

Proiectare cu Microprocesoare

Curs 8

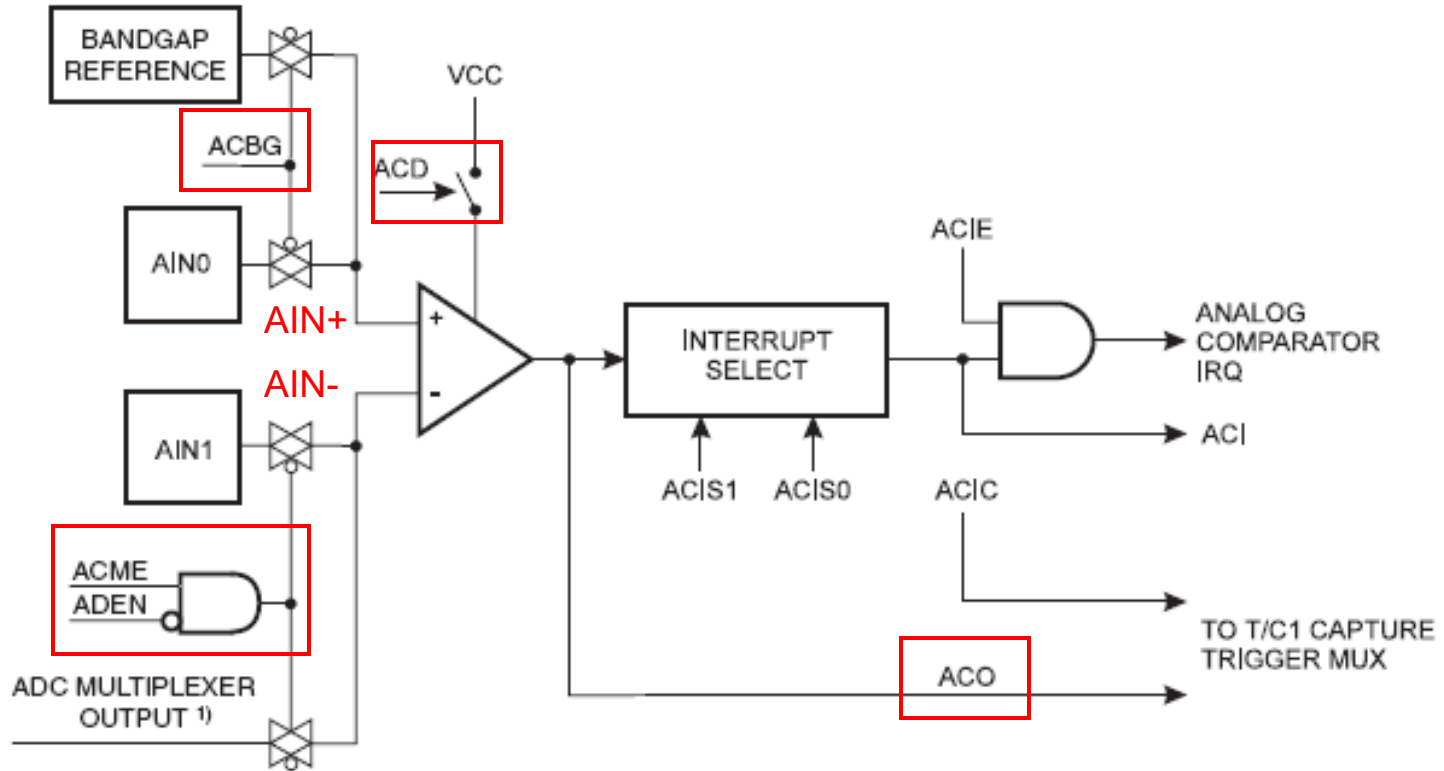
Utilizarea semnalelor analogice

An 3 CTI

Semestrul I

Lector: Răzvan Itu

Comparator analogic



Compară valorile analogice de pe AIN+ (pozitiv) și AIN- (negativ)

Dacă $(AIN+) > (AIN-) \rightarrow ACO = 1$

Activarea comparatorului analogic: bitul 7 (**ACD**) din registrul **ACSR**

AIN+ (Intrare): un semnal extern pe pinul **AIN0**, sau o tensiune de referință internă, selectată prin **ACBG** (bitul 6 din **ACSR**)

AIN- (Intrare): semnal extern (**AIN1**) (**ACME=0** sau **ADEN=1**) sau intrare de pe pinii **AD_{0..7}** (**ACME=1** și **ADEN=0**).

Comparator analogic

| | | | | | | | | | |
|---------------|---|------|---|---|---|-------|-------|-------|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| (0x7B) | – | ACME | – | – | – | ADTS2 | ADTS1 | ADTS0 | ADCSRB |
| Read/Write | R | R/W | R | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

ADCSRB – Registrul B de stare și control al convertorului ADC

- Bit 6 – ACME: Analog Comparator Multiplexer Enable

ACME ← 1, dacă convertorul ADC este deconectat (**ADEN=0**) atunci **AD_{0..7}** se conectează la AIN-

ACME ← 0, semnalul extern AIN1 este conectat la AIN-

Intrare multiplexată pentru comparatorul analogic (ATmega328P / UNO)

| ACME | ADEN | MUX2..0 | Intrare negativă pentru AC (AIN-) |
|------|------|---------|-----------------------------------|
| 0 | x | xxx | AIN1 |
| 1 | 1 | xxx | AIN1 |
| 1 | 0 | 000 | ADC0 |
| 1 | 0 | 001 | ADC1 |
| 1 | 0 | 010 | ADC2 |
| 1 | 0 | 011 | ADC3 |
| 1 | 0 | 100 | ADC4 |
| 1 | 0 | 101 | ADC5 |
| 1 | 0 | 110 | ADC6 |
| 1 | 0 | 111 | ADC7 |

Comparator analogic

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---|------|---|---|------|-------|-------|-------|--------|
| (0x7B) | - | ACME | - | - | MUX5 | ADTS2 | ADTS1 | ADTS0 | ADCSRB |
| Read/Write | R | R/W | R | R | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Intrare multiplexată pentru comparatorul analogic (ATmega2560 / MEGA)

ACME **ADEN** **MUX5** **MUX2..0** Intrare negativă pentru AC (AIN-)

| | | | | |
|---|---|---|-----|-------|
| 0 | x | x | xxx | AIN1 |
| 1 | 1 | x | xxx | AIN1 |
| 1 | 0 | 0 | 000 | ADC0 |
| 1 | 0 | 0 | 001 | ADC1 |
| 1 | 0 | 0 | 010 | ADC2 |
| 1 | 0 | 0 | 011 | ADC3 |
| 1 | 0 | 0 | 100 | ADC4 |
| 1 | 0 | 0 | 101 | ADC5 |
| 1 | 0 | 0 | 110 | ADC6 |
| 1 | 0 | 0 | 111 | ADC7 |
| 1 | 0 | 1 | 000 | ADC8 |
| 1 | 0 | 1 | 001 | ADC9 |
| 1 | 0 | 1 | 010 | ADC10 |
| 1 | 0 | 1 | 011 | ADC11 |
| 1 | 0 | 1 | 100 | ADC12 |
| 1 | 0 | 1 | 101 | ADC13 |
| 1 | 0 | 1 | 110 | ADC14 |
| 1 | 0 | 1 | 111 | ADC15 |

Comparator analogic

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|------------|-------------|------------|------------|-------------|-------------|--------------|--------------|-------------|
| 0x30 (0x50) | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | ACSR |
| Read/Write | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | N/A | 0 | 0 | 0 | 0 | 0 | |

ACSR – Registrul de stare și control al comparatorului analogic

- **Bit 7 – ACD: Analog Comparator Disable**

ACD \leftarrow 1, Comparatorul analogic este deconectat (reduce consumul)

- **Bit 6 – ACBG: Analog Comparator Bandgap Select**

ACBG \leftarrow 1, tensiune de referință fixă (bandgap) la (AN+)

ACBG \leftarrow 0, tensiune de referință externă pe pinul AIN0 conectată la (AN+)

- **Bit 5 – ACO: Analog Comparator Output**

Ieșirea comparatorului analogic este sincronizată și conectată la bitul ACO (sincronizarea introduce o întârziere de 1...2 perioade de ceas)

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

ACI \leftarrow 1, setat de hardware, când ieșirea comparatorului trebuie să genereze întrerupere conform selecției din ACIS1 și ACIS0. Întreruperea este generată dacă ACIE și SREG(I) sunt 1.

