



Proiectarea cu Micro-Procesoare

Lector: Mihai Negru

An 3 – Calculatoare și Tehnologia Informației

Seria B

Curs 9: Procesarea Semnalelor Analogice

<http://users.utcluj.ro/~negrum/>



- **Comparator analogic:** compară un semnal de intrare cu alt semnal de intrare, sau cu tensiuni de referință, indicând printr-un bit relația dintre aceste tensiuni (care este mai mare), sau producând cereri de întrerupere
- **Convertor analog/digital pe 10 biți,** cu 16 intrări analogice multiplexate: convertește semnalul analogic într-o valoare numerică între 0 și 1023.



- **Conversia semnalului analogic în semnal digital**
 - Transformarea unui semnal analogic (tensiune) de intrare într-o valoare digitală pe **10 biți**
 - Intrarea poate fi “single end” – tensiunea de intrare se calculează între pinul de intrare și GND, sau diferențială – diferența de tensiune între doi pini de intrare
 - Pentru intrarea diferențială, se poate utiliza amplificare (Gain)

$$ADC = \frac{V_{IN} * 1024}{V_{REF}}$$

ADC: 0...1023

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot Gain \cdot 512}{V_{REF}}$$

ADC: -512...511, complement față de 2

- V_{REF} poate fi:
 - AVCC – conectat în placă la VCC
 - A_{REF} – tensiune externă de referință
 - Tensiune de referință internă **2.56 V**



Convertor Analog/Digital



ATmega 328P

Rezoluie – 10 biți

Single ended – 8 canale multiplexate

Intrare senzor temperatura

Optional – ajustare stanga / dreapta
pentru citirea rezultatului ADC

Interval tensiune intrare: 0 ... VCC

Tensiune de referinta: 1.1 V interna

Moduri de functionare:

- Free running sau Single Conversion
- Interrupt on ADC Conv. Complete
- Sleep mode noise canceller

ATmega 2560

Rezoluie – 10 biți

Single ended – 16 canale multiplexate

Diferențială – 14 canale

Diferențială – 4 canale cu amplificare
(optionala 10x sau 200x)

Optional – ajustare stanga / dreapta
pentru citirea rezultatului ADC

Interval tensiune intrare: 0 ... VCC

Interval tensiune intrare: 2.7 V – VCC

Tensiune de referinta: Selectabil 2.56 V
sau 1.1 V (interne)

Moduri de functionare:

- Free running sau Single Conversion
- Interrupt on ADC Conv. Complete
- Sleep mode noise canceller

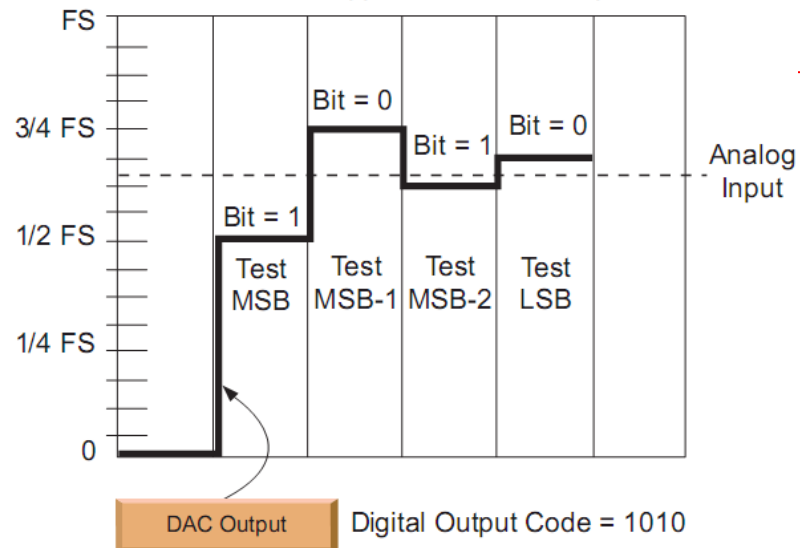
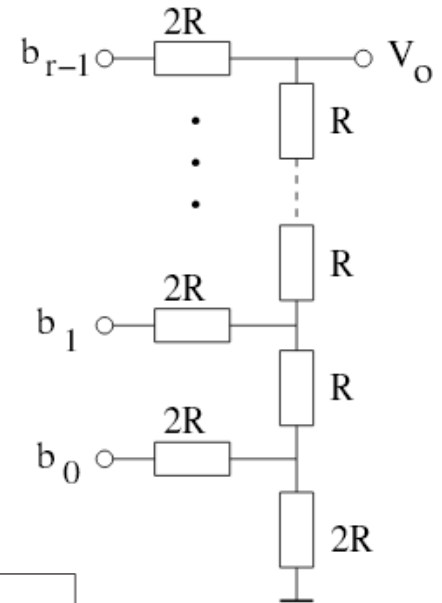
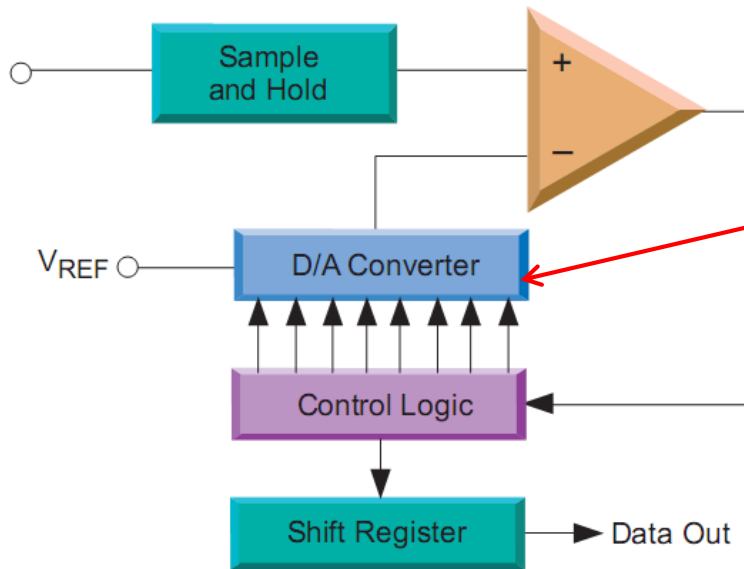


Convertor Analog/Digital



- **Principiul de funcționare**

- Comparații succesive cu o tensiune de referință
- Principiul este similar cu căutarea binară

