

## Sisteme de calcul în timp real

- Laboratorul I – Introducerea în domeniul procesării în timp real –

### I. Introducere:

**Un semnal analogic** reprezintă o mărime fizică, electrică sau neelectrică, variabilă sau fixă în timp. În domeniul Ingineriei Electrice, semnalele fizice, de cele mai multe ori, se rezumă la mărimile electrice precum tensiunea sau curentul. Mărimile neelectrice sunt transformate în tensiune variabilă cu ajutorul **traductoarelor**.

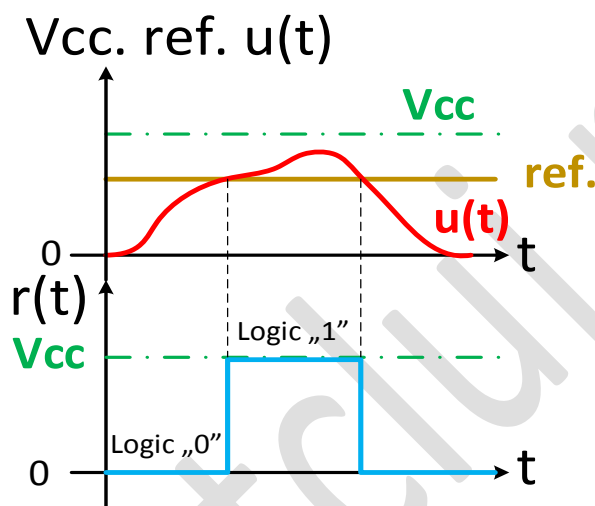


Fig. 1 – Diverse semnale și evoluția acestora în timp:

1. „Vcc” și „ref.” – semnale analogice continue invariabile în timp
2. „u(t)” – semnale analogice discontinue variabile în timp
3. „r(t)” (albastru) - semnale digitale discontinue variabile în timp

**Achiziționarea datelor** furnizate de către un semnal fizic, reprezintă procesul prin care, un sistem de calcul, sau un circuit electronic, preia parametrii semnalului în vederea înregistrării, stocării, prelucrării acestuia. Parametrii esențiali ai unui semnal sunt reprezentați de: amplitudine, frecvență de variație, defazaj și forma de undă. În cazul în care forma de undă a semnalului poate descrie legea de variație a acestuia conform unei funcții matematice bine definite în timp, atunci, semnalul dat, are un **caracter deterministic**, mai precis, variația mărimii poate fi aproximată sau prognozată în timp, la fel și intervalul de timp în care apare mărimea. Pe baza informațiilor preluate de către sistemul de calcul și achiziție, se pot **furniza valori noi de semnal**, conform **legilor de variație prescrise** în aplicația dată. Acest procedeu poartă numele de **procesare numerică sau digitală**.

**Prelucrarea numerică** a unui semnal analogic reprezintă de asemenea suma tuturor operațiilor de alterare sau corectare a parametrilor acestuia. În cadrul sistemelor moderne de achiziție și furnizare de semnal, aceste operații fundamentale, sunt realizate de către două etaje electronice, anume:

- convertorul analog - digital (eng. Analog to Digital Converter – ADC);
- convertorul digital – analog (eng. Digital to Analog Converter – DAC);

Cu ajutorul convertorului analog – digital, pot fi achiziționate semnalele analogice de către sistemul de calcul, iar prin intermediul convertorului digital analog, pot fi generate noi semnale analogice sub formă de tensiuni variabile sau fixe în timp.

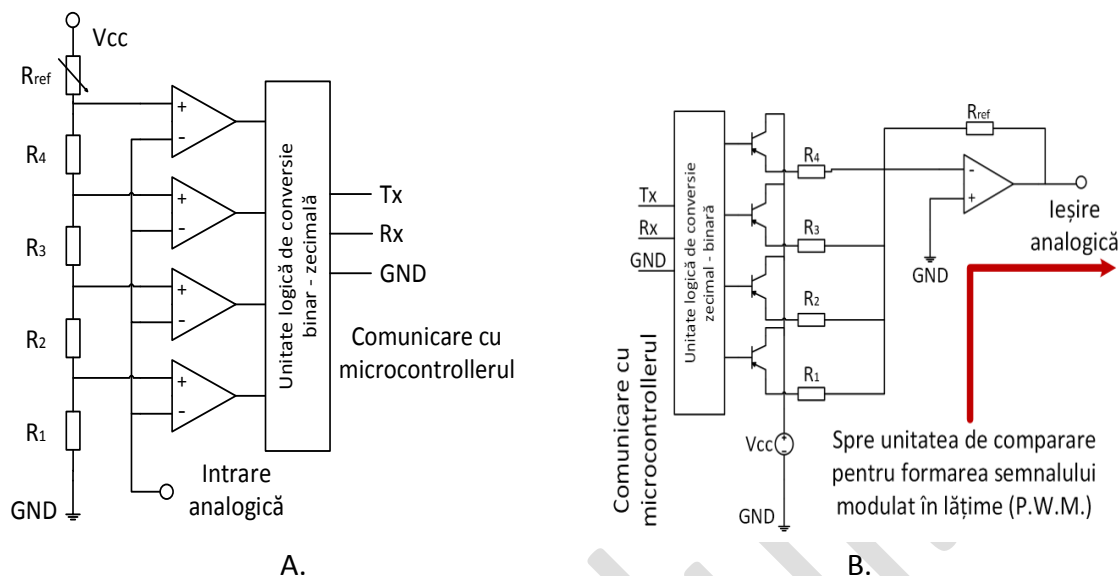


Fig. 2 – Schema electronică a:

A. Convertorului analog – digital - B. Convertorului digital – analog

Există situația în care, convertorul digital – analog furnizează mărimea sa de ieșire ca și mărime de referință a **unității de generare a semnalelor modulate** (eng. Pulse Width Modulation). Astfel, în cadrul etajului electronic de generare a semnalului modulat în lățime are loc o **comparație** dintre **unda purtătoare** (de formă triunghiulară) și **unda modulatorie** (DAC), care reprezintă **tensiunea variabilă furnizată de către convertorul digital – analog**. Semnalul rezultat al acestui etaj electronic, va fi **dreptunghiular**, având **lățimea impulsului variabilă** în funcție de **nivelul tensiunii furnizată de către convertorul digital analog**.

