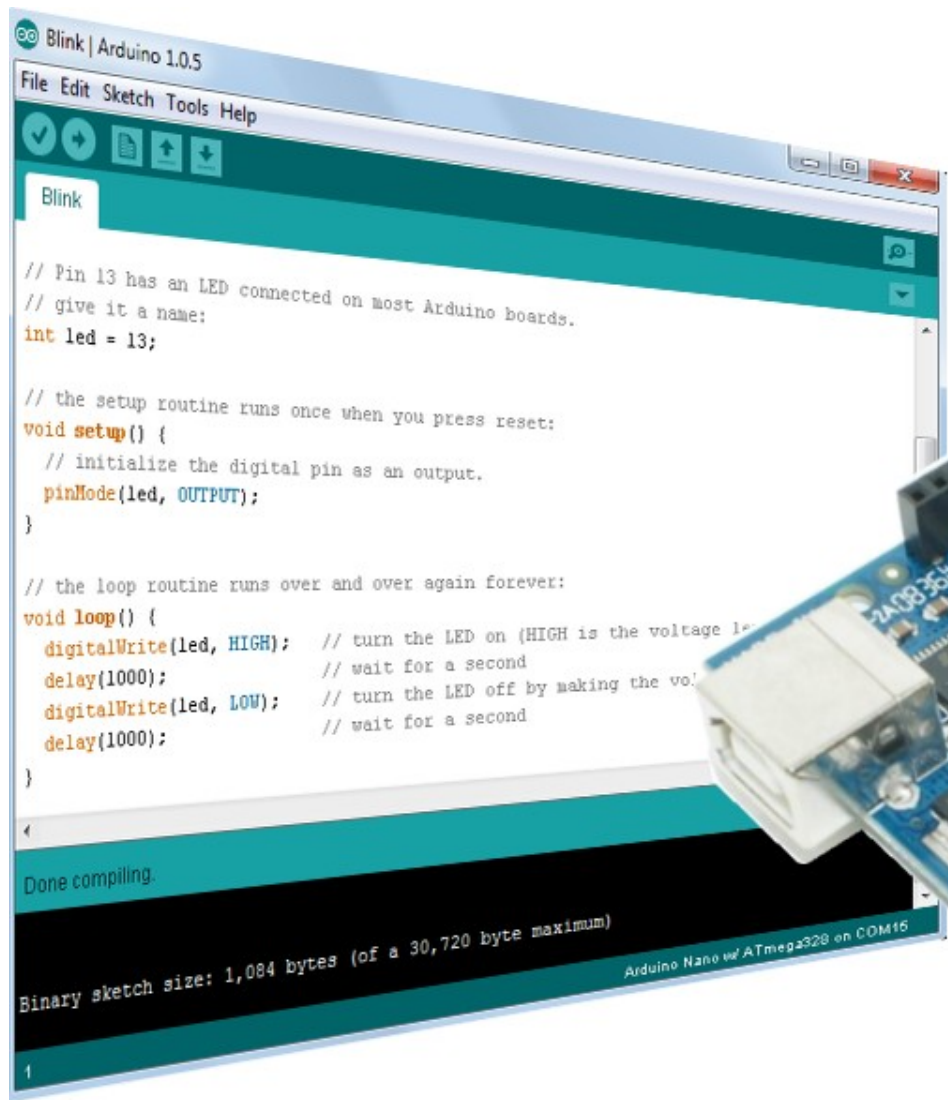


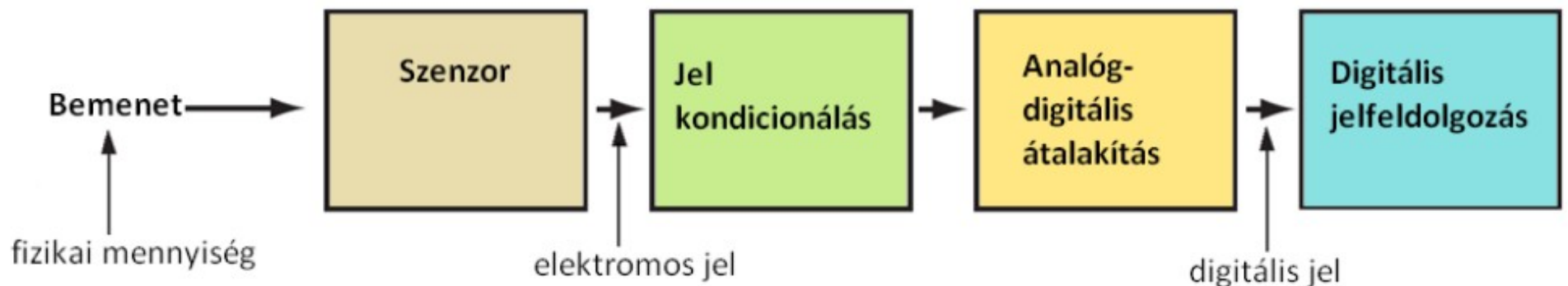
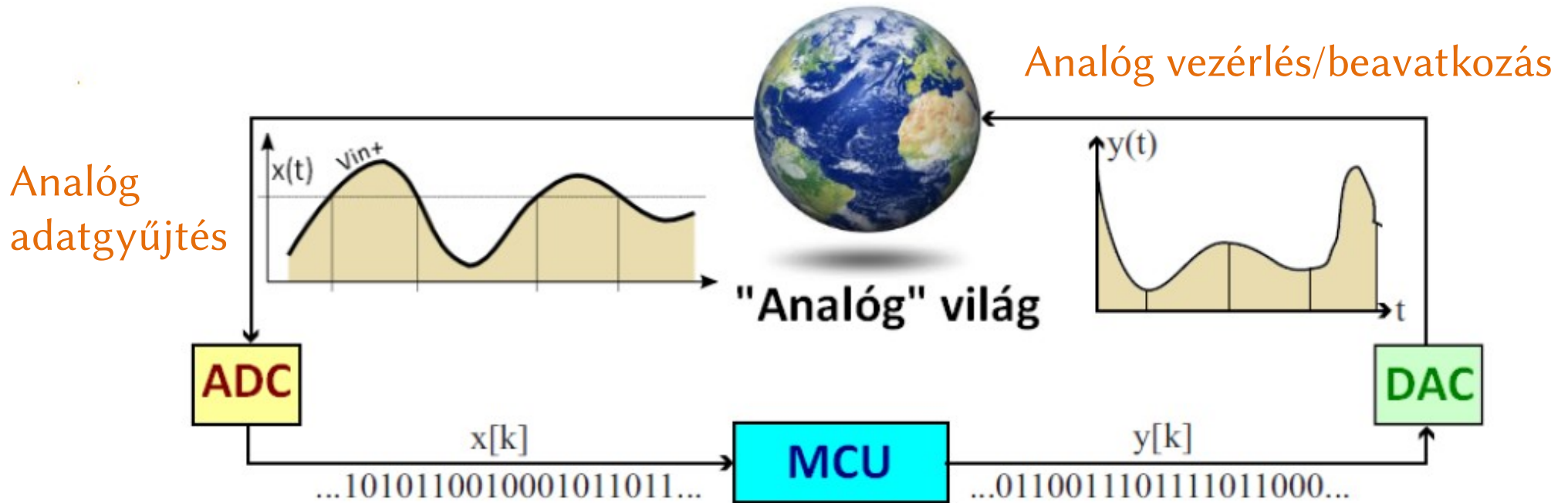
Arduino tanfolyam kezdőknek és haladóknak



4. Az analóg-digitális átalakító (ADC) használata

Analóg jelfeldolgozás

- Analóg világban élünk, de digitális mikrovezérlővel dolgozunk...



Az analóg adatgyűjtő ág elemei

❑ Szenzor:

- A folytonos fizikai mennyiségeket (pl. hőmérséklet, nyomás, páratartalom, sebesség, áramlási sebesség, elmozdulás, gyorsulás, szöggyorsulás) elektromos jellé alakítja (feszültséggé vagy árammá).

❑ Jel kondicionálás (szűrés, erősítés, stb.):

- A mérendő mennyiség elektromos jellé alakítása után még szűrésre, jel erősítésre, impedancia illesztésre is szükség lehet, hogy az analóg-digitális átalakító (ADC) bemeneti tartományába transzformáljuk az átalakítandó jelet.

❑ Analóg-Digitális Átalakító (ADC):

- *Bemenet:* a mérendő jel
- *Kimenet:* a mérendő jellel arányos számot reprezentáló digitális kód

Analóg–digitális átalakító (ADC)

- Az ADC feladata az, hogy diszkrét kódokká alakítsa a bejövő jelet
- A konverzió digitális értéke (N_{ADC}):

- ❖ Végkitérés: $N_{ADC} = 1023$, ha a bemenő jel $\geq V_{R+} - 1.5 \cdot \text{LSB}$
- ❖ Nulla: $N_{ADC} = 0$, ha a bemenő jel $\leq V_{R-} + 0.5 \text{ LSB}$
- ❖ Közbeeső értékekre:

