

# NI myRIO Project Essentials Guide

Ed Doering





# NI myRIO Project Essentials Guide

Ed Doering  
Electrical and Computer Engineering Department  
Rose-Hulman Institute of Technology

Printed February 9, 2016. Download the latest version at <http://www.ni.com/myrio/project-guide>.

©2013 National Technology and Science Press.

All rights reserved. Neither this book, nor any portion of it, may be copied or reproduced in any form or by any means without written permission of the publisher.

NTS Press respects the intellectual property of others, and we ask our readers to do the same. This book is protected by copyright and other intellectual property laws. Where the software referred to in this book may be used to reproduce software or other materials belonging to others, you should use such software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

LabVIEW and National Instruments are trademarks of National Instruments.

All other trademarks or product names are the property of their respective owners.

**Additional Disclaimers:** The reader assumes all risk of use of this book and of all information, theories, and programs contained or described in it. This book may contain technical inaccuracies, typographical errors, other errors and omissions, and out-of-date information. Neither the author nor the publisher assumes any responsibility or liability for any errors or omissions of any kind, to update any information, or for any infringement of any patent or other intellectual property right.

Neither the author nor the publisher makes any warranties of any kind, including without limitation any warranty as to the sufficiency of the book or of any information, theories, or programs contained or described in it, and any warranty that use of any information, theories, or programs contained or described in the book will not infringe any patent or other intellectual property right. THIS BOOK IS PROVIDED "AS IS." ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, ARE DISCLAIMED.

No right or license is granted by publisher or author under any patent or other intellectual property right, expressly, or by implication or estoppel.

IN NO EVENT SHALL THE PUBLISHER OR THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, COVER, ECONOMIC, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS BOOK OR ANY INFORMATION, THEORIES, OR PROGRAMS CONTAINED OR DESCRIBED IN IT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND EVEN IF CAUSED OR CONTRIBUTED TO BY THE NEGLIGENCE OF THE PUBLISHER, THE AUTHOR, OR OTHERS. Applicable law may not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

# Contents

1	Introduction	1
I	NI myRIO Starter Kit	3
2	Discrete LED	5
3	Seven-Segment LED Display	9
4	Pushbutton Switch	13
5	DIP Switches	17
6	Relay	21
7	Potentiometer	27
8	Thermistor	31
9	Photocell	35
10	Electret Microphone	39
11	Buzzer/Speaker	43
12	Motor	47
13	Rotary Encoder	51
14	Photointerrupter	55
15	Hall-Effect Sensor	59

---

16 Piezoelectric-Effect Sensor	63
<b>II NI myRIO Mechatronics Kit</b>	<b>67</b>
17 Servo	69
18 Motor Adapter for NI myRIO and Geared Motor	75
19 IR Range Finder	81
20 Sonic Range Finder	85
21 Accelerometer	89
22 Gyroscope	95
23 Compass	101
24 Ambient Light Sensor	107
<b>III NI myRIO Embedded Systems Kit</b>	<b>111</b>
25 Keypad	113
26 LCD Character Display – UART Interface	117
27 LCD Character Display – SPI Interface	121
28 LCD Character Display – I <sup>2</sup> C-bus Interface	125
29 LED Matrix	129
30 Serial EEPROM	135
31 Bluetooth Module	141
32 Digital Potentiometer	147
33 Temperature Sensor	151
34 MEMS Microphone	157

---

<b>IV Additional Devices</b>	<b>163</b>
35 H-Bridge and Geared Motor	165
36 USB Flash Drive	171
37 Webcam	175
38 GPS Receiver	179
39 RFID Reader	183
<b>V Integrated Project Ideas</b>	<b>189</b>
40 Handheld Meter	191
41 Wireless Sensor	193
42 Data Logger	195
43 NTP Clock	197
44 Steer By Wire	199
45 Digital Thermometer	201
46 3-D Color Controller	203
47 QR Code Scanner	205
48 RPN Calculator	207
49 Hotel Room Safe Controller	209
50 Tachometer	211
51 Scanning Sensor	213
52 On-Off Control System	215
53 Security Camera	217
54 Compass with Tilt Correction	219

---

55	Guitar Tuner	221
56	Music Maker	223
57	Digital Bubble Level	225
58	Weather Station	227
59	EEPROM Programmer	229
<b>VI</b>	<b>Appendices</b>	<b>231</b>
A	MPX and MSP Connector Diagrams	233
B	Build a Stand-Alone Application	235
C	NI myRIO Starter Kit Data Sheets	237
D	Video Tutorial Links	239



# 1 Introduction

Welcome to the *NI myRIO Project Essentials Guide*, and welcome to the exciting world of NI myRIO! This book serves as your guide to interfacing your NI myRIO to the wide variety of sensors, actuators, and displays that you will need for your projects. Each chapter concentrates on a specific component or device using a mixture of text and video to guide you through the learning process necessary to successfully integrate the component or device into your system.

For example, consider the photocell of Chapter 9 on page 35, one of the many parts included with the NI myRIO Starter Kit. You'll get immediate hands-on experience with the photocell as you build a simple demonstration circuit, connect it to one of the NI myRIO expansion ports, and run a provided LabVIEW program to confirm that the photocell works properly. After that, study a video tutorial that explains photocell operating principles and learn how to design a suitable interface circuit. Next, try your hand with the suggested modifications to the demonstration LabVIEW code to deepen your understanding of the photocell behavior and LabVIEW programming techniques. At last you will be ready to tackle the suggested systems integration project in which you combine the photocell with other devices.

This book is intended for students at the junior or senior year, especially those students engaged in capstone projects or research. A background in electrical and computer engineering is ideal, but

students pursuing other disciplines will find the level of tutorial detail to be more than adequate.

This document is fully hyperlinked for section and figure references, and all video links are live hyperlinks. Open the PDF version of this document for the most efficient way to access all of the links; click a video hyperlink to automatically launch the video in your browser. Within the PDF, use `ALT+leftarrow` to navigate back to a starting point.

The book is organized into six major parts: Parts I through III detail the components and devices included in the NI myRIO Starter Kit, Mechatronics Kit, and Embedded Systems Kit; Part IV includes additional chapters (USB flash drive, webcam, GPS receiver, and RFID reader); Part V provides numerous ideas for complete projects that integrate two or more devices and components; and Part VI (Appendices) includes NI myRIO connector diagrams, a tutorial on building a stand-alone application (power-on your NI myRIO and the application runs immediately), collected data sheet links, and links to all video tutorials.

## 1.1 Acknowledgements

This book represents the contributions of many, and I gratefully acknowledge the support of these individuals at National Instruments: Gretchen Edelman for her technical support and encouragement regarding the book's content and organization; Margaret Barrett for inviting me to participate in the NI myRIO Beta Program; Sam Kristoff for timely advice regarding LabVIEW programming techniques; Charles Yarbrough for discussions of myRIO hardware design; and Andrew Watchorn for his ongoing support of university academic programs. I am also grateful to Tom Robbins at NTS Press for his editorial support on this project.

Ed Doering  
Department of Electrical and Computer Engineering  
Rose-Hulman Institute of Technology  
Terre Haute, IN 47803

`doering@rose-hulman.edu`

**Part I**

**NI myRIO Starter Kit**



# 2 Discrete LED

LEDs, or light-emitting diodes, provide simple yet essential visual indicators for system status and error conditions. Figure 2.1 shows some typical LEDs; the clear LEDs on the right are included in the NI myRIO Starter Kit.



Figure 2.1: Typical LEDs including standard red and green, high-efficiency in various colors, and RGB.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the essential concepts related to LEDs:
  - (a) An LED is a diode that permits only one-way current,

- (b) The LED forward-bias voltage drop varies with color (wavelength),
  - (c) The interface circuit design depends on knowledge of the DIO output resistance and source voltage,
  - (d) LEDs may be direct-connected to the DIO under some circumstances, and
2. Select a suitable current-limiting resistor (or no resistor) based on the LED type.

## 2.1 Component Demonstration

Follow these steps to demonstrate correct operation of the discrete LED component.

**Select these parts from the NI myRIO Starter Kit:**

- Two  $100\ \Omega$  resistors connected in series or two  $470\ \Omega$  resistors connected in parallel
- Two-terminal LED
- Breadboard
- Jumper wires, M-F ( $2\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 2.2 on the following page. The discrete LED interface circuit requires two connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. Anode  $\rightarrow$  B/+3.3V (pin 33)
2. LED control  $\rightarrow$  B/DIO0 (pin 11)

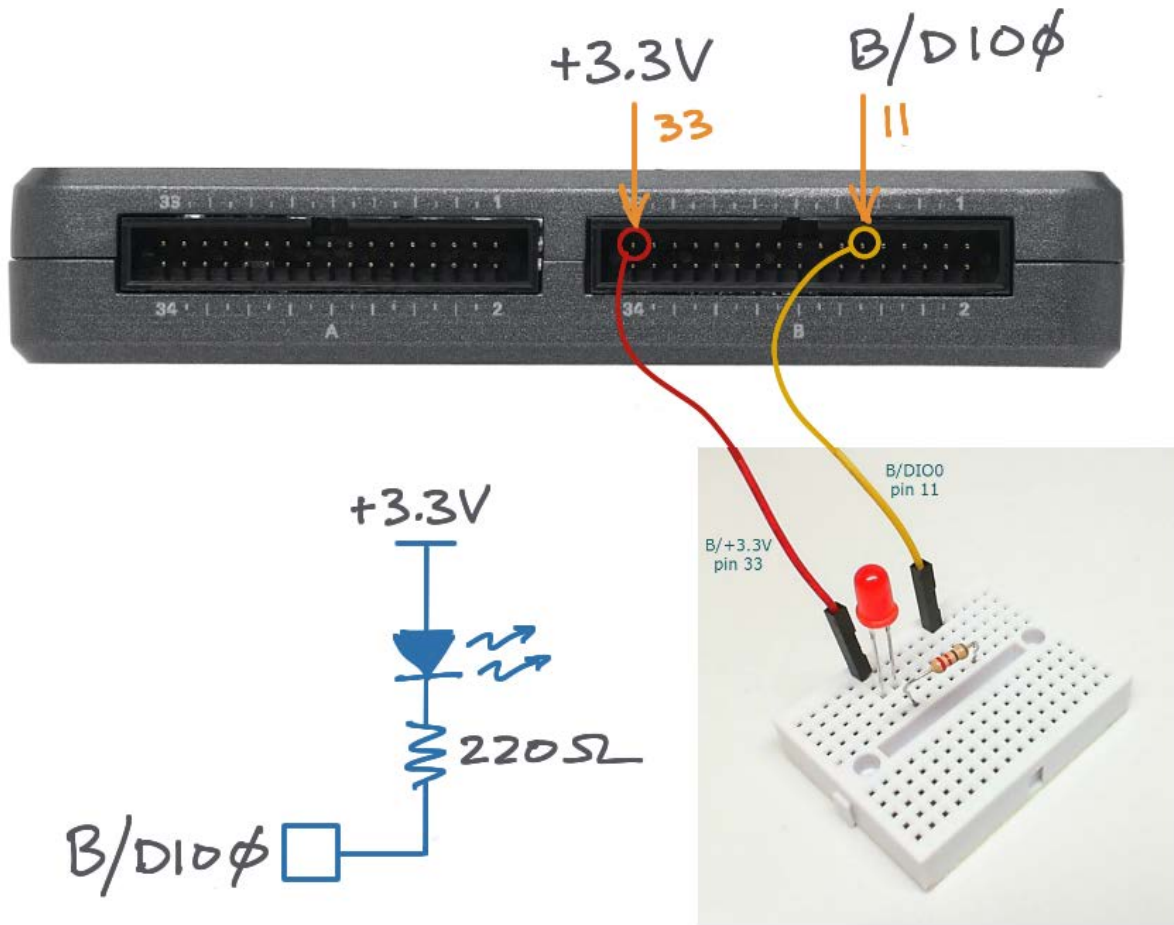


Figure 2.2: Demonstration circuit for discrete LED: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project Discrete LED demo.lvproj contained in the subfolder Discrete LED demo,

- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window

showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The schematic diagram indicates a single  $220\ \Omega$  current-limiting resistor; you may instead use two series-connected  $100\ \Omega$  resistors or two parallel-connected  $470\ \Omega$  resistors from the NI myRIO Starter Kit. Also, all of the LEDs in the Starter Kit have clear plastic lenses with a wide variety of colors. You may wish to try multiple LEDs to investigate your color options.

Your discrete LED should be blinking on and off in synchronism with the front-panel indicator digital output state. Click the enable blinker front-panel button to disable blinking and to enable the digital level button; click this button to manually set the digital output state either high or low. Because this interface circuit is the *sinking current* form (explained in the next section), the LED is active when the digital output is in the *low* state, i.e., this is an active-low LED interface circuit.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct LED orientation — the diode conducts current in one direction only; remove the LED and reinsert it in the opposite direction, and
- Correct resistor value — use an ohmmeter to verify that the resistance is near  $220\ \Omega$ .

## 2.2 Interface Theory

**Interface circuit:** Study the video *Discrete LED Interfacing Theory* ([youtu.be/9-R1GpVgFW0](https://youtu.be/9-R1GpVgFW0), 6:55) to learn the basics of LEDs and the two types of interface circuits (current-sinking and current-sourcing). Also study *LED Current Management* ([youtu.be/JW-19uXrWNU](https://youtu.be/JW-19uXrWNU), 15:06) to learn about the voltage-current characteristics of the various types of diodes included in the NI myRIO Starter Kit (standard, superbright, and RGB), to learn principles of operation of the LED interface circuit including the current-sinking and current-sourcing forms, and to learn how to choose the size of the current-limiting resistor.

**LabVIEW programming:** Study the video *Digital Output Express VI* ([youtu.be/Y8mKdsMAqrU](https://youtu.be/Y8mKdsMAqrU), 2:21) to learn how to access all of the available digital outputs with the NI myRIO Digital Output Express VI, including single output, multiple outputs, and connector type.

## 2.3 Basic Modifications

Study the video *LED Demo Walk-Through* ([youtu.be/SHJ-vu4jorU](https://youtu.be/SHJ-vu4jorU), 2:03) to learn the design principles of Discrete LED demo, and then try making these modifications to the interface circuit and to `Main.vi`:

1. Add a front-panel control to adjust the blink frequency specified in Hertz; at what frequency does the blinking become imperceptible?
2. Blink two adjacent LEDs to simulate a railroad crossing signal.
3. Blink the green and blue LEDs of the RGB LED using the same LabVIEW code as the railroad crossing signal; refer to Figure 2.3 on the following page for the RGB LED pinout diagram. Use the current-sourcing interface circuit.

4. Create an LED variable-intensity dimmer with the PWM Express VI to create a pulse-width modulated digital output. Set the Frequency to the default constant value of 1000 Hz and create a pointer slide control to adjust the duty cycle; select the logarithmic mapping option for the control, too. Add some code to account for the active-low current-sinking LED interface (bonus points with a Boolean control to select between current-sinking and current-sourcing interfaces).

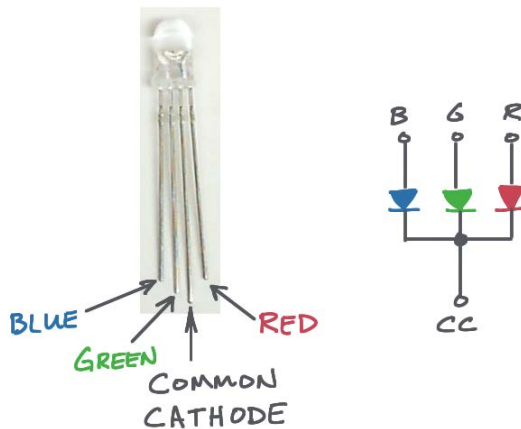


Figure 2.3: RGB LED pins and schematic diagram.

## 2.4 Integrated Project Ideas

Now that you know how to use the discrete LED consider integrating it with other devices to create a complete system, for example:

- *3-D Color Controller* (46)
- *Digital Bubble Level* (57)
- *NTP Clock* (43)

## 2.5 For More Information

- *LED Mixed Bag (5mm)* by SparkFun ~ Need more LEDs? The LED Mixed Bag from SparkFun offers the same type as those in the NI myRIO Starter Kit; also search SparkFun for many other sizes and types of LEDs:

<http://www.sparkfun.com/products/9881>

- *Engineering Thursday: LED Light Boxes* by SparkFun ~ Glowing multi-color boxes as household art:

<http://www.sparkfun.com/news/1210>

- *Use LEDs as photodiodes* by EDN ~ LEDs can detect light, too, making them an interesting type of photosensor:

<http://www.edn.com/design/led/4363842/>

Use-LEDs-as-photodiodes

- *LED Lighting Applications* by OSRAM Opto Semiconductors ~ LEDs are everywhere these days, including outdoor street lighting, architectural illumination, downlights (i.e., ceiling lights), flashlights, and greenhouses:

<http://ledlight.osram-os.com/applications/>



# 3 Seven-Segment LED Display

Displays based on seven LED segments arranged in an “8” pattern provide a simple means to display numbers 0 to 9 and some letters of the alphabet. Figure 3.1 shows the NI myRIO Starter Kit seven-segment display.

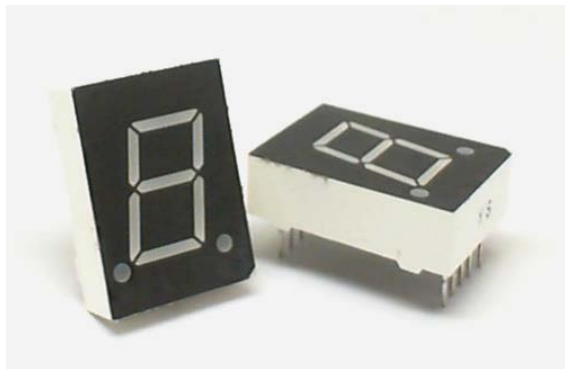


Figure 3.1: NI myRIO Starter Kit seven-segment display.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the array of LEDs wired with a common-anode connection,
2. Design the interface circuit by applying knowledge of the DIO output resistance, source voltage, LED voltage-current characteristic, and
3. Recognize that blue LEDs may be direct-connected to the NI myRIO DIO without

current-limiting resistors.

## 3.1 Component Demonstration

Follow these steps to demonstrate correct operation of the seven-segment LED component.

**Select these parts:**

- Seven-segment LED display, <http://www.sparkfun.com/datasheets/Components/LED/YSD-160AB3C-8.pdf>
- Breadboard
- Jumper wires, M-F (9×)

**Build the interface circuit:** Refer to the pin diagram and recommended breadboard layout shown in Figure 3.2 on page 11.

**TIP:** Use the resistor color code for the DIO wire colors, e.g., black (0) for B/DIO0, brown (1) for B/DIO1, and so on.

The interface circuit requires nine connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. Common anode (CA) → B/+3.3V (pin 33)
2. Segment *a* → B/DIO0 (pin 11)
3. Segment *b* → B/DIO1 (pin 13)
4. Segment *c* → B/DIO2 (pin 15)
5. Segment *d* → B/DIO3 (pin 17)
6. Segment *e* → B/DIO4 (pin 19)

7. Segment *f* → B/DIO5 (pin 21)
8. Segment *g* → B/DIO6 (pin 23)
9. Decimal point → B/DIO7 (pin 25)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Seven-Segment LED demo.lvproj` contained in the subfolder `Seven-Segment LED demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Toggle the eight front-panel switches to activate and deactivate each segment *a* through *g* as well as the decimal point; refer again to Figure 3.2 on the facing page to see the standard labeling scheme for the segments. Activating the front-panel switch should cause the corresponding segment to light.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,

- Black “Run” button on the toolbar signifying that the VI is in run mode,
- Correct wiring — ensure that your wiring order is correct and that you have started at B/DIO0 (pin 11), and
- Ensure that the seven-segment LED common anode (CA) terminal connects to the power supply B/+3.3V (pin 33).

### 3.2 Interface Theory

**Interface circuit:** Each of the seven line segments as well as the decimal point is an individual LED, each with its own anode and cathode. To conserve the number of electrical connections on the display all of the eight anodes are tied together and brought out as the “common anode” pin. Study the video *Seven-Segment LED Interfacing Theory* ([youtu.be/P0ER0VXvfSw](https://youtu.be/P0ER0VXvfSw), 4:11) to learn about the voltage-current characteristics of the individual LED segment, why the segment controls are active-low, and why no current-limiting resistors are required for this particular device.

**LabVIEW programming:** Study the video *Digital Output Express VI* ([youtu.be/Y8mKdsMAqrU](https://youtu.be/Y8mKdsMAqrU), 2:21) to learn how to access all of the available digital outputs with the NI myRIO Digital Output Express VI, including single output, multiple outputs, and choice of connector. In addition, study the video *Digital Output Low-Level VIs* ([youtu.be/WvnInG3ffqY](https://youtu.be/WvnInG3ffqY), 4:53) to learn how to access and use the lower-level code created by the Express VI to connect Boolean arrays directly to the digital outputs, i.e., as a bus.

### 3.3 Basic Modifications

Study the video *Seven-Segment LED Demo Walk-Through* ([youtu.be/ejyOo\\_k9K10](https://youtu.be/ejyOo_k9K10), 2:03) to learn the design principles of 7-segment

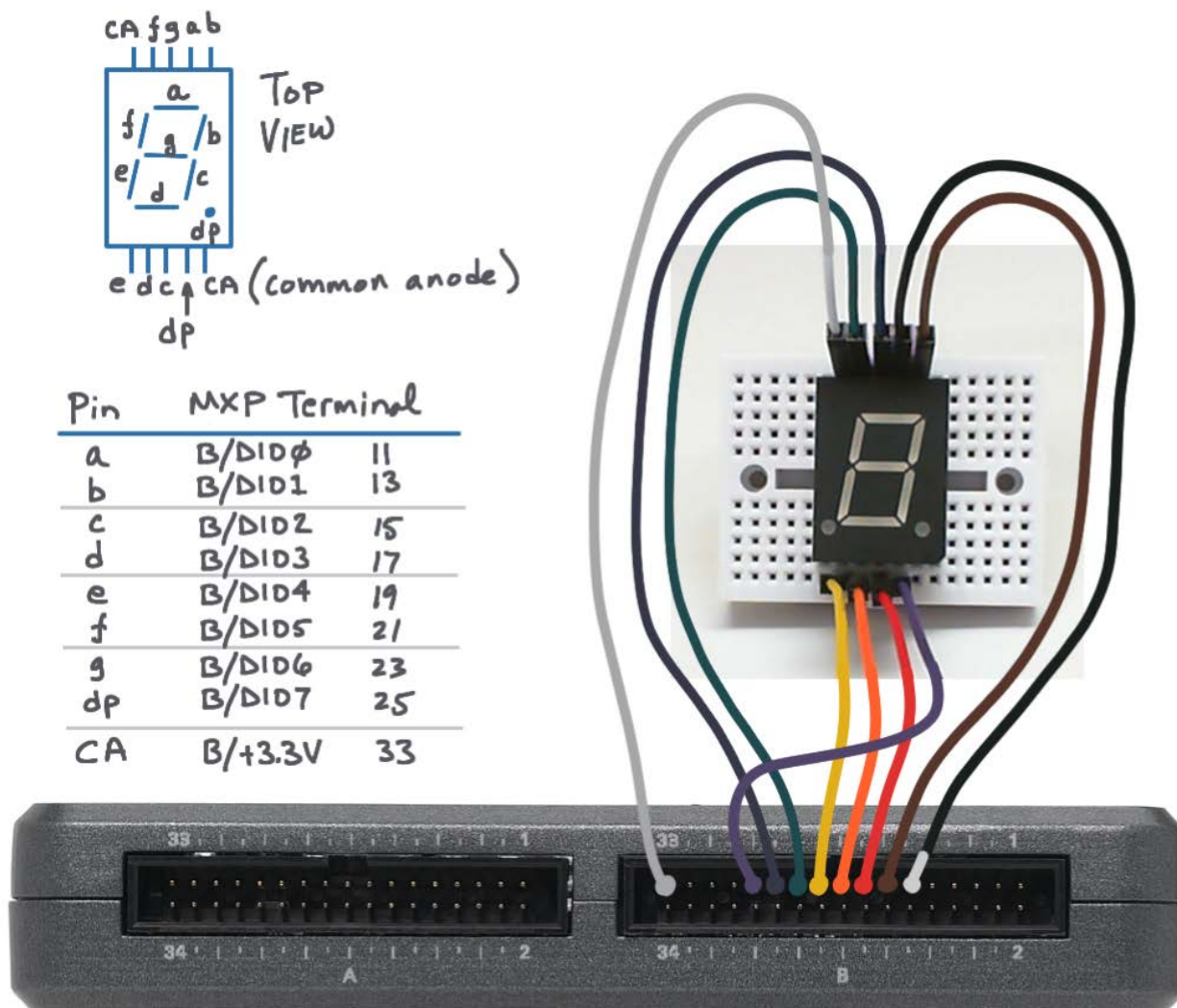


Figure 3.2: Demonstration circuit for seven-segment display: pin diagram, recommended breadboard layout, and connection to eight digital I/O terminals on NI myRIO MXP Connector B.

LED demo.lvproj, and then try making these modifications to Main.vi:

1. Maintain the same behavior, but convert the Digital Output Express VI to its underlying code, and then connect the array-style front-panel control directly to the Write VI from the NI myRIO Advanced I/O → Digital I/O subpalette. Also create a front-panel control to make user-selectable digital I/O channels.
2. Display a numerical (integer) front-panel control value as its corresponding pattern on the seven-segment display; a case structure with Boolean array constants works nicely here (see <http://cnx.org/content/m14766/latest/?collection=col10440>). Display the values 0 to 9 and a dash for values greater than 9 (bonus points for a hexadecimal display to include values A to F).
3. Create a rotating chase sequence in which a single active segment appears to move around the periphery of the display. Make the speed adjustable, and also include a control to reverse the direction of rotation. Consider a single Boolean array constant connected to Rotate 1D Array in the Programming → Array subpalette. TIP: Use Quick Drop (Ctrl+Space) to search for a programming element by name.

### 3.4 Integrated Project Ideas

Now that you know how to use the seven-segment display consider integrating it with other devices to create a complete system, for example:

- *NTP Clock* (43)

### 3.5 For More Information

- *7-Segment Display - LED (Blue)* by SparkFun ~ Need more display digits? This is the same (or

very similar) product:

<http://www.sparkfun.com/products/9191>

- *World's Simplest Font* by Twyman Enterprises ~ TrueType font for seven-segment displays; use this font and your favorite wordprocessor to quickly translate your text phrases into suitable segment patterns:

<http://www.twyman.org.uk/Fonts>

- *Nixie Tubes and Projects* by Neonixie ~ Before seven segment displays you would have used *nixie tubes*; see [http://en.m.wikipedia.org/wiki/Nixie\\_tube](http://en.m.wikipedia.org/wiki/Nixie_tube) for the origin of "nixie." You can buy nixie tubes and related projects here:

<http://neonixie.com/>

# 4 Pushbutton Switch

Pushbutton switches – also called momentary contact switches – serve as basic user-interface devices as well as simple sensors, e.g., bump sensors. Figure 4.1 pictures the pushbutton integrated with the rotary encoder covered in a later chapter.



Figure 4.1: NI myRIO Starter Kit pushbutton switch integrated with rotary encoder.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss the essential concepts associated with a pushbutton switch:
  - (a) Pushbutton switch appears as a short circuit when pressed, otherwise as an open circuit,
  - (b) Interface circuit to the digital input relies on the DIO internal pull resistors to eliminate the need for additional components (pull-up on MXP Connectors A and B, pull-down on MSP Connector C),
  - (c) Block diagram views the switch as a Boolean (two-level) signal that is either active-high or active-low depending on the type of pull resistor, and
2. Apply software-based edge detection to convert a pushbutton press into a trigger event.

## 4.1 Component Demonstration

Follow these steps to demonstrate correct operation of the pushbutton switch.

**Select these parts from the NI myRIO Starter Kit:**

- Pushbutton switch (integrated with rotary encoder), <http://www.sparkfun.com/datasheets/Components/TW-700198.pdf>
- Breadboard
- Jumper wires, M-F (2×)

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 4.2 on the facing page.

**TIP:** Flatten the two tabs on either side of the rotary encoder so that it sits flush on the breadboard surface.

The pushbutton switch interface circuit requires two connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. Pushbutton Terminal 1 → B/DIO0 (pin 11)
2. Pushbutton Terminal 2 → B/GND (pin 12)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Pushbutton demo.lvproj` contained in the subfolder `Pushbutton demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the input state of three DIOs, one on each connector. The states of the Connector A and B DIOs should be high due to the internal pull-up resistors, and the Connector C DIO should be low because of the internal pull-down resistor. Press the pushbutton and you should

see the B/DIO0 state indicator change to low; release the pushbutton and the state should go high again.

The demo VI also counts pushbutton presses detected on B/DIO0. Stop and restart the VI to clear the counter value.

Disconnect Pushbutton Terminal 1 and reconnect to A/DIO0 (pin 11). Confirm that the A/DIO0 state indicator changes in response to pushbutton presses.

Disconnect Pushbutton Terminal 1 and reconnect to C/DIO0 (pin 11). Does the C/DIO0 state indicator change in response to a pushbutton press? Can you explain the behavior you observe?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 4.2 Interface Theory

**Interface circuit:** The pushbutton switch normally appears as an open circuit and becomes a short circuit when pressed. The pushbutton may be connected directly to the digital input without any additional components because of the internal pull resistors on the NI myRIO DIO lines.

Study the video *Pushbutton Interfacing Theory* ([youtu.be/e7UcL5Ycpho](https://youtu.be/e7UcL5Ycpho), 4:24) to learn about the DIO pull resistors and how to properly connect the pushbutton switch for pull-up resistors (MXP Connectors A and B) and pull-down resistors (MSP Connector C). Study *Detect a Switch Signal Transition* ([youtu.be/GYBmRJ\\_qMrE](https://youtu.be/GYBmRJ_qMrE), 4:42) to learn

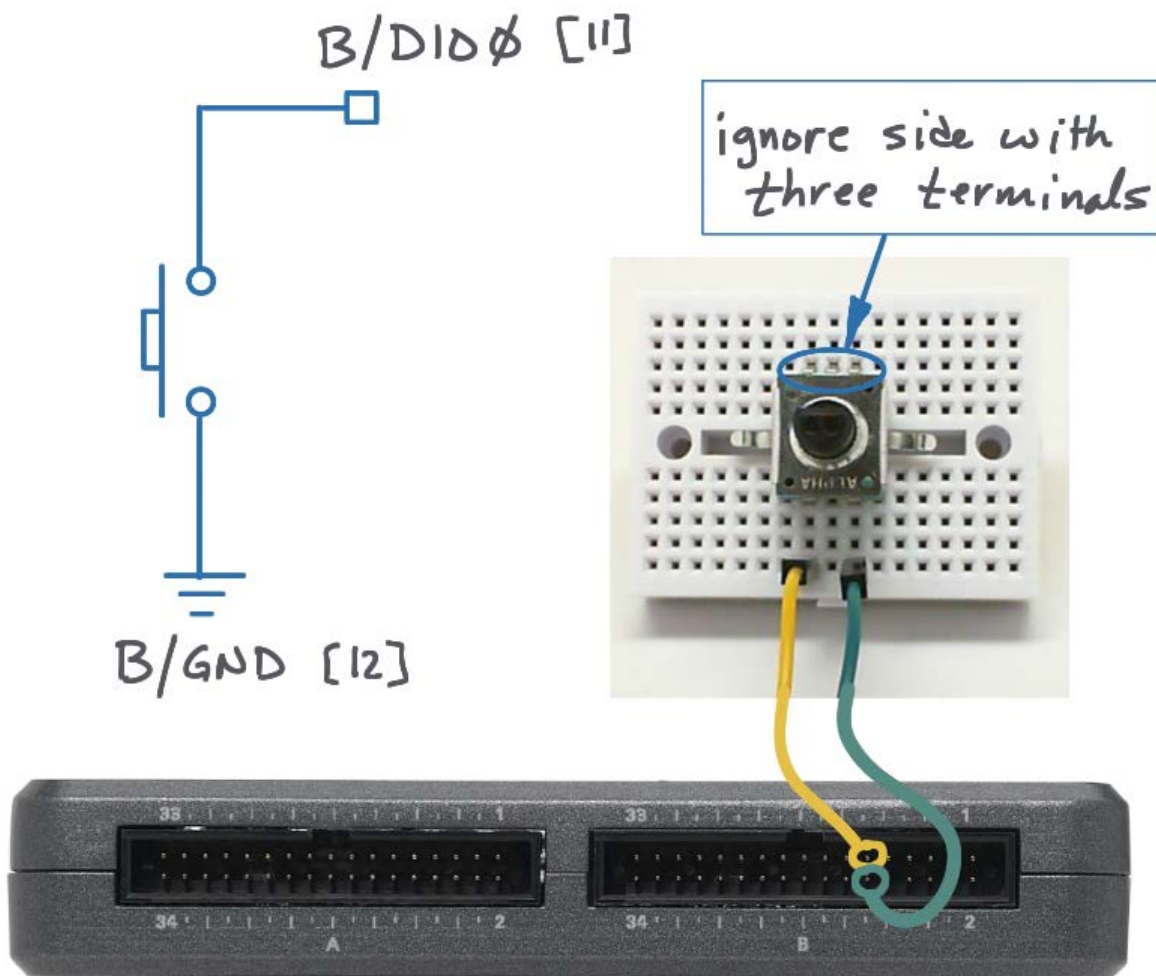


Figure 4.2: Demonstration circuit for pushbutton switch: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

how to detect a switching signal transition inside a software while-loop structure.

to learn how to use Digital Input Express VI to sense the state of the pushbutton.

**LabVIEW programming:** Study the video *Digital Input Express VI* ([youtu.be/litSwKgOmZA](https://youtu.be/litSwKgOmZA), 1:53)

### 4.3 Basic Modifications

Study the video *Pushbutton Demo Walk-Through* ([youtu.be/Xm1A4Cw2POU](https://youtu.be/Xm1A4Cw2POU), 3:16) to learn the design principles of `Pushbutton` demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add the onboard LED Express VI (myRIO | Onboard subpalette) as an indicator on the output of the edge detector (the AND gate). Confirm that the LED flashes briefly when you press the pushbutton.
2. Experiment with different values of loop speed by adjusting the value of `Wait (ms)`; you may find it more convenient to change the constant to a front-panel control. At what value does the VI introduce noticeable delay responding to the pushbutton press?
3. Adjust the `presses` counter behavior to count pushbutton *releases* instead of presses.
4. Adjust the `presses` counter behavior to count both pushbutton presses and releases. HINT: Try a single exclusive-OR gate from the `Programming | Boolean` subpalette.
5. Modify the loop termination condition so that the VI runs only while the pushbutton is pressed.

NOTE: The SPDT (single-pole single-throw) slide switch included with the NI myRIO Starter Kit (see Figure 4.3) can connect to myRIO in the same way as the pushbutton switch. Simply use the middle terminal and either end terminal in place of the pushbutton terminals. Use the slide switch anytime you need to maintain the DIO at a specific level, for example, as a mode setting for your NI myRIO program.



Figure 4.3: SPDT (single-pole single-throw) switch included with the NI myRIO Starter Kit.

- *Applications by Knitter-Switch* ~ Learn about the myriad practical applications for switches as well as the wide variety of switch types:  
[http://www.knitter-switch.com/p\\_applications.php](http://www.knitter-switch.com/p_applications.php)

### 4.4 For More Information

- *Mini Push Button Switch* by SparkFun ~ A handy switch for circuit boards:  
<http://www.sparkfun.com/products/97>



# 5 DIP Switches

DIP switches bundle multiple SPST switches together into a single component; “DIP” stands for “dual in-line package,” the standard IC package style that is breadboard compatible, and “SPST” means “single pole, single throw,” the simplest possible switch type. Figure 5.1 pictures two popular DIP switch styles: a standard DIP switch containing eight SPST switches and a 16-position rotary DIP switch that manipulates the open-and-closed states of four SPST switches in a binary sequence.

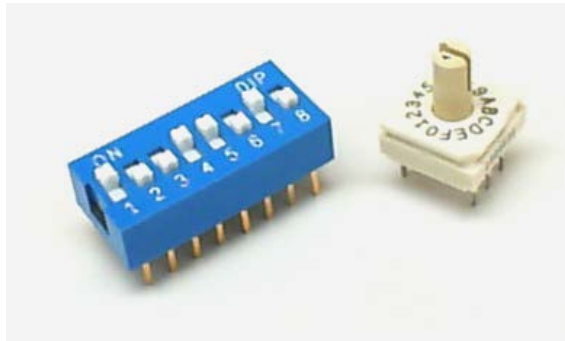


Figure 5.1: NI myRIO Starter Kit DIP switches: DIP switch (blue) and 16-position rotary DIP switch.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the following concepts related to switches and the NI myRIO interface:

- (a) DIP switch bundles  $N$  SPST switches into a single component with each switch appearing as a short circuit in one position and as an open circuit in the other,
  - (b)  $2^N$ -position rotary switch bundles  $N$  SPST switches into a single component; rotating the dial create a binary sequence of open-closed switch states,
2. Interface a switch on any of the NI myRIO connectors without using additional components by using the the DIO internal pull resistors (pull-up on MXP connectors, pull-down on MSP connector), and
  3. Interpret the combined switch open-closed patterns as an integer numerical value, binary array, and individual bit fields.

## 5.1 Component Demonstration

Follow these steps to demonstrate correct operation of the DIP switches.

### Select these parts from the NI myRIO Starter Kit:

- DIP switch, [http://www.resonswitch.com/p\\_rs\\_rsr.htm](http://www.resonswitch.com/p_rs_rsr.htm)
- Rotary DIP switch, <http://www.mantech.co.za/datasheets/products/ERD1-5.pdf>
- Breadboard
- Jumper wires, M-F (14×)
- Small screwdriver

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 5.2 on the facing page. The interface circuit for the DIP switches requires five connections to NI myRIO MXP Connector A and nine connections to Connector B (see Figure A.1 on page 233):

1. DIP Switch 8 → B/DIO0 (pin 11)
2. DIP Switch 7 → B/DIO1 (pin 13)
3. DIP Switch 7 → B/DIO2 (pin 15)
4. DIP Switch 5 → B/DIO3 (pin 17)
5. DIP Switch 4 → B/DIO4 (pin 19)
6. DIP Switch 3 → B/DIO5 (pin 21)
7. DIP Switch 2 → B/DIO6 (pin 23)
8. DIP Switch 1 → B/DIO7 (pin 25)
9. DIP Switch common → B/GND (pin 8)
10. Rotary DIP 1 → A/DIO0 (pin 11)
11. Rotary DIP 2 → A/DIO1 (pin 13)
12. Rotary DIP 4 → A/DIO2 (pin 15)
13. Rotary DIP 8 → A/DIO3 (pin 17)
14. Rotary DIP C (common) → A/GND (pin 20)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `DIP Switches demo.lvproj` contained in the subfolder `DIP Switches demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the individual switch states of the two DIP switches. An open switch appears as a high state because both MXP connectors A and B include pull-up resistors on each DIO.

With the rotary DIP switch dial at Position 0 all A/DIO state indicators should be active; turn the dial counter-clockwise one click to Position F and all indicator should be dark. Try clicking through the remaining positions and observe the binary sequence, remembering that the switches appear active-low.

Try each of the eight switches on the DIP switch and confirm that you can individually activate the indicators for A/DIO state. Is the switch open or closed in the “up” position?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using both Connectors A and B and that you have the correct pin connections, and
- Connecting wires link all eight of the lower pins of the DIP switch to ground.

## 5.2 Interface Theory

**Interface circuit:** Each SPST switch in the standard DIP switch appears either as an open circuit or as a short circuit depending on the switch position “up”

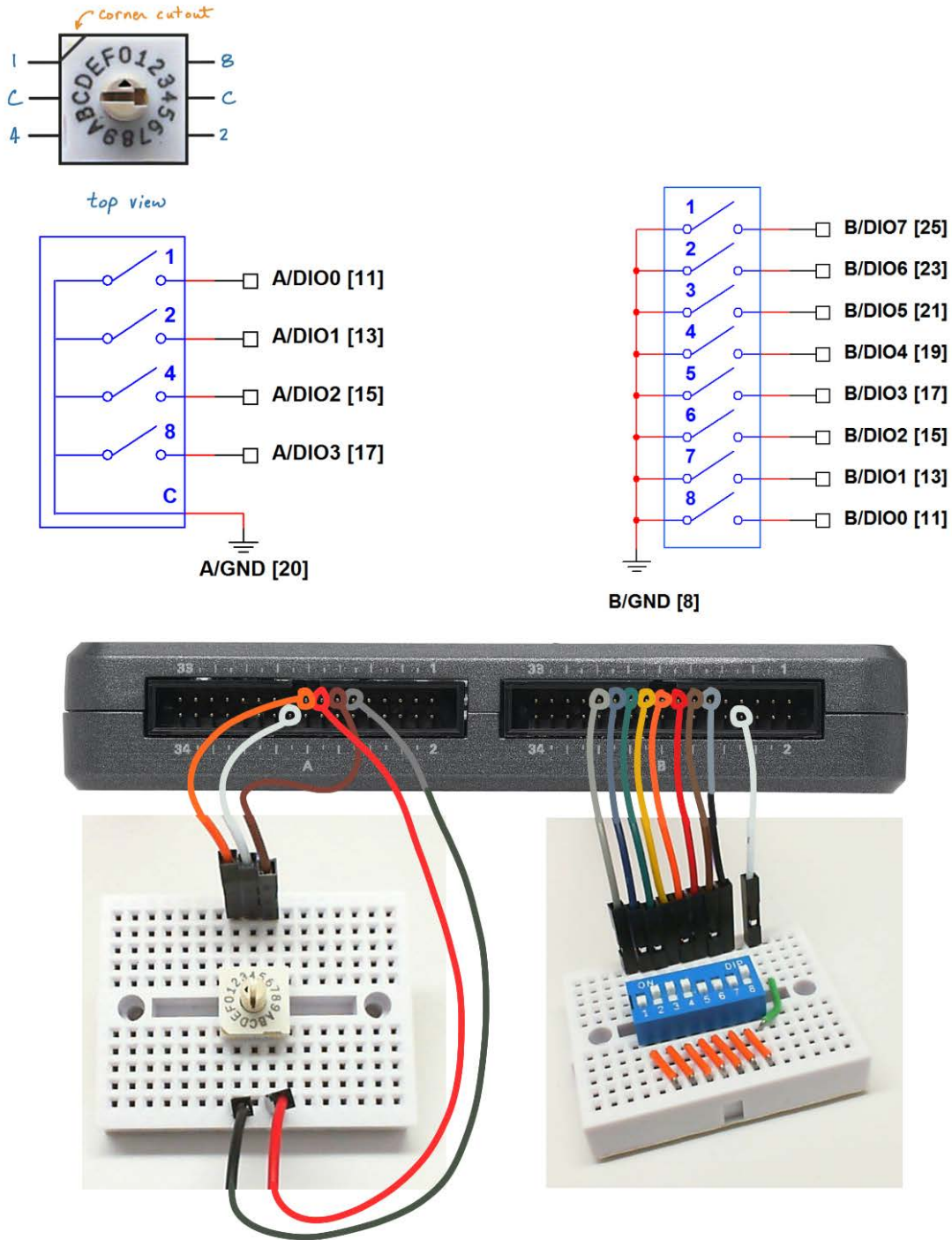


Figure 5.2: Demonstration circuit for DIP switches: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connectors A and B.

or “down.” The 16-position rotary switch opens all four SPST switches at Position 0 and then applies an ascending binary sequence to switch closings as the dial rotates clockwise. The DIP switches may connect directly to the digital input without any additional components because of the internal pull resistors on the NI myRIO DIO lines.

Each DIP switch can be interpreted by software in a number of different ways, including: single integer numerical value, single Boolean array pattern, and groups of binary patterns or numerical values called bitfields.

Study the video *Pushbutton Interfacing Theory* ([youtu.be/e7UcL5Ycpho](https://youtu.be/e7UcL5Ycpho), 4:24) to learn about the DIO pull resistors and how to properly connect a single SPST switch for pull-up resistors (MXP Connectors A and B) and for pull-down resistors (MSP Connector C). Each of the SPST switches on the standard DIP switch must have one terminal tied either low to ground (for pull-up resistors) or high to the power supply (for pull-down resistors); it is customary to tie all of the terminals on a given side, but you could use any pattern that you like.

Study the video *DIP Switch Interfacing Theory* ([youtu.be/KNzEyRwcPIg](https://youtu.be/KNzEyRwcPIg), 7:16) to learn more about the DIP switch and the rotary DIP switch, especially various ways that you can interpret the switch patterns as meaningful information in software.

**LabVIEW programming:** Study the video *Run-Time Selectable I/O Channels* ([youtu.be/uJW7CaL6L5c](https://youtu.be/uJW7CaL6L5c), 1:54) to learn how to use the low-level Digital Input VIs to select the DIP switch connector pins from the front panel instead of editing the VI itself.

### 5.3 Basic Modifications

Study the video *DIP Switch Demo Walk-Through* ([youtu.be/ZMyYRSsQCac](https://youtu.be/ZMyYRSsQCac), 2:30) to learn the design principles of DIP Switches demo, and then try

making these modifications to the block diagram of `Main.vi`:

1. Display the DIP switch pattern as an 8-bit unsigned integer (UINT8 data type) using the right-most switch as the least-significant bit (LSB) and the “down” position as logical 0.
2. Display the DIP switch pattern as three distinct fields as follows: Field 1 (bits 2:0) = 3-bit integer, Field 2 (bits 6:3) = 4-bit integer, and Field 3 = single-bit Boolean.
3. Display the 16-position rotary DIP switch pattern as a 4-bit integer displayed in both decimal and in hexadecimal.
4. Move either or both of the DIP switches to the MSP Connector C (remember, these have pull-down resistors) and repeat some of the previous exercises. Use a single strategically-placed “NOT” gate to avoid changing other parts of the block diagram. Also remember to connect the DIP switch common terminal to C/+5V (pin 20).

### 5.4 For More Information

- *2-Wire Controlled Digital DIP Switch* by Maxim Integrated ~ An electronic replacement for mechanical DIP switches, the DS3904 contains microcontroller-controlled *nonvolatile variable resistors* that offer smaller footprint, higher reliability, and lower cost:

<http://www.maximintegrated.com/app-notes/index.mvp/id/238>

# 6 Relay

Low-power digital outputs lack the necessary current drive to operate motors, lights, and other high-current appliances. Relays bridge the power gap using a relatively low-power magnetic coil to control a switch designed to carry large currents. Figure 6.1 shows the NI myRIO Starter Kit SPDT relay.



Figure 6.1: NI myRIO Starter Kit relay.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss the relay principles of operation,

2. Size the interface transistor to accommodate the relay coil current,
3. Explain why the rectifier is necessary to suppress back-emf voltage spiking when the transistor turns on and off,
4. Design the interface circuit to work with DIOs that have internal pull resistors, and
5. Ensure that the relay remains off during NI myRIO power-on and reset.

## 6.1 Component Demonstration

Follow these steps to demonstrate correct operation of the relay.

**Select these parts from the NI myRIO Starter Kit:**

- Relay, <http://www.cndongya.com/pdf/relayjzc-11f.pdf>
- 1N4001 general-purpose rectifier, <http://www.vishay.com/docs/88503/1n4001.pdf>
- ZVP2110A p-channel enhancement-mode MOSFET, <http://www.diodes.com/datasheets/ZVP2110A.pdf>
- Breadboard
- Jumper wires, M-F (3×)

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 6.2 on page 23. Note that the three relay switch contacts do not sit on tenth-inch centers,

therefore this side of the relay must hang off the side of the breadboard. The interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. 5-volt power supply → B/+5V (pin 1)
2. Ground → B/GND (pin 6)
3. Relay control → B/DIO0 (pin 11)

#### Run the demonstration VI:

- **Download** <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Relay demo.lvproj` contained in the subfolder `Relay demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Your relay should be clicking in synchronism with the front-panel indicator DIO state. Click the cycle front-panel button to disable automatic mode and to enable the manual button; click this button to manually set the digital output state either high or low. The relay control signal is active-low, therefore the coil is energized when the DIO is low.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset

causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct transistor orientation — the transistor has a rounded shape on one side, and
- Correct rectifier orientation — when the rectifier is backwards the relay coil will never reach the voltage level necessary to turn on.

## 6.2 Interface Theory

**Interface circuit:** The relay contains an electromagnet coil that operates a spring-loaded switch. The coil current is approximately 100 mA, well beyond the current drive limits of the NI myRIO digital output. The interface circuit uses a p-channel enhancement FET as a switch to turn the coil current on and off and a rectifier to protect the transistor from large back-emf voltage when the transistor shuts off the coil current.

Study the video *Relay Interfacing Theory* ([youtu.be/jLFL9\\_EW1wI](https://youtu.be/jLFL9_EW1wI), 11:11) to learn more about the relay principles of operation and interface circuit design principles including: sizing the transistor for relay coil current, importance of the rectifier to deal with back-emf voltage spiking, and circuit topologies for DIOS with internal pull-up resistors (MXP connector) and internal pull-down resistors (MSP connector).

**LabVIEW programming:** Study the video *Run-Time Selectable I/O Channels* ([youtu.be/uJW7CaL6L5c](https://youtu.be/uJW7CaL6L5c), 1:54) to learn how to use the low-level Digital I/O VIs Open, Write, and Close to create a VI with run-time selectable DIO channels, i.e., select the

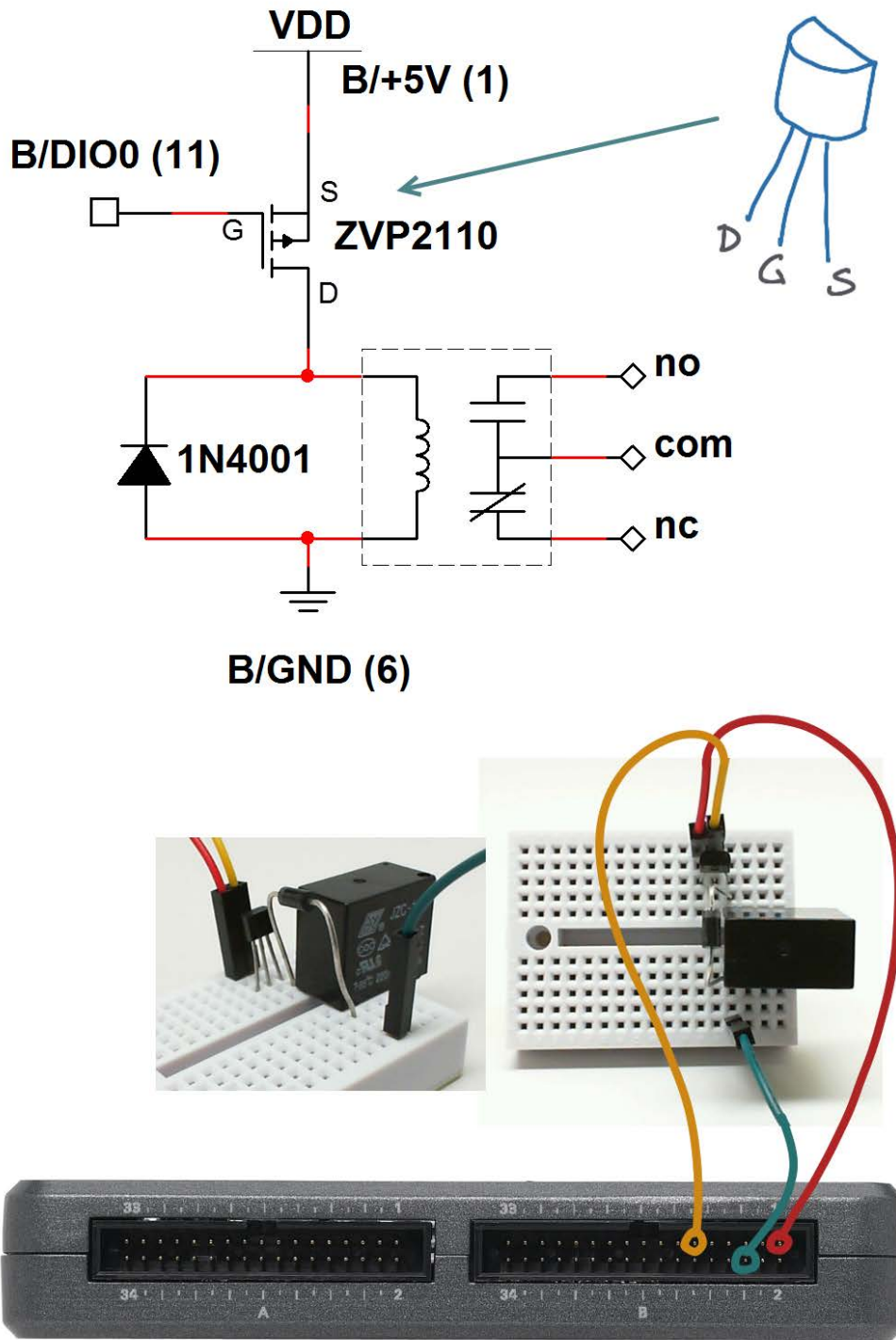


Figure 6.2: Demonstration circuit for relay: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

DIO channel directly on the front panel rather than editing the VI itself.

### 6.3 Basic Modifications

Study the video *Relay Demo Walk-Through* ([youtu.be/W2iukd8WVIA](https://youtu.be/W2iukd8WVIA), 3:30) to learn the design principles of `Relay demo.lvproj`, and then try making these modifications:

1. Add a front-panel control to adjust the cycle frequency specified in Hertz; at what frequency is the relay unable to keep up?
2. Blink two LEDs to simulate a railroad crossing signal; use the three relay contacts “normally open” (NO), “normally closed” (NC), and “common” (COM) as shown in Figure 6.2 on the previous page.
3. Build and test the interface circuit for a relay controlled from the MSP connector; see Figure 6.3 on the facing page. Select C/DIO7 (pin 18) as the relay control line on the myRIO DIO channel front-panel control. Power the circuit from the MSP connector with C/+5V (pin 20) and the digital ground C/DGND (pin 19).

