

Sisteme cu microprocesoare

Laboratorul nr. 5

- Mediul de simulare Matlab – Simulink -
- și platforma de dezvoltare Arduino -

Obiective

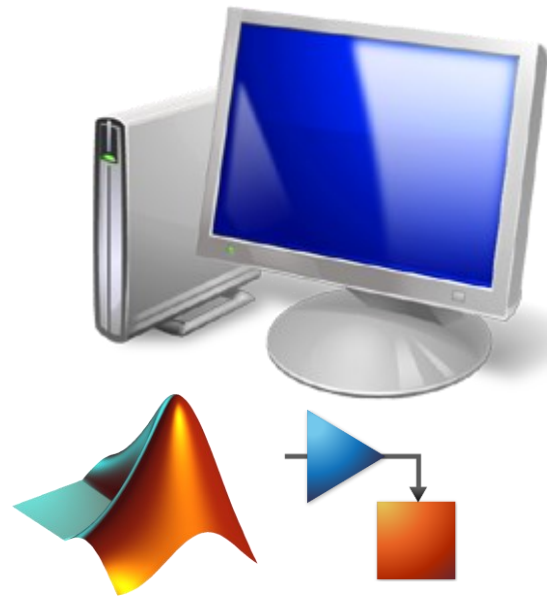
- Stabilirea legăturii dintre platforma de dezvoltare Arduino și mediul Matlab Simulink rulând pe un calculator gazdă prin intermediul unui cod – program de comunicație și decizie din memoria platformei Arduino;
- Dezvoltarea unor aplicații simple de achiziție și control în timp real cu ajutorul mediului Matlab – Simulink și al calculatorului gazdă;

Legătura dintre platforma Arduino și calculatorul gazdă rulând mediul de simulare Matlab - Simulink

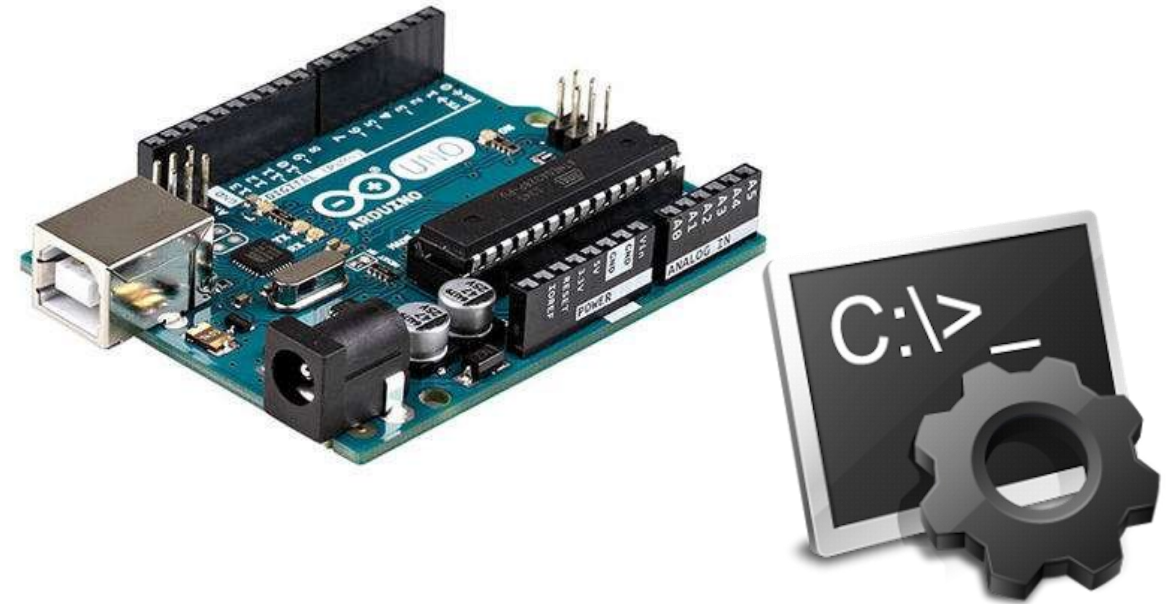
- **Există două componente – program necesare în vederea realizării unui model Simulink capabil să interacționeze cu platforma în timp real:**
 1. Programul încărcat în memoria platformei Arduino (adio.pde);
 2. Modelul Simulink, de pe calculatorul gazdă, în care există blocurile de comunicație Serial;

Legătura dintre platforma Arduino și calculatorul gazdă rulând mediul de simulare Matlab – Simulink ^{[1][2][3][4][5]}

Calculatorul gazdă



Platforma Arduino



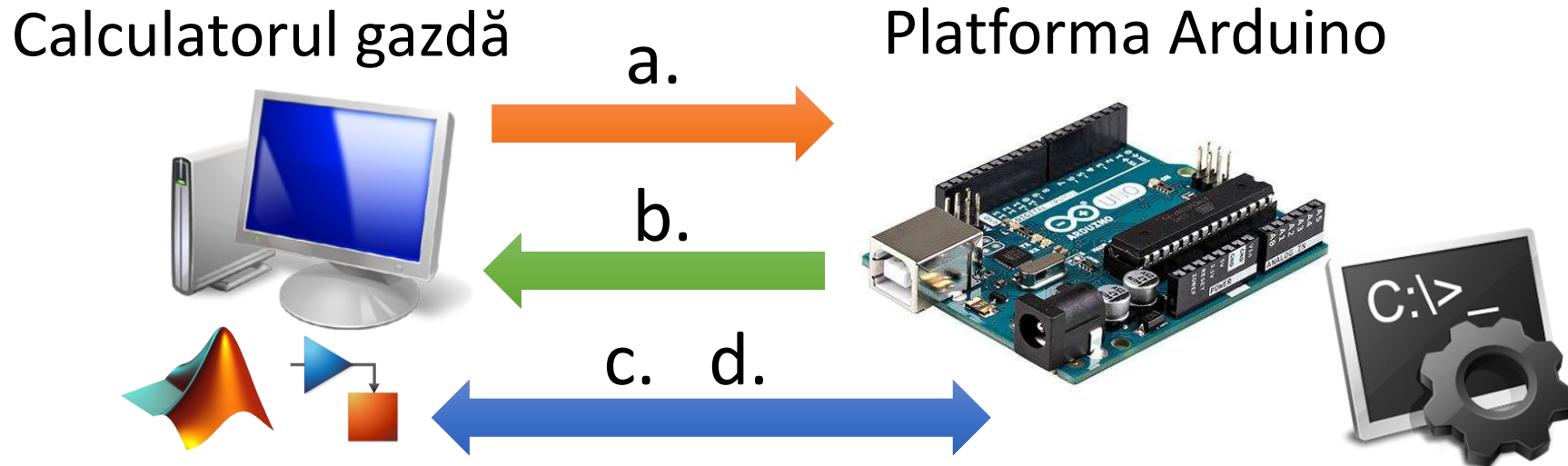
Comunicație
Serial

Rulează modelul de control
Matlab - Simulink

Rulează codul – program pentru stabilirea
comunicației cu modelul Matlab - Simulink

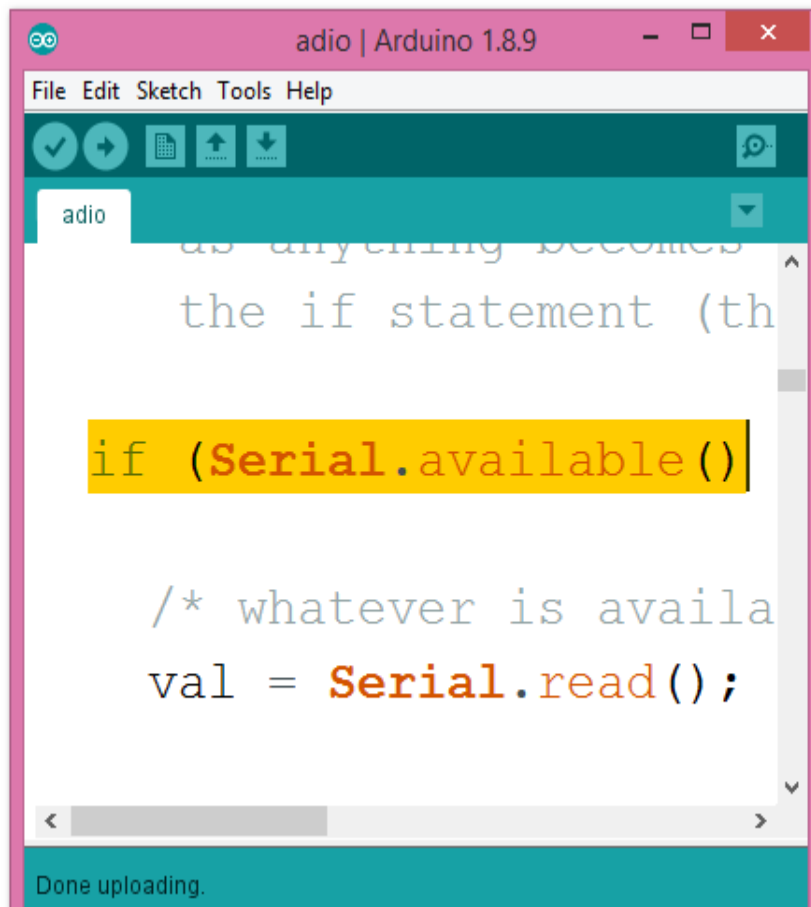
Programul încărcat în memoria platformei Arduino

- **Reprezintă fișierul specific pentru configurarea platformei Arduino în scopul:**
 - a. Acceptării comenzilor dinspre modelul Matlab – Simulink pe cale Serial spre Arduino;
 - b. Furnizării comenzilor dinspre Arduino înspre modelul Matlab – Simulink (achiziție);
 - c. Reglementării parametrilor de funcționare (timp de eșantionare, timp de simulare etc.);
 - d. Interacțiunii în timp real a modelului Matlab – Simulink cu platforma Arduino;