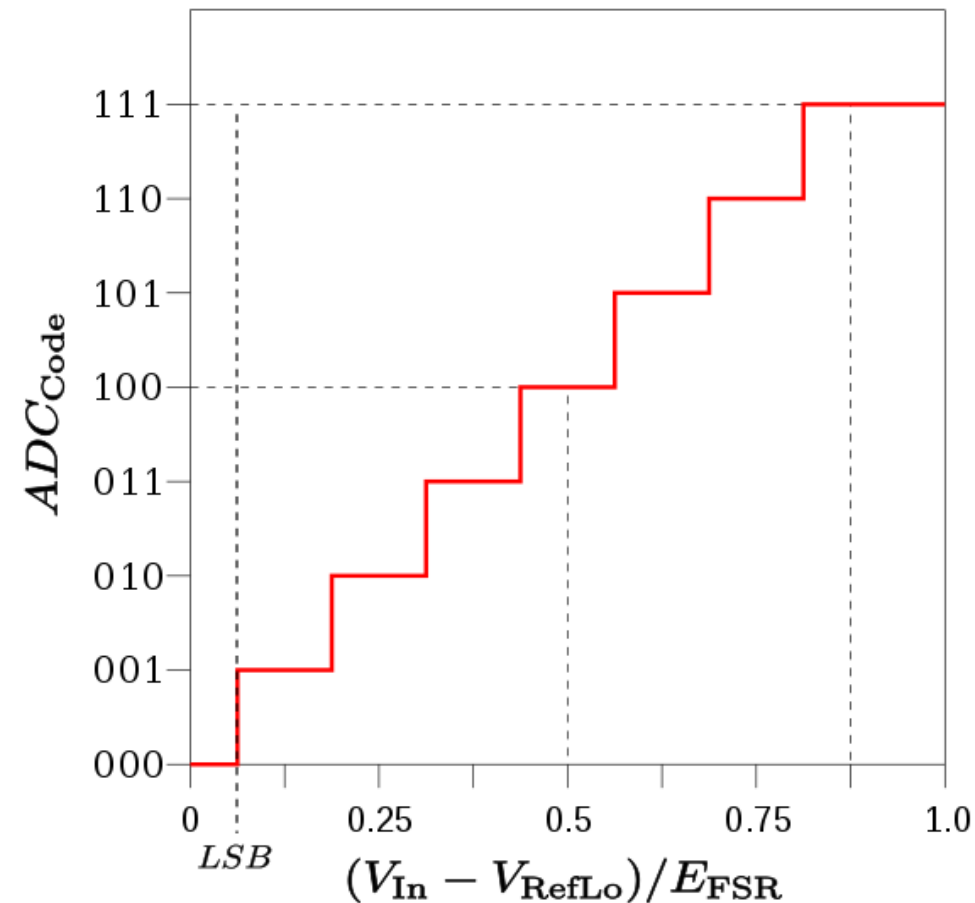
$$N_{ADC} = 1024 * (V_{IN} - V_{R-}) / (V_{R+} - V_{R-})$$

- A fenti képletből V_{IN} -t kifejezve, ezt kapjuk:

$$V_{IN} = (V_{R+} - V_{R-}) * N_{ADC} / 1024 + V_{R-}$$

- V_{R-} általában = 0

Egy 3-bites átalakító
ideális átviteli függvénye



Az ADC-t kezelő függvények

- **analogReference(*típus*)** – az analóg bemenetek viszonyítási (referencia) feszültségét konfigurálhatjuk vele
- A választható opciók:
 - DEFAULT** – a tápfeszültség a referencia (5V helyett inkább 4,75 V)
 - INTERNAL** – a belső 1,1 V-os referencia
 - EXTERNAL** – külső forrásból a V_{ref} lábra adhatunk feszültséget (0-5V)
- **analogRead(*pin*)** – elindít egy mérést a megadott analóg bemeneten (A0–A7) és a visszatérési érték a konverzió eredménye lesz (0 – 1023 közötti egész szám)

```
void setup() {
    analogReference(DEFAULT);    // Vcc a referencia
    Serial.begin(9600);         // Soros port konfigurálása
}
void loop() {
    int val = analogRead(A3);    // Az A3 bemenet feszültségét mérjük
    Serial.println(val);        // Kiíratjuk az eredményt
}
```

adc_potmeter.ino

- Mérjük meg a potméterrel leosztott feszültséget és írassuk ki!

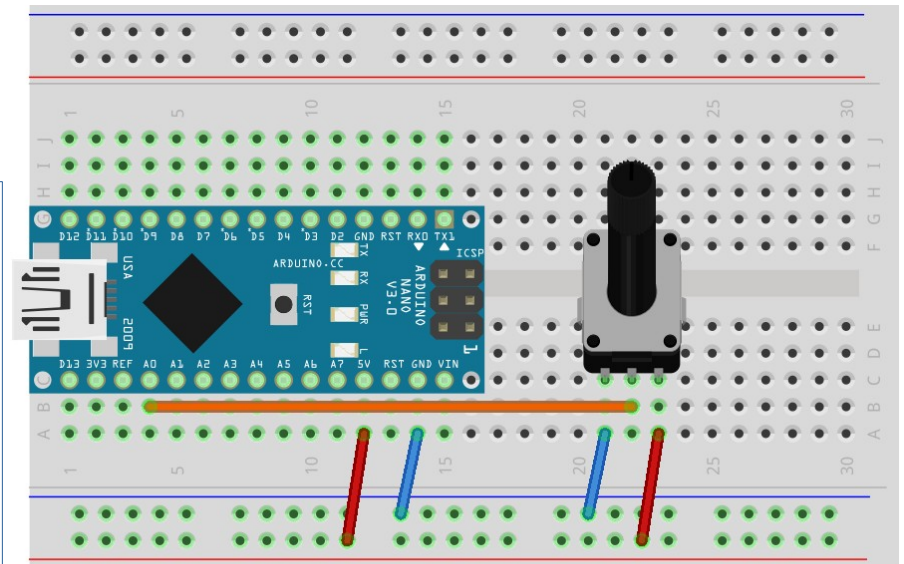
```
void setup() {  
  Serial.begin(9600);  
  analogReference(DEFAULT);  
  Serial.println("ADC potmeter reading");  
}
```

VCC a referencia

A0 a bemenet

```
void loop() {  
  int reading = analogRead (A0);  
  float voltage=(reading*4.75)/1024.0;  
  Serial.print("ADC: ");  
  Serial.print(reading);  
  Serial.print(" Voltage: ");  
  Serial.print(voltage, 2);  
  Serial.println(" V");  
  delay(2000);  
}
```