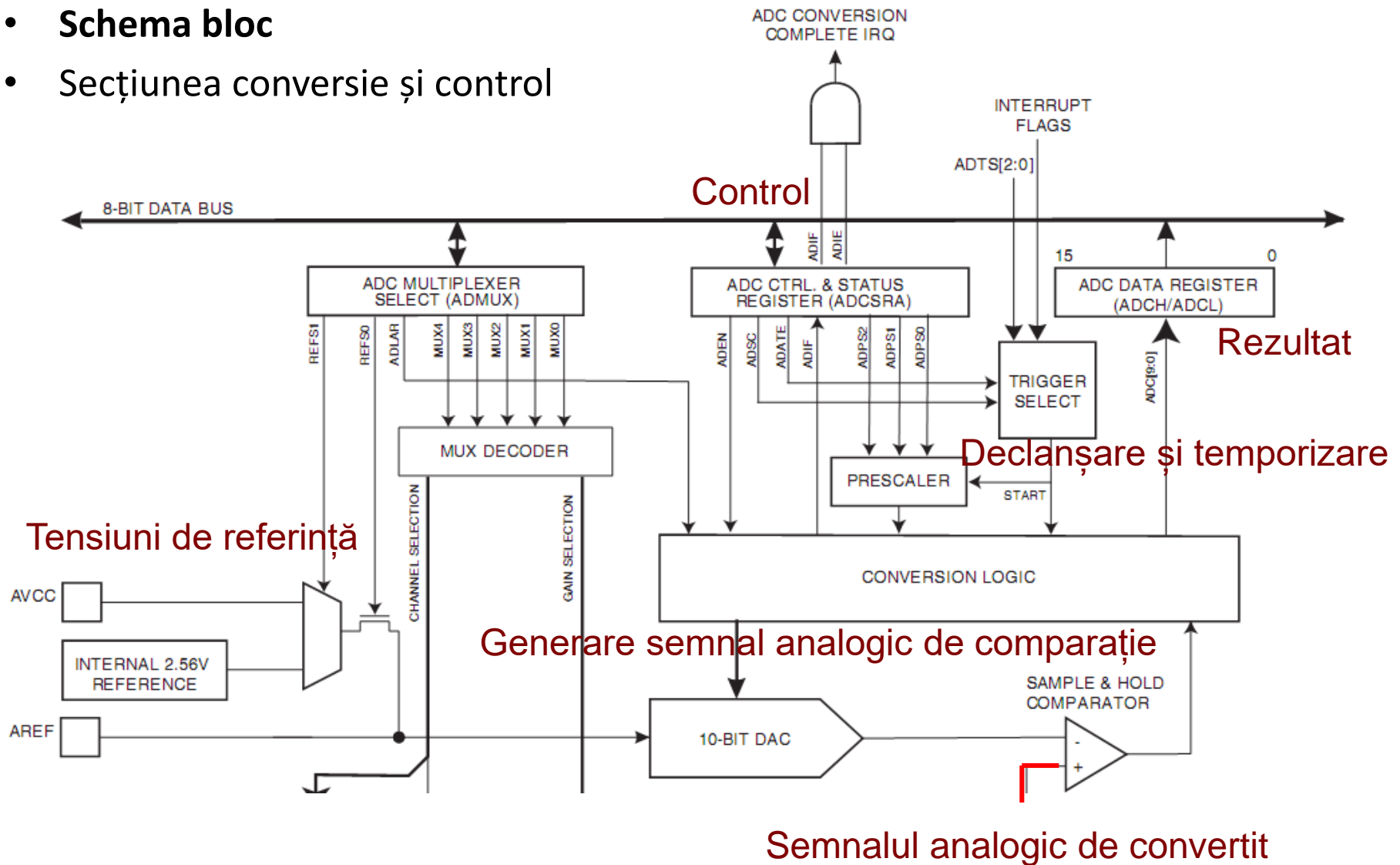




Convertor Analog/Digital



- Schema bloc
- Secțiunea conversie și control



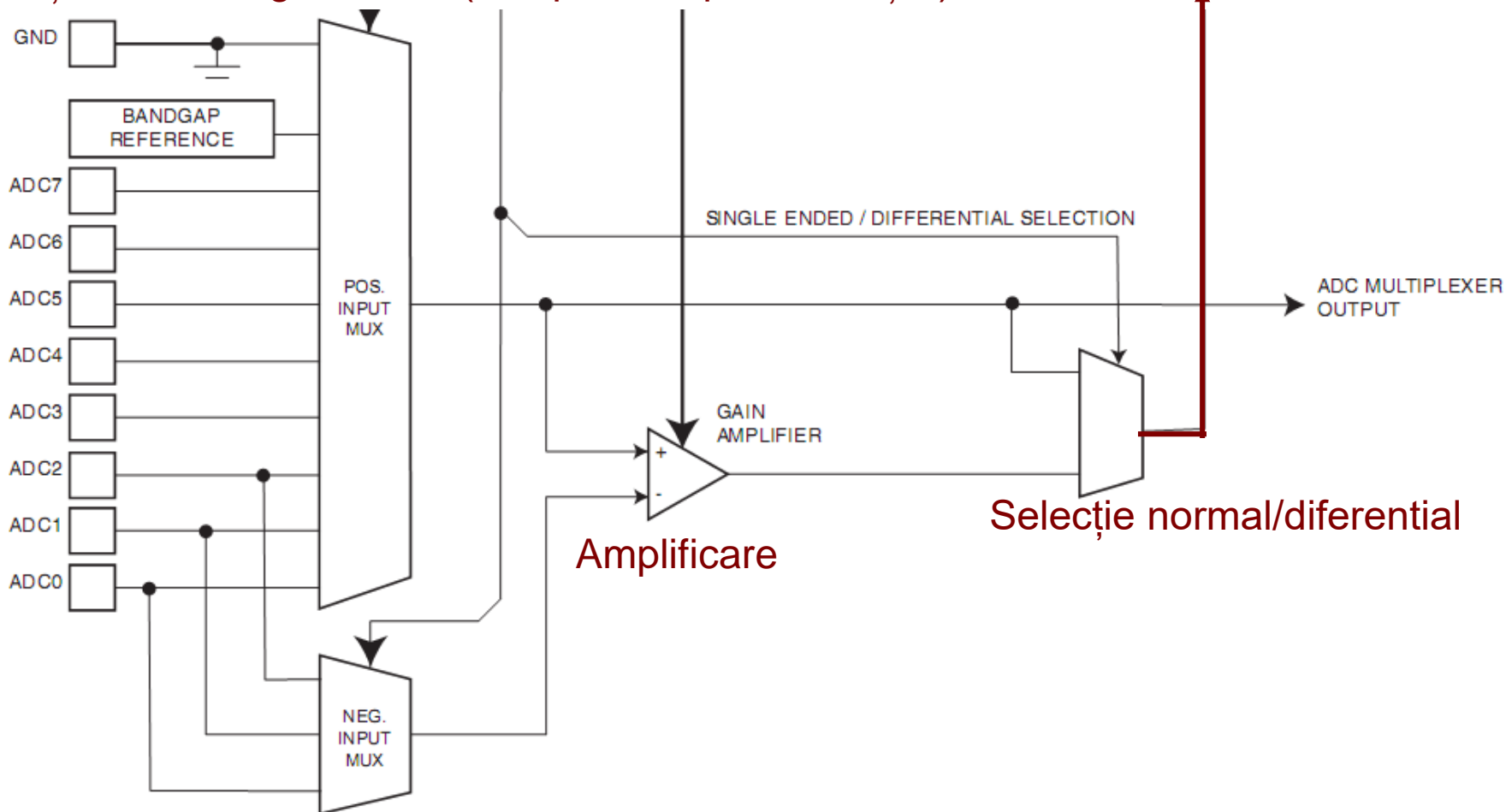


Convertor Analog/Digital



- **Schema bloc** – secțiunea de selecție a intrării

Selecție intrări single ended (sau pozitive pt. diferențial)



Către modul de conversie

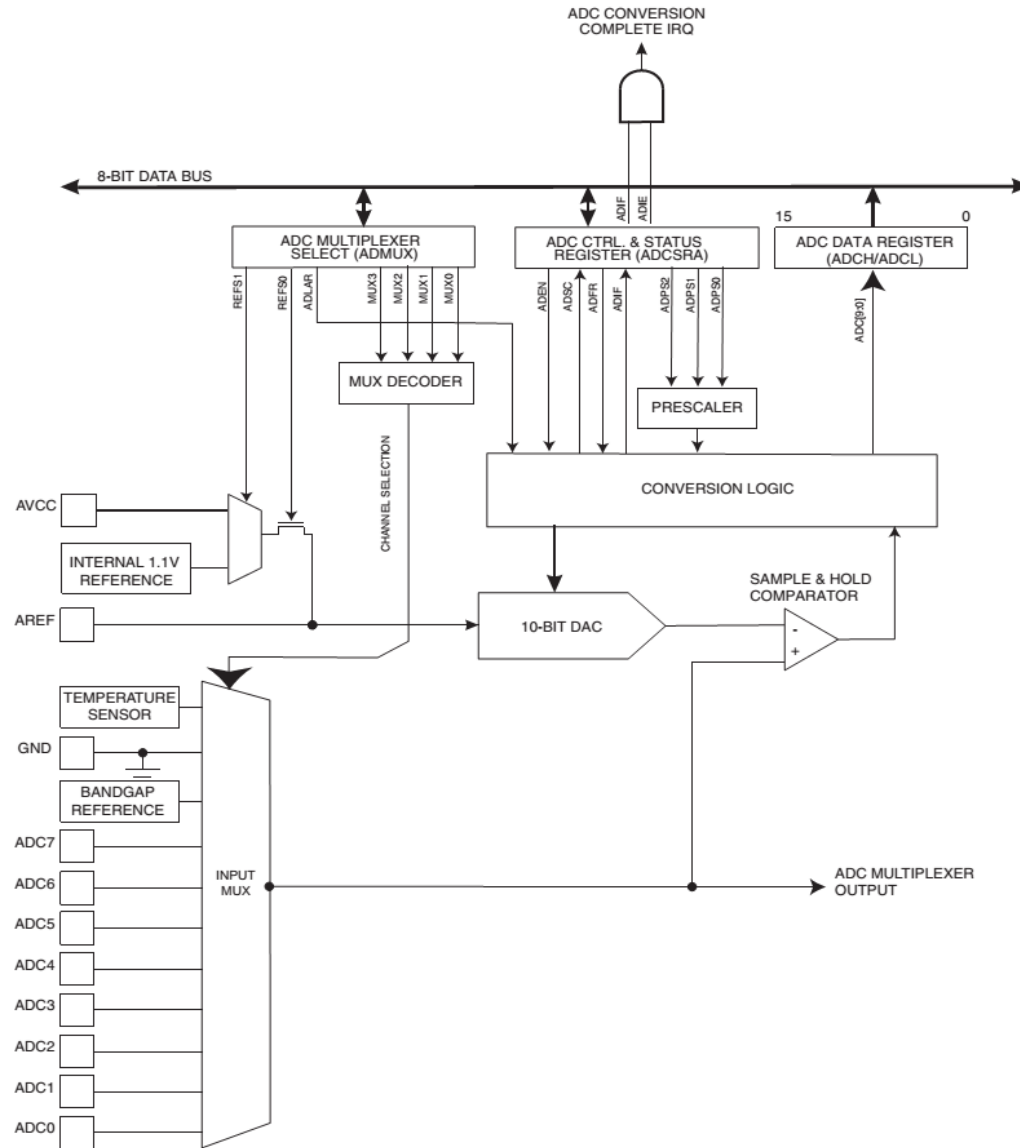
Amplificare

Selecție normal/diferential

Selecție intrări negative (pt. Diferențial)