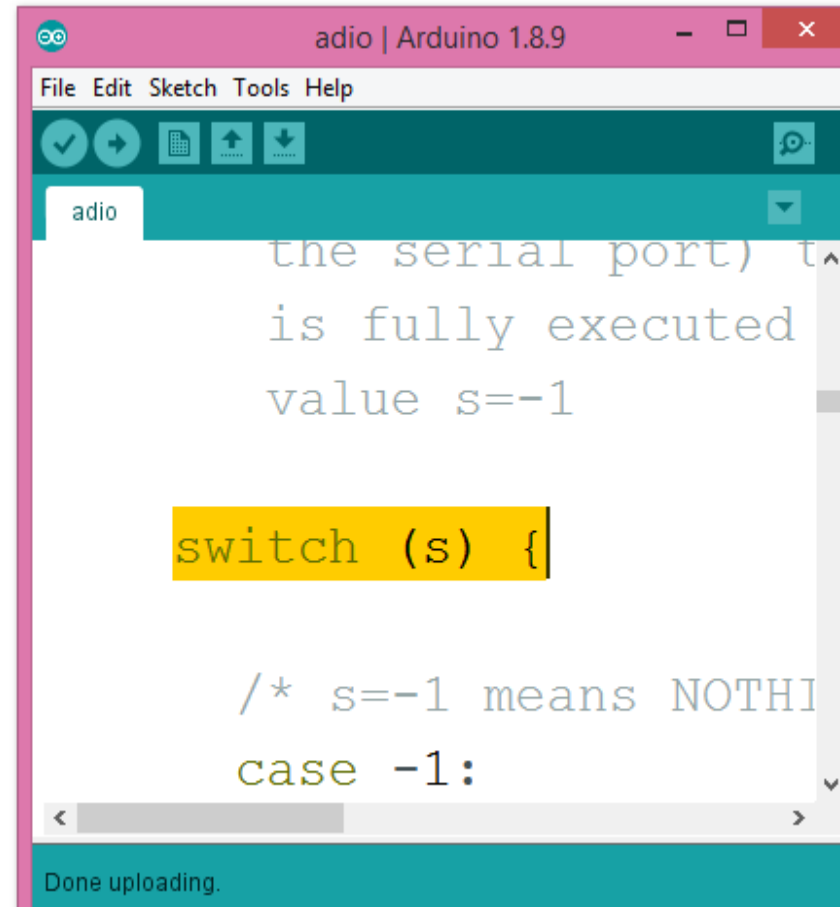
Programul încărcat în memoria platformei Arduino

- Este construit în jurul unei structuri de tip „switch – case”:
 - ✓ Și are rolul de a „răspunde” la comenzile primite de la model prin comunicație Serial;



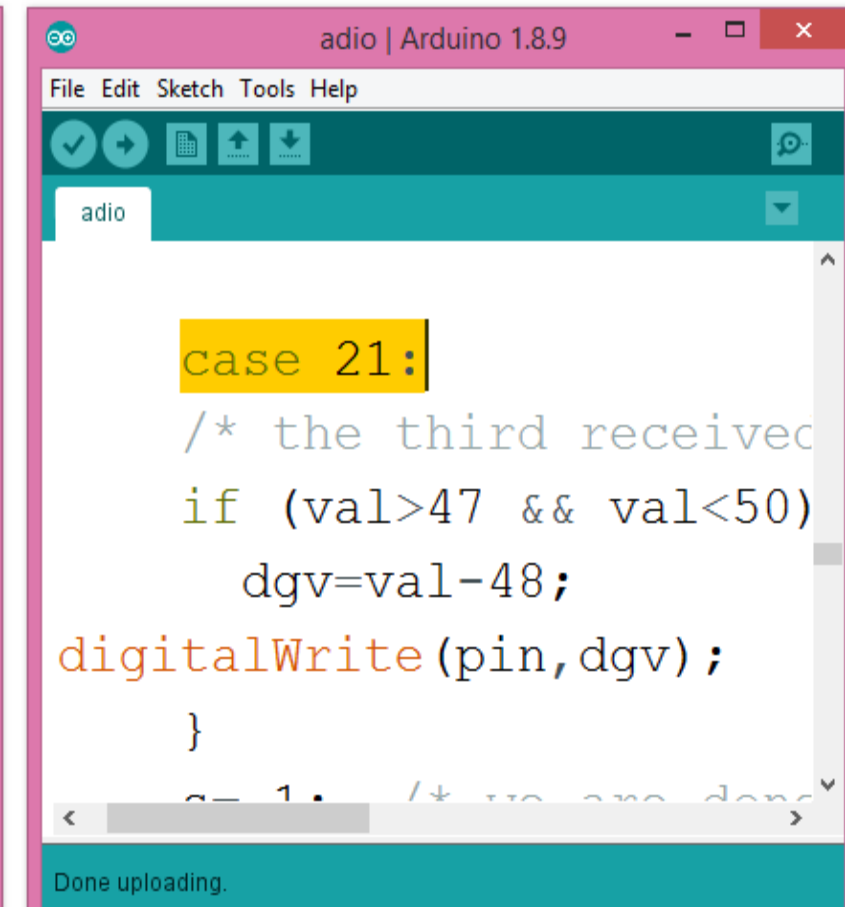
```
File Edit Sketch Tools Help
adio
as anything becomes
the if statement (th
if (Serial.available())
/* whatever is availa
val = Serial.read();
Done uploading.
```

A.



```
File Edit Sketch Tools Help
adio
the serial port) t
is fully executed
value s=-1
switch (s) {
/* s=-1 means NOTHI
case -1:
Done uploading.
```

B.

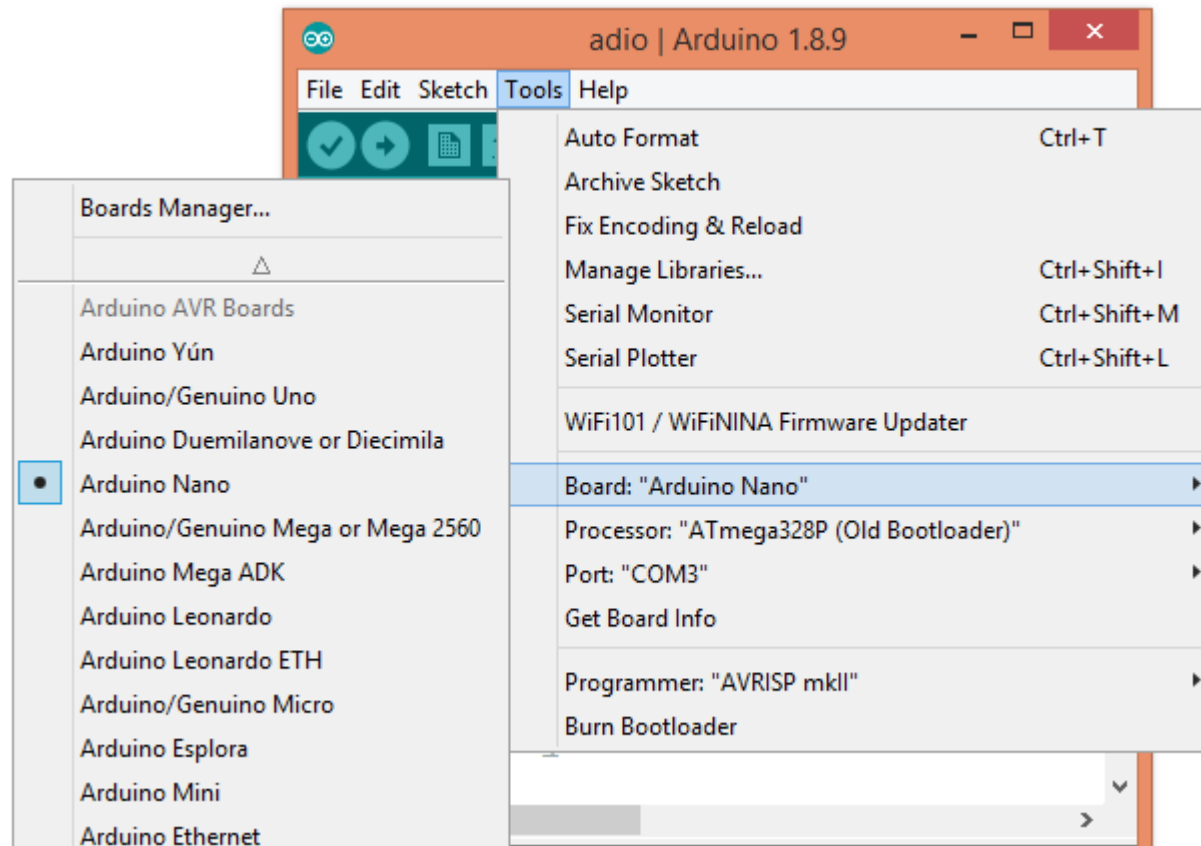


```
File Edit Sketch Tools Help
adio
case 21:
/* the third received
if (val>47 && val<50)
dgv=val-48;
digitalWrite (pin, dgv) ;
}
Done uploading.
```

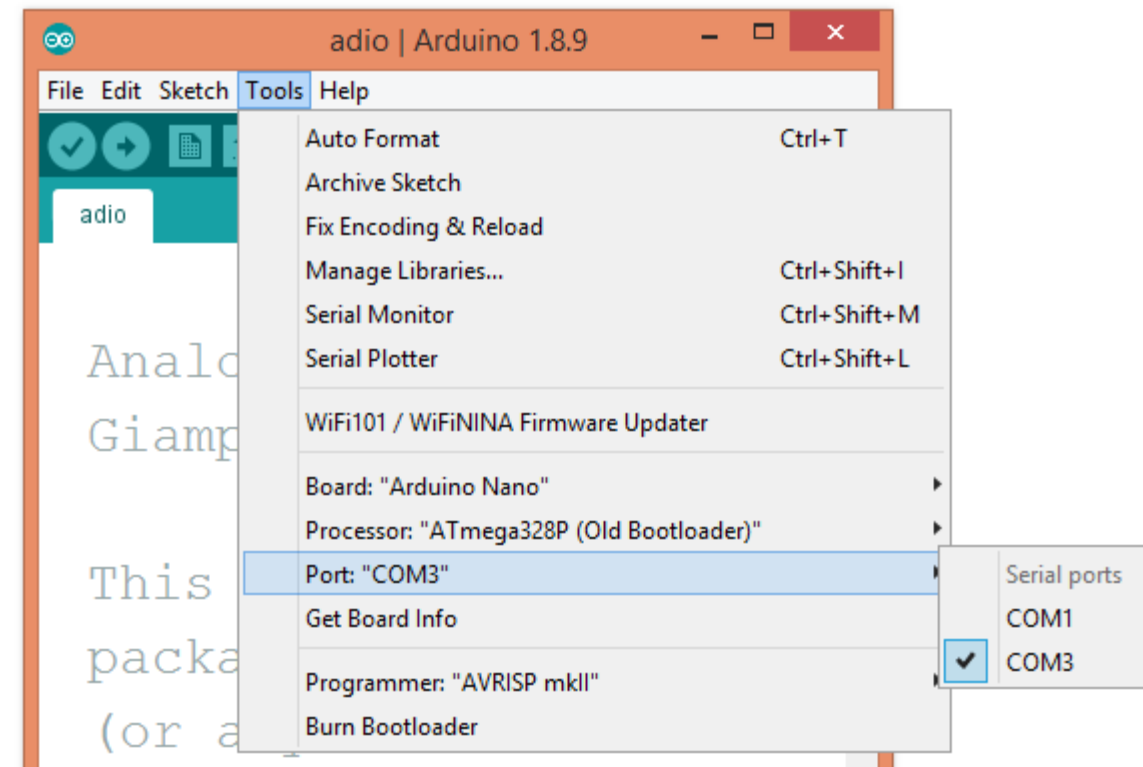
C.