VCC
tényleges
értéke



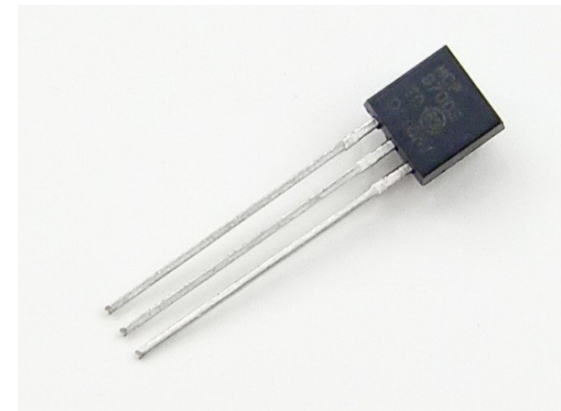
fritzing

```
COM5  
ADC potmeter reading  
ADC: 551 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V  
ADC: 550 Voltage: 2.53 V
```

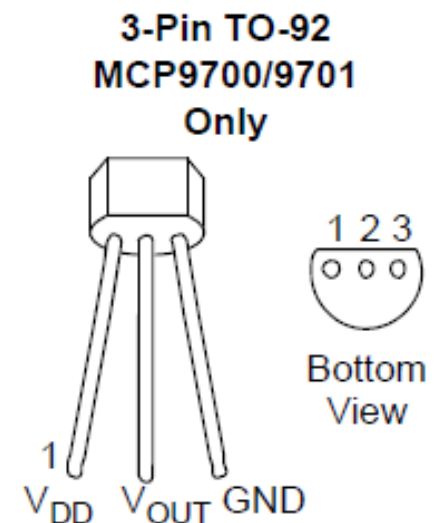
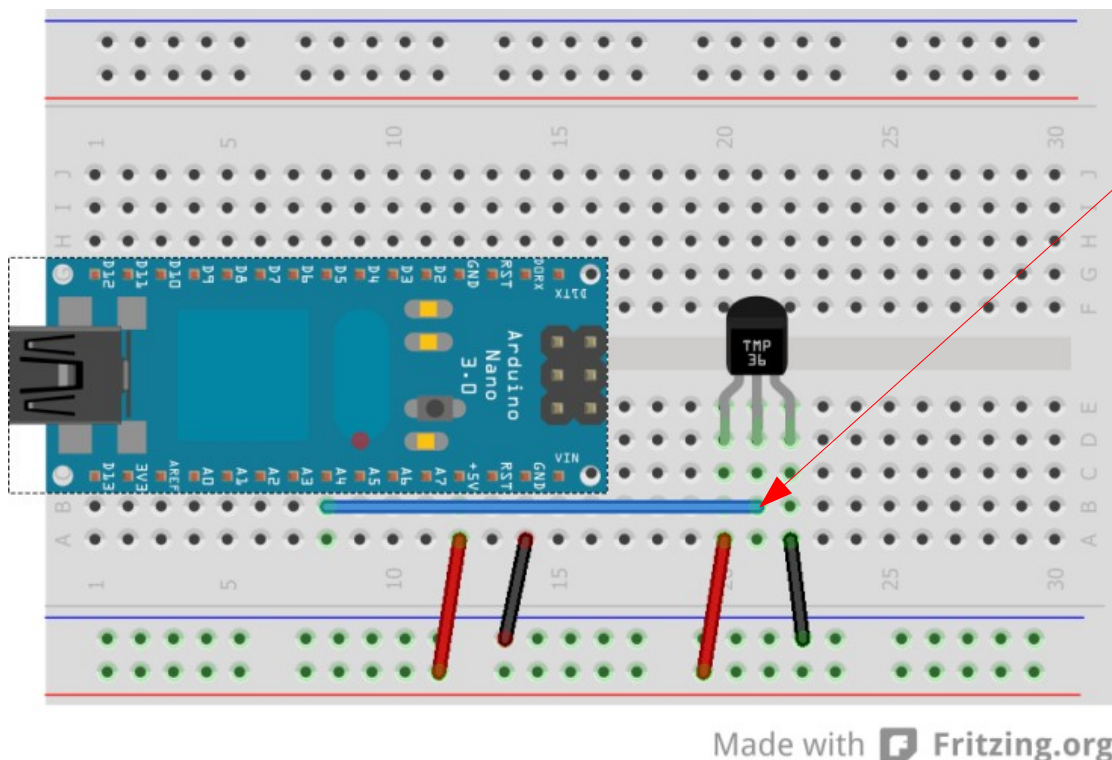
Hőmérséklet mérése analóg hőmérővel

■ Microchip MCP9700

- ❖ $V_{DD} = 2,5 - 5,5 \text{ V}$
- ❖ Mérési tart.: $-40 - 150 \text{ }^{\circ}\text{C}$
- ❖ Érzékenység: $10 \text{ mV / }^{\circ}\text{C}$
- ❖ Nullapont: $500 \text{ mV @ } 0 \text{ }^{\circ}\text{C}$



Most az **A4** bemenetre kötöttük a hőmérőt, de más bemenetet is használhatunk!