### 6.4 Integrated Project Ideas

Now that you know how to use the relay consider integrating it with other devices to create a complete system, for example:

- *On-Off Control System* (52)
- *NTP Clock* (43)

### 6.5 For More Information

- *Using Relays (Tips & Tricks)* by Jumper One  
~ Learn how to reduce relay switching time and minimize relay current for battery-powered applications:

<http://jumperone.com/2011/10/using-relays>



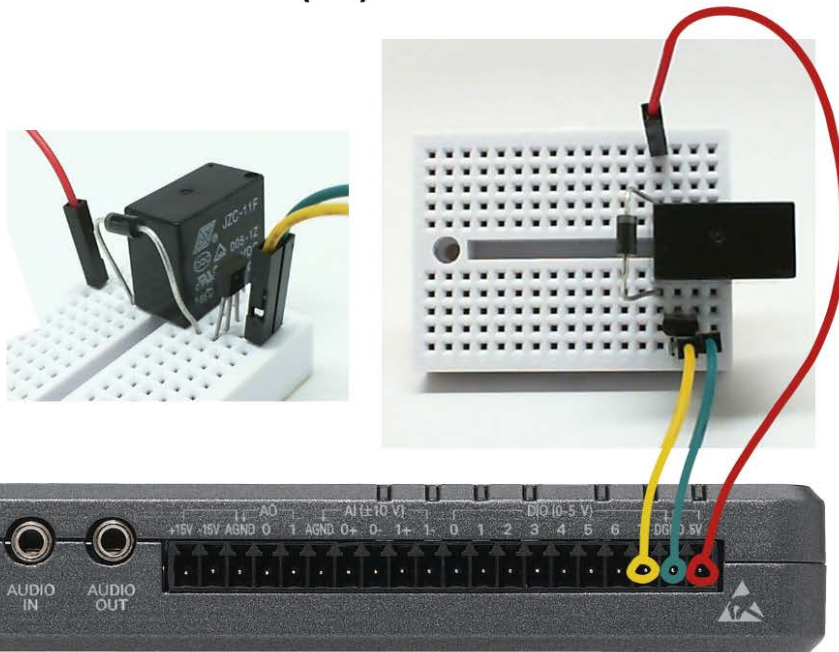
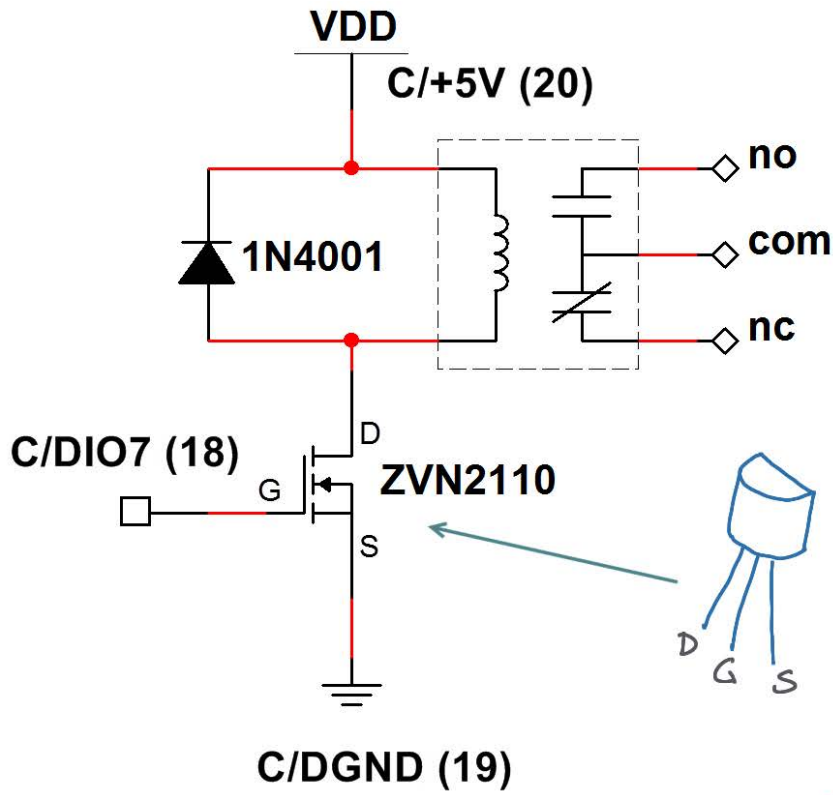


Figure 6.3: Relay interface circuit suitable for internal pull-down resistors on NI myRIO MSP Connector C.



# 7 Potentiometer

A *potentiometer* is a three-terminal variable resistor. When connected to a power supply to form a voltage divider a potentiometer acts as a proportional rotation sensor. Figure 7.1 pictures the NI myRIO Starter Kit potentiometer.



Figure 7.1: NI myRIO Starter Kit potentiometer.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss how the potentiometer can be used as either one variable resistor or two series-connected complementary variable resistors,

2. Connect a potentiometer as a voltage divider to produce a voltage proportional to rotation angle, and
3. Select the potentiometer resistance to minimize power consumption and to minimize loading effects.

## 7.1 Component Demonstration

Follow these steps to demonstrate correct operation of the potentiometer.

**Select these parts from the NI myRIO Starter Kit:**

- Potentiometer, 10 k $\Omega$ , <http://www.supertech.com.tw/electronic/resistors/potentiometers/PDF/rotary3/23/R0904N.pdf>
- Breadboard
- Jumper wires, M-F (3 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 7.2 on page 29.

**TIP:** Flatten the two tabs on either side of the potentiometer so that it sits flush on the breadboard surface.

The potentiometer interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. Pot Terminal 1 → B/GND (pin 16)
2. Pot Terminal 2 → B/AIO (pin 3)
3. Pot Terminal 3 → B/+5V (pin 1)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Potentiometer demo.lvproj` contained in the subfolder `Potentiometer demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the voltage on the analog input B/AIO. Turn the potentiometer dial and you should observe a corresponding change in the voltage sensed on the analog input. Because the potentiometer acts as an adjustable voltage divider between ground and the +5-volt supply, you should observe that a full rotation of the potentiometer dial from one extreme to the other causes the voltage to change from 0 to 5 volts.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 7.2 Interface Theory

**Interface circuit:** The potentiometer provides a fixed resistance between the two outer terminals, while the middle terminal connects to a movable contact point that effectively makes the potentiometer appear as two variable resistors. As one resistor increases in value, the other resistor decreases by the same amount. Wiring the potentiometer between ground and the power supply produces a voltage divider with voltage output proportional to the position of the contact. Connecting this variable voltage to the NI myRIO analog input provides a convenient sensing technique for angular position.

Study the video *Potentiometer Characteristics* ([youtu.be/3gwwF9rF\\_zU](https://youtu.be/3gwwF9rF_zU), 7:51) to learn about the potentiometer as a variable voltage source, and also to learn about proper sizing of the potentiometer to minimize power required and also to minimize loading effects that could distort the measurement.

**LabVIEW programming:** Study the video *Analog Input Express VI* ([youtu.be/N6Mi-VjBlmc](https://youtu.be/N6Mi-VjBlmc), 2:00) to learn how to use Analog Input Express VI to measure the voltage divider output voltage.

## 7.3 Basic Modifications

Study the video *Potentiometer Demo Walk-Through* ([youtu.be/RYeKIuU6DX8](https://youtu.be/RYeKIuU6DX8), 3:07) to learn the design principles of `Potentiometer demo`, and then try making these modifications to the block diagram of `Main.vi`:

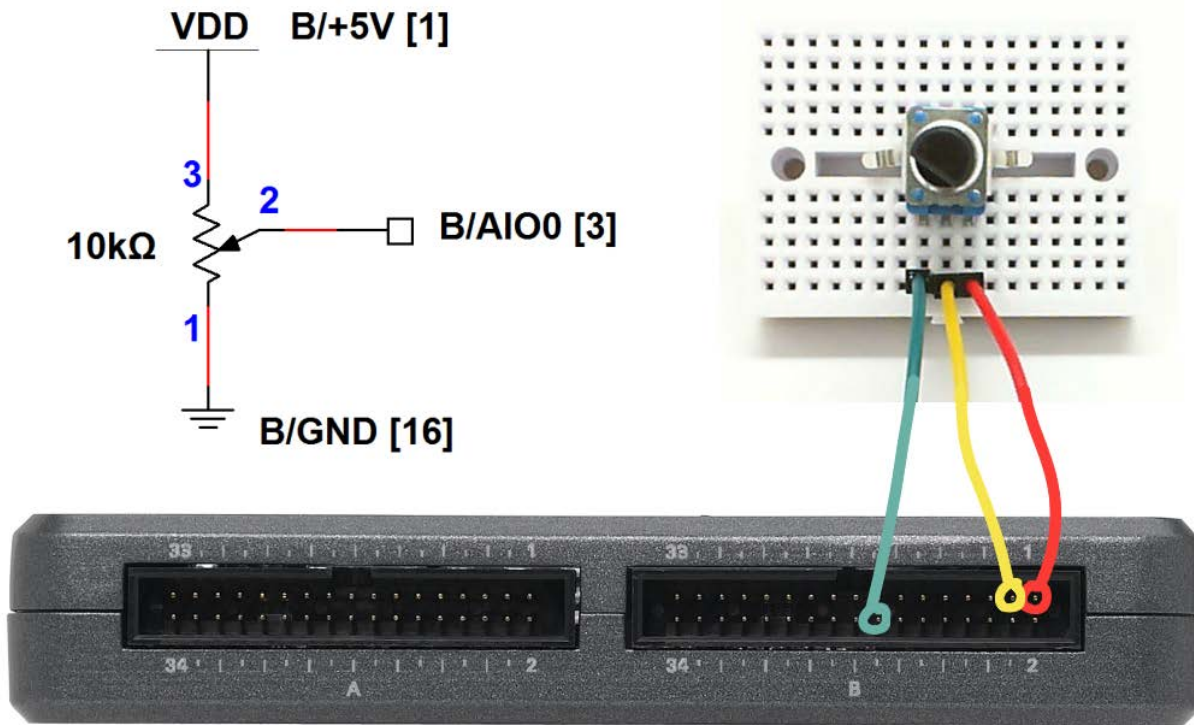


Figure 7.2: Demonstration circuit for potentiometer: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

1. Make the front-panel control selectable,
2. Replace dial indicator with another type of indicator,
3. Connect to a DIO input and monitor it's state inside a loop; adjust the pot voltage to locate the hysteresis edges of the digital input, and
4. Make a bargraph indicator of the pot wiper position with onboard LEDs.

## 7.4 Integrated Project Ideas

Now that you know how to use the potentiometer consider integrating it with other devices to create a complete system, for example:

- *Steer By Wire* (44)

## 7.5 For More Information

- *Potentiometer* by Resistorguide ~ Describes a variety of potentiometer types and characteristics:  
<http://www.resistorguide.com/potentiometer>



# 8 Thermistor

The *thermistor* – a contraction of “thermal” and “resistor” – is a two-terminal semiconductor device whose resistance varies with temperature. Most thermistors are of the *negative temperature coefficient* (NTC) type, meaning their resistance varies inversely with temperature. Figure 8.1 pictures the NI myRIO Starter Kit thermistor.

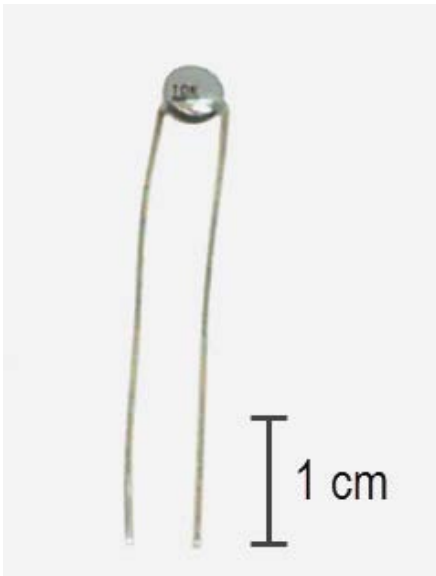


Figure 8.1: NI myRIO Starter Kit thermistor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Explain the thermistor principles of operation,
2. Measure the thermistor resistance a voltage divider and analog input,
3. Convert the measured resistance to temperature with the Steinhart-Hart thermistor equation, and
4. Size the voltage-divider resistor for the best measurement sensitivity and range.

## 8.1 Component Demonstration

Follow these steps to demonstrate correct operation of the thermistor.

### Select these parts from the NI myRIO Starter Kit:

- Thermistor, 10 k $\Omega$ , EPCOS B57164K103J, [http://www.epcos.com/inf/50/db/ntc\\_09/LeadedDisks\\_\\_B57164\\_\\_K164.pdf](http://www.epcos.com/inf/50/db/ntc_09/LeadedDisks__B57164__K164.pdf)
- Resistor, 10 k $\Omega$
- 0.1  $\mu$ F ceramic disk capacitor, marking “104” , <http://www.avx.com/docs/Catalogs/class3-sc.pdf>
- Breadboard
- Jumper wires, M-F (4 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 8.2 on page 33. The interface circuit requires four connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. 5-volt power supply  $\rightarrow$  B/+5V (pin 1)
2. Ground  $\rightarrow$  B/GND (pin 6)

3. Temperature measurement → B/AI0 (pin 3)
4. Supply voltage measurement → B/AI1 (pin 5)

Measure the resistance of the 10 k $\Omega$  resistor with an ohmmeter, as this value is required for the LabVIEW VI.

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Thermistor demo.lvproj` contained in the subfolder `Thermistor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

Enter the measured resistance of the 10 k $\Omega$  resistor as R [ohms].

**Expected results:** The demo VI displays the measured resistance of your thermistor; expect to see a value close to 10 k $\Omega$  at room temperature. Try heating the thermistor by gently pinching the thermistor body with your finger tips; you may also use a drinking straw or hair dryer to blow warm air on the thermistor. You should observe the resistance going down. How low can you make the resistance?

Use a plastic sandwich bag filled with two ice cubes or crushed ice. Surround the thermistor with

ice and you should observe the resistance going up. How high can you make the resistance?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 8.2 Interface Theory

**Interface circuit:** Constructing a voltage divider from a thermistor and a fixed-value resistor offers an effective and easy-to-build interface circuit. Placing the thermistor in the top branch of the divider makes the measured voltage increase with higher temperature.

Study the video *Thermistor Characteristics* ([youtu.be/US406sjBUxY](https://youtu.be/US406sjBUxY), 4:54) to learn more about thermistor characteristics and the *Steinhart-Hart thermistor equation* that converts measured thermistor resistance to temperature in degrees Kelvin. Study *Thermistor Resistance Measurement* ([youtu.be/PhZ2Q1CrwuQ](https://youtu.be/PhZ2Q1CrwuQ), 6:10) to learn how to measure the thermistor resistance with a voltage divider, and also how to size the resistor  $R$  for best measurement sensitivity and range. Also take a look at *Measure Resistance with a Voltage Divider* ([youtu.be/9KUVD7RkxNI](https://youtu.be/9KUVD7RkxNI), 9:44) for a more complete treatment of voltage dividers as a measurement technique.

**LabVIEW programming:** Study the video *Analog Input Express VI* ([youtu.be/N6Mi-VjBlmc](https://youtu.be/N6Mi-VjBlmc), 2:00)



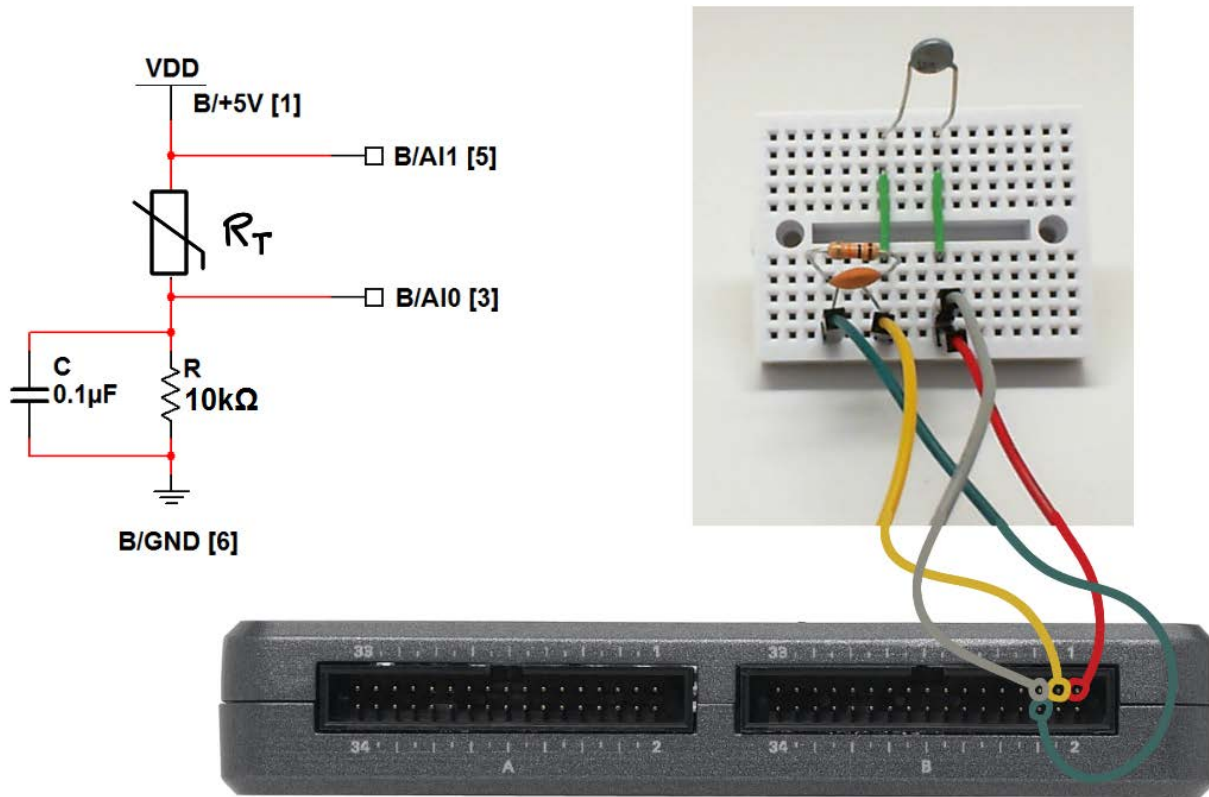


Figure 8.2: Demonstration circuit for thermistor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXF Connector B.

to learn how to use the Analog Input Express VI to measure the voltage divider's primary output as well as the voltage divider supply voltage.

### 8.3 Basic Modifications

Study the video *Thermistor Demo Walk-Through* ([youtu.be/xi0VIpGpf4w](https://youtu.be/xi0VIpGpf4w), 2:28) to learn the design principles of Thermistor demo, and then try making these modifications to the block diagram of Main.vi:

1. Add the necessary computation to convert the measured resistance to temperature in degrees Celsius; display the temperature on the large front-panel dial indicator. Use the built-in VIs Mathematics | Elementary | Natural Logarithm and Mathematics | Polynomial | Polynomial Evaluation. Use the polynomial coefficient values presented in the earlier video.
2. Modify your temperature display to display in degrees Fahrenheit.
3. Create a Boolean indicator to indicate when the measured temperature exceeds (or falls below)

a preset threshold.

## 8.4 Integrated Project Ideas

Now that you know how to use the thermistor consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Digital Thermometer* (45)
- *On-Off Control System* (52)
- *Weather Station* (58)

## 8.5 For More Information

- *Thermistors* by National Instruments ~ thermistor characteristics and the Steinhart-Hart thermistor equation:

<http://zone.ni.com/reference/en-XX/help/370466V-01/measfunds/thermistors>

- *NTC Thermistors* by Vishay ~ Learn about thermistor principles of operation, selection criteria, design equations, and example circuits and applications:

<http://www.vishay.com/docs/29053/ntcintro.pdf>

# 9 Photocell

A *photocell* is a two-terminal device fabricated from cadmium sulfide (CdS) and with resistance that varies with illumination in the visible spectrum of 400 to 700 nm. The photocell pictured in Figure 9.1 has a resistance that varies over many orders of magnitude: 10 k $\Omega$  at moderate illumination, less than 100  $\Omega$  at high illumination, and more than 10 M $\Omega$  in darkness.



Figure 9.1: NI myRIO Starter Kit photocell.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the photocell principles of operation,
2. Measure the photocell resistance with a voltage divider and analog input, and

3. Size the voltage-divider resistor for the best measurement sensitivity and range.

## 9.1 Component Demonstration

Follow these steps to demonstrate correct operation of the photocell.

**Select these parts from the NI myRIO Starter Kit:**

- Photocell, API PDV-P9203, [http://www.advancedphotonix.com/ap\\_products/pdfs/PDV-P9203.pdf](http://www.advancedphotonix.com/ap_products/pdfs/PDV-P9203.pdf)
- Resistor, 10 k $\Omega$
- Breadboard
- Jumper wires, M-F (3 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 9.2 on the following page. The interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. 5-volt power supply  $\rightarrow$  B/+5V (pin 1)
2. Ground  $\rightarrow$  B/GND (pin 6)
3. Photocell measurement  $\rightarrow$  B/AI0 (pin 3)

Measure the resistance of the 10 k $\Omega$  resistor with an ohmmeter, as this value is required for the LabVIEW VI.

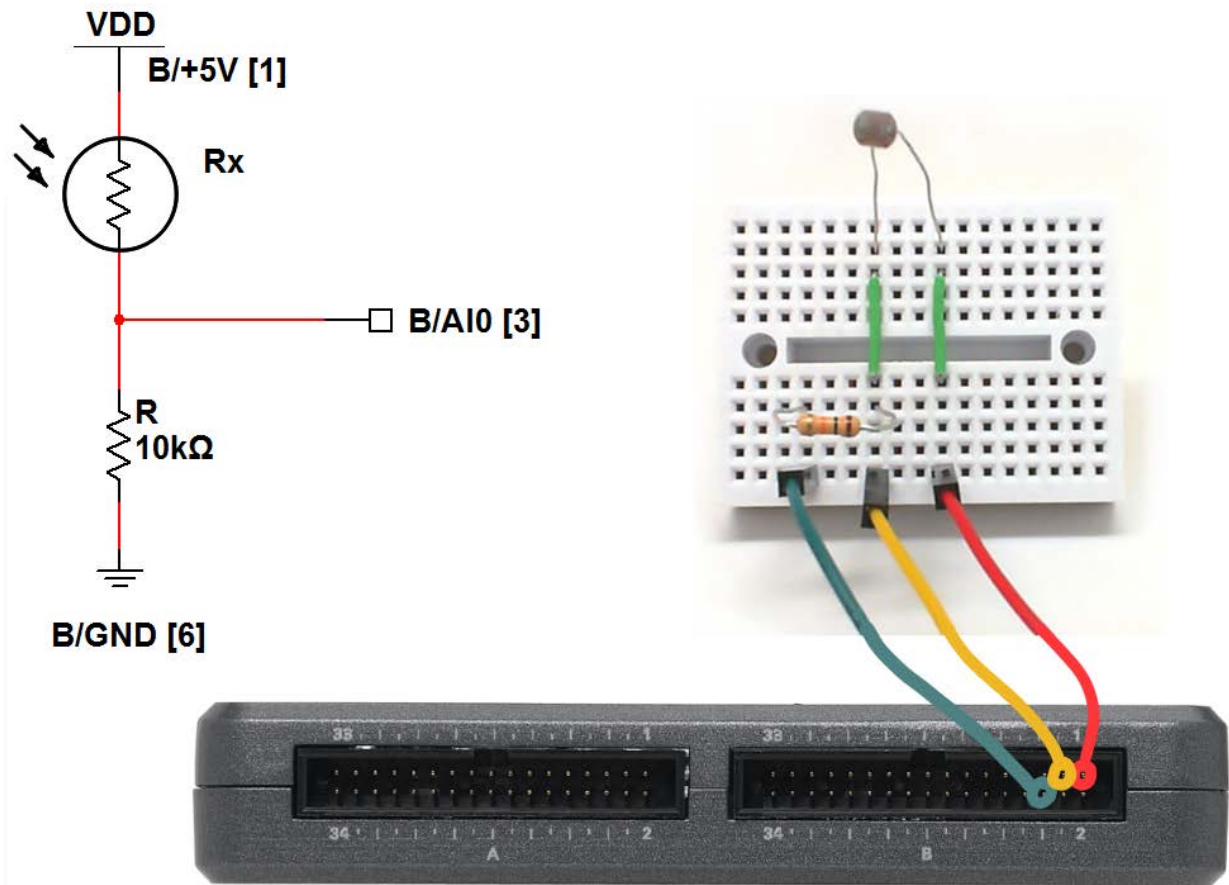


Figure 9.2: Demonstration circuit for photocell: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project Photocell demo.lvproj contained in the subfolder Photocell demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,

- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start

automatically.

**Expected results:** The demo VI displays the measured resistance of your photocell; expect to see a values in the range  $1\text{ k}\Omega$  to  $10\text{ k}\Omega$  at moderate illumination. Try blocking the light with a cover or cylindrical shroud such as a black straw. You should observe the resistance going up. How high can you make the resistance?

Use a flashlight or bright LED as illumination; you should observe the resistance going down. How low can you make the resistance?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 9.2 Interface Theory

**Interface circuit:** Constructing a voltage divider from a photocell and a fixed-value resistor offers an effective and easy-to-build interface circuit. Placing the photocell in the top branch of the divider makes the measured voltage increase with more illumination.

Study the video *Photocell Characteristics* ([youtu.be/geNeoFUjMjQ](https://youtu.be/geNeoFUjMjQ), 4:45) to learn about photocell characteristics, and then study *Measure Resistance with a Voltage Divider* ([youtu.be/9KUVD7RkxNI](https://youtu.be/9KUVD7RkxNI), 9:44) to learn how to

measure the photocell resistance with a voltage divider, and also how to properly choose the resistance  $R$  to maximize measurement sensitivity and range.

**LabVIEW programming:** Study the video *Analog Input Express VI* ([youtu.be/N6Mi-VjBlmc](https://youtu.be/N6Mi-VjBlmc), 2:00) to learn how to use Analog Input Express VI to measure the voltage divider's primary output.

## 9.3 Basic Modifications

Study the video *Photocell Demo Walk-Through* ([youtu.be/jZQqsc5GmoY](https://youtu.be/jZQqsc5GmoY), 3:07) to learn the design principles of Photocell demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add a Boolean front-panel control to make the voltage divider configuration user-selectable, i.e., one state of the control corresponds to the photocell in the lower branch while the other state selects the upper branch. Confirm that your modification works properly by swapping positions of the photocell and resistor.
2. Create a "room lights ON" detector with a suitable node from the Programming | Comparison subpalette and a Boolean front-panel indicator. Include a user-selectable threshold resistance as a front-panel numerical control.
3. Because analog inputs are not as plentiful as digital inputs, create the same "room lights ON" detector behavior, but do the comparison directly at a digital input; study *Resistive-Sensor Threshold Detector* ([youtu.be/TqLXJroefTA](https://youtu.be/TqLXJroefTA), 9:22) to learn the design procedure.

## 9.4 Integrated Project Ideas

Now that you know how to use the photocell consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Scanning Sensor* (51)
- *Music Maker* (56)
- *Weather Station* (58)

## 9.5 For More Information

- *Photocell Tutorial* by Digital DIY ~ An excellent tutorial with many types of interface circuits for analog and digital inputs:

[http://digital-diy.com/general-electronics/  
269-photocell-tutorial.html](http://digital-diy.com/general-electronics/269-photocell-tutorial.html)

- *Photocells* by Adafruit ~ A good overview of CdS photocells with interesting applications such as light-based motor control, line-following robots, and laser pointer breakbeam sensor:

<http://learn.adafruit.com/photocells/overview>

- *Photocell Tutorial* by Digital DIY ~ Several different light detector circuits, including a dual-photocell version that acts as a bistable latch:

[http://digital-diy.com/general-electronics/  
269-photocell-tutorial.html](http://digital-diy.com/general-electronics/269-photocell-tutorial.html)

# 10 Electret Microphone

A microphone serves as an acoustic sensor to record audio signals and to monitor acoustic level. *Electret* microphones provide good performance at low cost. The Chenyun CY-502 electret microphone pictured in Figure 10.1 is designed to connect directly to laptop computer soundcard inputs, and can also connect directly to the NI myRIO AUDIO IN jack.



Figure 10.1: NI myRIO Starter Kit electret microphone.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss the electret microphone operating principle based on parallel-plate capacitance,

2. Apply the “phantom power” provided by the NI myRIO AUDIO IN jack to power the electret microphone JFET impedance-level shifter, and
3. Increase the microphone gain with a simple inline preamplifier circuit powered by the NI myRIO AUDIO IN jack.

## 10.1 Component Demonstration

Follow these steps to demonstrate correct operation of the microphone.

**Select these parts from the NI myRIO Starter Kit:**

- Chenyun CY-502 computer microphone, [http://www.chenyun.cc/product\\_en.asp?ProductID=859](http://www.chenyun.cc/product_en.asp?ProductID=859)

**Build the interface circuit:** This microphone integrates a standard 3.5 mm audio jack that can plug directly into the NI myRIO AUDIO IN jack, consequently no special circuit is required. Refer to the diagram in Figure 10.2 on the next page.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project *Electret Microphone demo.lvproj* contained in the subfolder *Electret Microphone demo*,



Figure 10.2: Demonstration circuit for electret microphone connected directly to AUDIO IN.

- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the audio signal detected by the microphone as an oscilloscope display.

Try whistling, speaking, singing, or any other

sound, and you should see the corresponding waveform. You will need to position the microphone quite close to the sound source.

Click the **Stop** button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Microphone plugged into the AUDIO IN jack instead of the adjacent AUDIO OUT jack.

## 10.2 Interface Theory

**Interface circuit:** The Chenyun CY-502 electret microphone produces a monophonic (single



channel) audio signal of fractions of a volt up to more than a volt depending on the proximity and intensity of the sound source. Study the video *Electret Microphone Principles and Preamplifier Circuit* ([youtu.be/izJni0PM0bI](http://youtu.be/izJni0PM0bI), 8:48) to learn about the electret microphone principle of operation and the NI myRIO equivalent circuit for the AUDIO IN that provides power to the electret microphone as well as AC signal coupling; the video also describes a simple three-component audio preamplifier circuit (Figure 10.3 on the following page) that applies a gain factor of over 100× to greatly improve the sensitivity of the microphone.

### 10.3 Basic Modifications

Study the video *Electret Mic Demo Walk-Through* ([youtu.be/kZoFwQRYz98](http://youtu.be/kZoFwQRYz98), 2:52) to learn the design principles of Electret Microphone demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add a real-time amplitude spectrum display; use the point-by-point version of the built-in Amplitude and Phase Spectrum VI located on the Signal Processing | Point by Point | Spectral subpalette.
2. Add a VU (volume) meter that displays signal intensity; use the AC output of the built-in AC & DC Estimator PtByPt VI located on the Signal Processing | Point by Point | Sig Operation subpalette.
3. Try building the simple three-component audio preamplifier circuit of Figure 10.3 on the next page if you find that the electret microphone provides insufficient sensitivity for your application. You may wish to build this circuit on a small “perf board” (perforated plastic board with holes drilled on 0.1” centers) with a jack on one side for the CY-502 and a plug on the other side; use a jack on the output side to use the audio cable included with your NI myRIO. The optional 2.2 kΩ resistor connected to the NI myRIO

five-volt supply doubles the signal headroom to ±2 volts and maximizes the signal available to the AUDIO IN analog-to-digital converter.

### 10.4 Integrated Project Ideas

Now that you know how to use the electret microphone consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Guitar Tuner* (55)

### 10.5 For More Information

- *Computer Microphones* by Hobby-Hour.com ~ Describes the tip-ring-sleeve jack wiring and power supply connections for electret microphones and computer soundcards:  
[http://www.hobby-hour.com/electronics/computer\\_microphone.php](http://www.hobby-hour.com/electronics/computer_microphone.php)
- *Electret Microphones* by Open Music Labs ~ An excellent detailed teardown and explanation of electret microphone operating principles:  
<http://www.openmusiclabs.com/learning/sensors/electret-microphones>
- *Preamp to Electret Mic* by Instructables ~ A simple and effective preamplifier to raise the electret mic voltage level to line level:  
<http://www.instructables.com/id/Pre-amp-to-electret-mic>

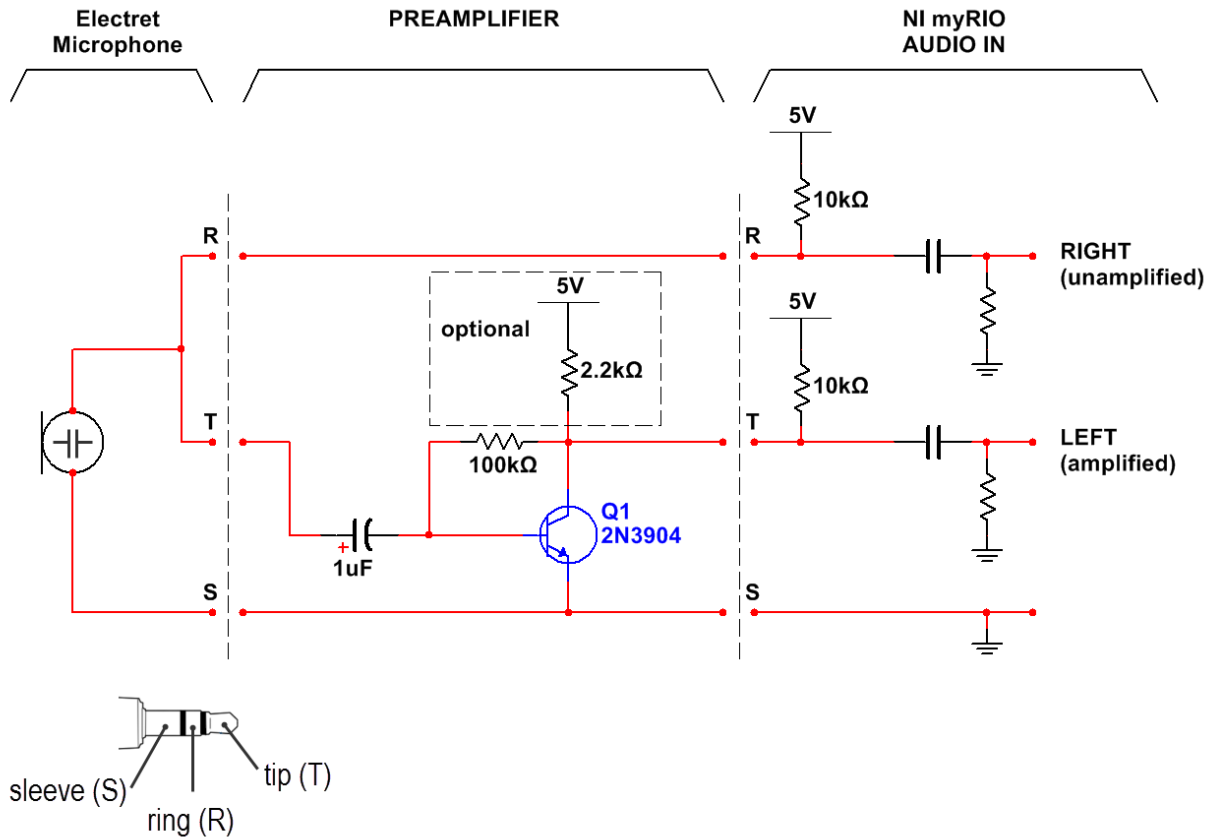


Figure 10.3: Preamplifier circuit to connect between the electret microphone and AUDIO IN. The NI myRIO AUDIO IN jack provides power to both the electret microphone and the preamplifier circuit. The optional 2.2 kΩ resistor connected to the NI myRIO five-volt supply doubles the signal headroom to  $\pm 2$  volts.

# 11 Buzzer/Speaker

The buzzer/speaker pictured in Figure 11.1 (also called a magnetic transducer) generates tones over much of the audible frequency spectrum. The speaker coil is designed for on-off operation (5 volts at 80 mA when on), therefore cycling the speaker with a square wave created from a digital output is the most straightforward way to operate the speaker. A transistor-based interface circuit is required to drive the coil.

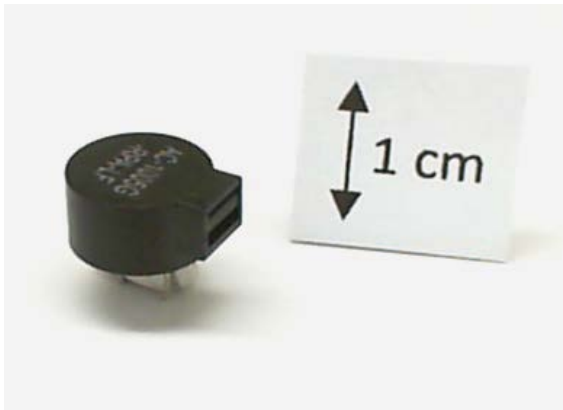


Figure 11.1: NI myRIO Starter Kit buzzer/speaker.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the buzzer/speaker principle of operation based on an electromagnetic coil that vibrates a small diaphragm,

2. Select a transistor to drive the coil from a digital output,
3. Protect the transistor from back-emf voltage spikes when the transistor switches the coil current on and off, and
4. Design the interface circuit to work with digital outputs that include either pull-up or pull-down resistors.

## 11.1 Component Demonstration

Follow these steps to demonstrate correct operation of the buzzer/speaker.

**Select these parts from the NI myRIO Starter Kit:**

- Buzzer/speaker, Soberton GT-0950RP3, <http://www.soberton.com/product/gt-0950rp3>
- 1N3064 small-signal diode, <http://www.fairchildsemi.com/ds/1N/1N3064.pdf>
- 2N3904 npn transistor, <http://www.fairchildsemi.com/ds/MM/MMBT3904.pdf>
- Resistor, 1.0 k $\Omega$
- Breadboard
- Jumper wires, M-F (3 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 11.2 on page 45. Note that the two buzzer/speaker terminals do not sit on tenth-inch centers, however, they fit just fine in two

diagonally-adjacent breadboard holes. The interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. 5-volt power supply → B/+5V (pin 1)
2. Ground → B/GND (pin 6)
3. Buzzer/speaker control → B/PWM0 (pin 27)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Buzzer-Speaker demo.lvproj` contained in the subfolder `Buzzer-Speaker demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Your buzzer/speaker should be making a low-frequency tone at 40 Hz. Move the front-panel control frequency [Hz] to increase the frequency. At what frequency is the sound output level the highest? What is the highest frequency that you can hear? Try blocking the port (rectangular hole) – how does this affect the sound at different frequencies?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset

causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not hearing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct transistor orientation — the transistor has a rounded shape on one side, and
- Correct diode orientation — when the diode is backwards the buzzer/speaker coil will never reach the voltage level necessary to turn on.

## 11.2 Interface Theory

**Interface circuit:** The buzzer/speaker contains an electromagnetic coil that vibrates a small diaphragm. The coil current is approximately 80 mA, well beyond the current drive limits of the NI myRIO digital output. The interface circuit uses an NPN transistor as a switch to turn the coil current on and off and a diode to protect the transistor from large back-emf voltage when the transistor abruptly shuts off the coil current.

Study the video *Buzzer/speaker Characteristics* ([youtu.be/8IbTWH9MpV0](https://youtu.be/8IbTWH9MpV0), 5:14) to learn more about the buzzer/speaker characteristics and interface circuit design principles including: sizing the transistor for relay coil current, importance of the diode to deal with back-emf voltage spiking, and circuit topologies for DIOs with internal pull-up resistors (MXP connector) and internal pull-down resistors (MSP connector).

**LabVIEW programming:** Study the video *PWM Express VI* ([youtu.be/mVN9jfwXleI](https://youtu.be/mVN9jfwXleI), 2:41) to learn how to use the PWM Express VI (PWM = pulse-width modulation) to create a squarewave output on B/PWM0 (pin 27) in the frequency range 40 Hz to 40 kHz with adjustable pulse width.

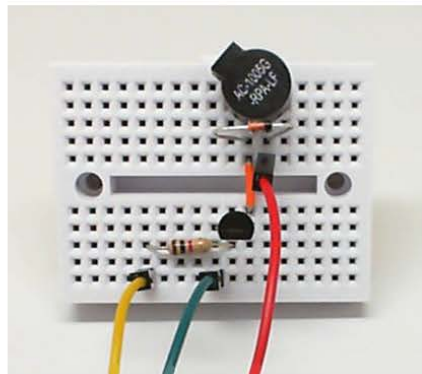
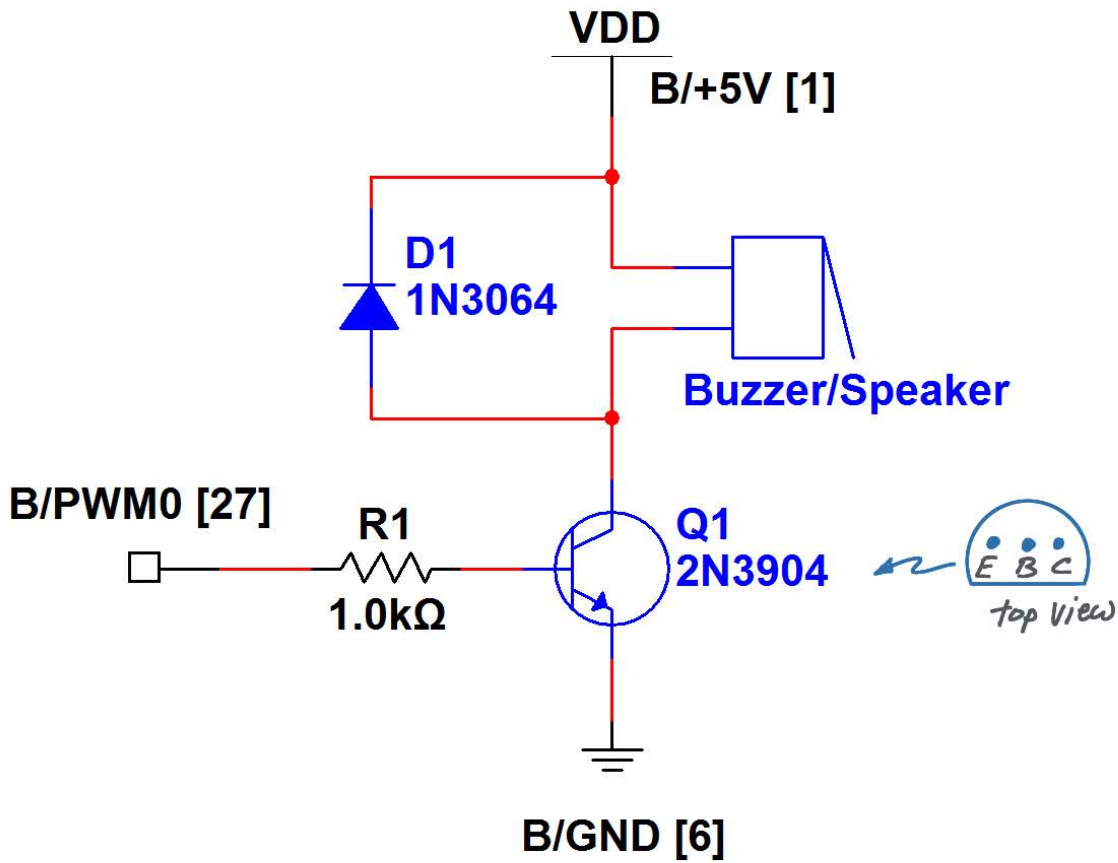


Figure 11.2: Demonstration circuit for buzzer/speaker: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

### 11.3 Basic Modifications

Study the video *Buzzer/speaker Demo Walk-Through* ([youtu.be/kW4v16GuAFE](https://youtu.be/kW4v16GuAFE), 2:06) to learn the design principles of `Buzzer-Speaker` demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add a front-panel control to adjust the waveform pulse width (called duty cycle) between 0 % and 100 %. How does a narrow pulse (low duty cycle) affect the tone quality at various frequencies?
2. Create a two-tone alarm signal with a Boolean front-panel control as an enable. Review `Discrete LED demo.lvproj` from Section 2.3 on page 7 to learn how to make a two-state oscillator.

### 11.4 Integrated Project Ideas

Now that you know how to use the buzzer/speaker consider integrating it with other devices to create a complete system, for example:

- *Music Maker* (56)
- *NTP Clock* (43)

### 11.5 For More Information

- *Products* by Soberton Inc. ~ Discover related products to the buzzer/speaker in the NI myRIO Starter Kit:

<http://www.soberton.com/products>

# 12 Motor

The low-voltage DC motor pictured in Figure 12.1 provides sufficient mechanical power to drive small fans or to spin lightweight objects. While the voltage is relatively low at 1.5 to 4.5 volts, the current can get as high as several hundred milliamps or even several amps in stall (blocked rotor) conditions. For this reason a power MOSFET is used as a motor driver. Figure 12.1 shows the NI myRIO Starter Kit DC motor.

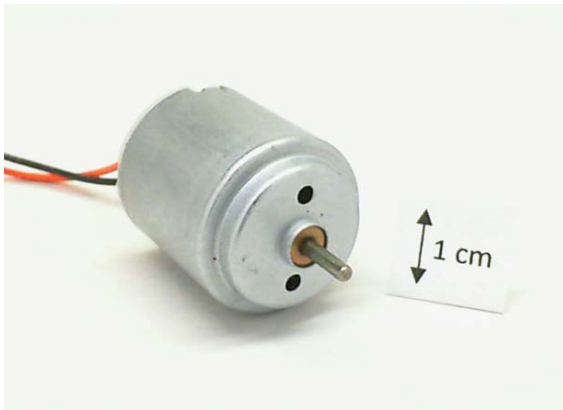


Figure 12.1: NI myRIO Starter Kit DC motor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the DC motor principle of operation,
2. Size the power transistor to drive the motor under various load conditions,

3. Protect the transistor from back-emf voltage spikes when the transistor switches the motor current on and off,
4. Design a voltage level-shifting circuit from 3.3 volts to 5 volts, and
5. Design the interface circuit to work with digital outputs that include either pull-up or pull-down resistors.

## 12.1 Component Demonstration

Follow these steps to demonstrate correct operation of the motor.

**Select these parts from the NI myRIO Starter Kit:**

- DC motor, [http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e\\_catalog.cgi?CAT\\_ID=ff\\_180phsh](http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e_catalog.cgi?CAT_ID=ff_180phsh)
- 1N4001 general-purpose rectifier, <http://www.vishay.com/docs/88503/1n4001.pdf>
- ZVN2110A n-channel enhancement-mode MOSFET, <http://www.diodes.com/datasheets/ZVN2110A.pdf>
- ZVP2110A p-channel enhancement-mode MOSFET, <http://www.diodes.com/datasheets/ZVP2110A.pdf>
- IRF510 n-channel enhancement-mode power MOSFET, <http://www.vishay.com/docs/91015/sihf510.pdf>
- Breadboard
- Jumper wires, M-F (4×)

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 12.2 on the facing page. The interface circuit requires four connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. 5-volt power supply → B/+5V (pin 1)
2. 3.3-volt power supply → B/+3.3V (pin 33)
3. Ground → B/GND (pin 30)
4. Motor control → B/DIO8 (pin 27)

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Motor demo.lvproj` contained in the subfolder `Motor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Click the DIO state button to set the digital output to its low state and your motor should spin at high speed, then click the button again to stop the motor. Note that the motor driver interface circuit is *active low*.

Click the **Stop** button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct transistor orientation — carefully follow the pin diagrams for each transistor; especially note that the IRF510 has the gate pin on the side rather than in the middle as do the lower-power MOSFETS, and
- Correct rectifier orientation — when the rectifier is backwards the motor will never reach the voltage level necessary to turn on.

## 12.2 Interface Theory

**Interface circuit:** The motor requires approximately 180 mA (at 3.3 V) when unloaded and over 1000 mA when running at maximum efficiency, three times higher than the maximum available current from all three NI myRIO connectors combined. Stalling the motor due to excessive loading or blocking the rotor demands even higher current because effective resistance of the motor is less than 1  $\Omega$ . For these reasons the IRF510 n-channel enhancement power MOSFET serves as a high-current solid-state switch to operate the motor. Because the IRF510 gate-to-source threshold voltage  $V_{GS(th)}$  ranges from 2 to 4 V the NI myRIO DIO output voltage of 3.3 V is not sufficient to turn on the IRF510. The two low-power MOSFETS arranged as a standard CMOS logic inverter supplied by the 5 V supply act as a 3.3-to-5 V *level shifter* to ensure that the IRF510 gate voltage is either 0 V (off) or 5 V (on).

Study the video *Motor Interfacing Theory* ([youtu.be/C\\_22XZaL5TM](https://youtu.be/C_22XZaL5TM), 6:49) to learn more about the motor principles of operation and interface circuit design principles including: sizing the power transistor for motor current under various load conditions, importance of the rectifier to deal with back-emf voltage spiking, level-shifting



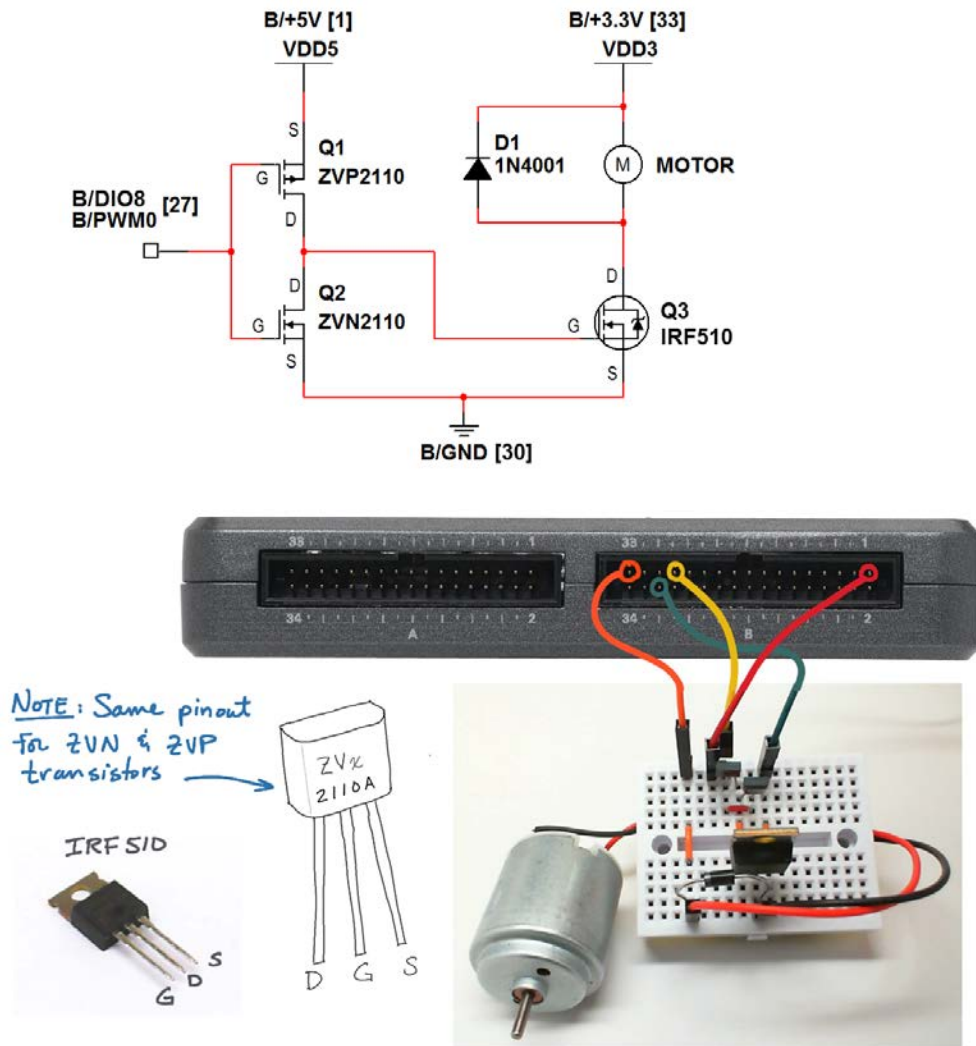


Figure 12.2: Demonstration circuit for DC motor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

circuit for 3.3-to-5 V, and required modification to connect the interface to the MSP connector with integral pull-down resistors.

**LabVIEW programming:** Study the video *PWM Express VI* ([youtu.be/mVN9jfwXleI](https://youtu.be/mVN9jfwXleI), 2:41) to learn how to use the PWM Express VI to create a pulse-width modulated square wave to provide variable-speed motor operation.

### 12.3 Basic Modifications

Study the video *Motor Demo Walk-Through* ([youtu.be/UCqFck0CLpc](https://youtu.be/UCqFck0CLpc), 1:56) to learn the design principles of *Motor demo*, and then try making these modifications to the block diagram of *Main.vi*:

1. Create variable-speed motor operation as follows:
  - Replace the existing Digital Output Express VI with the PWM Express VI. Choose the PWM channel as B/PWM0, the same connector pin as B/DIO8 (pin 27). Choose the remaining dialog box options so that both frequency and duty cycle are available as inputs,
  - Create pointer slide front-panel controls for each; right-click on each control and choose “Visible items” and the “Digital display,” and
  - Right-click on the frequency control, select “Scale” and then “Mapping,” and choose “Logarithmic.” Also, double-click the upper limit of your frequency control and enter “40000” and then similarly set the lower limit to “40.”

Experiment with both the duty cycle and frequency. What frequency minimizes audible PWM noise and maximizes your ability to create very slow motor speeds? What do you notice about restarting the motor after it stops? If you have a DMM ammeter handy, you may wish to observe the motor current under various conditions including mechanical loading, free running, and start-up.

2. Insert additional code to deal with the fact that the motor control interface is active-low. That is, you want 0 % duty cycle to turn the motor off rather than causing maximum speed as it does now.
3. Add a Boolean front-panel control as a motor enable. Trying using a **Select** node under the **Programming | Comparison** subpalette to set the duty cycle either to 0 or to the value of the front-panel duty cycle control.
4. Disconnect the motor control line and re-connect to C/PWM0 (pin 14) on MSP Connector C; adjust your VI to refer to this channel, too. You should observe that the motor is on due to the internal pull-down resistor. Now counter the effect of the internal pull-down with a 4.7 k $\Omega$  external pull-up resistor connected between the motor control line and the +5-volt supply. The motor should now remain off when the NI myRIO first powers on or after executing a myRIO reset.

### 12.4 Integrated Project Ideas

Now that you know how to use the DC motor consider integrating it with other devices to create a complete system, for example:

- *Tachometer* (50)
- *On-Off Control System* (52)

### 12.5 For More Information

- *Brushed DC Motor Fundamentals* by Microchip ~ Learn about DC motor principles of operation, drive circuits, direction control with an H-bridge, and speed control with Hall-effect sensors as feedback:

<http://ww1.microchip.com/downloads/en/AppNotes/00905B.pdf>

# 13 Rotary Encoder

A *rotary encoder*, also known as a *quadrature encoder*, combines a rotary knob and two switches that open and close in a staggered fashion as the knob turns. The knob angle and rotation direction can be sensed with suitable decoding of the switching waveforms. Figure 13.1 pictures the NI myRIO Starter Kit rotary encoder.