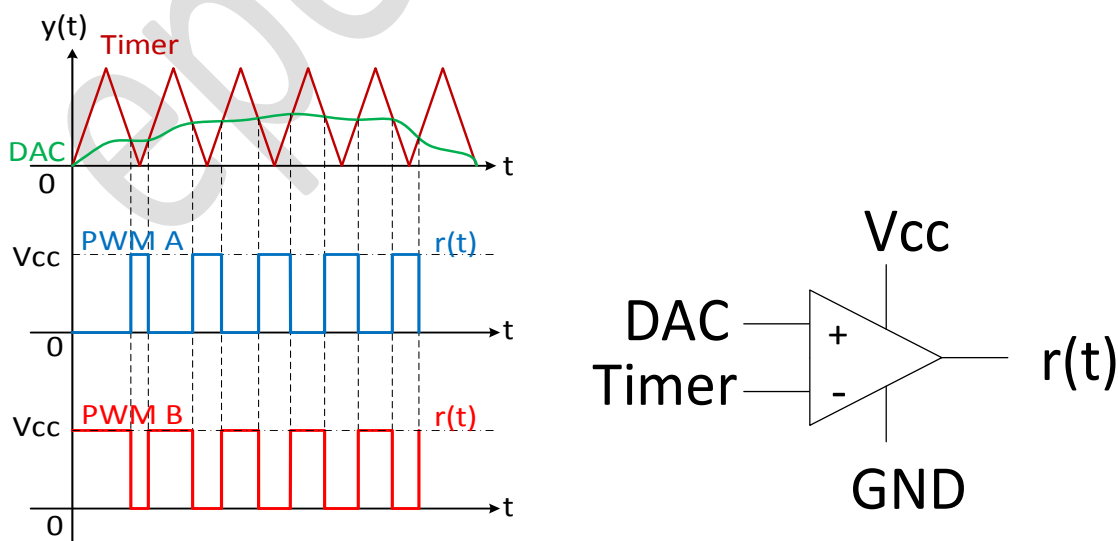


Fig. 3 – Generarea semnalului modulat în lățime (eng. Pulse Width Modulation – PWM)

Semnalul triunghiular sau unda purtătoare (eng. Carrier Wave - CW), este generată de către un circuit digital electronic numit numărător (eng. Counter), care se incrementează până la valoarea rezoluției acestuia (ex. un numărător pe 8 biți – are 255 de valori). Frecvența de convergență a numărătorului este dată de rezoluția sa și de frecvența oscilatorului (cristalului de cuarț). Revenirea la zero, se face prin resetarea numărătorului.

**Platforma Arduino**, poate fi utilizată pentru prelucrarea semnalelor, deoarece beneficiază de etajele menționate mai sus, având următoarele specificații:

- microcontroller ATmega 328P – având frecvența de lucru 16 [MHz];
- convertor analog – digital având rezoluția de 10 biți – 8 intrări analogice;
- convertor digital – analog având rezoluția de 8 biți – 6 ieșiri digitale PWM;
- comparatoare – 13 intrări și ieșiri digitale de uz general;
- conexiune USB – Serial pentru a realiza legătura între platformă și calculatorul gazdă;
- memorie flash 32 [kB], memorie RAM 2 [kB];