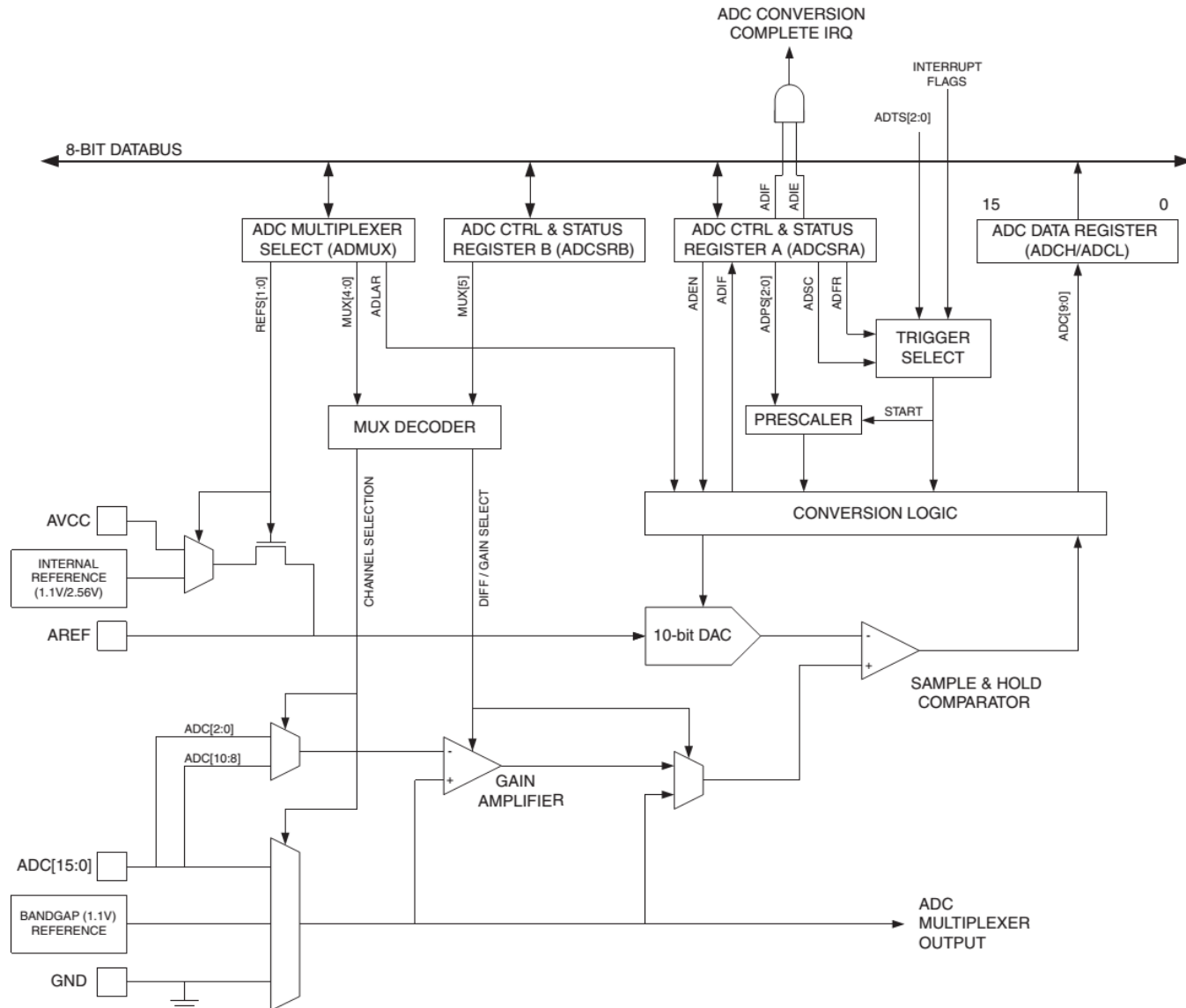


Convertor Analog/Digital ATmega 328P





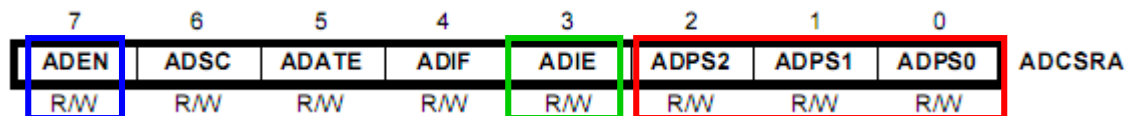
Convertor Analog/Digital ATmega 2560





- **Configurare convertor**

- Activare – setarea bitului **ADEN** din registrul **ADCSRA**
- Folosire întreruperi la terminarea conversiei – setarea bitului **ADIE** din **ADCSRA**
- Selecție frecvență pentru ceasul convertorului: bitii **ADPS2:0**



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

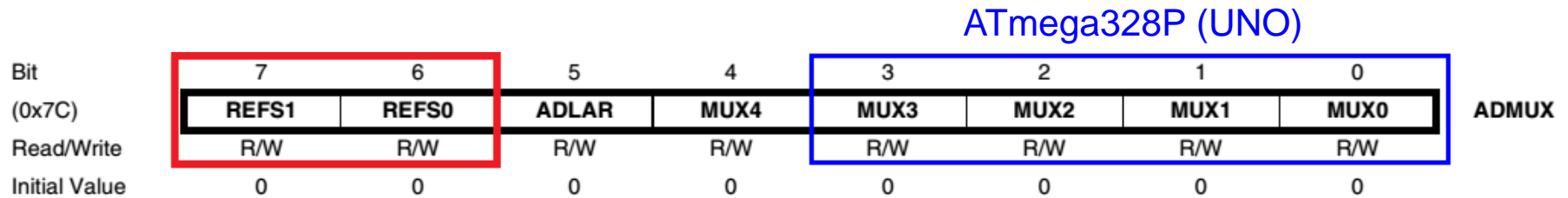


Convertor Analog/Digital



Selecția tensiunilor de referință:

- biții REFS1:0 din ADMUX (ADC Multiplexer Selection Register)



ATmega2560 (MEGA)

REFS1	REFS0	Voltage Reference Selection ⁽¹⁾
0	0	AREF, Internal V_{REF} turned off
0	1	AV_{CC} 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

ATmega328P (UNO)

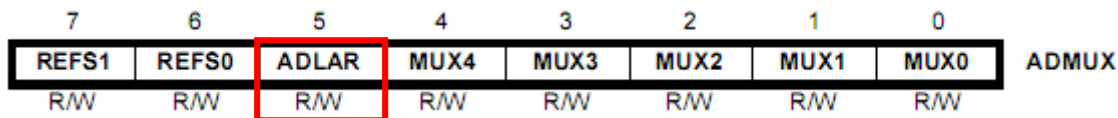
REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin



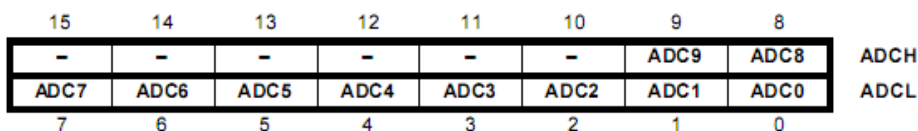
Convertor Analog/Digital



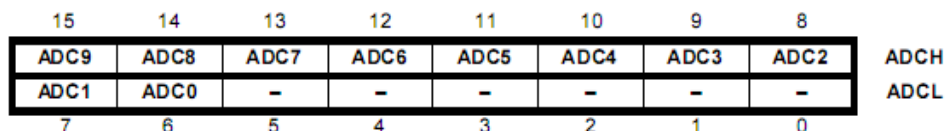
- **Configurare date de ieșire și accesare rezultate**
- Disponerea celor 10 biți de date este controlată de bitul **ADLAR** (ADC left adjust) din **ADMUX**