ACI \leftarrow 0, setat de hardware dacă se execută AC-ISR, sau de software

Comparator analogic

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|------------|-------------|------------|------------|-------------|-------------|--------------|--------------|-------------|
| 0x30 (0x50) | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | ACSR |
| Read/Write | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | N/A | 0 | 0 | 0 | 0 | 0 | |

ACSR – Registrul de stare și control al comparatorului analogic

- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

ACIE \leftarrow 1 și SREG(I) \leftarrow 1, Activează întreruperea comparatorului analogic

ACIE \leftarrow 0, Dezactivează întreruperea pentru comparatorul analogic

- **Bit 2 – ACIC: Analog Comparator Input Capture Enable**

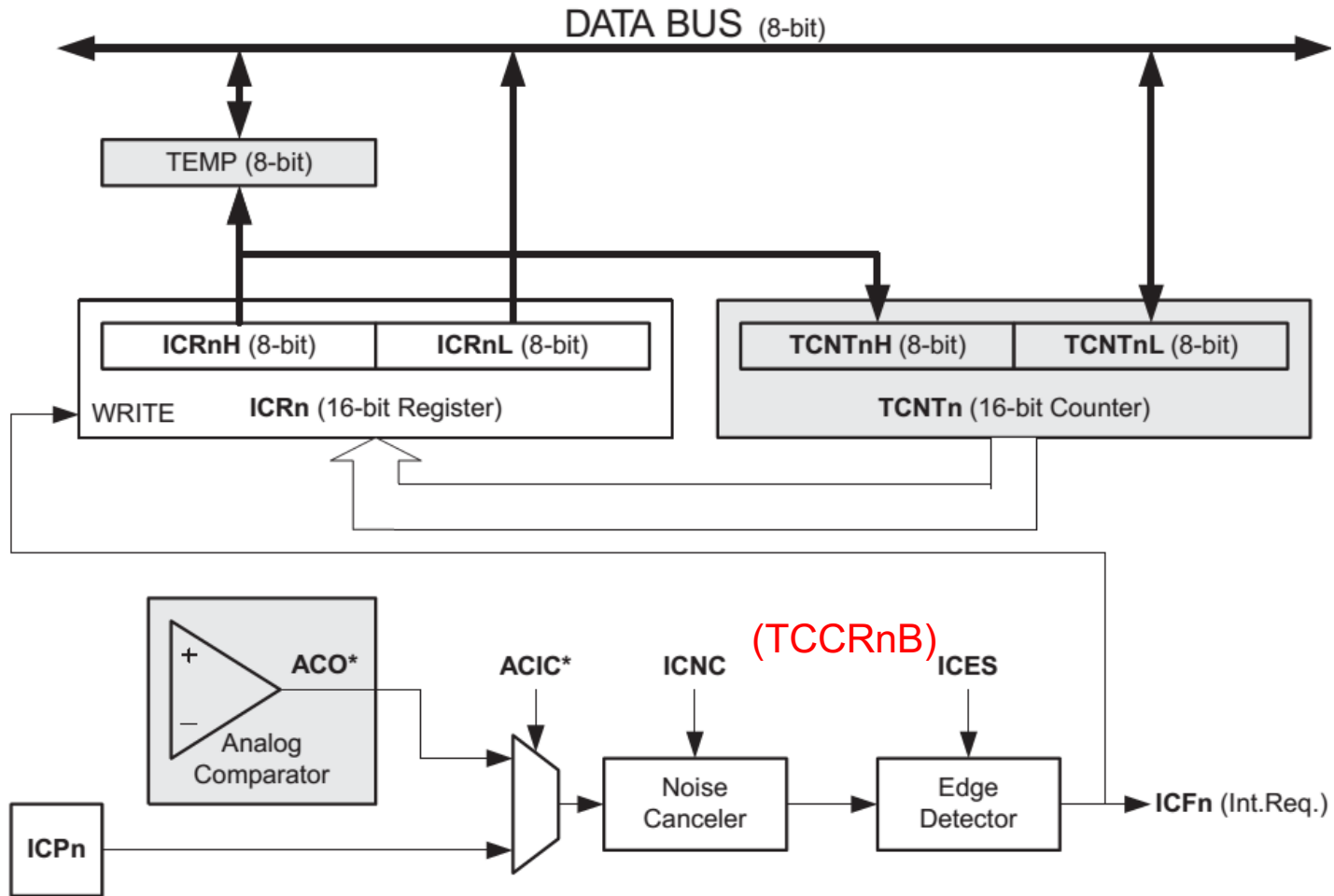
ACIC \leftarrow 1, **întreruperea Input Capture a temporizatorului Timer/Counter1** va fi declanșată de comparatorul analogic dacă bitul ICIE1 din registrul (TIMSK) este activat

ACIC \leftarrow 0, Comparatorul analogic nu este conectat la Timer/Counter1 input capture

- **Biții 1, 0 – ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

| ACIS1 | ACIS0 | Modul de întrerupere |
|-------|-------|---|
| 0 | 0 | Întrerupere pe comutarea ieșirii comparatorului |
| 0 | 1 | Rezervat |
| 1 | 0 | Întrerupere pe front descrescător la ieșirea comparatorului |
| 1 | 1 | Întrerupere pe front crescător la ieșirea comparatorului |

Comparatorul analogic și Timer1



Captură: $ICF1 \leftarrow 1$ și $WRITE \leftarrow 1 \Rightarrow ICR1 = TCNT1$;

ICR1 \Rightarrow Timp precis pentru evenimente externe (se pot măsura frecvențe, factor de umplere, etc)

Comparatorul analogic și Timer1

Exemplu: măsurarea capacității unui condensator

$$v(t) = V_{cc}(1 - \exp(-t/T)) \quad (1)$$

$$T = R2 * C \quad (2)$$

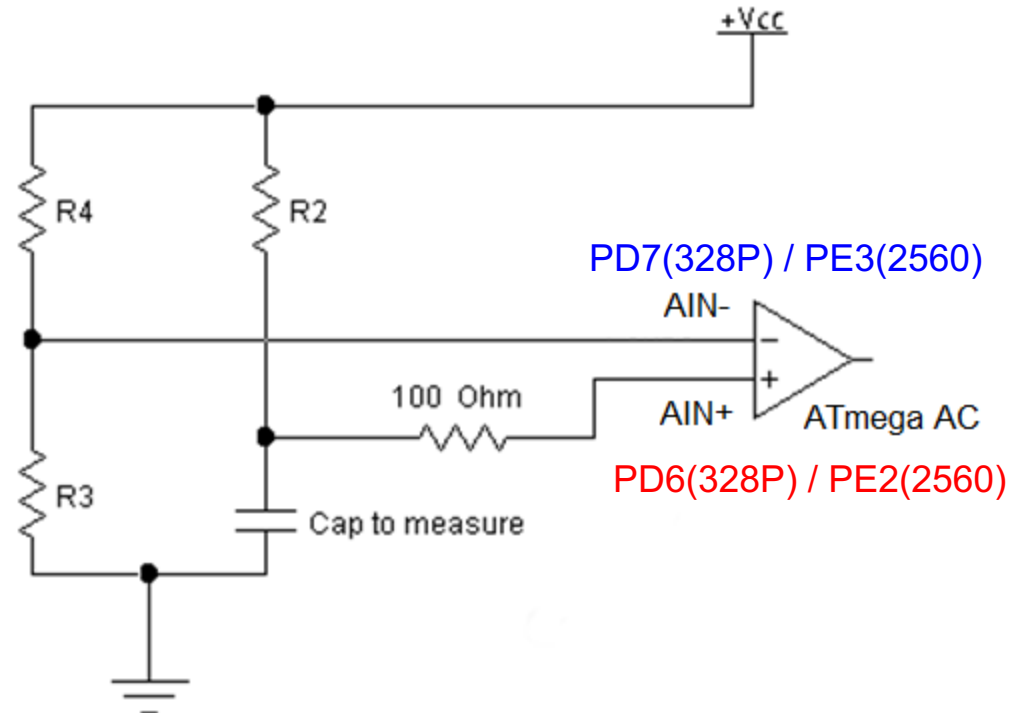
$$R2 \gg (100 \text{ ohm})$$

Algoritm:

1. Setează PORTnx(AIN-) intrare
2. Configurare AC și Timer1
3. Setează PORTny(AIN+) ca ieșire și scrie "0" (descarcă condensator)
4. Setează PORTny(AIN+) ca intrare și pornește Timer1.
Condensatorul începe încărcarea

ISR pentru timer 1 capture:

1. Citește registrul ICR1
2. Convertește ICR1 în sec $\Rightarrow t$
3. Calculează C din (1) + (2) + (3)



ISR va fi declanșată când tensiunea pe condensator, V_+ , este egală cu V_- :

$$v(t) = V_{cc} * R3 / (R3 + R4) \quad (3)$$

Convertor analog/ digital (ADC)

Atmega 328P

- Rezoluție 10-biți
- 8 Canale de intrare “single ended”
- Intrare pentru senzor de temperatură
- Ajustare opțională la stânga pentru rezultatul conversiei
- 0 ... V_{CC} Domeniul de tensiuni de intrare
- 1.1 V tensiune internă de referință
- Conversie continuă sau la cerere
- Poate genera întreruperi la finalizarea conversiei
- Reducerea zgomotului prin mod Sleep

Rezultatul unei conversii single end:

Rezultatul unei conversii diferențiale:

ATmega2560

- Rezoluție 10-biți
- 8 Canale de intrare “single ended”
- 14 canale de intrare diferențială
- 4 intrări diferențiale cu amplificare opțională 10× sau 200×
- Ajustare opțională la stânga pentru rezultatul conversiei
- 0 ... V_{CC} Domeniul de tensiuni de intrare
- Tensiune de referință internă selectabilă 2.56V sau 1.1 V
- Conversie continuă sau la cerere
- Poate genera întreruperi la finalizarea conversiei
- Reducerea zgomotului prin mod Sleep

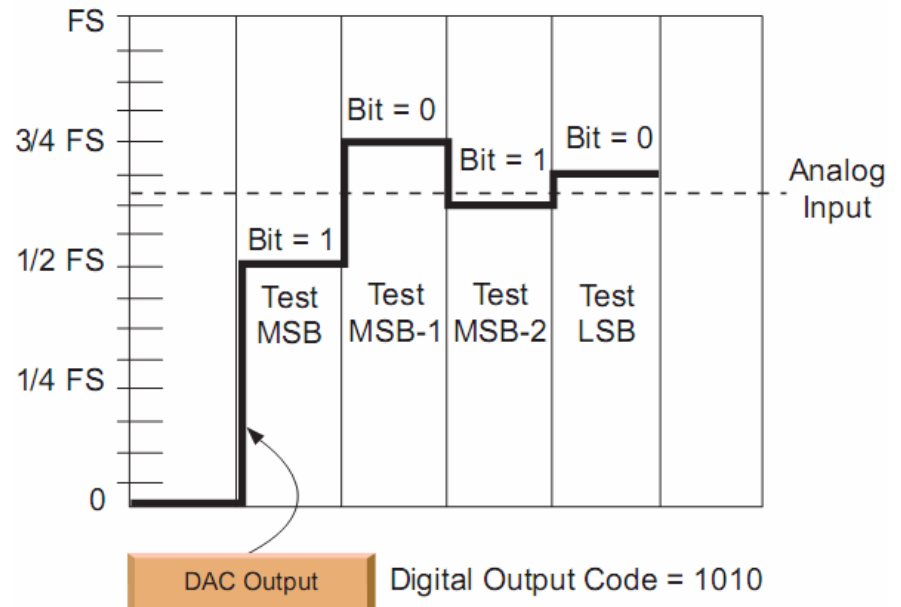
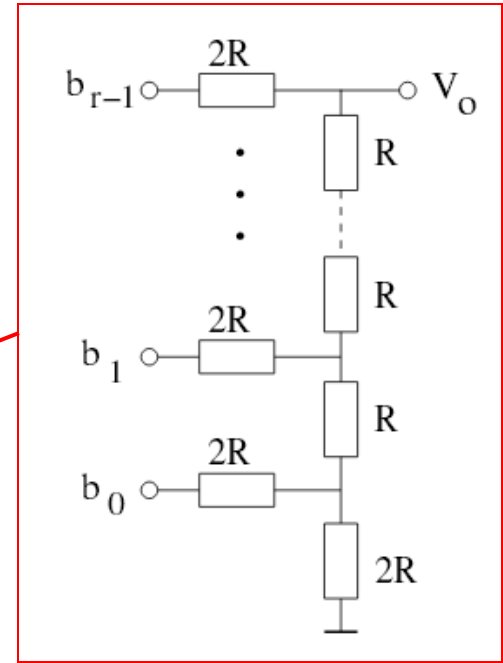
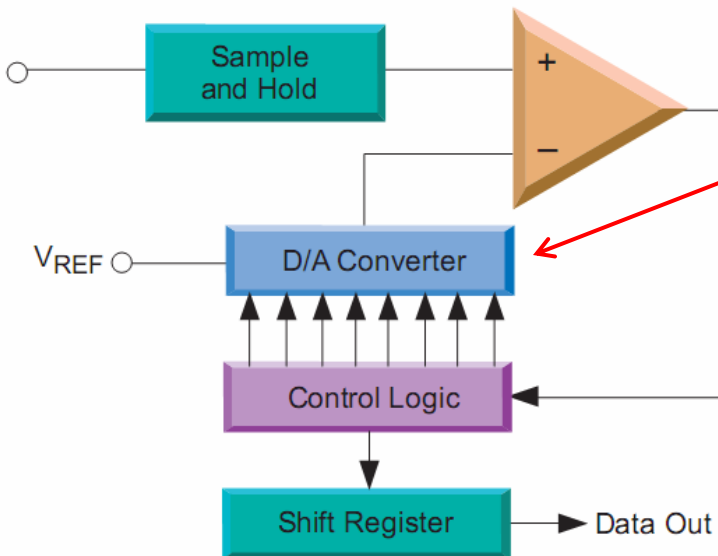
$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