Figure 13.1: NI myRIO Starter Kit rotary encoder.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss essential concepts related to rotary encoders:
  - (a) Quadrature waveforms A and B,
  - (b) NI myRIO built-in encoder inputs and associated LabVIEW VI Encoder to indicate counts (position) and direction,
  - (c) Switch bounce that requires debouncing circuitry for reliable operation, and
2. Connect the rotary encoder “common” terminal to work with digital inputs that include either internal pull-up or pull-down resistors.

## 13.1 Component Demonstration

Follow these steps to demonstrate correct operation of the rotary encoder.

### Select these parts from the NI myRIO Starter Kit:

- Rotary encoder, <http://www.mantech.co.za/Datasheets/Products/F-11E.pdf>
- Resistor, 10 k $\Omega$  (2 $\times$ )
- 0.01  $\mu$ F ceramic disk capacitor, marking “103” (2 $\times$ ), <http://www.avx.com/docs/Catalogs/class3-sc.pdf>
- Breadboard
- Jumper wires, M-F (5 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 13.2 on page 53.

**TIP:** Flatten the two tabs on either side of the rotary encoder so that it sits flush on the breadboard surface.

The rotary encoder interface circuit requires three connections to the NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. Encoder A → B/ENC.A (pin 18)
2. Encoder A → B/DIO0 (pin 11)
3. Encoder B → B/ENC.B (pin 22)
4. Encoder B → B/DIO1 (pin 13)
5. Encoder COM → B/GND (pin 20)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project Rotary Encoder demo.lvproj contained in the subfolder Rotary Encoder demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the encoder A and B switch states as either “open” or “closed.” Slowly turn the encoder shaft clockwise and you should observe the following sequence: ‘A’ switch state closed with ‘B’ switch state open, then both

closed, then A open with B closed, and finally both open again. You should also observe that both switches are open when the encoder shaft is at rest in one of its twelve *detente* positions. Rotate the shaft in the counter-clockwise direction and you should see a similar sequence, but with switch B closing first.

The demo VI also maintains a counter of A/B switch transitions, and should increment by four counts for each click of the encoder in the clockwise direction and decrement by four counts for each click in the counter-clockwise direction. A front-panel indicator also displays the counter direction. Click the Reset Counter control to clear the counter to zero; click again to continue counting.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 13.2 Interface Theory

**Interface circuit:** The rotary encoder translates shaft rotation into a pair of switch opening/closing patterns known as *quadrature encoding*. The pair of switches called A and B connect directly to one of four NI myRIO encoder inputs shared with the standard digital input/output (DIO) terminals. The Encoder Express VI decodes the switching patterns to produce a count value as well as the counter direction. These outputs indicate the relative position of the rotary encoder shaft since the last time the counter was initialized.

NOTE: The additional connections to B/DIO0 and B/DIO1 simply provide another way to observe

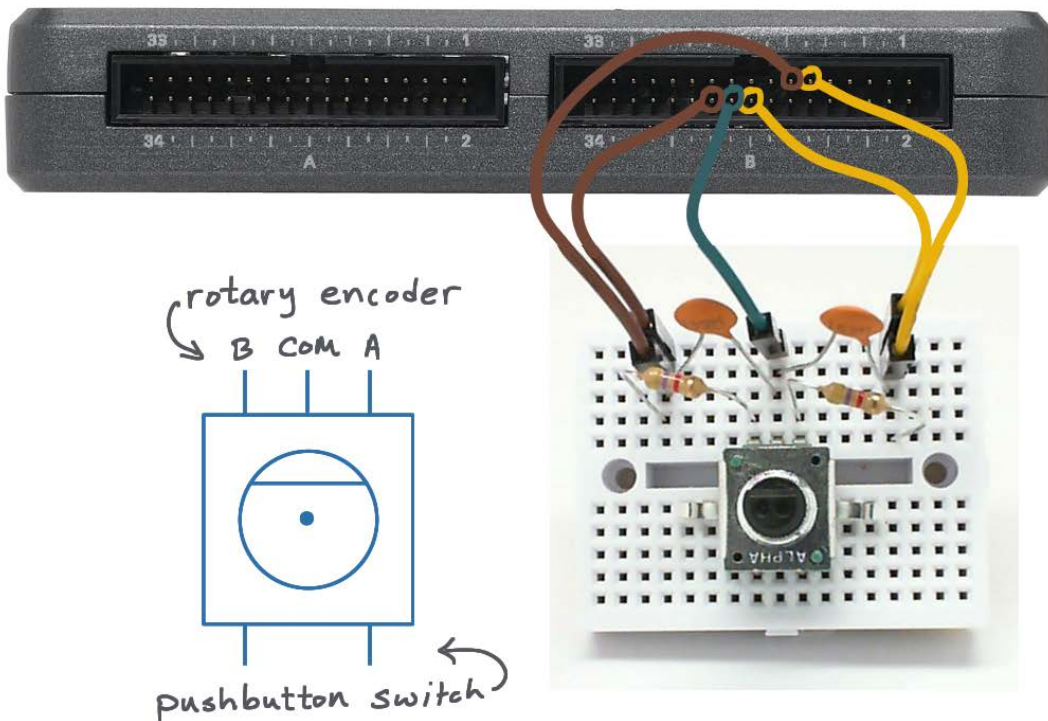
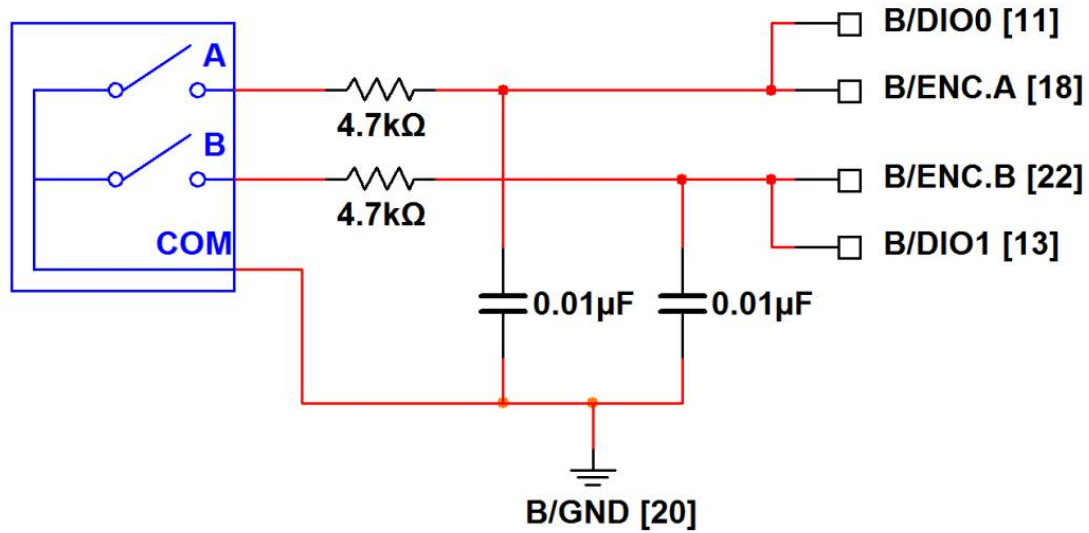


Figure 13.2: Demonstration circuit for rotary encoder: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

the switching activity; they are not required for decoding.

Study the video *Rotary Encoder Interfacing Theory* ([youtu.be/CpwGXZX-5Ug](https://youtu.be/CpwGXZX-5Ug), 10:08) to learn about the rotary encoder principles of operation, the quadrature waveforms produced by Switches A and B, interfacing techniques for the NI myRIO MXP and MSP connectors, and dealing with *switch bounce* that when ignored causes erroneous decoding of the switching waveforms.

### 13.3 Basic Modifications

Study the video *Rotary Encoder Demo Walk-Through* ([youtu.be/nmG1RqhQ6Rw](https://youtu.be/nmG1RqhQ6Rw), 3:15) to learn the design principles of Rotary Encoder demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Temporarily disconnect the two capacitors from the circuit, thereby removing the switch debouncing circuit. Experiment with various shaft rotation speeds and see if you can observe any relationship between rotation speed and counting error. Replace the capacitors and see if you can cause any count errors to occur – remember that each detente click corresponds to four counts.
2. Create a front-panel indicator to display the number of full encoder shaft revolutions.
3. Add the onboard button (myRIO | Onboard | Button Express VI) as another way to reset the counter value.
4. Add two wires to use the pushbutton on the decoder.

### 13.4 Integrated Project Ideas

Now that you know how to use the rotary encoder consider integrating it with other devices to create a complete system, for example:

- *Steer By Wire* (44)

### 13.5 For More Information

- *Quadrature Encoder Velocity and Acceleration Estimation with CompactRIO and LabVIEW FPGA* by National Instruments ~ A good review of quadrature encoders:

<http://www.ni.com/white-paper/3921/en>

- *Quadrature Encoding in a Rotary Encoder* by Robot Room ~ Look at the insides of a rotary encoder to better understand how it works; see the adjacent pages in this article, too:

<http://www.robotroom.com/Counter5.html>

- *Rotary Encoder: H/W, S/W or No Debounce?* by HiFiDUINO ~ A nice discussion of the rotary encoder switch bounce problem, with hardware and software solutions:

<http://hifiduino.wordpress.com/2010/10/20/rotaryencoder-hw-sw-no-debounce>

# 14 Photointerrupter

A *photointerrupter*, also called a *photogate*, combines an infrared LED *emitter* to produce an optical light path and an infrared *detector* to sense when the light path is broken. A photointerrupter serves as the basis for position and speed measurement when combined with timers and suitable targets of alternating opacity and transparency. Figure 14.1 pictures the NI myRIO Starter Kit photointerrupter.



Figure 14.1: NI myRIO Starter Kit photointerrupter.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Size the current-limiting resistor for the infrared emitter, and
2. Apply software-based edge detection to count photointerrupter events.

## 14.1 Component Demonstration

Follow these steps to demonstrate correct operation of the photointerrupter.

**Select these parts from the NI myRIO Starter Kit:**

- Photointerrupter, <http://sharpmicroelectronics.com/download/gpl1a57hr-epdf>
- Resistor, 470  $\Omega$
- Breadboard
- Jumper wires, M-F (3 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 14.2 on the following page. The photointerrupter interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +5-volt supply  $\rightarrow$  B/+5V (pin 1)
2. Ground  $\rightarrow$  B/GND (pin 6)
3. Photointerrupter output  $\rightarrow$  B/DIO0 (pin 11)



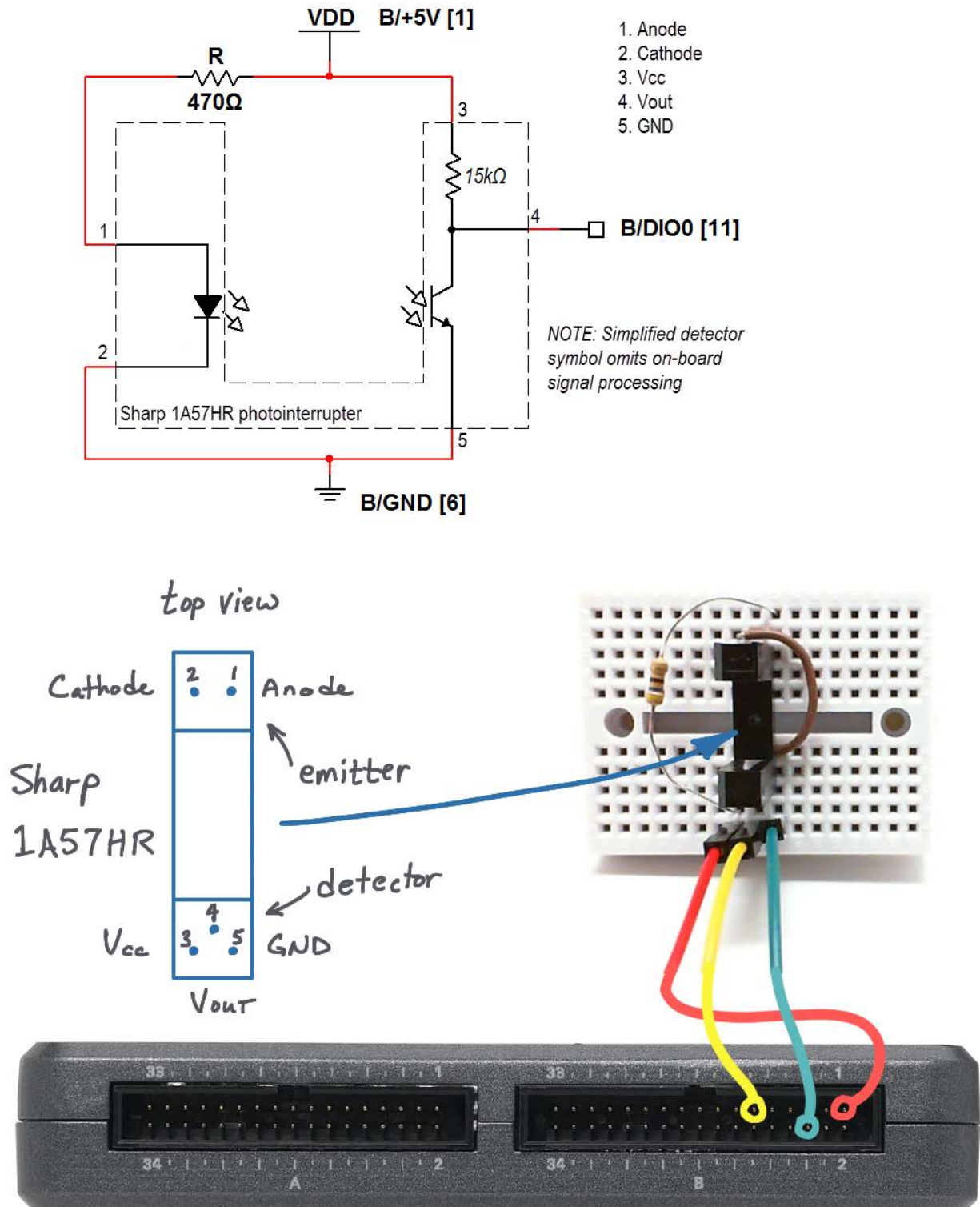


Figure 14.2: Demonstration circuit for photointerrupter: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.



**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Photointerrupter demo.lvproj` contained in the subfolder `Photointerrupter demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the state of the photointerrupter output and an event count. The state should normally be high. Block the photointerrupter optical path with an opaque card, your finger tip, and a sheet of paper. How much opacity does the photointerrupter require to trigger the event counter?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and

- Correct resistor value – ensure that you have a 470  $\Omega$  resistor (yellow-violet-brown) and not a 470 k $\Omega$  resistor (yellow-violet-yellow).

## 14.2 Interface Theory

**Interface circuit:** The photointerrupter places an infrared LED as the photoemitter on one side of the gap and a photodetector with signal conditioning circuitry on the other. The output is normally 5 volts and drops to zero volts when the optical path is blocked. The photointerrupter output may be connected directly to any NI myRIO digital input (MXP and MSP connectors).

Study the video *Photointerrupter Characteristics* ([youtu.be/u1FVfEvSdkg](https://youtu.be/u1FVfEvSdkg), 4:59) to learn more about the photointerrupter characteristics including the voltage output details and requirements for sizing the current-limiting resistor for the infrared LED emitter.

**LabVIEW programming:** Study the video *Digital Input Low-Level VIs* ([youtu.be/4nzt7THqU8U](https://youtu.be/4nzt7THqU8U), 4:09) to learn how to use the low-level Digital Input VIs to sense the state of the photointerrupter.

## 14.3 Basic Modifications

Study the video *Photointerrupter Demo Walk-Through* ([youtu.be/yuzNb1ZDbv4](https://youtu.be/yuzNb1ZDbv4), 3:22) to learn the design principles of `Photointerrupter demo`, and then try making these modifications to the block diagram of `Main.vi`:

1. Add the onboard LED Express VI (myRIO | Onboard subpalette) as an indicator on the output of the edge detector (the AND gate). Confirm that the LED flashes briefly when you block the photointerrupter optical path.
2. Measure and display the elapsed time between photointerrupter events – use the Elapsed Time

Express VI (Programming | Timing subpalette) inside a case structure with the selector terminal connected to the edge detector output.

## 14.4 Integrated Project Ideas

Now that you know how to use the photointerrupter consider integrating it with other devices to create a complete system, for example:

- *Tachometer* (50)

## 14.5 For More Information

- *How do I set up photogates for data collection?* by Vernier ~ Review of various ways to use a photointerrupter to take time-based measurements such as elapsed time and speed:

<http://vernier.com/ti1/1623>

- *Switch Tips: Photointerrupter switches* by Machine Design ~ Replace mechanical pushbuttons with a photointerrupter-based pushbutton to improve reliability and maximize equipment life:

<http://machinedesign.com/archive/>

[switch-tips-photointerrupter-switches](http://machinedesign.com/archive/switch-tips-photointerrupter-switches)

# 15 Hall-Effect Sensor

The *Hall effect* provides an elegant way to sense magnetic fields as a change in voltage. Hall-effect sensors provide either digital or analog outputs, with the former finding numerous applications in detecting proximity and sensing position and speed, while the latter can map the field strength pattern of a magnet. Figure 15.1 shows a photo of a typical Hall-effect sensor.



Figure 15.1: NI myRIO Starter Kit Hall effect sensor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss the Hall effect principle,
2. Discuss the two types of Hall-effect sensor behavior (latching and switching), and
3. Connect an open-collector sensor output to digital inputs with either pull-up resistors (MXP connector) or pull-down resistors (MSP connector).

## 15.1 Component Demonstration

Follow these steps to demonstrate correct operation of the Hall effect sensor.

**Select these parts from the NI myRIO Starter Kit:**

- US1881 Hall-effect latch, <http://www.melexis.com/Hall-Effect-Sensor-ICs/Hall-Effect-Latches/US1881-140.aspx>
- 0.1  $\mu\text{F}$  ceramic disk capacitor, marking “104” , <http://www.avx.com/docs/Catalogs/class3-sc.pdf>
- Breadboard
- Jumper wires, M-F (3 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 15.2 on page 61. The Hall effect sensor interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +5-volt supply  $\rightarrow$  B/+5V (pin 1)
2. Ground  $\rightarrow$  B/GND (pin 6)

### 3. Hall-effect sensor output → B/DIO0 (pin 11)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Hall-Effect Sensor demo.lvproj` contained in the subfolder `Hall-Effect Sensor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the state of the Hall-effect sensor output and the most-recent magnetic pole (north or south) applied to the marked side. The US1881 behaves as a *latch*, meaning that a magnetic field of the opposite pole must be applied to flip the latch to its opposite state. Find as many magnets as you can — refrigerator magnets work nicely — and experiment with the sensitivity of the sensor (how close does the magnet need to be to flip the state) as well as the location of the north and south poles on the magnet. How many distinct pole locations can you detect on the magnet? The answer may surprise you!

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and
- Correct orientation of the US1881 — as you face the labeled side of the sensor you have  $V_{DD}$  on the left, ground in the middle, and the output on the right.

## 15.2 Interface Theory

**Interface circuit:** The *Hall effect* relies on the fact that electrons moving across magnetic field lines experience a force. A current source establishes a current in a small semiconductor known as a *Hall plate*, and when subjected to a magnetic field the electrons tend to deflect to one side of the plate, leaving behind positive charges on the other side of the plate. The complete Hall-effect sensor on the US1881 senses the charge displacement as a voltage, applies amplification and other signal conditioning operations, and indicates the sensed magnetic pole type as the state of an open-drain output suitable for digital inputs.

Study the video *Hall-Effect Sensor Interfacing Theory* ([youtu.be/T9GP\\_cnz7rQ](https://youtu.be/T9GP_cnz7rQ), 9:48) to learn more about the Hall effect, the various types of Hall effect sensor output behaviors (latch, switch, and linear), and interface circuit techniques for both the MXP and MSP connectors.

**LabVIEW programming:** Study the video *Digital Input Low-Level VIs* ([youtu.be/4nZr7THqU8U](https://youtu.be/4nZr7THqU8U), 4:09) to learn how to use the low-level Digital Input VIs to sense the state of the Hall-effect sensor.

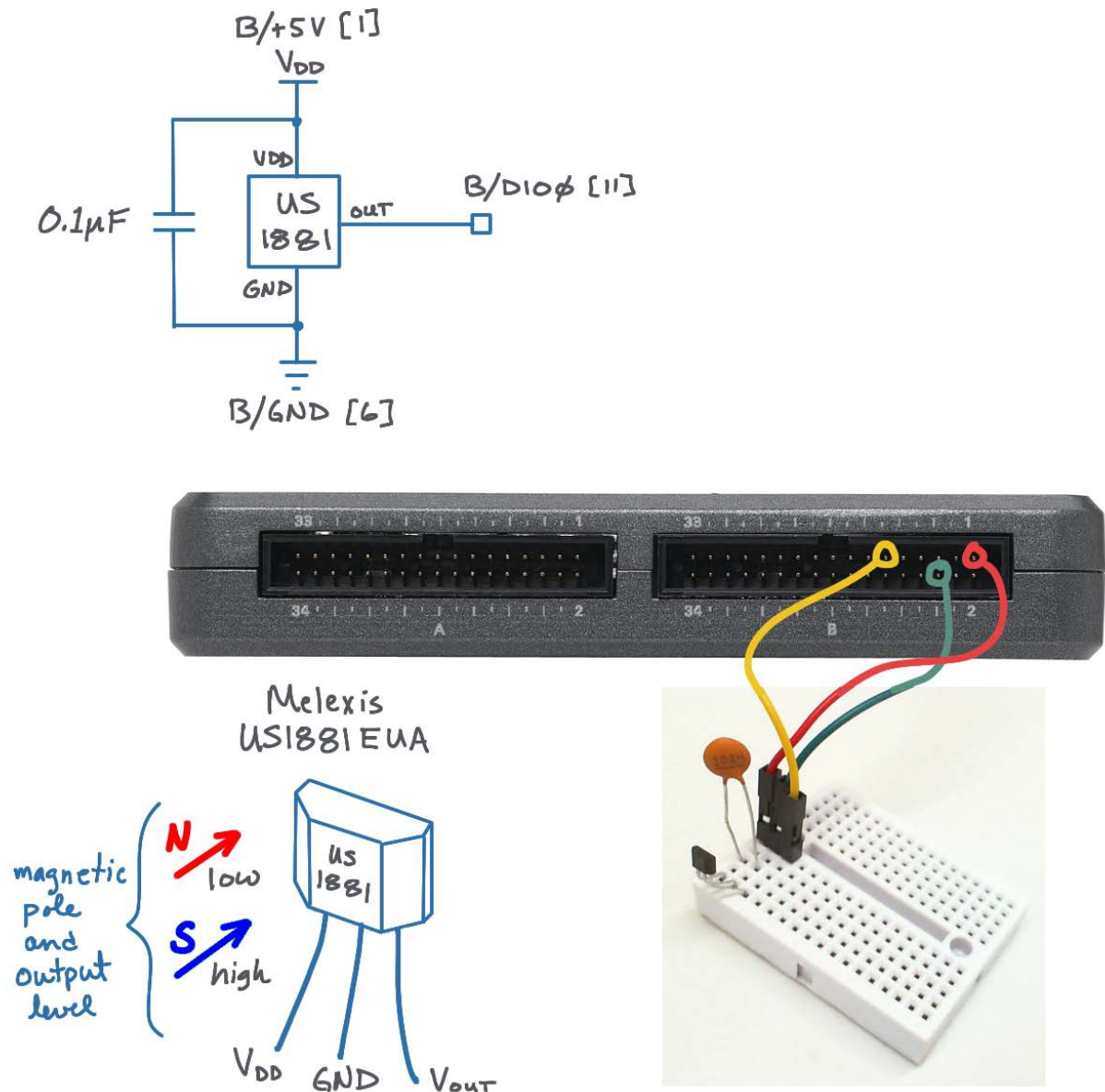


Figure 15.2: Demonstration circuit for Hall effect sensor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

### 15.3 Basic Modifications

Study the video *Hall-Effect Sensor Demo Walk-Through* ([youtu.be/BCJLg-WbIK4](http://youtu.be/BCJLg-WbIK4), 2:36) to learn the design principles of `Hall-Effect Sensor demo`, and then try making these modifications to the block diagram of `Main.vi`:

1. Add the onboard LED Express VI (`myRIO | Onboard subpalette`) to indicate the Hall-effect sensor output state.

### 15.4 Integrated Project Ideas

Now that you know how to use the Hall effect sensor consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)

### 15.5 For More Information

- *What is the Hall Effect?* by Melexis.com ~ Includes an excellent animation of the charge displacement phenomenon due to a magnetic field:  
<http://www.melexis.com/Assets/What-is-the-Hall-Effect-3720.aspx>
- *Hall Applications Guide* by Melexis.com ~ Slide-by switch, proximity sensor, rotary interrupt switch, pushbutton, flowmeter, liquid-level sensing, brushless DC motor commutation, and many more; includes detailed discussion of magnet properties:  
<http://www.melexis.com/Assets/Hall-Applications-Guide--3715.aspx>
- *A Strange Attraction: Various Hall Effect Sensors* by bildr.org ~ Video demonstration of latch, switch, and linear-mode behavior:  
<http://bildr.org/2011/04/various-hall-effect-sensors>

# 16 Piezoelectric-Effect Sensor

A *piezoelectric sensor* sandwiches a piezoelectric material between two plate electrodes. Deforming the sensor causes a charge displacement that appears as a measurable voltage between the plates, much like a parallel-plate capacitor. Combining the piezoelectric sensor with a charge-to-voltage converter provides a useful sensor to detect shock and vibration. Figure 16.1 pictures the NI myRIO Starter Kit piezo sensor.



Figure 16.1: NI myRIO Starter Kit piezo sensor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the piezoelectric effect,
2. Design a charge-to-voltage converter as an interface circuit to the analog input, and
3. Set the gain of the charge-to-voltage converter for different applications including a deflection sensor and a highly-sensitive vibration/shock sensor.

## 16.1 Component Demonstration

Follow these steps to demonstrate correct operation of the piezo sensor.

**Select these parts from the NI myRIO Starter Kit:**

- Piezo film sensor, DT-series, [http://meas-spec.com/product/t\\_product.aspx?id=2478](http://meas-spec.com/product/t_product.aspx?id=2478)
- AD8541 rail-to-rail single-supply op amp, <http://www.analog.com/AD8541>
- 100 pF ceramic disk capacitor, marking “101” , <http://www.vishay.com/docs/45171/kseries.pdf>
- 0.001  $\mu\text{F}$  ceramic disk capacitor, marking “102” , <http://www.vishay.com/docs/45171/kseries.pdf>
- Resistor, 10 M $\Omega$
- Resistor, 10 k $\Omega$  (2 $\times$ )
- Breadboard
- Jumper wires, M-F (3 $\times$ )

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 16.2 on the following page. The piezo sensor interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +5-volt supply  $\rightarrow$  B/+5V (pin 1)
2. Ground  $\rightarrow$  B/GND (pin 6)
3. Sensor output  $\rightarrow$  B/A10 (pin 3)

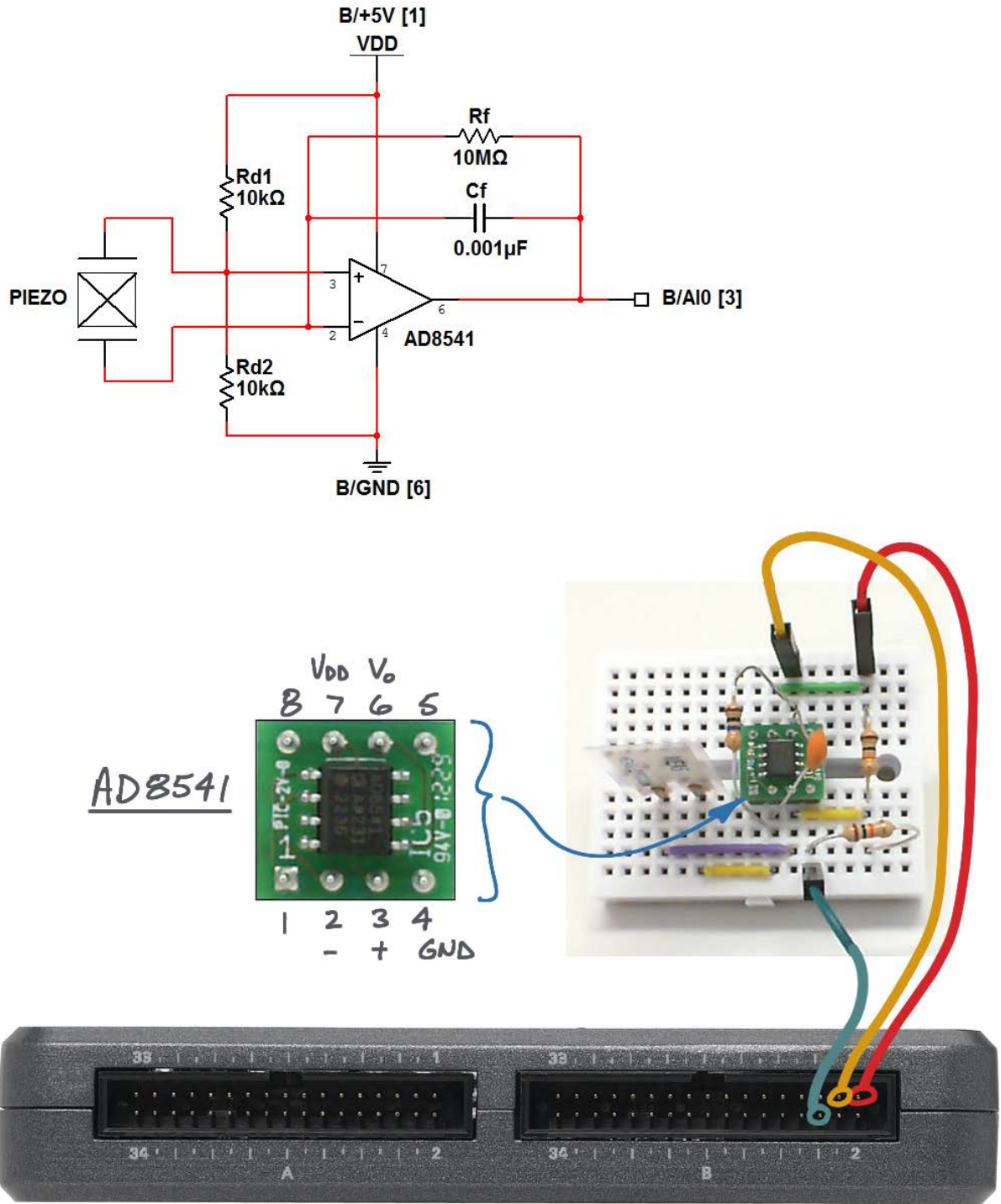


Figure 16.2: Demonstration circuit for piezo sensor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.



**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Piezoelectric-Effect Sensor demo.lvproj` contained in the subfolder `Piezoelectric-Effect Sensor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the piezo sensor amplifier output much like an oscilloscope display. You should observe that the voltage is a constant close to 2.5 volts. Flex the piezo sensor and you should see a corresponding change in the voltage; observe the difference as you flex the sensor one way and then the other. Next, try flicking the sensor and jostling the breadboard. You should see a decaying sinusoid similar to an underdamped second-order system. Experiment to determine the sensitivity of the sensor, i.e., what is the lightest disturbance that still registers on the display?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct op amp wiring – be certain that you understand the pinout of the AD8541, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 16.2 Interface Theory

**Interface circuit:** The piezo sensor acts like a charge pump (current source) when flexed. The interface circuit is a *charge-to-voltage converter* based on an op amp.

Study the video *Piezo Sensor Interfacing Theory* ([youtu.be/dHaPUJ7n-UI](https://youtu.be/dHaPUJ7n-UI), 5:13) to learn about the piezoelectric effect, charge-to-voltage converters, and choice of feedback capacitor to control the sensitivity of the sensor.

## 16.3 Basic Modifications

Study the video *Piezoelectric-Effect Sensor Demo Walk-Through* ([youtu.be/b1me4f-3iOE](https://youtu.be/b1me4f-3iOE), 2:54) to learn the design principles of `Piezoelectric-Effect Sensor demo`, and then try making these modifications to the block diagram of `Main.vi`:

1. Add the onboard LED Express VI (myRIO | Onboard subpalette) as a “bump” indicator; use the Programming | Comparison | In Range and Coerce VI as the upper-limit and lower-limit range test for the analog input Read VI. Confirm that the LED flashes briefly when you bump the sensor. Try one or more of the following methods to increase the sensitivity of your bump sensor:
  - Increase the charge-to-voltage converter gain by 10× by changing the feedback capacitance to 100 pF.

- Right-click on the waveform chart Y-axis, select autoscaling mode, and observe the minimum and maximum values of the sensor waveform when the sensor is at rest. Use this information to more precisely set the threshold values for the In Range and Coerce VI.
- Add mass to the sensor tab (modeling clay works well).

Try to make the bump indicator flash at the slightest disturbance such as tapping on the table.

## 16.4 Integrated Project Ideas

Now that you know how to use the piezo sensor consider integrating it with other devices to create a complete system, for example:

- *Wireless Sensor* (41)
- *Data Logger* (42)

## 16.5 For More Information

- *Fundamentals of Piezoelectric Shock and Vibration Sensors* by Digi-Key TechZone ~ Detect excessive shock and vibration that could damage equipment:  
<http://www.digikey.com/us/es/techzone/sensors/resources/articles/fundamentals-of-piezoelectric-sensors.html>
- *Bicycle Anti-Theft Alarm Circuit Diagram* by Electronic Circuit Diagrams & Schematics ~ Alarm circuit based on the same piezo sensor in the NI myRIO Starter Kit:  
<http://circuitsstream.blogspot.com/2013/05/bicycle-anti-theft-alarm-circuit-diagram.html>
- *Signal Conditioning Piezoelectric Sensors* by Texas Instruments ~ Piezoelectric sensor principle of operation and circuit model, voltage-mode amplifier and charge-mode amplifier circuits:  
<http://www.ti.com/lit/an/sloa033a/sloa033a.pdf>

## **Part II**

# **NI myRIO Mechatronics Kit**



# 17 Servo

A servo motor — usually called a *servo* for short — combines a DC motor, gearbox, potentiometer, and controller electronics to provide relatively precise angular position control of a rotating shaft. Servos provide motion control for robot arms, rotating sensor scanners, and other actuators. Servos have been used for many years in remote-controlled airplanes, cars, and boats to manipulate control surfaces (aileron, elevator, rudder) and steering. Figure 17.1 pictures the GWS S03N STD servo included in the NI myRIO Mechatronics Kit; the kit also includes the GWS S35+ XF continuous-rotation sensor that can serve as the basis of a robot drive train.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss the main components of a servo feedback control system: pulse-style command input, controller, DC motor, gearbox, and potentiometer as an angular position sensor,
2. Create a pulse-width modulated (PWM) signal to command the servo spline to a desired angle,
3. Null any nonideal offsets in the angular position, and
4. Explain the fundamental difference between a standard servo and a servo intended for continuous operation.



Figure 17.1: NI myRIO Mechatronics Kit servo.

## 17.1 Component Demonstration

Follow these steps to demonstrate correct operation of the servo.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Servo, GWS S03N STD, <http://gwsus.com/english/product/servo/standard.htm>
- Jumper wires, M-F (3×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 17.2 on the next page; the servo requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. Vcc (red) → B/+5V (pin 1)
2. Ground (black) → B/GND (pin 6)
3. Command signal (white) → B/PWM0 (pin 27)

Make certain that you are using the correct servo (GWS S03N STD); the NI myRIO Mechatronics Kit includes a continuous-rotation servo with similar appearance.

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Servo demo.lvproj` contained in the subfolder `Servo demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI includes a pointer slide control to adjust the servo angle. Move the slider and confirm that the servo shaft turns in response. You may also use the Page-Up and Page-Down keys to move the slider. Which slider values (positive or negative) correspond to clockwise motion?

The servo includes a parts kit that includes a variety of “servo arms,” also known as “servo horns” (see Figure 17.3 on page 72). Attach the two-arm servo horn to the servo “spline” (the gear-shaped rotating shaft of the servo) so that you can more easily see the servo rotation angles.

The slider is calibrated in “percent full scale” (%FS). Estimate the servo angle at 100%FS and then at –100%FS. Use the direct entry box at the top of the slider to apply step changes, e.g., +100%FS to –100%FS; how quickly does the servo rotate between these two angles?

The default limits on the slider allow a 2× “over travel” factor. At what %FS values does the servo reach its rotation limits?

The servo command input is a variable-width pulse with 1.0 ms at –100%FS and doubling to 2.0ms at +100%FS; the midpoint pulse width 1.5 ms — called the neutral-position pulse width — corresponds to 0%FS. The pulse must be repeated at a sufficiently fast rate and yet not too fast. Try adjusting the `freq [Hz]` control to a lower frequency (say, 10 Hz) and to a higher frequency (say, 200 Hz), each time moving the position slider to command different angles. Note the two indicators under this control that indicate the pulse width and “duty cycle” (percent of time that the pulse is active) of the signal sent to the servo command input. What happens to the servo angle at these frequency extremes? Experiment to determine a range of frequencies that yields satisfactory control of the servo angle.

Disconnect the servo and replace it with the continuous-rotation servo (GWS GWS S35+ XF); this servo uses a slightly different connector: red = +5 V, brown = ground, and yellow = command input. Investigate the behavior of this servo and compare it to your earlier results.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

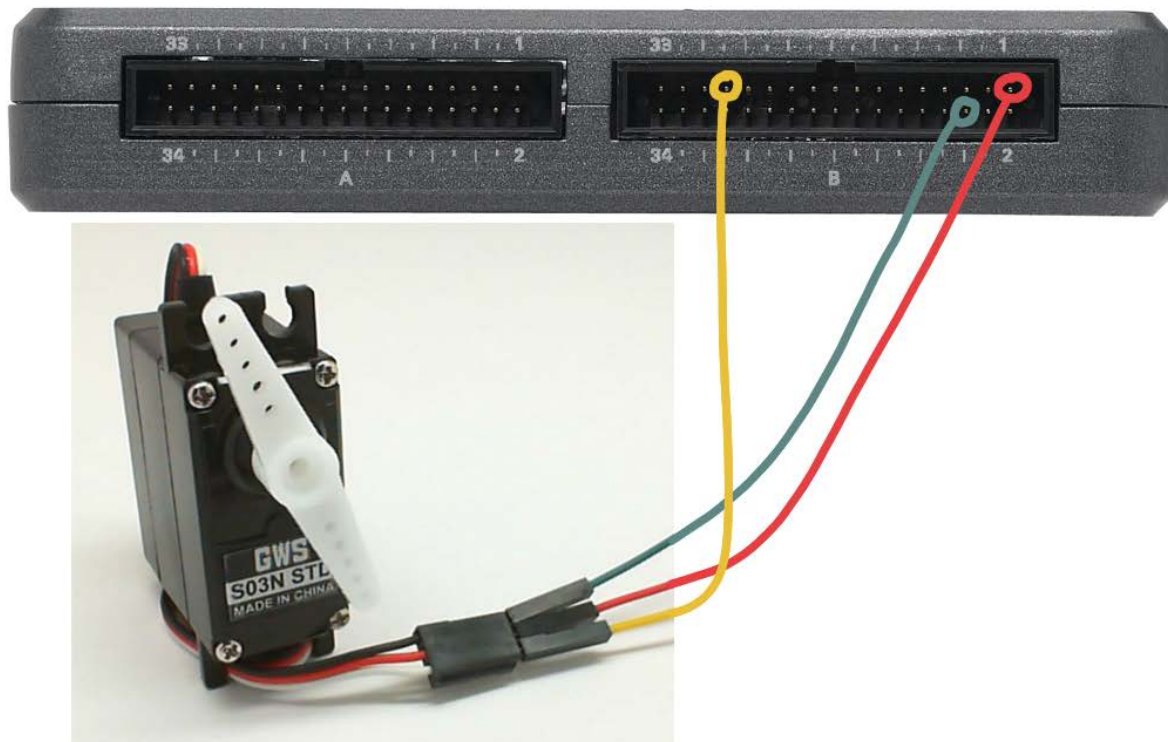
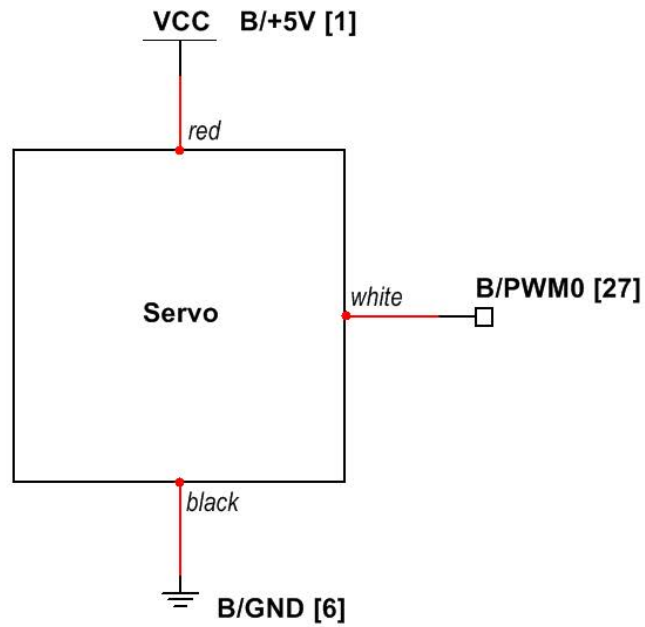


Figure 17.2: Demonstration setup for the GWS S03N STD servo connected to NI myRIO MXP Connector B.

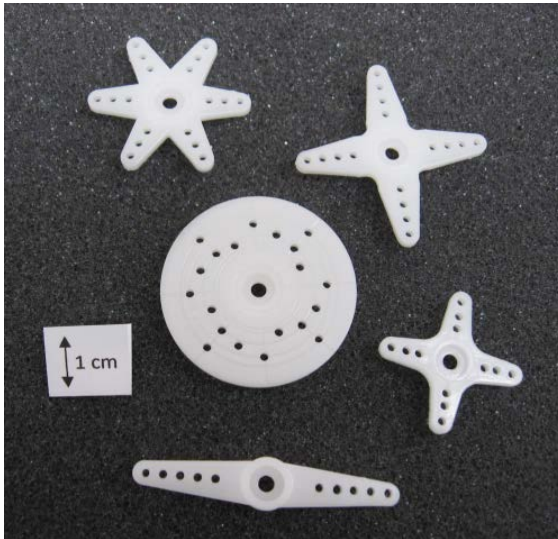


Figure 17.3: Servo control horns included with the GWS S03N STD.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and
- Correct servo connections — double-check your connections, and ensure that you have connected the red line to the +5-volt supply, the black line to ground, and the white line to the PWM0 output.

## 17.2 Interface Theory

**Interface circuit:** The servo adjusts its shaft angle according to the *command input*, a periodic pulse that varies in width between 1.0 and 2.0 ms. A pulse width centered between these two limits (1.5 ms) commands the servo to its neutral (center) position.

The servo requires the 5-volt power supply and a single connection to one of the pulse-width modulation (PWM) outputs available on the NI myRIO.

Study the video *Servo Interfacing Theory* ([youtu.be/DOu5AvSDP2E](https://youtu.be/DOu5AvSDP2E), 7:18) to learn about servo applications, the internal components of the servo (controller, motor drive, gearbox, and potentiometer), the principle of operation of the servo's feedback control system, and the technique used to make a continuous-rotation servo.

**LabVIEW programming:** Study the video *PWM Express VI* ([youtu.be/mVN9jfwXleI](https://youtu.be/mVN9jfwXleI), 2:41) to learn how to use the PWM VIs to open a channel to a PWM output and to set the pulse width and pulse repetition rate.

## 17.3 Basic Modifications

Study the video *Servo Demo Walk-Through* ([youtu.be/QXHe0DFbUdc](https://youtu.be/QXHe0DFbUdc), 4:23) to learn the design principles of *Servo demo*, and then try making these modifications to the block diagram of *Main.vi*:

1. Add the necessary computation to calibrate the pointer slide control in degrees of rotation. Use the same technique as the null offset code, but now with a multiplicative scale factor (be sure to initialize the feedback node to 1). Test your code as follows:
  - Null the offset,
  - Set the servo angle to zero and note the position of the servo horn arm,
  - Adjust the servo angle until the arm rotates +90 degrees,
  - Click and then release your "scale" button, and then
  - Enter 90 degrees into the point slide control direct entry box and confirm that the arm rotation is exactly 90 degrees from center.



2. Make the servo follow an *angular position trajectory*, i.e., a sequence of angles stored in an array:
  - Replace the while-loop with a for-loop (right-click on the for-loop and select the Conditional Terminal option),
  - Create an array using the Sine Pattern generator found in the Signal Processing | Sig Generation subpalette; select an amplitude for the sine pattern that will move the servo horn arm throughout its possible range of travel,
  - Change the pointer slide control to an indicator, and
  - Connect the Sine Pattern output through the for-loop frame to the position control wire.

## 17.4 Integrated Project Ideas

Now that you know how to use the servo consider integrating it with other devices to create a complete system, for example:

- *Steer By Wire* (44)
- *Hotel Room Safe Controller* (49)
- *Scanning Sensor* (51)
- *NTP Clock* (43)

## 17.5 For More Information

- *Actuators and Servos* by Society of Robots ~ Many practical details on servos:  
[http://www.societyofrobots.com/actuators\\_servos.shtml](http://www.societyofrobots.com/actuators_servos.shtml)
- *Servo Control* by PC Control Learning Zone ~ Another good tutorial on servos:  
[http://www.pc-control.co.uk/servo\\_control.htm](http://www.pc-control.co.uk/servo_control.htm)



# 18 Motor Adapter for NI myRIO and Geared Motor

The Motor Adapter for NI myRIO connects directly to an MXP port and integrates an H-bridge motor controller and power manager into a trouble-free solution to drive two DC motors using pulse-width modulation (PWM) speed control. Both the motor adapter and the compatible geared motor are included with the NI myRIO Mechatronics Kit (Figure 18.1). The Motor Adapter supports motor voltages in the 6 to 16 V range with up to 1.5 A current, accommodates quadrature encoder feedback for position and speed control, senses the motor current and voltage as waveforms for monitoring purposes, supports servo motors and stepper motors, and includes current overlimit protection.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the H-bridge principle of operation,
2. Explain how to control the motor speed and direction of rotation,
3. Sketch the quadrature encoded output waveforms of the shaft position sensor for clockwise and counterclockwise rotation,
4. Apply the LabVIEW PWM Express VI to control the motor speed, and
5. Apply the LabVIEW Encoder Express VI to measure the motor shaft position and speed.



Figure 18.1: NI myRIO Mechatronics Kit motor adapter and geared motor.

## 18.1 Component Demonstration

Follow these steps to demonstrate correct operation of the motor adapter and geared motor.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Motor Adapter for NI myRIO), <http://store.digilentinc.com/motor-adapter-for-ni-myrio>

- Geared motor, <http://digilentinc.com/Products/Detail.cfm?NavPath=2,403,625&Prod=MT-MOTOR>
- Small screwdriver

You will also need a DC power supply between 6 V to 12 V; a 9-volt battery will also work quite well.

**Build the interface circuit:** Refer to the diagram shown in Figure 18.2 on the next page. Simply connect the Motor Adapter to MXP Connector A, attach the geared motor, and attach the power supply as shown. Use a small Phillips or straight-bladed screwdriver to loosen and tighten the screws on the motor supply voltage connector.

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Motor Adapter & Geared Motor demo.lvproj` contained in the subfolder `Motor Adapter & Geared Motor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Increase the PWM duty (duty cycle) slider until the motor begins to turn. You should see the motor turn, the Counter Value

indicator value increment in the positive direction, the Counter Direction indicator display “Counting Up,” and activity on the motor current and voltage waveform displays. Continue increasing the PWM duty slider to speed up the motor.

Reduce the motor speed and then click the Direction button to reverse the motor direction. You should see the direction indicator change to “Counting Down” and the counter value decrementing.

Click the Reset Counter button and then experiment with the motor speed. Note that the counter now serves as a speed indicator reporting the number of “counts” measured in a 100 millisecond interval.

Try varying the PWM freq slide control to adjust the rate at which pulses are applied to the H-bridge controller’s “Enable” input. Look for relationships between the PWM frequency and duty cycle especially in terms of these points:

1. What PWM frequency range makes it easiest to start the stopped motor when increasing the PWM duty cycle from zero?
2. What PWM frequency range causes audible tones?
3. What PWM frequency range is best for very slow rotation?

Bring the motor speed back to zero and then reset the counter, leaving the Reset Counter switch in its “off” state. Manually rotate the magnet on the back of the motor and observe the Counter Value display. How many counts do you see for one complete revolution of the magnet?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,

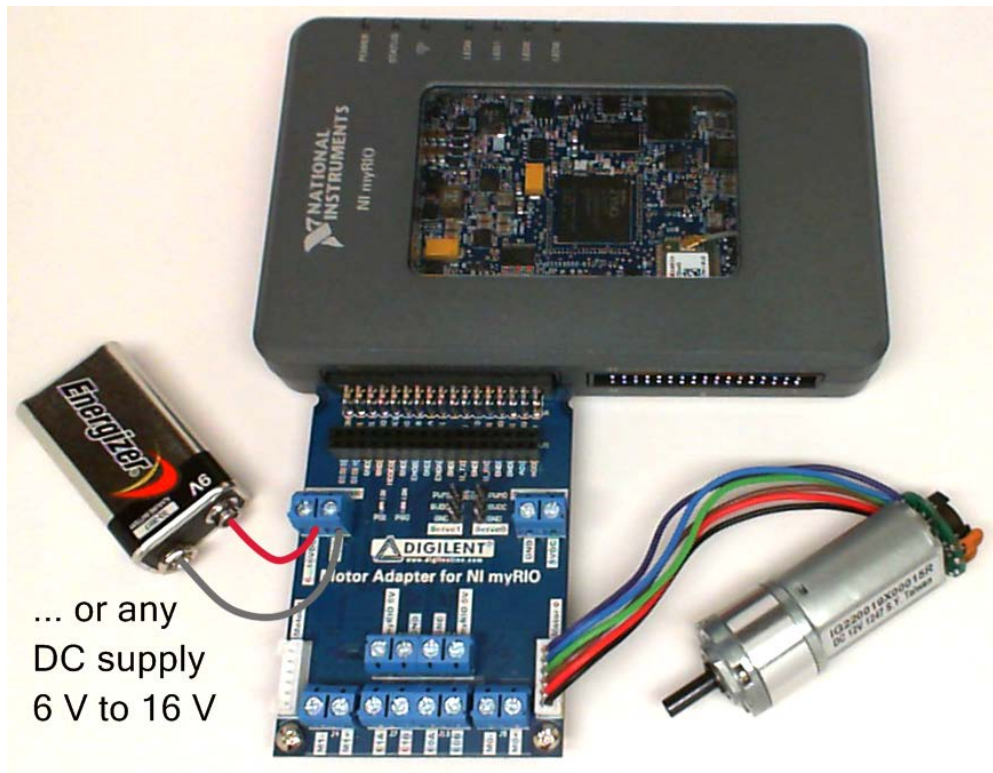


Figure 18.2: Demonstration circuit for motor adapter and geared motor: connected to NI myRIO MXP Connector A with motor and motor power supply connections.

- Correct MXP connector terminals — ensure that you are using Connector A for the motor adapter,
- Correct DC power supply terminals — double-check your connections, and ensure that you have not reversed the DC power supply terminals, and
- Glowing “power good” LED indicators (PG0 and PG1) on the motor adapter board.

## 18.2 Interface Theory

**Interface circuit:** The Motor Adapter for NI myRIO provides a flexible and trouble-free way to connect DC motors, servo motors, and stepper motors

to your myRIO. Study the video *Motor Adapter for NI myRIO and Geared Motor Interfacing Theory* ([youtu.be/bkiopa5nxiE](https://youtu.be/bkiopa5nxiE), 8:54) to learn more about the features of this board as well as the geared motor, to understand the high-level functional behavior of the adapter, to learn how to connect motors and power supplies to the adapter, and to become aware of the various control and feedback/status signals offered at the MXP interface.

Study the video *H-Bridge and Geared Motor Interfacing Theory* ([youtu.be/W526ekpR8q4](https://youtu.be/W526ekpR8q4), 11:26) to learn more about H-bridge principles of operation, speed control with pulse-width modulation (PWM), Hall-effect sensors and quadrature-encoded

sensor signals, and an overview of closed-loop control of the motor position and speed by using the sensor signals as feedback. This video discusses the PmodHB5 H-bridge specifically, but all of the theoretical concepts apply to the motor adapter, as well. See Chapter 35 on page 165 for more details.

**LabVIEW programming:** Study the video *PWM Express VI* ([youtu.be/mVN9jfwXleI](https://youtu.be/mVN9jfwXleI), 2:41) to learn how to use the PWM Express VI to create a pulse-width modulated square wave to provide variable-speed motor operation.

### 18.3 Basic Modifications

Study the video *Motor Adapter for NI myRIO Demo Walk-Through* ([youtu.be/wg8FQ7kEmdE](https://youtu.be/wg8FQ7kEmdE), 8:08) to learn the design principles of Motor Adapter & Geared Motor demo, and then try making these modifications to the block diagram of `Main.vi`:

- Determine the number of shaft encoder counts for one revolution of the gearbox shaft output (hint: you need to know the gearbox ratio and the number of encoder counts for one revolution of the sensor magnet), and then add block diagram code to display the gearbox output shaft angle in degrees.
  - Add a display to show the gearbox output shaft RPS (revolutions per second).
  - Add front-panel meters for motor voltage and current; use the "DC" output of the AC & DC Estimator.vi located in the Signal Processing | Signal Operation subpalette. Note that this VI requires at least three waveform periods to properly estimate the DC value.
  - Add a front-panel meter for motor power by taking the product of voltage and current measured in the previous step. Investigate the motor current, voltage, and power as you vary the motor loading.
- Try the servo motor connection; see Chapter 17 on page 69 for complete details. Remember that you can connect the servo motor power terminal block "5VDC" to the "myRIO 5V" to use myRIO to power your servo motor.
  - Add Boolean indicators for the fault ("FLT") and power-good ("PG") status signals.