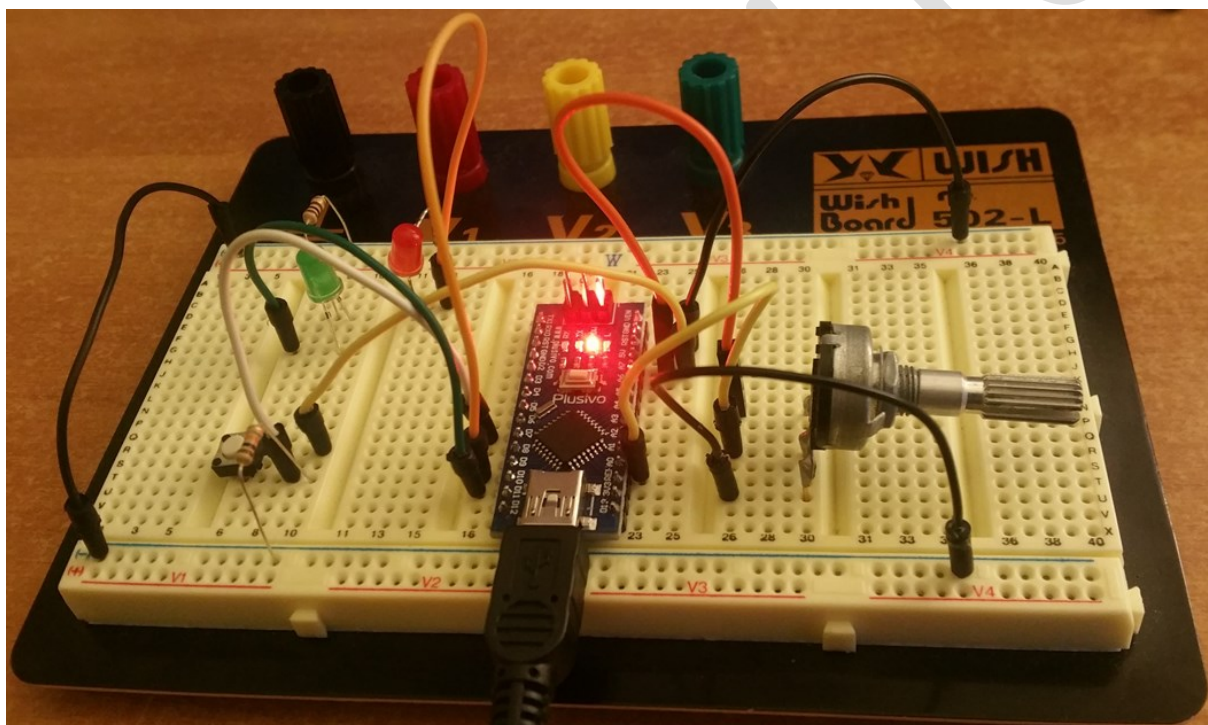


Fig. 4 - Platforma de dezvoltare Arduino Nano montată pe o plăcuță de test

Prin intermediul platformei Arduino, conectată la calculatorul gazdă, se pot realiza aplicații de achiziție, monitorizare și control în timp real. Toate procesele enumerate pot fi asistate de calculator, atâta timp, cât între platformă și calculator există un **protocol de comunicație** care permite **manipularea perifericelor platformei** (ex. intrări / ieșiri digitale, intrări analogice etc.). Totodată, este necesar, ca datele preluate, să poată fi introduse într-un **mediu de simulare și testare**, în vederea **validării algoritmului de control și achiziție în timp real**, prin **interacțiunea directă** cu aparatura atașată la platforma Arduino (ex. un convertor electronic de putere sau un sistem de acționare electrică).

## II. Implementare:

Mediile de simulare și testare precum MathWorks Matlab – Simulink, National Instruments LabVIEW, Altair / SolidThinking Embed (VisSIM), PLECS – Plexim sau SciLab, pot **interacționa în mod direct, în timpul simulării cu perifericele unei platforme de achiziție și furnizare / generare de semnal**. Pentru a permite comunicarea în timp real a modelului conceput în oricare dintre mediile enumerate anterior, este necesar ca:

- în cadrul mediului de simulare să existe pachete de instrumente aferente platformei de lucru (ex. pachetul ArduinoIO al mediului Matlab – Simulink sau pachetul LIFA\_CORE și LIFA\_BASE ale mediului LabVIEW – eng. LabVIEW Interface for Arduino);
- pe platforma de lucru să ruleze un program / aplicație prin intermediul căreia să poată fi accesate perifericele de achiziție și furnizare a semnalului (eng. LIFA firmware, ADIO.pde). Acest program are rolul de a prelua „comenzile” furnizate de către mediul de simulare.

**Mediul Matlab – Simulink** – reprezintă unul dintre mediile specializate în domeniul conceperii strategiilor de control automat. În cadrul acestuia, există o gamă largă și variată de platforme și dispozitive periferice, care permit achiziția și furnizarea semnalelor. În categoria dispozitivelor compatibile cu acest mediu de simulare, intră și platforma Arduino. Prin intermediul pachetului „Arduino IO”, mediul Matlab – Simulink poate interacționa cu intrările și ieșirile digitale / analogice ale platformei Arduino.

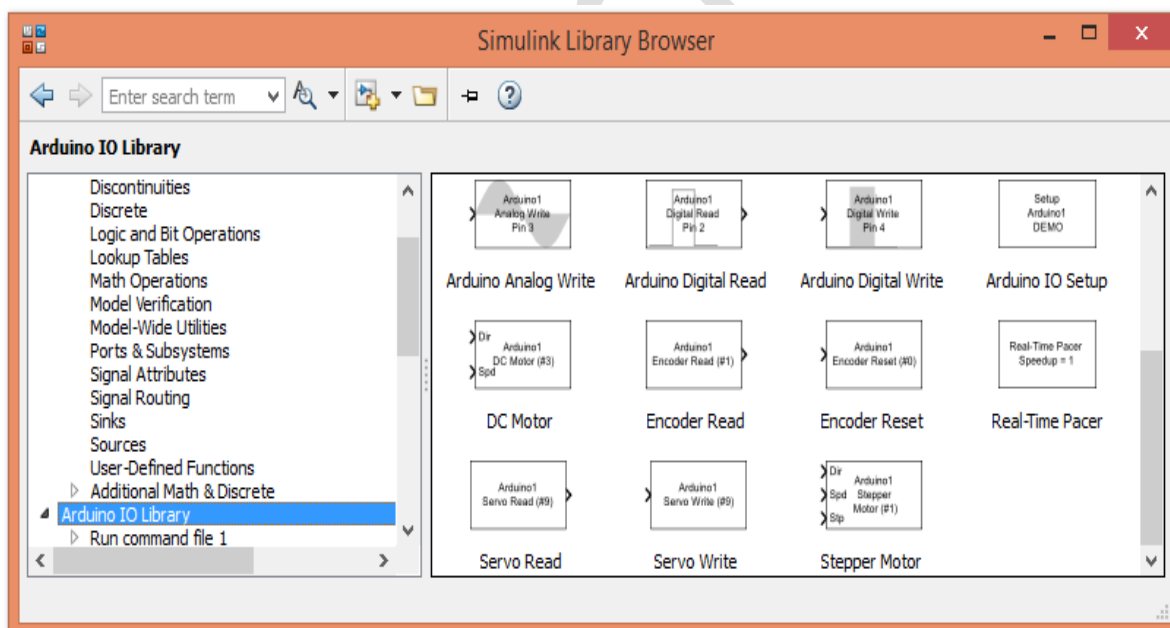


Fig. 5 – Paleta de instrumente „Arduino IO”