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

ADC

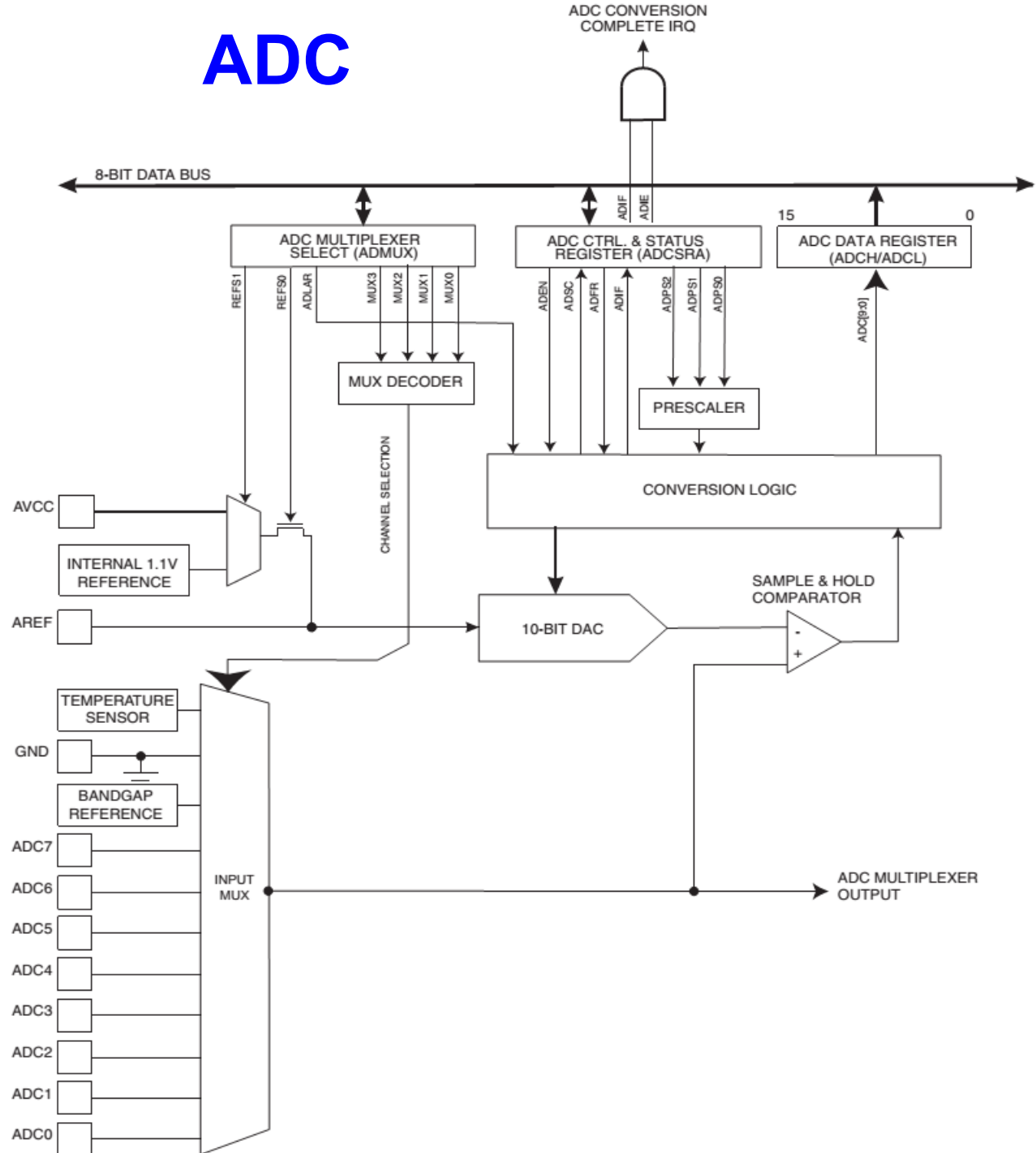
Principiul ADC:

- Comparații succesive cu tensiuni de referință



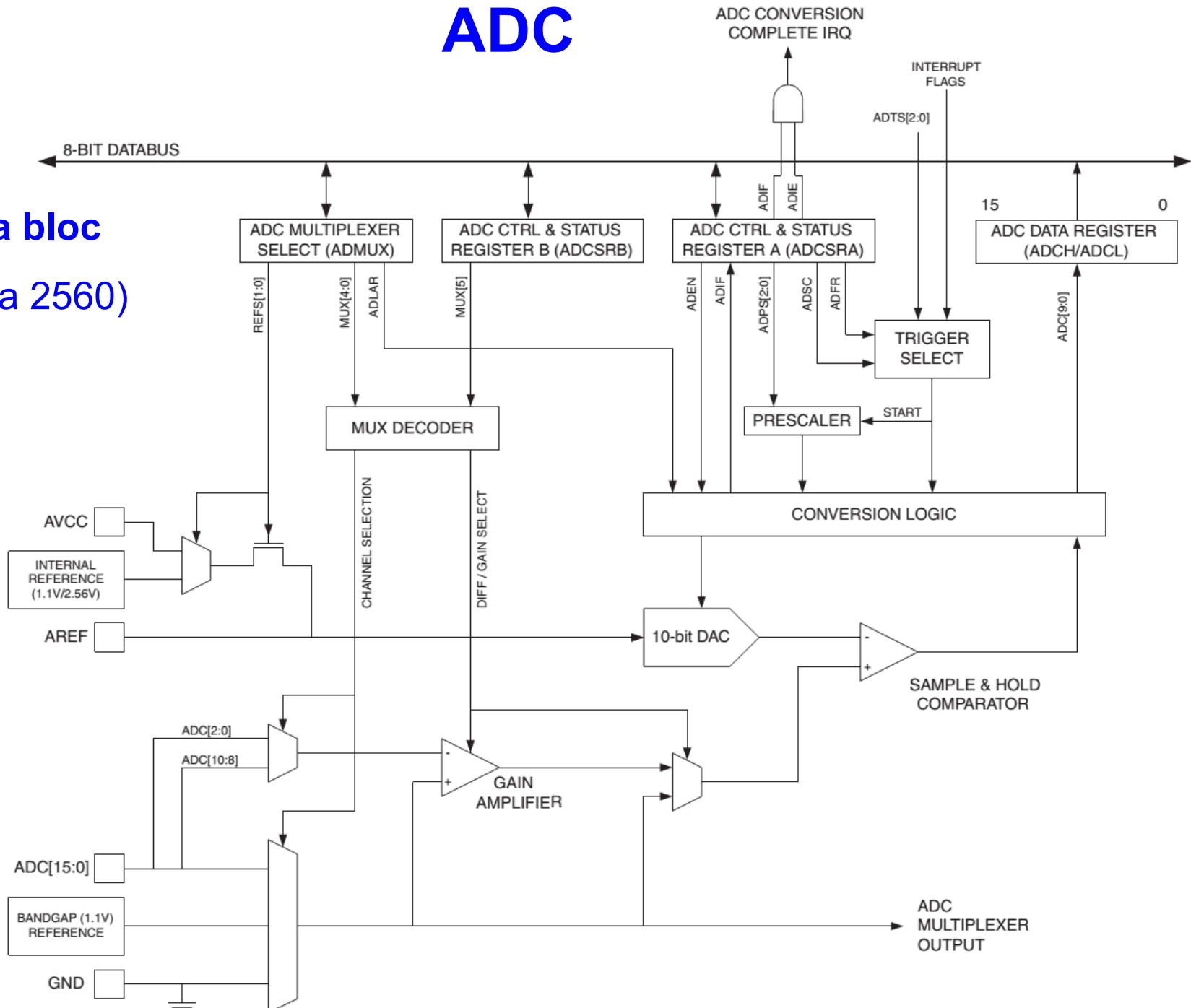
ADC

Schema bloc (Atmega 328P)



ADC

Schema bloc (ATmega 2560)



ADC

Configurare ADC

ADCSRA – Registrul A de stare și control al ADC



ADEN – activare ADC (ADEN=1)

ADIE – activare întrerupere ADC (ADIE= 1 & SREG(I)=1 ⇒ ADC IRQ activat)

ADPS2 ..0 – divizarea frecvenței ceasului

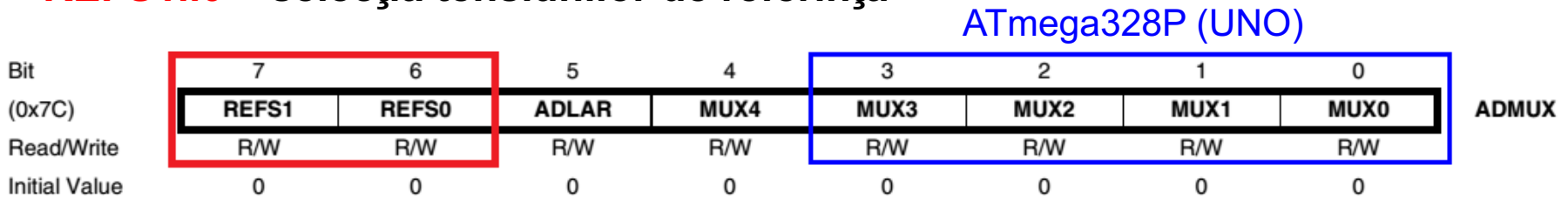
| ADPS2 | ADPS1 | ADPS0 | Division Factor |
|-------|-------|-------|-----------------|
| 0 | 0 | 0 | 2 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 |
| 1 | 0 | 0 | 16 |
| 1 | 0 | 1 | 32 |
| 1 | 1 | 0 | 64 |
| 1 | 1 | 1 | 128 |

ADC

Configurare ADC

ADMUX – selecția multiplexorului intrărilor ADC

REFS1..0 – selecția tensiunilor de referință



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 |

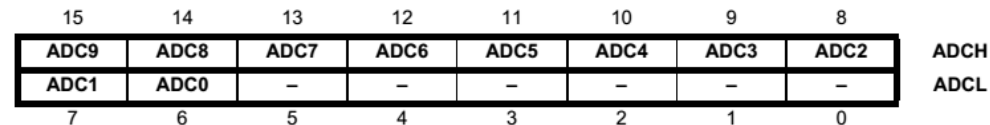
ATmega328P (UNO)

| 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 |

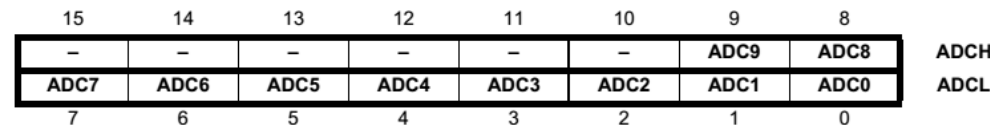
ADLAR: Ajustare rezultat ADC la stânga

ADLAR ← 1, rezultat aliniat la stânga (dacă doar ADCH este citit – rezultat pe 8 biți – rezoluție mai mică)

$$ADCH = V_{in} * 256 / V_{ref}$$



ADLAR ← 0, rezultat aliniat dreapta



ADC

MUX5:0: Selecția canalului analogic și a amplificării

ATmega328P

ATmega2560

| 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) |

Note: 1. For Temperature Sensor.