### 18.4 For More Information

- Motor Adapter for NI myRIO Reference Manual* by Digilent ~ Reference manual for the motor adapter:  
[https://reference.digilentinc.com/ni:mxp\\_motor\\_adapter:refmanual](https://reference.digilentinc.com/ni:mxp_motor_adapter:refmanual)
- Motor Adapter for NI myRIO Schematic* by Digilent ~ Schematic diagram of the motor adapter:  
[https://reference.digilentinc.com/\\_media/ni:motoradapter\\_sch.pdf](https://reference.digilentinc.com/_media/ni:motoradapter_sch.pdf)
- A4973 Full-Bridge PWM Motor Driver* by Allegro ~ Datasheet for the H-bridge motor driver at the heart of the motor adapter:  
<http://www.allegromicro.com/en/Products/Motor-Driver-And-Interface-ICs/Brush-DC-Motor-Drivers/A4973.aspx>
- TPS24711 2.5V to 18V High Efficiency Power-Limiting Hot-Swap Controller* by Texas Instruments ~ Datasheet for the power manager used by the motor adapter:  
<http://www.ti.com/product/tps24711>
- Motor/Gearbox* by Digilent ~ Need more geared motors for your project? The IG22 is supplied by Digilent:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,403,625&Prod=MT-MOTOR>
- IG22 Geared Motor Datasheet* by Sha Yang Ye ~ Datasheet for the motor and planetary gearbox:  
<http://www.geared-motor.com/english/pdf/IG-22GM-01&02.pdf>
- Two-Channel Encoder Datasheet* by Sha Yang Ye ~ Datasheet for the two-channel quadrature

encoder:

[http://www.geared-motor.com/english/pdf/  
Magnetic-Encoders.pdf](http://www.geared-motor.com/english/pdf/Magnetic-Encoders.pdf)





# 19 IR Range Finder

An IR range finder uses a beam of reflected infrared light to sense the distance between the sensor and a reflective target. The range to an object is proportional to the reciprocal of the IR range finder's output voltage. Applications of an IR range finder include ranging and object detection for robots, proximity sensing, and touchless switches. Figure 19.1 pictures the IR range finder included with the NI Mechatronics Kit.



Figure 19.1: NI myRIO Mechatronics Kit rangefinderIR.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the features of the SHARP GP2Y0A21YK0F IR range finder,
2. Discuss the principle of operation of IR range finders, and
3. Apply calibration techniques using single or multiple measurements.

## 19.1 Component Demonstration

Follow these steps to demonstrate correct operation of the photocell.

**Select these parts from the NI myRIO Mechatronics Kit:**

- IR range finder, <http://www.digilentinc.com/Data/Products/IR-RANGE-SENSOR/IR%20Range%20Sensor%20rm.pdf>
- Jumper wires, M-F (3×)

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 19.2 on the next page. The interface circuit requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. 5-volt power supply → B/+5V (pin 1)
2. Ground → B/GND (pin 6)
3. Output signal → B/A10 (pin 3)

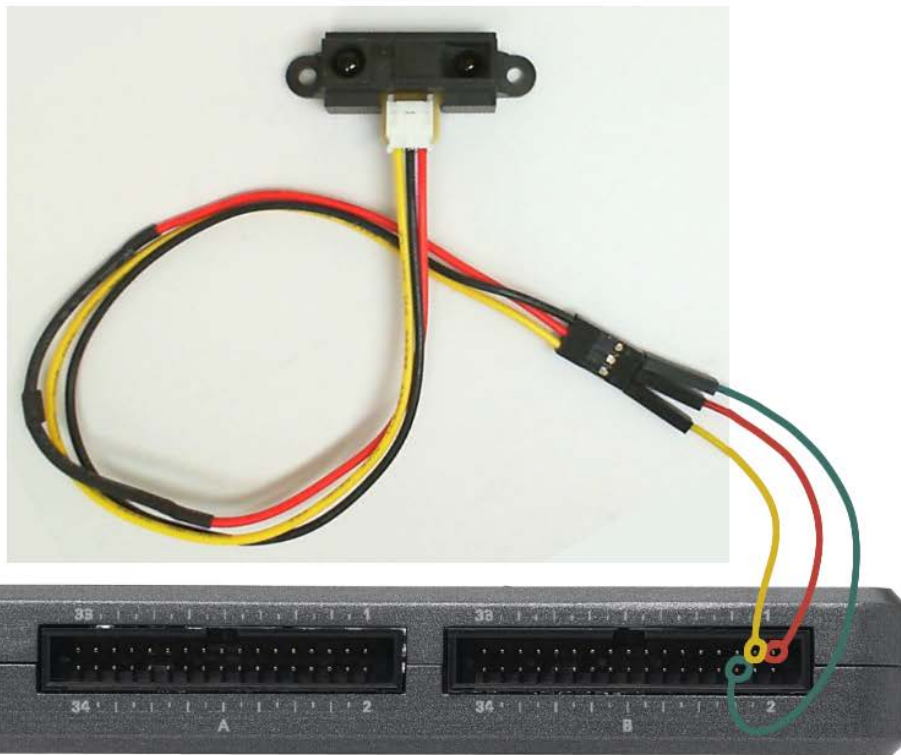
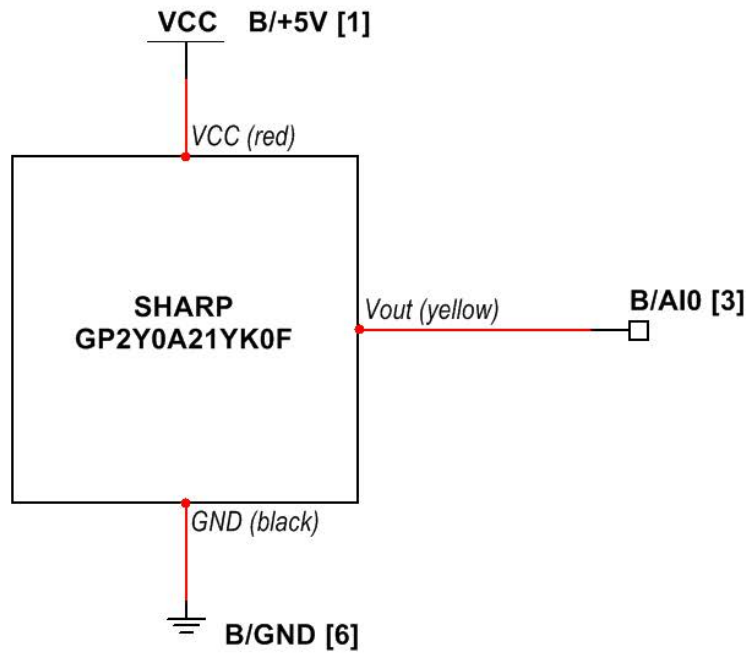


Figure 19.2: Demonstration setup for rangefinderIR connected to NI myRIO MXP Connector B.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project IR Range Finder demo.lvproj contained in the subfolder IR Range Finder demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays both the IR range finder output voltage and the range in centimeters to a reflective target. Use a meter stick or yard stick to measure the distance from the back side of the sensor to a reflective target placed somewhere between 0 cm and 80 cm. Compare the measured range to the known range; how well do these values match?

Record the sensor voltage  $V_O$  when the target is placed at a known range  $R$  of between 10 cm and 40 cm, calculate the calibration scale factor  $K_{\text{scale}} = R \times V_O$ , and then enter this value into the Kscale [cm-V] front-panel control. Repeat your earlier range measurements; do you observe improved measurement accuracy?

Try moving the target to ranges closer than 10 cm. You should observe that the measured range begins to increase even though the target is obviously very

close to the sensor. You can mitigate this non-ideal behavior by ensuring that the sensor is mounted at least 10 cm from the nearest target. With a robot platform, for example, mount the sensor towards the interior of the platform rather than at the edge.

Use a smartphone camera or webcam to view the IR range sensor. Infrared light appears blue on such a digital image sensor. Which side of the sensor is the IR emitter?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

## 19.2 Interface Theory

**Interface circuit:** The SHARP GP2Y0A21YK0F IR range finder creates an output voltage  $V_O$  that varies in inverse proportion with range to a reflective target.

Study the video *IR Range Finder Interfacing Theory* ([youtu.be/Xwr-j-2WT3k](https://youtu.be/Xwr-j-2WT3k), 9:59) to learn more about the IR range finder including features, principle of operation, calculating range from sensor voltage based on the geometric principle of similar triangles, and calibrating the sensor with single- or multi-measurement techniques.

**LabVIEW programming:** Study the video *Analog Input Express VI* ([youtu.be/N6Mi-VjBlmc](https://youtu.be/N6Mi-VjBlmc), 2:00) to learn how to use Analog Input Express VI to measure the voltage divider’s primary output.

### 19.3 Basic Modifications

Study the video *IR Range Finder Demo Walk-Through* ([youtu.be/BFge1RQxJ\\_E](https://youtu.be/BFge1RQxJ_E), 3:51) to learn the design principles of IR Range Finder demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add a Boolean control to disable or enable the averaging function.
2. Display the range in inches instead of centimeters.
3. Add a proximity detection feature: activate an onboard LED when the range is lower than a threshold value entered on the front panel.
4. Add an “out of range” Boolean indicator when the range exceeds 80 cm.
5. Add an enhanced proximity detection feature: one of three onboard LEDs to indicate “in range,” “too close,” or “too far.”

### 19.4 Integrated Project Ideas

Now that you know how to use the `rangefinderIR` consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Scanning Sensor* (51)

### 19.5 For More Information

- *IR Range Sensor Reference Manual* by Digilent ~ Reference manual for the IR range finder:  
<http://www.digilentinc.com/Data/Products/IR-RANGE-SENSOR/IR%20Range%20Sensor%20rm.pdf>
- *GP2Y0A21YK0F Datasheet* by SHARP Microelectronics ~ Datasheet for the GP2Y0A21YK0F Distance Measuring Sensor Unit:  
[http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk\\_e.pdf](http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf)

# 20 Sonic Range Finder

A sonic range finder measures the round trip time-of-flight of short pulses of sound just beyond human hearing (42 kHz) reflected back (echoed) from a target. Multiplying the time-of-flight by the speed of sound in air yields the distance (range) to the target. Applications of a sonic range finder include noncontact distance measurement, object detection, proximity sensing, and robotic world mapping. Figure 20.1 pictures the sonic range finder included with the NI Mechatronics Kit.

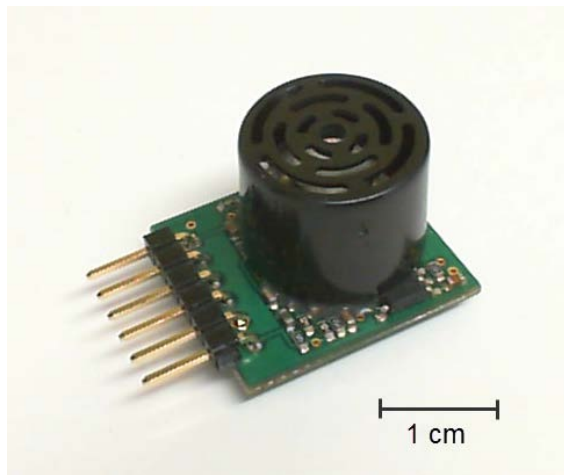


Figure 20.1: NI myRIO Mechatronics Kit sonic range finder.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the features of the MaxBotix MB1010 sonic range finder,
2. Determine the measured range from the UART and analog outputs,
3. Discuss the principle of operation of sonic range finders,
4. Compensate the measurement for air temperature, and
5. Understand the beam characteristics presented in the datasheet.

## 20.1 Component Demonstration

Follow these steps to demonstrate correct operation of the sonic range finder.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Sonic range finder, [http://maxbotix.com/documents/MB1010\\_Datasheet.pdf](http://maxbotix.com/documents/MB1010_Datasheet.pdf)
- Jumper wires, F-F (3×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 20.2 on page 87; the sonic range finder requires three connections to NI myRIO MXP Connector A (see Figure A.1 on page 233):

1. VCC → A/+3.3V (pin 33)
2. GND → A/GND (pin 30)
3. TX → A/UART.RX (pin 10)

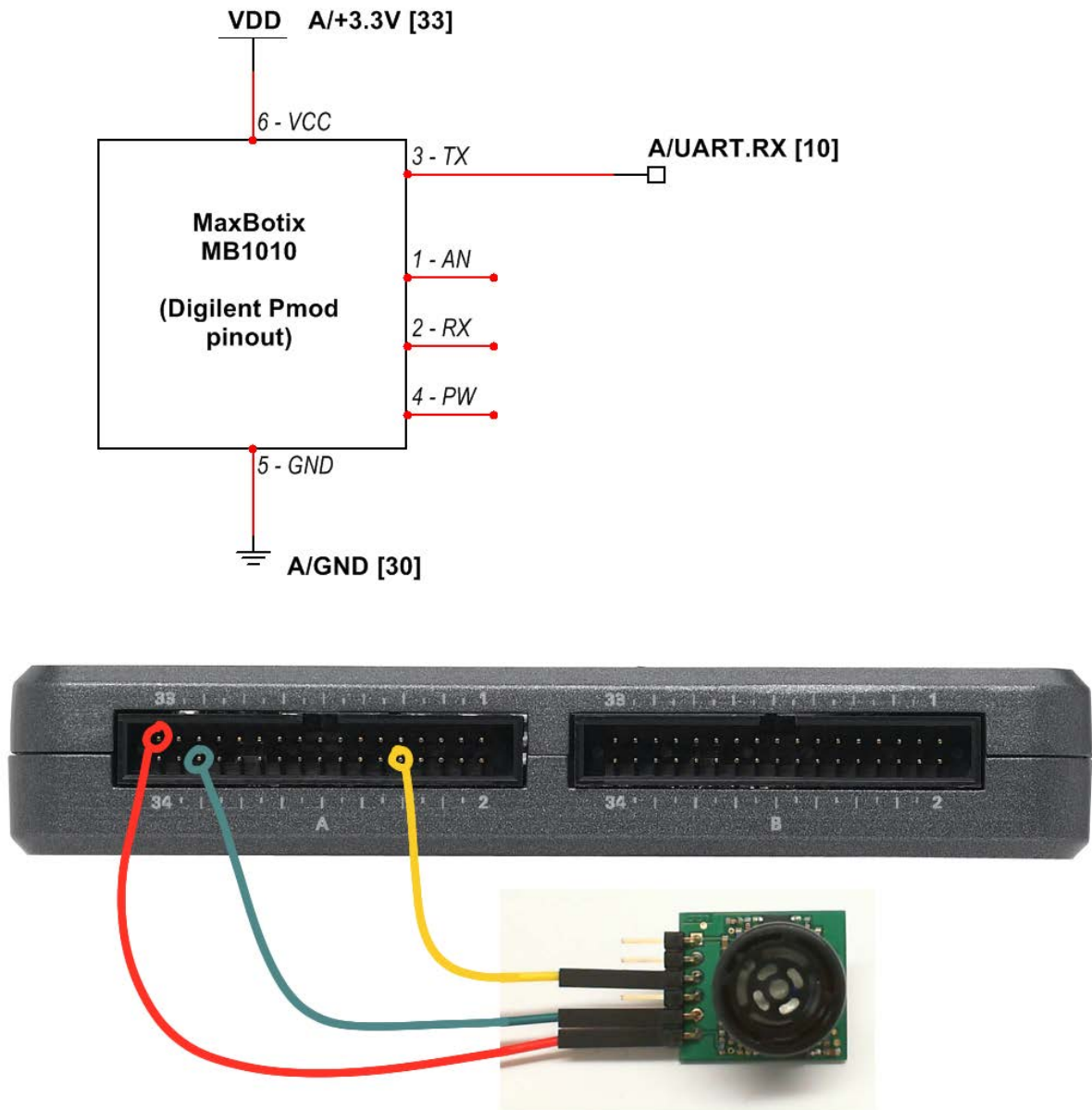


Figure 20.2: Demonstration setup for sonic range finder connected to NI myRIO MXP Connector A.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project Sonic Range Finder demo.lvproj contained in the subfolder Sonic Range Finder demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the measured range in inches as a horizontal fill slide and digital display. The output characters generated by the MB1010 appear in the lower left corner.

Allow at least 14 inches of clearance when you power on the range finder to allow its power-on calibration to work properly, and then try holding an object at known distances from the range finder. You should find that the range finder accurately displays the distance to the object.

Try holding the object very close to the transducer, i.e., closer than six inches. Note how the range finder reports a minimum range of six inches.

Double-click the upper limit of the horizontal fill slide and change its value to 254; this is the maximum range of the MB1010. Position the sonic range finder in an open area of at least 22 feet. Are you able to see the maximum possible range measurement?

Try objects of various thicknesses such as a pen or pencil. The range finder cannot “see” small-diameter objects unless they are sufficiently close. Can you determine the maximum detectable range?

Try placing an object off to the side. The range finder beam pattern looks a bit like a flashlight beam — narrow near the transducer and fanning out farther away. Can you determine the beam width at several different ranges?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A and that you have the correct pin connections, and
- Correct MB1010 connector terminals — double-check your connections, and ensure that you have connected the NI myRIO UART “RX” line to the MB1010 “TX” terminal; also check that you have not accidentally crossed the power supply connections, and

NOTE: Double-check the “RX” and “TX” connections again should you see the message “Timeout expired before operation completed” or similar; this message indicates that NI myRIO did not receive the expected characters from the MB1010 “TX” output.

## 20.2 Interface Theory

**Interface circuit:** The transducer of the MaxBotix MB1010 sonic range finder (also known as the LV-MaxSonar-EZ1) generates short pulses of sound that when echoed back from an object can be detected by the same transducer. The time-of-flight of the

pulse combined with the speed of sound yields the distance, or range, to the object. The MB1010 provides range measurements in digital format via UART, analog output, and pulse-width output.

Study the video *Sonic Range Finder Interfacing Theory* ([youtu.be/UcpmrcJR\\_D8](http://youtu.be/UcpmrcJR_D8), 9:26) to learn about the sonic range finder principle of operation, applications, MaxBotix MB1010 features, signal connections, temperature compensation technique, and beam pattern.

Study *Serial Communication: UART* ([youtu.be/odN66E85J5E](http://youtu.be/odN66E85J5E), 7:56) to better understand the signaling waveforms between UART transmitters and receivers.

### 20.3 Basic Modifications

Study the video *Sonic Range Finder Demo Walk-Through* ([youtu.be/-MVa9Hk1gKI](http://youtu.be/-MVa9Hk1gKI), 3:33) to learn the design principles of *Sonic Range Finder* demo, and then try making these modifications to the block diagram of `Main.vi`:

- Update the front panel to display the range in centimeters.
- Add a proximity detector (Boolean indicator or onboard LED) with an adjustable threshold level control on the front panel. For example, when the threshold is set to 20 inches, any object ranging less than 20 inches would activate the proximity detector indicator.
- Add an Analog Input Express VI to display the analog output “AN” of the MB1010. Convert the voltage to range and then compare this value to the UART value. Do you find any difference between using the analog output and the digital output?
- Create an on-demand version of the VI: add two Digital Output Express VIs to hold the MB1010 “RX” normally low with a short high-level pulse to trigger a single measurement.

### 20.4 Integrated Project Ideas

Now that you know how to use the sonic range finder consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Scanning Sensor* (51)

### 20.5 For More Information

- *MB1010 Datasheet* by MaxBotix ~ Datasheet for the MB1010 Sonic Range Finder:

[http://maxbotix.com/documents/MB1010\\_Datasheet.pdf](http://maxbotix.com/documents/MB1010_Datasheet.pdf)

- *Temperature Compensation for Sonic Range Finders* by MaxBotix ~ The speed of sound varies significantly with temperature. This page presents an equation for calculating speed of sound for any given temperature that can be used as the basis of temperature compensation:

[http://www.maxbotix.com/documents/Temperature\\_Compensation.pdf](http://www.maxbotix.com/documents/Temperature_Compensation.pdf)

- *Sonic Range Finder Tutorials* by MaxBotix ~ Tutorials on all aspects of sonic range finders:

<http://maxbotix.com/tutorials.htm>



# 21 Accelerometer

An accelerometer measures the position of an on-chip suspended *proof mass* and reports motion of the proof mass as acceleration in “gees,” i.e.,  $1g = 9.81\text{m/s}^2$ . Accelerometers measure both static acceleration for leveling, tilt, and drop sensor applications as well as dynamic acceleration for shock and vibration data acquisition. Figure 21.1 pictures the NI myRIO Mechatronics Kit accelerometer based on the Analog Devices ADXL345 triple-axis digital accelerometer with I<sup>2</sup>C-bus serial communications. The ADXL345 provides a high degree of flexibility and includes on-chip event detection including single-tap and double-tap, activity, and free-fall.

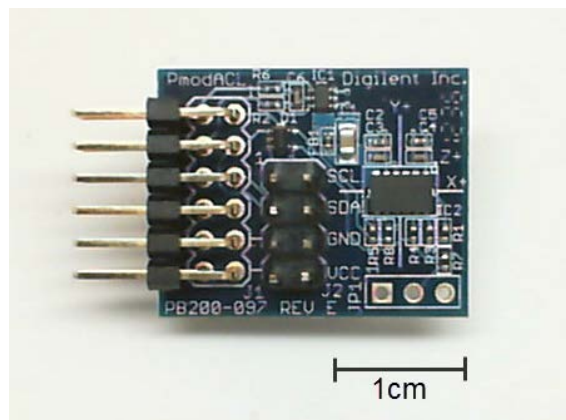


Figure 21.1: NI myRIO Mechatronics Kit accelerometer.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Configure the accelerometer for data rate, resolution, and range,
2. Set up single-tap detection and interrupt output pins, and
3. Read and display acceleration values.

## 21.1 Component Demonstration

Follow these steps to demonstrate correct operation of the accelerometer.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Accelerometer (PmodACL), [http://digilentinc.com/Data/Products/PMOD-ACL/PmodACL\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-ACL/PmodACL_rm.pdf)
- Jumper wires, F-F (6×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 21.2 on the next page; the accelerometer requires six connections to NI myRIO MXP Connector A (see Figure A.1 on page 233):

1. +3.3-volt supply → A/+3.3V (pin 33)
2. Ground → A/GND (pin 30)
3. Serial data (SDA) → A/I2C.SDA (pin 34)
4. Serial clock (SCL) → A/I2C.SCL (pin 32)
5. Interrupt #1 → A/DIO0 (pin 11)
6. Interrupt #2 → A/DIO0 (pin 13)

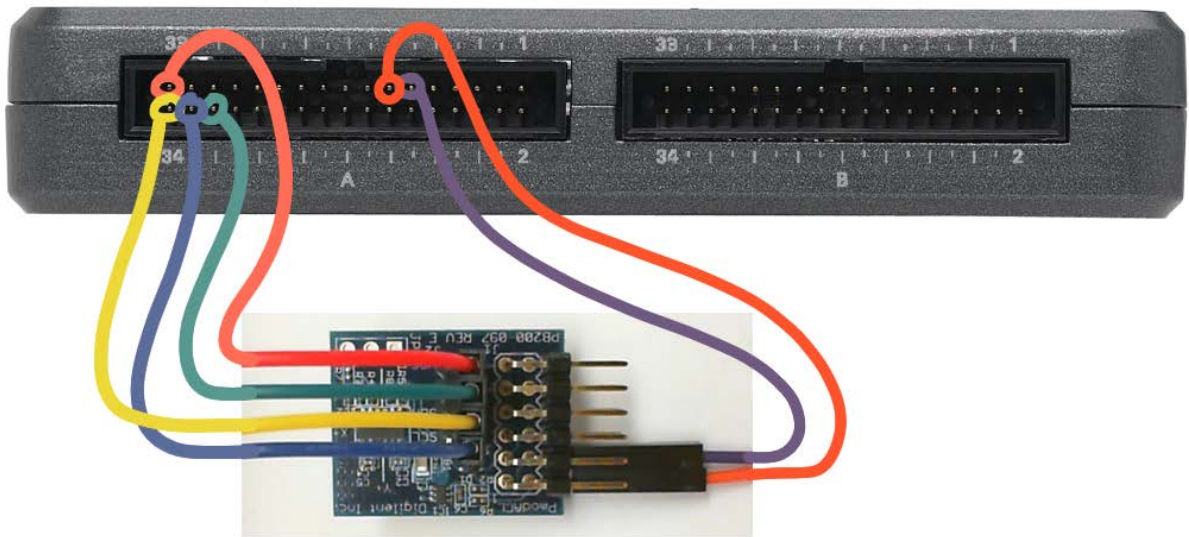
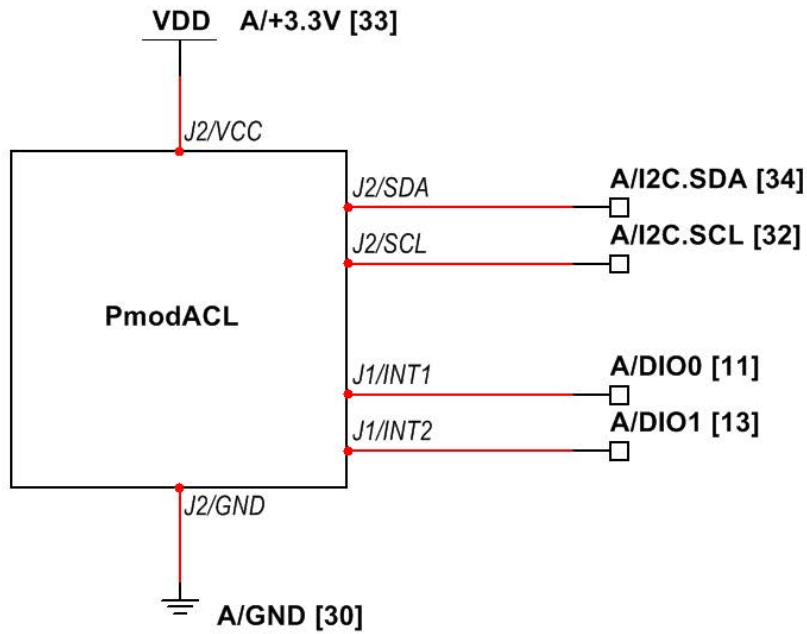


Figure 21.2: Demonstration setup for accelerometer connected to NI myRIO MXP Connector A.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project Accelerometer demo.lvproj contained in the subfolder Accelerometer demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the triple-axis accelerometer values in three formats: as the six bytes retrieved from the accelerometer “DATA” registers, as three signed integers formed by combining the two bytes retrieved per axis, and as a waveform chart. You will soon learn how to convert these values to “gees.” The VI also displays the contents of the “INTERRUPT\_SOURCE” register. Before running the main loop the VI configures the accelerometer registers for data rate, resolution, range, and single-tap detection on the X-axis.

Run the VI and then observe the front-panel display as you shake the accelerometer. Look carefully at the PmodACL board to see the “X+” and “Y+” coordinate system axes. Try shaking the accelerometer along a particular axis and then correlate this motion with what you see on the front panel. The coordinate system follows the right-hand rule, therefore the

“Z+” axis point up from the top of the board.

Next, try some static (constant) acceleration measurements that form the basis of level and tilt sensing applications. For example, place the edge of the board marked “X+” on a level surface, and then rock the board back and forth as you observe the front panel. What do you observe about the sign of the measured X-axis acceleration? Note the position of the board when the X-axis acceleration reaches its maximum deviation from zero. Can you form an idea of how a static acceleration measurement could be converted into a tilt sensor reporting its results in degrees off-center?

Lastly, try tapping the edge of the board on the table or tap the edge with your finger. The NI myRIO onboard LED0 will flash to indicate that a single-tap event was detected by the accelerometer. Single-tap detection is only enabled along the X-axis. Try tapping on the three different axes as you observe the LED.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A and that you have the correct pin connections,
- Correct PmodACL connector terminals — double-check your connections, and ensure that you have connected the NI myRIO I<sup>2</sup>C-bus “SDA” line to the PmodACL “SDA” terminal on connector J2 and the “SCL” line to the “SCL” terminal; also check that you have not accidentally crossed the power supply connections, and
- Correct PmodACL interrupt terminals to the NI myRIO DIO lines — the front-panel only updates in response to the “data ready” interrupt.

**NOTE:** Double-check the “SDA” and “SCL” connections should you see the message “Error -36011 occurred at myRIO Write I2C.vi” or similar; this message indicates that NI myRIO did not receive an expected acknowledgement from the PmodACL I<sup>2</sup>C-bus interface.

## 21.2 Interface Theory

**Interface circuit:** The Analog Devices ADXL345 triple-axis accelerometer at the heart of the PmodACL supports the I<sup>2</sup>C-bus serial interface (commonly denoted “I2C”); the ADXL345 also supports the SPI serial interface, however, this chapter concentrates exclusively on the I<sup>2</sup>C-bus interface. The ADXL345 includes two interrupt outputs designated “INT1” and “INT2.” These pins provide access to eight different interrupt sources which may be enabled as needed. Thirty addressable registers provide access to the three acceleration measured values as well as a wide variety of configuration options.

Study the video *Accelerometer Interfacing Theory* ([youtu.be/uj76-JtT\\_xk](https://youtu.be/uj76-JtT_xk), 15:55) to learn about accelerometer principle of operation, applications, Analog Devices ADXL345 features, and Digilent PmodACL connections and sensor axis orientation. Continue studying the video to learn the necessary details to configure the ADXL345 registers and read accelerometer data. The ADXL345 is a relatively complex device offering a wide range of options; the video tutorial therefore concentrates on the subset of features that will quickly get you up and running. The detailed example shows you: how to set the data rate, resolution, and range; how set up single-tap threshold and duration values and direct the single-tap interrupt to an output pin; and how to read the three sets of data registers and convert these accelerometer values into measured acceleration in “g” units.

Study *Serial Communication: I2C*

([youtu.be/7CgNF78pYQM](https://youtu.be/7CgNF78pYQM), 8:47) to better understand the signaling waveforms between I<sup>2</sup>C-bus transmitters and receivers.

## 21.3 Basic Modifications

Study the video *Accelerometer Demo Walk-Through* ([youtu.be/-\\_GWEsrfxU4](https://youtu.be/-_GWEsrfxU4), 8:37) to learn the design principles of Accelerometer demo, and then try making these modifications to the block diagram of *Main.vi*:

1. Change the device address to some other value. What LabVIEW error message do you see?
2. Add the necessary computation to display the acceleration in “g” units.
3. Add code to read the ADXL345 device ID register (address 0x00) and to generate an error condition if the ID does not match the expected value; in this way your VI can detect that the correct I<sup>2</sup>C-bus device is attached to the NI myRIO I2C terminals. Study the ADXL345 datasheet register map (Table 19) to determine the expected value. Select one of the pop-up dialog box VIs from the Programming | Dialog & User Interface subpalette to display your error message.
4. Compare the accelerometer performance to that of the NI myRIO onboard accelerometer by setting up an additional set of front-panel indicators; see the Accelerometer Express VI located in the myRIO | Onboard subpalette. Use a piece of tape to affix the PmodACL to the myRIO case.

## 21.4 Integrated Project Ideas

Now that you know how to use the accelerometer consider integrating it with other devices to create a complete system, for example:

- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Digital Bubble Level* (57)

## 21.5 For More Information

- *PmodACL Reference Manual* by Digilent ~ Reference manual for the accelerometer:  
[http://digilentinc.com/Data/Products/PMOD-ACL/PmodACL\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-ACL/PmodACL_rm.pdf)
- *PmodACL Schematics* by Digilent ~ Schematic diagram of the accelerometer:  
[http://digilentinc.com/Data/Products/PMOD-ACL/PmodACL\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-ACL/PmodACL_sch.pdf)
- *ADXL345 Datasheet* by Analog Devices ~ Datasheet for the ADXL345 Digital Accelerometer:  
<http://www.analog.com/ADXL345>
- *UM10204 I<sup>2</sup>C-bus Specification and User Manual* by NXP Semiconductors ~ A complete treatment of the I<sup>2</sup>C-bus standard, including timing diagrams and multi-master systems:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)



# 22 Gyroscope

A gyroscope measures the angular velocity (rate of rotation) about an axis. The rate is reported in degrees per second and when integrated yields relative angular displacement. A three-axis gyroscope mounted to a robot platform yields 3-D attitude (pitch, roll, and yaw) and provides valuable information to maintain stability of the vehicle. Figure 22.1 pictures the NI myRIO Mechatronics Kit gyroscope based on the STMicroelectronics L3G4200D triple-axis digital-output gyroscope with I<sup>2</sup>C-bus serial communications.

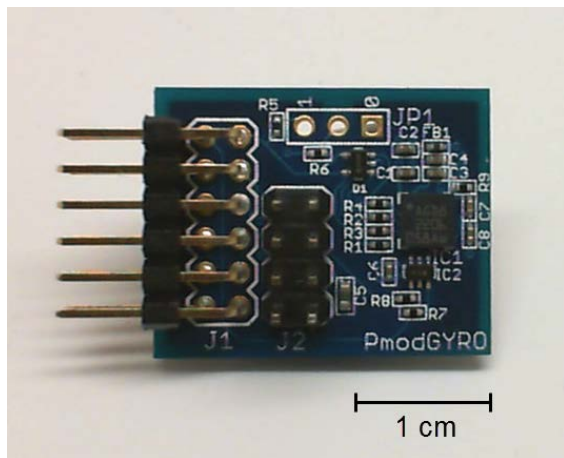


Figure 22.1: NI myRIO Mechatronics Kit gyroscope.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Configure the gyroscope for data rate and full-scale range,
2. Set up an interrupt output for onboard threshold limit detection,
3. Read and display angular rate values, and
4. Deal with non-zero values of zero-rate level output bias.

## 22.1 Component Demonstration

Follow these steps to demonstrate correct operation of the gyroscope.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Gyroscope (PmodGYRO), [https://www.digilentinc.com/Data/Products/PMOD-GYRO/PmodGYRO\\_rm\\_RevA.pdf](https://www.digilentinc.com/Data/Products/PMOD-GYRO/PmodGYRO_rm_RevA.pdf)
- Jumper wires, F-F (6×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 22.2 on page 97; the gyroscope requires six connections to NI myRIO MXP Connector A (see Figure A.1 on page 233):

1. +3.3-volt supply → A/+3.3V (pin 33)
2. Ground → A/GND (pin 30)
3. Serial data (SDA) → A/I2C.SDA (pin 34)
4. Serial clock (SCL) → A/I2C.SCL (pin 32)
5. Interrupt #2 → A/DIO0 (pin 11)

## 6. Interrupt #1 → A/DIO0 (pin 13)

### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Gyroscope demo.lvproj` contained in the subfolder `Gyroscope demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI shows the six-byte rate output from the L3G4200D gyroscope, formatted values as 16-bit signed integers, and a waveform chart to show the time history of the three gyroscope rates. The VI also shows the integrated rate values which yield relative angular displacement. The integrators may be reset to zero by clicking `reset integrator` and then clicking again.

Hold the PmodGYRO board right side up and facing you with the word “PmodGYRO” in the lower right corner. The X-axis points to the right, the Y-axis point up, and the Z-axis points out of the board toward you. Rotate the board about each of these three axes. You should observe that counter-clockwise rotation about the sensitive axis correspond to positive values in the angular rate

displays at the top of the VI front panel. What do you observe as you vary the rotation rate?

The lower-half of the VI front panel contains the angular displacement displays. Hold the PmodGYRO board in an initial position and then click the `reset integrator` two times. Rotate the board around a single axis and then rotate it back to the original position. You should observe that the displacement value becomes a large positive (or negative) value away from zero displacement and then returns to the same value.

The L3G4200D gyroscope output contains a non-ideal effect called *zero-rate level*. Set the PmodGYRO on a table undisturbed and then reset the integrator. The rate values on top may be difficult to read due to the rapid data rate, but you should expect to see the displacement values on the bottom begin a linear ramp away from zero. Even when the gyroscope is perfectly stationary (this is the “zero rate” input) the output contains some residual non-zero value (“zero-rate level”) which causes the integrator output magnitude to increase indefinitely. You can estimate the zero-rate level as follows: reset the integrator, allow a relatively long time to elapse, e.g., one minute (60 seconds), stop the VI, and then divide the measured displacement by 60 seconds. Compare this value to the typical value you see for the rate indicator. Note that you can adjust the limits of the angular rate plot by double-clicking the upper and lower values on the “amplitude” axis. Select a narrower range to better appreciate the output characteristics of the gyroscope including the zero-rate level offset and high-frequency noise level.

LED0 on the NI myRIO displays the Z-axis HIGH-limit threshold interrupt generated by the gyroscope. Try a quick rotation about the Z-axis to activate this LED.

Click the `Stop` button or press the escape key to stop the VI and to reset NI myRIO.



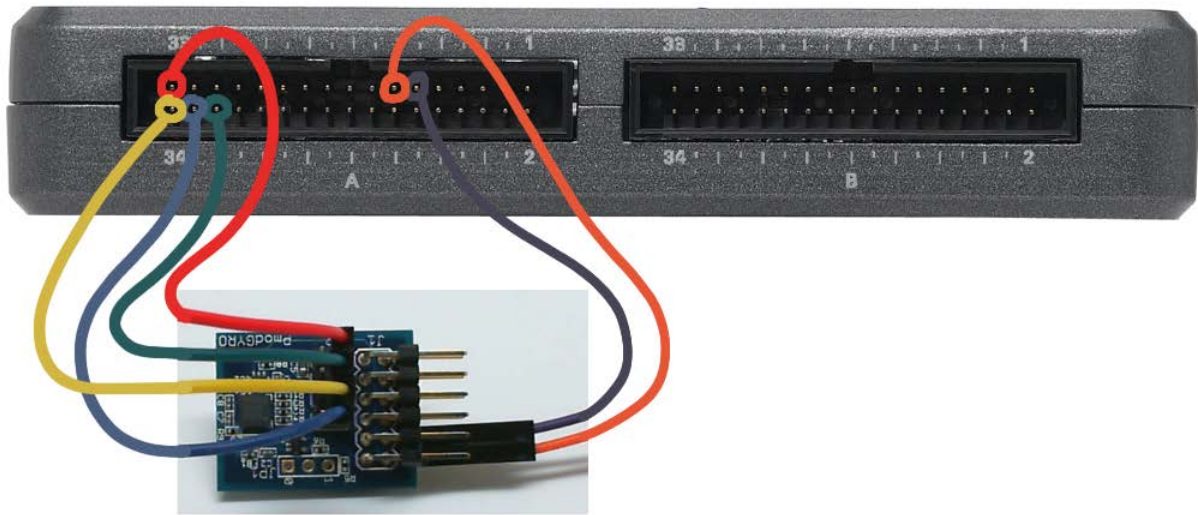
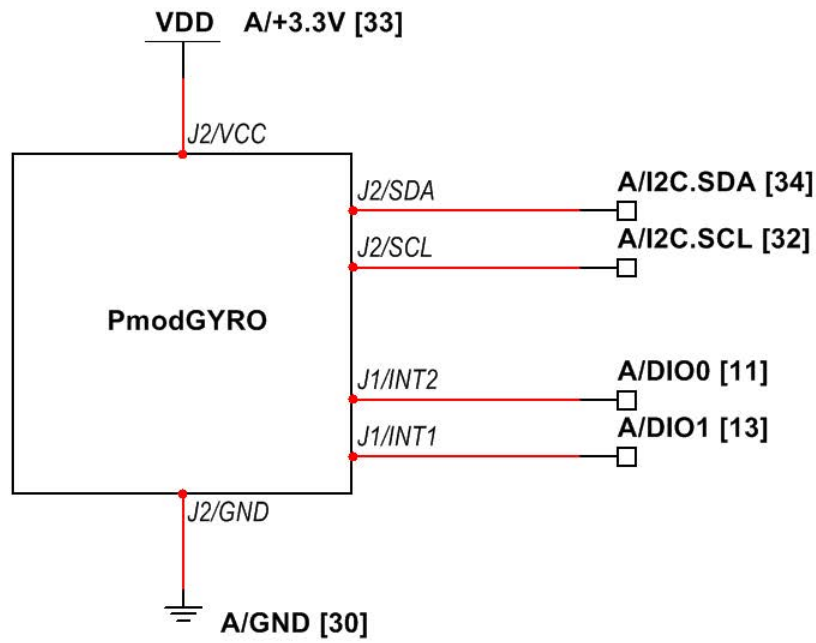


Figure 22.2: Demonstration setup for gyroscope connected to NI myRIO MXP Connector A.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A and that you have the correct pin connections,
- Correct PmodGYRO connector terminals — double-check your connections, and ensure that you have connected the NI myRIO I<sup>2</sup>C-bus “SDA” line to the PmoDAcL “SDA” terminal on connector J2 and the “SCL” line to the “SCL” terminal; also check that you have not accidentally crossed the power supply connections, and
- Correct PmodGYRO interrupt terminals to the NI myRIO DIO lines — the front-panel only updates in response to the “data ready” interrupt.

**NOTE:** Double-check the “SDA” and “SCL” connections should you see the message “Error -36011 occurred at myRIO Write I2C.vi” or similar; this message indicates that NI myRIO did not receiving an expected acknowledgement from the PmodGYRO I<sup>2</sup>C-bus interface.

## 22.2 Interface Theory

**Interface circuit:** The STMicroelectronics L3G4200D triple-axis gyroscope at the heart of the the PmodGYRO supports the I<sup>2</sup>C-bus serial interface (commonly denoted “I2C”); the L3G4200D also supports the SPI serial interface, however, this chapter concentrates exclusively on the I<sup>2</sup>C-bus interface. The L3G4200D includes two interrupt outputs designated “INT1” and “INT2.” These pins provide access to various interrupt sources which may be enabled as needed. Twenty-six addressable registers provide access to the three angular rate measured values as well as a wide variety of configuration options.

Study the video *Gyroscope Interfacing Theory* ([youtu.be/5JDkwG2rr1o](https://youtu.be/5JDkwG2rr1o), 13:48) to learn about gyroscope principles of operation, applications, STMicroelectronics L3G4200D features, and Digilent PmodGYRO connections and sensor axis orientation. Continue studying the video to learn the necessary details to configure the L3G4200D registers, read angular rate data, and integrate the rate measurements to obtain relative angular displacement. The L3G4200D is a relatively complex device offering a wide range of options; the video tutorial therefore concentrates on the subset of features that will quickly get you up and running. The detailed example shows you how to do the following:

- Set the data rate and full-scale range
- Set up a high-level threshold to generate a comparator-style interrupt on the “INT1” output pin
- Generate a “data ready” interrupt on the “INT2” output pin
- Read the three sets of data registers and convert these rate values into measured acceleration in “dps” (degrees per second) units
- Integrate the rate values to obtain relative angular displacement, and
- Estimate the zero-rate level offset to reduce integrator ramp-up.

Study *Serial Communication: I2C* ([youtu.be/7CgNF78pYQM](https://youtu.be/7CgNF78pYQM), 8:47) to better understand the signaling waveforms between I<sup>2</sup>C-bus transmitters and receivers.

## 22.3 Basic Modifications

Study the video *Gyroscope Demo Walk-Through* ([youtu.be/o\\_iuY0M3yDk](https://youtu.be/o_iuY0M3yDk), 6:36) to learn the design principles of Gyroscope demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Change the device address to some other value. What LabVIEW error message do you see?

2. Add the necessary computation to display the angular rate in “dps” units (degrees per second); refer to the theory video or the datasheet to obtain the necessary sensitivity scale factor to convert “LSB” (least-significant bit) units to dps units.
  3. Add code to read the “WHO\_AM\_I” device ID register
  4. Add code to read the L3G4200D device ID register (address 0x0F) and to generate an error condition if the ID does not match the expected value; in this way your VI can detect that the correct I<sup>2</sup>C-bus device is attached to the NI myRIO I2C terminals. Study the L3G4200D datasheet register map (Table 18) to determine the expected value. Select one of the pop-up dialog box VIs from the Programming | Dialog & User Interface subpalette to display your error message.
  5. Add the necessary computation to subtract the zero-rate level.
- *L3G4200D Datasheet* by STMicroelectronics ~ Datasheet for the L3G4200D Digital-Output Gyroscope:  
<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf>
  - *L3G4200D Datasheet* by STMicroelectronics ~ *TA0343: Everything about STMicroelectronics 3-axis digital MEMS gyroscopes* — This technical article provides an excellent discussion of the gyroscope principles of operation as well as techniques to deal with zero-rate level:  
[http://www.st.com/st-web-ui/static/active/en/resource/technical/document/technical\\_article/DM00034730.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/technical_article/DM00034730.pdf)
  - *UM10204 I<sup>2</sup>C-bus Specification and User Manual* by NXP Semiconductors ~ A complete treatment of the I<sup>2</sup>C-bus standard, including timing diagrams and multi-master systems:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)

## 22.4 Integrated Project Ideas

Now that you know how to use the gyroscope consider integrating it with other devices to create a complete system, for example:

- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *3-D Color Controller* (46)

## 22.5 For More Information

- *PmodGYRO Reference Manual* by Digilent ~ Reference manual for the gyroscope:  
[http://digilentinc.com/Data/Products/PMOD-GYRO/PmodGYRO\\_rm\\_RevA.pdf](http://digilentinc.com/Data/Products/PMOD-GYRO/PmodGYRO_rm_RevA.pdf)
- *PmodGYRO Schematics* by Digilent ~ Schematic diagram of the gyroscope:  
[http://digilentinc.com/Data/Products/PMOD-GYRO/PmodGYRO\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-GYRO/PmodGYRO_sch.pdf)



# 23 Compass

The familiar compass with its magnetic north-seeking needle has a digital counterpart in the Honeywell HMC5883L three-axis compass at the heart of the Digilent PmodCMPS pictured in Figure 23.1 and included in the NI myRIO Mechatronics Kit. The HMC5883L generates 12-bit measurements delivered by I<sup>2</sup>C-bus serial communications at rates of up to 160 Hz and offers multiple sensitivity ranges for compassing (finding the Earth's magnetic north pole) or to measure stronger magnetic fields up to 8 gauss.

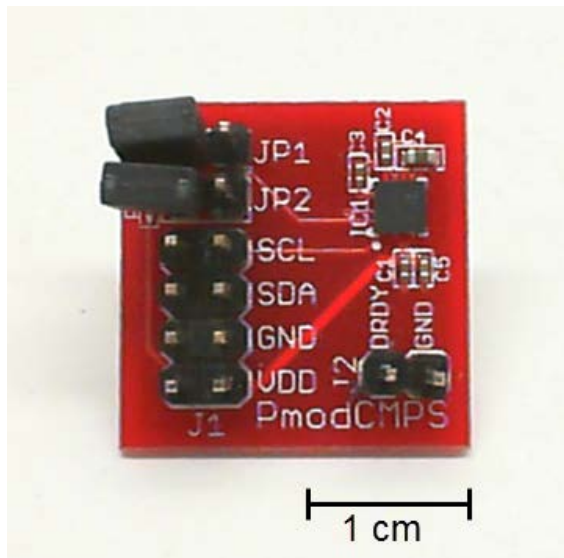


Figure 23.1: NI myRIO Mechatronics Kit compass.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss compassing principles including Earth's magnetic field inclination angle and declination angle (location-dependent difference between magnetic north and true north),
2. Interpret the three-axis compass output to find true north direction,
3. Configure the compass for data rate, range, and continuous measurements, and
4. Read and display compass direction.

## 23.1 Component Demonstration

Follow these steps to demonstrate correct operation of the compass.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Compass (PmodCMPS), <http://digilentinc.com/Data/Products/PMOD-CMPS/PmodCMPS-rm-revA1.pdf>
- Jumper wires, F-F (5×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 23.2 on page 103; the compass requires five connections to NI myRIO MXP Connector A (see Figure A.1 on page 233):

1. +3.3-volt supply (VDD) → A/+3.3V (pin 33)
2. Ground (GND) → A/GND (pin 30)
3. Serial data (SDA) → A/I2C.SDA (pin 34)
4. Serial clock (SCL) → A/I2C.SCL (pin 32)

#### 5. Data ready (DRDY) #1 → A/DIO0 (pin 11)

NOTE: Leave both jumpers JP1 and JP2 disconnected; these jumpers enable 2.2 k $\Omega$  pull-up resistors that are already included on the NI myRIO SCA and SCL lines.

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Compass demo.lvproj` contained in the subfolder `Compass demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the three-axis compass values in three formats: as the six bytes retrieved from the compass “DATA” registers, as three signed integers formed by combining the two bytes retrieved per axis, and as a waveform chart. You will soon learn how to convert these values to magnetic field strength in gauss. Before entering the main loop the VI configures the compass registers for averaging mode, data rate, range, and continuous-measurement mode.

Run the VI and then observe the front-panel display as you move and rotate the compass. Hold the board component-side up and level with the

ground, and then rotate the board while observing the X value; when X reaches its maximum positive value (with the board still level) the right side of the board points to magnetic north.

Hold the PmodCMPS board component-side up and level with the ground. Gradually tilt the board until the Z-axis reaches either a positive or negative extreme value and the X and Y axes indicate zero. The board is now normal to the Earth’s magnetic field direction, i.e., the Z-axis is in parallel with the Earth’s magnetic field lines. The compass sensor yields a positive Z value when the field direction is out of the component-side of the board, therefore you will see a negative value if you are located in the northern hemisphere and a positive value if you are in the southern hemisphere. The angle of the board indicates the magnetic field *inclination angle* which is 90 degrees at the north and south magnetic poles and a more acute (smaller) angle elsewhere.

Try holding the board near a magnet. How does the magnet’s field strength compare to that of the Earth’s field? If you like, you can change the sensitivity range to prevent the values from saturating. Stop the VI, open the block diagram (press Ctrl+E), set all three gain configuration bits of Configuration Register B to one (Boolean “true”), go back to the front panel (Ctrl+E again), and re-run the VI.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A and that you have the correct pin connections,
- Correct PmodCMPS connector terminals — double-check your connections, and ensure that you have connected the NI myRIO I<sup>2</sup>C-bus

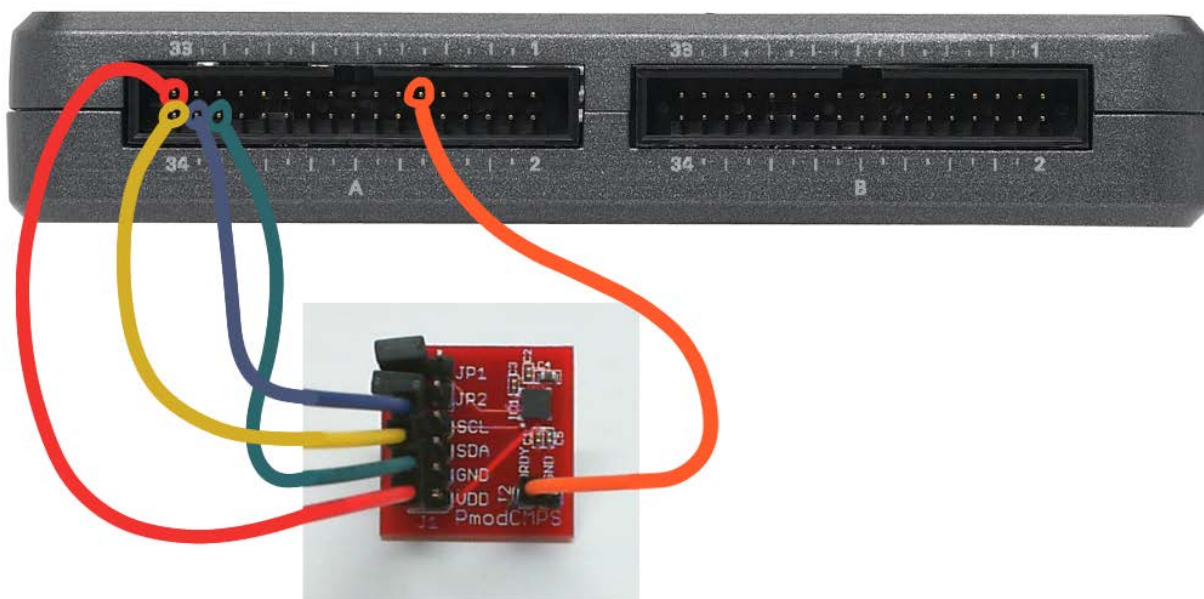
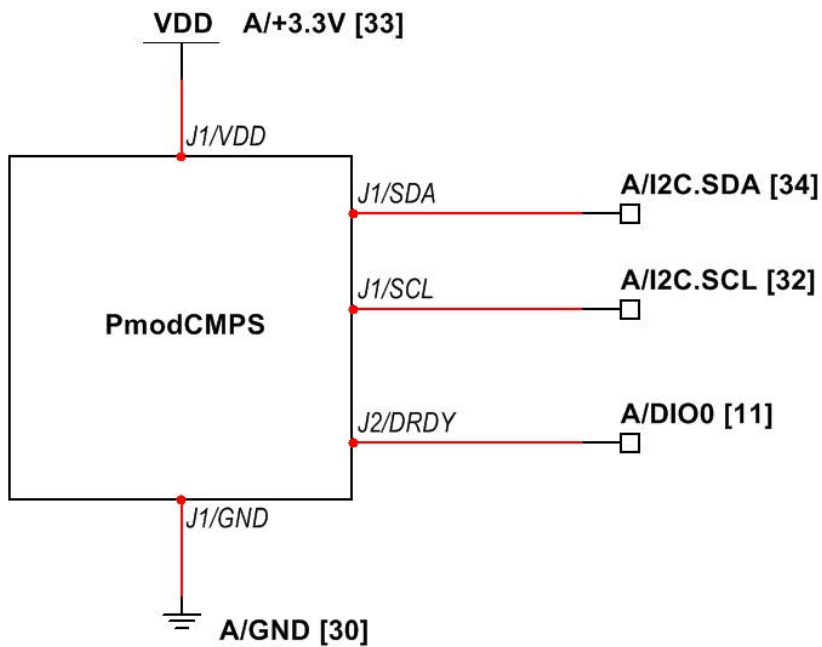


Figure 23.2: Demonstration setup for compass connected to NI myRIO MXP Connector A.

“SDA” line to the PmodCMPS “SDA” terminal on connector J1 and the “SCL” line to the “SCL” terminal; also check that you have not accidentally crossed the power supply connections, and

- Correct PmodCMPS data ready “DRDY” terminal to the NI myRIO DIO line — the front-panel only updates in response to the “data ready” interrupt.

**NOTE:** Double-check the “SDA” and “SCL” connections should you see the message “Error -36011 occurred at myRIO Write I2C.vi” or similar; this message indicates that NI myRIO did not receiving an expected acknowledgement from the PmodCMPS I<sup>2</sup>C-bus interface.