ADLAR = 0 – ajustare la dreapta



ADLAR = 1 – ajustare la stânga



- Citirea datelor: prima data se citește **ADCL**, apoi **ADCH**. La citirea ADCL, registrul ADCH ramâne cu aceeași valoare până este citit
- Dacă ADLAR = 1, se poate citi doar **ADCH**, ca rezultat pe 8 biți (rezoluție mai scăzută) $ADCH = V_{in} * 256 / V_{ref}$



Convertor Analog/Digital



- **Selecția intrărilor – Configurarea biților MUX din ADMUX**

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8 ⁽¹⁾
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

ATmega328P

Aceste tabele nu conțin toate combinațiile!
→ datasheet-ul corespunzător chipului

MUX5:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain	
000000	ADC0	N/A	N/A	N/A	
000001	ADC1				
000010	ADC2				
000011	ADC3				
000100	ADC4				
000101	ADC5				
000110	ADC6				
000111	ADC7				
001000 ⁽¹⁾	N/A	ADC0	ADC0	10x	
001001 ⁽¹⁾		ADC1	ADC0	10x	
001010 ⁽¹⁾		ADC0	ADC0	200x	
001011 ⁽¹⁾		ADC1	ADC0	200x	
001100 ⁽¹⁾		ADC2	ADC2	10x	
001101 ⁽¹⁾		ADC3	ADC2	10x	
001110 ⁽¹⁾		ADC2	ADC2	200x	
001111 ⁽¹⁾		ADC3	ADC2	200x	
010000		N/A	ADC0	ADC1	1x

.....

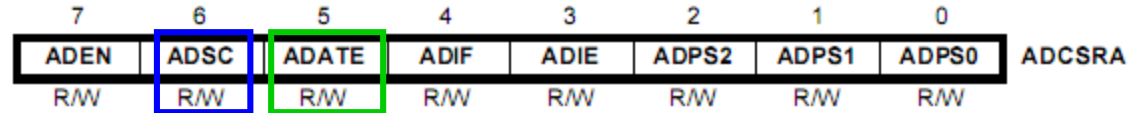
011110	1.1V (V_{BG})	N/A		
011111	0V (GND)	N/A		
100000	ADC8	N/A		
100001	ADC9			
100010	ADC10			
100011	ADC11			
100100	ADC12			
100101	ADC13			
100110	ADC14			
100111	ADC15			

ATmega2560

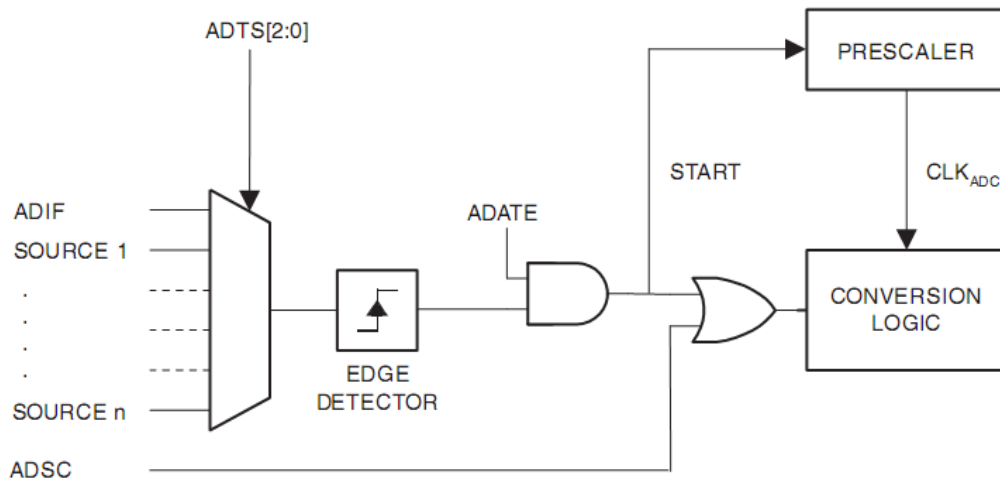


Convertor Analog/Digital

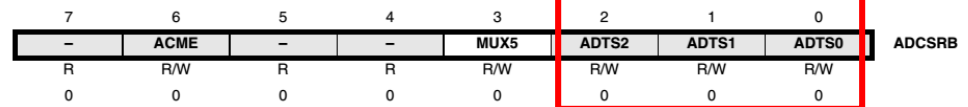
• Declanșare conversie



- La cerere – scrierea bitului **ADSC** din **ADCSRA**. Acest bit rămâne ‘1’ în timpul conversiei, și se șterge la terminare
- Automat, declanșat de variații ale semnalelor de intrare – dacă bitul **ADATE** din **ADCSRA** e setat. Una din surse este **ADIF** – flag-ul care semnalează terminarea unei conversii, și eventual cererea unei întreruperi. În acest caz, o nouă conversie începe când se termină cea anterioară.
- Sursele pentru declanșare automată → biții **ADTS** din **ADCSRB**



ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

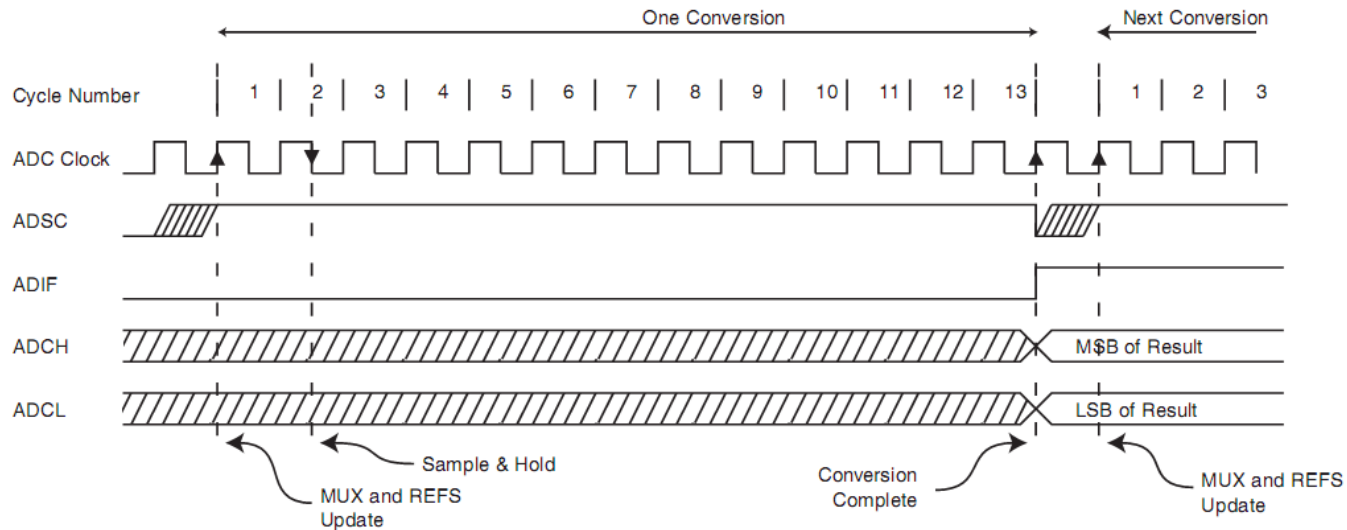




Convertor Analog/Digital



- **Timpi de conversie, diagrame de timp**



Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5
Normal conversions, differential	1.5/2.5	13/14



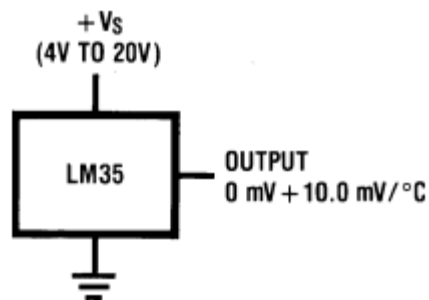
Convertor Analog/Digital



- **Măsurarea temperaturii**
- Senzor: National Semiconductor LM 35 (<http://www.ti.com/lit/ds/symlink/lm35.pdf>)
- $V_{output} = T * 0.01 [V] / [^{\circ}C]$
- Intrare single ended:

$$ADC = \frac{V_{IN} * 1024}{V_{REF}}$$

- Dacă $ADLAR = 1$, putem scrie: $ADCH = V_{in} * 256 / V_{ref}$
- $ADCH = V_{output} * 256 / V_{ref}$
- $ADCH = T * 2,56 / V_{ref}$
- Dacă V_{ref} este **tensiunea internă de referință de 2,56 V**, atunci **$ADCH = T [^{\circ}C]$**