Pentru toate variantele disponibile, consultați datasheet.

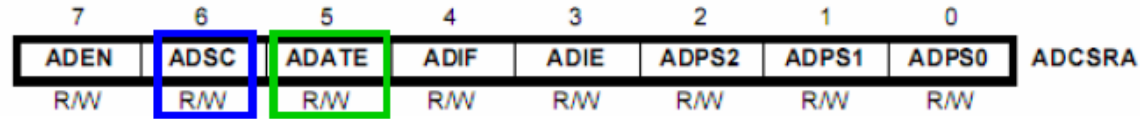
| MUX5:0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain | |
|-----------------------|--------------------|-----------------------------|-----------------------------|------|----|
| 000000 | ADC0 | N/A | N/A | | |
| 000001 | ADC1 | | | | |
| 000010 | ADC2 | | | | |
| 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 | | | |

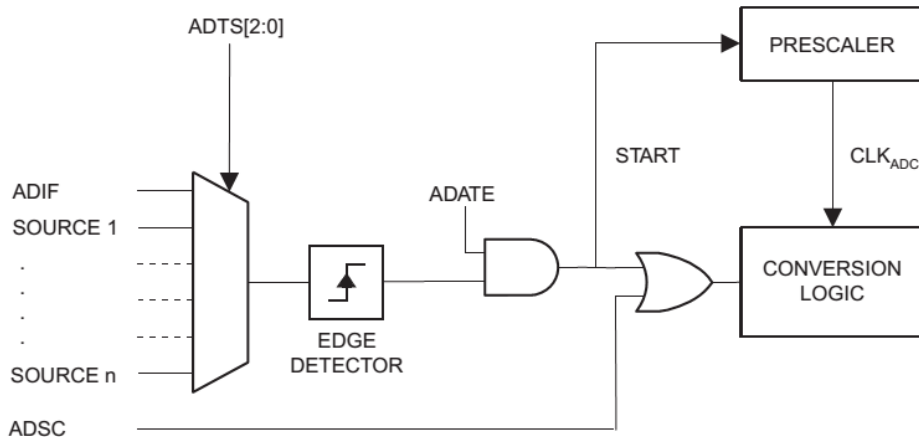
ADC

Pornirea conversiei

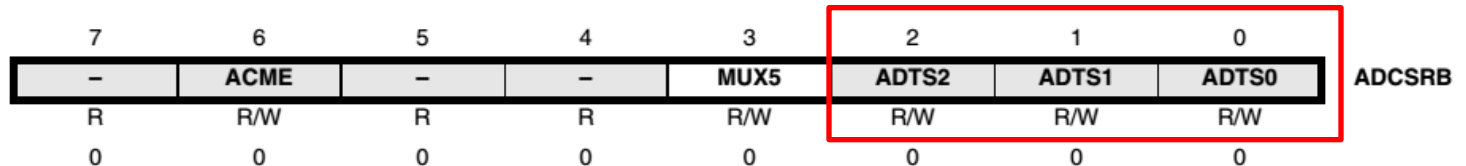


- La cerere prin: **ADSC = 1** (rămâne pe 1 până la finalizarea conversiei)
- Automat: **ADATE = 1** (Auto Trigger Enable)
 - O nouă conversie este pornită la finalul celei curente
 - Alte surse pot declanșa conversia automată:

ADC Auto Trigger Logic

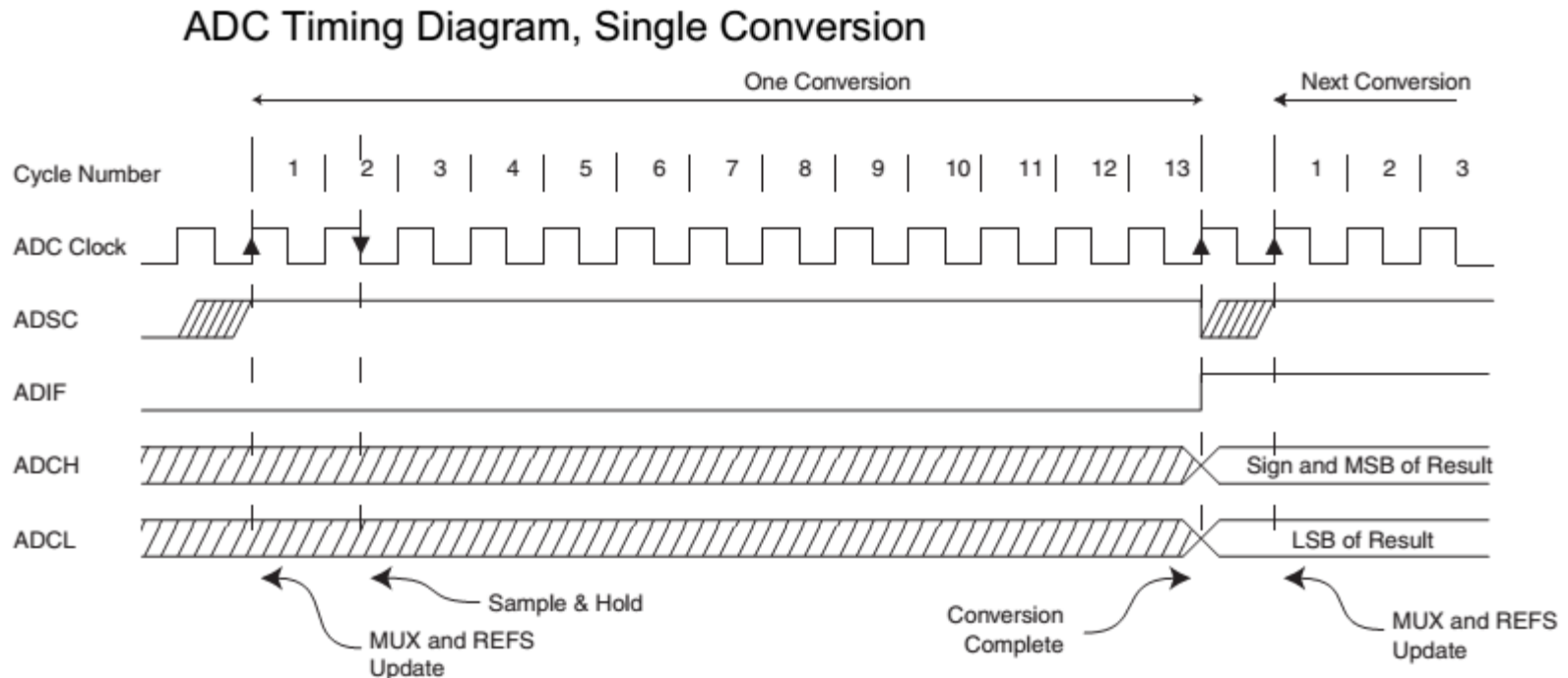


| ADTS2 | ADTS1 | ADTS0 | Trigger Source |
|-------|-------|-------|--------------------------------|
| 0 | 0 | 0 | Free Running mode |
| 0 | 0 | 1 | Analog Comparator |
| 0 | 1 | 0 | External Interrupt Request 0 |
| 0 | 1 | 1 | Timer/Counter0 Compare Match A |
| 1 | 0 | 0 | Timer/Counter0 Overflow |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