## 23.2 Interface Theory

**Interface circuit:** The Honeywell HMC5883L three-axis compass at the heart of the PmodCMPS board supports the I<sup>2</sup>C-bus serial interface (commonly denoted “I2C”). The HMC5883L includes a data-ready output designated “DRDY.” Twelve addressable registers provide access to the three compass measured values as well as a variety of configuration options.

Study the video *Compass Interfacing Theory* ([youtu.be/3WkJ7ssZmEc](https://youtu.be/3WkJ7ssZmEc), 12:47) to learn about compass applications, Honeywell HMC5883L features, Digilent PmodCMPS connections and sensor axis orientation, and compassing principles: magnetic north vs. true north (declination angle), inclination angle, and compass output corrections to read true north. Continue studying the video to learn the necessary details to configure the HMC5883L registers and read compass data. The detailed example shows you how to set the data rate, averaging mode, sensitivity range, and continuous-measurement mode, and also how to read the three sets of data registers and convert these compass values into measured compass in “gauss” units.

Study *Serial Communication: I2C* ([youtu.be/7CgNF78pYQM](https://youtu.be/7CgNF78pYQM), 8:47) to better un-

derstand the signaling waveforms between I<sup>2</sup>C-bus transmitters and receivers.

## 23.3 Basic Modifications

Study the video *Compass Demo Walk-Through* ([youtu.be/bWew4fHWVKo](https://youtu.be/bWew4fHWVKo), 7:48) to learn the design principles of Compass demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Change the device address to some other value. What LabVIEW error message do you see?
2. Add the necessary computation to display the compass measurements in “gauss” units. Note that the scale factor depends on the gain setting in Configuration Register B.
3. Add the X-axis and Y-axis offset corrections described in the theory video tutorial; these are necessary to improve the accuracy of the compass.
4. Add calculations to convert the X and Y value to polar form and display the angle on a front-panel dial indicator; see the built-in `Re/Im To Polar VI`. Assuming that you have already added the offset corrections code (previous modification), compare the accuracy of your compass measurement to that of another instrument. You can find many compass apps for your smartphone, for example.
5. Add code to read the three HMC5883L device ID registers (addresses 0x0A through 0x0C) and to generate an error condition if the ID does not match the expected value; in this way your VI can detect that the correct I<sup>2</sup>C-bus device is attached to the NI myRIO I2C terminals. Study the HMC5883L datasheet register map (Tables 18 through 20) to determine the expected values. Select one of the pop-up dialog box VIs from the Programming | Dialog & User Interface subpalette to display your error message.



## 23.4 Integrated Project Ideas

Now that you know how to use the compass consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Compass with Tilt Correction* (54)

## 23.5 For More Information

- *PmodCMPS Reference Manual* by Digilent ~ Reference manual for the compass:  
<http://digilentinc.com/Data/Products/PMOD-CMPS/PmodCMPS-rm-revA1.pdf>
- *PmodCMPS Schematics* by Digilent ~ Schematic diagram of the compass:  
[http://digilentinc.com/Data/Products/PMOD-CMPS/PmodCMPS\\_A1\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-CMPS/PmodCMPS_A1_sch.pdf)
- *HMC5883L Datasheet* by Honeywell ~ Datasheet for the HMC5883L Three-Axis Compass; click the “HMC5883L Brochure” link at the bottom of the page:  
<http://www.magneticsensors.com/three-axis-digital-compass.php>
- *Compass Heading Using Magnetometers* by Honeywell ~ Converting the three-axis compass measurements to a heading angle; click the “Literature” link and then select “AN203” from the “Application Notes” menu:  
<http://www.magneticsensors.com/three-axis-digital-compass.php>
- *Reference Desing: Low Cost Compass* by Honeywell ~ More details on the design of a practical compass; click the “Literature” link and then select “AN214” from the “Application Notes” menu:  
<http://www.magneticsensors.com/three-axis-digital-compass.php>
- *Applications of Magnetic Sensors for Low Cost Compass Systems* by Honeywell ~ Detailed discussion of compass design including compensation of error effects; click the “Literature” link and then select this article from the “Technical Articles” menu:  
<http://www.magneticsensors.com/three-axis-digital-compass.php>
- *Magnetic Declination* by NOAA National Geophysical Data Center ~ Magnetic declination angle maps for the US and world:  
<http://www.ngdc.noaa.gov/geomag/declination.shtml>
- *Estimated Values of Magnetic Field Properties* by NOAA National Geophysical Data Center ~ Look up your declination angle (difference between magnetic north and true north) and inclination angle (angle of Earth’s magnetic field) based on your US zip code or your city and country. Select either “Declination” or “Inclination” in the “Magnetic component” menu:  
<http://www.ngdc.noaa.gov/geomag-web/#igrfwmm>
- *UM10204 I<sup>2</sup>C-bus Specification and User Manual* by NXP Semiconductors ~ A complete treatment of the I<sup>2</sup>C-bus standard, including timing diagrams and multi-master systems:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)



# 24 Ambient Light Sensor

An ambient light sensor (ALS) provides a rudimentary “eye” that enables a robot to sense general lighting conditions, to track and follow a light source, and to follow a painted line pathway. The PmodALS pictured in Figure 24.1 integrates a phototransistor sensor with an 8-bit analog-to-digital converter (ADC) that communicates via the SPI serial bus.

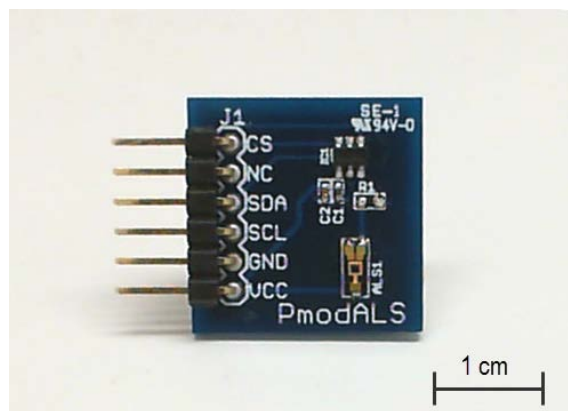


Figure 24.1: NI myRIO Mechatronics Kit ambient light sensor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the principles of operation of the phototransistor and analog-to-digital converter (ADS) combined on the PmodALS board,

2. State the timing requirements for the chip select and SPI clock, and
3. Interpret the format of the SPI data word output.

## 24.1 Component Demonstration

Follow these steps to demonstrate correct operation of the ambient light sensor.

**Select these parts from the NI myRIO Mechatronics Kit:**

- Ambient light sensor (PmodALS), <http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,1190&Prod=PMOD-ALS>
- Jumper wires, F-F (5×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 24.2 on the following page; the ambient light sensor requires five connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. VCC → B/+3.3V (pin 33)
2. GND → B/GND (pin 30)
3. SCL → B/SPI.CLK (pin 21)
4. SDA → B/SPI.MISO (pin 23)
5. CS → B/DIO0 (pin 11)

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done

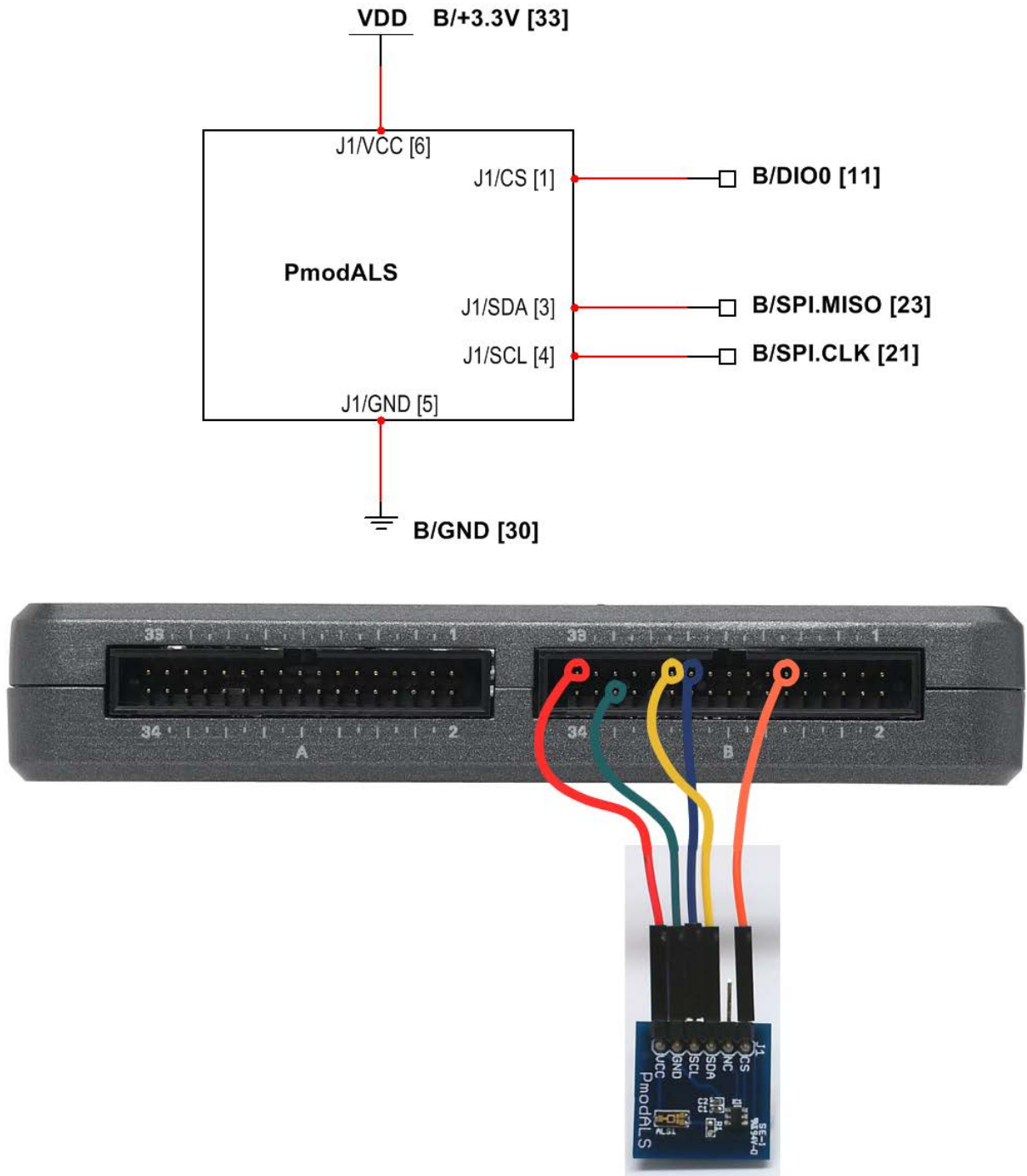


Figure 24.2: Demonstration setup for ambient light sensor connected to NI myRIO MXP Connector B.

so previously and unpack the contents to a convenient location,

- Open the project Ambient Light Sensor demo.lvproj contained in the subfolder Ambient Light Sensor demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the ambient light sensor 8-bit value as a dial gauge. Cover the light sensor (the component immediately above the “PmodALS” logo) and you should see the value drop towards zero. Shine a bright flashlight on the sensor to see the value increase towards the upper limit of 255. Expect to see a random variation of about  $\pm 1$  LSB (least significant bit).

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals, and
- Correct SPI connector terminals — double-check your connections, and ensure that you have connected the NI myRIO SPI “MISO” input to the

ambient light sensor “SDA” output and digital output DIO0 to the chip select input.

## 24.2 Interface Theory

**Interface circuit:** The Digilent PmodALS board combines the Vishay Semiconductors TEMT6000X01 phototransistor and the Texas Instruments ADC081S021 analog-to-digital converter (ADC). The ADC converts the phototransistor output voltage to an 8-bit value and transmits this measurement via SPI (serial peripheral interface).

Study the video *Ambient Light Sensor Interfacing Theory* ([youtu.be/zKnn1SskqRQ](https://youtu.be/zKnn1SskqRQ), 7:38) to learn more about the ambient light sensor theory of operation, SPI bus and chip select timing, and proper setup of the SPI Express VI to read sensor measurements. Study *Serial Communication: SPI* ([youtu.be/GaXtDamw5As](https://youtu.be/GaXtDamw5As), 7:02) to understand how the SPI Express VI configuration options relate to the signaling waveforms between SPI transmitters and receivers. In particular, see the discussion beginning at 4:29 to better understand the SPI Express VI “Advanced Options” for clock phase and polarity; the ADC081S021 requires “Trailing” edge clock phase and “High” clock polarity.

**LabVIEW programming:** Study the video *SPI Express VI* ([youtu.be/S7KkTeMfmc8](https://youtu.be/S7KkTeMfmc8), 5:51) to learn how to use the SPI Express VI.

## 24.3 Basic Modifications

Study the video *Ambient Light Sensor Demo Walk-Through* ([youtu.be/XcwEjM6TOig](https://youtu.be/XcwEjM6TOig), 3:02) to learn the design principles of Ambient Light Sensor demo, and then try making these modifications to the block diagram of Main.vi:

1. Display the ambient light level as a normalized value between 0 and 1.

2. Add a front-panel Boolean indicator or connection to the NI myRIO on-board LED that activates whenever the ambient light level exceeds a user-defined threshold.
  3. Convert the Wait constant to a front-panel control. Experiment with different loop times.
- *M68HC11 Reference Manual* by Freescale Semiconductor ~ Refer to Section 8 for a complete treatment of the SPI serial bus standard, including timing diagrams and multi-master systems:  
[http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/M68HC11RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf)

## 24.4 Integrated Project Ideas

Now that you know how to use the ambient light sensor consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Scanning Sensor* (51)
- *Music Maker* (56)
- *Weather Station* (58)

## 24.5 For More Information

- *PmodALS Reference Manual* by Digilent ~ Reference manual for the ambient light sensor board:  
<http://digilentinc.com/Data/Products/PMOD-ALS/PmodALS-rm-RevA.pdf>
- *PmodALS Schematics* by Digilent ~ Schematic diagram of the ambient light sensor board:  
[http://digilentinc.com/Data/Products/PMOD-ALS/PmodALS\\_A.1\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-ALS/PmodALS_A.1_sch.pdf)
- *ADC081S021 Data Sheet* by Texas Instruments ~ Complete information on the ADC081S021 analog-to-digital converter on the PmodALS board:  
<http://www.ti.com/product/adc081s021>
- *TEMT6000X01 Data Sheet* by Vishay Semiconductors ~ Complete information on the TEMT6000X01 ambient light sensor (phototransistor) on the PmodALS board:  
<http://www.vishay.com/product?docid=81579>

## **Part III**

# **NI myRIO Embedded Systems Kit**





# 25 Keypad

A keypad provides an essential component for a human user interface. Figure 25.1 pictures the the NI myRIO Embedded Systems Kit keypad. The keypad pushbutton switches wired in a 4×4 matrix can be scanned to determine single and multiple key presses.



Figure 25.1: NI myRIO Embedded Systems Kit keypad.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the matrix connection used by the keypad switch array,
2. Use internal pull-up and pull-down resistors to eliminate additional components, and
3. Determine which multi-keypress patterns can be uniquely decoded.

## 25.1 Component Demonstration

Follow these steps to demonstrate correct operation of the keypad.

Select these parts from the NI myRIO Embedded Systems Kit:

- 4x4 matrix keypad (PmodKYPD), <http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,940&Prod=PModKYPD>
- Jumper wires, F-F (9×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 25.2 on page 115; the keypad requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply → B/+3.3V (pin 33)
2. Column line 1 → B/DIO0 (pin 11)
3. Column line 2 → B/DIO1 (pin 13)
4. Column line 3 → B/DIO2 (pin 15)
5. Column line 4 → B/DIO3 (pin 17)

6. Row line 1 → B/DIO4 (pin 19)
7. Row line 2 → B/DIO4 (pin 21)
8. Row line 3 → B/DIO4 (pin 23)
9. Row line 4 → B/DIO4 (pin 25)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Keypad demo.lvproj` contained in the subfolder `Keypad demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI shows the state of the keypad buttons as a 4×4 grid and as a 1-D Boolean array of scan results. Try a variety of single-key combinations and confirm that only one LED indicator is active. Also confirm that the position of the keypad button indicator exactly matches the position of the button you press. Next, try several two-key combinations. Do they all work properly?

Experiment with multi-key combinations involving three or more keys. You should find that some work as expected, i.e., the front-panel indicator properly corresponds to the actual keys pressed, while other combinations produce an incorrect display. Try to identify a pattern that explains when

a given multi-key combination will or will not lead to an incorrect display.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and
- Correct keypad connector terminals — double-check your connections, and ensure that you have not accidentally interchanged the row and column lines.

## 25.2 Interface Theory

**Interface circuit:** The keypad contains sixteen single-throw (SPST) pushbuttons arranged in 4×4 grid; refer to Chapter 4 on page 13 for a complete treatment of pushbutton interfacing to both the MXP and MSP connectors. Because the keypad normally receives a sequence of single-button presses and occasionally two- or even three-button combination presses, the matrix connection based on shared row lines and column lines requires only eight connections to the NI myRIO DIO lines instead of the perhaps more expected sixteen connections that would be required to properly sense all  $2^{16} = 65536$  possible switch open-closed patterns.

Study the video *Keypad Interfacing Theory* ([youtu.be/oj2-CYSnyo0](https://youtu.be/oj2-CYSnyo0), 13:10) to learn more about the matrix approach to keypad switch wiring, the reasoning behind the pull-up resistors on the row lines, the keypad scanning process of driving column lines with a low-level voltages and reading the row lines, and the rule that explains when a multi-key button combination will produce “ghost”

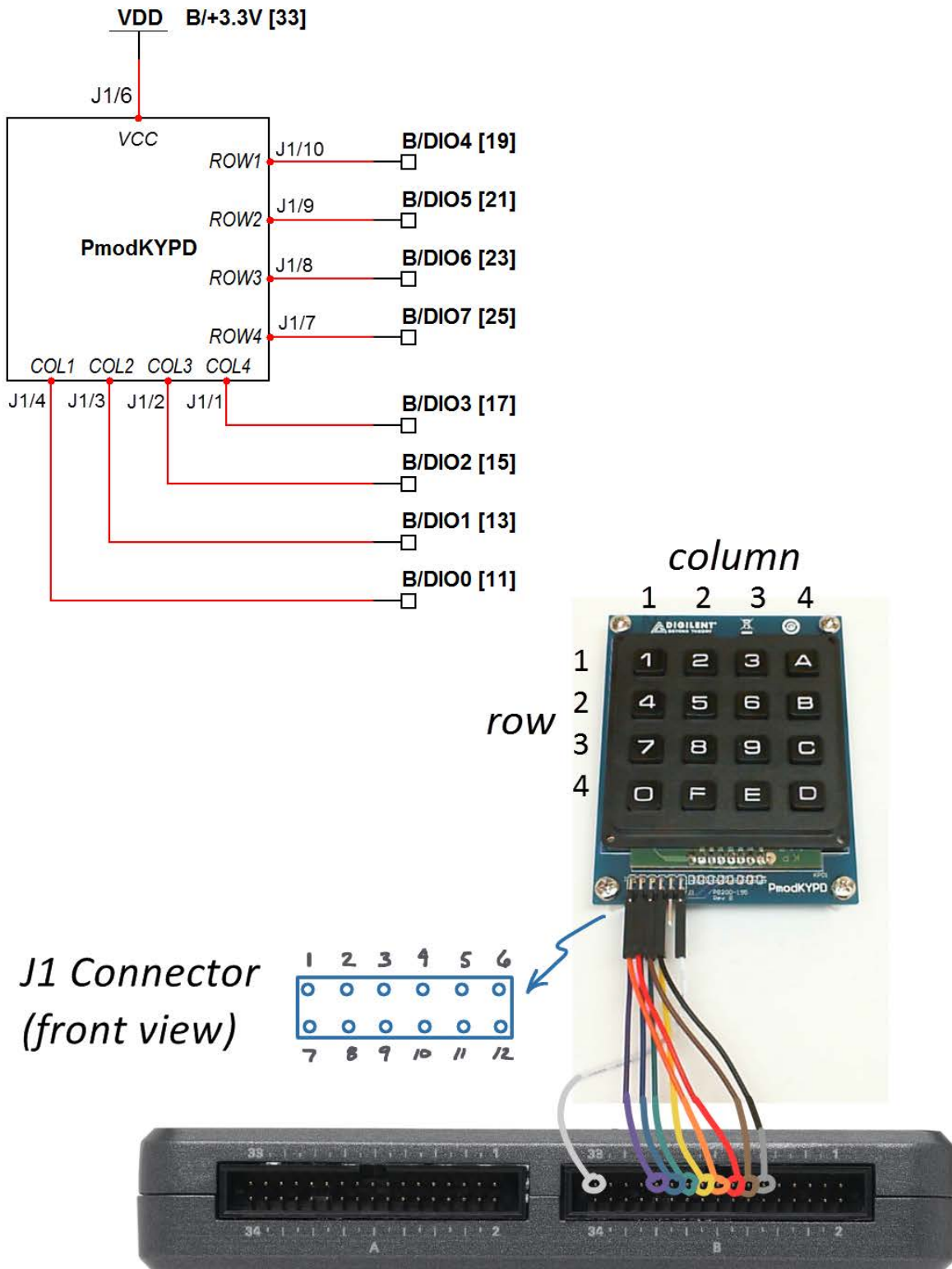


Figure 25.2: Demonstration setup for keypad connected to NI myRIO MXP Connector B.

keypresses (false reads). Look for the “challenge” offered in the video, and see if you can determine the maximum possible multi-key combination that does not produce “ghosting.”

**LabVIEW programming:** Study the video *Digital Output Low-Level VIs* ([youtu.be/WvnInG3ffqY](https://youtu.be/WvnInG3ffqY), 4:53) to learn how to use the low-level Digital Output VIs set the DIOs to either high impedance or to a defined voltage level.

### 25.3 Basic Modifications

Study the video *Keypad Demo Walk-Through* ([youtu.be/7r\\_LwcDa2AM](https://youtu.be/7r_LwcDa2AM), 4:58) to learn the design principles of Keypad demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Temporarily remove the Transpose 2D Array node and observe the new behavior of the Keypad buttons front-panel indicator. Thinking about the Scan display, can you explain how the transpose operation corrects the Keypad buttons information display?
2. Add the Boolean Array to Number node just before the Scan indicator connection, and then create a numerical indicator. Try various single-button and multi-button keypress combinations until you understand the relationship between the keypress and the indicated numerical value.
3. Create a numerical indicator whose value matches the value printed on each button of the keypad, with the letters corresponding to hexadecimal values; the indicator should display  $-1$  when no key is pressed. Consider using the Boolean Array to Number node and a case structure.
4. Re-connect the display to the MSP connector and updated the DIO values accordingly. Note that the MSP connector includes internal pull-down resistors instead of pull-up resistors as on the

MXP connectors. You can either rely on the  $10\text{ k}\Omega$  pull-up resistors on the PmodKYPD to override the approximately  $40\text{ k}\Omega$  MSP pull-downs (use the MSP 5-volt supply for the PmodKYPD), or you can connect the PmodKYPD “V” terminal to *ground* thereby converting the  $10\text{ k}\Omega$  resistors to pull-down resistors – this latter connection requires that you also change the column drives to *high* level with a Boolean “T” instead of the current “F” value.

### 25.4 Integrated Project Ideas

Now that you know how to use the keypad consider integrating it with other devices to create a complete system, for example:

- *Hotel Room Safe Controller* (49)
- *RPN Calculator* (48)

### 25.5 For More Information

- *PmodKYPD Reference Manual* by Digilent ~ Reference manual for the  $4\times 4$  keypad; scroll to the bottom of the page:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,940&Prod=PMODKYPD>
- *PmodKYPD Schematics* by Digilent ~ Schematic diagram of the  $4\times 4$  keypad; scroll to the bottom of the page:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,940&Prod=PMODKYPD>

# 26 LCD Character Display – UART Interface

An LCD character display provides an excellent means for your LabVIEW block diagram code to display measurements, status, and conditions with ASCII character strings. The LCD character display also provides visual feedback for a user interface. Figure 26.1 pictures the NI myRIO Embedded Systems Kit LCD character display which supports three different serial communications standards; this chapter focuses on the UART interface and the next two chapters cover the SPI and I<sup>2</sup>C-bus interfaces to the display.

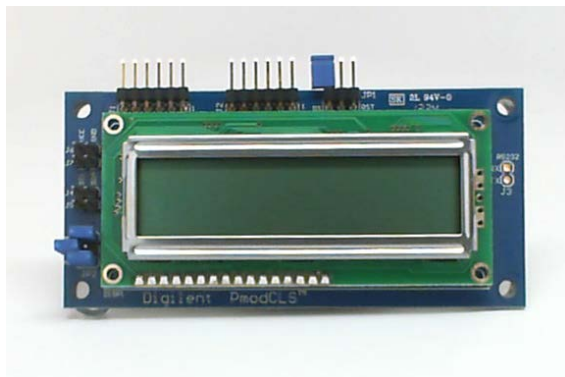


Figure 26.1: NI myRIO Embedded Systems Kit LCD character display.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Configure the display for UART serial communications at a desired baud rate,
2. Send characters directly to appear on display, and
3. Send escape sequences to adjust the display modes.

## 26.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LCD character display.

Select these parts from the NI myRIO Embedded Systems Kit:

- LCD character display with serial interface (PmodCLS), <http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS>
- Jumper wires, F-F (3×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 26.2 on the next page; the LCD character display requires three connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply → B/+3.3V (pin 33)
2. Ground → B/GND (pin 30)
3. UART receiver → B/UART.TX (pin 14)

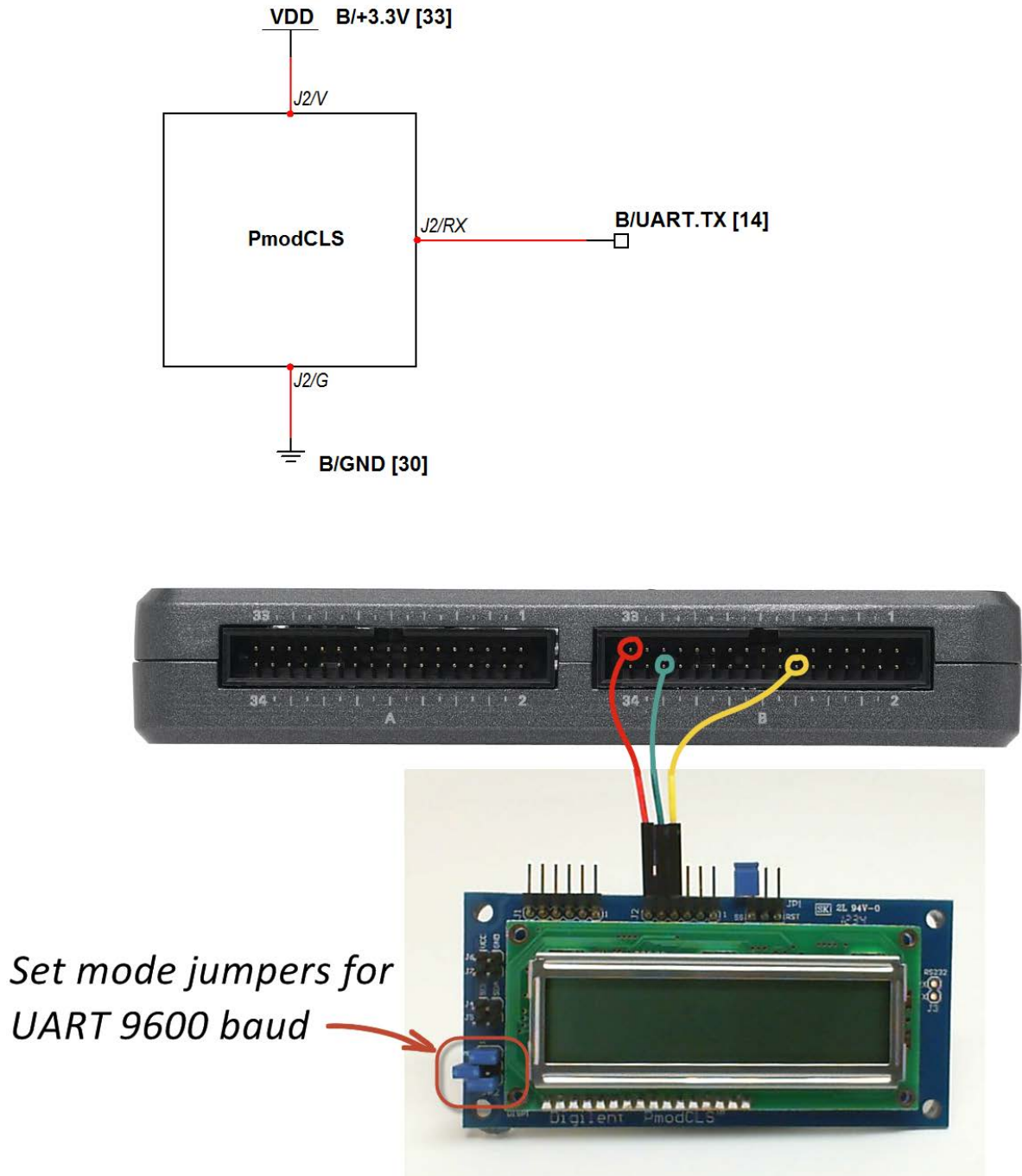


Figure 26.2: Demonstration setup for LCD character display connected to NI myRIO MXP Connector B. Remember to set the mode jumpers as shown for UART serial communications.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `LCD (UART) demo.lvproj` contained in the subfolder `LCD (UART) demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the state of the NI myRIO on-board 3-axis accelerometer as three values (*X*, *Y*, and *Z* directions) and the state of the built-in pushbutton on the bottom side of myRIO; press the button to see the state change from 0 to 1. Change the orientation of myRIO to see the accelerometer values change, and shake myRIO to see larger acceleration values. You may wish to hold or tape the LCD character display to the top of myRIO, too.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,

- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct LCD character display connector terminals — double-check your connections, and ensure that you have connected the NI myRIO UART “transmit” output to the LCD character display “receive” input; also check that you have not accidentally crossed the power supply connections, and
- Correct mode jumper settings – refer again to the circled region of Figure 26.2 on the preceding page for the proper settings.

## 26.2 Interface Theory

**Interface circuit:** The LCD character display supports three serial communication standards: UART (universal asynchronous receiver-transmitter), SPI (serial peripheral interface), and I<sup>2</sup>C-bus (inter-IC). This chapter concentrates on the UART interface while the next two chapters cover the other interfaces; the functionality of the LCD character display itself remains independent of the selected communication standard. Simply send ASCII characters to the display’s UART receive line to make them appear on the display screen. Use “escape sequences” to configure other aspects of the display such as homing the cursor, displaying and blinking the cursor, scrolling the display, and so on.

Study the video *LCD Character Display Interfacing Theory* ([youtu.be/m0Td7KbhvdI](https://youtu.be/m0Td7KbhvdI), 10:36) to learn more about the LCD character display operation with UART interface including setting the baud rate, sending display information and escape sequences, and available configuration instructions. Study *Serial Communication: UART* ([youtu.be/odN66E85J5E](https://youtu.be/odN66E85J5E), 7:56) to understand how the UART Express VI configuration options relate to the signaling waveforms between UART transmitters and receivers.

**LabVIEW programming:** Study the video *UART Express VI* ([youtu.be/0FMnkFDsGQs](https://youtu.be/0FMnkFDsGQs), 5:29) to learn how to use the UART Express VI to read and write character strings, including strings with special characters.

### 26.3 Basic Modifications

Study the video *LCD (UART) Demo Walk-Through* ([youtu.be/JsEMMnIWg4k](https://youtu.be/JsEMMnIWg4k), 3:44) to learn the design principles of LCD (UART) demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Interchange the two display rows when the NI myRIO on-board button is pressed; revert to the original display when the button is released.
2. Display the full character set – Cycle through all possible 256 8-bit patterns and see if you can spot the ASCII code for the “degrees” symbol.
3. Create an instruction tester – Make two string controls, one for the instruction and a second for the display; use Boolean controls to disable sending either one.

### 26.4 Integrated Project Ideas

Now that you know how to use the LCD character display consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *NTP Clock* (43)
- *Digital Thermometer* (45)
- *QR Code Scanner* (47)
- *RPN Calculator* (48)
- *Hotel Room Safe Controller* (49)
- *Tachometer* (50)
- *Compass with Tilt Correction* (54)
- *Guitar Tuner* (55)

- *Digital Bubble Level* (57)
- *EEPROM Programmer* (59)

### 26.5 For More Information

- *PmodCLS Reference Manual* by Digilent ~ Reference manual for the LCD character display:  
[http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS\\_rm\\_RevD-E.pdf](http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS_rm_RevD-E.pdf)
- *PmodCLS Schematics* by Digilent ~ Schematic diagram of the LCD character display:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS>



# 27 LCD Character Display – SPI Interface

An LCD character display provides an excellent means for your LabVIEW block diagram code to display measurements, status, and conditions with ASCII character strings. The LCD character display also provides visual feedback for a user interface. Figure 27.1 pictures the NI myRIO Embedded Systems Kit LCD character display which supports three different serial communications standards. This chapter focuses on the SPI interface to the display, while the previous chapter covers the UART interface and the next chapter covers the I<sup>2</sup>C-bus interface.

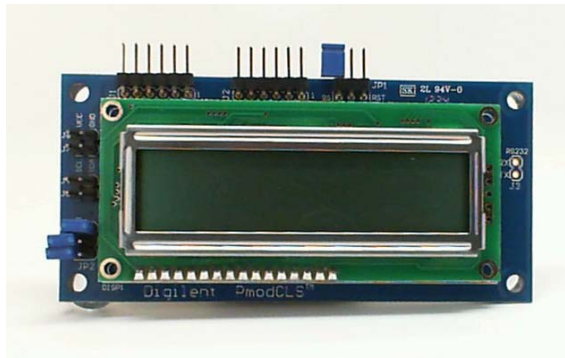


Figure 27.1: NI myRIO Embedded Systems Kit LCD character display.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Configure the display for SPI serial communications at a desired clock frequency,
2. Send characters directly to appear on display, and
3. Send escape sequences to adjust the display modes.

## 27.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LCD character display.

**Select these parts from the NI myRIO Embedded Systems Kit:**

- LCD character display with serial interface (PmodCLS), <http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS>
- Jumper wires, F-F (4×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 27.2 on the next page; the LCD character display requires four connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply → B/+3.3V (pin 33)
2. Ground → B/GND (pin 30)
3. SPI receiver → B/SPI.MOSI (pin 25)
4. SPI clock → B/SPI.CLK (pin 21)

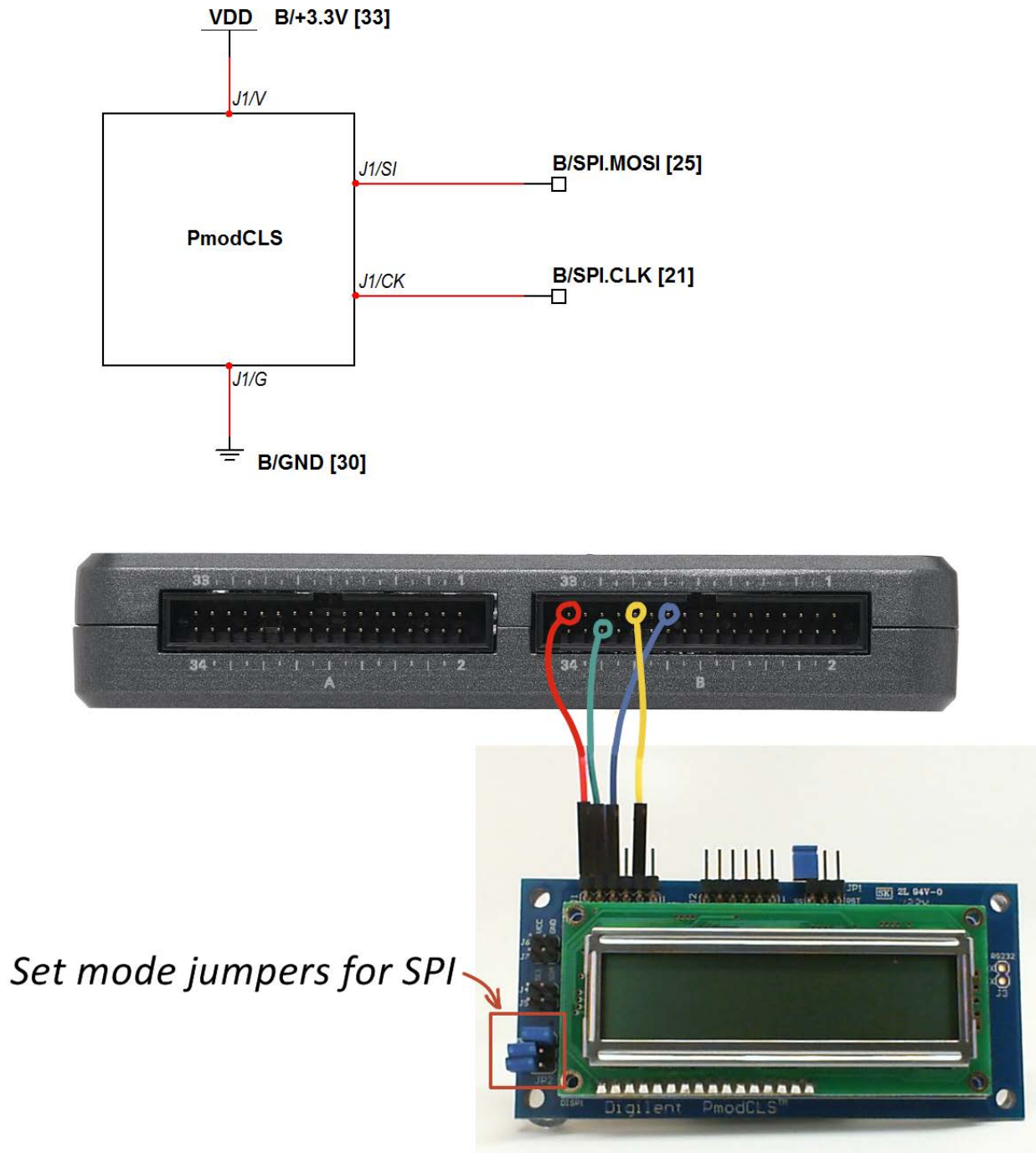


Figure 27.2: Demonstration setup for LCD character display connected to NI myRIO MXP Connector B. Remember to set the mode jumpers as shown for SPI serial communications.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `LCD (SPI) demo.lvproj` contained in the subfolder `LCD (SPI) demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the state of the NI myRIO on-board 3-axis accelerometer as three values (*X*, *Y*, and *Z* directions) and the state of the built-in pushbutton on the bottom side of myRIO; press the button to see the state change from 0 to 1. Change the orientation of myRIO to see the accelerometer values change, and shake myRIO to see larger acceleration values. You may wish to hold or tape the LCD character display to the top of myRIO, too.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,

- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct LCD character display connector terminals — double-check your connections, and ensure that you have connected the NI myRIO SPI “MOSI” output to the LCD character display “MOSI” input; also check that you have not accidentally crossed the power supply connections, and
- Correct mode jumper settings – refer again to the circled region of Figure 27.2 on the preceding page for the proper settings.

## 27.2 Interface Theory

**Interface circuit:** The LCD character display supports three serial communication standards: UART (universal asynchronous receiver-transmitter), SPI (serial peripheral interface), and I<sup>2</sup>C-bus (inter-IC). This chapter concentrates on the SPI interface while the adjacent chapters cover the other interfaces; the functionality of the LCD character display itself remains independent of the selected communication standard. Simply send ASCII characters to the display’s SPI receive line to make them appear on the display screen. Use “escape sequences” to configure other aspects of the display such as homing the cursor, displaying and blinking the cursor, scrolling the display, and so on.

Study the video *LCD Character Display Interfacing Theory* ([youtu.be/m0Td7KbhvdI](https://youtu.be/m0Td7KbhvdI), 10:36) to learn more about the LCD character display operation including setting the baud rate, sending display information and escape sequences, and available configuration instructions. Study *Serial Communication: SPI* ([youtu.be/GaXtDamw5As](https://youtu.be/GaXtDamw5As), 7:02) to understand how the SPI Express VI configuration options relate to the signaling waveforms between SPI transmitters and receivers.

**LabVIEW programming:** Study the video *SPI Express VI* ([youtu.be/S7KkTeMfmc8](https://youtu.be/S7KkTeMfmc8), 5:51) to learn how to use the SPI Express VI to read and write character strings, including strings with special characters; the video *UART Express VI* ([youtu.be/0FMnkFDsGQs](https://youtu.be/0FMnkFDsGQs), 5:29) provides additional useful string formatting techniques.

### 27.3 Basic Modifications

Study the video *LCD (SPI) Demo Walk-Through* ([youtu.be/oOXYryu4Y-c](https://youtu.be/oOXYryu4Y-c), 4:23) to learn the design principles of LCD (SPI) demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Interchange the two display rows when the NI myRIO on-board button is pressed; revert to the original display when the button is released.
2. Display the full character set – Cycle through all possible 256 8-bit patterns and see if you can spot the ASCII code for the “degrees” symbol.
3. Create an instruction tester – Make two string controls, one for the instruction and a second for the display; use Boolean controls to disable sending either one.
4. Experiment with the serial clock frequency: what is the highest possible frequency that still provides reliable communication with the LCD? What is the lowest frequency that still updates the LCD display without introducing noticeable lag? Remember to change the speed of *both* SPI Express VIs. NOTE: You must power cycle the LCD display each time you change the serial clock frequency because while the display automatically adapts to the SPI clock frequency, it does so only one time.

### 27.4 Integrated Project Ideas

Now that you know how to use the LCD character display consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *NTP Clock* (43)
- *Digital Thermometer* (45)
- *QR Code Scanner* (47)
- *RPN Calculator* (48)
- *Hotel Room Safe Controller* (49)
- *Tachometer* (50)
- *Compass with Tilt Correction* (54)
- *Guitar Tuner* (55)
- *Digital Bubble Level* (57)
- *EEPROM Programmer* (59)

### 27.5 For More Information

- *PmodCLS Reference Manual* by Digilent ~ Reference manual for the LCD character display:  
[http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS\\_rm\\_RevD-E.pdf](http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS_rm_RevD-E.pdf)
- *PmodCLS Schematics* by Digilent ~ Schematic diagram of the LCD character display:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS>
- *M68HC11 Reference Manual* by Freescale Semiconductor ~ Refer to Section 8 for a complete treatment of the SPI serial bus standard, including timing diagrams and multi-master systems:  
[http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/M68HC11RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf)

# 28 LCD Character Display – I<sup>2</sup>C-bus Interface

An LCD character display provides an excellent means for your LabVIEW block diagram code to display measurements, status, and conditions with ASCII character strings. The LCD character display also provides visual feedback for a user interface. Figure 28.1 pictures the NI myRIO Embedded Systems Kit LCD character display which supports three different serial communications standards. This chapter focuses on the I<sup>2</sup>C-bus interface to the display, while the previous two chapters cover the UART and SPI interfaces.

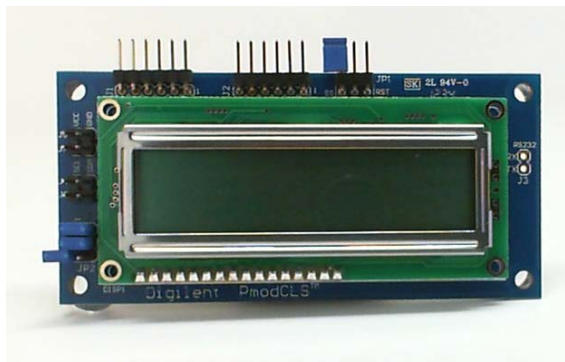


Figure 28.1: NI myRIO Embedded Systems Kit LCD character display.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Configure the display for I<sup>2</sup>C-bus serial

- communications,
2. Send characters directly to appear on display, and
3. Send escape sequences to adjust the display modes.

## 28.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LCD character display.

Select these parts from the NI myRIO Embedded Systems Kit:

- LCD character display with serial interface (PmodCLS), <http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS>
- Jumper wires, F-F (4×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 28.2 on the next page; the LCD character display requires four connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply → B/+3.3V (pin 33)
2. Ground → B/GND (pin 30)
3. Serial data (SD) → B/I2C.SDA (pin 34)
4. Serial clock (SC) → B/I2C.SCL (pin 32)

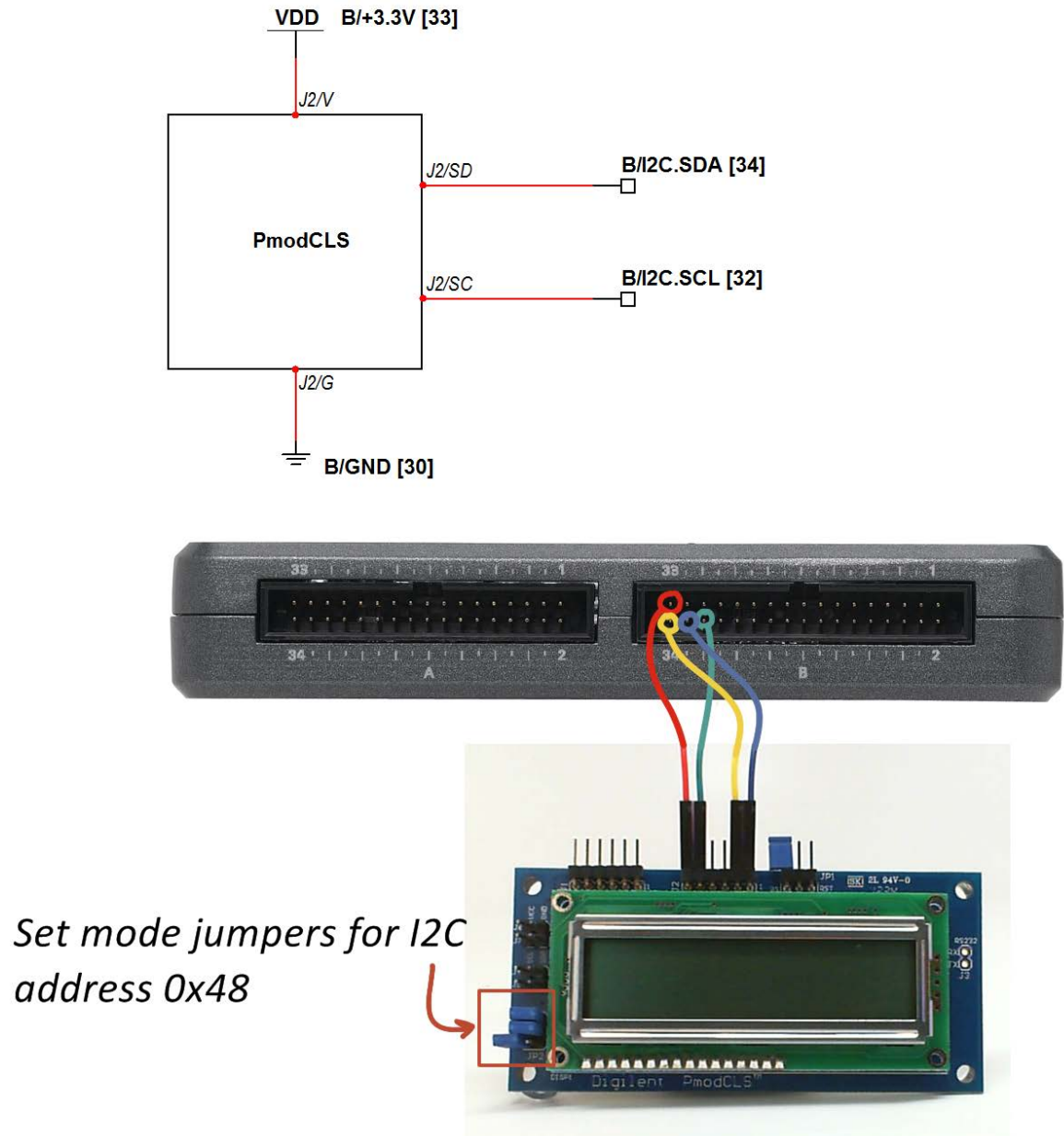


Figure 28.2: Demonstration setup for LCD character display connected to NI myRIO MXP Connector B. Remember to set the mode jumpers as shown for I<sup>2</sup>C-bus serial communications.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project LCD (I2C) demo.lvproj contained in the subfolder LCD (I2C) demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the state of the NI myRIO on-board 3-axis accelerometer as three values (*X*, *Y*, and *Z* directions) and the state of the built-in pushbutton on the bottom side of myRIO; press the button to see the state change from 0 to 1. Change the orientation of myRIO to see the accelerometer values change, and shake myRIO to see larger acceleration values. You may wish to hold or tape the LCD character display to the top of myRIO, too.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,

- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct LCD character display connector terminals — double-check your connections, and ensure that you have connected the NI myRIO I<sup>2</sup>C-bus “SDA” line to the LCD character display “SD” terminal on connector J2 and the “SCL” line to the “SC” terminal; also check that you have not accidentally crossed the power supply connections, and
- Correct mode jumper settings – refer again to the circled region of Figure 28.2 on the preceding page for the proper settings.

**NOTE:** Double-check the “SDA” and “SCL” connections should you see the message “Error -36011 occurred at myRIO Write I2C.vi” or similar; this message indicates that NI myRIO did not receiving an expected acknowledgement from the LCD character display I<sup>2</sup>C-bus interface.

## 28.2 Interface Theory

**Interface circuit:** The LCD character display supports three serial communication standards: UART (universal asynchronous receiver-transmitter), SPI (serial peripheral interface), and I<sup>2</sup>C-bus (inter-IC). This chapter concentrates on the I<sup>2</sup>C-bus interface (commonly denoted “I2C”) while the previous two chapters cover the other interfaces; the functionality of the LCD character display itself remains independent of the selected communication standard. Simply send ASCII characters to the display’s I<sup>2</sup>C-bus receive line to make them appear on the display screen. Use “escape sequences” to configure other aspects of the display such as homing the cursor, displaying and blinking the cursor, scrolling the display, and so on.

Study the video *LCD Character Display Interfacing Theory* ([youtu.be/m0Td7KbhvdI](https://youtu.be/m0Td7KbhvdI), 10:36) to

learn how to send display information, escape sequences, and configuration instructions to the LCD character display. Study *Serial Communication: I2C* ([youtu.be/7CgNF78pYQM](https://youtu.be/7CgNF78pYQM), 8:47) to understand how the I2C Express VI configuration options relate to the signaling waveforms between I<sup>2</sup>C-bus transmitters and receivers.

**LabVIEW programming:** Study the video *UART Express VI* ([youtu.be/0FMnkFDsGQs](https://youtu.be/0FMnkFDsGQs), 5:29) to learn useful string formatting techniques.

### 28.3 Basic Modifications

Study the video *LCD (I2C) Demo Walk-Through* ([youtu.be/qbD31AeqOMk](https://youtu.be/qbD31AeqOMk), 4:32) to learn the design principles of LCD (I2C) demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Interchange the two display rows when the NI myRIO on-board button is pressed; revert to the original display when the button is released.
2. Display the full character set – Cycle through all possible 256 8-bit patterns and see if you can spot the ASCII code for the “degrees” symbol.
3. Create an instruction tester – Make two string controls, one for the instruction and a second for the display; use Boolean controls to disable sending either one.
4. Experiment with the serial clock frequency; does the LCD character display appear to support both “Standard mode” and “Fast mode” frequencies? Remember to change the speed of *both* I2C Express VIs.
5. Try disconnecting either I<sup>2</sup>C-bus line from the LCD character display. What LabVIEW error message do you see?
6. Change the “slave” address to some other value. What LabVIEW error message do you see?

### 28.4 Integrated Project Ideas

Now that you know how to use the LCD character display consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *NTP Clock* (43)
- *Digital Thermometer* (45)
- *QR Code Scanner* (47)
- *RPN Calculator* (48)
- *Hotel Room Safe Controller* (49)
- *Tachometer* (50)
- *Compass with Tilt Correction* (54)
- *Guitar Tuner* (55)
- *Digital Bubble Level* (57)
- *EEPROM Programmer* (59)

### 28.5 For More Information

- *PmodCLS Reference Manual* by Digilent ~ Reference manual for the LCD character display:  
[http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS\\_rm\\_RevD-E.pdf](http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS_rm_RevD-E.pdf)
- *PmodCLS Schematics* by Digilent ~ Schematic diagram of the LCD character display:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS>
- *UM10204 I<sup>2</sup>C-bus Specification and User Manual* by NXP Semiconductors ~ A complete treatment of the I<sup>2</sup>C-bus standard, including timing diagrams and multi-master systems:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)



# 29 LED Matrix

The LED matrix pictured in Figure 29.1 is an 8×8 grid with one green LED and one red LED behind each circular dot. The LED matrix uses the same row/column wiring scheme as the keypad of Chapter 25 with two column lines (one for each LED) and a shared row line.

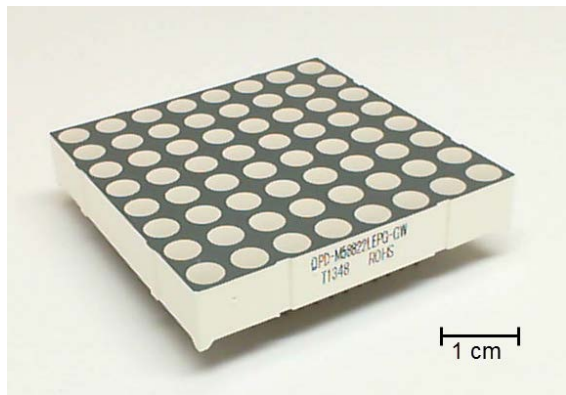


Figure 29.1: NI myRIO Embedded Systems Kit LED matrix.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the LED matrix wiring diagram,
2. Explain the raster display technique to display arbitrary patterns,
3. Describe the need to alternately activate the green and red LEDs, and

4. Design an animation sequence.

## 29.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LED matrix.

### Select these parts:

- LED matrix, <http://www.sparkfun.com/datasheets/Components/SanYoung-Medium-RG.pdf>
- Breadboard (2×)
- Jumper wires, M-F (24×)

**Build the interface circuit:** Refer to the pin diagram shown in Figure 29.2 on page 131 and recommended breadboard layout shown in Figure 29.3 on page 132.

**TIP:** Use the resistor color code for the DIO wire colors, e.g., black (0) for B/DIO0, brown (1) for B/DIO1, and so on.

