



## DOSSIER RESSOURCE 1 : DR1

### ✓ Documentation technique du capteur TMP36 :



## 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.

### FUNCTIONAL BLOCK DIAGRAM

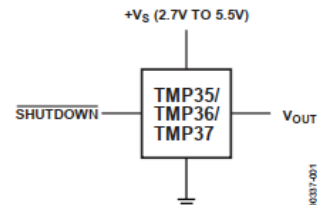


Figure 1.

### PIN CONFIGURATIONS

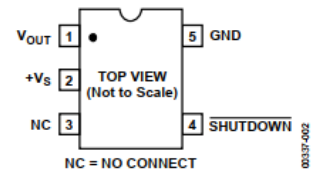


Figure 2. RJ-5 (SOT-23)

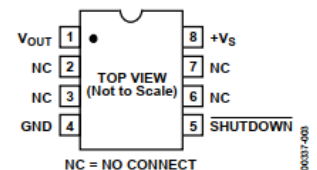


Figure 3. R-8 (SOIC\_N)

### ✓ Documentation technique de la thermistance CTN100 :

#### Caractéristiques techniques

Type de coefficient de température : CTN  
Résistance à 25°C : 100Ω  
Tolérance : ±10%  
Puissance maximale : 450mW  
Longueur : 5.5mm  
Profondeur : 5mm  
Hauteur : 6.5  
Température d'utilisation minimum : -55°C  
Température d'utilisation maximum : +125°C



## DOSSIER RESSOURCES

### DOSSIER RESSOURCE 2

#### ✓ Présentation du logiciel « Tinkercad » :

Cette ressource a pour objectif la mise en œuvre du simulateur de la carte Arduino : "**Tinkercad**"



#### ○ Etape 1 :

La première étape consiste à se connecter à votre classe virtuelle :

- Sélectionner le **lien** ci-dessous :

<https://www.tinkercad.com/joinclass/.....>

- Puis entrer votre **pseudo** :

Bienvenue dans

Classe virtuelle

Quel est ton surnom?

Entrer le pseudo

Et voilà!

#### Pseudo :

Votre « nom+1<sup>ère</sup> lettre de votre prenom », sans majuscule, sans accent et sans espace.

- Puis : **Et voilà.**



### DOSSIER RESSOURCE 3

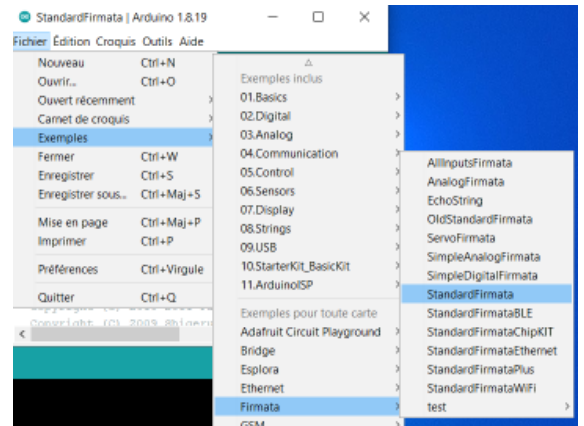
#### ✓ Programmation de la carte Arduino en langage python :



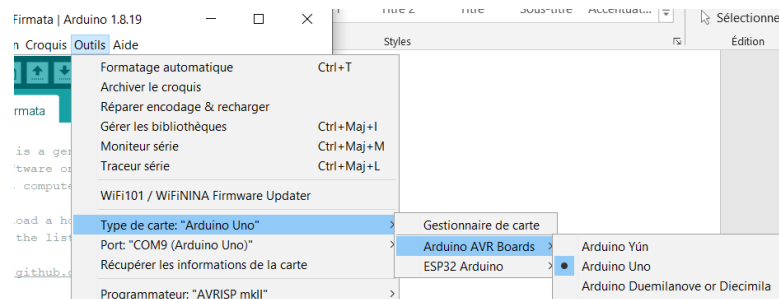
##### ○ Etape 1 :

La première étape consiste à installer un « firmware » sur la carte pour qu'elle puisse communiquer en Python.

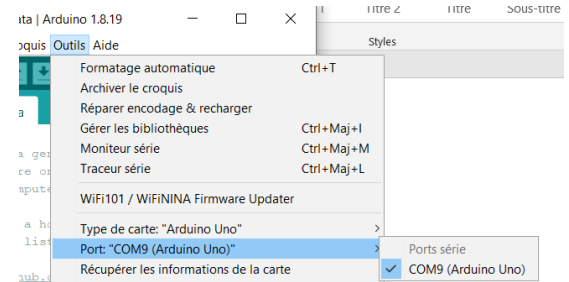
- Ouvrir le logiciel **Arduino IDE** ;
- Sélectionner : **Fichier > Exemples > Firmata > StandardFirmata** :



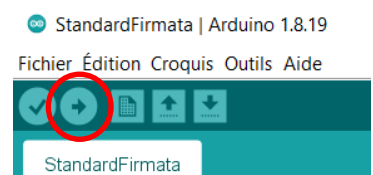
- Brancher la carte Arduino Uno en USB.  
Choisir le type de carte : **Outils > Type de carte > Arduino Uno** :



- Sélectionner la carte : **Outils > Port > COM .....** :  
(La carte connectée apparaît dans la liste) :



- Téléverser le microprogramme (firmware) sur la carte :  **Icône ->**



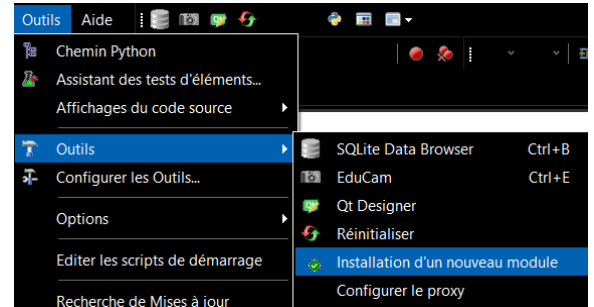
- La carte est prête pour être utilisée avec un IDE Python comme EduPython. Ce firmware reste ensuite sur la carte le temps de l'activité, même si elle est déconnectée ou éteinte.



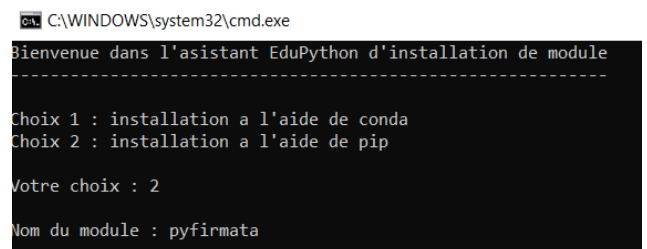
### ○ Etape 2 :

La seconde étape consiste à installer la bibliothèque « Pyfirmata » qui renferme les commandes Python compréhensibles par la carte Arduino dans Edupython.

- Démarrer : **EduPython** ;
- Faire : **Outils > installation d'un nouveau module** :



- Un menu apparaît : **Choisir 2** et ensuite taper **pyfirmata** :



- Faire : **Entrer** puis suivre les instructions.

### ○ Etape 3 :

Voici un code qui permet de tester (sans aucun matériel) si tout est opérationnel.

Il consiste à faire clignoter 10x la LED qui se trouve d'origine sur la carte et qui est reliée à la sortie digitale 13 :