ADC

Timpi de conversie, diagrame de timp



ADC Conversion Time

| 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 |

ADC

Cod exemplu:

rcall ADC_init

loop:

```
rcall start_ADC_conversion ; Pornire conversie  
rcall wait_ADC_complete ; Așteaptă finalizare conversie  
rcall ADC_read ; Citire rezultat în 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 ; Activare ADC, viteză maximă (clock div. ratio = 2)  
out ADCSRA, r16
```

ret

start_ADC_conversion:

```
sbi ADCSRA, ADSC ; ADC start, setează bitul ADSC din ADCSRA
```

ret

wait_ADC_complete:

```
sbic ADCSRA, ADSC ; Când ADSC=0, conversia e finalizată  
rjmp wait_ADC_complete
```

ret

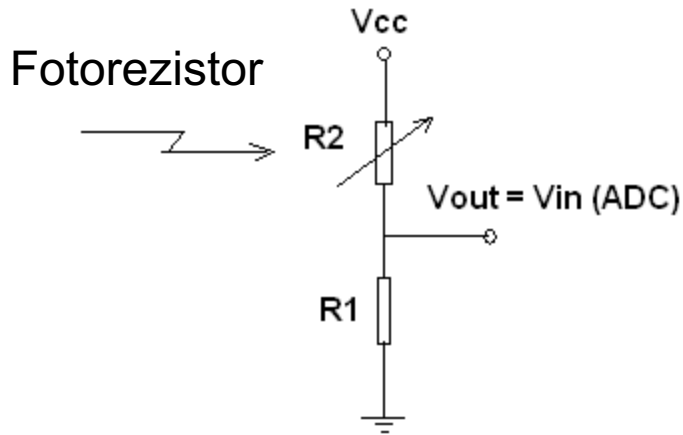
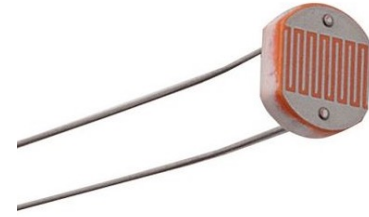
ADC_read:

```
in r16, ADCH ; ADCH – temperatura pe 8 biți
```

ret

ADC

Exemplu 2: măsurare intensitate luminoasă



$R2 = 200 \Omega$ (lumină puternică) $1.4 \text{ M}\Omega$ (întuneric)

$R1 = \text{constant}$ (ex: 20 K)

$$V_{OUT} \propto V_{CC} \frac{R_1}{R_1 \propto R_2}$$

Mod intrare single ended, $ADLAR = 1$ (rezoluție mică): $ADCH = V_{in} * 256 / V_{ref}$

Calibrare senzor:

- Citire ADCH pentru întuneric: $ADCH_{MIN}$
- Citire ADCH pentru lumină puternică: $ADCH_{MAX}$

Măsurare:

- Calcul luminozitate relativă B [%]:

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

ADC

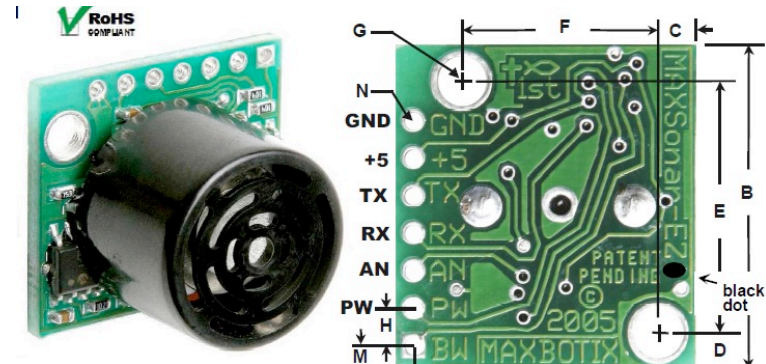
Exemplu 3: măsurare distanță cu senzor ultrasonic

LV-MaxSonar®-EZ0™ High Performance Sonar Range Finder

(http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf)

AN – ieșire analogică scalată cu ($V_{CC}/512$) per inch. O tensiune de alimentare de 3.3V produce $\sim 6.4\text{mV/in} \approx 2.56\text{ mV/cm}$

Domeniu de distanțe: 6-in (15 cm) ... 254 in (645 cm)
cu rezoluție de 1 inch.



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

ADC_Init:

```
ldi r16, 0b11000011 ; Vref=2,56 V internal, ADLAR=0 (Data Shift right - full 1024 bit resolution), ADC3 single ended
```

```
out ADMUX, r16
```

```
ldi r16, 0b10000000 ; Activare ADC, viteză maximă
```

```
out ADCSRA, r16
```

ret

ADC_read:

```
in r20, ADCL //blochează accesul ADC la registrul de date
```

```
in r21, ADCH //eliberează accesul ADC la registrul de date
```

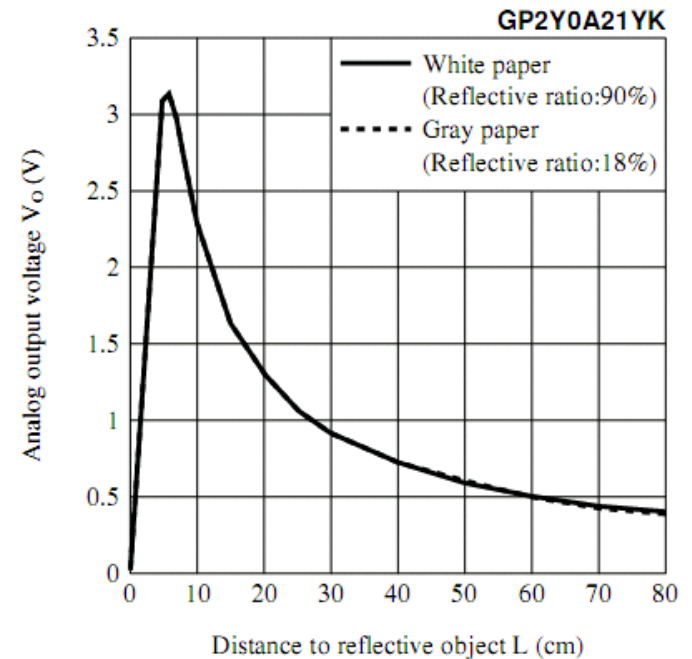
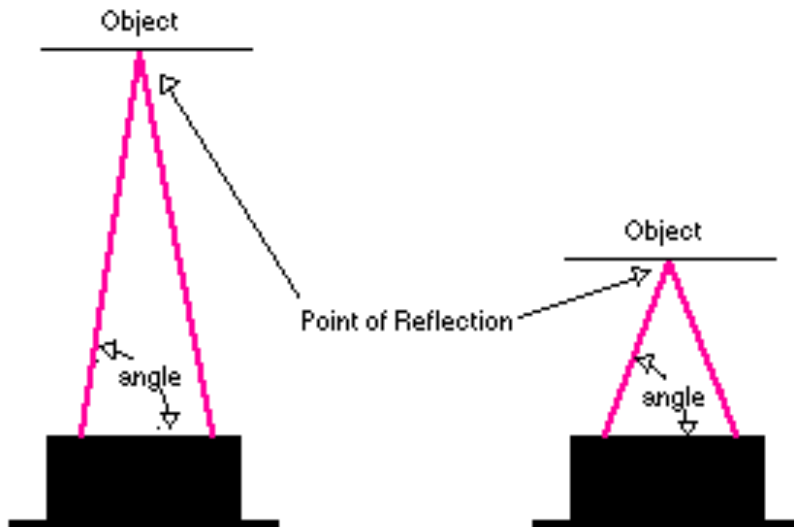
```
// r21:r20 = d[cm]
```