Programul încărcat în memoria platformei Arduino

- În acest sens, se va proceda la încărcarea programului „adio.pde” în memorie:
 - ✓ Încărcarea programului se poate realiza prin intermediul aplicației ArduinoIDE;
 - ✓ Se vor parcurge toate etapele necesare pentru configurarea platformei și portului Serial;



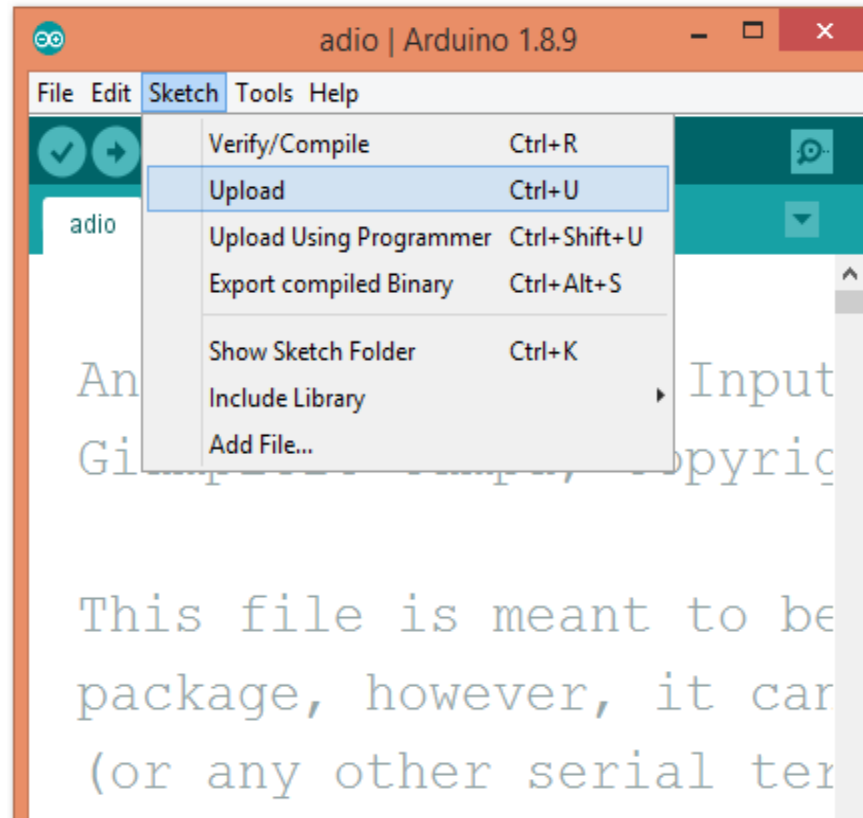
A.



B.

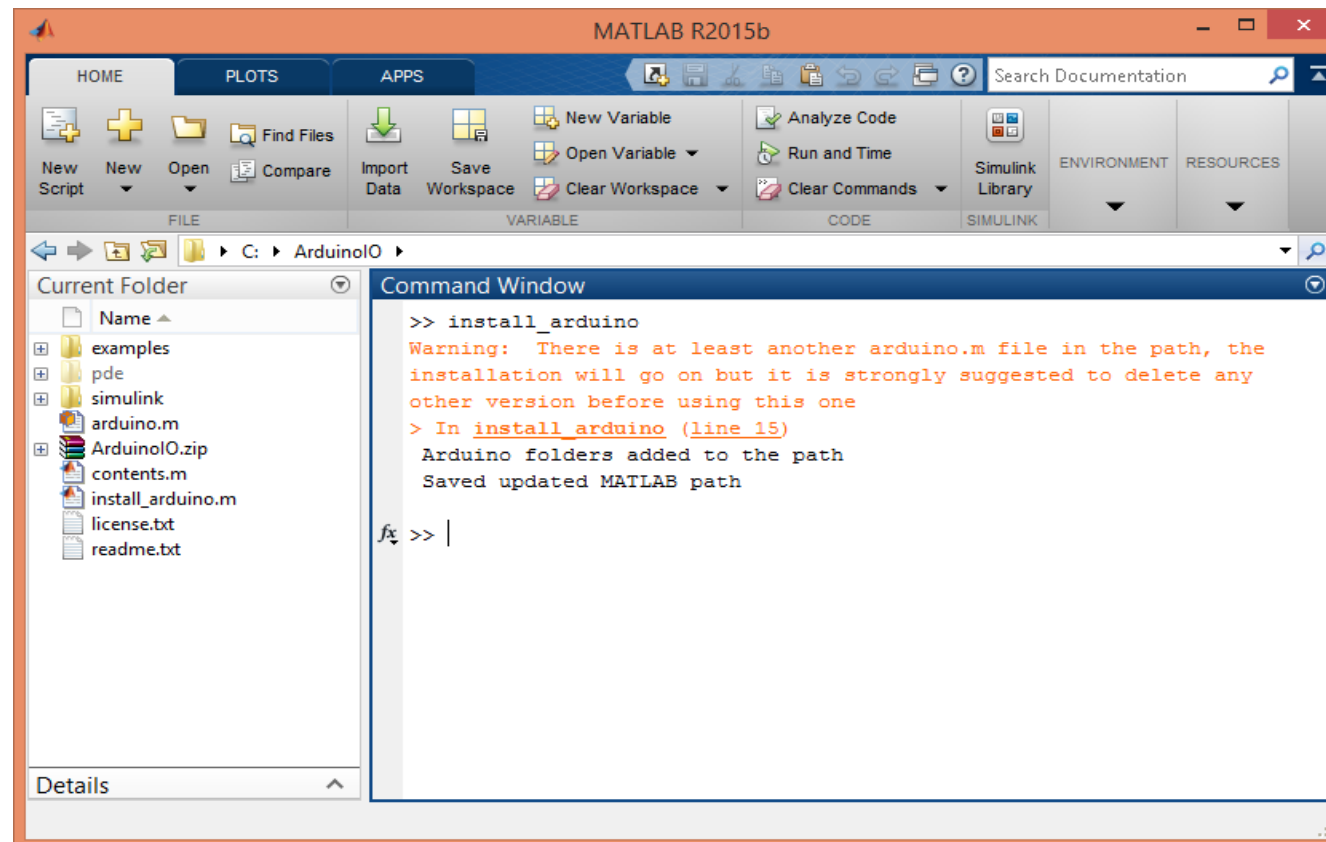
Programul încărcat în memoria platformei Arduino

- În acest sens, se va proceda la încărcarea programului „adio.pde” în memorie:
 - ✓ Încărcarea programului se poate realiza prin intermediul aplicației ArduinoIDE;
 - ✓ Se vor parcurge toate etapele necesare pentru configurarea platformei și portului Serial;



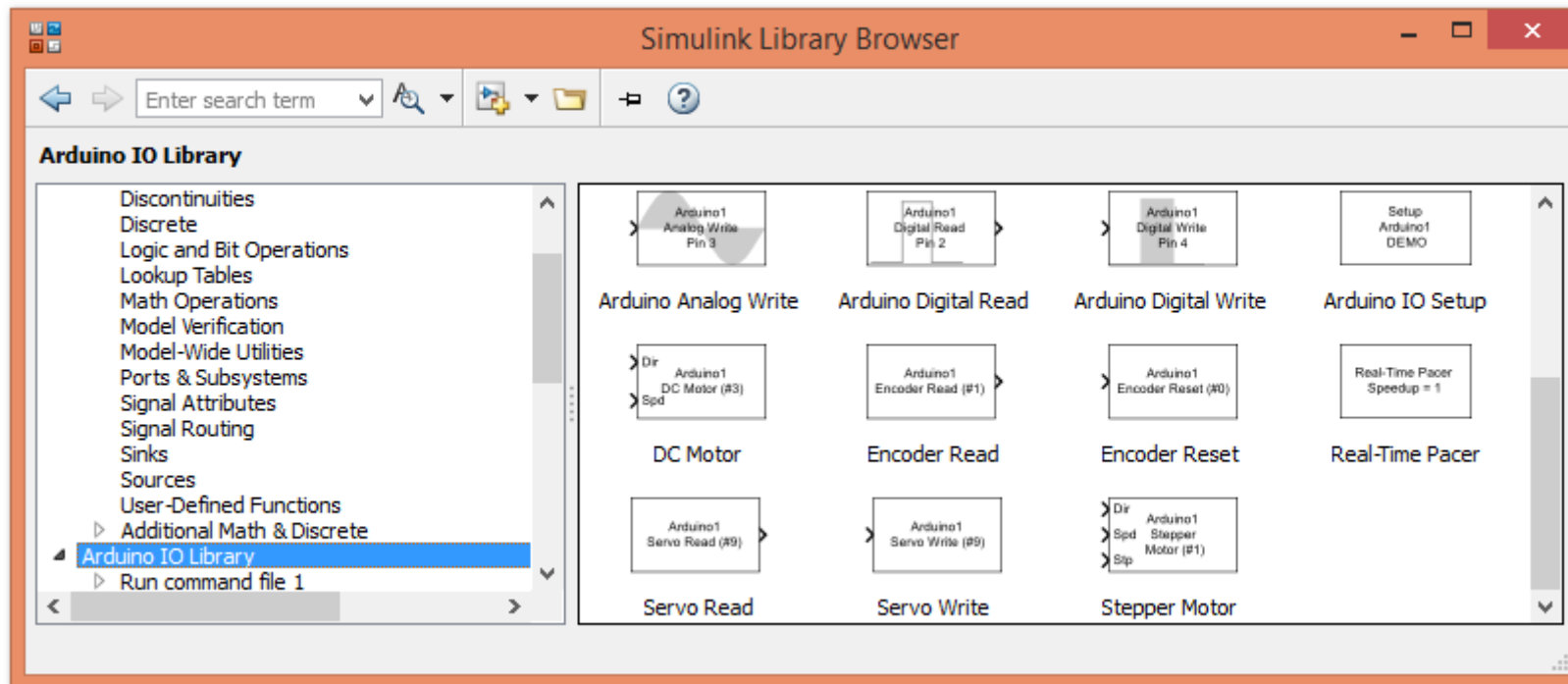
Modelul Simulink de pe calculatorul gazdă