AnalogThermometer.ino

```
void setup () {  
    Serial.begin(9600);  
    analogReference(INTERNAL);  
    Serial.println("Analóg hőmérő");  
}  
  
void loop () {  
    long reading = analogRead (A4);  
    //--- Átszámítjuk mV-ba  
    long voltage = reading*1100/1024;  
    Serial.print (voltage);  
    Serial.print (" mV, ");  
    //--- Átszámítjuk Celsius fokokra  
    float tempC = (voltage - 500)/10.0;  
    Serial.print (tempC,1);  
    Serial.print (" °C, ");  
    //--- Celsiusból Fahrenheit fokokba  
    float tempF = (tempC * 9/5) + 32;  
    Serial.print (tempF,1);  
    Serial.println (" °F");  
    delay (5000);  
}
```

Vref = 1,1 V

Hőmérés MCP9700A
hőmérővel

Az előző oldali példában az A4 bemenetre kötöttük a hőmérőt, de köthetjük volna másik bemenetre is!

Analóg hőmérő

763 mV, 26.3 °C, 79.3 °F
771 mV, 27.1 °C, 80.8 °F
770 mV, 27.0 °C, 80.6 °F
770 mV, 27.0 °C, 80.6 °F
771 mV, 27.1 °C, 80.8 °F
770 mV, 27.0 °C, 80.6 °F
770 mV, 27.0 °C, 80.6 °F
771 mV, 27.1 °C, 80.8 °F
770 mV, 27.0 °C, 80.6 °F
768 mV, 26.8 °C, 80.2 °F
779 mV, 27.9 °C, 82.2 °F

AnalogThermometer2.ino

```
void setup () {  
  Serial.begin(9600);  
  analogReference(INTERNAL);  
  Serial.println("Analóg hőmérő átlagolással");  
}
```

```
void loop () {  
  long reading = 0;  
  for (int i = 0; i < 1100; i++) {  
    reading += analogRead(A4);  
  }  
  long voltage = reading/1024;  
  Serial.print (voltage);  
  Serial.print (" mV, ");  
  float tempC = (voltage-500)/10.0;  
  Serial.print (tempC, 1);  
  Serial.print (" °C, ");  
  float tempF = (tempC*9/5)+32;  
  Serial.print (tempF, 1);  
  Serial.println (" °F");  
  delay (5000);  
}
```

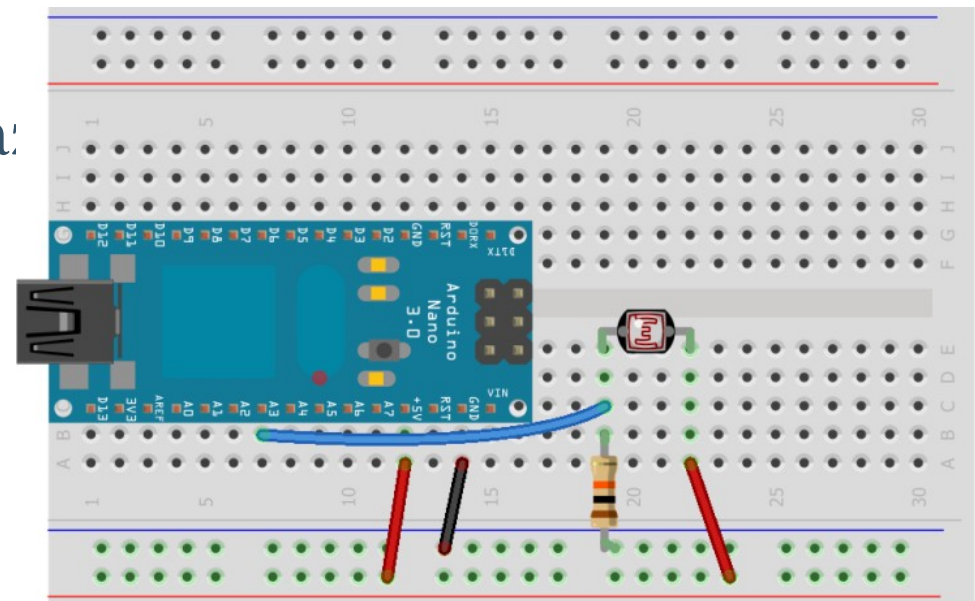
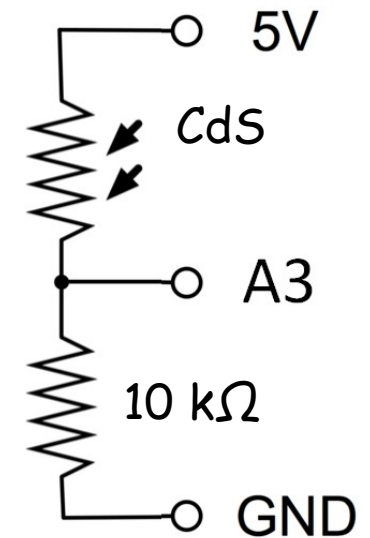
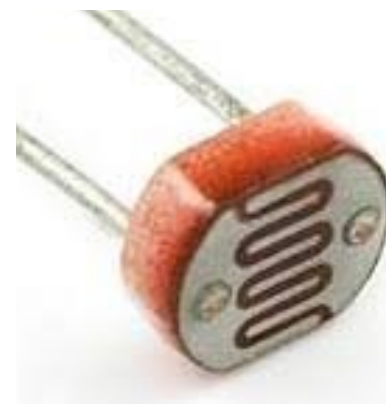
Hőmérés **MCP9700A**
hőmérővel, átlagolással

1100 db mérés eredményét összeadjuk
és nem szorzunk 1100-zal!

