Laborator 7 – Utilizarea intrarilor analogice

Microcontrollerele sunt capabile sa detecteze semnale binare, 0 sau 1, cum este de exemplu starea unui buton (apasat sau ridicat). Aceste semnale se numesc semnale digitale. Cand un microcontroller este alimentat de la 5 V el intelege valoarea tensiunii de 5V ca si 1 logic si 0V ca si 0 logic. Cu toate acestea noi avem nevoie sa masuram si altfel de semnale in lumea reala, semnale intermediare valorilor extreme (de exemplu, 2.57 V). Aceste semnale contin informatie relevanta in nivelul tensiunii lor, si poarta numele de semnale analogice. Un ADC (convertor analog la digital) converteste un semnal analogic la un numar. Prin intermediul acestui dispozitiv avem posibilitatea de a interfata tot felul de periferice la microcontrollerul nostru si de a masura informatiile analogice din jurul nostru. Nu toti pinii de pe arduino pot face astfel de conversii. Pinii care pot fi folositi impreuna cu senzori analogici sunt pinii care au un ‚A' im fata numelui lor pe placa. In figura 1 se observa pinii analogici de pe Arduino mega incercuiti cu rosu, iar in figura 2 se poate observa locatia pinilor analogici de pe Arduino UNO.

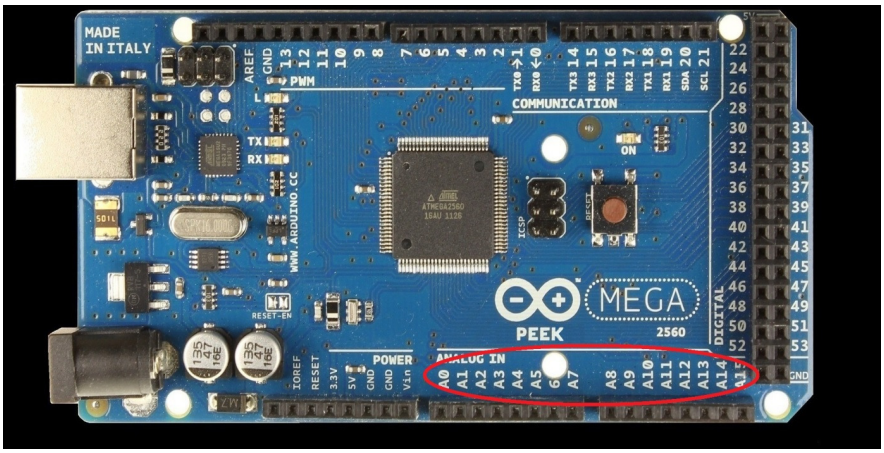


Fig 1 Pinii Analogici din Arduino Mega

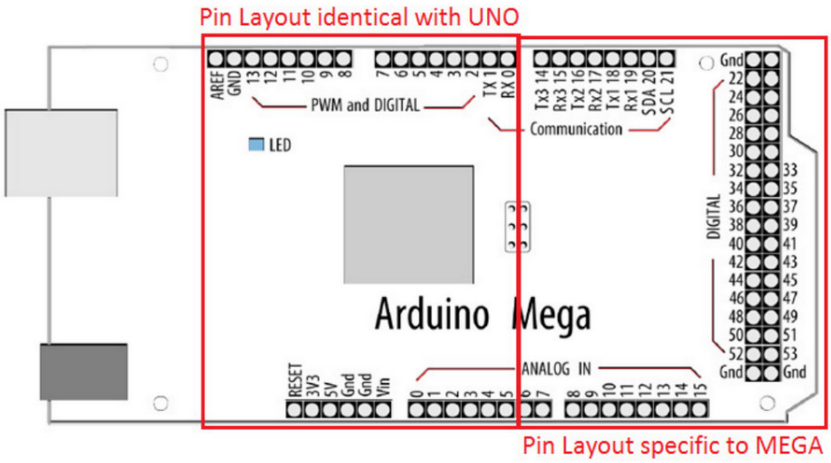


Figure 2. Arduino UNO and Mega pin layout

ADC-urile pot varia mult intre diferite tipuri de microcontrollere. Spre exemplu pe Arduino Mega

avem ADC-uri care au o precizie de 10 biti. Acest lucru inseamna ca aceste convertoare pot detecta pana la 1024 valori. Exista si adc-uri care au rezolutie de 8 sau 16 biti. ADC-ul intoarce o valoare ratiometrica. Asta inseamna ca adc-ul considera 5 V ca 1023 si orice valoare mai mica ca si 5V va fi construita ca si o fractie intre 5V si 1023 (2).