- În mediul Matlab – Simulink se va instala paleta de instrumente ArduinoIO:
 - ✓ Prin intermediul comenzii „install_arduino” se va realiza acest lucru;
 - ✓ Este necesar ca pachetul „ArduinoIO” să fie în spațiul de lucru al mediului Simulink;



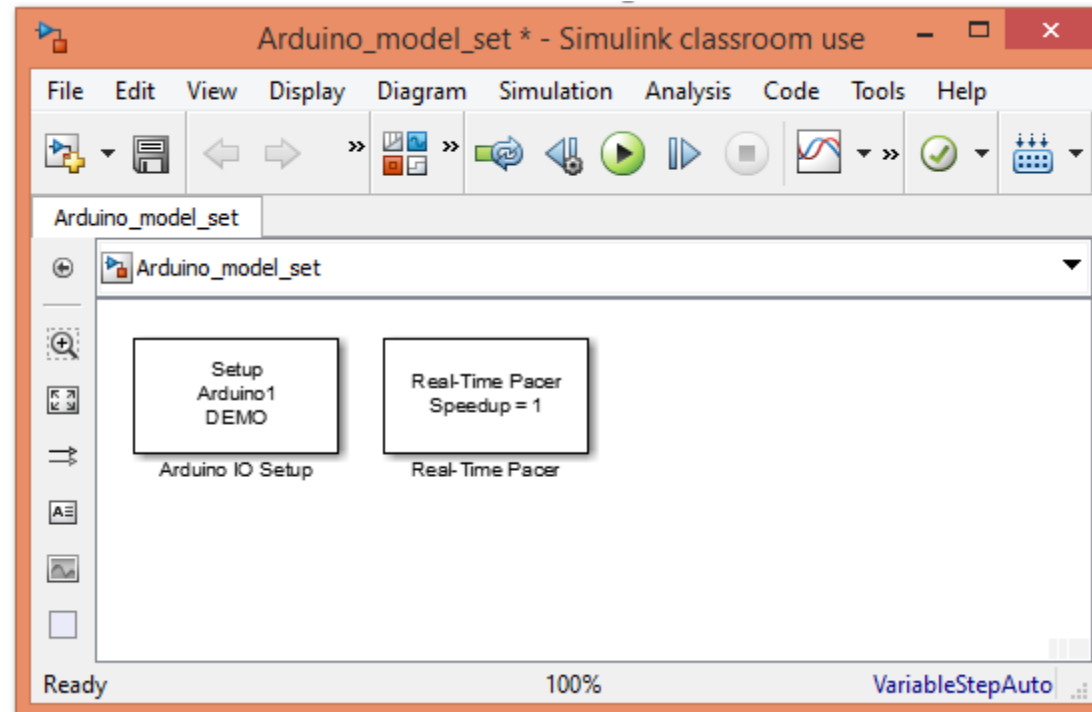
Modelul Simulink de pe calculatorul gazdă

- Având platforma Arduino conectată la calculatorul gazdă:
 - ✓ Se vor particulariza parametrii de identificare ai platformei;
 - ✓ Se vor particulariza parametrii pentru bunul mers al simulării în timp real;
 - ✓ Se vor concepe modele de control în timp real pe baza instrumentelor ArduinoIO;



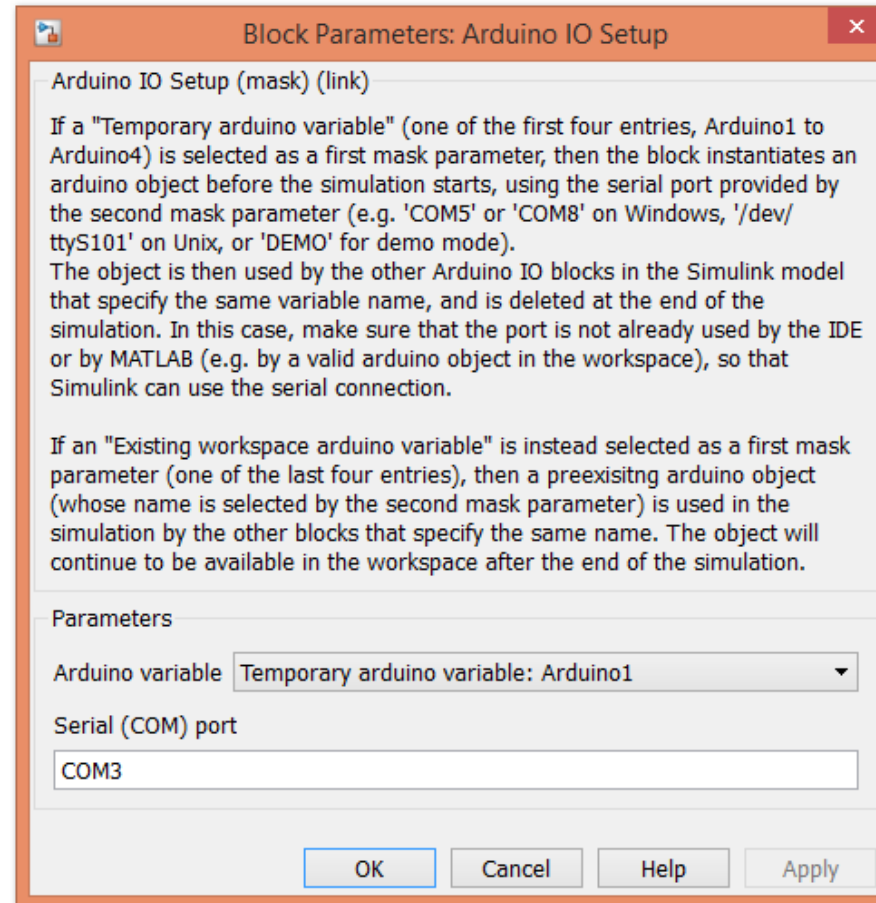
Modelul Simulink de pe calculatorul gazdă

- În cadrul unui model Simulink nou se vor introduce blocurile principale:
 - ✓ „Arduino IO Setup” – pentru inițializarea conexiunii dintre model și platforma Arduino;
 - ✓ „Real - Time Pacer” – pentru stabilirea parametrilor de temporizare ai simulării;



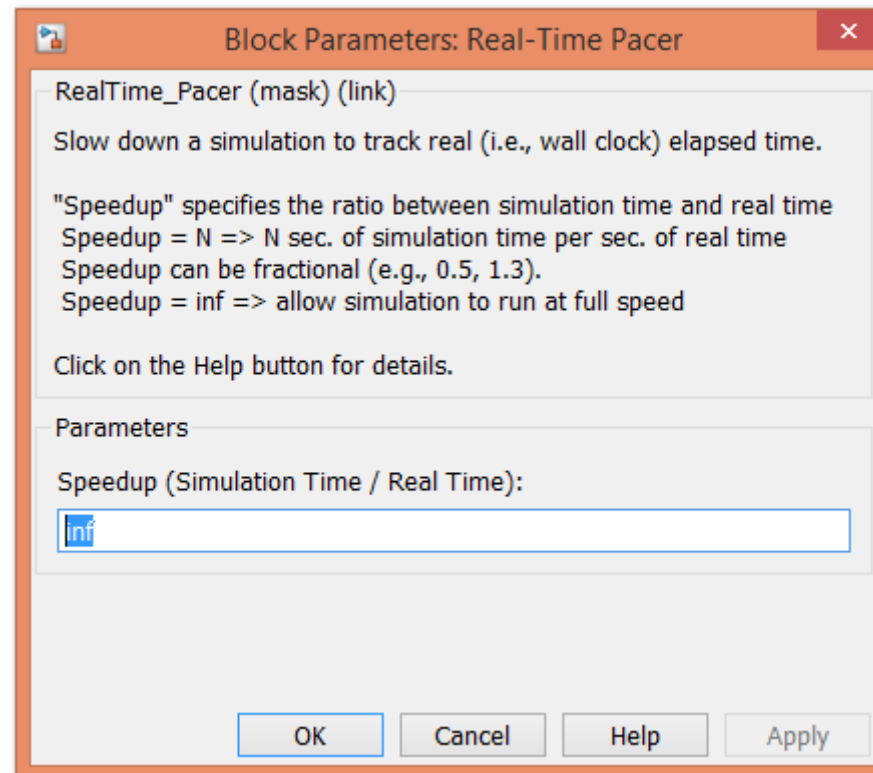
Modelul Simulink de pe calculatorul gazdă

- În cadrul aceluiaș model Simulink se vor particulariza parametrii precum:
 - ✓ Portul de comunicație Serial („COMx”) din cadrul blocului „Arduino IO Setup”;