Analóg hőmérő átlagolással
772 mV, 27.2 °C, 81.0 °F
771 mV, 27.1 °C, 80.8 °F
772 mV, 27.2 °C, 81.0 °F
772 mV, 27.2 °C, 81.0 °F
771 mV, 27.1 °C, 80.8 °F
771 mV, 27.1 °C, 80.8 °F
771 mV, 27.1 °C, 80.8 °F
771 mV, 27.1 °C, 80.8 °F
771 mV, 27.1 °C, 80.8 °F
771 mV, 27.1 °C, 80.8 °F

Fénymérés fényérzékeny ellenállással

- A kapcsolás feszültségosztóként működik, amelyikben a felső tag egy CdS fényérzékeny ellenállás, melynek ellenállása a megvilágítástól függően széles határok között változik. A megvilágítás hatására az ellenállása csökken...
- Az ellenállásosztó közös pontját a **A3** analóg bemenetre kötöttük
- Az eredményt soros porton kiküldjük a számítógépre, s a terminálablakban jelenik meg.



Made with  Fritzing.org

Photoresistor.ino

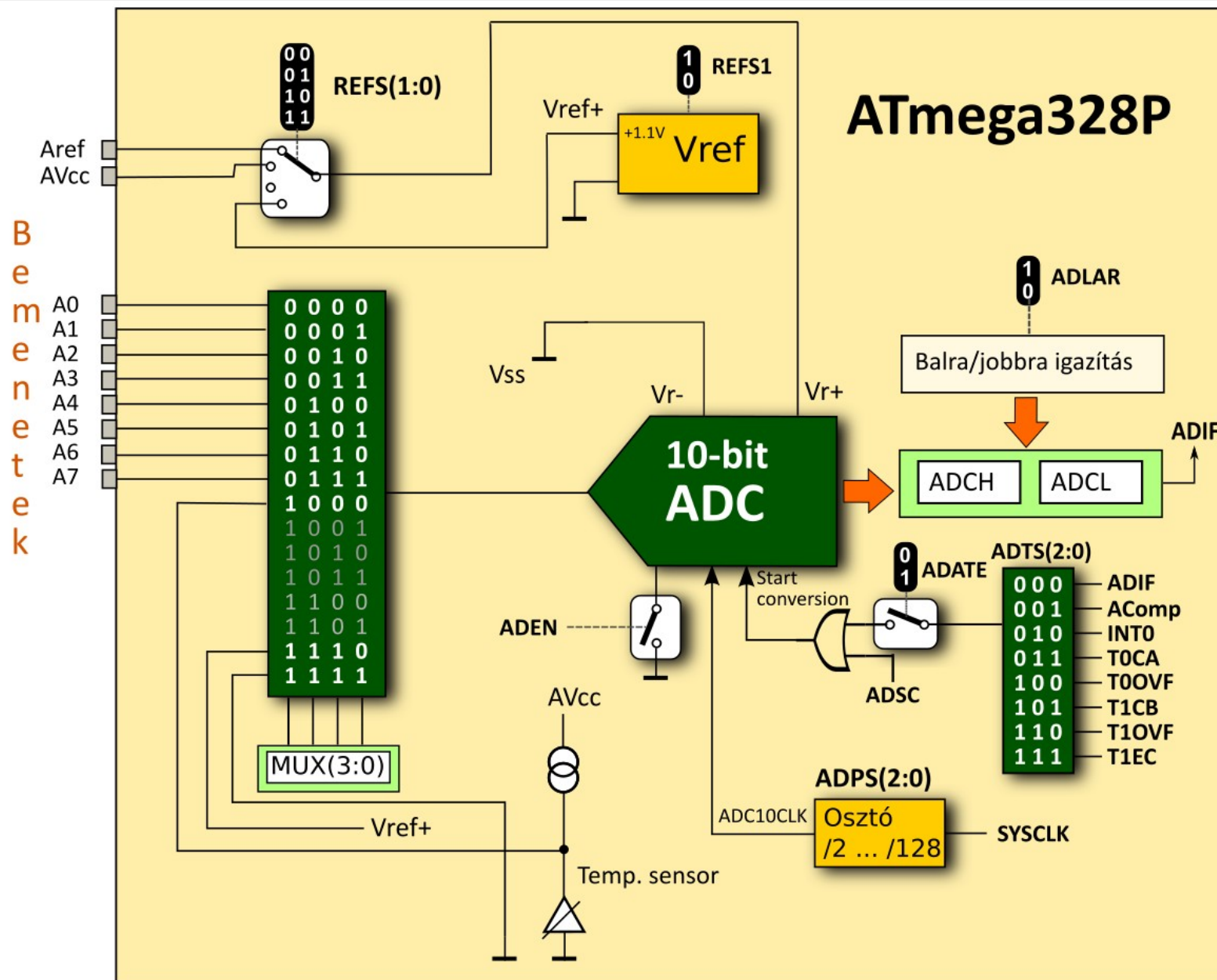
```
void setup () {
  Serial.begin(9600);
  analogReference(DEFAULT); //VCC a referencia
  Serial.print("Photoresistor");
}

void loop () {
  long reading = 0;
  for(int i=0; i<4750; i++) {
    reading += analogRead(A3);
  }
  // Trükkös osztás 1024-gyel
  float voltage = reading>>10;
  Serial.print(voltage,0);
  Serial.print(" mV, ");
  // Átszámítás kOhm-ra
  // Rx = VCC*10k/voltage - 10k
  float rx = 47500/voltage - 10;
  Serial.print(rx,3);
  Serial.println(" kOhm");
  delay (5000);
}
```

