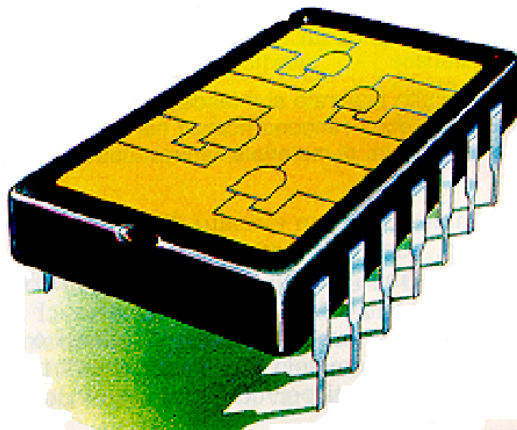


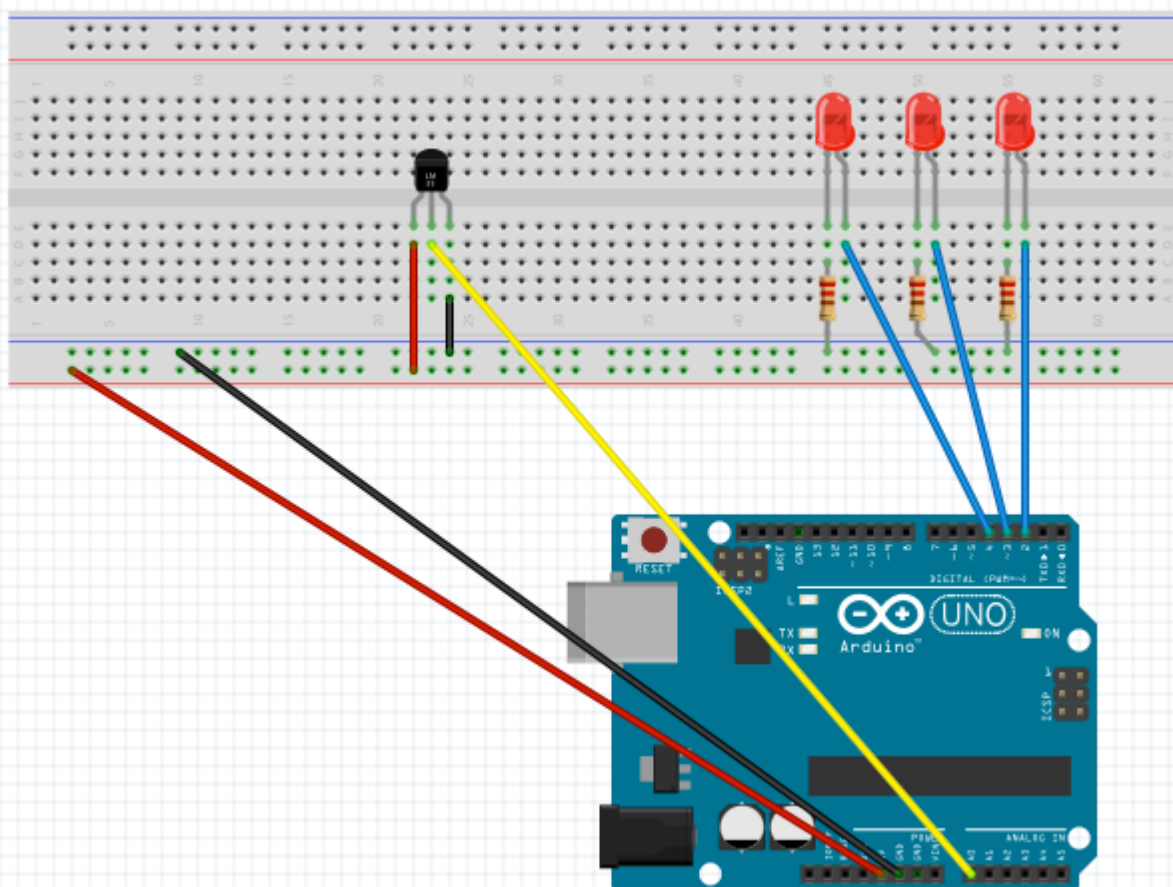
Lez. N. 10 Amorometro



Materiali

Arduino
3 Led
3 resistenze da 220 Ω
1 TMP 36
Jumper vari

Tanta pazienza!



Introduzione



In questo circuito c'è l'introduzione di un sensore che misura la temperatura esterna. I led sono degli attuatori, eseguono quello che gli viene comandato dal programma inserito nel microprocessore, mentre il TMP 36 è un sensore sensibile alla temperatura esterna. Oltre ai pin digitali ci sono anche i pin analogici destinati a cogliere i segnali che provengono dal mondo esterno. Sono A0, A1, A2, A3, A4, A5.