DS005516-3





- Exemplu – masurarea temperaturii ATmega 2560

```
rcall ADC_init
```

```
loop:
```

```
    rcall start_ADC_conversion ; Starts a conversion
```

```
    rcall wait_ADC_complete ; Wait to complete the current conversion
```

```
    rcall ADC_read ; Read the result in r16
```

```
    rjmp loop
```

```
ADC_Init:
```

```
    ldi r16, 0b11100011 ; Vref=2,56 V internal, ADLAR=1 (Data Shift left)  ADC3 single ended
```

```
    out ADMUX, r16
```

```
    ldi r16, 0b10000000 ; Activate ADC, max. speed (clock div. ratio = 2)
```

```
    out ADCSRA, r16
```

```
ret
```

```
start_ADC_conversion: sbi ADCSRA, ADSC ; ADC start, set ADSC bit in ADCSRA
```

```
ret
```

```
wait_ADC_complete:  sbic ADCSRA, ADSC ; When ADSC=0, conversion is finished
```

```
    rjmp wait_ADC_complete
```

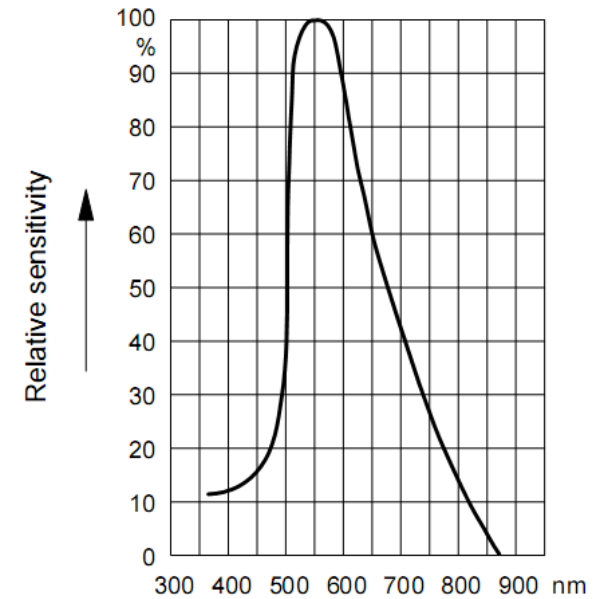
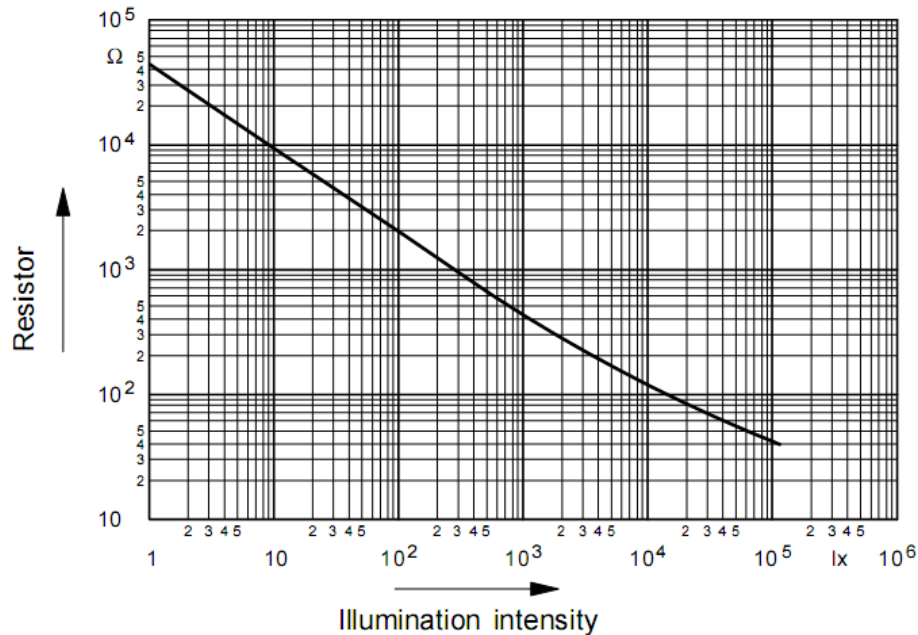
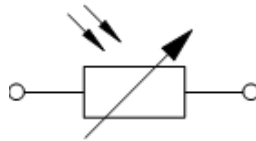
```
ret
```

```
ADC_read:           in r16, ADCH ; ADCH – temperature on 8 bits
```

```
ret
```

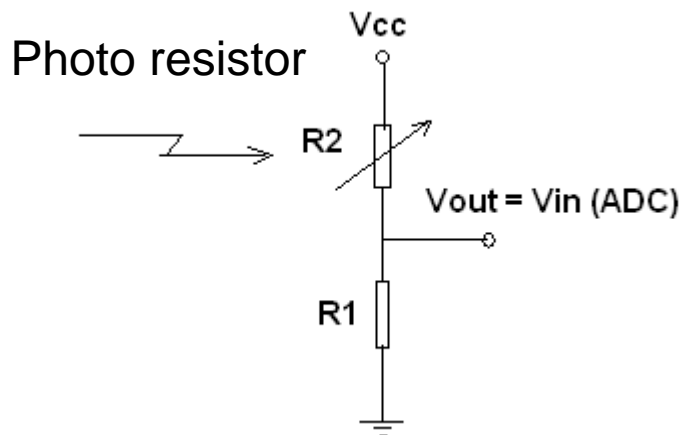
- **Măsurarea intensității luminoase**

- Soluția cea mai simplă: folosirea unei fotorezistențe (sulfură de cadmiu)
- Rezistența scade pe măsură ce intensitatea luminoasă crește



- **Măsurarea intensității luminoase (continuare)**

- Fotorezistența împreună cu o rezistență fixă formează un divizor de tensiune
- Relația dintre R_F și intensitatea luminoasă trebuie calibrată
- V_{OUT} se introduce la o ADC intrare single ended, GND comun
- ADLAR = 1 (low resolution): $ADCH = V_{in} * 256 / V_{ref}$



$R_2 = 200 \Omega$ (bright light) $1.4 M\Omega$ (dark)

$R_1 = \text{constant}$ (ex: 20 K)

$$V_{OUT} = V_{CC} \frac{R_1}{R_1 + R_2}$$

Calibrare sensor:

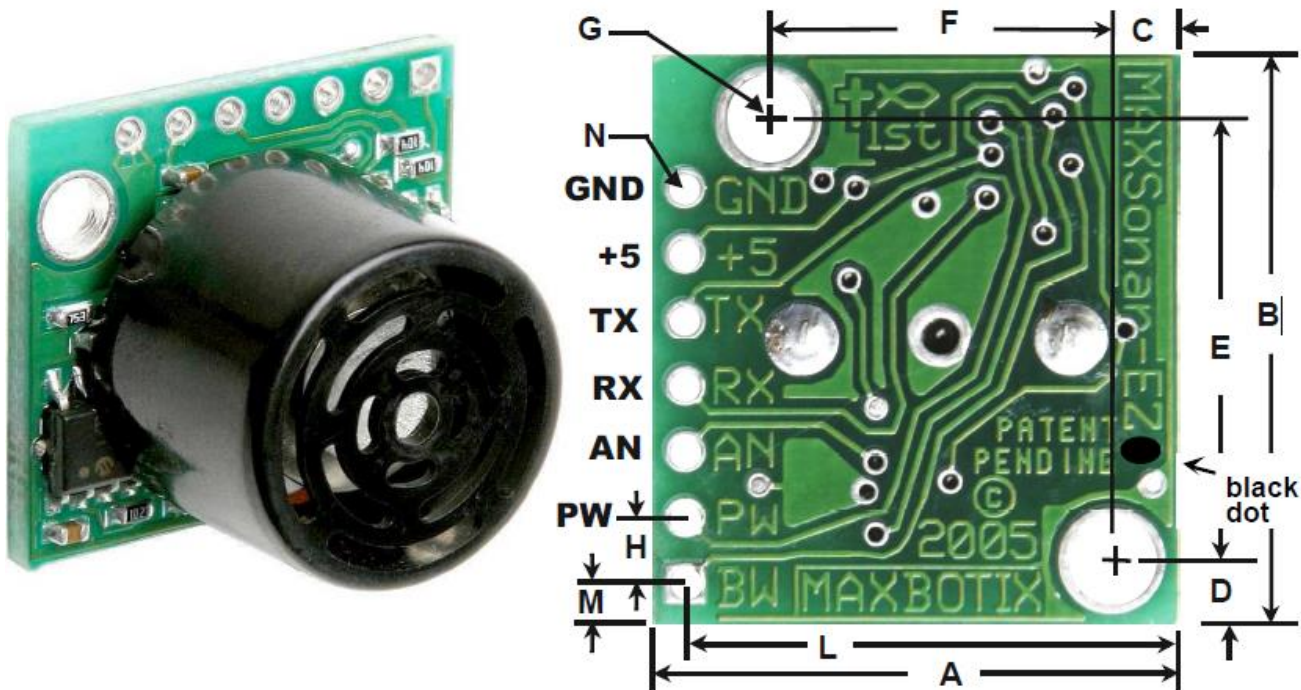
- Se măsoară ADCH pentru cea mai mică luminozitate (dark): $ADCH_{MIN}$
- Se măsoară ADCH pentru cea mai mare luminozitate: $ADCH_{MAX}$