În cadrul paletii de instrumente „Arduino IO” există toate funcțiile specifice platformei, precum: „AnalogRead()”, „AnalogWrite()”, „DigitalRead()”, „DigitalWrite()” etc. Pe baza instrumentelor puse la dispoziție de către pachetul „Arduino IO”, pot fi dezvoltate modele Simulink prin intermediul cărora, dezvoltatorul poate controla aplicația prin modificarea parametrilor în timpul simulării.

Totodată, la fel de important ca și pachetul „Arduino IO” este și programul „adio.pde” care necesită a fi încărcat în memoria platformei Arduino, în vederea stabilirii comunicației dintre modelul Simulink și perifericele platformei.

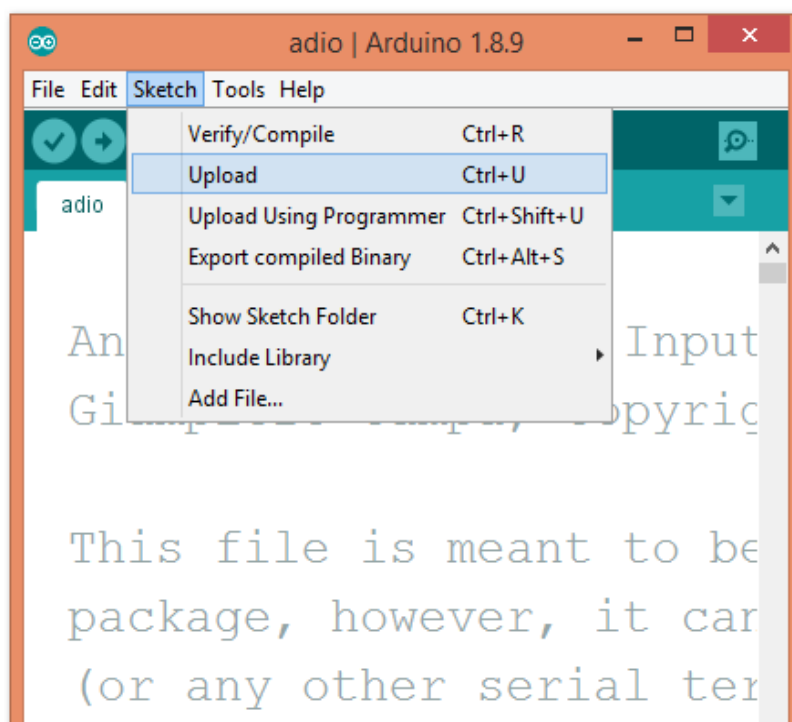


Fig. 6 – Încărcarea programului de comunicație „adio.pde”

În urma parcurgerii etapelor descrise anterior, între calculatorul gazdă și platforma Arduino, se va crea un canal de comunicație Serial bidirecțional, permițând mediului Simulink atât să preieie semnalele achiziționate, cât și să furnizeze alte semnale prin intermediul plăcuței de dezvoltare Arduino.

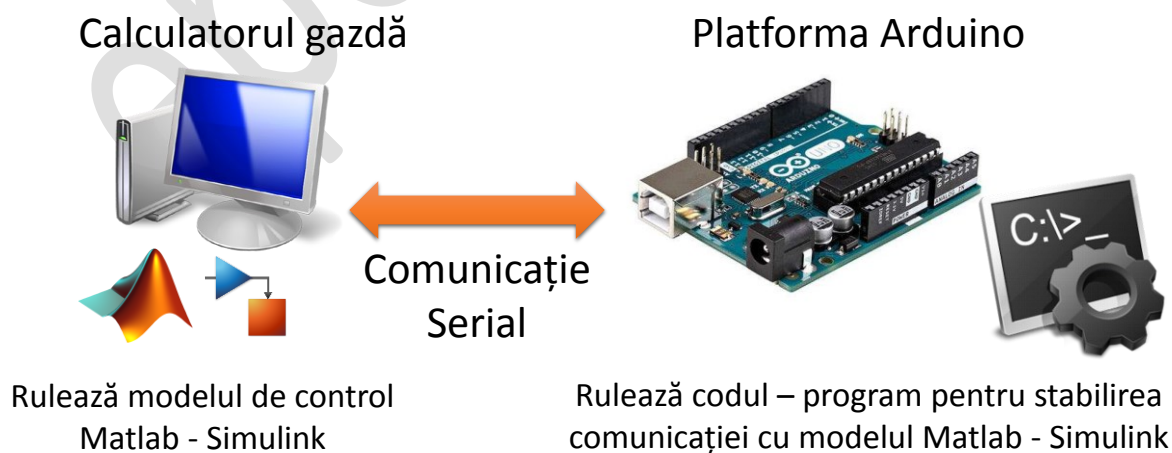


Fig. 7 – Stabilirea comunicației dintre mediul Matlab Simulink și platforma Arduino

În vederea realizării modelelor Simulink de interacțiune cu platforma Arduino, este necesară parcurgerea următoarelor etape inițiale:

**A. Introducerea blocurilor esențiale în vederea stabilirii legăturii dintre model și platformă:**

„Arduino IO Setup” – pentru a inițializa conexiunea Serial. În cadrul acestui bloc se va stabili portul de comunicație Serial (ex. COM „x”);

„Real – Time Pacer” – pentru a sincroniza execuția modelului cu execuția programului de pe platforma de dezvoltare. În cadrul acestui bloc, se va stabili viteza de execuție a modelului.