$$\frac{5}{1023} = \frac{Citirea\ de\ la\ ADC}{Valoare\ tensiune\ masurata} \qquad (2)$$

Pinii analogici de pe arduino pot fi folositi si ca pini de tip I/O general (GPIO), ei avand aceleasi functionalitati ca si pinii digitali, in cazul in care cei oferiti de placa nu sunt suficienti. Pinii analogici de pe placa au la randul lor rezistente pull up. Sintaxa arduino pentru activarea acestor rezistori este similara cu cea de la pinii digitali: digitalWrite(A0, HIGH);//pinul A0 fiind setat ca input. Pentru citirea unei valori de la un senzor se foloseste comanda analogRead().

!!Atentie
Comanda analogRead() nu va functiona corect daca pinul de pe care incercati sa cititi a fost setat ca si pin de iesire.

Datasheetul ATMEGA mai avertizeaza despre folosirea senzorilor analogici in pozitii apropiate. In momentul in care realizam o citire, daca executam rapid o comutare intre pozitiile pe care dorim sa le citim, se vor introduce zgomote in citirea semnalului. Se recomanda folosirea unui mic delay() inaintea citirii unei valori analogice consecutive. O alta functie importanta legata de utilizarea senzorilor analogici o reprezinta analogReference().

Pentru a masura o tensiune analogica trebuie sa existe o tensiune de referinta fata de care sa o raportam. Functia analogReference() seteaza tensiunea maxima cu care sa efectuam masuratoarea.

Configuratii posibile pentru aceasta referinta sunt :
- DEFAULT – foloseste tensiunea de referinta a placii (5V pentru placile Arduino care folosesc tensiune de 5V sau 3.3 V pentru placi cu tensiune de referinta de 3.3 V).
- INTERNAL – seteaza o tensiune de referinta de 1.1 V. Poate fi folosita pe placile care contin ATMEGA 328 spre exemplu dar nu poate fi folosita pe ATMEGA2560.
- INTERNAL1V1 – Tensiune de referinta de 1.1 V folosita pe placile MEGA
- INTERNAL2V56 – tensiune de referinta de 2.56 V aceasta e valabila doar pe placile MEGA
- EXTERNAL – tensiune aplicata pinului AREF. Aceasta tensiune este intre 0 si 5V.

Folositi tensiunea de referinta cea mai buna pentru senzorul analogic utilizat. Ideal, referinta trebuie sa fie valoarea maxima pe care o poate genera senzorul analogic, pentru a obtine o rezolutie cat mai buna la conversia in digital. Daca referinta este mai mica decat valoarea maxima pe care o poate avea semnalul, tensiunea ce depaseste valoarea de referinta nu va putea fi cuantizata, ea generand la digitizare valoarea de saturatie 1023.

Dureaza aproximativ 100 microsecunde (0.0001 s) pentru a citi o intrare analogica, astfel incat rata maxima de citire este 10000 valori pe secunda.

Nu folositi o tensiune de referinta externa negativa (<0V) sau mai mare de 5V pe pinul AREF!
Daca folositi o tensiune externa de referinta, configurati referinta ca externa apeland analogReference() inainte de a apela functia analogRead().

Exemplul 1

In exemplul urmator vom citi valoarea de la un potentiometru liniar. Valoarea citita va fi afisata pe LCD. Circuitul pentru acest exemplu este ilustrat in figura 3 (legati pinii VCC si GND ai potentiometrului la +5V si GND de pe placa, si semnalul de iesire la un pin analogic).