Le novità sono tante e tutte degne di spiegazione approfondita.

La tensione sul piedino varia da 0 a 5V, ma potrebbe essere anche decimale (es. 2,5V) perciò bisogna utilizzare il prefisso **float** (sta per floating point, o virgola mobile). Fino a questo

progetto abbiamo usato sempre il prefisso **int** che sta per intero. C'è anche il termine **const** sta per costante. È un qualificatore di variabile che modifica il comportamento della variabile, rendendola una "variabile di sola lettura". Questo significa che la variabile può essere usata come ogni altro tipo di variabile ma che il suo valore non può essere cambiato.

Codice

```
// AMOROMETRO
```

```
const int sensorPin = A0;  
const float baselineTemp = 20.0;
```

```
void setup()  
{
```

```
  Serial.begin(9600);
```

```
  for(int pinNumber = 2; pinNumber < 5; pinNumber++)  
  {  
    pinMode(pinNumber, OUTPUT);  
    digitalWrite(pinNumber, LOW);  
  }
```

```
}
```

```
void loop()  
{  
  int sensorVal = analogRead(sensorPin);  
  Serial.print("Sensor Value: ");  
  Serial.print(sensorVal);
```

```
  float voltage = (sensorVal/1024.0)*5.0;
```

```
  Serial.print(" , Volts: ");  
  Serial.print(" , degrees C: ");
```

```

float temperature =(voltage - .5) * 100;
Serial.println(temperature);

if(temperature < baselineTemp + 2)
{

digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
}

else if(temperature>= baselineTemp+2 && temperature < baselineTemp + 4)
{

digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);

}

else if(temperature >= baselineTemp + 6)
{

digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);

}
delay(10);
}

```

Codice spiegazione

<pre>const int sensorPin = A0; const float baselineTemp = 20.0;</pre>	<i>const</i> definisce la variabile che può essere intera o decimale, ma deve essere sempre la stessa. <i>baselineTemp</i> è la temperatura di inizio, assunta a 20°C. 20.0 perché è in notazione <i>float</i> (decimale) con la convenzione anglosassone che adopera il punto al posto della virgola.
<pre>void setup() { Serial.begin(9600);</pre>	<i>void setup()</i> è una funzione che serve per inizializzare le variabili, per impostare lo stato dei pin, per far partire le librerie da usare e per impostare le comunicazioni seriali <i>Serial.begin(9600)</i> baud ovvero bit al secondo

<pre>for(int pinNumber =2; pinNumber<5; pinNumber++) { pinMode(pinNumber, OUTPUT); digitalWrite(pinNumber, LOW); }</pre>	Istruzione for/ciclo for Il ciclo for ripete un'istruzione o un blocco di istruzioni per un numero prefissato di volte e utilizza un contatore per incrementare un blocco di istruzioni e terminare il ciclo. <u>Sintassi</u> for(inizializzazione; condizione; incremento) {istruzioni} <u>inizio</u> <i>pinNumber = 2</i> <u>condizione</u> <i>pinNumber<5</i> <u>incremento</u> <i>pinNumber++</i> (incremento di uno) In questo modo sono settati (pinMode) i pin 2, 3, 4 come OUTPUT. Dovranno partire, sempre i pin 2,3, 4, come digitalWrite LOW (basso o 0)
<pre>void loop() { int sensorVal = analogRead(sensorPin); Serial.print("Sensor Value: "); Serial.print(sensorVal); float voltage = (sensorVal/1024.0)*5.0; Serial.print(" , Volts: "); Serial.print(" , degrees C: "); float temperature =(voltage - .5) * 100; Serial.println(temperature);</pre>	void loop() dopo la creazione della funzione setup() che inizializza e imposta i valori iniziali, la funzione loop() esegue ciclicamente il programma definito al suo interno. Nella variabile <i>sensorVal</i> è inserito il valore letto da <i>analogRead</i> di <i>sensorPin</i>, prima variabile definita e la legge nel pin A0. Stampa sul monitor seriale la frase "Sensor Value: " Stampa sul monitor seriale il valore assunto dalla variabile <i>sensorVal</i>. Quindi sul monitor seriale compare: <i>Sensor Value: il valore della variabile sensorVal</i> Ora i calcoli per avere la temperatura corretta. Devo creare una variabile, voltage, e immagazzinarla in un float. <i>sensorVal</i> è quello che legge il sensore, si divide per 1024 e lo si moltiplica per 5. Si stampa " , Volts: " Si stampa " , degrees C: "