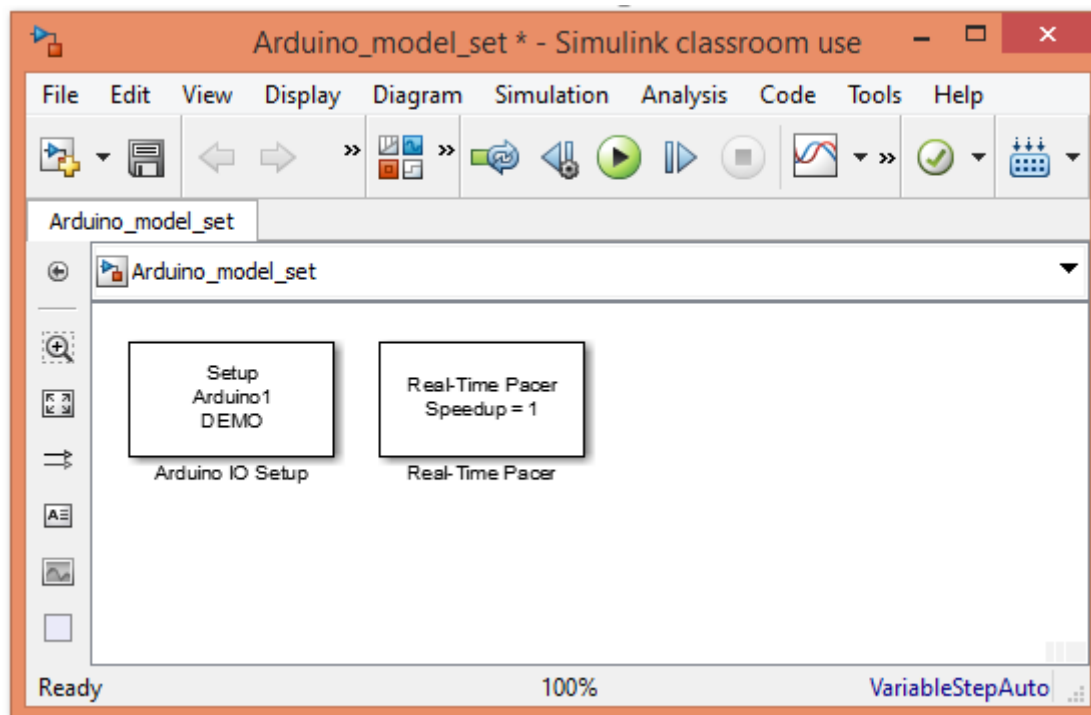


Fig. 8 – Introducerea blocurilor pentru inițializare

**B. Particularizarea timpului total de simulare, a timpului de eșantionare și a metodei de rezolvare a modelului matematic:**

În cadrul ferestrei de particularizare a parametrilor de simulare (din meniul „Simulation” – „Model Configuration Parameters”), se va stabili timpul total de simulare ca și infinit („inf”), metoda de rezolvare a modelului matematic:

- Type „Fixed-step”;
- Solver „discrete (no continuous states)”;
- timpul de eșantionare „Fixed-step size (fundamental sample time): „1e-3”;

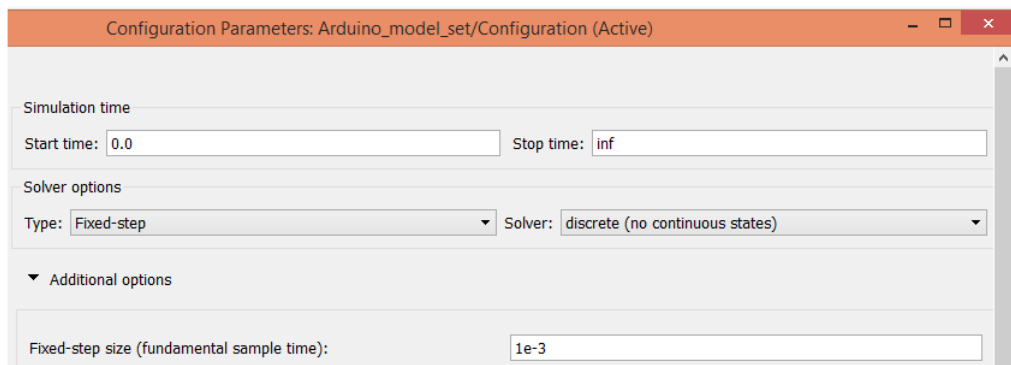



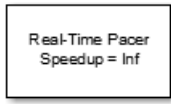
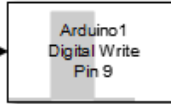




Fig. 9 – Stabilirea parametrilor de simulare

În urma particularizărilor efectuate, se vor putea implementa diverse aplicații de comandă și control asistate de calculator, precum:

- furnizarea semnalelor digitale pe baza ieșirilor digitale;
- achiziția semnalelor digitale de pe baza intrărilor digitale;