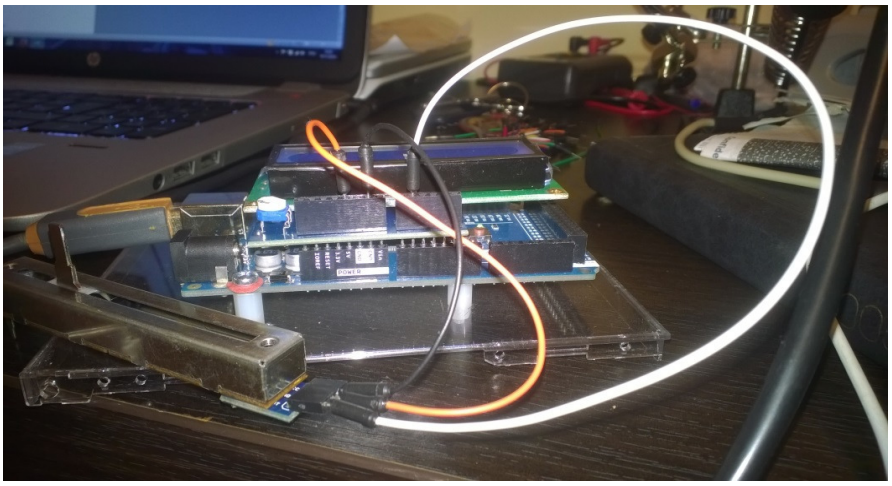
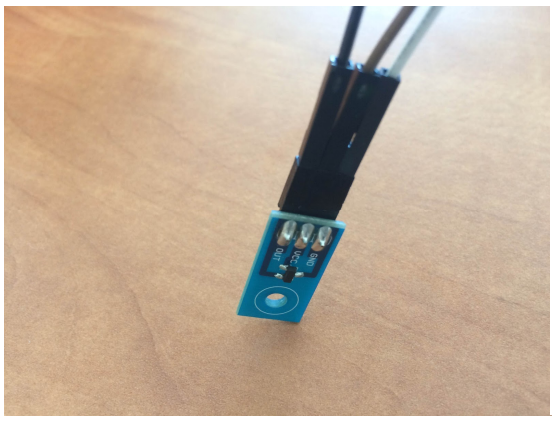


Fig 3 Conexiunile pentru exemplul cu senzorii analogici

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup()
{
        analogReference(DEFAULT); //setarea tensiunii de referinta la tensiunea default
        lcd.begin(16, 2); //initializarea LCD ului
        lcd.setCursor(0,0);
        lcd.print("Cititi senzor");
        pinMode(A1, INPUT); // setarea pinului analogic A1 ca si pin de input
        digitalWrite(A1, HIGH); //activarea rezistorului pull up pentru pinul A1
}
void loop()
{
        int val = analogRead(A1); //citirea valorii analogice
        lcd.setCursor(0,1);
        lcd.print(val);
}
```

Exemplul 2 citire temperatura:

Senzor de temperatura folosind LM50 http://www.ti.com/lit/ds/symlink/lm50.pdf



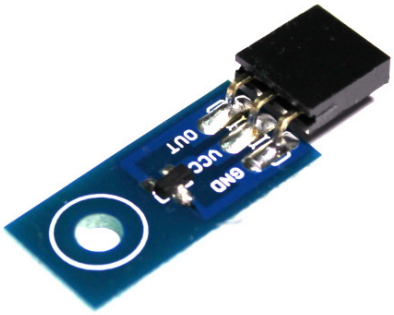


Figura 4 Senzorul de temperature utilizat

Caracteristici:

• Iesire liniara +10.0 mV/°C = 0.01V/°C

• Domeniu de temperaturi –40°C ... +125°C

• Deplasament constant +500 mV pentru citirea temperaturilor negative

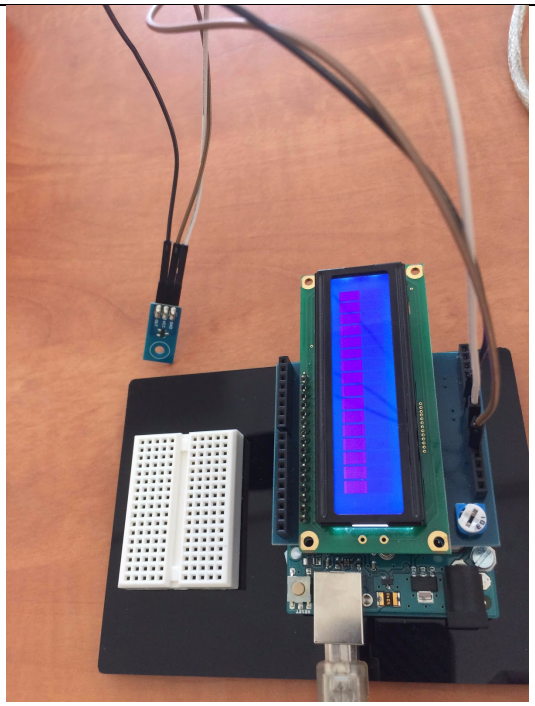


Figura 5. Montajul pe care il aveti de realizat pentru exemplul 2

Examplul 2 – Citire temperatura de la senzor, face media a 10 citiri consecutive, si trimite catre PC

Realizati montajul din figura 5 iar apoi testate codul de mai jos.