ret

ADC

SHARP GP2XX, familie de senzori pentru distanța bazați pe reflexia IR

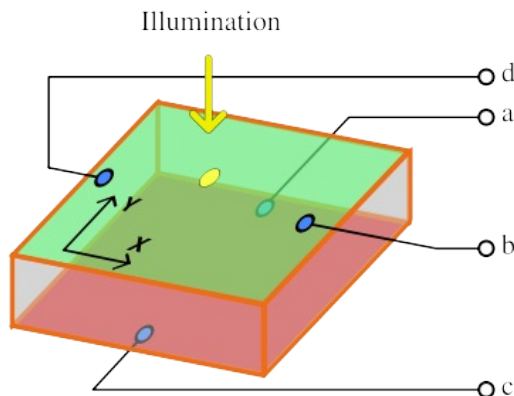
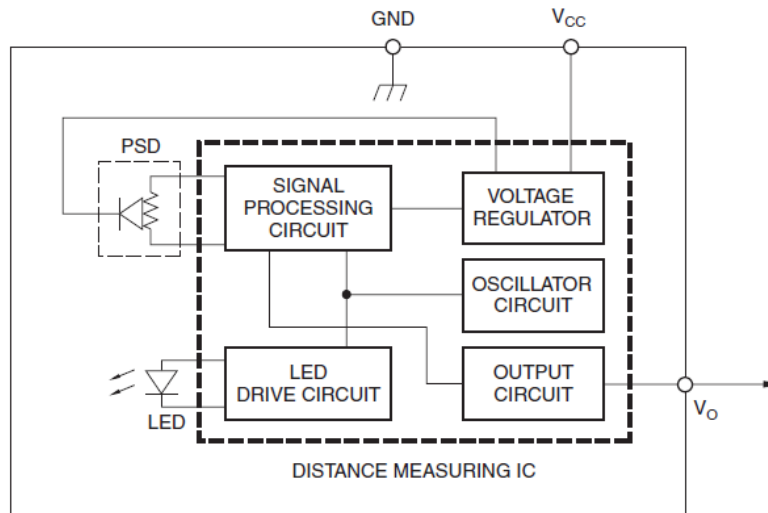
- Folosește triangulația pentru calculul distanței
- Se măsoară unghiul sub care se întoarce raza emisă
- Iesire analogică, **neliniară**
- Cost redus
- Usor de montat, robust



ADC

SHARP GP2XX, familie de senzori pentru distanța bazați pe reflexia IR

- Folosește o fotodioda sensibilă la poziție
- Pe baza răspunsului ei, se determină unghiul de reflexie



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

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

ADC

Accelerometru ADXL335

- Masoara acceleratia pe 3 axe, de la -3... 3 g
- Alimentare intre 1.8 ... 3.6 V
- Iesire pentru 0 G: $V_{cc}/2$
- Sensibilitate tipica pentru $V_{cc}=3.3V$: 300 mv/G

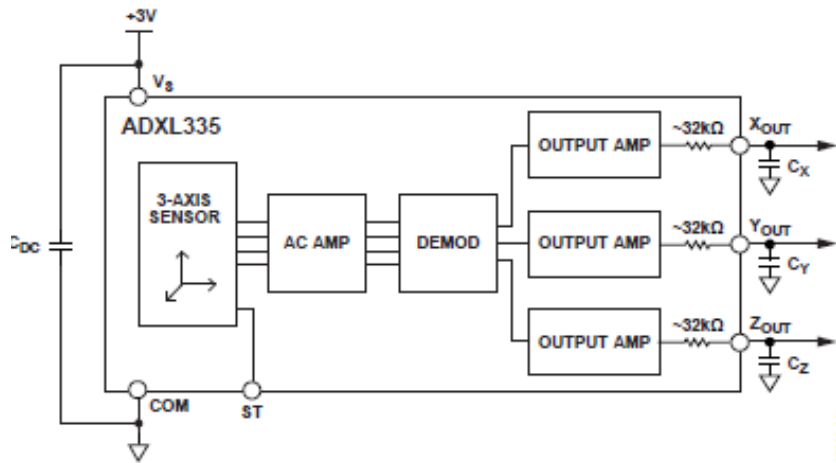
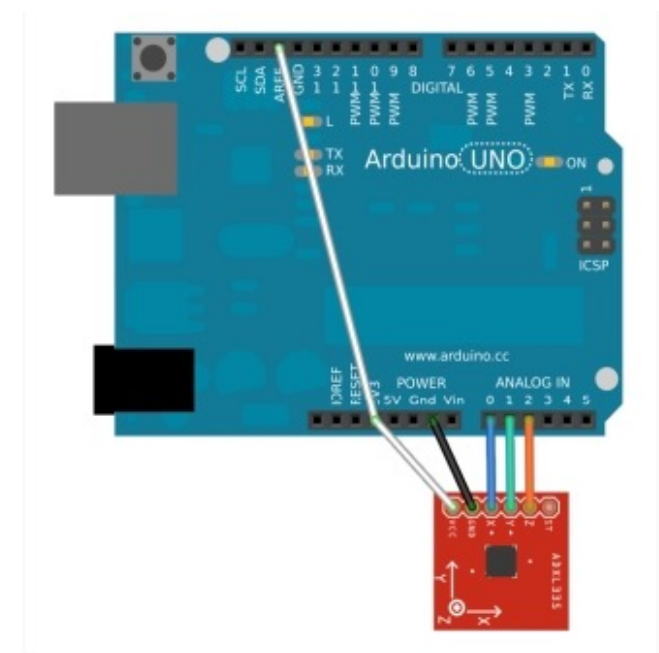
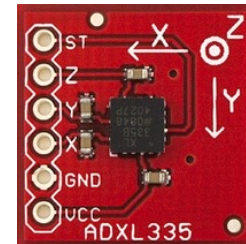
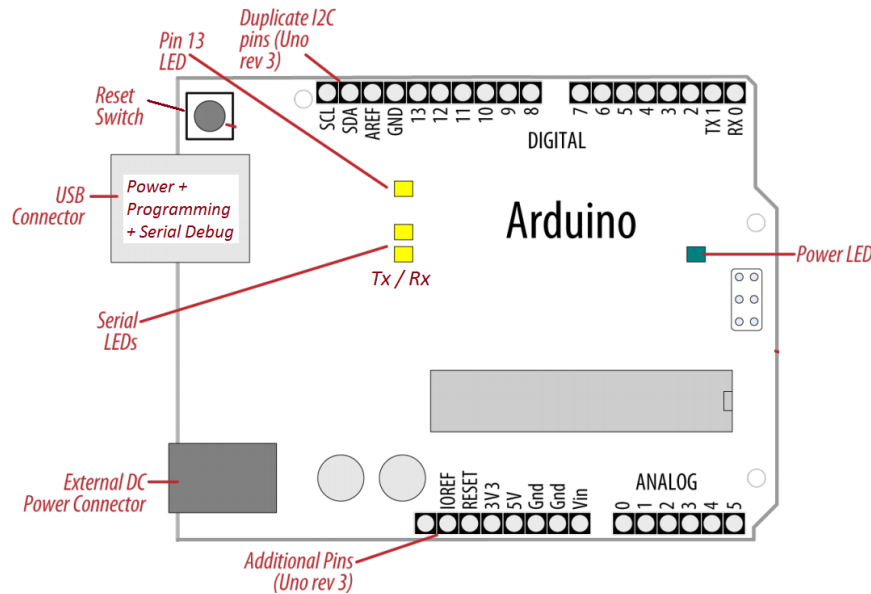


Figure 1.



| | |
|-----------------|--------------|
| 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