	Si crea una ulteriore variabile temperature dove risiederà il valore della temperatura corrente. La formula è: $temperature = (voltage - 5) * 100$ stampa il valore a fine linea il datasheet di TMP 36 dice che ad ogni variazione di 10 mv si ha una differenza esatta di 1 °C.
<pre> if(temperature < baselineTemp + 2) { digitalWrite(2, LOW); digitalWrite(3, LOW); digitalWrite(4, LOW); } else if(temperature >= baselineTemp+2 && temperature < baselineTemp + 4) { digitalWrite(2, HIGH); digitalWrite(3, HIGH); digitalWrite(4, LOW); } else if(temperature >= baselineTemp + 6) { digitalWrite(2, HIGH); digitalWrite(3, HIGH); digitalWrite(4, HIGH); } </pre>	Istruzioni condizionali L'istruzione if permette di eseguire blocchi di codice diversi in base al risultato di una condizione booleana. SE ALLORA ALTRIMENTI if else condizione booleana x==y //x uguale a y x!= y // x diverso da y x<y // x minore di y x>y // x maggiore di y x<=y //x minore o uguale a y x>=y //x maggiore o uguale a y <i>baselineTemp=20 °C</i> <i>temperature=21 °C (supponiamo)</i> SE <i>temperature</i> è minore di <i>baselineTemp+2</i> ALLORA esegui le istruzioni indicate ALTRIMENTI SE <i>temperature</i> è compresa tra 22 e 24 Esegui le istruzioni indicate ALTRIMENTI SE <i>temperature</i> è maggiore o uguale a 26 Esegui le istruzioni indicate.
OPERATORI BOOLEANI	&& operatore AND operatore OR ! operatore NOT
OPERATORI COMPOSTI	++ (incremento) – (decremento) x++; // il valore di x viene reso prima

	disponibile e poi viene incrementato ++x; // il valore della variabile x prima viene incrementato di 1 e poi reso disponibile stessa procedura per il decremento
--	---



Low Voltage Temperature Sensors

Data Sheet

TMP35/TMP36/TMP37

FEATURES

- Low voltage operation (2.7 V to 5.5 V)
- Calibrated directly in °C
- 10 mV/°C scale factor (20 mV/°C on TMP37)
- ±2°C accuracy over temperature (typ)
- ±0.5°C linearity (typ)
- Stable with large capacitive loads
- Specified -40°C to +125°C, operation to +150°C
- Less than 50 µA quiescent current
- Shutdown current 0.5 µA max
- Low self-heating
- Qualified for automotive applications

APPLICATIONS

- Environmental control systems
- Thermal protection
- Industrial process control
- Fire alarms
- Power system monitors
- CPU thermal management

GENERAL DESCRIPTION

The TMP35/TMP36/TMP37 are low voltage, precision centigrade temperature sensors. They provide a voltage output that is linearly proportional to the Celsius (centigrade) temperature. The TMP35/TMP36/TMP37 do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the -40°C to +125°C temperature range.

The low output impedance of the TMP35/TMP36/TMP37 and its linear output and precise calibration simplify interfacing to temperature control circuitry and ADCs. All three devices are intended for single-supply operation from 2.7 V to 5.5 V maximum. The supply current runs well below 50 µA, providing very low self-heating—less than 0.1°C in still air. In addition, a shutdown function is provided to cut the supply current to less than 0.5 µA.

The TMP35 is functionally compatible with the LM35/LM45 and provides a 250 mV output at 25°C. The TMP35 reads temperatures from 10°C to 125°C. The TMP36 is specified from -40°C to +125°C, provides a 750 mV output at 25°C, and operates to 125°C from a single 2.7 V supply. The TMP37 is functionally compatible with the LM50. Both the TMP35 and TMP36 have an output scale factor of 10 mV/°C.

FUNCTIONAL BLOCK DIAGRAM

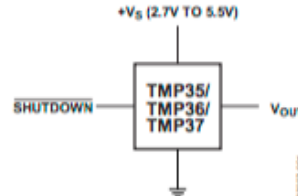


Figure 1.

PIN CONFIGURATIONS

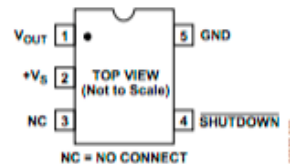


Figure 2. R-5 (SOT-23)

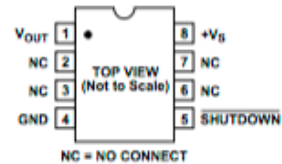


Figure 3. R-8 (SOIC_N)



Figure 4. T-3 (TO-92)

The TMP37 is intended for applications over the range of 5°C to 100°C and provides an output scale factor of 20 mV/°C. The TMP37 provides a 500 mV output at 25°C. Operation extends to 150°C with reduced accuracy for all devices when operating from a 5 V supply.

The TMP35/TMP36/TMP37 are available in low cost 3-lead TO-92, 8-lead SOIC_N, and 5-lead SOT-23 surface-mount packages.

Rev. H
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

Document Feedback