The interface circuit requires eight connections to NI myRIO MXP Connector A and sixteen connections to Connector B (see Figure A.1 on page 233):

1. Row 1 (1) → A/DIO0 (pin 11)
2. Row 2 (2) → A/DIO1 (pin 13)
3. Row 3 (3) → A/DIO2 (pin 15)
4. Row 4 (4) → A/DIO3 (pin 17)
5. Row 5 (21) → A/DIO4 (pin 19)
6. Row 6 (22) → A/DIO5 (pin 21)

7. Row 7 (23) → A/DIO6 (pin 23)
8. Row 8 (24) → A/DIO7 (pin 25)
9. Column 1 green (5) → B/DIO0 (pin 11)
10. Column 2 green (6) → B/DIO1 (pin 13)
11. Column 3 green (7) → B/DIO2 (pin 15)
12. Column 4 green (8) → B/DIO3 (pin 17)
13. Column 5 green (9) → B/DIO4 (pin 19)
14. Column 6 green (10) → B/DIO5 (pin 21)
15. Column 7 green (11) → B/DIO6 (pin 23)
16. Column 8 green (12) → B/DIO7 (pin 25)
17. Column 1 red (20) → B/DIO8 (pin 27)
18. Column 2 red (19) → B/DIO9 (pin 29)
19. Column 3 red (18) → B/DIO10 (pin 31)
20. Column 4 red (17) → B/DIO11 (pin 18)
21. Column 5 red (16) → B/DIO12 (pin 22)
22. Column 6 red (15) → B/DIO13 (pin 26)
23. Column 7 red (14) → B/DIO14 (pin 32)
24. Column 8 red (13) → B/DIO15 (pin 34)

**TIP:** *Short on jumper wires? You can connect a subset of the array, e.g., rows 1 to 4 and columns 1 to 4 for a green-only display that requires only eight jumper wires.*

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project LED Matrix demo.lvproj contained in the subfolder LED Matrix demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys

(downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The VI begins in “automatic pattern” mode and the display should show a converging square animation in which the display changes every half second. Move the pattern loop time pointer slide to adjust the animation speed. Do you notice anything unusual for loop times below 100 ms?

Click the auto pattern switch and then try clicking buttons on the green 2-D array control. You should see the corresponding green LED activate on the display. Take a moment to try each of the rows and columns to ensure that your wiring is correct. Repeat for the red 2-D array control.

The green pattern and red pattern indicators under the array controls show the state of the 8×8 2-D Boolean arrays as a single 64-bit hexadecimal value. Select Edit | Reinitialize Values to Default to clear all 64 array buttons and then click the lower-right button. Note how this corresponds to the least-significant bit in the 64-bit value. Click the upper-left button; this is the most-significant bit. Click more buttons to determine the ordering of the remaining bits. These 64-bit pattern codes provide a convenient way for you to create individual display patterns to be used for animations.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black “Run” button on the toolbar signifying that the VI is in run mode, and
- Correct wiring — ensure that your wiring order is correct (it is easy to make a mistake).

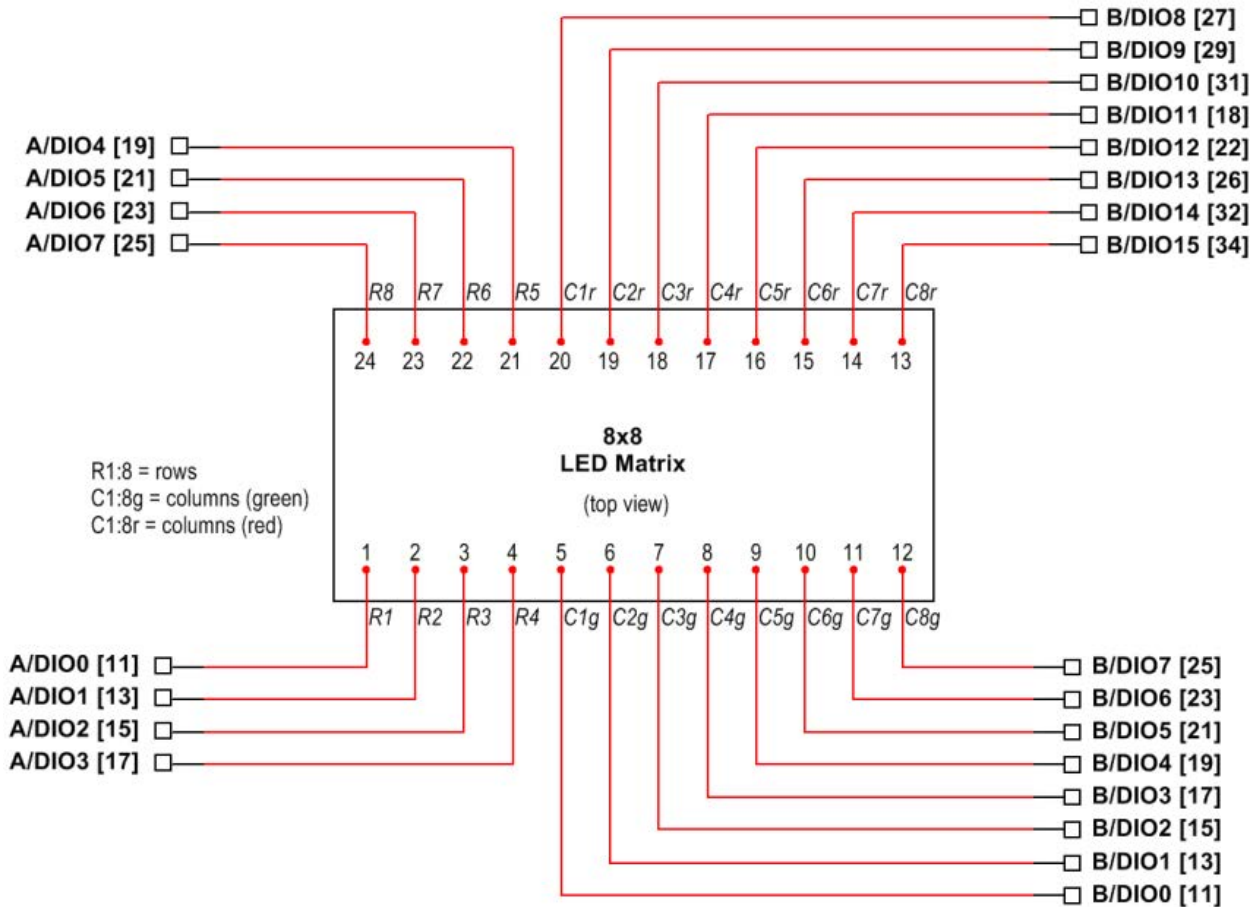


Figure 29.2: Demonstration circuit for LED matrix: pin diagram and connection to 24 digital I/O terminals on NI myRIO MXP Connectors A and B.

### 29.2 Interface Theory

**Interface circuit:** Each circular dot of the 8x8 LED matrix contains a green LED and a red LED. The cathode of each diode pair connects to the same horizontal row wire. The anode of all the green LEDs in the same column connects to the same vertical column wire; the anodes of the red LEDs are similarly connected, thereby providing individual

control of the two LEDs at each dot. Activating both LEDs can color mix to yellow.

Study the video *LED Matrix Interfacing Theory* ([youtu.be/vsBjZBLdeNc](https://youtu.be/vsBjZBLdeNc), 9:51) to learn more about the LED matrix wiring diagram and pin connections, electrical interfacing (especially the substantial voltage/current mismatch between the red and green LEDs), and the raster display technique that

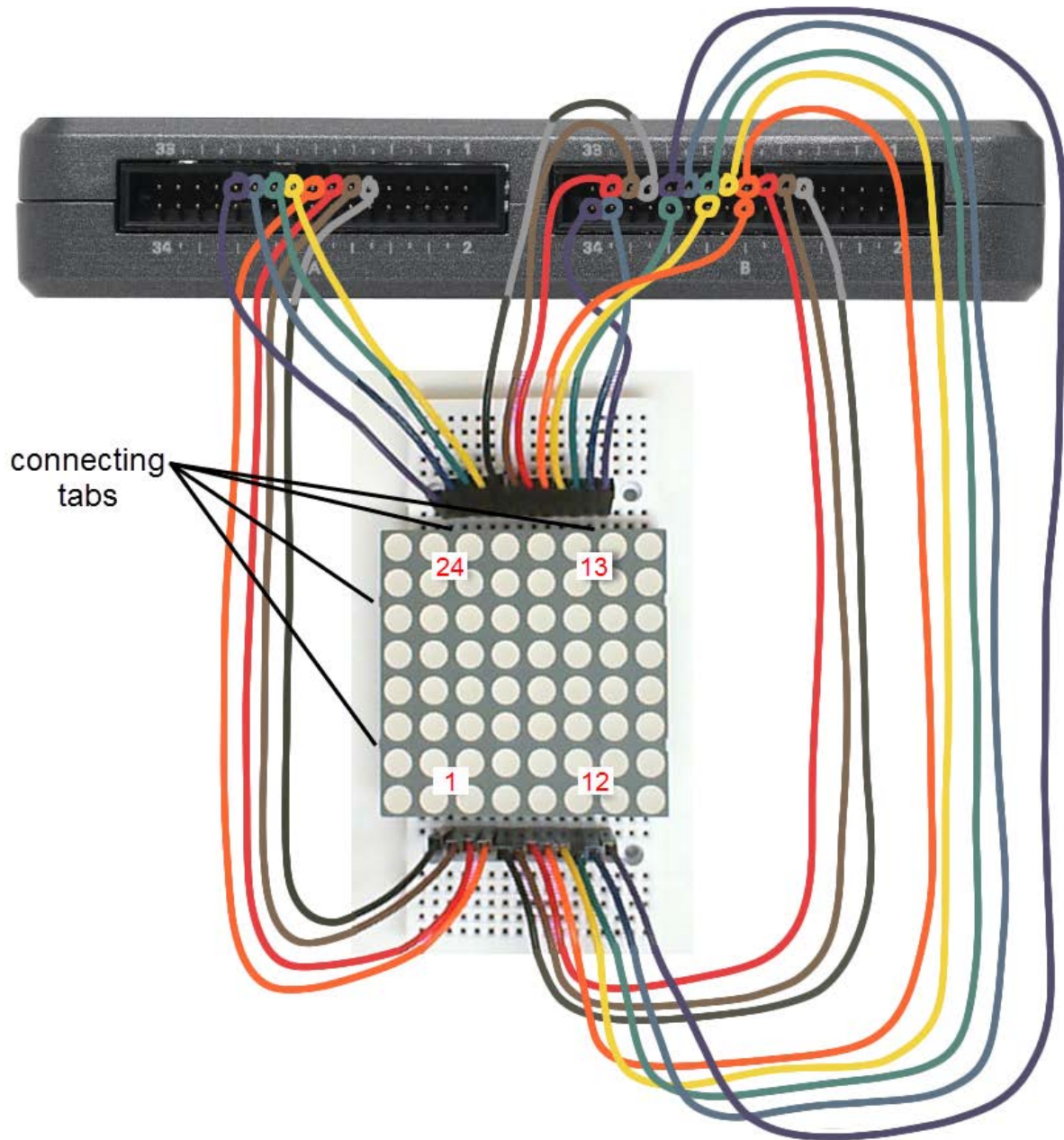


Figure 29.3: Demonstration circuit for LED matrix: recommended breadboard layout.

permits an arbitrary pattern of dots to be selected.

**NOTE:** *The interfacing technique described in this chapter requires no additional parts and also helps you to understand how to set up a raster display (also known as a multiplexed display). However, 24 DIOs uses more than half of the available 40 DIOs and also ties up shared resources such the SPI, I<sup>2</sup>C-bus PWM, and encoder. Furthermore, the raster scan code requires a significant amount of a CPU effort. Consider instead using the MAX7219 LED Display Driver (see “For More Information” section at the bottom of the page) to completely offload the detailed control of the LED matrix; you simply transfer the desired patterns via the SPI bus. This technique becomes mandatory should your application need more LED matrices to build a larger display.*

**LabVIEW programming:** Study the video *Digital Output Low-Level VIs* ([youtu.be/WvnInG3ffqY](https://youtu.be/WvnInG3ffqY), 4:53) to learn how to use the low-level VIs to connect Boolean arrays directly to the digital outputs, i.e., as a bus.

### 29.3 Basic Modifications

Study the video *LED Matrix Demo Walk-Through* ([youtu.be/Bqq63sKwQKE](https://youtu.be/Bqq63sKwQKE), 12:14) to learn the design principles of LED Matrix demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Finish the “Auto pattern generator” loop and update the main loop to add a pattern codes array for the red LEDs.
2. Create a set of 64-bit pattern codes to make your own animation, and then update the 1-D array constant in the “Auto pattern generator” loop. You may find it more convenient to change the constant to a front-panel control. You may also

want to use “Array Size” from the Programming | Array subpalette instead of the constant “4” to make your code accommodate an arbitrary number of 64-bit patterns.

3. Think of a way to map one or more measurements such as the onboard accelerometer output or analog input voltages onto the display. For example, you could make eight bar graphs to show eight analog input voltages, or you could map the *X* and *Y* outputs from the accelerometer onto a Cartesian grid. Your mapping destination is the 2-D Boolean array global variables.
4. Add a pointer slide to vary the loop time of the timed loop: pull down on the “Right Data Node” on the upper-right interior of the timed loop to expose the “Next Loop Iteration Timing” terminal (appears at *dt*), and then create a front-panel control to interactively adjust the loop time. Set the lower limit to 1 ms to avoid selecting 0 ms (this hangs the application).
5. Observe the effects of jitter on the display: change the timed loop to a standard while loop (right-click on the loop frame and select “Replace with While Loop”) and then add a 1 ms time delay. You should notice that the display intensity has a slight random flicker because the real-time processor now has more freedom to process background tasks. The timed loop, on the other hand, guarantees a precise loop time.

### 29.4 Integrated Project Ideas

Now that you know how to use the LED matrix consider integrating it with other devices to create a complete system, for example:

- *Digital Bubble Level* (57)
- *NTP Clock* (43)

## 29.5 For More Information

- *8×8 LED Matrix Datasheet* by SparkFun ~ Datasheet for the 8×8 LED matrix:  
<http://www.sparkfun.com/datasheets/Components/SanYoung-Medium-RG.pdf>
- *MAX7219 LED Display Driver with SPI Interface* by Maxim Integrated ~ Includes power transistors and raster display controller to completely offload the detailed control of each LED. Simply send the patterns via the SPI serials bus:  
<http://www.maximintegrated.com/MAX7219>

# 30 Serial EEPROM

An EEPROM (Electrically Erasable Programmable Read-Only Memory) provides nonvolatile data storage that persists even when power is absent. EEPROMs find application as sensor calibration tables, digital keys, and adaptive devices. The Microchip 25LC040A serial EEPROM (Figure 30.1) offers 4 kilobits of storage organized as an array of 512 8-bit bytes accessible via the SPI serial bus.

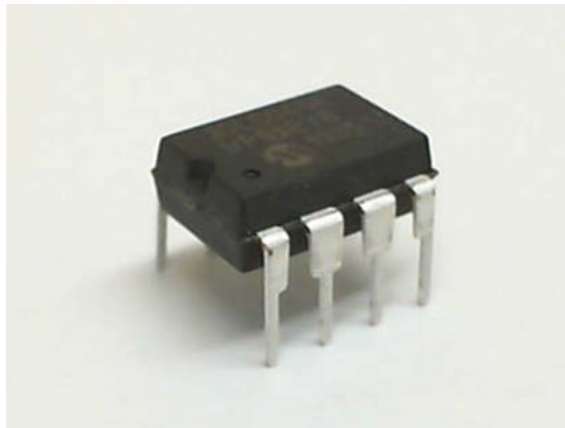


Figure 30.1: NI myRIO Embedded Systems Kit serial EEPROM.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the Microchip 25LC040A features and device pinout,

2. Apply the 25LC040A instruction set to read and write data to the memory array and status register, and
3. Use the LabVIEW low-level SPI interface VIs to communicate with the 25LC040A.

## 30.1 Component Demonstration

Follow these steps to demonstrate correct operation of the serial EEPROM.

**Select these parts from the NI myRIO Embedded Systems Kit:**

- Microchip 25LC040A serial EEPROM, <http://www.microchip.com/25LC040A>
- Jumper wires, M-F (7×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 30.2 on the next page; the serial EEPROM requires six connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. VCC → B/+3.3V (pin 33)
2. VSS → B/GND (pin 30)
3. SI → B/SPI.MOSI (pin 25)
4. SO → B/SPI.MISO (pin 23)
5. SCK → B/SPI.SCLK (pin 21)
6.  $\overline{CS}$  → B/DIO0 (pin 11)
7.  $\overline{WP}$  → B/DIO0 (pin 13)

Be sure to include a short jumper wire to connect the  $\overline{HOLD}$  input to the NI myRIO 3.3-volt supply.



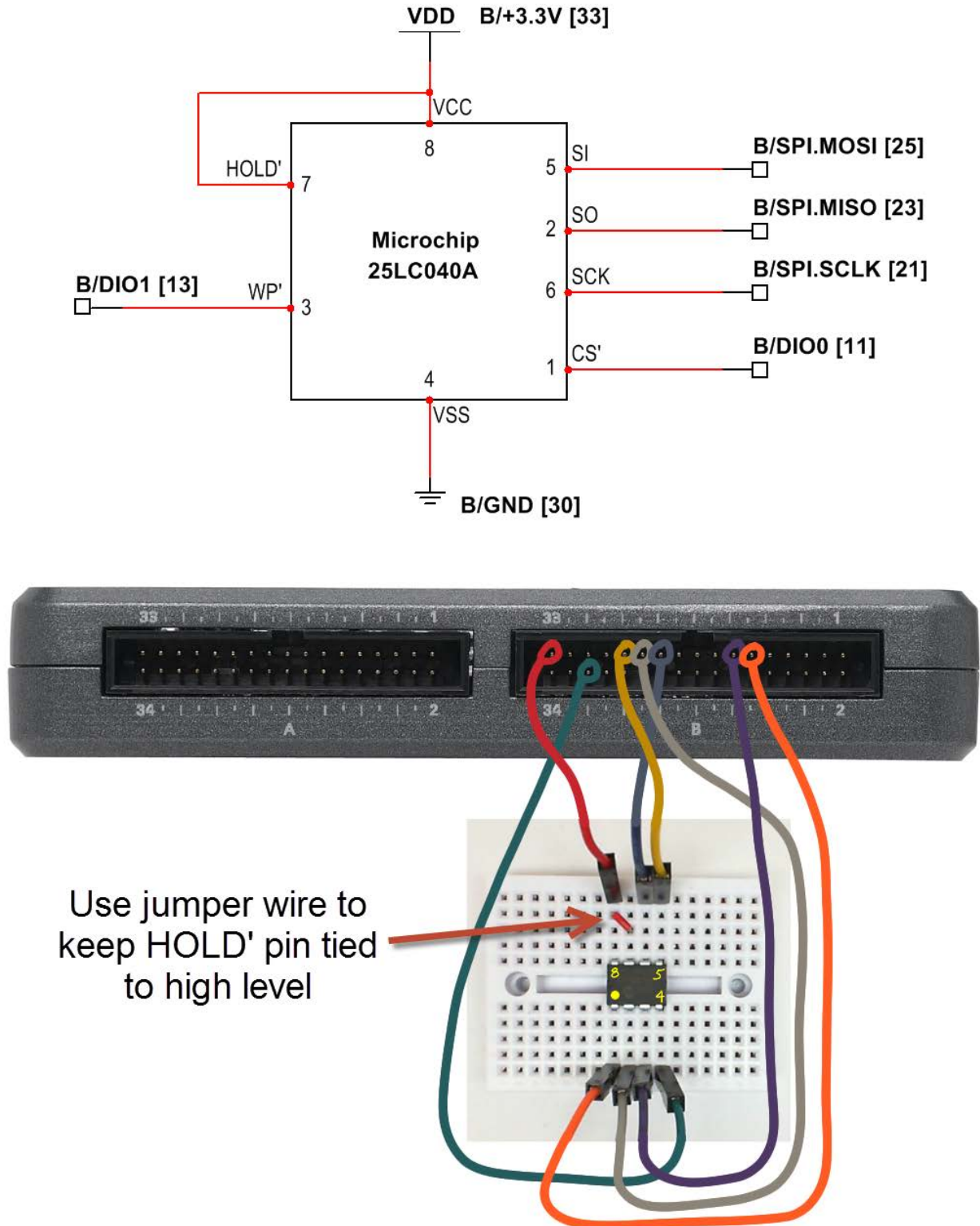


Figure 30.2: Demonstration setup for serial EEPROM connected to NI myRIO MXP Connector B.



**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `EEPROM demo.lvproj` contained in the subfolder `EEPROM demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI provides an Action control to select among four different actions (read entire memory array, write to the array, read the STATUS register, and write to the STATUS register) and three associated data panels (data read from the memory array, data to write, and STATUS byte to write). The VI performs the selected action only one time when the Action control value changes, therefore an Idle action must be selected between two successive actions of the same type such as writing to the array.

Select the Read entire array action. If your EEPROM has not yet been programmed, i.e., it is still fresh from the factory, you should expect to see the EEPROM array indicator filled with 255 (hexadecimal FF) and the associated graph plotting a constant of the same value as a function of address (512 total).

Select the Write to array action. By default the VI performs a single-byte write of value 0 to Address 0.

Observe that the EEPROM array and graph are both empty now, therefore you need to select Read entire array to update these two displays. Because the Write protect control is initially set (this holds the  $\overline{WP}$  pin low) you should observe that the array is still entirely filled with value 255. Click the Write protect button to enable writing, select Write to array, and then select Read entire array. You should now see that Address 0 contains the value zero.

Try writing some single-byte values to other addresses, and confirm that the plotted values appear where you expect. Type a new value into the EEPROM array index control to view other regions of the array indicator, e.g., type 511 to see the last array value.

Click on the Bytes to write control to enter more values for a multi-byte write. You can write up to sixteen bytes (one “page”) in a given write operation provided that the address begins on a page boundary (lower four bits of the address are zero). Right-click on the Bytes to write, select “Data Operations” followed by “Delete Element” to remove values from the array control. You may also reinitialize the entire front panel to the original settings by selecting “Edit | Reinitialize Values to Default.”

Select Read STATUS register to view the contents of the EEPROM’s STATUS register; expect to see a zero value. Enter the hexadecimal value 0x08 into the STATUS to write control, and then select Write STATUS register followed by Read STATUS register to confirm that the STATUS register has been updated. With this setting the upper half of the memory is write protect regardless of the state of the  $\overline{WP}$  pin. Try to write data to a low address (0 to 127) and then to a high address (128 to 511); remember to deselect the Write protect switch. Confirm that the value at the lower address changes while the value at the higher address does not. Write a 0x00 value to the STATUS register when you are finished to return the register to its as-shipped default value (the two “block protect” bits are nonvolatile).

Click the Stop button or press the escape key to

stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and
- Correct 25LC040A pins connections — double-check your connections, and ensure that you have not reversed the NI myRIO B/SPI.MOSI and B/SPI.MISO lines; also check that you have not accidentally crossed the power supply connections.

## 30.2 Interface Theory

**Interface circuit:** The Microchip 25LC040A offers 4 kilobits of nonvolatile storage organized as 512 8-bit bytes. The memory array is rated for one million erase/write cycles with data retention over 200 years. The array may be read in any fashion from a single byte up to the entire array in a single read operation. Data may be written to the array either as individual bytes or up to a single page of 16 bytes provided that the starting address is a page boundary.

Study the video *EEPROM Interfacing Theory* ([youtu.be/RxRwyDOCeRw](https://youtu.be/RxRwyDOCeRw), 11:18) to learn more about the 25LC040A features, applications, pinout for the 8-pin DIP package, instruction set, and necessary configuration settings for the NI myRIO SPI VIs. The video continues by detailing the timing of the chip select  $\overline{CS}$  necessary to read from and write to the array and STATUS register (including considerations for writing to the entire array) and concludes by describing how to use the  $\overline{HOLD}$  pin to suspend an SPI bus transaction momentarily.

Study *Serial Communication: SPI* ([youtu.be/GaXtDamw5As](https://youtu.be/GaXtDamw5As), 7:02) to better un-

derstand the signaling waveforms between SPI transmitters and receivers.

**LabVIEW programming:** Study the video *SPI Express VI* ([youtu.be/S7KkTeMfmc8](https://youtu.be/S7KkTeMfmc8), 5:51) to learn how to use the SPI VIs to read and write data.

## 30.3 Basic Modifications

Study the video *EEPROM Demo Walk-Through* ([youtu.be/UNdVUnYHE4U](https://youtu.be/UNdVUnYHE4U), 13:07) to learn the design principles of EEPROM demo, and then try making these modifications to the block diagram of *Main.vi*:

1. Add an action Write entire array that will fill the memory array:
  - Add the item “Write entire array” to the Action control,
  - Create an array of 512 values that will produce a recognizable pattern on the graph display, e.g., Sine Pattern from the Signal Processing | Sig Generation subpalette,
  - Use Reshape Array from the Programming | Array subpalette to form a  $32 \times 16$  2-D array (32 pages of 16 bytes each),
  - Duplicate the “Write to array” subdiagram of the case structure, and
  - Wrap the operation in a for-loop structure.

Note that the 5-ms delay *must* be used between page writes.

## 30.4 Integrated Project Ideas

Now that you know how to use the serial EEPROM consider integrating it with other devices to create a complete system, for example:

- *EEPROM Programmer* (59)

### 30.5 For More Information

- *25LC040A Data Sheet* by Microchip ~ Data sheet for the Microchip 25LC040A serial EEPROM:

<http://www.microchip.com/25LC040A>

- *M68HC11 Reference Manual* by Freescale Semiconductor ~ Refer to Section 8 for a complete treatment of the SPI serial bus standard, including timing diagrams and multi-master systems:

[http://www.freescale.com/files/](http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf)

[microcontrollers/doc/ref\\_manual/M68HC11RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf)



# 31 Bluetooth Module

Bluetooth radios provide short-range (up to 20 m) data communication between mobile appliances such as smartphones, laptop computers, headsets, and audio systems. The Roving Networks (now Microchip) RN42 Bluetooth module at the heart of the Digilent PmodBT2 pictured in Figure 31.1 provides a simple-to-use interface between an embedded controller and another Bluetooth-capable device. The PmodBT2 module serve as a wireless alternative to a UART-style data cable.

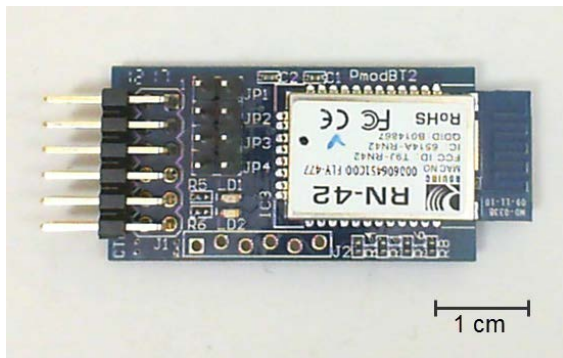


Figure 31.1: Digilent PmodBT2 module based on the Roving Networks RN42.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the essential features and pinout of the RN42 Bluetooth module,

2. Pair the RN42 with a laptop or desktop computer to create a virtual COM port,
3. Access the COM port interactively with a terminal emulator and with a LabVIEW VI,
4. Establish a wireless data link, and
5. Configure the RN42 name, pairing code, and profile.

## 31.1 Component Demonstration

Follow these steps to demonstrate correct operation of the Bluetooth module.

**Select these parts from the NI myRIO Embedded Systems Kit:**

- Bluetooth module (PmodBT2), [http://digilentinc.com/Data/Products/PMOD-BT2/PmodBT2\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-BT2/PmodBT2_rm.pdf)
- Jumper wires, F-F (8×)

You will also need a Bluetooth-capable laptop or desktop computer.

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 31.2 on page 143; the Bluetooth module requires eight connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply (VCC3V3) → B/+3.3V (pin 33)
2. Ground (GND) → B/GND (pin 30)
3. TX → B/UART.RX (pin 10)

4. RX → B/UART.TX (pin 14)
5. STATUS → B/DIO0 (pin 11)
6. RTS → B/DIO1 (pin 13)
7. CTS → B/DIO2 (pin 15)
8. RESET → B/DIO3 (pin 17)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Bluetooth demo.lvproj` contained in the subfolder `Bluetooth demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Older desktop computers equipped with a serial communications (“COM”) port could connect directly to the NI myRIO UART and exchange data. The RN42 defaults to the Serial Port Profile (SPP), meaning that the RN42 appears to the desktop as a virtual COM port and appears to the myRIO as a physical COM port. In other words the RN42 wirelessly extends the desktop COM port to the myRIO.

The demo VI displays data characters received by the RN42 from the Bluetooth-capable laptop or desktop computer and transmits the NI myRIO system date and time when the characters “d” and

“t” are received. The incoming character indicator uses the “backslash codes” display mode so that non-printing characters such as carriage return appear as “\r” and the ASCII hex code indicator shows the associated hexadecimal code of the character. The VI also displays the lower four bits of the received character on the myRIO onboard LEDs. The RN42 “STATUS” and “RTS” outputs are displayed as LED indicators, and its “RESET” and “CTS” inputs are controlled with the pushbuttons.

Press RESET to reset the module; you should see that the green status LED on the PmodBT2 is off. Release the RESET button and you should see the status LED blinking at medium speed to indicate that the RN42 is discoverable and waiting for a connection.

Pair the RN42 with your Bluetooth-capable laptop or desktop computer using the default RN42 pairing code “1234.” View the video tutorial *Bluetooth Demo Walk-Through* ([youtu.be/LFCThGa681A](http://youtu.be/LFCThGa681A), 15:08) through 2:30 to see the step-by-step Bluetooth pairing procedure for the Windows 7 operating system; the pairing process is similar for Mac OS. Make note of the COM port that was created for the RN42 SPP service.

Download and run the PuTTY terminal emulator from <http://www.putty.org>. Make the following settings (this is demonstrated in the same video tutorial beginning at 2:52):

1. Connection type = Serial,
2. Serial line = the COM port created from pairing the RN42,
3. Speed = 115200 baud (click the “Serial” category on the bottom left),
4. Data bits = 8,
5. Stop bits = 1, and
6. Flow control = None.

You may wish to save this configuration so that you can load it later on; type a name for “Saved Sessions” and then click the “Save” button.

Click “Open” to open the connection to the COM port. Observe that the PmodBT2 green status LED

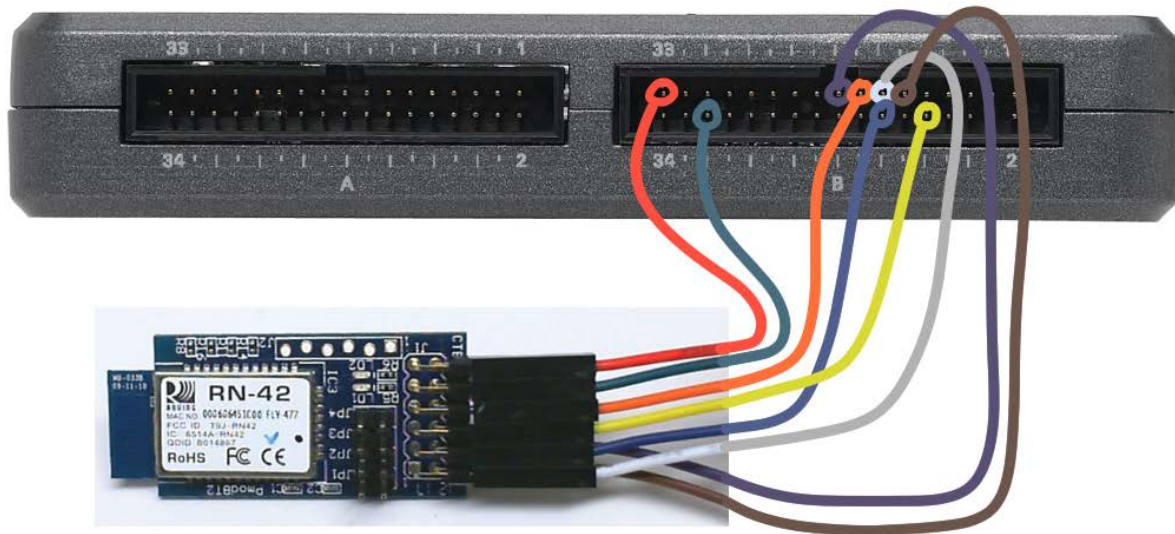
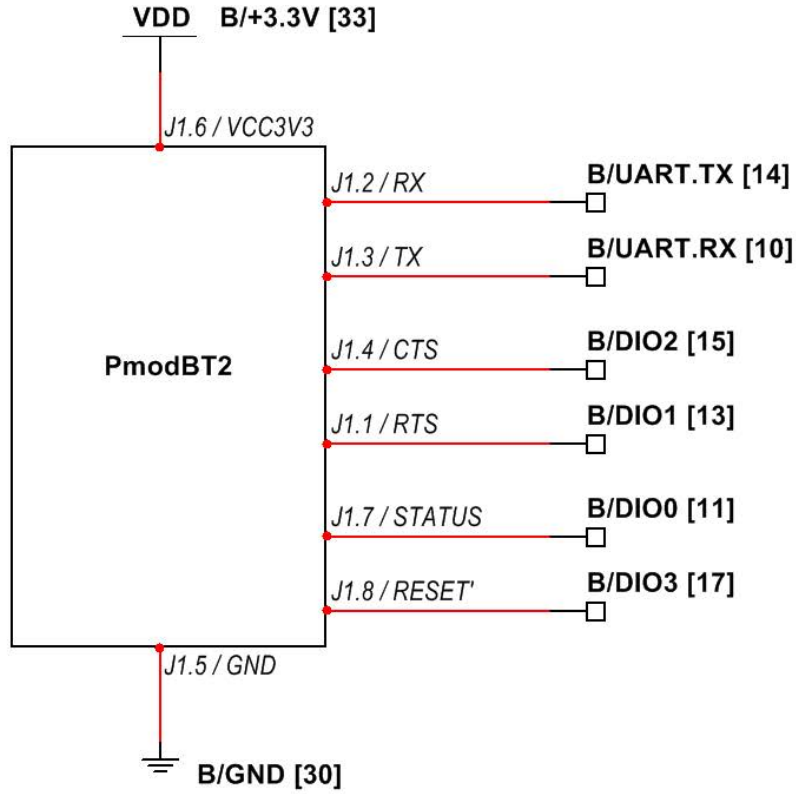


Figure 31.2: Demonstration setup for Bluetooth module connected to NI myRIO MXP Connector B.

goes dark, and you should also expect to see the STATUS front-panel indicator active; these two indications show that the RN42 has established a data connection with another Bluetooth device. Type characters into the PuTTY terminal emulator window and expect to see those same characters appear on the VI front panel. Type “t” and “d” and confirm that you see the NI myRIO system time and date. Click **enable echo** and type some more characters; you should now see the characters as you type them in the PuTTY terminal emulator window.

Click the CTS (“Clear To Send”) button and type several characters. When CTS is active the RN42 suspends transmitting from its UART and you should observe that no characters are received by the NI myRIO. Click the CTS button again and you should see the buffered characters appear in quick succession on the incoming character indicator.

Close the PuTTY window; you will see the PmodBT2 green status LED blinking again and the STATUS front-panel indicator inactive to show that the RN42 is waiting for a new data connection.

Open a PuTTY window again and type \$\$\$ to enter command mode. By now the default 60-second configuration time window has expired and nothing special will happen. Observe the PmodBT2 green status LED as you click the RESET button two times; do you notice how the blink rate is faster now? This medium-speed rate indicates that the RN42 can accept a request to enter command mode. Type \$\$\$ again and you will see the CMD prompt appear; also observe that the status LED blinks at high speed while in command mode. Type “h” to display help on all of the available commands. In particular, try “d” to display basic settings, “e” to display extended settings, and “v” to show the firmware version. Type -- to exit command mode.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct PmodBT2 terminals — double-check your connections, and ensure that you have connected the NI myRIO UART “receive” input to the RN42 “transmit” output; also check that you have not accidentally crossed the power supply connections.

## 31.2 Interface Theory

**Interface circuit:** The Roving Networks RN42 is a self-contained full-qualified Bluetooth Class 2 module with integrated antenna. By default the RN42 operates in “slave” mode, may be paired with the code “1234,” and offers the Serial Port Profile (SPP) service. By connecting the NI myRIO UART directly to the RN42 UART, the RN42 appears as a virtual COM port on a Bluetooth-capable laptop or desktop computer.

Study the video *Bluetooth Interfacing Theory* ([youtu.be/WidjSMNU1QM](https://youtu.be/WidjSMNU1QM), 15:34) to learn more about the RN42 as the basis of a wireless *serial cable replacement* application, including essential RN42 features and PmodBT2 pinout, pairing the RN42 with a computer, accessing the virtual COM port with a terminal emulator and with a LabVIEW desktop application, and accessing the RN42 UART with the NI myRIO low-level UART VIs.

## 31.3 Basic Modifications

Study the video *Bluetooth Demo Walk-Through* ([youtu.be/LFCThGa681A](https://youtu.be/LFCThGa681A), 15:08) to learn the design principles of Bluetooth demo, and then



try making these modifications to the block diagram of `Main.vi`:

1. Create a desktop LabVIEW VI to interact with the COM port; review *Bluetooth Interfacing Theory* ([youtu.be/WidjSMNU1QM](https://youtu.be/WidjSMNU1QM), 15:34) for ideas.
2. Add additional cases to return other internal states of the NI myRIO. For example, return the value of the onboard accelerometer X-axis when the “x” character is received.
3. Add a data streaming feature: when the “a” character is received begin transmitting the three onboard accelerometer values once every 100 ms (three space-delimited values followed by a carriage-return and linefeed combination); continue streaming out data until the “a” character is received again.

Microchip ~ Complete information on configuring the RN42 while in command mode; select the “Bluetooth Advanced User Manual” PDF document at the bottom of the page:

<http://microchip.com/RN42>

- *Bluetooth Basics* by SparkFun ~ Excellent tutorial on Bluetooth and wireless mobile devices:

<http://learn.sparkfun.com/tutorials/bluetooth-basics/all>

- *Bluetooth Resources* by Bluetooth Developer Portal ~ Comprehensive site devoted to designing with Bluetooth:

<http://developer.bluetooth.org>

### 31.4 Integrated Project Ideas

Now that you know how to use the Bluetooth module consider integrating it with other devices to create a complete system, for example:

- *Wireless Sensor* (41)

### 31.5 For More Information

- *PmodBT2 Reference Manual* by Digilent ~ Reference manual for the Bluetooth module:

[http://digilentinc.com/Data/Products/PMOD-BT2/PmodBT2\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-BT2/PmodBT2_rm.pdf)

- *PmodBT2 Schematics* by Digilent ~ Schematic diagram of the Bluetooth module:

[http://digilentinc.com/Data/Products/PMOD-BT2/PmodBT2\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-BT2/PmodBT2_sch.pdf)

- *RN42 Datasheet* by Microchip ~ Datasheet for RN42 Bluetooth module; select the “RN42” PDF document at the bottom of the page:

<http://microchip.com/RN42>

- *Bluetooth Data Module Command Reference and Advanced Information User’s Guide* by



# 32 Digital Potentiometer

A digital potentiometer is a three-terminal variable resistor that behaves just like its mechanical analog potentiometer counterpart (see Chapter 7 on page 27) but with a digital numerical value to set the virtual knob position. The digital potentiometer provides a convenient replacement for mechanical potentiometers, offers computer-adjustable gain control for amplifiers, and software adjustment of trim potentiometers used to null resistive sensor circuit offsets. Figure 32.1 pictures the NI myRIO Embedded Systems Kit digital potentiometer with 10 k $\Omega$  end-to-end resistance and 8-bit resolution; SPI serial bus conveys the digital wiper position.

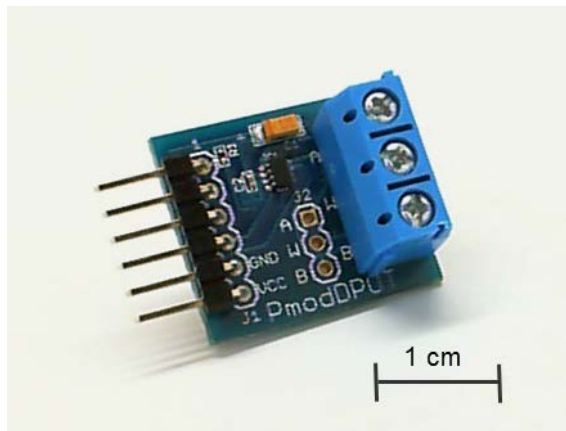


Figure 32.1: NI myRIO Embedded Systems Kit digital potentiometer.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Adjust the digital potentiometer wiper position with SPI serial communications,
2. Properly connect and operate the digital potentiometer as either a rheostat or as a potentiometer (voltage divider), and
3. Discuss the principles of the “virtual wiper” as implemented by an array of solid-state switches.

## 32.1 Component Demonstration

Follow these steps to demonstrate correct operation of the digital potentiometer.

**Select these parts from the NI myRIO Embedded Systems Kit:**

- Digital potentiometer (PmodDPOT),  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,1075&Prod=PMOD-DPOT>
- Jumper wires, F-F (5 $\times$ )
- Jumper wires, M-F (3 $\times$ )
- Small screwdriver

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 32.2 on page 149; the digital potentiometer requires five connections to NI myRIO MXP Connector A and three connections to MXP Connector B (see Figure A.1 on page 233):

1. +5-volt supply  $\rightarrow$  A/+5V (pin 1)

2. Ground → A/GND (pin 6)
3. SPI receiver → A/SPI.MOSI (pin 25)
4. SPI clock → A/SPI.CLK (pin 21)
5. Chip select → A/DIO0 (pin 11)
6. “A” → B/+5V (pin 1)
7. “B” → B/GND (pin 6)
8. “W” → B/AI0 (pin 3)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Dpot demo.lvproj` contained in the subfolder `Dpot demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI provides a front-panel slide control to set the virtual wiper position as an 8-bit value. The five-volt power supply connected across the end terminals of the potentiometer creates a proportional variable voltage at the wiper terminal “W” which is read by an analog input and displayed on the indicator dial. Move the slide and you should see a corresponding change on the dial position. Press the page-up and page-down keys to make single-bit changes to the digital value.

If you have an ohmmeter handy, disconnect all three potentiometer terminals from NI myRIO and then measure the resistance between the “W” and “B” terminals as you vary the digital wiper position; repeat for the “W” and “A” terminals. If the measurement does not seem sufficiently stable, try connecting either “A” or “B” to one of the NI myRIO ground terminals.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A to power the digital potentiometer board and Connector B to establish the potentiometer variable voltage, and
- Correct SPI connector terminals — double-check your connections, and ensure that you have connected the NI myRIO SPI “MOSI” output to the digital potentiometer “SDI” input and digital output DIO0 to the chip select input.

## 32.2 Interface Theory

**Interface circuit:** The Digilent PmodDPOT board provides a convenient set of interface connectors for the Analog Devices AD5160 digital potentiometer. The digital potentiometer provides the conventional trio of terminals like a mechanical potentiometer, and an 8-bit value between 0 and 255 transmitted via SPI (serial peripheral interface) sets the virtual wiper position by closing exactly one of 256 solid-state switches that establish the connection point to a string of 256 equal-valued resistors between terminals “A” and “B.”

Study the video *Digital Potentiometer Interfacing Theory* ([youtu.be/C4iBQjWn70I](https://youtu.be/C4iBQjWn70I), 9:15) to learn

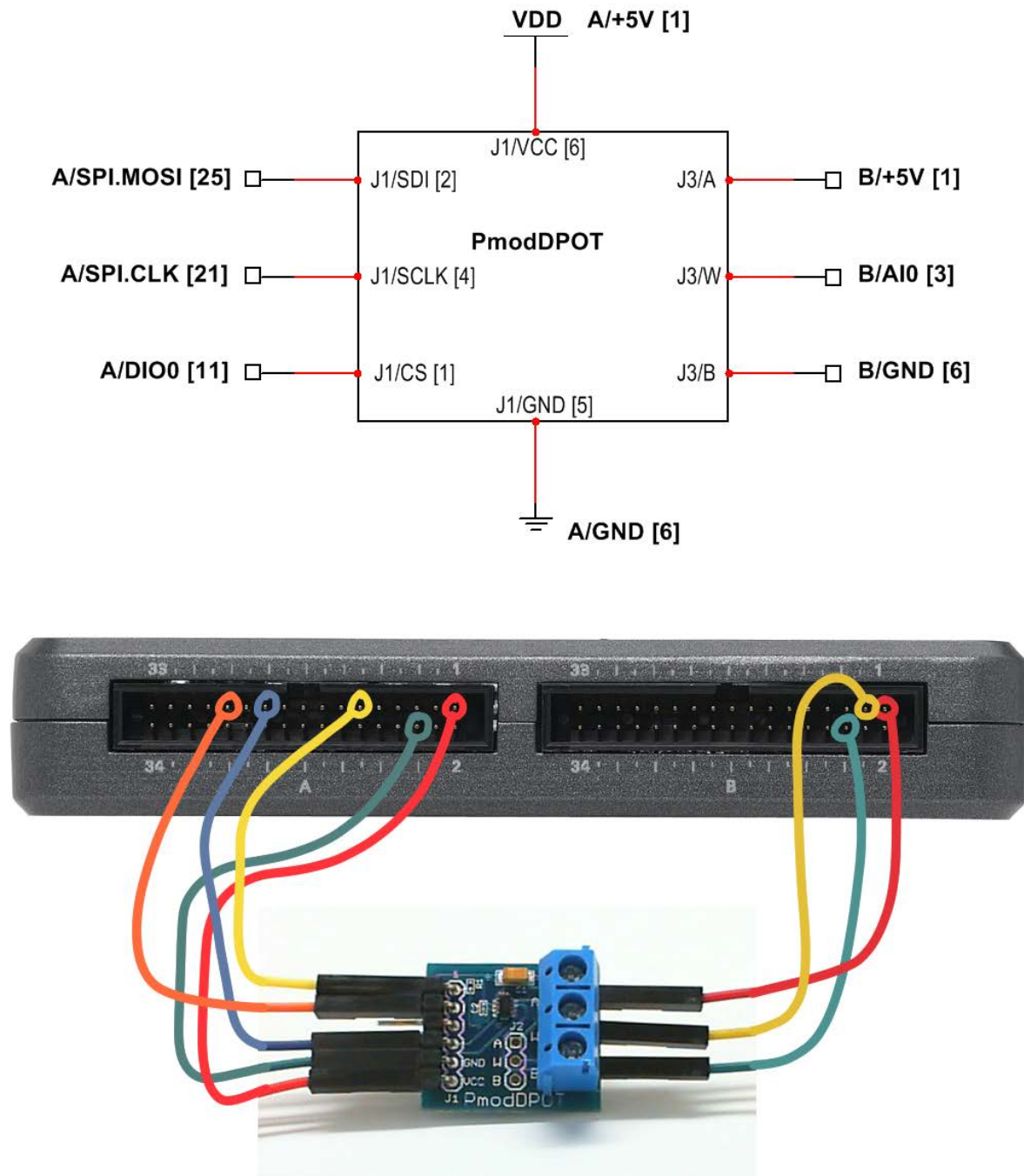


Figure 32.2: Demonstration setup for digital potentiometer connected to NI myRIO MXP Connector B. Use the M-F jumper wires and a screwdriver for the three potentiometer terminals.

more about the digital potentiometer theory of operation including SPI bus and chip select timing, internal switch array circuit, and design equations for using the digital potentiometer in rheostat mode (a single variable resistance) and in potentiometer mode (a voltage divider providing an adjustable voltage). Study *Serial Communication: SPI* ([youtu.be/GaXtDamw5As](http://youtu.be/GaXtDamw5As), 7:02) to understand how the SPI Express VI configuration options relate to the signaling waveforms between SPI transmitters and receivers.

**LabVIEW programming:** Study the video *SPI Express VI* ([youtu.be/S7KkTeMfmc8](http://youtu.be/S7KkTeMfmc8), 5:51) to learn how to use the SPI Express VI.

### 32.3 Basic Modifications

Study the video *Dpot Demo Walk-Through* ([youtu.be/dtwX0j5vvy4](http://youtu.be/dtwX0j5vvy4), 4:57) to learn the design principles of Dpot demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Interchange the “A” and “B” connections, and confirm that the analog voltage decreases as you increase the digital value.
2. Evaluate the linearity of the digital potentiometer: Change the while-loop structure to a for-loop, create an array of the analog voltage at each digital value, and then plot the analog voltage as a function of digital voltage.
3. Continue the linearity evaluation of the previous step by plotting the difference of the measured analog voltage and the ideal analog voltage. This difference plot makes it much easier to identify any trends of nonlinearity.

### 32.4 For More Information

- *PmodDPOT Reference Manual* by Digilent ~ Reference manual for the digital potentiometer board: [http://digilentinc.com/Data/Products/PMOD-DPOT/PmodDPOT\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-DPOT/PmodDPOT_rm.pdf)
- *PmodDPOT Schematics* by Digilent ~ Schematic diagram of the digital potentiometer board: [http://digilentinc.com/Data/Products/PMOD-DPOT/PmodDPOT\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-DPOT/PmodDPOT_sch.pdf)
- *AD5160 Data Sheet* by Analog Devices ~ Complete information on the AD5160 that serves as the heart of the digital potentiometer board: <http://www.analog.com/ad5160>
- *M68HC11 Reference Manual* by Freescale Semiconductor ~ Refer to Section 8 for a complete treatment of the SPI serial bus standard, including timing diagrams and multi-master systems: [http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/M68HC11RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf)

# 33 Temperature Sensor

A temperature sensor provides a useful indication of environmental conditions. Figure 33.1 pictures the NI myRIO Embedded Systems Kit temperature sensor based on the Microchip TCN75A temperature sensor with I<sup>2</sup>C-bus serial communications. The sensor offers  $\pm 1^{\circ}\text{C}$  accuracy over the range  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ , nine to twelve bits resolution, and conversion times from 30 to 240 ms. The sensor also provides an “alert” output that triggers when the measured temperature exceeds a user-adjustable limit.

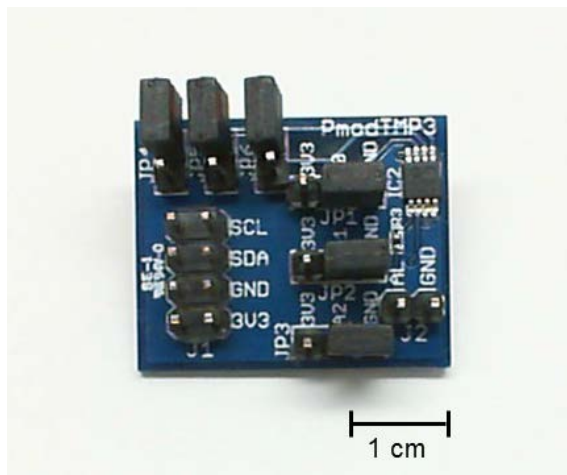


Figure 33.1: NI myRIO Embedded Systems Kit temperature sensor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the continuous conversion, one-shot, and shutdown operating modes,
2. Configure the ALERT output polarity, comparator, and interrupt modes, and
3. Read and interpret the ambient temperature.

## 33.1 Component Demonstration

Follow these steps to demonstrate correct operation of the temperature sensor.

**Select these parts from the NI myRIO Embedded Systems Kit:**

- Temperature sensor (PmodTMP3), [http://digilentinc.com/Data/Products/PMOD-TMP3/PmodTMP3-rm-revA\\_1.pdf](http://digilentinc.com/Data/Products/PMOD-TMP3/PmodTMP3-rm-revA_1.pdf)
- Jumper wires, F-F (5×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 33.2 on page 153; the temperature sensor requires five connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply → B/+3.3V (pin 33)
2. Ground → B/GND (pin 30)
3. Serial data (SDA) → B/I2C.SDA (pin 34)
4. Serial clock (SCL) → B/I2C.SCL (pin 32)

Ensure that *all* PmodTMP3 jumpers are set exactly as shown.

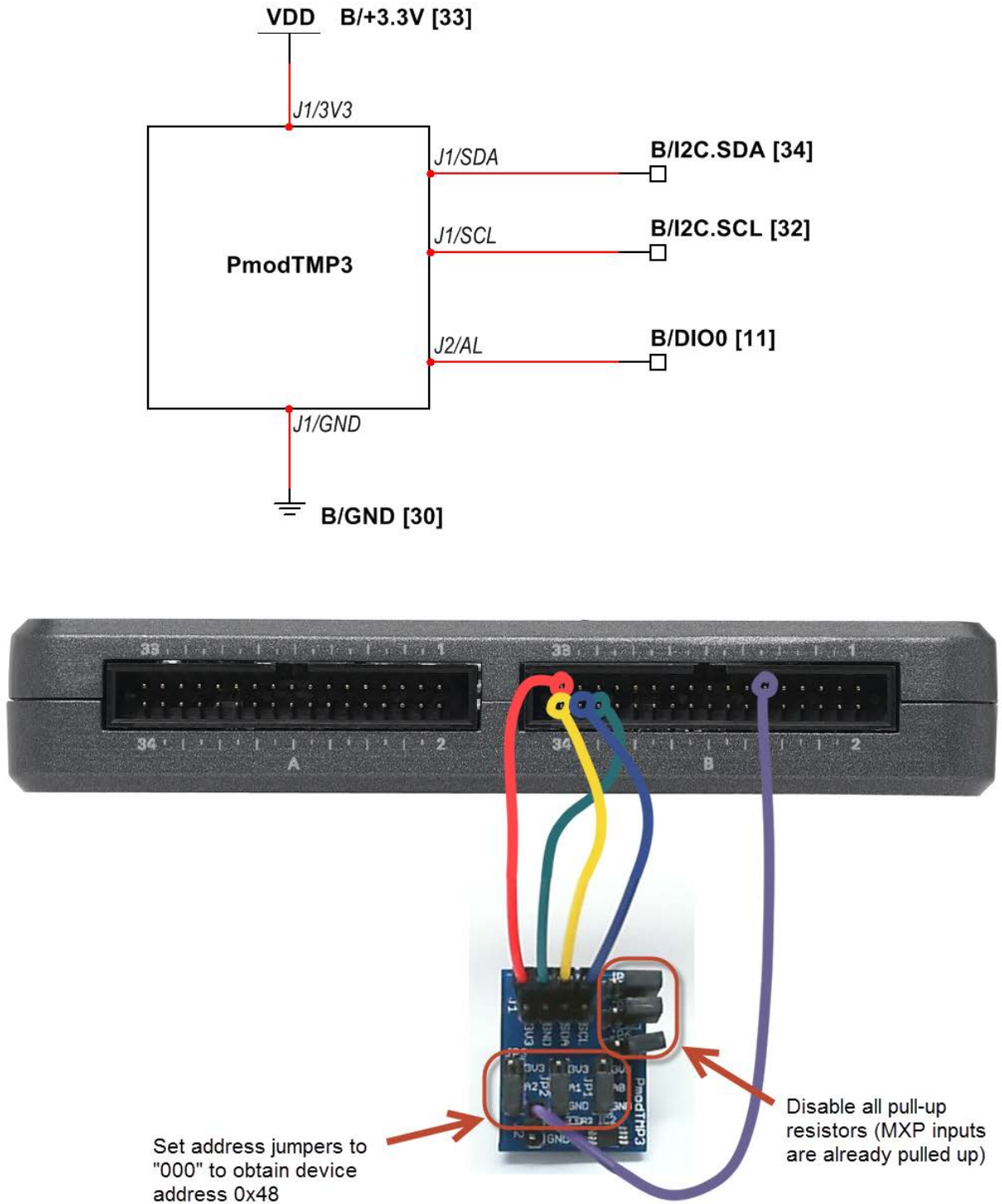


Figure 33.2: Demonstration setup for temperature sensor connected to NI myRIO MXP Connector B. Set *all* jumpers exactly as shown.