```
float resolutionADC = .0049 ; // rezolutia implicita (pentru referinta 5V) = 0.049 [V] / unitate
float resolutionSensor = .01 ; // rezolutie senzor = 0.01V/°C

void setup() {
        Serial.begin(9600);
}
void loop(){
        Serial.print("Temp [C]: ");
        float temp = readTempInCelsius(10, 0); // citeste temperatura de 10 ori, face media
        Serial.println(temp); // afisare
        delay(200);
}
float readTempInCelsius(int count, int pin) {
 // citeste temperatura de count ori de pe pinul analogic pin
        float sumTemp = 0;
        for (int i =0; i < count; i++) {
                int reading = analogRead(pin);
                float voltage = reading * resolutionADC;
                float tempCelsius = (voltage - 0.5) / resolutionSensor ;
                // scade deplasament, converteste in grade C
                sumTemp = sumTemp + tempCelsius; // suma temperaturilor
        }
        return sumTemp / (float)count; // media returnata
}
```

Utilizarea ADC-urilor folosind registre AVR

Precum am vazut mai sus senzorii genereaza o tensiune variabila, dependent de o marime fizica masurata (nivelul de lumina, nivelul de temperatura, elongatia, flexibilitatea, sunetul etc…).

Majoritatea MCU-rilor AVR contin ADC incorporat.

AVR-ul Atmega 2560 contine un singur ADC pe 10 biti cu o rata de esantionare maxima de 15kS/s la rezolutia maxima. Metoda de esantionare folosita este successive approximation type ADC. Atmega 2560 ofera posibilitatea de a selecta din 16 pini analogici. Un anumit pin analogic este selectat printr-un process de multiplexare. Domeniul de variatie al intrarii este 0 … Vcc unde Vcc este de obicei 5V.

Masuratori diferentiale ale tensiunii pot fi facute folosind patru canale diferentiale independente. O

schema simplificata a subsistemelor ADC-ului sunt ilustrate in figura 6.

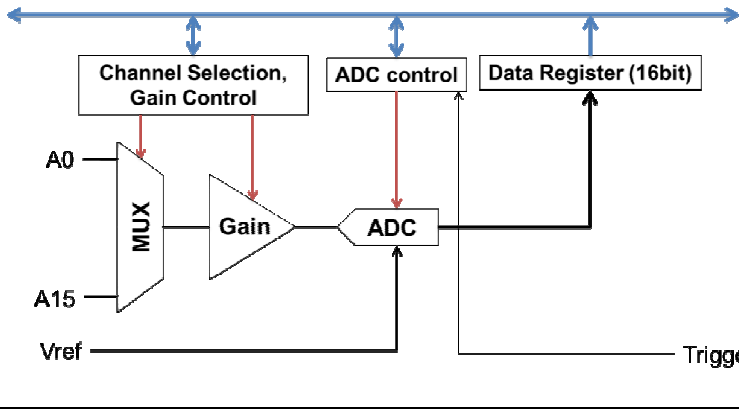


Figura 6. Metoda simplificata de a ilustra procesul de achizitie de la ADC-uri

ADC-ul poate fi operat in urmatoarele 3 moduri:

1. Single Conversion: inceputa scriind un 1 logic in bitul de incepere conversie al ADC-ului
2. Triggered conversion: conversia e inceputa in momentul in care avem un front crescator la triggerul setat
3. Free running: Urmatoarea conversie incepe imediat dupa ce conversia precedenta a fost terminata.

Inainte de a folosi ADC e necesar sa selectam tensiunea de referinta si frecventa clockului ADC-ului. Optiunile pentru tensiunea de referinta sunt ilustrate in tabelul 1 si sunt setate utilizand bitii 6 si 7 in registrul ADCMUX, asa cum e ilustrat in figura 7.

REFS1	REFS0	Voltage Reference Selection[1]
0	0	AREF, Internal V_{REF} turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Internal 1.1V Voltage Reference with external capacitor at AREF pin