4750 db mérés eredményét összeadjuk és nem szorzunk Vref-fel!

Photoresistor
3145 mV, 5.103 kOhm
3149 mV, 5.084 kOhm
3036 mV, 5.646 kOhm
1393 mV, 24.099 kOhm
1322 mV, 25.930 kOhm
4406 mV, 0.781 kOhm
4449 mV, 0.677 kOhm
4134 mV, 1.490 kOhm
2856 mV, 6.632 kOhm
2836 mV, 6.749 kOhm

Az ADC blokkvázlata



Bemenetek

Az ADC regiszterei

- ADMUX – referenciafeszültség és bemeneti csatorna választó

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

REFS – 00: EXTERNAL, 01: DEFAULT (V_{cc}), 11: INTERNAL (1,1V)

ADLAR – Az eredmény igazítása **0**: jobbra, **1**: balra

MUX – bemenetválasztás (**0000-0111**: A0-A7, **1000**: belső hőmérő, **1110**: 1,1V-os referencia, **1111**: GND)

- ADCH és ADCL adatregiszterek

	15	14	13	12	11	10	9	8	
ADLAR = 0	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
ADLAR = 1	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	

Az ADC regiszterei

■ ADCSRA – vezérlő és állapotregiszter I.

7	6	5	4	3	2	1	0	
ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

ADEN: ADC engedélyezése, **ADSC:** konverzió indítás, **ADIF:** konverzió vége jelzőbit, **ADIE:** megszakításkérés engedélyezés, **ADPS[2:0]:** előszámláló választás (/2 ... /128)

■ ADCSRB – vezérlő és állapotregiszter II.

7	6	5	4	3	2	1	0	
-	ACME	-	-	-	ADTS2	ADTS1	ADTS0	ADCSRB
R	R/W	R	R	R	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

ACME – az analóg komparátorhoz rendeli a bemeneti multiplexert (ADEN = 0 esetén)
ADTS[2:0] – konverziót indító (trigger) jelforrás választása:

000: Szabadonfutó mód (ADIF)
001: Analóg komparátor
010: Külső megszakítás (INT0)
011: Timer0 Compare esemény A

100: Timer0 Túlcsordulás
101: Timer1 Compare esemény B
110: Timer1 Túlcsordulás
111: Timer1 Capture esemény

Az ADC regiszterei

DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Ha 1-et írunk valamelyik bitbe, az letiltja a hozzá tartozó **A0 – A5** kivezetés digitális bemeneti bufferét. **A6** és **A7** csak analóg bemenet, ezeknél nincs mit letiltani...

A beépített hőmérő használata

- ATmega328p beépített hőmérő a 0b1000 ADC csatornán érhető el

```
long readTemp() {  
    long result;  
    // Hőmérés , 1V1 belső referenciával  
    ADMUX = _BV(REFS1) | _BV(REFS0) | _BV(MUX3);  
    delay(20); // Vref beállási idő  
    ADCSRA |= _BV(ADSC); // Konverzió indítása  
    while (bit_is_set(ADCSRA,ADSC)); // Konverzió végére vár  
    result = ADCL; // Előbb ezt kell kiolvasni  
    result |= ADCH<<8; // Magasabb helyiértékű bitek  
    result = (result - 125) * 1075; // Hőmérséklet * 10 000 Celsius fokban  
    return result;  
}
```

Mérés belső referenciával, a 0b1000 csatornában

```
void setup() {  
    Serial.begin(9600);  
    Serial.println("Belső hőmérő");  
}  
  
void loop() {  
    long temp = readTemp();  
    Serial.print("Temperature = ");  
    Serial.print(temp / 10000.0f,1);  
    Serial.println(" °C");  
    delay(5000);  
}
```

Eredmények
(egy kis melegítéssel)

beepitett_homero.ino

Belső hőmérő
Temperature = 25.6 °C
Temperature = 25.6 °C
Temperature = 25.8 °C
Temperature = 25.8 °C
Temperature = 25.9 °C
Temperature = 26.1 °C
Temperature = 25.9 °C
Temperature = 25.7 °C
Temperature = 25.8 °C
Temperature = 26.8 °C
Temperature = 26.9 °C
Temperature = 27.5 °C