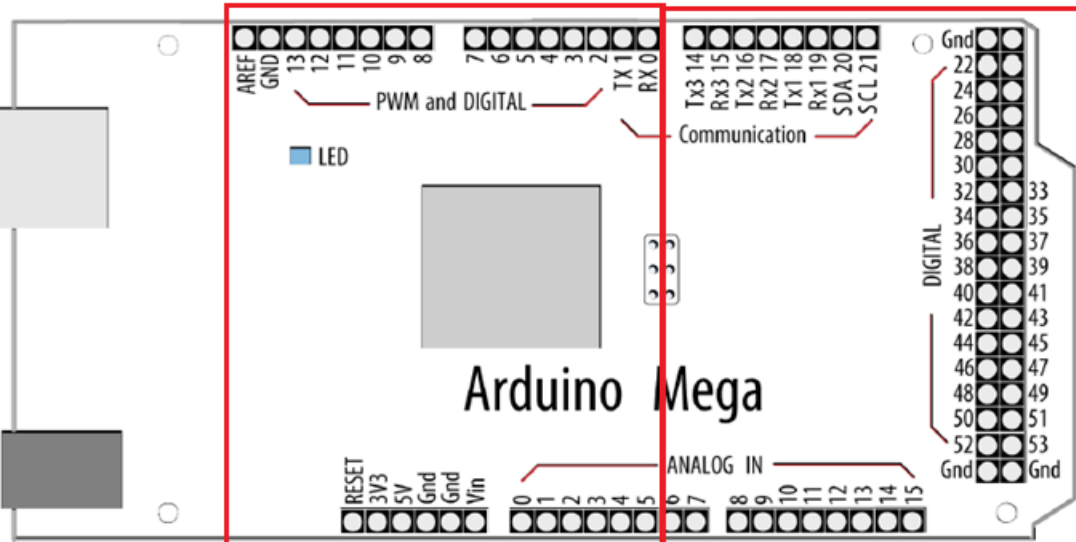


Arduino UNO: A0 .. A5

Arduino MEGA: A0 .. A15

- Pinii analogici sunt intrari pentru convertorul ADC al microcontrollerului.
- ADC are rezolutia de 10 biti, returnand valori intre 0 si 1023

Pin Layout identical with UNO



Pin Layout specific to MEGA

Alti pini

AREF (intrare) – tensiune de referinta externa pentru ADC

IOREF (iesire) – tensiune de referinta pentru shield-uri

Procesare semnal analogic cu Arduino

- **Functia principala a pinilor analogici:** **citirea semnalelor analogice**
- Pinii analogici au si functia de pin digital de uz general, ca si pinii digitali.
Exemplu:

```
pinMode(A0, OUTPUT);  
digitalWrite(A0, HIGH);
```

- Pinii analogici au de asemenea rezistente **pull up resistors**, care functioneaza in acelasi fel ca rezistentele pinilor digitali. Ele sunt activate scriind HIGH pe pinul configurat ca intrare.

```
digitalWrite(A0, HIGH); // activare pullup la A0 configurat ca input.
```

Activarea unei rezistente pull up va influenta valorile citite cu analogRead() !!!

Functii:

analogRead(pin) – citeste o valoare de pe un pin analogic

analogReference(type) - configureaza tensiunea de referinta care va fi folosita pentru intrarea analogica (i.e. valoarea maxima a tensiunii de intrare masurabila pe pin-ul analogic)

Procesare semnal analogic cu Arduino

analogReference(type) – configureaza tensiunea de referinta care va fi folosita pentru intrarea analogica.

type - care referinta este folosita:

- DEFAULT: tensiunea referinta implicita, de 5 V (pentru UNO & MEGA)
- INTERNAL: tensiune interna de referinta, 1.1 V la UNO (*nu exista la Arduino Mega*)
- INTERNAL1V1: tensiune interna de referinta 1.1V (*doar Arduino Mega*)
- INTERNAL2V56: tensiune interna 2.56V (*doar Arduino Mega*)
- EXTERNAL: tensiune de referinta externa, aplicata la pinul AREF (**0 ... 5V**).

Dupa schimbarea tensiunii de referinta, prima citire cu analogRead() poate fi eronata !!!

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(). In caz contrar, veti pune in contact tensiunea de referinta interna, generata in mod activ, cu tensiunea externa, putand cauza scurtcircuit si distrugerea microcontrollerului. !!!

Procesare semnal analogic cu Arduino

int *digital_value* **analogRead**(*pin*) – citește o valoare de pe pinul analogic specificat

- O valoare analogică între 0 .. RANGE va produce un număr *digital_value* între 0 și 1023.
- Rezoluția de măsurare este deci: RANGE volți / 1024 unități.
- Pentru referința DEFAULT (5V) rezoluția devine:
resolutionADC = .0049 volți (4.9 mV) / unitate.
- Pentru a converti valoarea citită *digital_value* la tensiunea analogică:
Voltage = ***resolutionADC*** * ***digital_value***
- Durează aproximativ 100 microsecunde (0.0001 s) pentru a citi o intrare analogică, astfel încât rata maximă de citire este 10000 valori pe secundă.

Dacă pinul analogic nu este conectat la nimic, valoarea returnată de `analogRead()` va fluctua în funcție de mai mulți factori (e.g. ce tensiuni sunt pe ceilalți pini analogici, apropierea mâinii de placă...) !!!

Procesare semnal analogic cu Arduino

Exemplu – Citirea tensiunii unui potentiometru conectat la intrarea analogica (<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 * resolution;  // 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 temperatura folosind LM50 <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:

<http://www.robofun.ro/senzori/vreme/senzor-temperatura-brick>

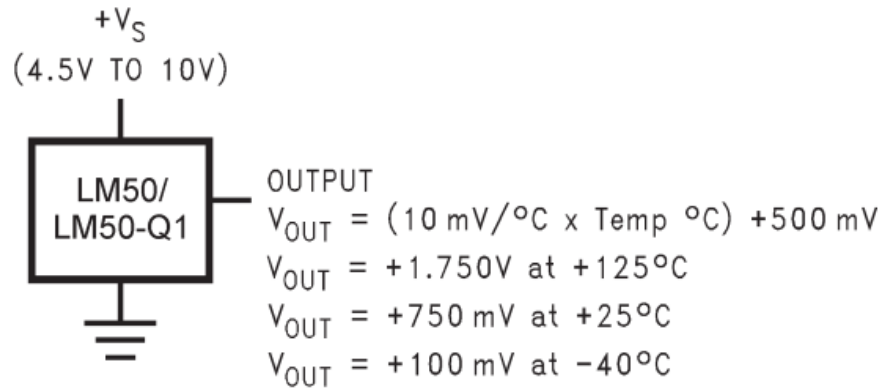
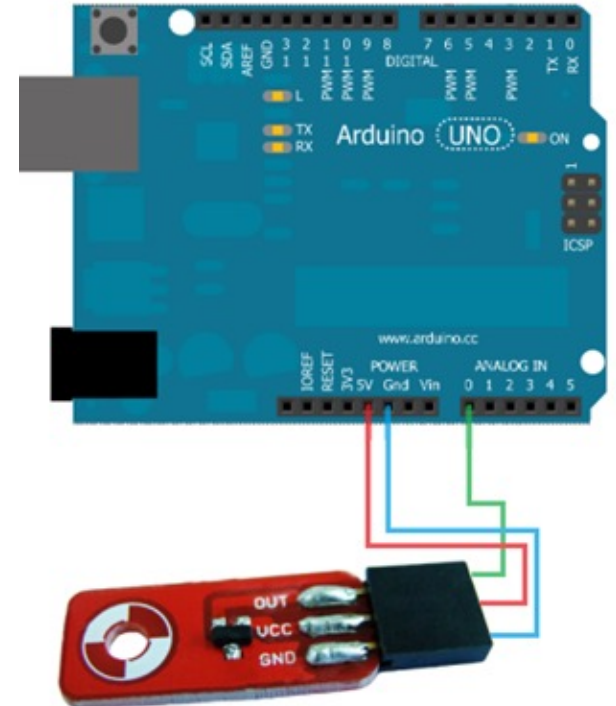


Figure 2. Full-Range Centigrade Temperature Sensor Application (-40°C to $+125^{\circ}\text{C}$)



Procesare semnal analogic cu Arduino

Exemplu – Citire temperatura de la senzor, face media a 10 citiri consecutive, si trimite catre 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 * resolution;
    float tempCelsius = (voltage - 0.5) / resolutionSensor ; // scade deplasament, converteste in grade C
    sumTemp = sumTemp + tempCelsius;                          // suma temperaturilor
  }
  return sumTemp / (float)count;                             // media returnata
}
```

Procesare semnal analogic cu Arduino

Example – Masurare distante cu sonarul LV EZ0 (rezolutie 10mV / inch \cong 0.01V)

http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf

```
const int sensorPin = 1;           // lesire 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 * resolution;
    float distance = voltage / resolutionSensor; // conversie tensiune in distanta
    sumDist = sumDist + distance;
  }
  return sumDist / (float)count;
}
```