#### 1. Furnizarea semnalelor digitale pe baza ieșirilor digitale:

Pe baza acestui model, se va realiza semnalizarea cu un șir de opt diode electroluminiscente, mai precis balearea succesivă a șirului. Aceeaș aplicație poate fi utilizată în vederea furnizării comenzii unui motor electric pas cu pas. Pentru implementarea modelului următoarele blocuri sunt necesare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 Arduino IO Setup	Arduino IO Setup	Arduino IO Library
 Real-Time Pacer	Real – Time Pacer	Arduino IO Library
 Arduino Digital Write	Arduino Digital Write	Arduino IO Library
 Display	Display	Simulink → Sinks
 Scope	Scope	Simulink → Sinks
 Counter Limited	Counter Limited	Simulink → Sources
 Rate Transition	Rate Transition	Simulink → Signal Attributes

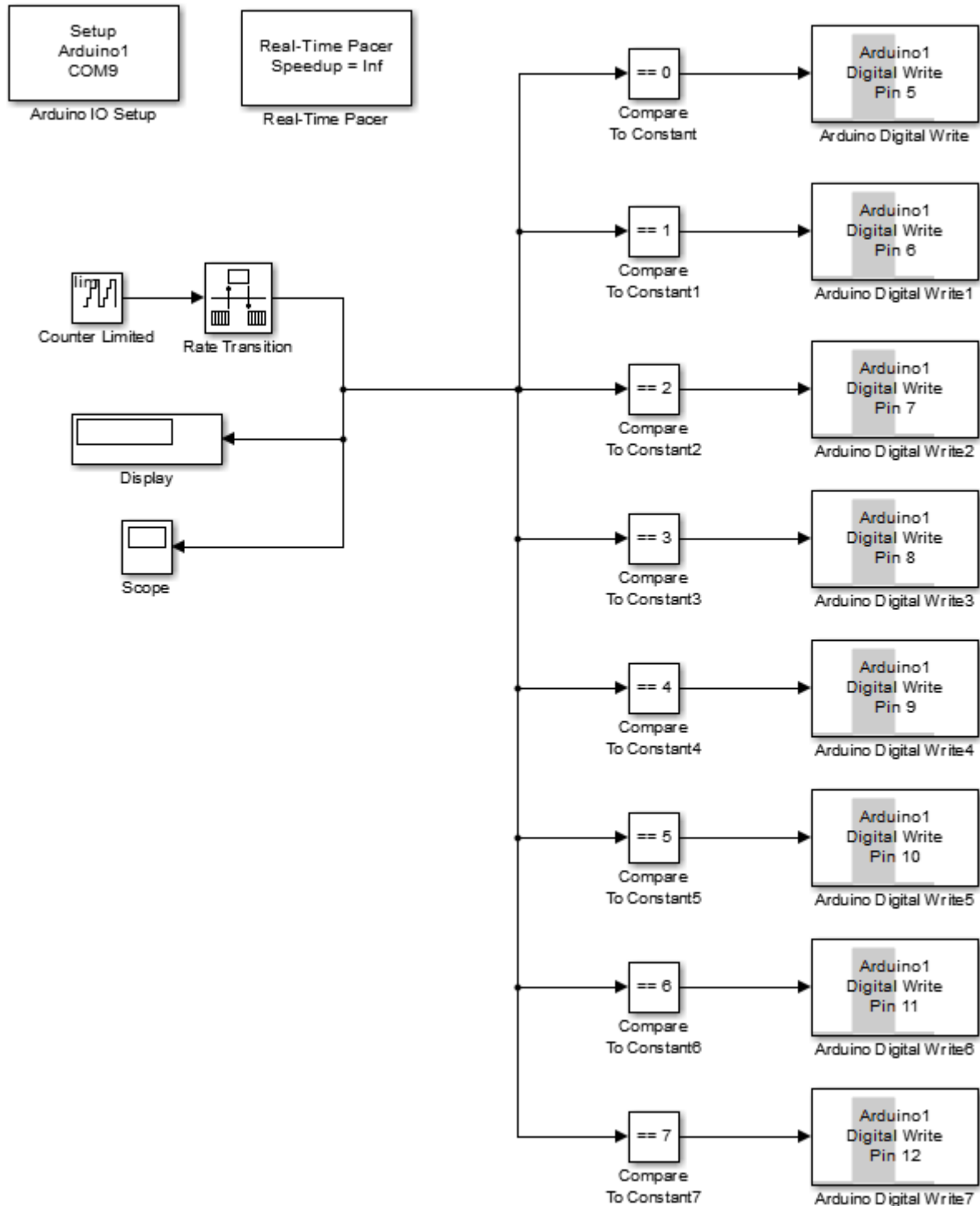
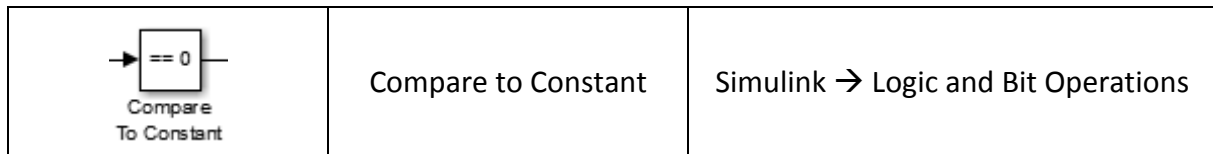


Fig. 10 – Modelul Simulink pentru furnizarea semnalelor digitale



În cadrul modelului, rolul blocului „Limited Counter” (numărător cu limitare), este de a incrementa numărul ieșirii digitale pe care urmează a fi furnizat semnalul digital. Șirul de comparatoare, verifică valoarea numerică și comută intrarea digitală când valoarea numerică este egală cu referința dată.