A tápfeszültség meghatározása

- ATmega328p belső referenciája a 0b1110 ADC csatornán érhető el

```
long readVcc() {  
    long result;  
    // 1.1V referenciafeszültség mérése, AVcc a referencia  
    ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);  
    delay(20); // Vref beállási idő  
    ADCSRA |= _BV(ADSC); // Konverzió indítása  
    while (bit_is_set(ADCSRA,ADSC)); // Konverzió végére vár  
    result = ADCL; // Előbb ezt kell kiolvasni  
    result |= ADCH<<8; // Magasabb helyiértékű bitek  
    result = 1100L * 1024L / result; // AVcc kiszámítása [mV]  
    return result; // 1100 mV Vref * 1024 ADC felbontás  
}  
  
void setup() {  
    Serial.begin(9600);  
    Serial.println("Tápfeszültség meghatározása belső referenciával");  
}  
  
void loop() {  
    long vcc = readVcc();  
    Serial.print("Vcc = ");  
    Serial.print(vcc);  
    Serial.println(" mV");  
    delay(5000);  
}
```

tapfeszultseg_meghatarozasa.ino

Tápfeszültség meghatározása belső referenciával

Vcc = 4752 mV

Vcc = 4752 mV

Vcc = 4752 mV

Vcc = 4752 mV

Vcc = 4752 mV

Vcc = 4752 mV

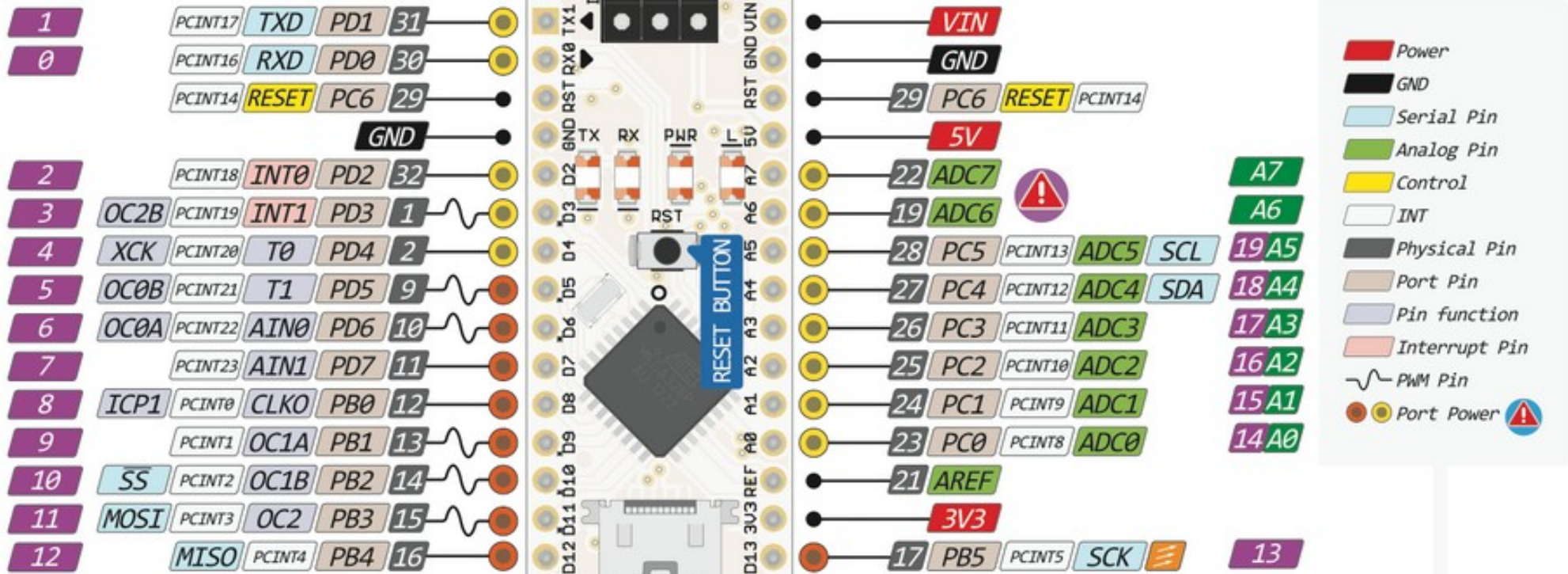
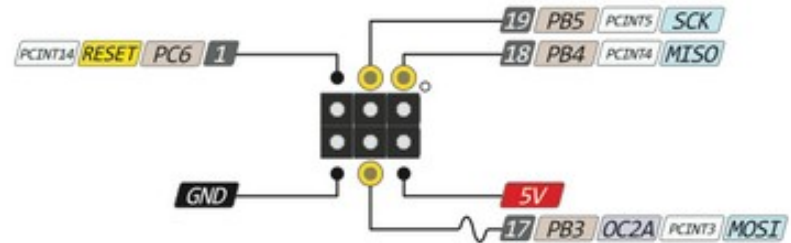
Futási
eredmény

Az Arduino nano kártya kivezetései



NANO PINOUT

The power sum for each pin's group should not exceed 100mA



Absolute MAX per pin 40mA recommended 20mA

Absolute MAX 200mA for entire package

Analog exclusively Pins

Ellenállás színkódok