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Tabelul 1: Tensiunea de referinta a adc-ului.

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 7: Registrul de multiplexare ADCMUX

- **Bit 5 – ADLAR – ADC Left Adjust Result** – Se face ‘1’ pentru a putea face left adjust la rezultatul obtinut.

- **Bits 4:0 – MUX4:0 – Analog Channel and Gain Selection Bits** – Pe acesti biti selectam canalul cu care dorim sa lucram. Initial acesti biti sunt 0.

  Asadar pentru a initializa ADMUX scriem : ADMUX = (1<<REFS0);

Prescallerul controleaza clockul intern al ADC-ului. De regula metoda de conversie a ADC-ului necesita un clock cu o frecventa intre 50 si 200 kHz. Daca am avea nevoie de o rezolutie mai slaba de conversie, 8 biti spre exemplu, am avea nevoie de un clock si mai mare pentru rata de esantionare. Procesul de conversie necesita 13-14 cicluri de ceas din partea adc-ului. Optiunile de subdivizare sunt prezentate in tabelul 2 de mai jos.

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Table 2. Valorile de subdivizare ale ceasului de esantionare

Acesti biti sunt setati in registrul Adc Control and Status Register (ADCSRA) precum se vede in figura 8 de mai jos.

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 8. ADCSRA Register

Semnificatia bitilor din registrul ADCSRA:

- **Bit 7 – ADEN – ADC Enable** – Asa cum ii sugereaza si numele, acest bit face enable la functionalitatea de ADC. Daca acest bit nu ar fi setat, operatiile de ADC nu ar putea avea loc, iar portul pe care operatiile se desfasoara se va comporta ca un port classic GPIO.

- **Bit 6 – ADSC – ADC Start Conversion** – Daca faceti acesti bit 1, incepe o conversie analog/digitala. Acest bit ramane ‘1’ cat timp conversia e in lucru, dupa care redevine 0. In mod normal operatiile de conversie folosind ADC-uri au nevoie de 13 pulsuri de ceas. Cu toate astea prima data cand procesul este pornit durata conversiei este mai mare, 25 de cicluri de ceas – acest lucru se intampla deoarece prima data cand se porneste procesul se executa si pasii de initializare.

- **Bit 5 – ADATE – ADC Auto Trigger Enable** – Setand acest bit la ‘1’ se permite operatia de aut-triggering pentru ADC. ADC-ul este pornit automat la fiecare front crescator al semnalului de tact al ADC-ului. Pentru mai multe detalii va trebuii sa va uitati in datasheetul avr-ului la registrul SFIOR.

- **Bit 4 – ADIF – ADC Interrupt Flag** – In momentul in care o conversie s-a finalizat si registrii s-au actualizat, aceasta valoare devine ‘1’. Acest bit este de regula folosit pentru a verifica daca conversia s-a finalizat sau este inca in curs de desfasurare.