Measurement:

- Compute the light brightness B [%]:

$$B[\%] = \frac{ADC - ADC_{MIN}}{ADC_{MAX} - ADC_{MIN}} * 100$$

- **LV-MaxSonar-EZ0 Sonar pentru detecția obstacolelor**
 - Comunicare serială (UART), Baud 9600, 8 biți de date, 1 bit stop, fără paritate
 - leșire analogică, **Vcc/512** Volți per inch (1 inch = 2.54 cm)
 - leșire PWM, **0.147 ms / inch**
 - Frecvența ultrasunetelor: 42 KHz
 - Distanța: 0-6.45 m, depinde foarte mult de dimensiunea obstacolului
 - Utilizarea cea mai simplă: folosind **convertorul ADC**





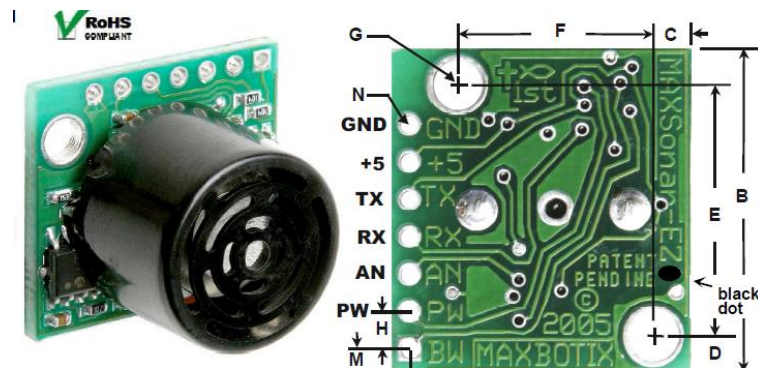
Convertor Analog/Digital



- **Exemplu: masurarea distantei cu senzor Sonar**
 - LV-MaxSonar®-EZ0™ High Performance Sonar Range Finder [7]

AN – ieșire analogică, cu factor de scalare ($V_{CC}/512$) per inch. O alimentare de 3.3V produce $\sim 6.4\text{mV/in} \approx 2.56\text{mV/cm}$
Domeniu de distanta: 6-in (15 cm) .. 254 in (645 cm)
rezolutie 1-inch

$$ADC = \frac{V_{IN} \times 1024}{V_{REF}} = \frac{2.56\text{mV} \times d[\text{cm}] \times 1024}{2.56[\text{V}]} \approx d[\text{cm}]$$



ADC_Init:

```
ldi r16, 0b11000011
out ADMUX, r16
ldi r16, 0b10000000
out ADCSRA, r16
```

ret

ADC_read:

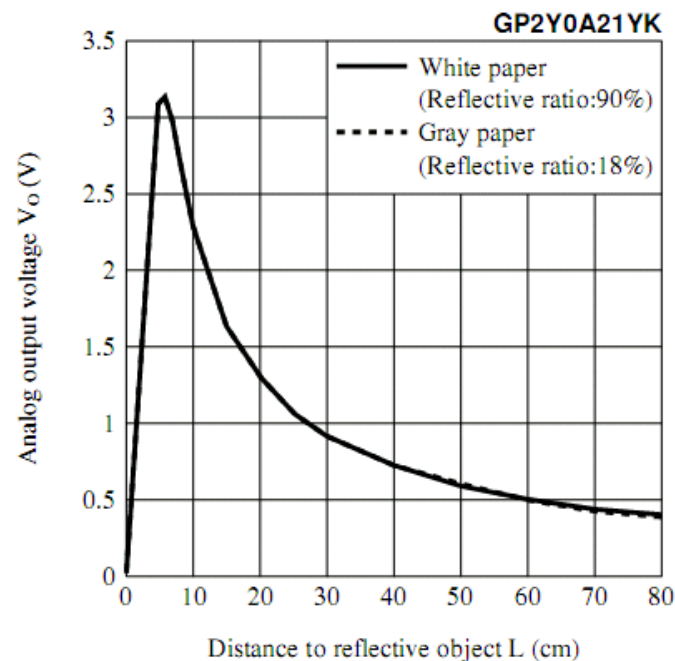
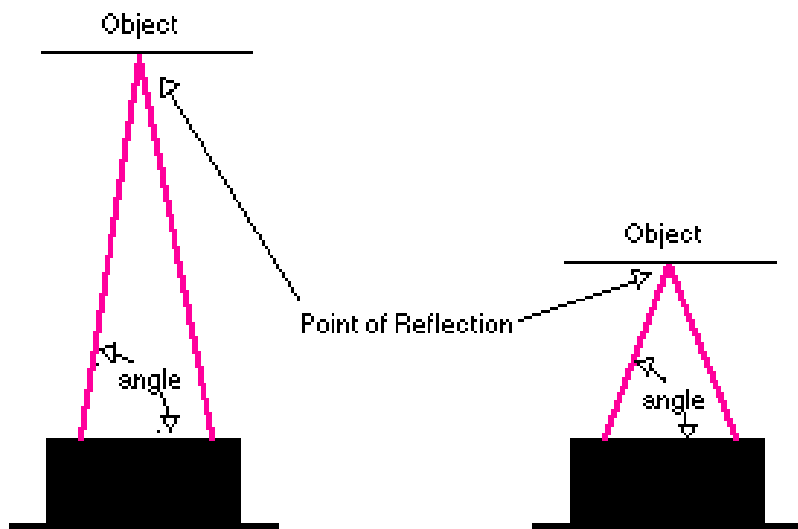
```
in r20, ADCL
in r21, ADCH
```

ret

```
; Vref=2,56 V internal, ADLAR=0 (Data Shift right – full 1024 bit resolution),
; ADC3 single ended
; Activare ADC, viteza maxima
```

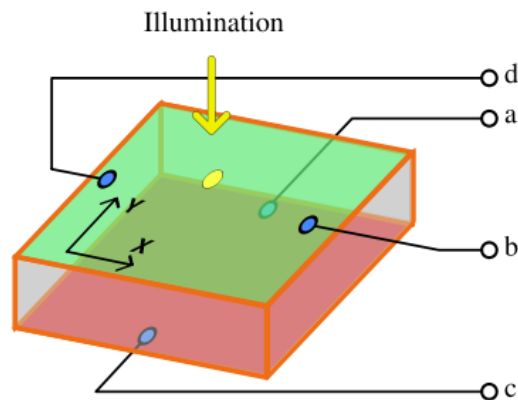
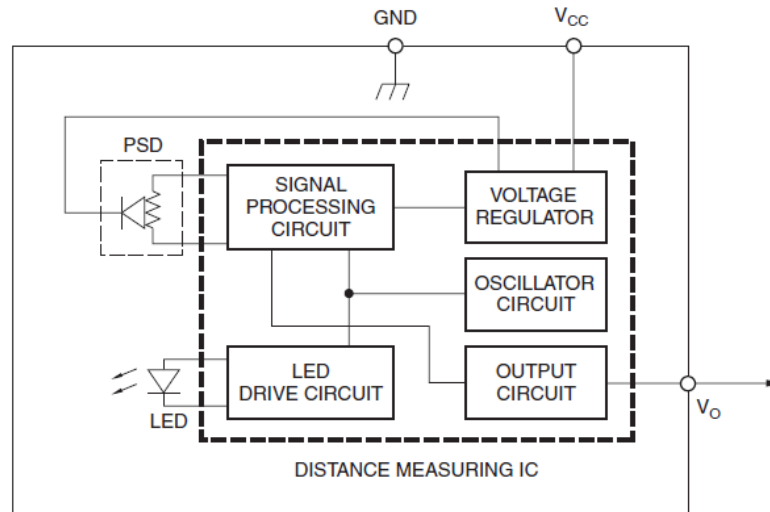
```
//ADC citire registru inferior
//ADC citire registru superior // r21:r20 = d[cm]
```

- **SHARP GP2XX, familie de senzori pentru distanță bazați pe reflexia IR**
 - Folosește triangulația pentru calculul distanței
 - Se măsoară unghiul sub care se întoarce raza emisă
 - Leșire analogică, **neliniară**
 - Cost redus (approx. 10 usd)
 - Ușor de montat, robust