**Run the demonstration VI:**

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Temperature Sensor demo.lvproj` contained in the subfolder `Temperature Sensor demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the `Run` button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the ambient temperature in degrees Celsius as measured by the Microchip TCN75A temperature sensor with an accuracy of  $\pm 1^\circ\text{C}$ ; expect to see a temperature reading that matches your room temperature. Double-click the maximum and minimum values of the thermometer indicator to change its display range. Try heating the sensor by a finger touch or by using a drinking straw or hair dryer to blow warm air on the thermistor. What is the highest temperature that you observe?

Use a plastic sandwich bag filled with an ice cube or crushed ice. Surround the temperature sensor with ice and you should observe the measured temperature going down. What is the lowest temperature that you observe?

The alert LED indicator displays the state of the TCN75A “ALERT” output. You should

observe that the alert indicator asserts when the temperature exceeds  $26^\circ\text{C}$  and then de-asserts when the temperature falls below  $24.5\pm 1^\circ\text{C}$ .

The VI displays the contents of the four TCN75A registers: ambient temperature (primary measurement), configuration, temperature limit (to activate the ALERT output), and temperature hysteresis lower-limit; stop the VI, edit the block diagram, save and re-run to change the values in these last three registers.

Click the `Stop` button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black `Run` button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct `PmodTMP3` connector terminals — double-check your connections, and ensure that you have connected the NI myRIO I<sup>2</sup>C-bus “SDA” line to the `PmodTMP3` “SDA” terminal on connector J1 and the “SCL” line to the “SCL” terminal; also check that you have not accidentally crossed the power supply connections, and
- Correct `PmodTMP3` address jumper settings — refer again to the circled region of Figure 33.2 on the next page for the proper settings.

NOTE: Double-check the “SDA” and “SCL” connections should you see the message “Error -36011 occurred at myRIO Write I2C.vi” or similar; this message indicates that NI myRIO did not receive an expected acknowledgement from the `PmodTMP3` I<sup>2</sup>C-bus interface.

### 33.2 Interface Theory

**Interface circuit:** The Microchip TCN75A temperature sensor packaged by the PmodTMP3 supports the I<sup>2</sup>C-bus serial interface (commonly denoted “I2C”). Two 8-bit registers provide the measured temperature in degrees Celsius every 30 ms for 9-bit resolution and every 240 ms for 12-bit resolution; each additional bit doubles the conversion time. The open-drain “ALERT” output provides a setpoint limit to indicate when the temperature exceeds a user-defined limit.

Study the video *Temperature Sensor Interfacing Theory* ([youtu.be/HwzTgYp5nF0](https://youtu.be/HwzTgYp5nF0), 10:03) to learn how to read the measured ambient temperature, how to configure the device to adjust the resolution and “ALERT” output behavior including polarity, temperature limit setpoint, hysteresis, and fault queue (the number of consecutive conversion cycles for which a fault condition must be true before asserting the ALERT output), and how to put the device into shutdown mode (power-saving idle) with one-shot measurement on demand. Study *Serial Communication: I2C* ([youtu.be/7CgNF78pYQM](https://youtu.be/7CgNF78pYQM), 8:47) to understand how the I2C Express VI configuration options relate to the signaling waveforms between I<sup>2</sup>C-bus transmitters and receivers.

### 33.3 Basic Modifications

Study the video *Temperature Sensor Demo Walk-Through* ([youtu.be/10ib10sojds](https://youtu.be/10ib10sojds), 6:25) to learn the design principles of Temperature Sensor demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Change the device address to some other value. What LabVIEW error message do you see?
2. Add the necessary computation to display the temperature in degrees Fahrenheit.
3. Move the configuration register code inside the while-loop and convert the Boolean array

constant to a front-panel control; in this way you can more easily experiment with the configuration options.

4. Make front-panel controls for the temperature limit setpoint and hysteresis values in degrees Celsius, and then move the associated I2C Express VIs inside the while-loop so that these values can be adjusted while the VI is running. Confirm that the alert indicator behaves as expected when you adjust these two temperature-related controls. What happens when the hysteresis value is higher than the limit setpoint value?
5. Add a waveform chart indicator for measured temperature and then change the while-loop time interval to a much larger value such as one minute. Plot the long-term temperature profile over a 24-hour period.
6. Repeat the previous step, and modify the block diagram code so that the temperature sensor operates in shutdown mode and performs a one-shot measurement on demand. Remember to wait for the necessary conversion time before reading the temperature register: 30 ms for 9-bit resolution, and double the time for each additional bit up to 240 ms for 12-bit resolution.
7. Add a minimum and maximum recorded temperature display with reset buttons.

### 33.4 Integrated Project Ideas

Now that you know how to use the temperature sensor consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Digital Thermometer* (45)
- *On-Off Control System* (52)
- *Weather Station* (58)

### 33.5 For More Information

- *PmodTMP3 Reference Manual* by Digilent ~ Reference manual for the temperature sensor:  
[http://digilentinc.com/Data/Products/PMOD-TMP3/PmodTMP3-rm-revA\\_1.pdf](http://digilentinc.com/Data/Products/PMOD-TMP3/PmodTMP3-rm-revA_1.pdf)
- *PmodTMP3 Schematics* by Digilent ~ Schematic diagram of the temperature sensor:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,1202&Prod=PMOD-TMP3>
- *TCN75A Datasheet* by Microchip ~ Datasheet for the TCN75A 2-Wire Serial Temperature Sensor:  
<http://www.microchip.com/TCN75A>
- *UM10204 I<sup>2</sup>C-bus Specification and User Manual* by NXP Semiconductors ~ A complete treatment of the I<sup>2</sup>C-bus standard, including timing diagrams and multi-master systems:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)



# 34 MEMS Microphone

A microphone serves as an acoustic sensor to record audio signals and monitor acoustic level. The Analog Devices ADMP504 microphone pictured in Figure 34.1, a MEMS (micro electro-mechanical sensor) device with an on-board amplifier, finds application in smartphones. The device is packaged on a DIP carrier for use on a breadboard.

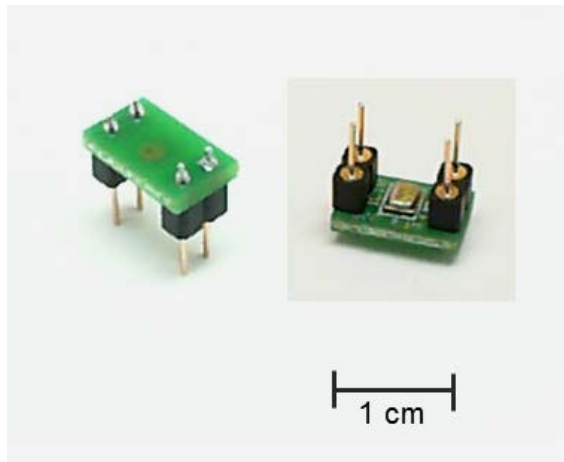


Figure 34.1: NI myRIO Starter Kit MEMS microphone. The photo on the right shows the microphone mounted to the underside of the DIP carrier.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the output characteristics of the

- ADMP504 MEMS microphone,
- Block the DC component of the microphone output,
- Amplify the microphone signal with an op amp, and
- Read the microphone signal with an analog input.

## 34.1 Component Demonstration

Follow these steps to demonstrate correct operation of the microphone.

**Select these parts from the NI myRIO Starter Kit:**

- ADMP504 ultra-low-noise microphone, <http://www.analog.com/ADMP504>
- OP37 low-noise precision high-speed op amp, <http://www.analog.com/OP37>
- AD8541 rail-to-rail single-supply op amp, <http://www.analog.com/AD8541>
- 0.1  $\mu\text{F}$  ceramic disk capacitor, marking "104" , <http://www.avx.com/docs/Catalogs/class3-sc.pdf>
- 1.0  $\mu\text{F}$  electrolytic capacitor , <http://industrial.panasonic.com/www-data/pdf/ABA0000/ABA0000CE12.pdf>
- Resistor, 10  $\text{k}\Omega$  (3 $\times$ )
- Resistor, 100  $\text{k}\Omega$
- Breadboard
- Jumper wires, M-F (5 $\times$ )
- 3.5 mm stereo audio cable (provided with NI myRIO)

- Test clips (2×)

**Build the interface circuit:** Refer to the schematic diagram and recommended breadboard layout shown in Figure 34.3 on the next page. The interface circuit requires four connections to NI myRIO Connectors B and C (see Figures A.1 on page 233 and A.2 on page 234):

1. +15-volt power supply → C/+15V (pin 1)
2. -15-volt power supply → C/-15V (pin 2)
3. +3.3-volt power supply → B/+3.3V (pin 33)
4. Ground → C/AGND (pin 3)
5. Mic output → AUDIO IN

Connect the 3.5 mm stereo audio cable to the AUDIO IN. Use test clips to connect the other plug tip (left channel) to the mic output and the plug sleeve to ground; refer to Figure 34.2.

Alternatively you may build the interface circuit designed for direct connection to the MXP analog input; refer to Figure 34.4 on page 160. This version requires five connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +5-volt power supply → B/+5V (pin 1)
2. Ground → B/GND (pin 6)
3. +3.3-volt power supply → B/+3.3V (pin 33)
4. Ground → B/GND (pin 30)
5. Mic amplifier output → B/AIO (pin 3)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project MEMS Microphone demo.lvproj contained in the subfolder MEMS Microphone demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and

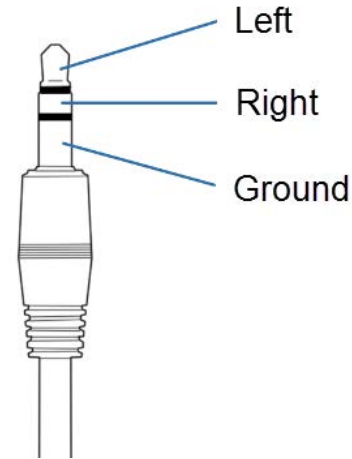


Figure 34.2: 3.5 mm stereo audio cable plug connections for ground, left channel, and right channel.

- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R. Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI displays the audio signal detected by the microphone as an oscilloscope display. Select the appropriate mic input depending on the interface circuit you built: AudiIn/Left or B/AIO. Also, double-click the upper and lower limits of the waveform chart and set to -2.5 and 2.5 for AudiIn/Left and to 0 and 5 for B/AIO.

Try whistling, speaking, singing, or make any other sound and you should see the corresponding waveform. Note that the waveform is centered about zero when using the audio input and centered about approximately 2.5 volts when using the analog input.

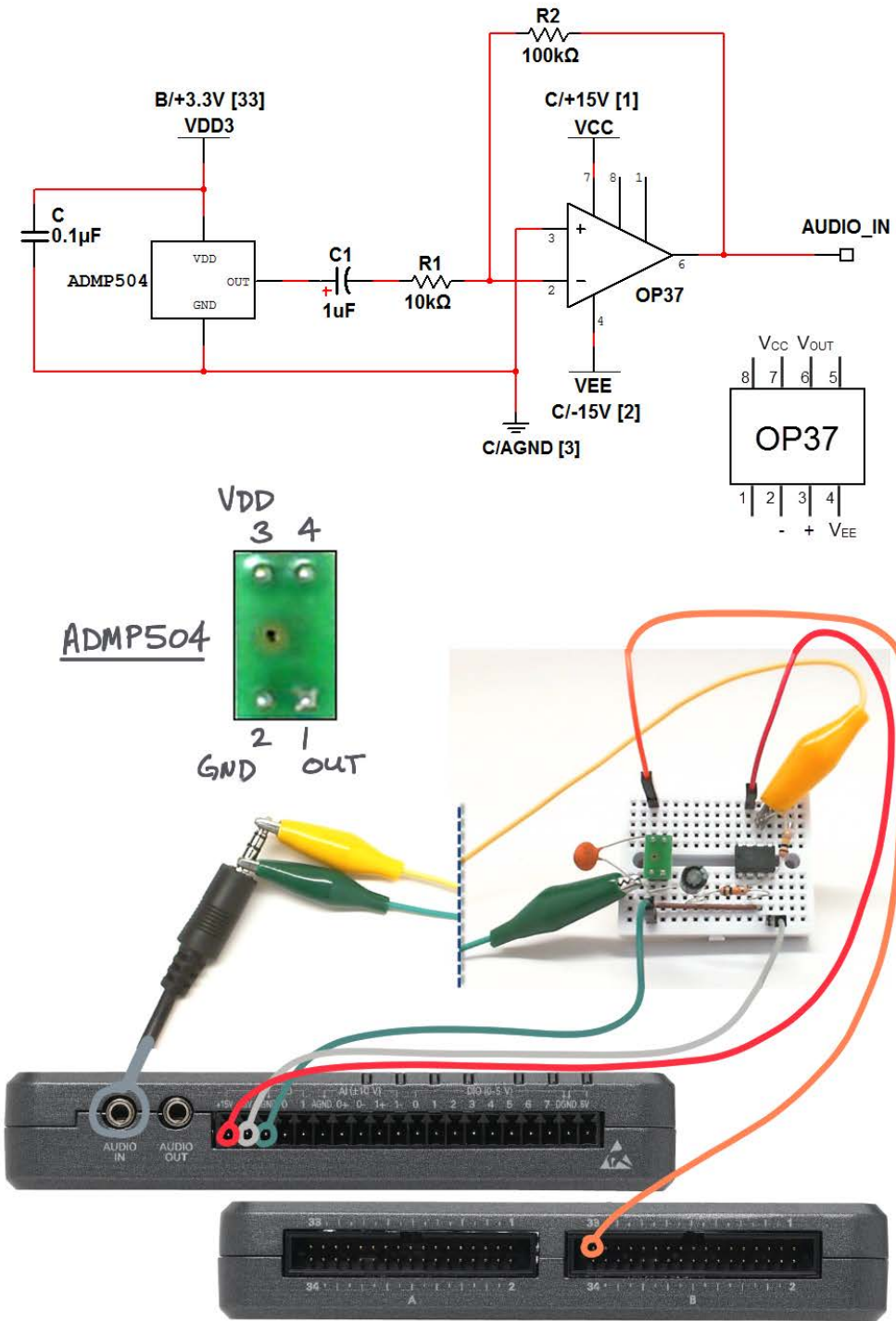


Figure 34.3: Demonstration circuit for MEMS microphone with AUDIO IN: schematic diagram, recommended breadboard layout, and connection to NI myRIO Connectors B and C.

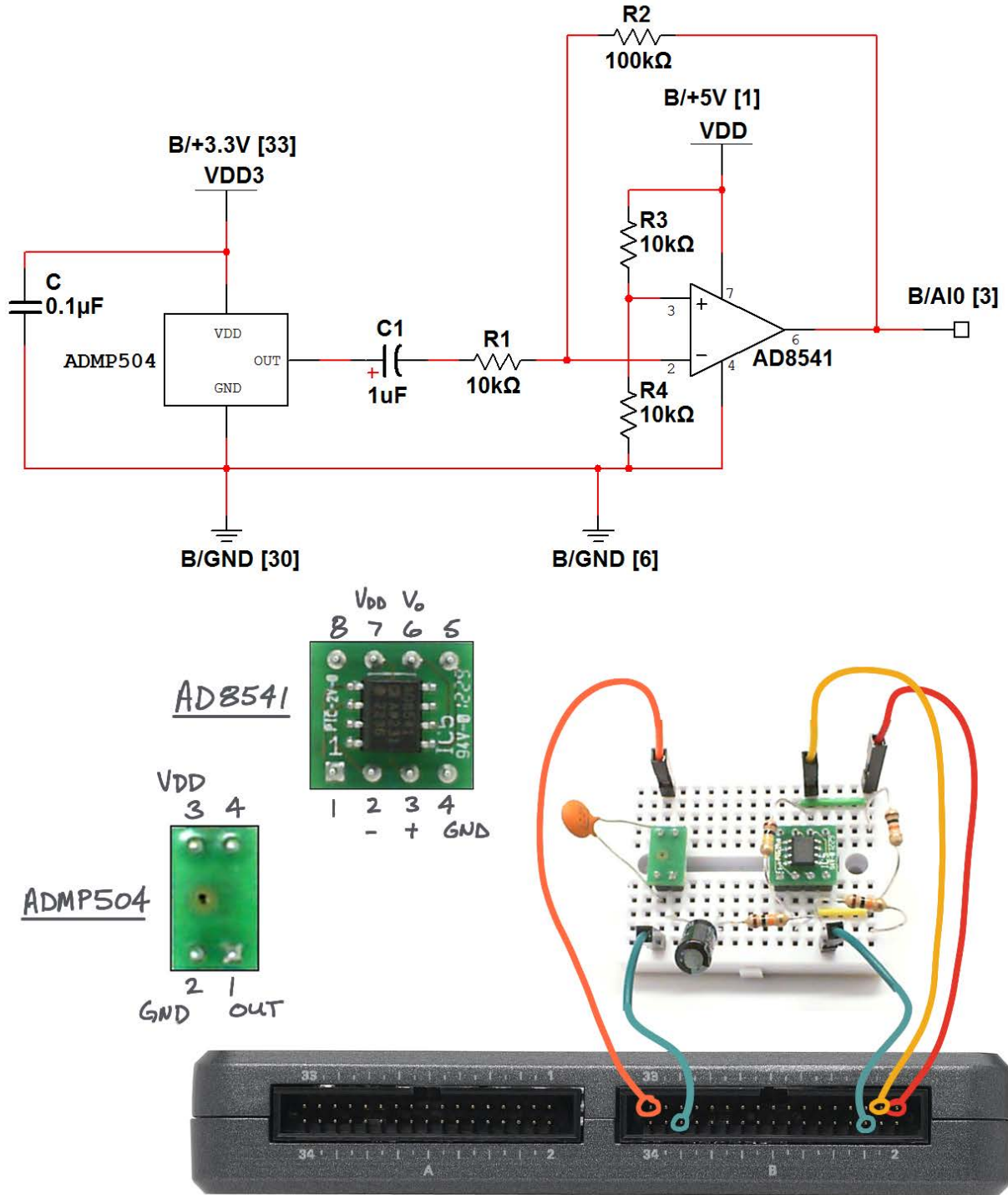


Figure 34.4: Demonstration circuit for MEMS microphone with analog input (AI): schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.



Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B (and possibly Connector C) and that you have the correct pin connections, and
- Use the B/A10 (pin 3) jumper wire as probe to check the following signal points:
  - ADMP504 output: 0.8 volt DC offset with signal at up to  $\pm 0.25$  volts
  - Negative terminal of 1  $\mu$ F capacitor: same as ADMP504 output but with zero DC offset (for OP37) or 2.5 V offset (AD8541)
  - Noninverting (+) op amp terminal: zero (for OP37) or 2.5 V offset (AD8541)
  - Inverting (–) op amp terminal: zero (for OP37) or 2.5 V offset (AD8541); if some other level, double-check all of your op amp connections or try another op amp

## 34.2 Interface Theory

**Interface circuit:** The ADMP504 produces a maximum voltage of 0.25 volts with a DC offset of 0.8 V. Study the video *ADMP504 Microphone Interfacing Theory* ([youtu.be/991pj7yUmuY](http://youtu.be/991pj7yUmuY), 6:36) to learn about the microphone characteristics, the need for a power supply bypass capacitor, the DC blocking capacitor, and the inverting amplifier that boosts microphone output to level matched to the  $\pm 2.5$  V input range of the audio input and analog input.

## 34.3 Basic Modifications

Study the video *MEMS Mic Demo Walk-Through* ([youtu.be/2ZpI\\_uDwOg4](http://youtu.be/2ZpI_uDwOg4), 2:31) to learn the design principles of MEMS Microphone demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add a real-time amplitude spectrum display; use the point-by-point version of the built-in Amplitude and Phase Spectrum VI located on the Signal Processing | Point by Point | Spectral subpalette.
2. Add a VU (volume) meter that displays signal intensity; use the AC output of the built-in AC & DC Estimator PtByPt VI located on the Signal Processing | Point by Point | Sig Operation subpalette.
3. Create an anti-aliasing filter by adding a parallel capacitor with the feedback resistance.

## 34.4 Integrated Project Ideas

Now that you know how to use the MEMS microphone consider integrating it with other devices to create a complete system, for example:

- *Handheld Meter* (40)
- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *Guitar Tuner* (55)

## 34.5 For More Information

- *Microphone Array Beamforming with the ADMP504* by Analog Devices ~ Video demonstration of two ADMP504 microphones combined with DSP (digital signal processing) to create a virtual directional microphone:

<http://videos.analog.com/video/products/MEMS-sensors/1979997938001/Microphone-Array-Beamforming-with-the-ADMP504>

- *ADMP504 Flex Eval Board* by Analog Devices ~  
The ADMP504 packaged with a bypass capacitor  
and extension wires:

<http://www.analog.com/en/evaluation/>

EVAL-ADMP504Z-FLEX/eb.html

## **Part IV**

# **Additional Devices**



# 35 H-Bridge and Geared Motor

An H-bridge motor drive uses four power MOSFETs to direct current through a DC motor in one direction or the other, thereby allowing the motor controller to reverse the motor direction as needed. The MOSFETs also switch fast enough that pulse-width modulation (PWM) of the pair of active MOSFETs can adjust the motor speed. The H-bridge and geared motor included with the NI myRIO Mechatronics Kit (Figure 35.1) provides relatively high torque and is suitable for the drive train of a robotic platform. Moreover, the integrated shaft position quadrature encoder provides feedback suitable for motor position and speed control.



Figure 35.1: NI myRIO Mechatronics Kit H-bridge and geared motor.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the H-bridge principle of operation,
2. Explain how to control the motor speed and direction of rotation,
3. Sketch the quadrature encoded output waveforms of the shaft position sensor for clockwise and counterclockwise rotation,
4. Understand how to protect the MOSFET transistors from momentary short circuit current when changing the motor direction,
5. Apply the LabVIEW PWM Express VI to control the motor speed, and
6. Apply the LabVIEW Encoder Express VI to measure the motor shaft position and speed.

## 35.1 Component Demonstration

Follow these steps to demonstrate correct operation of the H-bridge and geared motor.

**Select these parts from the NI myRIO Mechatronics Kit:**

- H-bridge (PmodHB5), [http://digilentinc.com/Data/Products/PMOD-HB5/PmodHB5\\_RevD\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-HB5/PmodHB5_RevD_rm.pdf)
- Geared motor, <http://digilentinc.com/Products/Detail.cfm?NavPath=2,403,625&Prod=MT-MOTOR>
- Small screwdriver
- Jumper wires, F-F (6×)
- Jumper wires, M-F (2×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 35.2 on the facing page; the H-bridge and geared motor requires six connections to NI myRIO MXP Connector A and two connections to MXP Connector B (see Figure A.1 on page 233):

1. J2.VM → B/+5V (pin 1)
2. J2.GND → B/GND (pin 6)
3. J1.VCC → A/+3.3V (pin 33)
4. J1.GND → A/GND (pin 30)
5. J1.EN (enable) → A/PWM0 (pin 27)
6. J1.SA (sensor A) → A/ENCA (pin 18)
7. J1.SB (sensor B) → A/ENCB (pin 22)
8. J1.DIR (direction) → A/DIO0 (pin 11)

Use a small straight-bladed screwdriver to loosen and tighten the screws on the motor supply voltage connector J2.

#### Run the demonstration VI:

- **Download** <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project H-Bridge & Geared Motor demo.lvproj contained in the subfolder H-Bridge & Geared Motor demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** Click the Enable button and increase the PWM duty (duty cycle) slider until the motor begins to turn. You should see the motor turn, the Counter Value indicator value increment in the positive direction, and the Counter Direction indicator display “Counting Up.” Continue increasing the PWM duty slider to speed up the motor.

Bring the slider back down towards zero and then continue downwards to a negative value. You should see the direction indicator change to “Counting Down” and the counter value decrementing.

Click the Reset Counter button and then experiment with the motor speed. Note that the counter now serves as a speed indicator reporting the number of “counts” measured in a 100 millisecond interval.

Try varying the PWM freq slide control to adjust the rate at which pulses are applied to the H-bridge “enable” input. Look for relationships between the PWM frequency and duty cycle especially in terms of these points:

1. What PWM frequency range makes it easiest to start the stopped motor when increasing the PWM duty cycle from zero?
2. What PWM frequency range causes audible tones?
3. What PWM frequency range is best for very slow rotation?

Bring the motor speed back to zero and then reset the counter, leaving the Reset Counter switch in its “off” state. Manually rotate the magnet on the back of the motor and observe the Counter Value display. How many counts do you see for one complete revolution of the magnet?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,

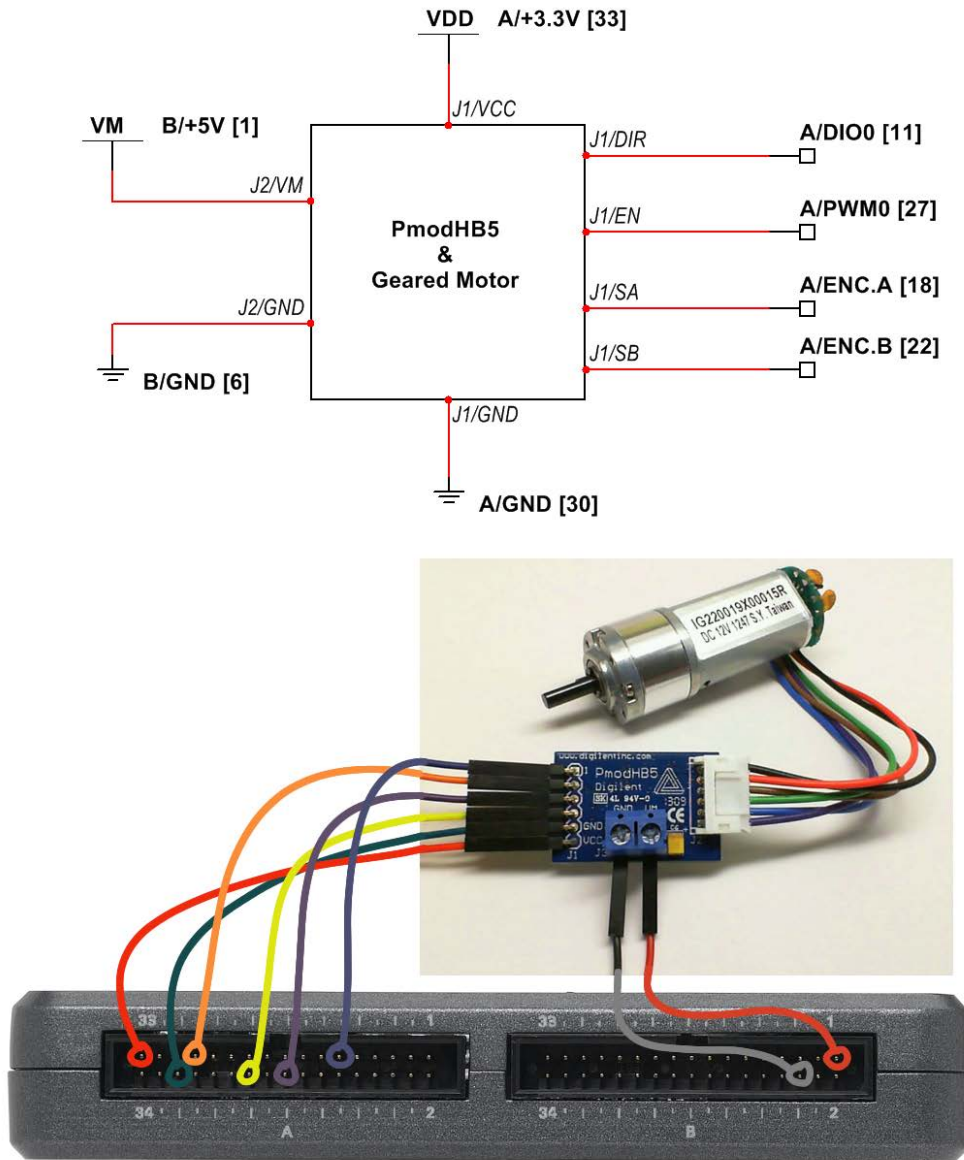


Figure 35.2: Demonstration circuit for H-bridge and geared motor: schematic diagram and connection to NI myRIO MXP Connectors A and B.

- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A for the H-bridge control and sensor signals and that you have the correct pin connections,
- Correct PmodHB5 terminals — double-check your connections, and ensure that you have not reversed the NI myRIO encoder inputs; also check that you have not accidentally crossed the power supply connections.

## 35.2 Interface Theory

**Interface circuit:** The PmodHB5 H-bridge motor driver provides motor enable and direction inputs and quadrature sensor outputs from the pair of Hall-effect sensors affixed to the back of the motor; a small circular magnet mounted to the motor shaft activates the sensors. The Sha Yang Ye IG22 geared motor connects directly to PmodHB5 via the JST-style connector. The screw terminal block on the HB5 accepts 18-gauge wire to accommodate motors up to 12 volts and 2 amps.

Study the video *H-Bridge and Geared Motor Interfacing Theory* ([youtu.be/W526ekpR8q4](https://youtu.be/W526ekpR8q4), 11:26) to learn more about the H-bridge principles of operation, PmodHB5 and IG22 features, speed control with pulse-width modulation (PWM), Hall-effect sensors and quadrature-encoded sensor signals, and an overview of closed-loop control of the motor position and speed by using the sensor signals as feedback.

**LabVIEW programming:** Study the video *PWM Express VI* ([youtu.be/mVN9jfwX1eI](https://youtu.be/mVN9jfwX1eI), 2:41) to learn how to use the PWM Express VI to create a pulse-width modulated square wave to provide variable-speed motor operation.

## 35.3 Basic Modifications

Study the video *H-Bridge and Geared Motor Demo Walk-Through* ([youtu.be/Q1UXVtVN-oQ](https://youtu.be/Q1UXVtVN-oQ), 6:01) to learn the design principles of H-Bridge & Geared Motor demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Determine the number of shaft encoder counts for one revolution of the gearbox shaft output (hint: you need to know the gearbox ratio and the number of encoder counts for one revolution of the sensor magnet), and then add block diagram code to display the gearbox output shaft angle in degrees.
2. Add a display to show the gearbox output shaft RPS (revolutions per second).
3. The block diagram has been intentionally designed to make it easy for you to add a PID (proportional/integral/derivative) controller to implement a closed-loop speed control system:
  - (a) Open up some vertical space immediately to the right of the Encoder Express VI (press “Ctrl” and left-mouse click and drag) and then disconnect the error cluster from the encoder VI,
  - (b) Place the PID controller located in the Real-Time | Function Blocks | Control subpalette in this space,
  - (c) Reconnect the error cluster to ensure that the PID VI executes *after* the Encoder VI and *before* the PWM VI,
  - (d) Create front-panel controls for the proportional gain, integral time, and derivative time values,
  - (e) Right-click on the PWM duty control and select “Change to indicator,”
  - (f) Connect the PID output to the PWM duty indicator,
  - (g) Create a control for the PID setpoint input, and
  - (h) Keep the Reset Counter switch active or change it to a true-valued constant.



Search the Internet with the phrase “Ziegler-Nichols PID tuning” to look for tutorials that discuss how to choose appropriate values for the three PID controller constants, and feel free to experiment on your own (start with proportional gain).

### 35.4 For More Information

- *PmodHB5 Reference Manual* by Digilent ~ Reference manual for the H-bridge module:  
[http://digilentinc.com/Data/Products/PMOD-HB5/PmodHB5\\_RevD\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-HB5/PmodHB5_RevD_rm.pdf)
- *PmodHB5 Schematics* by Digilent ~ Schematic diagram of the H-bridge module:  
[http://digilentinc.com/Data/Products/PMOD-HB5/PmodHB5\\_D\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-HB5/PmodHB5_D_sch.pdf)
- *Motor/Gearbox* by Digilent ~ Need more geared motors for your project? The IG22 is supplied by Digilent:  
<http://digilentinc.com/Products/Detail.cfm?NavPath=2,403,625&Prod=MT-MOTOR>
- *IG22 Geared Motor Datasheet* by Sha Yang Ye ~ Datasheet for the motor and planetary gearbox:  
<http://www.geared-motor.com/english/pdf/IG-22GM-01&02.pdf>
- *Two-Channel Encoder Datasheet* by Sha Yang Ye ~ Datasheet for the two-channel quadrature encoder:  
<http://www.geared-motor.com/english/pdf/Magnetic-Encoders.pdf>



# 36 USB Flash Drive

The NI myRIO includes a USB port with operating system support for USB flash drives to greatly extend to ability of NI myRIO to work with large data sets and to perform data logging tasks over long periods of time beyond the limitations of the onboard solid-state hard disk drive. Figure 36.1 pictures a USB flash drive inserted into the NI myRIO USB connector.



Figure 36.1: NI myRIO USB connector with USB flash drive.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Access the NI myRIO file system onboard disk drive with any of these methods:

- (a) Web browser,
  - (b) Network mapped drive,
  - (c) Interactive secure shell (SSH), and
2. Read and write files with LabVIEW.

## 36.1 Component Demonstration

Follow these steps to demonstrate correct operation of the USB flash drive.

### Select these parts:

- USB flash drive formatted for Windows FAT16 or FAT32

### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project USB Flash Drive demo.lvproj contained in the subfolder USB Flash Drive demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI illustrates a simple datalogging application. Insert the USB flash drive into the USB connector on the top of NI myRIO as shown in Figure 36.1 on the previous page, run the VI, and then move the myRIO to cause some activity on the X-axis acceleration indicator. The VI continually monitors the activity of the NI myRIO onboard accelerometer and then saves these measurements to a spreadsheet file in CSV format (comma-separated values) when you stop the VI; click the **Stop** button or press the escape key. Use the default base filename “logfile” or enter a new name. Note that the VI overwrites the file each time that you run the VI. Remove the USB flash drive and read the spreadsheet file with a spreadsheet application such as Excel. The first column is time and the second column is acceleration; plot these two columns as an X-Y graph to confirm that the plot matches that of the **Acquired data** waveform graph.

The demo VI also displays the USB flash drive capacity and free space as well as the files and folder on the top-level folder of the flash drive. Run the VI repeatedly with different logfile names and you should see these newly-created files appear in the far-left indicator.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black **Run** button on the toolbar signifying that the VI is in run mode, and
- USB flash drive inserted into the USB connector — The demo VI will generate an error message

if the USB flash drive is missing.

## 36.2 Interface Theory

**NI myRIO file system:** The NI myRIO RT (real-time) target runs the NI Linux Real-Time OS (operating system) which manages the onboard solid-state hard disk drive (HDD) with 387 MB capacity as well as the USB flash drive. Inserting a USB flash drive automatically mounts the drive as the folder /u. Most of the folders on the HDD are read-only, however, three folders provide read-write access: /home/lvuser, /home/webserver, and /tmp (this folder is automatically cleared after a reset). Access to the file system includes web browser, mapped network drive, and interactive command-line prompts with secure shell (SSH). A LabVIEW program can also directly read and write files as well as perform a wide variety of file management tasks.

Study the video *NI myRIO File System* ([youtu.be/BuREWnD6Eno](https://youtu.be/BuREWnD6Eno), 12:55) to learn about the NI myRIO file system, including accessing and downloading files from the onboard hard disk drive (HDD) with a web browser, mapping the onboard HDD as a network drive in Windows, programmatically creating text files and spreadsheet files with LabVIEW, and interactively navigating the NI myRIO file system through a secure shell (SSH) connection with the PuTTY SSH client.

## 36.3 Basic Modifications

Study the video *USB Flash Drive Demo Walk-Through* ([youtu.be/YlQukBt1lWI](https://youtu.be/YlQukBt1lWI), 8:22) to learn the design principles of USB Flash Drive demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Enable the accelerometer Y-axis and X-axis outputs and log these to the CSV spreadsheet file as two additional columns. Read the spreadsheet

file on a PC and confirm that the plotted data looks correct.

2. Add a feature to create a text file that contains the time and date at which the logfile was created. Use the **Get Time/Date VIs** in the **Programming | Timing** subpalette to create the character string.

### 36.4 Integrated Project Ideas

Now that you know how to use the USB flash drive consider integrating it with other devices to create a complete system, for example:

- *Data Logger* (42)
- *Security Camera* (53)
- *EEPROM Programmer* (59)

### 36.5 For More Information

- *Using WebDAV to Transfer Files to Your Real-Time Target* by National Instruments ~ Explains how to map the myRIO file system to a Windows drive:

<http://digital.ni.com/public.nsf/allkb/4EBE45E8A816B19386257B6C0071D025>

- *Working with File Paths on Real-Time Targets* by National Instruments ~ Best practices for specifying folder and file names (file paths):

<http://www.ni.com/tutorial/14669/en/>

- *What File Systems Can be Used With the NI Linux Real-Time OS?* by National Instruments ~ Recommended formatting of USB flash drives:

<http://digital.ni.com/public.nsf/allkb/4819E4118F382D4586257C1A00592CF3>



# 37 Webcam

NI myRIO includes support for USB webcams (Figure 37.1), and can work with multiple webcams attached via a USB hub. With the Vision Acquisition and Development modules installed you can acquire and process still images and video streams with a wide variety of image processing and machine vision tools.



Figure 37.1: webcam connected to NI myRIO USB port.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Acquire and process a single image,
2. Acquire and process a video stream,
3. Set the webcam's video mode (resolution and frame rate),

4. Set a webcam attribute such as saturation, and
5. Use NI-MAX to determine available video modes and attributes for your webcam.

## 37.1 Component Demonstration

Follow these steps to demonstrate correct operation of the webcam.

### Select these parts:

- USB webcam

### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project Webcam demo.lvproj contained in the subfolder Webcam demo,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open Main.vi by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

**NOTE:** You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** This project contains three distinct `Main.vi` top-level applications:

1. `Main - single image.vi` – This demo VI captures a single image and detects the edges in the image. Select your webcam with the `webcam` front-panel control and then run the VI. You should see the webcam image on the left side and its edge-detected version on the right side. Adjust the high threshold `HThresh` control to change the sensitivity of the edge detector.
2. `Main - video stream.vi` – This demo VI captures a video stream and processes the stream in real time. Select your webcam with the `webcam` front-panel control and then run the VI. You should see the webcam image on the left-hand side and its processed version on the right-hand side; the VI begins with no processing (pass-through mode). Select the type of video processing with the `processing` front-panel control. Adjust the high threshold `HThresh` control to change the sensitivity of the edge detector. Try the “hue” processing mode to display a gray level image where each gray level represents a different color; can you relate the color in the image to its gray tone? Also try the “luma” processing mode to display the gray-level equivalent of the source image. Click `adjust saturation` to enable adjustment of the webcam “Saturation” property and then move the horizontal pointer control. What level seems to provide the most pleasing color balance?
3. `Main - camera info.vi` – This demo VI displays information for all cameras that have ever been connected to your NI myRIO as well as available attributes and video modes for the selected webcam. Select a specific webcam with the `webcam` control and then run the VI to up-

date the display (you need to run the VI again each time you select another webcam). Click the up/down arrows on the array index control of all `cameras` to see the details on each camera. Identify one of the video modes (resolution and frame rate) besides the mode at the top of the list, make note of its number, and then enter this value into `Main - video stream.vi` (you need to re-run the VI to change the video mode). Confirm that the resolution matches that reported in the camera information display underneath the source image display on the left-hand side.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Webcam inserted into the USB connector.

## 37.2 Interface Theory

**NI myRIO file system:** The LabVIEW Vision and Motion VIs provide a comprehensive set of tools to acquire and process images as well as to extract useful numerical information from images. Connect a single webcam to the NI myRIO USB port or multiple webcams with a USB hub and then use the Vision and Motion VIs to quickly create an application.

Study the video *Webcam Interfacing Theory (Single Image)* ([youtu.be/lizzs9rBmYA](https://youtu.be/lizzs9rBmYA), 7:50) to learn the basics of acquiring and processing a single image, and then continue with *Webcam Interfacing Theory (Video Stream)* ([youtu.be/L7tMeKshd38](https://youtu.be/L7tMeKshd38), 10:59) to learn how to process a video stream in real time. Study *Webcam Interfacing Theory (Set Video Mode)* ([youtu.be//IxsioDcCuwA](https://youtu.be//IxsioDcCuwA), 5:13) to learn how to adjust the video mode (resolution and frame rate) and then study *Webcam Interfacing Theory (Set Attribute)* ([youtu.be/wcM6xfXOT6I](https://youtu.be/wcM6xfXOT6I), 4:49) to



learn how to adjust webcam attributes such as color saturation, brightness, and exposure.

### 37.3 Basic Modifications

Study three videos *Webcam Demo Walk-Through (Single Image)* ([youtu.be/Fup-ro7qWxk](https://youtu.be/Fup-ro7qWxk), 4:53) *Webcam Demo Walk-Through (Video Stream)* ([youtu.be/IbTN6pBu\\_EM](https://youtu.be/IbTN6pBu_EM), 8:19) and *Webcam Demo Walk-Through (Camera Information)* ([youtu.be/Pta6\\_REo41c](https://youtu.be/Pta6_REo41c), 3:17) to learn the design principles of Webcam demo, and then try making these modifications to the block diagram of `Main - video stream.vi`:

1. Add additional processing functions by extending the enumerated front-panel control and then adding new subdiagrams to the case structure. The Vision and Motion VIs contain a rich set of processing functions for you to try.

### 37.4 Integrated Project Ideas

Now that you know how to use the webcam consider integrating it with other devices to create a complete system, for example:

- *Wireless Sensor* (41)
- *Data Logger* (42)
- *Steer By Wire* (44)
- *QR Code Scanner* (47)
- *Scanning Sensor* (51)
- *Security Camera* (53)

### 37.5 For More Information

- *Machine Vision Concepts* by National Instruments ~ Click the “Table of Contents” link to learn more about vision basics, image processing and analysis, and machine vision:

[http://zone.ni.com/reference/en-XX/help/372916P-01/nivisionconcepts/machine\\_vision/](http://zone.ni.com/reference/en-XX/help/372916P-01/nivisionconcepts/machine_vision/)



# 38 GPS Receiver

All modern navigation relies on the Global Positioning System (GPS) satellite network to provide the 3-D coordinates (latitude, longitude, and altitude) of a GPS receiver anywhere on the planet with high precision and accuracy. While the entirety of the global positioning system represents a sophisticated and complex system, the PmodGPS (Figure 38.1) based on the GlobalTech Gms-u1LP GPS module provides a surprisingly simple way to access a wealth of information in addition to coordinates including precise time, date, heading, speed, and other useful information related to the specific satellites in view.

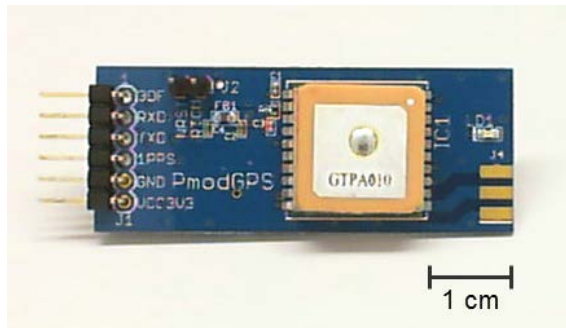


Figure 38.1: Digilent PmodGPS receiver based on the GlobalTech Gms-u1LP.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Discuss GPS concepts including trilateration to

- measure 3-D coordinates,
2. Capture the NMEA sentences (ASCII text strings) generated by the Gms-u1LP as an array of strings in LabVIEW,
  3. Parse the sentence to extract individual data fields, and
  4. Extract information from the data fields using LabVIEW Scan From String and Format Into String VIs.

## 38.1 Component Demonstration

Follow these steps to demonstrate correct operation of the GPS receiver.

**Select these parts:**

- GPS receiver (PmodGPS), [http://digilentinc.com/Data/Products/PMOD-GPS/PmodGPS\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-GPS/PmodGPS_rm.pdf)
- Jumper wires, F-F (5×)

**Build the interface circuit:** Refer to the schematic diagram shown in Figure 38.2 on the following page; the GPS receiver requires five connections to NI myRIO MXP Connector A (see Figure A.1 on page 233):

1. +3.3-volt supply (VCC3V3) → A/+3.3V (pin 33)
2. Ground (GND) → A/GND (pin 30)
3. UART transmit data (TXD) → A/UART.RX (pin 10)
4. UART receive data (RXD) → A/UART.TX (pin 14)
5. One-pulse-per-second (1PPS) → A/DIO0 (pin 11)

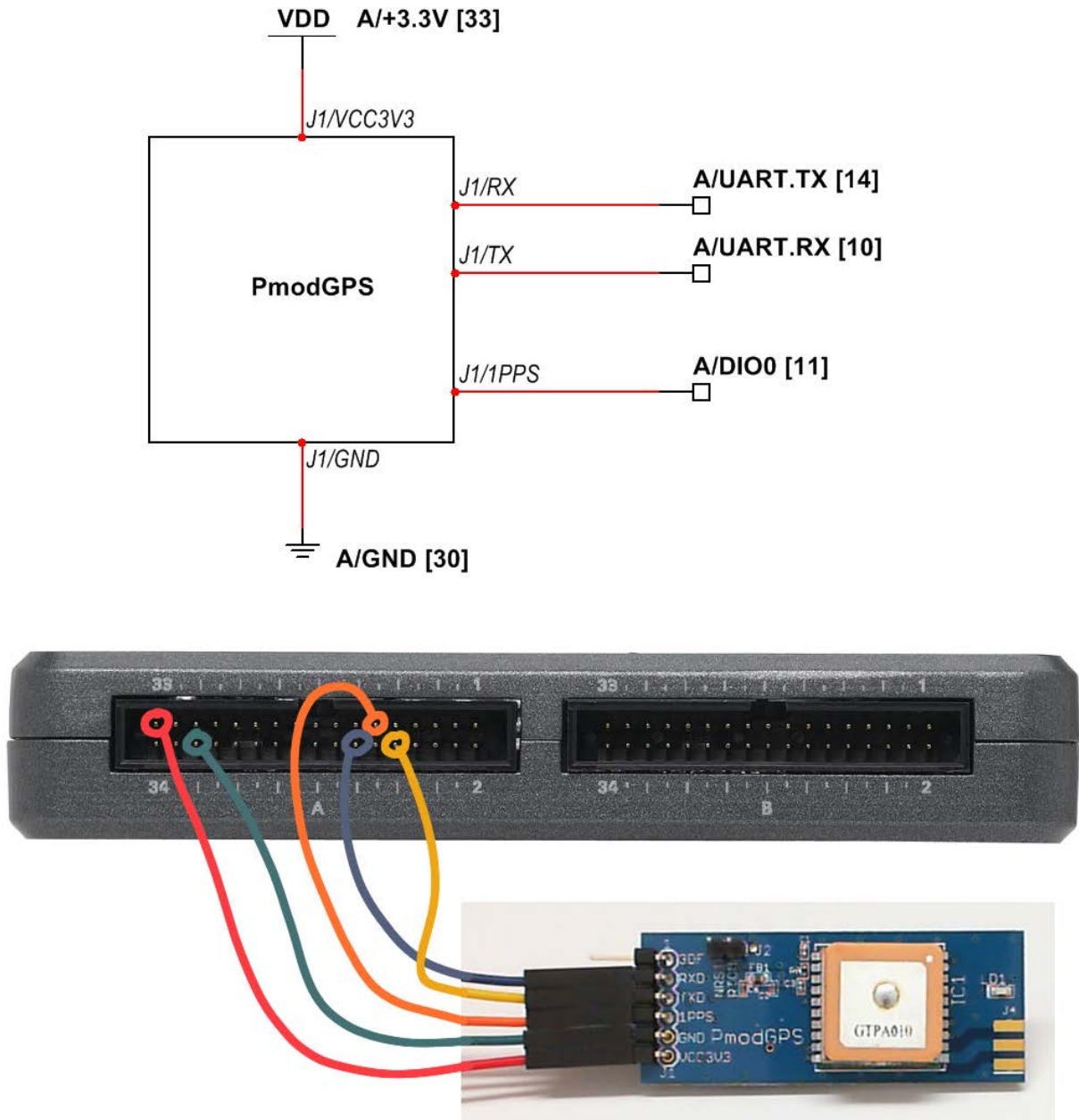


Figure 38.2: Demonstration setup for GPS receiver connected to NI myRIO MXP Connector A.

**Run the demonstration VI:**

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done so previously and unpack the contents to a convenient location,
- Open the project `GPS Receiver demo.lvproj` contained in the subfolder `GPS Receiver demo`,
- Expand the hierarchy button (a plus sign) for the `myRIO` item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the `Run` button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI includes three main front-panel indicator zones:

1. NMEA sentences on the bottom left displays the sentences (ASCII text strings) as received from the Gms-u1LP GPS module; the most-recent sentence appears at the bottom of the indicator,
2. Time, Latitude [deg], and Longitude [deg] at the top of the front panel show examples of formatted data extracted from the NMEA sentences, and
3. NMEA data fields (right side) from the “\$GPRMC” (Recommended Minimum Navigation Information) sentence.

Run the demo VI and you should see a new cluster of NMEA sentences appear once every second. On a “cold start” the Gms-u1LP GPS receiver requires about one minute to begin generating valid information; at this time the “1PPS” indicator on both the front panel and on the NI myRIO LED0 will begin

to flash. Note that overcast conditions and building obstructions may increase the time for the receiver to gain a fix on a sufficient number of satellites.

If you have a battery pack for your NI myRIO and a mobile computer you may want to observe the display as you move to other locations. How well does the `Speed over ground` value match your speed? Can you see the latitude and longitude indicators change value?

Find your time zone offset and enter this value into the time zone control. For example, Eastern Time Zone in the United States is `-5` indicating five hours west of Coordinated Universal Time (UTC) centered on the Prime Meridian intersecting Greenwich, England. Compare the value in the Time indicator (upper left corner) to the official time maintained by the National Institute of Standards and Technology at <http://nist.time.gov>. Do you observe any difference?

Click the `Stop` button or press the escape key to stop the VI and to reset NI myRIO.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black `Run` button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A and that you have the correct pin connections,
- Correct GPS receiver terminals — double-check your connections, and ensure that you have connected the NI myRIO UART “receive” input to the GPS receiver “transmit” output; also check that you have not accidentally crossed the power supply connections.

## 38.2 Interface Theory

**Interface circuit:** The GlobalTech Gms-u1LP is a self-contained GPS module with integrated antenna.

The GPS module generates “NMEA sentences,” i.e., ASCII text strings terminated by a carriage return and line feed combination. NMEA (National Marine Electronics Association) sentences accommodate a wide range of navigation information of which the GPS sentences form a rather small subset. Each sentence begins with “\$” followed a two-character “talker ID” (“GP” for a GPS sentence) and a three-character sentence type. The sentence body contains comma-delimited fields and ends with a check-sum. By default the Gms-u1LP emits ASCII characters at 9600 baud (8-bit, 1-stop) and generates a cluster of four sentences (and occasionally more) every second.

Study the video *GPS Receiver Interfacing Theory* ([youtu.be/ptp2bGU9EHU](http://youtu.be/ptp2bGU9EHU), 12:56) to learn more about general GPS concepts including a 2-D trilateration example that illustrates how an unknown position is determined from satellite signals, the Digilent GPS board and GlobalTop Gms-u1LP GPS module features, the five NMEA sentences generated by the Gms-u1LP module, and a detailed example of parsing the “\$GPRMC” sentence to obtain latitude, longitude, time, date, and other information.

### 38.3 Basic Modifications

Study the video *GPS Receiver Demo Walk-Through* ([youtu.be/SebcpkbYBd4](http://youtu.be/SebcpkbYBd4), 9:46) to learn the design principles of GPS Receiver demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Add an altitude front-panel indicator; refer to the “\$GPGGA” sentence.
2. Add a front-panel indicator to show the number of satellites in view; refer to the “\$GPGGA” or “\$GPGSV” sentence.
3. Add a display to show the distance from a reference latitude-longitude coordinate. Refer to the article “Great Circle Distance” referenced below to learn how to calculate the distance along the surface of a sphere between two points.

### 38.4 Integrated Project Ideas

Now that you know how to use the GPS receiver consider integrating it with other devices to create a complete system, for example:

- *Wireless Sensor* (41)
- *Data Logger* (42)

### 38.5 For More Information

- *PmodGPS Reference Manual* by Digilent ~ Reference manual for the GPS receiver:  
[http://digilentinc.com/Data/Products/PMOD-GPS/PmodGPS\\_rm.pdf](http://digilentinc.com/Data/Products/PMOD-GPS/PmodGPS_rm.pdf)
- *PmodGPS Schematics* by Digilent ~ Schematic diagram of the GPS receiver:  
[http://digilentinc.com/Data/Products/PMOD-GPS/PmodGPS\\_sch.pdf](http://digilentinc.com/Data/Products/PMOD-GPS/PmodGPS_sch.pdf)
- *Gms-u1LP Datasheet* by GlobalTop ~ Datasheet for Gms-u1LP GPS module based on the MediaTek GPS MT3329 chip set; this document contains detailed information on the NMEA sentences produced by the module. Try a general web search on the phrase “Gms-u1LP PDF” if your search at GlobalTop is unsuccessful:  
<http://www.gtop-tech.com>
- *Untangling the GPS Data String* by Institute of Agriculture and Natural Resources ~ A good article detailing the NMEA sentences related to GPS:  
<http://www.ianrpubs.unl.edu/eublic/live/ec157/build/ec157.pdf>
- *Great Circle Distance* by Wolfram ~ The Great Circle distance between any two points on a sphere may be used to approximate the distance between any two GPS latitude-longitude coordinates; this relatively simple technique ignores the fact that the Earth is an oblate spheroid (slightly flattened at the poles):  
<http://mathworld.wolfram.com/GreatCircle.html>

# 39 RFID Reader

RFID (radio-frequency identification) offers a non-contact scanning method for access control and inventory management. An RFID reader queries an RFID tag to determine the unique bit pattern encoded in the tag. Passive RFID tags power their on-board electronics from the magnetic field generated by the reader. Figure 39.1 shows the ID-Innovations ID-12LA RFID reader along with the breakout board that converts the 2 mm pin spacing to standard 0.1 inch pin spacing for breadboards.