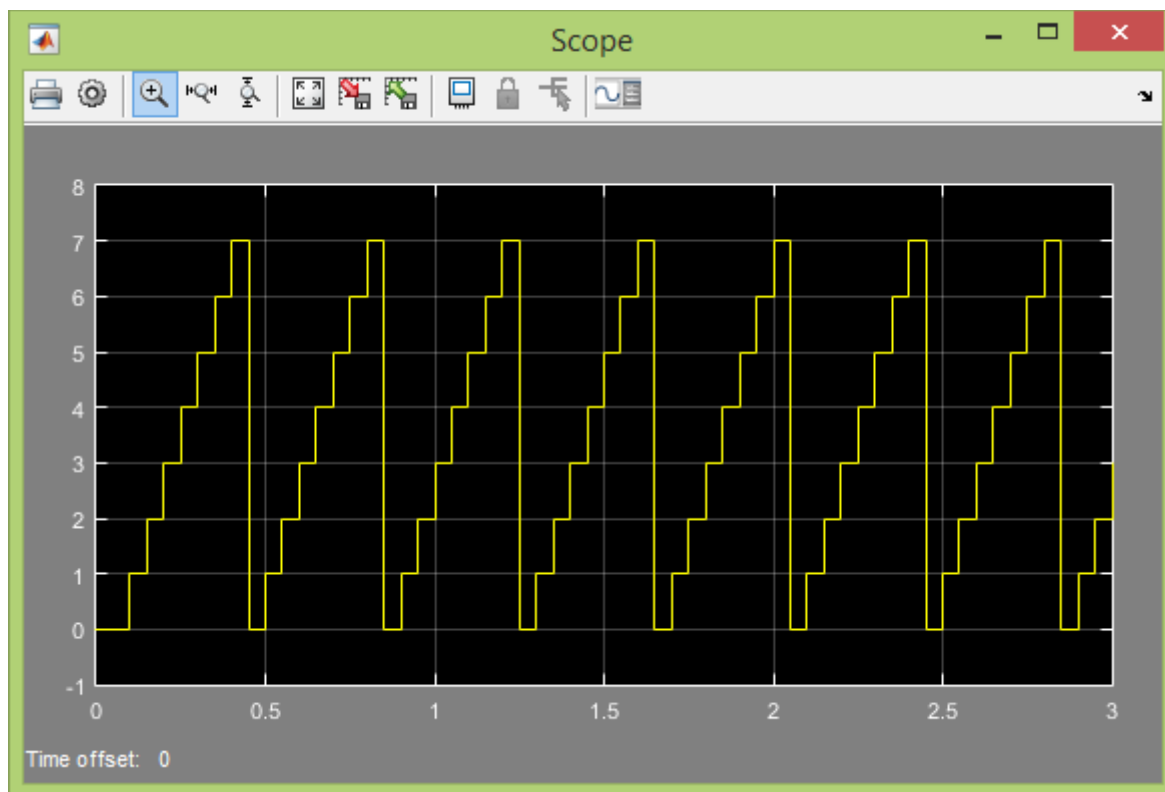


Fig. 11 – Semnalul furnizat de numărător

## 2. Achiziția semnalelor digitale de pe baza intrărilor digitale:

În cadrul acestei aplicații, baleerea șirului de diode se realizează fie la apăsarea repetată a butonului fizic atașat la intrarea digitală 13, fie la comutarea repetată a celui de-al doilea întreruptor virtual „Manual Switch”. Primul întreruptor virtual „Manual Switch” comută modul de lucru al aplicației (ex. modul software – comanda realizată din interfața grafică; modul hardware – comanda realizată din butonul fizic atașat la intrarea digitală 13). Blocurile „Rate Transition” asigură sincronizarea porțiunilor din modelul matematic (trecerea de la un timp de execuție mai mic la unul mai mare). Blocul „Data Type Conversion” asigură trecerea de la un tip de date la altul (ex. numere întregi sau fracționare). Bucla compusă din blocurile „Sum” + „Memory” + „Switch” + „Constant”, asigură incrementarea și resetarea valorii numerice, care controlează poziția diodei aprinse din șirul de opt diode electroluminiscente. Blocul „Edge Detector” furnizează un impuls digital la sesizarea unui front digital crescător. Comportamentul aplicației actuale este similar cu cel al aplicației precedente, doar că, în cazul de față, baleerea șirului se realizează în mod manual.