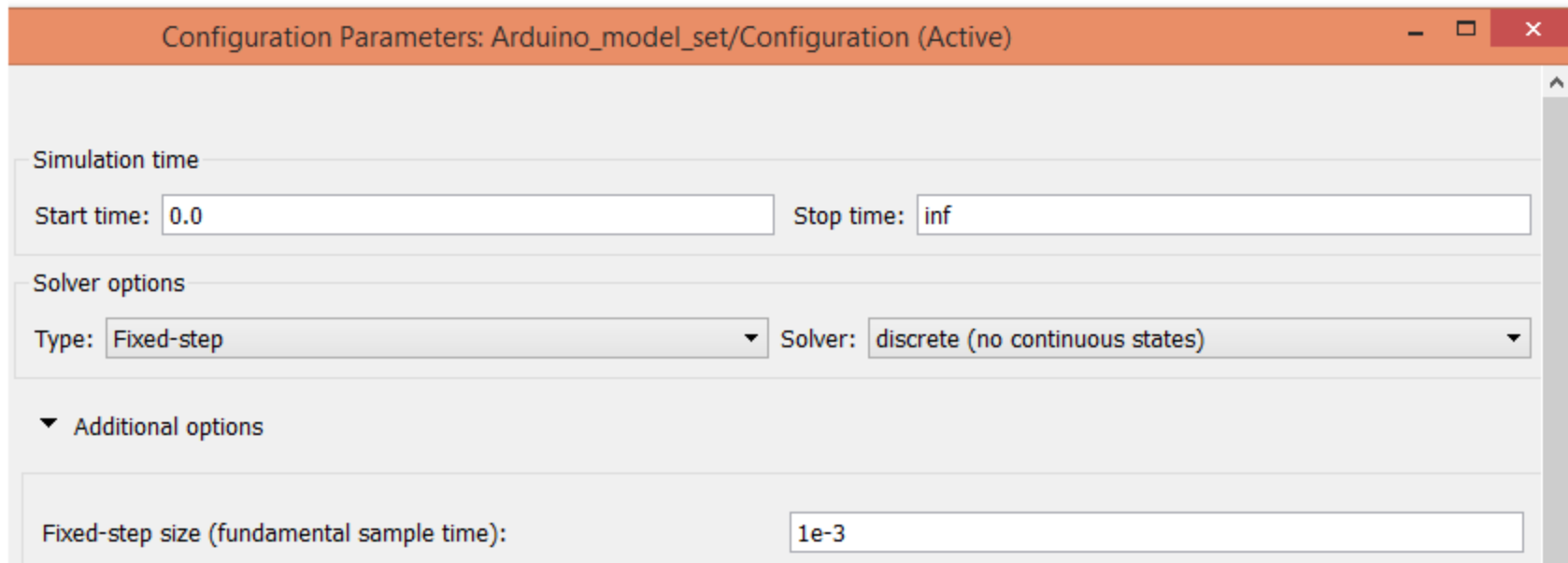
Modelul Simulink de pe calculatorul gazdă

- În cadrul aceluiaș model Simulink se vor particulariza parametrii precum:
 - ✓ Pasul de parcurgere în timp a modelului prin intermediul blocului „Real – Time Pacer”;



Modelul Simulink de pe calculatorul gazdă

- În cadrul aceluiaș model Simulink se vor particulariza parametrii precum:
 - ✓ Metoda de tratare a ecuațiilor din cadrul modelului (Solver options: Fixed-step);
 - ✓ Tipul metodei de rezolvare (Solver: discrete (no continuous state));
 - ✓ Timpul total de simulare (Stop time: inf);
 - ✓ Timpul total de eșantionare (Fixed-step size (fundamental sample time): 1e-3);


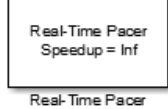

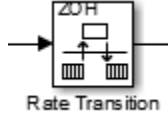

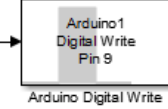


Aplicații

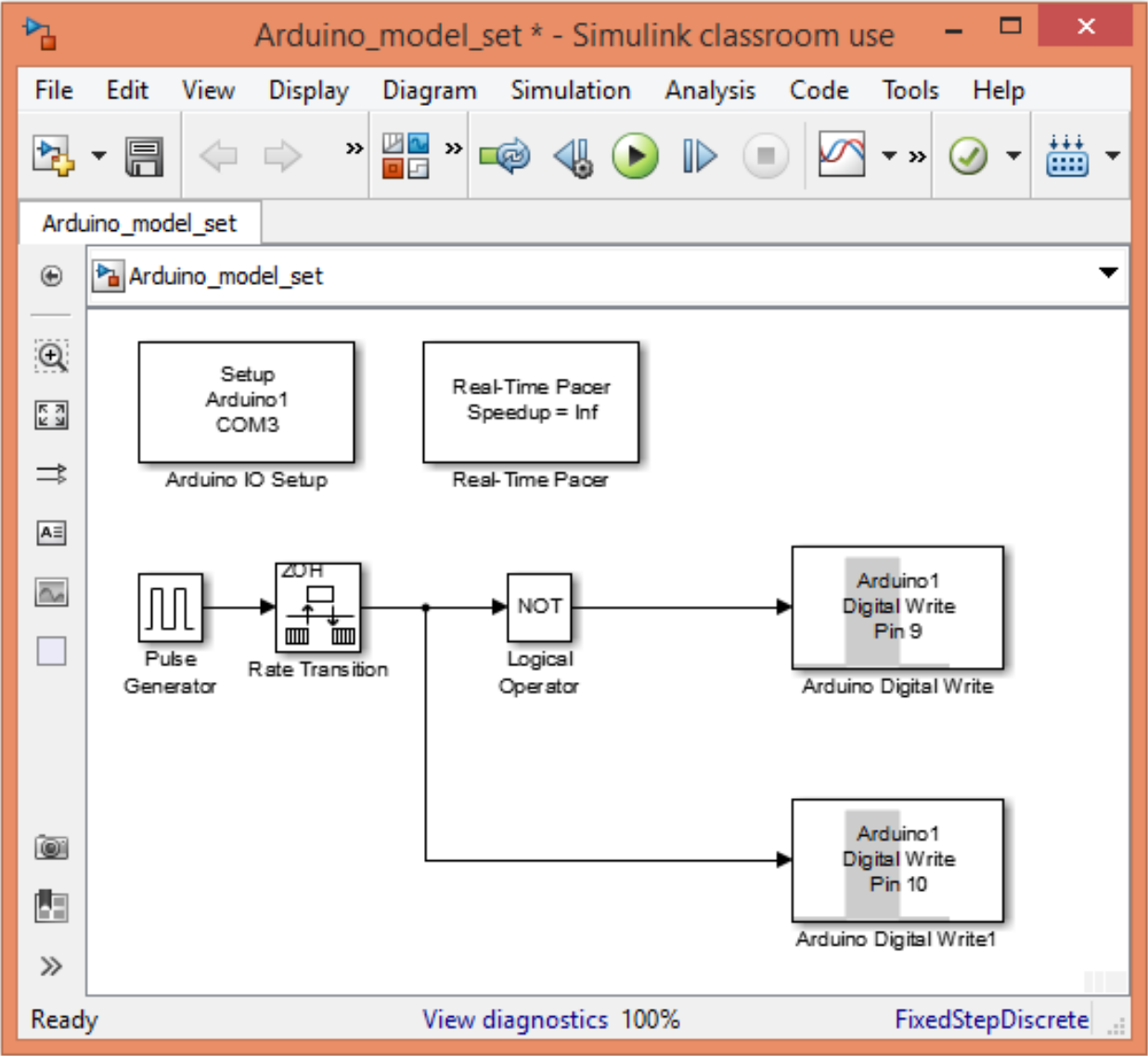
- **În urma particularizării parametrilor indicați, se propune realizarea aplicațiilor:**
 1. Semnalizare alternativă cu două diode electroluminiscente (LED);
 2. Variația intensității luminoase pe baza semnalelor modulate în lățime;
 3. Achiziția unui semnal digital;
 4. Achiziția unui semnal analogic;
 5. Generarea unor semnale de comandă în funcție de mărimea achiziționată;

Semnalizare alternativă cu două diode electroluminiscente

- Blocurile necesare în vederea realizării modelului se găsesc în categoriile următoare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 <p>Setup Arduino1 COM3 Arduino IO Setup</p>	Arduino IO Setup	Arduino IO Library
 <p>Real-Time Pacer Speedup = Inf Real-Time Pacer</p>	Real – Time Pacer	Arduino IO Library
 <p>Pulse Generator</p>	Pulse Generator	Simulink → Sources
 <p>ZOH Rate Transition</p>	Rate Transition	Simulink → Signal Attributes
 <p>NOT Logical Operator</p>	Logical Operator	Simulink → Logic and Bit Operations
 <p>Arduino1 Digital Write Pin 9 Arduino Digital Write</p>	Arduino Digital Write	Arduino IO Library

Semnalizare alternativă cu două diode electroluminiscente




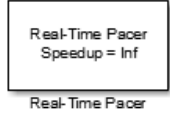

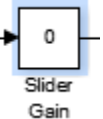
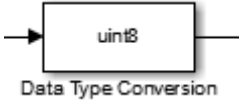
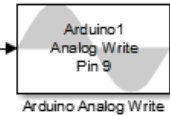