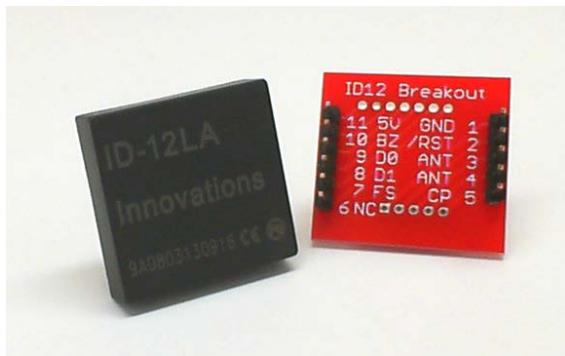


Figure 39.1: ID-Innovations ID-12LA RFID reader with breakout board.

**Learning Objectives:** After completing the activities in this chapter you will be able to:

1. Describe the EM4001 RFID tag standard,
2. Interpret the ID-12LA UART output format, and

3. Implement the checksum calculation.

## 39.1 Component Demonstration

Follow these steps to demonstrate correct operation of the RFID reader.

**Select these parts:**

- ID-Innovations ID-12LA RFID reader, <http://www.hobbytronics.co.uk/datasheets/sensors/ID-12LA-ID-20LA.pdf>
- Breakout board for ID-12LA RFID reader, <http://www.sparkfun.com/products/8423>
- Breakaway header, straight pin, <https://www.sparkfun.com/products/116>
- RFID tag, 125 kHz, EM4001 format (2×), <https://www.sparkfun.com/products/8310>
- LED [needs info]
- Jumper wires, M-F (4×)

You will also need access to a soldering station.

**Build the interface circuit:** The RFID reader breakout board requires soldering. Refer to Figure 39.2 on the next page to see how the reader and breakout board should look when finished. Begin by breaking off a 5-pin section and a 6-pin section of the breakaway header. Place the headers in a breadboard (longer pin down) to keep them properly aligned, and then fit the breakaway board over the headers *with the pin numbers facing down* and the SparkFun logo

facing up. Solder the short pins in place. When cool, remove the breakout board and then fit the RFID reader onto the breakout board. Solder the RFID reader pins on the other side of the breakout board.



Figure 39.2: RFID reader, breakout board, and headers after soldering.

Refer to the schematic diagram and recommended breadboard layout shown in Figure 39.3 on the facing page. Note that the compact layout requires several breadboard jumper connections *under* the RFID reader. The RFID reader requires four connections to NI myRIO MXP Connector B (see Figure A.1 on page 233):

1. +3.3-volt supply → B/+3.3V (pin 33)
2. Ground → B/GND (pin 30)
3. UART output D0 → B/UART.RX (pin 10)
4. Tag-in-range → B/DIO0 (pin 11)

#### Run the demonstration VI:

- Download <http://www.ni.com/academic/myrio/project-guide-vis.zip> if you have not done

so previously and unpack the contents to a convenient location,

- Open the project `RFID demo.lvproj` contained in the subfolder `RFID demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing `Ctrl+R`.

Expect to see a “Deployment Process” window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running.

NOTE: You may wish to select the “Close on successful completion” option to make the VI start automatically.

**Expected results:** The demo VI includes three main indicators:

1. tag detected lights anytime a valid RFID tag is within range of the RFID reader,
2. RFID tag bytes displays the sixteen bytes transmitted by the RFID reader corresponding to a valid RFID tag, and
3. data string extracts the data section of the message and displays it as a hexadecimal numerical value.

Hold one RFID tag card (Figure 39.4 on page 186) near the RFID reader and you should see the tag detected indicator light when you hold the card sufficiently nearby. The data string indicator should show a five-digit hexadecimal number (contains digits 0 to 9 and A to F), and the RFID tag bytes indicator should a sequence beginning with 0x02 (ASCII “start of text” character) and end with 0x03 (ASCII “end of text” character).

Try the other RFID tag card and confirm that you see a different value for the data string. Also, experiment with the minimum required distance for the



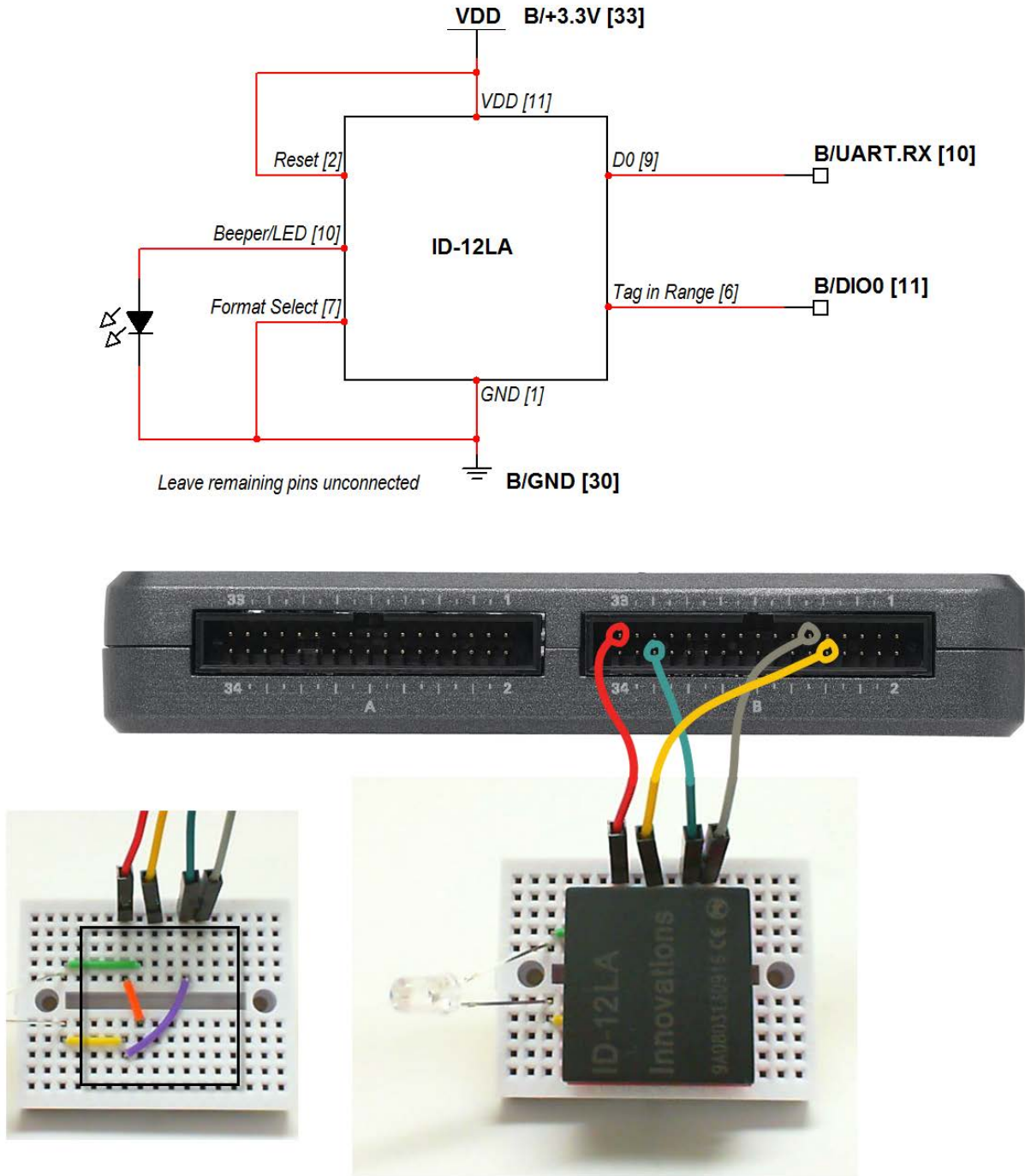


Figure 39.3: Demonstration setup for RFID reader connected to NI myRIO MXP Connector B. Note that the compact layout requires several breadboard jumper connections *under* the RFID reader.

RFID reader to scan the tag. Does waving the card or moving the card quickly make any difference?

Click the **Stop** button or press the escape key to stop the VI and to reset NI myRIO.



Figure 39.4: RFID card transponders.

**Troubleshooting tips:** Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct RFID reader terminals — double-check your connections, and ensure that you have connected the NI myRIO UART “receive” input to the RFID reader “D0” output; also check that you have not accidentally crossed the power supply connections, and
- Hidden jumpers — be certain that you have added the hidden jumper wires as shown in Figure 39.3 on the previous page.

## 39.2 Interface Theory

**Interface circuit:** The RFID reader uses an electromagnetic field to supply power to the RFID tag *and* to receive a transmitted digital signal from the tag. This noncontact method is popular for ID badges and inventory control. The ID-Innovations ID-12LA RFID reader reads RFID tags encoded with the EM4001 standard and formats the tag information into one of three formats, one of which is compatible with the NI myRIO UART serial communications port.

Study the video *RFID Reader Interfacing Theory* ([youtu.be/z1v0vCue83c](https://youtu.be/z1v0vCue83c), 11:29) to learn more about the RFID reader principles of operation, including the EM4001 tag standard, ID-12LA RFID reader configuration and UART data output format, and checksum calculation.

## 39.3 Basic Modifications

Study the video *RFID Demo Walk-Through* ([youtu.be/Jovn0kPJOKs](https://youtu.be/Jovn0kPJOKs), 5:18) to learn the design principles of RFID demo, and then try making these modifications to the block diagram of `Main.vi`:

1. Calculate the checksum for the data segment and compare to the checksum field of the RFID tag message; use a Boolean indicator to show whether or not a valid message was received from the RFID reader.

## 39.4 Integrated Project Ideas

Now that you know how to use the RFID reader consider integrating it with other devices to create a complete system, for example:

- *Hotel Room Safe Controller* (49)

### 39.5 For More Information

- *EM4001 Protocol Description* by Priority 1 Design  
~ Detailed tutorial on the EM4001 protocol used by RFID tag cards:

[http://www.priority1design.com.au/em4100\\_protocol.html](http://www.priority1design.com.au/em4100_protocol.html)

- *RFID Made Easy (AppNote 411)* by EM Micro-electronic ~ Everything you need to know about RFID: system principles, electromagnetic theory, antenna design, and data coding techniques:

<http://www.emmicroelectronic.com/webfiles/Product/RFID/AN/AN411.pdf>



## **Part V**

# **Integrated Project Ideas**



# 40 Handheld Meter

Handheld meters such as a digital multimeter (DMM) provide a convenient way to make measurements of voltage, current, and resistance. You can extend the idea of a handheld meter to measure a wide range of other physical parameters, too. Combine the LCD display with one or more sensors to create your own handheld meter:

- Measurement display: *LCD Display* (26, 27, 28)
- Light sensors: *Photocell* (9), *Ambient Light Sensor* (24)
- Temperature sensors: *Thermistor* (8), *Temperature Sensor* (33)
- Sound level sensors (measure the RMS of the audio waveform): *Electret Microphone* (10), *MEMS Microphone* (34)
- Range (distance) sensors: *IR Range Finder* (19), *Sonic Range Finder* (20)
- Magnetic field sensors: *Hall-Effect Sensor* (15), *Compass* (23)

Refer to the example LabVIEW application *Digital Tape Measure* that combines the sonic range finder with the LCD display.





# 41 Wireless Sensor

Wireless sensors provide useful information at a distance from the data acquisition system, often in areas that do not have power readily available or are otherwise inaccessible. Attach a battery pack to your myRIO and then combine the Bluetooth module with one or more sensors; you may also wish to include a local display to confirm that your wireless sensor is properly transmitting information:

- Wireless data link: *Bluetooth Module* (31)
- Local display: *LCD Display* (26, 27, 28)
- Light and vision sensors: *Photocell* (9), *Ambient Light Sensor* (24), *Webcam* (37)
- Temperature sensors: *Thermistor* (8), *Temperature Sensor* (33)
- Sound level sensors (measure the RMS of the audio waveform): *Electret Microphone* (10), *MEMS Microphone* (34)
- Range (distance) and position sensors: *IR Range Finder* (19), *Sonic Range Finder* (20), *GPS Receiver* (38)
- Magnetic field sensors: *Hall-Effect Sensor* (15), *Compass* (23)
- Motion and vibration sensors: *Piezoelectric-Effect Sensor* (16), onboard accelerometer, *Accelerometer* (21), *Gyroscope* (22)



# 42 Data Logger

A *data logger* monitors one or more sensors over an extended time period and saves the acquired data to a file for later review and analysis. A data logger takes measurements at a regular interval and immediately appends the time-stamped record to a file to ensure that data records are not lost in case of a power failure during the data logging session.

The sample project `Data Logger - Light+Temperature` uses the LabVIEW data logging and time-stamping VIs to record measurements from the temperature sensor and ambient light sensor to a USB flash drive. Use your desktop computer to create a configuration file on a USB flash drive called `config.txt` that contains a string for the base name of the data log file, an integer to indicate the number of milliseconds to wait between measurements, and an integer to indicate the total number of measurements for the session. For example, the line `temp+light 10000 360` will cause the program to create a data logging file (`.log` extension) with a time stamp in the file name to show when the data logging session started, to collect measurements every 10 seconds, and to collect 360 measurements (one hour's worth of data). Insert the USB flash drive in the NI myRIO USB port and then start the application. The onboard LED0 will flash each time the sensors are polled and a data record is appended to the log file. Press the NI myRIO onboard button to quit the data logging session early. An error condition such as absence of the flash drive or problems communicating with

the temperature sensor will cause all four onboard LEDs to light. Once you have created a data log file, reconnect the USB flash drive and then run the `Read log file` desktop application included in the project to plot the collected data. It is important to note that LabVIEW data logging VIs allow you to read the measurements of all of your sensors exactly as they were recorded, i.e., the data types are exactly the same. This way you do not need to make decisions about how to format numerical values to text files and possibly lose available resolution.

You can easily adjust the system time of your NI myRIO. With myRIO attached via USB, point your web browser to `172.22.11.2` and select the "Time Configuration" tab.

**NOTE:** *NI myRIO does not include battery back-up for the system clock, therefore the system clock suspends operation when power is removed.*

Study the data logger application VI and then adapt it to your own needs with other available sensors; consider using an LCD display to serve as a local display of the current measurement:

- Data log file storage: *USB Flash Drive* (36)
- Local display: *LCD Display* (26, 27, 28)
- Light and vision sensors: *Photocell* (9), *Ambient Light Sensor* (24), *Webcam* (37)
- Temperature sensors: *Thermistor* (8), *Temperature Sensor* (33)

- Sound level sensors (measure the RMS of the audio waveform): *Electret Microphone* (10), *MEMS Microphone* (34)
- Range (distance) and position sensors: *IR Range Finder* (19), *Sonic Range Finder* (20), *GPS Receiver* (38)
- Magnetic field sensors: *Hall-Effect Sensor* (15), *Compass* (23)
- Motion and vibration sensors: *Piezoelectric-Effect Sensor* (16), onboard accelerometer, *Accelerometer* (21), *Gyroscope* (22)

# 43 NTP Clock

Network Time Protocol (NTP) servers provide precise time and date information to any Internet-connected device. See <http://nist.time.gov> for a web-style interface. The LabVIEW project `NTP Clock` illustrates how to use the LabVIEW TCP/IP VIs to query an NTP time server for the current date and time. For example, opening a TCP/IP connection to `nist.time.gov` returns the string `56742 14-03-26 22:10:29 50 0 0 172.1 UTC(NIST) *` which can then be parsed to obtain the date (March 26, 2014) and the time (22:10:29, Coordinated Universal Time).

**NOTE:** *Enable Wi-Fi networking on your NI myRIO to allow NTP Clock to work properly. With myRIO connected via USB browse to 172.22.11.2, select the "Network Configuration" tab, select "Connect to wireless network," and then complete the remaining entries in the form. Your myRIO is ready to connect to the NTP server when you see a valid entry for "IPv4 Address."*

You can extend this project in a number of ways. Add the LCD display to show the date and time, purchase additional seven-segment display modules to make a traditional digital clock, use the LED matrix to create a binary clock display (each pair of rows displays a numerical value as a binary pattern), or devise a creative arrangement of discrete LEDs to indicate the time of day. You could also add an actuator or output device to "make something

happen" at specific times. As examples, use the speaker to chime at fifteen minute intervals and on the hour like London's "Big Ben," use the relay as a "clicker" instead of a chime, or move the servo motor horn as a pseudo "minute hand" (the horn rotates over an angle of 180 degrees during 60 seconds and then rapidly resets):

- Time and date display: *LCD Display* (26, 27, 28)
- LED-based displays: *Discrete LED* (2), *Seven-Segment LED Display* (3), *LED Matrix* (29)
- Actuators and output devices: *Relay* (6), *Buzzer/Speaker* (11), *Servo* (17)



# 44 Steer By Wire

The phrase “fly by wire” applies to aircraft systems in which sensors, control electronics, and servo motors replace the mechanical linkages between the pilot’s yoke and rudder pedals and the control surfaces including flaps, ailerons, and rudder. Engineers at Waterloo Labs applied this idea to create a “steer by wire” (actually wireless) system using an iPhone as the controller and a Compact RIO to control the steering column, accelerator and brakes; see the fascinating video demo at [youtu.be/\\_x5IziyOcAg](http://youtu.be/_x5IziyOcAg). Combine the servo motor with a rotary user input device to create your own steer-by-wire system with the servo motor controlling the position of steering wheels on a robotic platform. Use the accelerometer or gyroscope as a hand-held controller like the Waterloo Labs system; see the LabVIEW application *Steer-By-Wire* as an example of controlling the servo position using the X-axis of the NI myRIO onboard accelerometer. You can also use a sensor to make the steering system automatically respond to physical measurements such as light and temperature:

- Steering wheel actuator: *Servo* (17)
- Steering wheel control inputs: *Potentiometer* (7), *Rotary Encoder* (13)
- Light and vision sensors: *Photocell* (9), *Ambient Light Sensor* (24), *Webcam* (37)
- Temperature sensors: *Thermistor* (8), *Temperature Sensor* (33)
- Sound level sensors (measure the RMS of the audio waveform): *Electret Microphone* (10),

*MEMS Microphone* (34)

- Range sensors: *IR Range Finder* (19), *Sonic Range Finder* (20)
- Magnetic field sensors: *Hall-Effect Sensor* (15), *Compass* (23)
- Motion sensors: onboard accelerometer, *Accelerometer* (21), *Gyroscope* (22)





# 45 Digital Thermometer

Create a digital thermometer to measure the ambient temperature using either the thermistor or the I<sup>2</sup>C-bus temperature sensor and display the temperature on the LCD. Include features to display in Fahrenheit or celsius (use the NI myRIO onboard pushbutton to select between these options) and also display the minimum and maximum observed temperatures:

- Time and date display: *LCD Display* (26, 27, 28)
- Temperature sensors: *Thermistor* (8), *Temperature Sensor* (33)



# 46 3-D Color Controller

The gyroscope and accelerometer both provide three-axis measurements related to orientation of the sensor. The RGB LED mixes the primary light colors to produce any arbitrary color. Use one of these 3-D sensors to control the RGB LED color by mapping the X-axis to red, Y-axis to green, and Z-axis to blue. Use three PWM outputs to provide intensity control of each LED. Provide the option to calibrate the sensor values so that a given value on each axis produces the same intensity on each LED. Also investigate nonlinear mapping functions that will make the LED intensity appear to vary proportionally with the sensor variable:

- RGB LED: *Discrete LED* (2)
- 3-D sensors: Onboard accelerometer, *Accelerometer* (21), *Gyroscope* (22)



# 47 QR Code Scanner

QR codes are a popular type of 2-D bar code that encodes a website's URL. The LabVIEW Vision library includes the VI IMAQ Read QR Code to extract the URL as a string. A quick Internet search reveals many free QR code generating services, i.e., [www.qrstuff.com](http://www.qrstuff.com). Because you can enter arbitrary text, the generated QR code pattern is not restricted to URLs but may be used to represent any type of information that you wish. For example, you could create unique QR codes for an inventory control system in which each object or container has a QR code label that can be conveniently scanned by a webcam. As another example, each robot in a road race could be tagged with a QR code and the scanner could automatically identify each robot by name as it passes by a checkpoint.

Combine the webcam and the LCD display to create your own QR code scanner. The example LabVIEW project `QR Code Scanner` gets you started:

- Local display: *LCD Display* (26, 27, 28)
- *Webcam* (37)



# 48 RPN Calculator

Combine the keypad and LCD display to create your own version of the familiar pocket calculator. The LabVIEW project `RPN Calculator` gets you started with the basic framework of an RPN (reverse Polish notation) stack-based calculator. This project illustrates the “Simple State Machine” design pattern — the calculator waits for a button to be pressed on the keypad (“idle” state), moves to one of several calculation states depending on the button that was pressed, and then returns to the idle state to wait for the next key press. An RPN calculator requires you to enter two values and then select the operation. For example, add two values by first pressing the numerical keys for the first value, press “E” to enter the value onto the stack, press the numerical keys for the second value, and then press “A” to add the values. See the block diagram of `Main.vi` for the keypad layout.

The `RPN Calculator` project includes several VIs to serve as a high-level API (application program interface) to the keypad. The LabVIEW demonstration project `Keypad demo` presented in Chapter 25 on page 113 shows how to scan the keypad to determine which buttons are pressed at any given moment. This scanning operation forms the basis of the `scan keypad` VI included with the calculator project; `scan keypad` scans the keypad one time and returns a scan code integer representing the button or buttons that are pressed. The related VIs `open keypad` and `close keypad` simplify the process of opening DIO channels to the

row and column driver lines for the keypad.

The `read keypad` VI further simplifies the interface to the keypad by combining two instances of `scan keypad` inside loops, one to wait for all keys to be released and another to wait for a keypress. The VI blocks further execution until either a keypress is detected or the timeout period has expired (the timeout value can be selected with `open keypad`). With this VI the main application merely waits for `read keypad` to return a valid scan code which in turn can be converted to a numerical value with `scan code to number`; this latter VI also filters out multiple keypresses. In summary, these keypad API VIs greatly simplify the interface to the keypad to speed up development of any application that requires user input via the keypad.

Begin by adding the LCD display interface code to the idle state. You have two display rows available; consider displaying `Stack 0` and `Stack 1` or perhaps a text string to indicate the most-recent operation. Next, add more operations to the calculator (at the moment the calculator only supports addition, change sign, multiplication, squaring, and square root); because of the limited number of keys additional functions need to be added as “function keys.” For example, the square root operation is invoked by first pressing “F” to select a function key and then pressing “2” applies the square root operation. You may also want to change the stack data type from integer to double and then add more code to handle decimal points:

- Local display: *LCD Display* (26, 27, 28)
- *Keypad* (25)



# 49 Hotel Room Safe Controller

Some hotel rooms provide a small safe to protect your valuables. The digital controller for the hotel room safe works like this: place your belongings in the safe, close the door, type in a four-digit code (which you need to remember!) to lock the safe, and then type in the same code to open the safe again. The “Simple State Machine” design pattern illustrated in the LabVIEW project `RPN Calculator` applies quite well to the digital controller for this project. The states include “open” (safe is open and waiting for a four-digit code), “locking” (save four-digit code to memory, move the bolt to the locked position), “locked” (safe is locked and waiting for the same four-digit code), and “opening” (correct four-digit code entered, move the bolt to the open position).

Combine the keypad and LCD display to create your own hotel room safe controller. Use the high-level keypad API VIs included with the `RPN Calculator` project to ensure that your state machine sees only single-button presses, and the timeout feature of `open keypad` and `read keypad` comes in handy to deal with partially-entered codes, i.e., if someone enters three digits, gets distracted and then walks away, the partially-entered code should be cleared. You could use an LED to indicate the bolt position (open or locked) or use the servo motor as a more realistic model of the bolt. Also consider adding a master-key option based on an RFID tag to open the safe in the event of a forgotten four-digit code:

- Local display: *LCD Display* (26, 27, 28)
- *Keypad* (25)
- *Servo* (17)
- *RFID Reader* (39)



# 50 Tachometer

A tachometer measures the speed of rotating machinery in revolutions per minute (RPM). Measure the speed of the DC motor like this: attach a circular cardboard disk (about 3-inch diameter) with a single notch cut into the disk edge to the motor shaft, and then place the disk edge in the light path of the photointerrupter; the photointerrupter detector will generate one pulse per revolution. Take the reciprocal of the measured time between pulses (useful for low RPM) or measure the number of pulses during a given time interval (useful for high RPM). Use the “Step and Direction Signal” mode of the LabVIEW myRIO | Encoder with the “Reset” input wired to “True” to count the total number of photointerrupter pulses since the previous call. No pulses are missed, even at high pulse rates, because the LabVIEW myRIO | Encoder implements its counter in the FPGA fabric. Embed the encoder VI in a timed loop for highest precision and show the measured RPM on the LCD display:

- Local display: *LCD Display* (26, 27, 28)
- *Motor* (12)
- *Photointerrupter* (14)



# 51 Scanning Sensor

Use the servo as a turret-style scanning platform for a 180-degree sensor sweep. Select the largest-available circular servo horn (see Figure 17.3 on page 72) to create a large platform and then attach the desired sensor. Set up a for-loop structure to command the servo angle over a range of angles while collecting sensor measurements to an array. For example, with the sonic rangefinder as the sensor the array of measurements will indicate the distance to objects as a function of angle, thereby providing useful information for a higher-level robot navigation system:

- *Servo* (17)
- *IR Range Finder* (19)
- *Sonic Range Finder* (20)
- *Photocell* (9)
- *Ambient Light Sensor* (24)
- *Webcam* (37)



# 52 On-Off Control System

A room thermostat and air conditioner unit serves as a familiar example of an on-off controller (also known as a “bang-bang” controller). The thermostat continually monitors the room temperature with a sensor and compares the measured temperature to the desired room temperature setting. When the measured room temperature rises above the desired temperature the thermostat closes a relay to activate the air conditioner. Adding cool air to the room causes the measured temperature to fall, and once the measured temperature falls below the desired temperature the relay opens and the air conditioner shuts off. On-off controllers represent an inexpensive form of closed-loop feedback control system and perform satisfactorily for applications that can tolerate a certain degree of up-and-down behavior in the controlled variable.

On-off controllers include a *hysteresis* band to prevent the controller from cycling the control effort on and off too frequently. For example, suppose the thermostat sensor resolution is  $0.1^{\circ}\text{C}$  and the desired temperature is set to  $20.5^{\circ}\text{C}$ . Without hysteresis the thermostat would start the air conditioner at a measured temperature of  $20.6^{\circ}\text{C}$ , turn off a few moments later when the temperature returns to  $20.5^{\circ}\text{C}$ , and then resume cooling shortly after that when the temperature rises again to  $20.6^{\circ}\text{C}$ . With a  $0.2^{\circ}\text{C}$  hysteresis band centered about  $20.5^{\circ}\text{C}$  the measured temperature would need to exceed  $20.6^{\circ}\text{C}$  before the air conditioner turns one and then fall below  $20.4^{\circ}\text{C}$  to cause the air conditioner to turn

off. The width of the hysteresis band defines a trade-off between the amount of acceptable temperature variation and the cycle time of the air conditioner.

Create your own version of an on-off control system. Make or buy a fan to attach to the DC motor to serve as the “air conditioner” and use one of the available temperature sensors to measure the room temperature. Include a front-panel control for the hysteresis band value:

- *Motor* (12)
- *Relay* (6)
- *Thermistor* (8)
- *Temperature Sensor* (33)





# 53 Security Camera

Combine a webcam and a USB flash drive to create a security camera that monitors a room and saves time-stamped images when motion is detected. Consider using a pair of IMAQdx Grab instances to collect adjacent webcam frames, IMAQ Absolute Difference to determine the difference image between the two frames, IMAQ Cast Image to convert to a grayscale image, and IMAQ Quantify 2 to determine the average value of the difference image; a difference higher than a suitable threshold indicates a significant difference between the frames, i.e., motion.

As an alternative, compare the current webcam image to a reference image taken by pressing the onboard button or when the VI starts up. With this approach even a stationary intruder will be detected.

Use IMAQ Write File 2 to save image frames to the USB flash drive. Refer to the project `Data Logger - Light+Temperature` to learn how to access the NI myRIO system time for the time-stamping purpose:

- *Webcam* (37)
- *USB Flash Drive* (36)



# 54 Compass with Tilt Correction

The compass described in Chapter 23 on page 101 senses a 3-D magnetic field, and the tutorial video *Compass Interfacing Theory* ([youtu.be/3WkJ7ssZmEc](https://youtu.be/3WkJ7ssZmEc), 12:47) presents a basic method to calibrate the compass to indicate true north for the purpose of navigation. This basic method requires that the compass be held parallel to the ground, however, this may not always be convenient or even possible. A compass tilt correction procedure exists provided that an accelerometer is available to indicate the angle of the compass with respect to ground.

Study the article *Applications of Magnetic Sensors for Low Cost Compass Systems* linked in the “For More Information” section at the end of Chapter 23 to learn how to mathematically correct for compass tilt and then create your own tilt-compensated compass. Include an option to disengage tilt compensation so that you can make a study of the improved performance:

- Measurement display: *LCD Display* (26, 27, 28)
- *Compass* (23)
- Onboard accelerometer, *Accelerometer* (21)



# 55 Guitar Tuner

A microphone provides your NI myRIO the opportunity to “listen” to you as you tune your guitar. The built-in VI Extract Single Tone Information analyzes the microphone audio signal and returns the frequency of the tone with the highest amplitude, i.e., the fundamental frequency of a single guitar string.

A six-string guitar tuned according to equal temperament has the following string frequencies (see *Perception of Sound* <http://cnx.org/content/m15439> for a description of equal temperament and <http://cnx.org/content/m15440> for an interactive piano keyboard that displays pitch frequencies):

- String 1 (E4):  $440 \times 2^{-5/12} = 329.6$  Hz
- String 2 (B3):  $440 \times 2^{-10/12} = 246.9$  Hz
- String 3 (G3):  $440 \times 2^{-14/12} = 196.0$  Hz
- String 4 (D3):  $440 \times 2^{-19/12} = 146.8$  Hz
- String 5 (A2):  $440 \times 2^{-24/12} = 110.0$  Hz
- String 6 (E2):  $440 \times 2^{-29/12} = 82.41$  Hz

Create a display that at a minimum shows the frequency in Hz of the detected tone. Add more sophistication by detecting and displaying the nearest string name with an indication of the number of cents sharp or flat; one cent is one hundredth of a semitone. For example, suppose the detected frequency is 333.5 Hz. This is frequency is nearest String 1 but is too high (sharp) by 3.9 Hz. At the target frequency  $440 \times 2^{-5/12}$  an error  $e$  in cents causes the frequency to shift to  $440 \times 2^{(-5-e/100)/12}$ . Solving this equation for  $e$  yields an error of +20.2

cents or about 20% of a semitone sharp.

Make the display even more user friendly by displaying a bar graph under the detected frequency that indicates the direction of the tuning error (bar to the right for sharp, bar to the left for flat) with the bar length indicating the amount of error:

- Measurement display: *LCD Display* (26, 27, 28)
- *Electret Microphone* (10)
- *MEMS Microphone* (34)



# 56 Music Maker

Drive the speaker from a tone generator whose frequency and amplitude is controlled by sensors. For example, you could light level detected by a photosensor to control the frequency and the orientation of the accelerometer to control the amplitude:

- *Buzzer/Speaker* (11)
- *Photocell* (9)
- *Ambient Light Sensor* (24)
- *Onboard accelerometer, Accelerometer* (21)





# 57 Digital Bubble Level

Bubble levels are familiar to carpenters and to anyone who has hung picture frames. When level the bubble of a bubble level is centered in its liquid tube, otherwise the bubble drifts one way or the other. Any of the accelerometer axes can be used to determine the degree to which the axis points directly at the ground, i.e., a unit magnitude tells you that the axis is perfectly aligned with a plumb line.

Combine the accelerometer with an array of discrete LEDs, a row of the LED matrix, or the LCD display to emulate the bubble position. Add a numerical display to show the percent of slope:

- Measurement display: *LCD Display* (26, 27, 28)
- *Discrete LED* (2)
- *LED Matrix* (29)
- Onboard accelerometer, *Accelerometer* (21)



# 58 Weather Station

A weather station commonly includes sensors for temperature, humidity, wind speed (anemometer), wind direction, barometric pressure, and rainfall. Form an Internet search based on the sensor name with “I2C,” “SPI,” or “UART” and you will find a wide selection of available sensors to create your own weather station. Create a front-panel display that includes waveform charts to show the weather conditions over the past hour or longer interval of time.

The NI myRIO kits provide a few of the needed sensors to get you started:

- *Thermistor* (8)
- *Temperature Sensor* (33)
- *Photocell* (9)
- *Ambient Light Sensor* (24)



# 59 EEPROM Programmer

Create an EEPROM programmer application that reads a programming file from a USB flash drive, copies the data to the EEPROM, and then reads back the EEPROM data to confirm that it exactly matches the data from the USB flash drive file. Include an LCD display to show the status of the programming process:

- *USB Flash Drive* (36)
- *Serial EEPROM* (30)
- *LCD Display* (26, 27, 28)



**Part VI**

**Appendices**







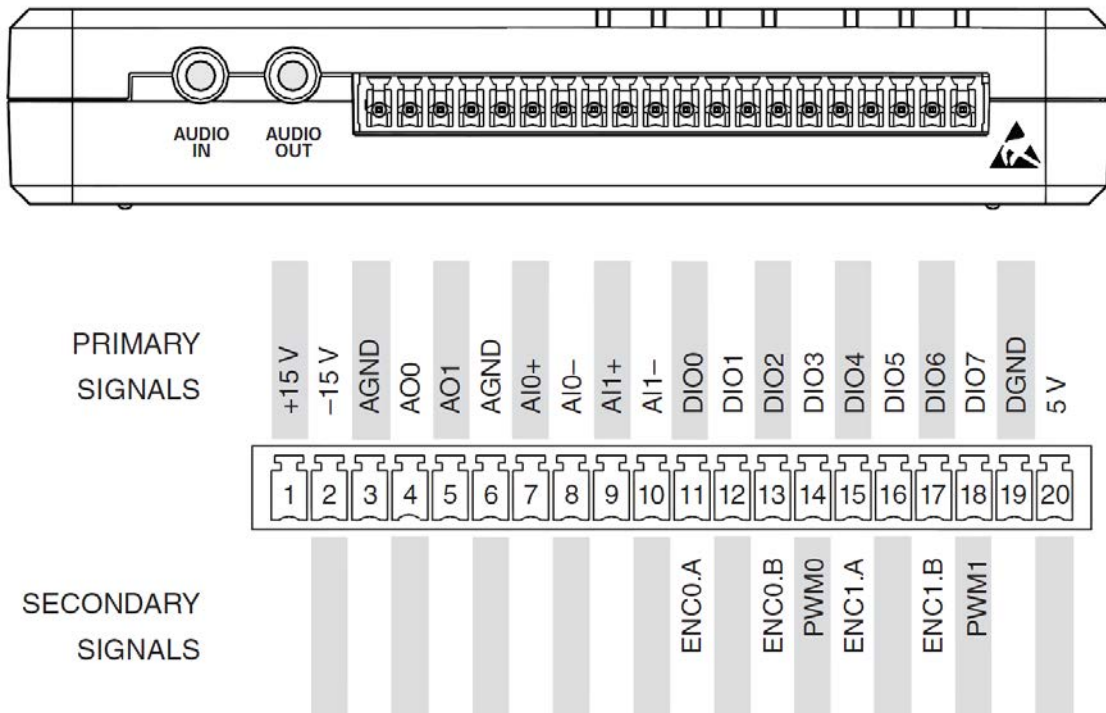


Figure A.2: MSP (miniSystem Port) connector diagram.

# B Build a Stand-Alone Application

During development you normally connect NI myRIO to your computer with a USB cable. After development is complete you can easily deploy your project as a stand-alone application stored on the myRIO solid-state hard drive that starts automatically when you power up the myRIO; no USB cable required. Study the tutorial *Deploy a Stand-Alone Application* ([youtu.be/JXoJECRS-eo](https://youtu.be/JXoJECRS-eo), 8:29) to learn the step-by-step procedure to create a build for the real-time (RT) target, deploy the build as the startup application, and how to disable the startup application, if necessary.



# C NI myRIO Starter Kit Data Sheets

## Switches

- SPDT switch, <http://www.switch.com.tw/product/slide23.html>
- DIP switch, [http://www.resonswitch.com/p\\_rs\\_rsr.htm](http://www.resonswitch.com/p_rs_rsr.htm)
- Rotary DIP switch, <http://www.mantech.co.za/datasheets/products/ERD1-5.pdf>
- Rotary encoder, <http://www.mantech.co.za/Datasheets/Products/F-11E.pdf>
- Pushbutton switch (integrated with rotary encoder), <http://www.sparkfun.com/datasheets/Components/TW-700198.pdf>

## Sensors

- Potentiometer, 10 k $\Omega$ , <http://www.supertech.com.tw/electronic/resistors/potentiometers/PDF/rotary3/23/R0904N.pdf>
- US1881 Hall-effect latch, <http://www.melexis.com/Hall-Effect-Sensor-ICs/Hall-Effect-Latches/US1881-140.aspx>
- AD22151G linear-output magnetic field sensor, <http://www.analog.com/AD22151>
- Piezo film sensor, DT-series, [http://meas-spec.com/product/t\\_product.aspx?id=2478](http://meas-spec.com/product/t_product.aspx?id=2478)
- Photointerrupter, <http://sharpmicroelectronics.com/download/gp1a57hr-epdf>
- Chenyun CY-502 computer microphone, [http://www.chenyun.cc/product\\_en.asp?ProductID=859](http://www.chenyun.cc/product_en.asp?ProductID=859)
- Thermistor, 10 k $\Omega$ , EPCOS B57164K103J, [http://www.epcos.com/inf/50/db/ntc\\_09/LeadedDisks\\_\\_B57164\\_\\_K164.pdf](http://www.epcos.com/inf/50/db/ntc_09/LeadedDisks__B57164__K164.pdf)
- AD22100 temperature sensor, <http://www.analog.com/AD22100>
- Photocell, API PDV-P9203, [http://www.advancedphotonix.com/ap\\_products/pdfs/PDV-P9203.pdf](http://www.advancedphotonix.com/ap_products/pdfs/PDV-P9203.pdf)

## Indicators

- Seven-segment LED display, <http://www.sparkfun.com/datasheets/Components/LED/YSD-160AB3C-8.pdf>

## Actuators

- Buzzer/speaker, Soberton GT-0950RP3, <http://www.soberton.com/product/gt-0950rp3>
- DC motor, [http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e\\_catalog.cgi?CAT\\_ID=ff\\_180psh](http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e_catalog.cgi?CAT_ID=ff_180psh)
- Relay, <http://www.cndongya.com/pdf/relayjzc-11f.pdf>

## Diodes, Transistors, and Active Devices

- 1N3064 small-signal diode, <http://www.fairchildsemi.com/ds/1N/1N3064.pdf>
- 1N4001 general-purpose rectifier, <http://www.vishay.com/docs/88503/1n4001.pdf>
- 2N3904 npn transistor, <http://www.fairchildsemi.com/ds/MM/MMBT3904.pdf>
- 2N3906 pnp transistor, <http://www.fairchildsemi.com/ds/2N/2N3906.pdf>
- AD8541 rail-to-rail single-supply op amp, <http://www.analog.com/AD8541>
- IRF510 n-channel enhancement-mode power MOSFET, <http://www.vishay.com/docs/91015/sihf510.pdf>
- OP37 low-noise precision high-speed op amp, <http://www.analog.com/OP37>
- ZVN2110A n-channel enhancement-mode MOSFET, <http://www.diodes.com/datasheets/ZVN2110A.pdf>
- ZVP2110A p-channel enhancement-mode MOSFET, <http://www.diodes.com/datasheets/ZVP2110A.pdf>

# D Video Tutorial Links

## Component Principles of Operation and Interfacing Techniques

- *Accelerometer Interfacing Theory* ([youtu.be/uj76-JtT\\_xk](https://youtu.be/uj76-JtT_xk), 15:55)
- *Ambient Light Sensor Interfacing Theory* ([youtu.be/zKnn1SskqRQ](https://youtu.be/zKnn1SskqRQ), 7:38)
- *Bluetooth Interfacing Theory* ([youtu.be/WidjSMNU1QM](https://youtu.be/WidjSMNU1QM), 15:34)
- *Compass Interfacing Theory* ([youtu.be/3WkJ7ssZmEc](https://youtu.be/3WkJ7ssZmEc), 12:47)
- *DIP Switch Interfacing Theory* ([youtu.be/KNzEyRwcPIg](https://youtu.be/KNzEyRwcPIg), 7:16)
- *Digital Potentiometer Interfacing Theory* ([youtu.be/C4iBQjWn7OI](https://youtu.be/C4iBQjWn7OI), 9:15)
- *EEPROM Interfacing Theory* ([youtu.be/RxRwyDOCeRw](https://youtu.be/RxRwyDOCeRw), 11:18)
- *Electret Microphone Principles and Preamplifier Circuit* ([youtu.be/izJni0PM0bI](https://youtu.be/izJni0PM0bI), 8:48)
- *Rotary Encoder Interfacing Theory* ([youtu.be/CpwGXZX-5Ug](https://youtu.be/CpwGXZX-5Ug), 10:08)
- *NI myRIO File System* ([youtu.be/BuREWnD6Eno](https://youtu.be/BuREWnD6Eno), 12:55)
- *GPS Receiver Interfacing Theory* ([youtu.be/ptp2bGU9EHU](https://youtu.be/ptp2bGU9EHU), 12:56)
- *Gyroscope Interfacing Theory* ([youtu.be/5JdkwG2rr1o](https://youtu.be/5JdkwG2rr1o), 13:48)
- *Hall-Effect Sensor Interfacing Theory* ([youtu.be/T9GP\\_cnz7rQ](https://youtu.be/T9GP_cnz7rQ), 9:48)
- *H-Bridge and Geared Motor Interfacing Theory* ([youtu.be/W526ekpR8q4](https://youtu.be/W526ekpR8q4), 11:26)
- *Photointerrupter Characteristics* ([youtu.be/u1FVfEvSdkg](https://youtu.be/u1FVfEvSdkg), 4:59)
- *Keypad Interfacing Theory* ([youtu.be/oj2-CYSnyo0](https://youtu.be/oj2-CYSnyo0), 13:10)
- *LCD Character Display Interfacing Theory* ([youtu.be/m0Td7KbhvdI](https://youtu.be/m0Td7KbhvdI), 10:36)
- *LED Matrix Interfacing Theory* ([youtu.be/vsBjZBLdeNc](https://youtu.be/vsBjZBLdeNc), 9:51)
- *Discrete LED Interfacing Theory* ([youtu.be/9-R1GPVgFW0](https://youtu.be/9-R1GPVgFW0), 6:55)
- *LED Current Management* ([youtu.be/JW-19uXrWNU](https://youtu.be/JW-19uXrWNU), 15:06)
- *Motor Adapter for NI myRIO and Geared Motor Interfacing Theory* ([youtu.be/bkiopa5nxiE](https://youtu.be/bkiopa5nxiE), 8:54)
- *ADMP504 Microphone Interfacing Theory* ([youtu.be/991pj7yUmuY](https://youtu.be/991pj7yUmuY), 6:36)
- *Motor Interfacing Theory* ([youtu.be/C\\_22XZaL5TM](https://youtu.be/C_22XZaL5TM), 6:49)
- *Photocell Characteristics* ([youtu.be/geNeoFUjMjQ](https://youtu.be/geNeoFUjMjQ), 4:45)
- *Piezo Sensor Interfacing Theory* ([youtu.be/dHaPUJ7n-UI](https://youtu.be/dHaPUJ7n-UI), 5:13)
- *Potentiometer Characteristics* ([youtu.be/3gwwF9rF\\_zU](https://youtu.be/3gwwF9rF_zU), 7:51)
- *Pushbutton Interfacing Theory* ([youtu.be/e7UcL5Ycpho](https://youtu.be/e7UcL5Ycpho), 4:24)
- *IR Range Finder Interfacing Theory* ([youtu.be/Xwr-j-2WT3k](https://youtu.be/Xwr-j-2WT3k), 9:59)

- *Sonic Range Finder Interfacing Theory* ([youtu.be/UcpmrcJR\\_D8](https://youtu.be/UcpmrcJR_D8), 9:26)
- *Relay Interfacing Theory* ([youtu.be/jLFL9\\_EWlwI](https://youtu.be/jLFL9_EWlwI), 11:11)
- *RFID Reader Interfacing Theory* ([youtu.be/z1v0vCue83c](https://youtu.be/z1v0vCue83c), 11:29)
- *Servo Interfacing Theory* ([youtu.be/DOu5AvSDP2E](https://youtu.be/DOu5AvSDP2E), 7:18)
- *Seven-Segment LED Interfacing Theory* ([youtu.be/P0ER0VXvfSw](https://youtu.be/P0ER0VXvfSw), 4:11)
- *Buzzer/speaker Characteristics* ([youtu.be/8IbTWH9MpV0](https://youtu.be/8IbTWH9MpV0), 5:14)
- *Detect a Switch Signal Transition* ([youtu.be/GYBmRJ\\_qMrE](https://youtu.be/GYBmRJ_qMrE), 4:42)
- *Temperature Sensor Interfacing Theory* ([youtu.be/HwzTgYp5nF0](https://youtu.be/HwzTgYp5nF0), 10:03)
- *Thermistor Characteristics* ([youtu.be/US406sjBUxY](https://youtu.be/US406sjBUxY), 4:54)
- *Thermistor Resistance Measurement* ([youtu.be/PhZ2QlCrwuQ](https://youtu.be/PhZ2QlCrwuQ), 6:10)

## Tutorials

- *Measure Resistance with a Voltage Divider* ([youtu.be/9KUVD7RkxNI](https://youtu.be/9KUVD7RkxNI), 9:44)
- *Resistive-Sensor Threshold Detector* ([youtu.be/TqLXJroefTA](https://youtu.be/TqLXJroefTA), 9:22)
- *Detect a Switch Signal Transition* ([youtu.be/GYBmRJ\\_qMrE](https://youtu.be/GYBmRJ_qMrE), 4:42)
- *Serial Communication: UART* ([youtu.be/odN66E85J5E](https://youtu.be/odN66E85J5E), 7:56)
- *Serial Communication: SPI* ([youtu.be/GaXtDamw5As](https://youtu.be/GaXtDamw5As), 7:02)
- *Serial Communication: I2C* ([youtu.be/7CgNF78pYQM](https://youtu.be/7CgNF78pYQM), 8:47)
- *Deploy a Stand-Alone Application* ([youtu.be/JXoJECRS-eo](https://youtu.be/JXoJECRS-eo), 8:29)

## LabVIEW Techniques for myRIO

- *Analog Input Express VI* ([youtu.be/N6Mi-VjBlmc](https://youtu.be/N6Mi-VjBlmc), 2:00)
- *Digital Input Express VI* ([youtu.be/litswKgOmZA](https://youtu.be/litswKgOmZA), 1:53)
- *Digital Input Low-Level VIs* ([youtu.be/4nzt7THqU8U](https://youtu.be/4nzt7THqU8U), 4:09)
- *Digital Output Express VI* ([youtu.be/Y8mKdsMAqrU](https://youtu.be/Y8mKdsMAqrU), 2:21)
- *Digital Output Low-Level VIs* ([youtu.be/WvnInG3ffqY](https://youtu.be/WvnInG3ffqY), 4:53)
- *PWM Express VI* ([youtu.be/mVN9jfwXleI](https://youtu.be/mVN9jfwXleI), 2:41)
- *Run-Time Selectable I/O Channels* ([youtu.be/uJW7CaL6L5c](https://youtu.be/uJW7CaL6L5c), 1:54)
- *SPI Express VI* ([youtu.be/S7KkTeMfmc8](https://youtu.be/S7KkTeMfmc8), 5:51)
- *UART Express VI* ([youtu.be/0FMnkFDsGQs](https://youtu.be/0FMnkFDsGQs), 5:29)

## LabVIEW Demo Project Walk-Throughs

- *Accelerometer Demo Walk-Through* ([youtu.be/-\\_GWEsrfxU4](https://youtu.be/-_GWEsrfxU4), 8:37)
- *Ambient Light Sensor Demo Walk-Through* ([youtu.be/XcwEjm6TOig](https://youtu.be/XcwEjm6TOig), 3:02)
- *Bluetooth Demo Walk-Through* ([youtu.be/LFCThGa681A](https://youtu.be/LFCThGa681A), 15:08)
- *Compass Demo Walk-Through* ([youtu.be/bWew4fHWVKo](https://youtu.be/bWew4fHWVKo), 7:48)
- *DIP Switch Demo Walk-Through* ([youtu.be/ZMyYRSsQCac](https://youtu.be/ZMyYRSsQCac), 2:30)



- *Dpot Demo Walk-Through* ([youtu.be/dtwXOj5vvy4](https://youtu.be/dtwXOj5vvy4), 4:57)
- *EEPROM Demo Walk-Through* ([youtu.be/UNdVUnYHE4U](https://youtu.be/UNdVUnYHE4U), 13:07)
- *Electret Mic Demo Walk-Through* ([youtu.be/kZoFwQRYz98](https://youtu.be/kZoFwQRYz98), 2:52)
- *Rotary Encoder Demo Walk-Through* ([youtu.be/nmG1RqhQ6Rw](https://youtu.be/nmG1RqhQ6Rw), 3:15)
- *GPS Receiver Demo Walk-Through* ([youtu.be/SebcpkbYBd4](https://youtu.be/SebcpkbYBd4), 9:46)
- *Gyroscope Demo Walk-Through* ([youtu.be/o\\_iuY0M3yDk](https://youtu.be/o_iuY0M3yDk), 6:36)
- *Hall-Effect Sensor Demo Walk-Through* ([youtu.be/BCJLg-WbIK4](https://youtu.be/BCJLg-WbIK4), 2:36)
- *H-Bridge and Geared Motor Demo Walk-Through* ([youtu.be/Q1UXVtVN-oQ](https://youtu.be/Q1UXVtVN-oQ), 6:01)
- *Photointerrupter Demo Walk-Through* ([youtu.be/yuzNb1ZDbv4](https://youtu.be/yuzNb1ZDbv4), 3:22)
- *Keypad Demo Walk-Through* ([youtu.be/7r\\_LwcDa2AM](https://youtu.be/7r_LwcDa2AM), 4:58)
- *LCD (I2C) Demo Walk-Through* ([youtu.be/qbD31AeqOMk](https://youtu.be/qbD31AeqOMk), 4:32)
- *LCD (SPI) Demo Walk-Through* ([youtu.be/oOXYryu4Y-c](https://youtu.be/oOXYryu4Y-c), 4:23)
- *LCD (UART) Demo Walk-Through* ([youtu.be/JsEMMnIWg4k](https://youtu.be/JsEMMnIWg4k), 3:44)
- *LED Matrix Demo Walk-Through* ([youtu.be/Bqq63sKwQKE](https://youtu.be/Bqq63sKwQKE), 12:14)
- *LED Demo Walk-Through* ([youtu.be/SHJ-vu4jorU](https://youtu.be/SHJ-vu4jorU), 2:03)
- *MEMS Mic Demo Walk-Through* ([youtu.be/2ZpI\\_uDwOg4](https://youtu.be/2ZpI_uDwOg4), 2:31)
- *Motor Demo Walk-Through* ([youtu.be/UCqFck0CLpc](https://youtu.be/UCqFck0CLpc), 1:56)
- *Motor Adapter for NI myRIO Demo Walk-Through* ([youtu.be/wg8FQ7kEmdE](https://youtu.be/wg8FQ7kEmdE), 8:08)
- *Photocell Demo Walk-Through* ([youtu.be/jZQqsc5GmoY](https://youtu.be/jZQqsc5GmoY), 3:07)
- *Piezoelectric-Effect Sensor Demo Walk-Through* ([youtu.be/b1me4f-3i0E](https://youtu.be/b1me4f-3i0E), 2:54)
- *Potentiometer Demo Walk-Through* ([youtu.be/RYeKIuU6DX8](https://youtu.be/RYeKIuU6DX8), 3:07)
- *Pushbutton Demo Walk-Through* ([youtu.be/Xm1A4Cw2POU](https://youtu.be/Xm1A4Cw2POU), 3:16)
- *IR Range Finder Demo Walk-Through* ([youtu.be/BFge1RQxJ\\_E](https://youtu.be/BFge1RQxJ_E), 3:51)
- *Sonic Range Finder Demo Walk-Through* ([youtu.be/-MVa9HklgKI](https://youtu.be/-MVa9HklgKI), 3:33)
- *Relay Demo Walk-Through* ([youtu.be/W2iukd8WVIA](https://youtu.be/W2iukd8WVIA), 3:30)
- *RFID Demo Walk-Through* ([youtu.be/Jovn0kPJOKs](https://youtu.be/Jovn0kPJOKs), 5:18)
- *Servo Demo Walk-Through* ([youtu.be/QXHe0DFbUdc](https://youtu.be/QXHe0DFbUdc), 4:23)
- *Seven-Segment LED Demo Walk-Through* ([youtu.be/ejyOo\\_k9Kl0](https://youtu.be/ejyOo_k9Kl0), 2:03)
- *Buzzer/speaker Demo Walk-Through* ([youtu.be/kW4v16GuAFE](https://youtu.be/kW4v16GuAFE), 2:06)
- *Temperature Sensor Demo Walk-Through* ([youtu.be/10ib10sojds](https://youtu.be/10ib10sojds), 6:25)
- *Thermistor Demo Walk-Through* ([youtu.be/xi0VIpGpf4w](https://youtu.be/xi0VIpGpf4w), 2:28)
- *USB Flash Drive Demo Walk-Through* ([youtu.be/Y1QukBt1lWI](https://youtu.be/Y1QukBt1lWI), 8:22)