Percepția Mediului

- **SHARP GP2XX, familie de senzori pentru distanță bazați pe reflexia IR**
 - Folosește o fotodiodă sensibilă la poziție
 - Pe baza răspunsului ei, se determină unghiul de reflexie



$$x = k_x \cdot \frac{I_b - I_d}{I_b + I_d}$$

$$y = k_y \cdot \frac{I_a - I_c}{I_a + I_c}$$

Percepția Mediului

• Accelerometru ADXL335

- Măsoară accelerația pe 3 axe, de la -3 g ... 3 g
- Alimentare între 1.8 V ... 3.6 V
- Ieșire pentru 0 G: $V_{cc} / 2$
- Sensibilitate tipică pentru $V_{cc} = 3.3V$: 300 mV/G

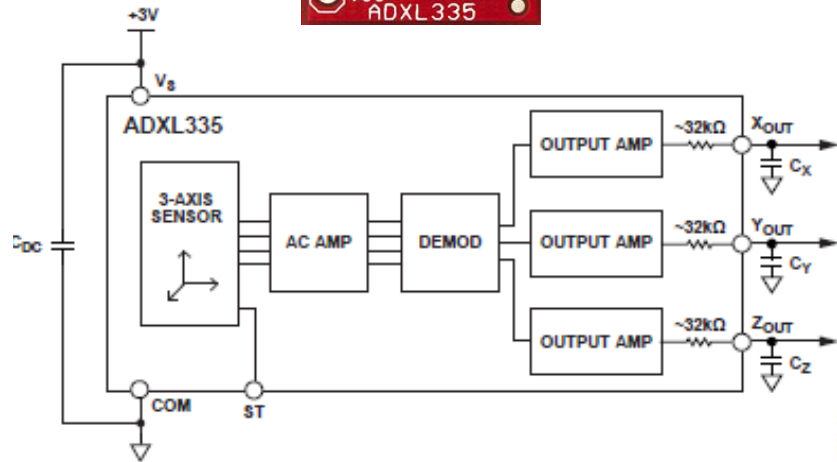
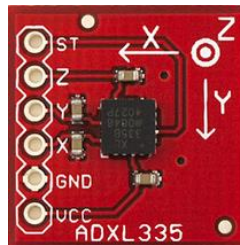
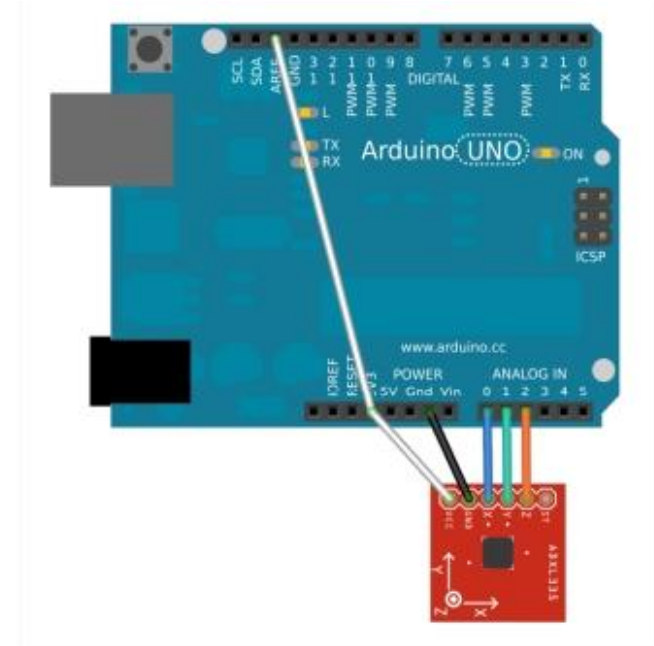


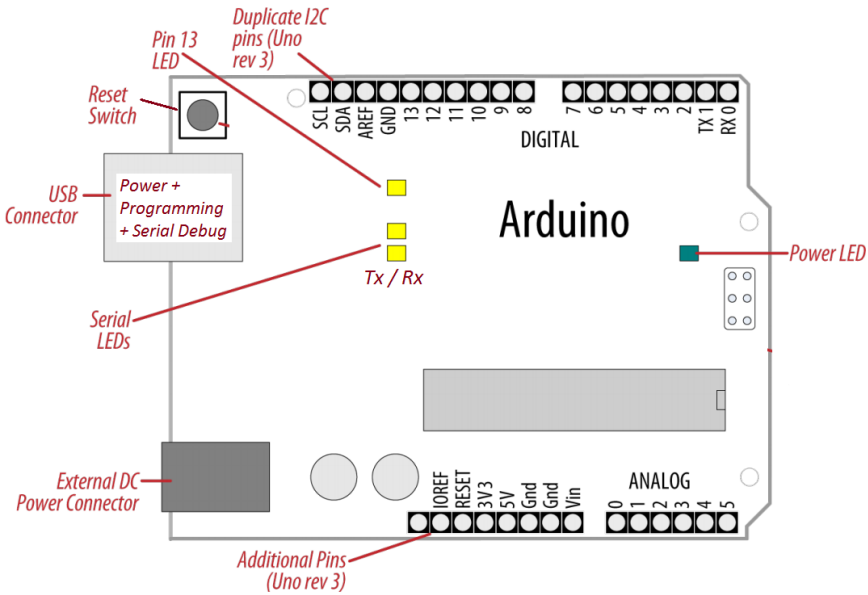
Figure 1.



Arduino 3.3 V	ADXL335 VCC
Arduino GND	ADXL335 GND
Arduino Analog0	ADXL335 X
Arduino Analog1	ADXL335 Y
Arduino Analog2	ADXL335 Z
Arduino 3.3	Arduino AREF



Procesare semnal analogic cu Arduino



Arduino UNO: A0 .. A5

Arduino MEGA: A0 .. A15

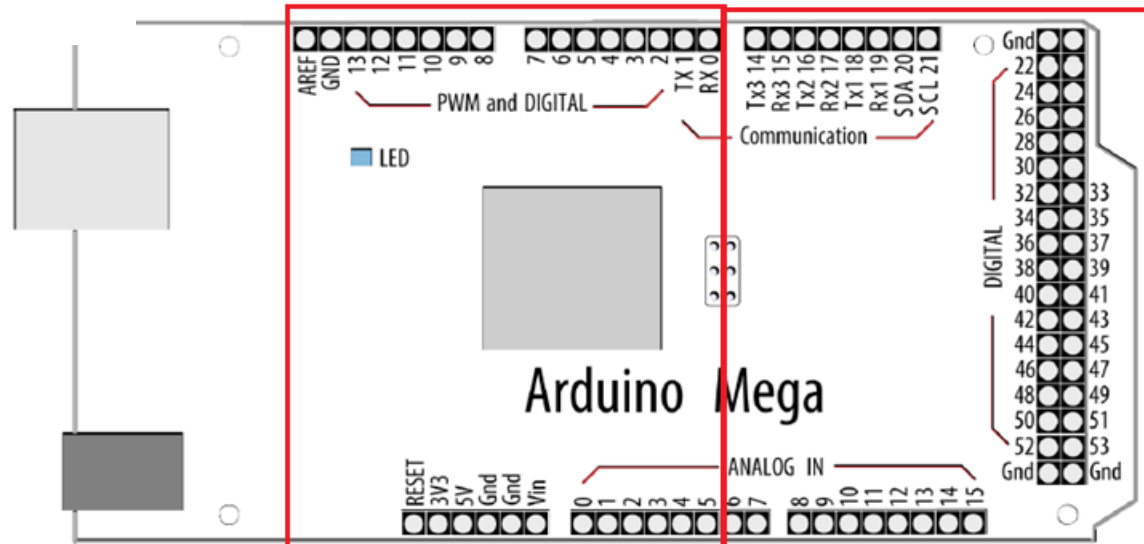
- Pinii analogici sunt intrări pentru convertorul ADC al microcontrollerului.
- ADC are rezoluția de 10 biți, returnând valori între 0 și 1023

Alți pini:

A_{REF} (intrare) – tensiune de referință externă pentru ADC

IOREF (ieșire) – tensiune de referință pentru shield-uri

Pin Layout identical with UNO



Pin Layout specific to MEGA



- **Funcția principală a pinilor analogici: citirea semnalelor analogice**
- Pinii analogici au și funcția de pin digital de uz general, ca și pinii digitali. Exemplu:
 - `pinMode(A0, OUTPUT);`
 - `digitalWrite(A0, HIGH);`
- Pinii analogici au de asemenea rezistențe **pull up resistors**, care funcționează în același fel ca rezistențele pinilor digitali. Ele sunt activate scriind HIGH pe pinul configurat ca intrare.
 - `digitalWrite(A0, HIGH); // activare pullup la A0 configurat ca input.`
- **Activarea unei rezistențe pull up va influența valorile citite cu `analogRead()` !!!**
- **Funcții:**
 - **`analogRead(pin)`** – citește o valoare de pe un pin analogic
 - **`analogReference(type)`** – configurează tensiunea de referință care va fi folosită pentru intrarea analogică (i.e. valoarea maximă a tensiunii de intrare măsurabilă pe pin-ul analogic)