Semnalizare alternativă cu două diode electroluminiscente

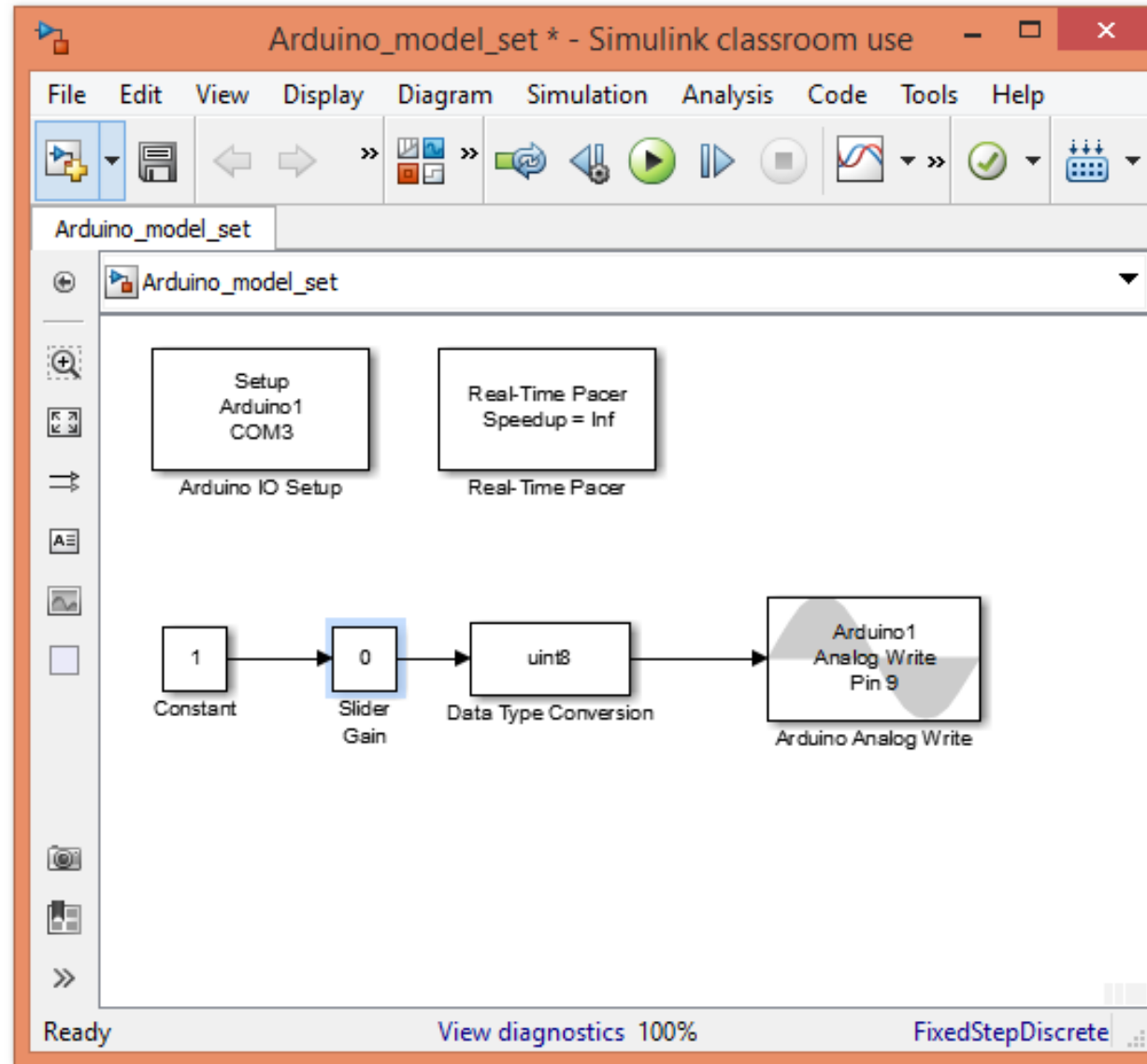
Material video demonstrativ nr. 1

Variația intensității luminoase pe baza semnalelor modulate în lățime

- Blocurile necesare în vederea realizării modelului se găsesc în categoriile următoare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 <p>Setup Arduino1 COM3 Arduino IO Setup</p>	Arduino IO Setup	Arduino IO Library
 <p>Real-Time Pacer Speedup = Inf Real-Time Pacer</p>	Real – Time Pacer	Arduino IO Library
 <p>1 Constant</p>	Constant	Simulink → Sources
 <p>0 Slider Gain</p>	Slider Gain	Simulink → Math Operations
 <p>uint8 Data Type Conversion</p>	Data Type Conversion	Simulink → Signal Attributes
 <p>Arduino1 Analog Write Pin 9 Arduino Analog Write</p>	Arduino Analog Write	Arduino IO Library

Variația intensității luminoase pe baza semnalelor modulate în lățime




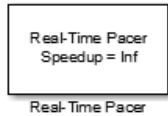
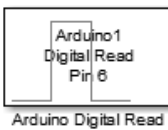
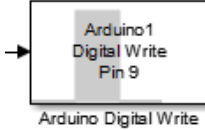

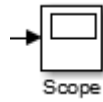


Variația intensității luminoase pe baza semnalelor modulate în lățime

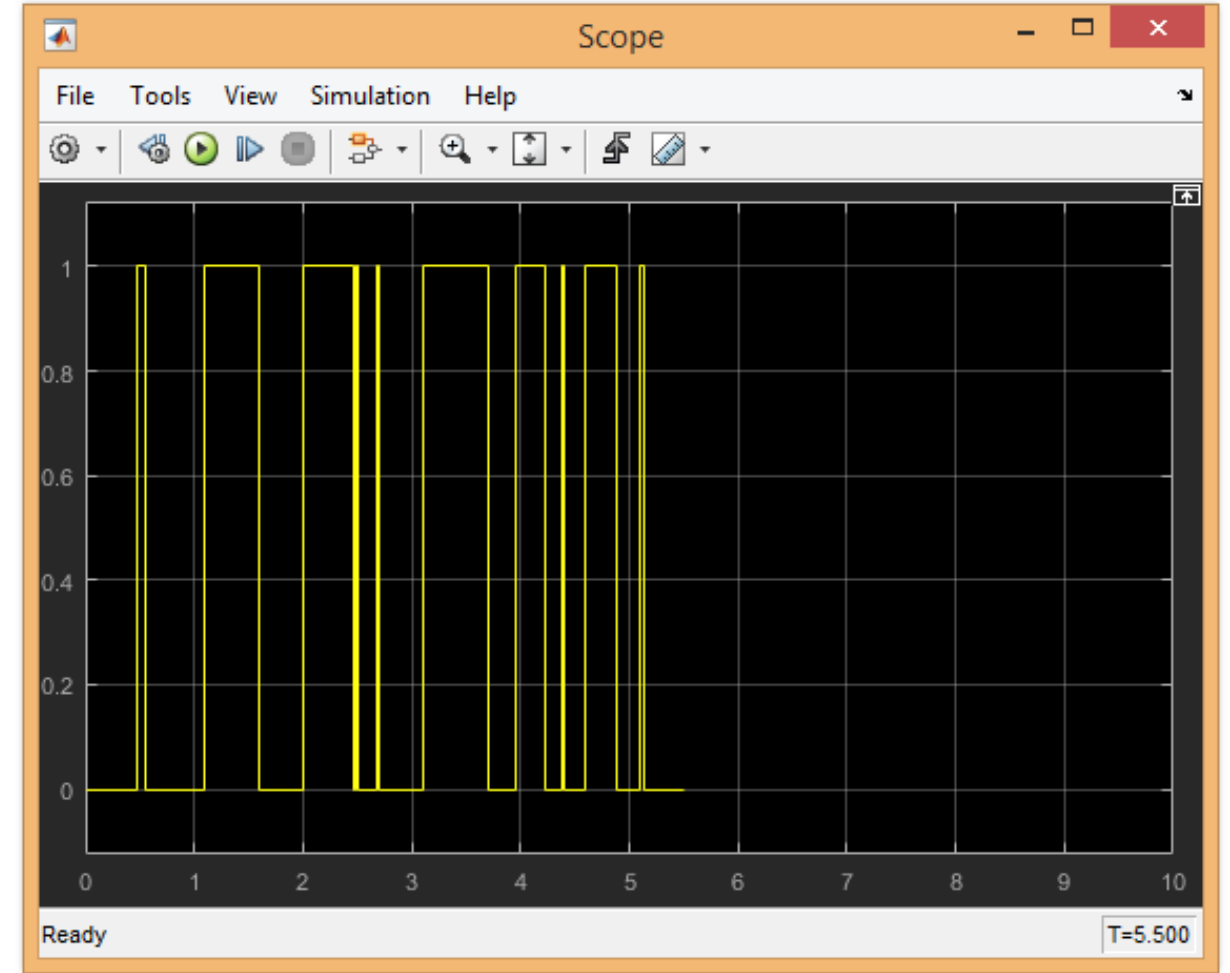
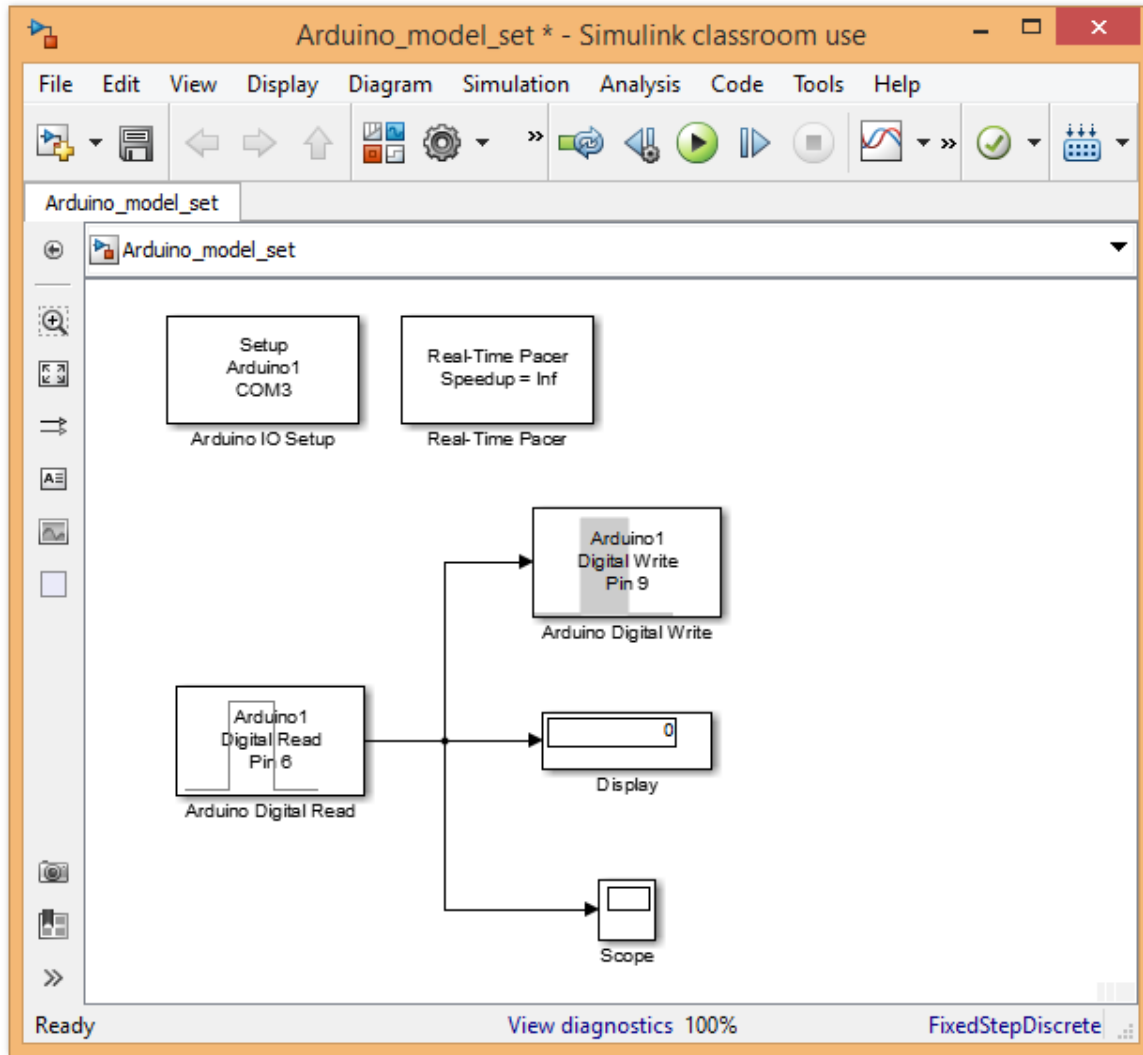
[Material video demonstrativ nr. 2](#)