- **Bit 3 – ADIE – ADC Interrupt Enable** – Cand acest bit este setat la ‘1’, intreruperea de ADC este activata. Acest bit e folosit in cazul intreruperilor de ADC.

- **Bits 2:0 – ADPS2:0 – ADC Prescaler Select Bits** – Acestia sunt bitii de prescaller pe care i-am detaliat mai sus.

ADCL si ADCH – ADC Data Registers

In urma unei conversii rezultatul ADC-ului este stocat in acesti registrii.De vreme ce ADC-ul are o rezolutie pe 10 biti nu este sufficient sa avem doar un registru. Asadar se folosesc 2 registrii- ADCL si ADCH (octetii low si high ai conversiei). Ambii impreuna pot fi numiti ADCW (compilatorul va genera codul pentru a inlocui ADCW cu ADCL si ADCH).

In figura 9 sunt ilustrati atat cei doi registrii, cat si efectul setarii bitului ADLAR la 1.

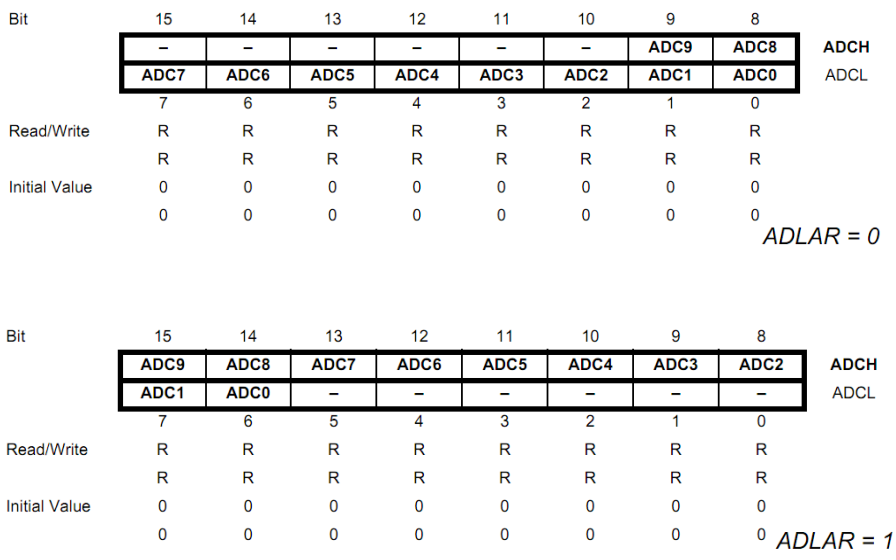


Figure 9. Ilustrarea registrilor ADCL ,ADCH si ilustrarea functionalitatii bitului ADLAR.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup()
{
        //16MHz/128 = 125kHz clockul de referinta la ADC

        ADCSRA |= ((1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0));
        ADMUX |= (1<<REFS0);      //Seteaza tensiunea de referinta la Avcc (5v)
```

```
        ADCSRA |= (1<<ADEN);      //Activteaza ADC
        ADCSRA |= (1<<ADSC);
}

void loop()
{
        int val = read_adc(0); //citirea valorii analogice
        lcd.setCursor(0,1);
        lcd.print(val);
}


uint16_t read_adc(uint8_t channel)
{
        ADMUX &= 0xE0;          //sterge bitii MUX0-4
        ADMUX |= channel&0x07;   //Seteaza bitii in MUX0-2 pentru noul canal din care va
        // trebui sa citim
        ADCSRB = channel&(1<<3); //Seteaza MUX5
        ADCSRA |= (1<<ADSC);      //incepe conversia
        while(ADCSRA & (1<<ADSC));  //Asteapta pana cand conversia se termina

         return ADCW;
}
```

Lucru individual

1. Implementati exemplele din laborator si intrebati cadrul didactic in legatura cu orice nelamurire aveti legata de partea de senzori pe AVR.
2. Folosind un sensor de lumina implementati cu instructiuni Arduino si instructiuni AVR o functionalitate de tipul night light: cand nivelul de lumina de pe sensor scade dati un PWM mai mare la un led.
3. Folosind functia micros(), comparati viteza de conversie ADC a functiei Arduino analogRead, cu viteza functiei read_adc() data ca exemplu.
4. Creati un sistem termostat. Folosind butoanele, setati temperatura dorita, care va fi afisata in acest timp pe LCD. Dupa setare, pe LCD se va afisa temperatura citita de la senzor. Daca temperatura ambientala este mai mica decat cea dorita, afisati pe LCD mesajul “Incalzire”. Cand temperatura ambientala este mai mare sau egala cu cea dorita, mesajul “Incalzire” va disparea.