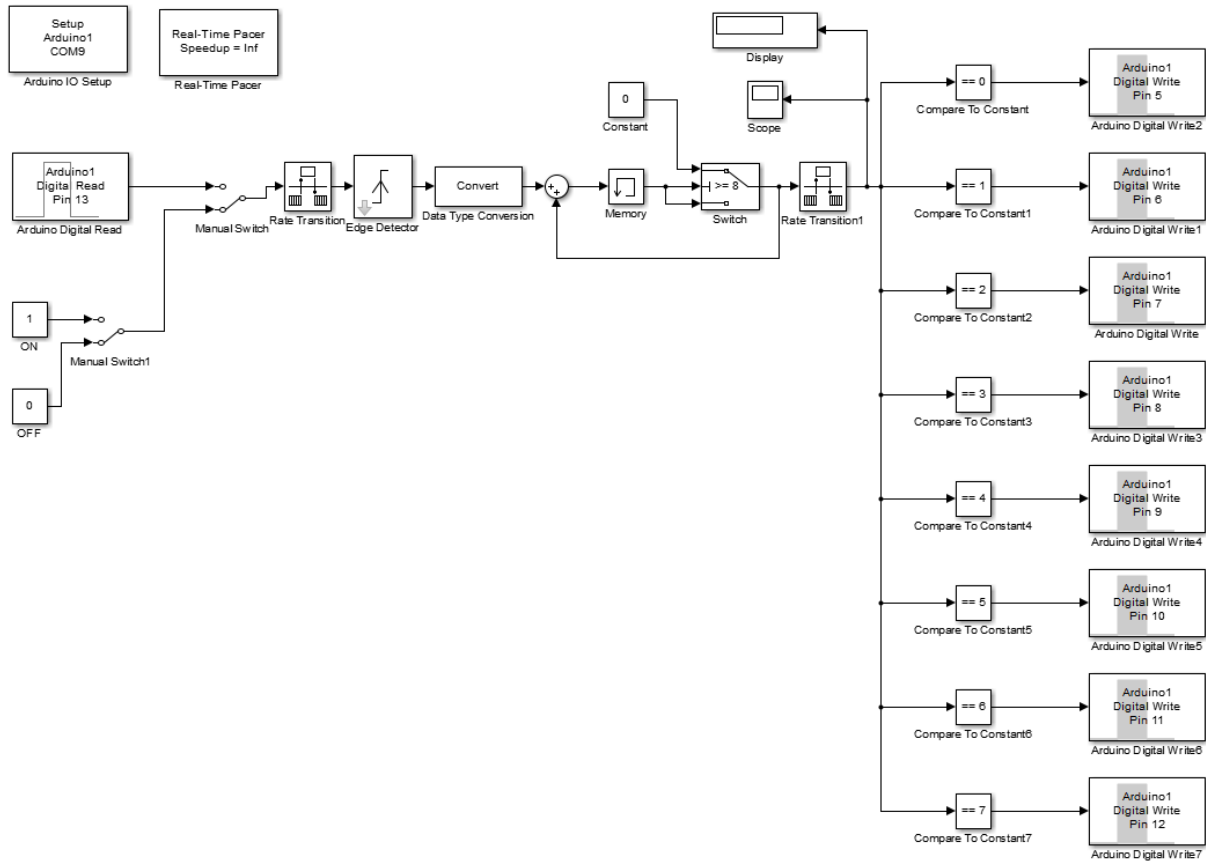
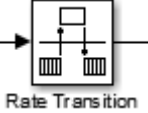
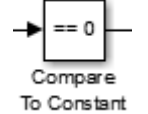
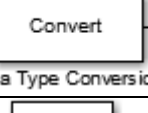
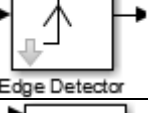
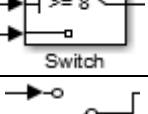
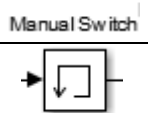
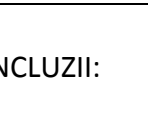


Fig. 12 – Modelul pentru achiziționarea și furnizarea unui semnal digital

În vederea realizării acestui model următoarele blocuri sunt necesare:

Simbol bloc	Denumire bloc	Categoria din care face parte
 Setup Arduino1 COM3 Arduino IO Setup	Arduino IO Setup	Arduino IO Library
 Real-Time Pacer Speedup = Inf Real-Time Pacer	Real – Time Pacer	Arduino IO Library
 Arduino1 Digital Write Pin 9 Arduino Digital Write	Arduino Digital Write	Arduino IO Library
 Display	Display	Simulink → Sinks
 Scope	Scope	Simulink → Sinks
 Counter Limited	Counter Limited	Simulink → Sources

 Rate Transition	Rate Transition	Simulink → Signal Attributes
 Compare To Constant	Compare to Constant	Simulink → Logic and Bit Operations
 Data Type Conversion	Data Type Conversion	Simulink → Signal Attributes
 Edge Detector	Edge Detector	Simscape → SimPowerSystems → Specialized Technology → Control & Measurements → Logic
 Switch	Switch	Simulink → Signal Routing
 Manual Switch	Manual Switch	Simulink → Signal Routing
 Memory	Memory	Simulink → Discrete

### III. CONCLUZII:

- Cu ajutorul mediului Matlab – Simulink și platforma Arduino, pot fi dezvoltate aplicații de comandă și control în timp real asistate de calculator, fără a fi necesară re-programarea platformei Arduino și re-inscripționarea memoriei acestuia;
- Varietatea paletelor de instrumente ale mediului Matlab – Simulink facilitează implementarea diversificată a soluțiilor pentru problemele de Inginerie Electrică;
- Gradul de funcționalitate al platformei Arduino este mult mai ridicat, atunci când se utilizează în cadrul unui mediu de simulare și testare precum Matlab - Simulink;

### IV. BIBLIOGRAFIE:

1. Michigan University – Control Tutorials and Matlab – Simulink – ArduinoIO:

[http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities\\_IOPack](http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_IOPack)

2. MathWorks – Legacy MATLAB and Simulink Support for Arduino:

<https://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino>

3. Biblioteca UTCLUJ – Editura UTPress Cluj-Napoca – „Sisteme Embedded în Inginerie Electrică” – Ioana – Cornelia GROS, Lucian – Nicolae PINTILIE, Teodor Crișan PANĂ; ISBN: 2020 ISBN 978-606-737-431-5; <https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/431-5.pdf>

Realizat de: Asist. drd. Pintilie Lucian - Nicolae  
Pentru disciplina: „Sisteme de calcul în timp real”  
Adresă de e-mail: [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)