Achiziția unui semnal digital

- Blocurile necesare în vederea realizării modelului se găsesc în categoriile următoare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 <p>Setup Arduino1 COM3 Arduino IO Setup</p>	Arduino IO Setup	Arduino IO Library
 <p>Real-Time Pacer Speedup = Inf Real-Time Pacer</p>	Real – Time Pacer	Arduino IO Library
 <p>Arduino1 Digital Read Pin 6 Arduino Digital Read</p>	Arduino Digital Read	Arduino IO Library
 <p>Arduino1 Digital Write Pin 9 Arduino Digital Write</p>	Arduino Digital Write	Arduino IO Library
 <p>Display</p>	Display	Simulink → Sinks
 <p>Scope</p>	Scope	Simulink → Sinks

Achiziția unui semnal digital




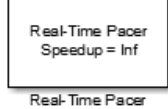
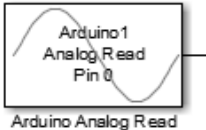
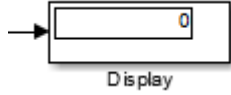
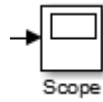


Achiziția unui semnal digital

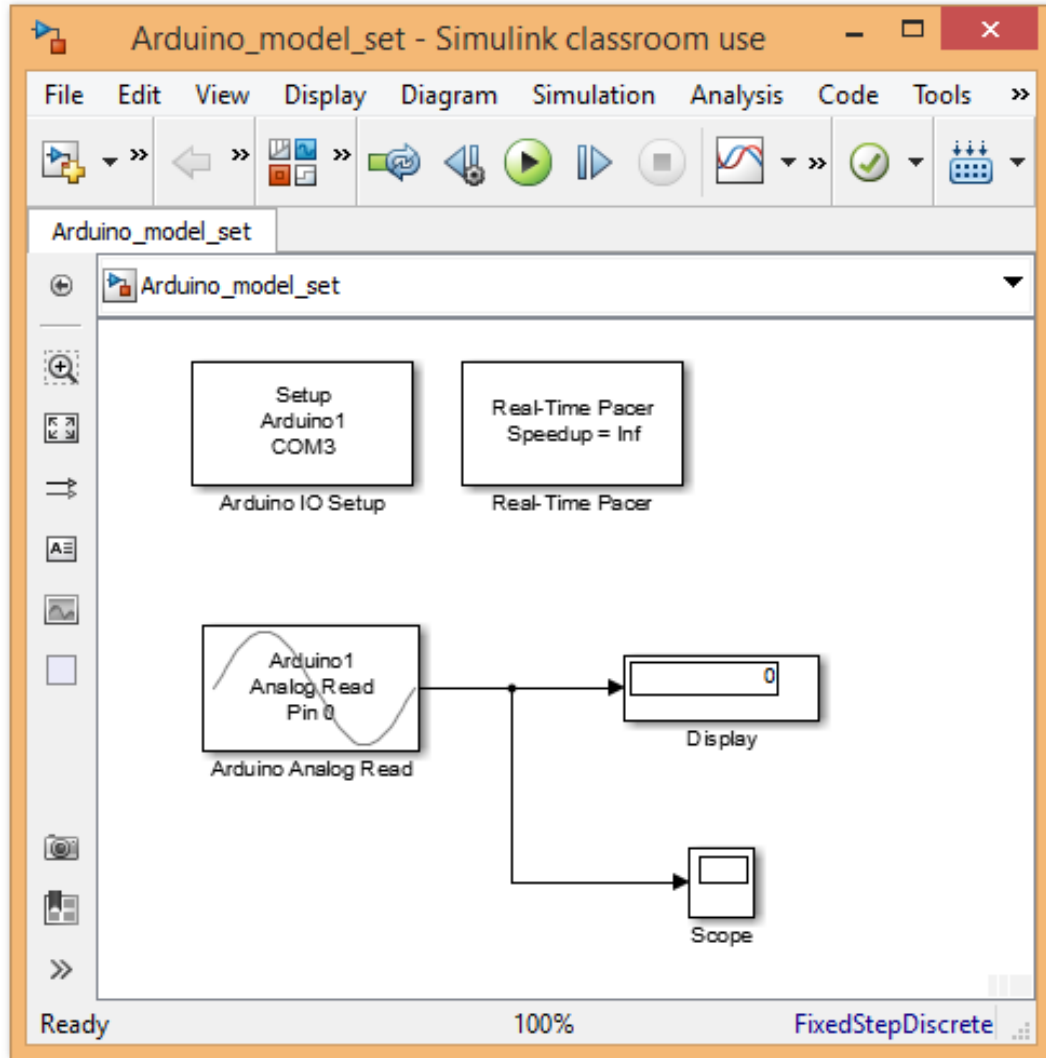
[Material video demonstrativ nr. 3](#)

Achiziția unui semnal analogic

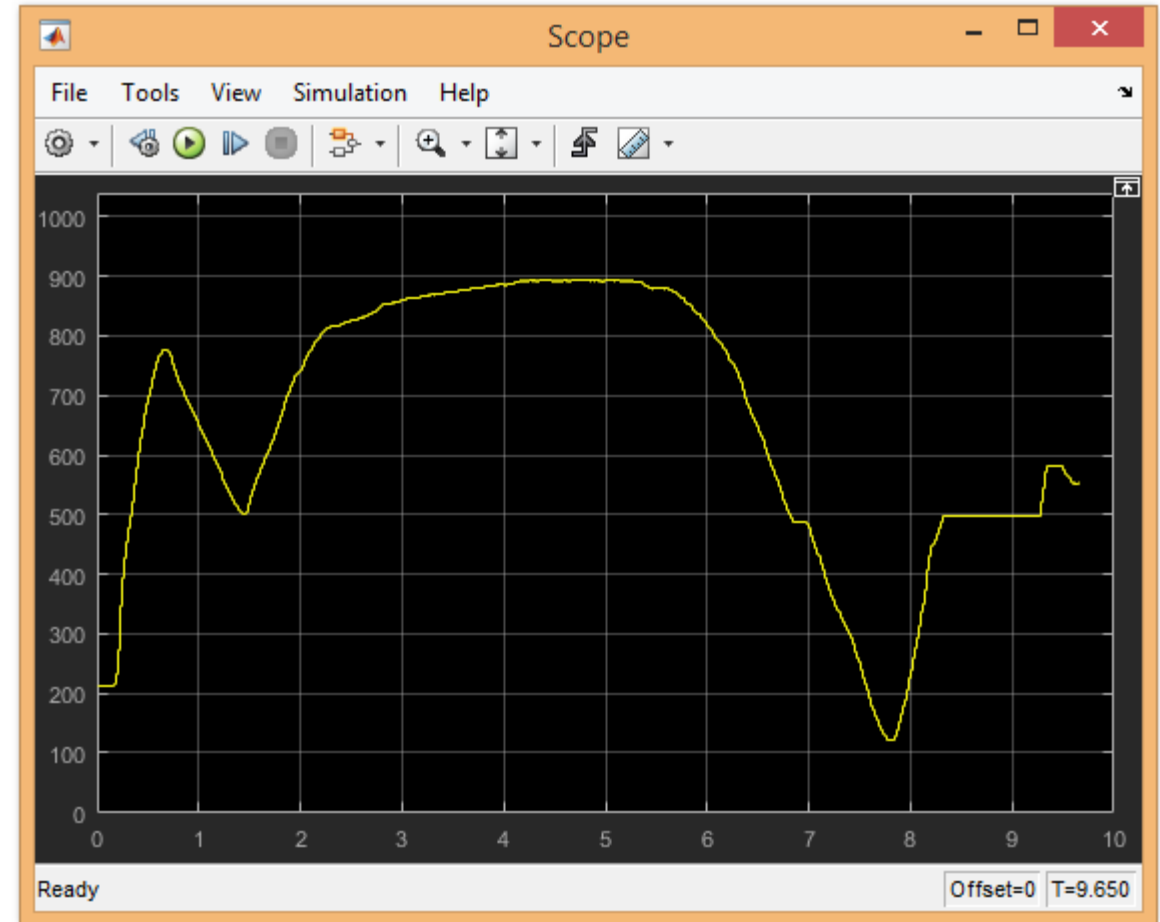
- Blocurile necesare în vederea realizării modelului se găsesc în categoriile următoare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 <p>Setup Arduino1 COM3 Arduino IO Setup</p>	Arduino IO Setup	Arduino IO Library
 <p>Real-Time Pacer Speedup = Inf Real-Time Pacer</p>	Real – Time Pacer	Arduino IO Library
 <p>Arduino1 Analog Read Pin 0 Arduino Analog Read</p>	Arduino Analog Read	Arduino IO Library
 <p>Display</p>	Display	Simulink → Sinks
 <p>Scope</p>	Scope	Simulink → Sinks

Achiziția unui semnal analogic



A.



B.


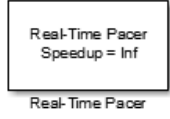
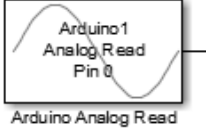
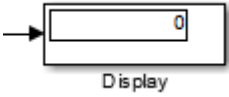



Achiziția unui semnal analogic

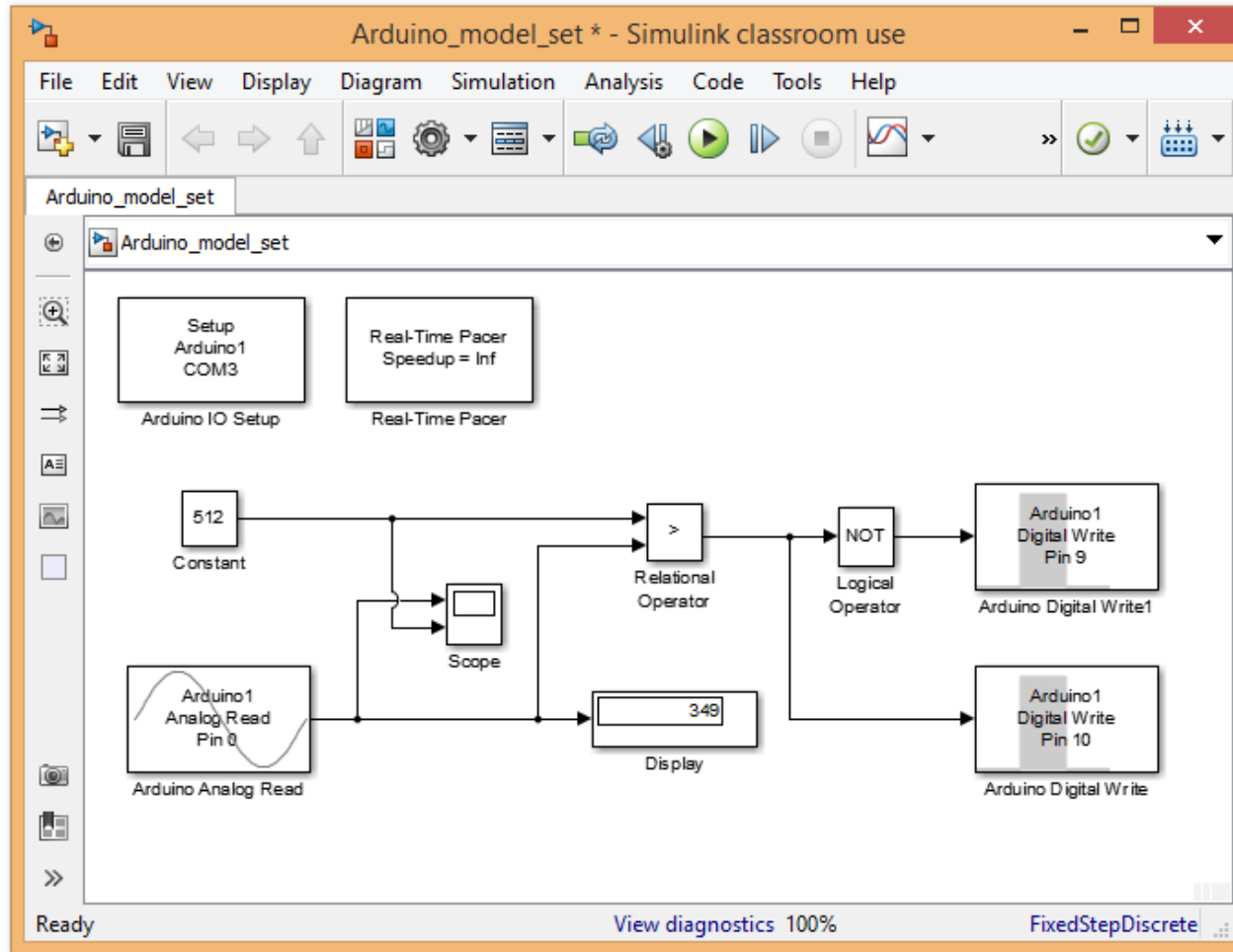
Material video demonstrativ nr. 4

Generarea unor semnale de comandă în funcție de mărimea achiziționată

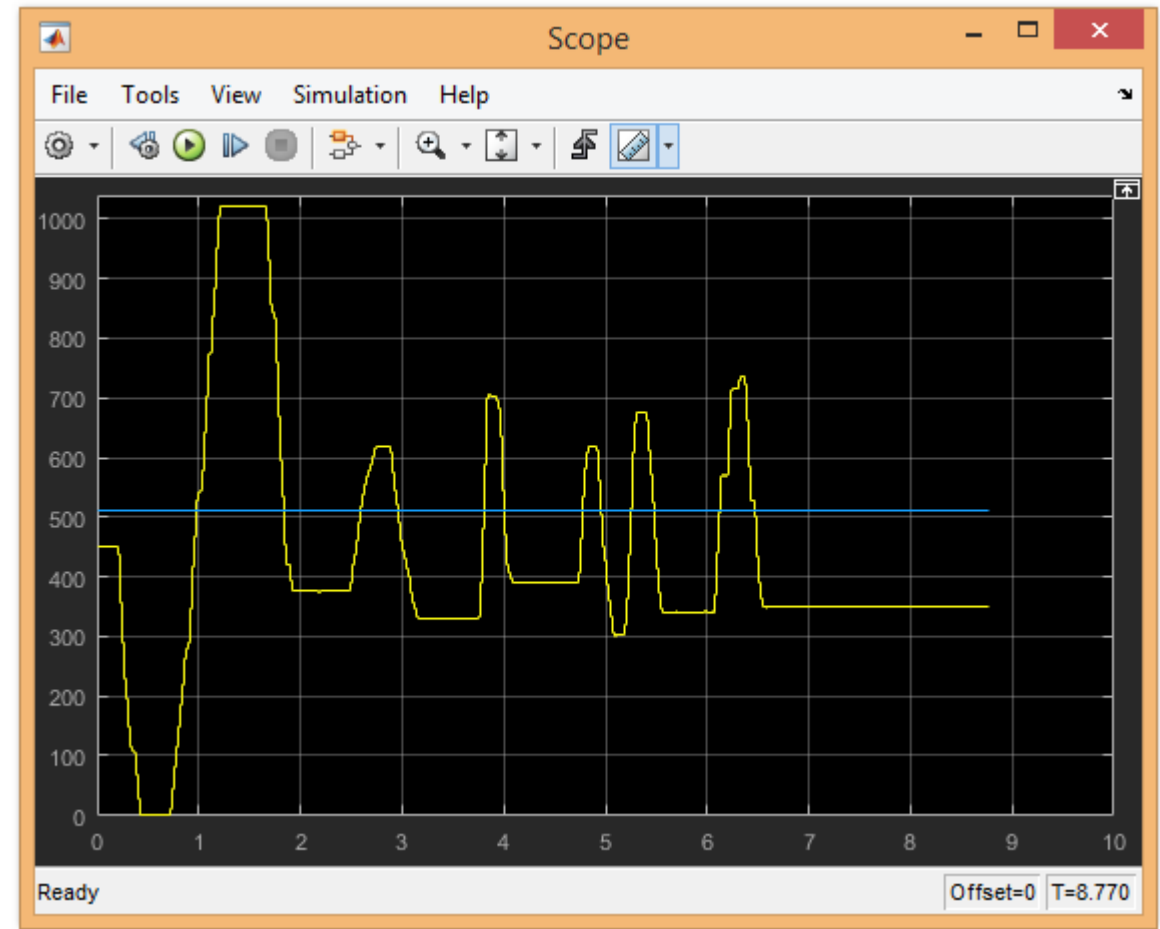
- Blocurile necesare în vederea realizării modelului se găsesc în categoriile următoare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 <p>Setup Arduino1 COM3 Arduino IO Setup</p>	Arduino IO Setup	Arduino IO Library
 <p>Real-Time Pacer Speedup = Inf Real-Time Pacer</p>	Real – Time Pacer	Arduino IO Library
 <p>Arduino1 Analog Read Pin 0 Arduino Analog Read</p>	Arduino Analog Read	Arduino IO Library
 <p>Display</p>	Display	Simulink → Sinks
 <p>Scope</p>	Scope	Simulink → Sinks

Generarea unor semnale de comandă în funcție de mărimea achiziționată



A.



B.

A photograph of a breadboard circuit. A Raspberry Pi Zero is connected to a breadboard. A USB cable is plugged into the board. Several jumper wires are connected to the breadboard. A small LED is lit. The background is a light-colored surface.

Generarea unor semnale de comandă în funcție de mărimea achiziționată

[Material video demonstrativ nr. 5](#)

Concluzii

- ✓ Utilizând mediul Matlab – Simulink în strânsă legătură cu platforma Arduino, pot fi dezvoltate astfel, aplicații complexe de achiziție și control datorită faptului că, în acest mediu de simulare există posibilitatea unei abordări sistemice a programului realizat;
- ✓ Programarea platformei Arduino pe baza blocurilor funcționale, reprezintă esența abordării sistemice;
- ✓ Simularea în timp real permite interacțiunea utilizatorului cu parametrii procesului direct din model, prin intermediul calculatorului gazdă;
- ✓ Procedul de analiză și depanare în timp real a unei bucle de control, poartă denumirea de Rapid Control Prototyping. Utilizarea mediului Matlab – Simulink și platforma Arduino, împreună cu pachetul „Arduino IO”, reprezintă un procedeu similar!

Bibliografie

1. <https://www.pngguru.com/free-transparent-background-png-clipart-uhxim;>
2. https://commons.wikimedia.org/wiki/File:Matlab_Logo.png;
3. <https://www.pngguru.com/free-transparent-background-png-clipart-enxmn;>
4. <https://www.robofun.ro/platforme-de-dezvoltare/arduino-uno-r3.html>
5. <https://www.pngwave.com/png-clip-art-bsctu/download>
6. Ioana – Cornelia GROS, Lucian – Nicolae PINTILIE, Teodor Crișan PANĂ – „SISTEME EMBEDDED ÎN INGINERIE ELECTRICĂ - GHID DE APLICAȚII” – Editura UTPress Cluj – Napoca, 2020 ISBN 978-606-737-431-5:
(<https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/431-5.pdf>);