- **analogReference(type)** – configurează tensiunea de referință care va fi folosită pentru intrarea analogică.
- **type** – stabilește referința folosită:
 - DEFAULT: tensiunea referință implicită, de 5 V (pentru UNO & MEGA)
 - INTERNAL: tensiune internă de referință, 1.1 V la UNO (*nu există la Arduino Mega*)
 - INTERNAL1V1: tensiune internă de referință 1.1V (*doar Arduino Mega*)
 - INTERNAL2V56: tensiune internă 2.56V (*doar Arduino Mega*)
 - EXTERNAL: tensiune de referință externă, aplicată la pinul AREF (**0 ... 5V**).
- După schimbarea tensiunii de referință, prima citire cu analogRead() poate fi eronată !!!
- **Nu folosiți o tensiune de referință externă negativă (<0V) sau mai mare de 5V pe pinul AREF! Dacă folosiți o tensiune externă de referință, configurați referința ca externă apelând analogReference() înainte de a apela funcția analogRead().** În caz contrar, veți pune în contact tensiunea de referință internă, generată în mod activ, cu tensiunea externă, putând cauza scurtcircuit și distrugerea microcontrollerului. !!!



Procesare semnal analogic cu Arduino



- `int digital_value analogRead(pin)` – citește o valoare de pe pinul analogic specificat
- O valoare analogica între 0 .. RANGE va produce un numar `digital_value` între 0 și 1023.
- Rezoluția de măsurare este deci: $RANGE \text{ volți} / 1024 \text{ unități}$.
- Pentru referința DEFAULT (5V) rezoluția devine:

$$\text{resolutionADC} = .0049 \text{ volti (4.9 mV) / unitate.}$$

- Pentru a converti valoarea citită `digital_value` la tensiunea analogică:

$$\text{Voltage} = \text{resolutionADC} * \text{digital_value}$$

- Pentru a converti voltajul la o valoare fizica masurata in [X] folositi:

$$\text{Measurement [X]} = \text{Voltage [V]} / \text{Sensor_resolution [V]} / [X]$$

- Durează aproximativ 100 microsecunde (0.0001 s) pentru a citi o intrare analogică, astfel încât rata maximă de citire este 10,000 valori pe secundă.
- Dacă pinul analogic nu este conectat la nimic, valoarea returnată de `analogRead()` va fluctua în functie de mai multi factori (e.g. ce tensiuni sunt pe ceilalți pini analogici, apropierea mâinii de placă...) !!!



- **Exemplu** – Citirea tensiunii unui potențiomtru conectat la intrarea analogică (<http://arduino.cc/en/Reference/AnalogRead>)

```
int analogPin = 3;           // aici se va conecta cursorul potentiometrului
                             // celelalte terminale ale potentiometrului se conecteaza la +5V si GND
int val = 0;                 // variabila in care se va citi valoarea analogica
float voltage;              // tensiunea calculata, in [mV]
float resolutionADC = 4.9;  // rezolutia in mV pentru referinta implicita de 5 V
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  val = analogRead(analogPin); // citire intrare analogica, pentru referinta 5 V
  voltage = val * resolutionADC; // conversie in mV
  Serial.print("Digital value = ");
  Serial.println(val);         // transmitere la PC
  Serial.print("Voltage [mV] = ");
  Serial.println(voltage);
}
```



Procesare semnal analogic cu Arduino

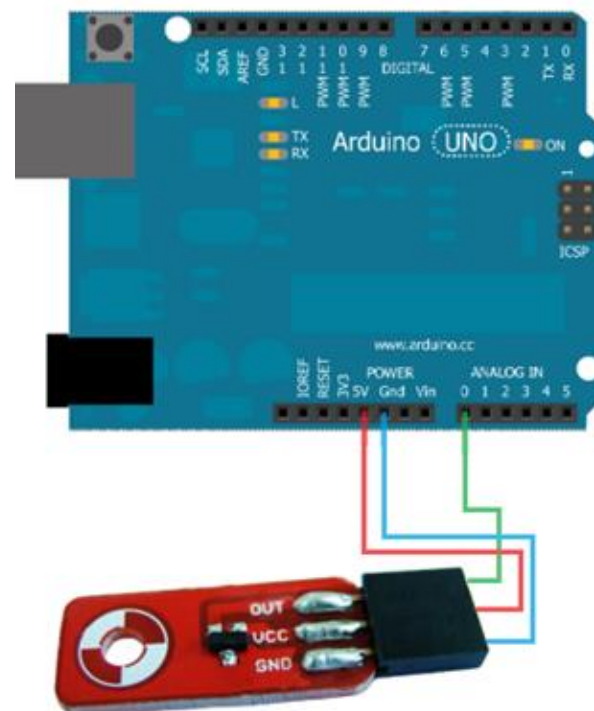
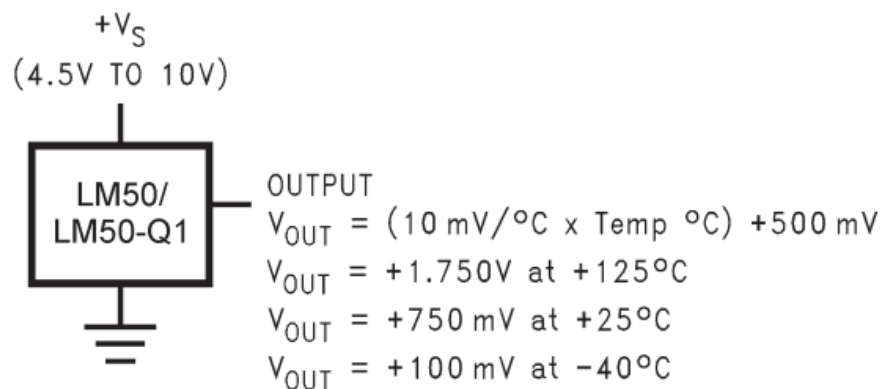


- **Senzor de temperatură folosind LM50 [5]**

<http://www.ti.com/lit/ds/symlink/lm50.pdf>

- **Caracteristici:**

- Iesire liniara $+10.0 \text{ mV}/^{\circ}\text{C} = 0.01\text{V}/^{\circ}\text{C}$
- Domeniu de temperaturi $-40^{\circ}\text{C} \dots +125^{\circ}\text{C}$
- Deplasament constant $+500 \text{ mV}$ pentru citirea temperaturilor negative
- Circuitul LM50 este inclus in senzorul de temperatura Brick [6]





- **Exemplu** – Citire temperatură de la senzor, face media a 10 citiri consecutive, și trimite către PC

```
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
}
```



- **Exemplu** – Măsurare distanțe cu sonarul LV EZ0 (rezoluție 10mV / inch \cong 0.01V)

```
const int sensorPin = 1;           // Iesire sonar conectata la A1
float resolutionADC = .0049;       // rezolutia implicita (pentru referinta 5V) = 0.049 [V] / unitate
float resolutionSensor = .01;      // rezolutie senzor = 0.01V/ inch

void setup() {
  Serial.begin(9600);
}

void loop() {
  float distance = readDistance(10, sensorPin);           // distanta in inch, media a 10 citiri
  Serial.print("Distance [inch]: "); Serial.println(distance); // afisare distanta in inch
  Serial.print("Distance [cm]: "); Serial.println(distance*2.54); // afisare distanta in cm, 1 inch=2.54 cm
  delay(200);
}

float readDistance(int count, int pin) {
  // citeste de 10 ori distanta, si face media
  float sumDist = 0;
  for (int i =0; i < count; i++) {
    int reading = analogRead(pin);
    float voltage = reading * resolutionADC;
    float distance = voltage / resolutionSensor; // conversie tensiune in distanta
    sumDist = sumDist + distance;
  }
  return sumDist / (float)count;
}
```




Referințe



1. Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V datasheet
2. Atmel Atmega64 datasheet
3. <http://arduino.cc/en/>
4. <http://arduino.cc/en/Reference/AnalogRead>
5. <http://www.ti.com/lit/ds/symlink/lm50.pdf>
6. <http://www.robofun.ro/senzori/vreme/senzor-temperatura-brick>
7. http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf