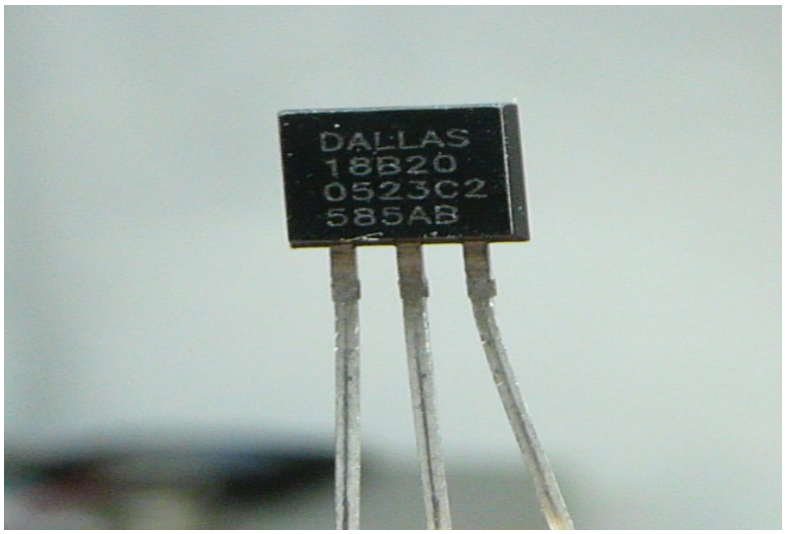


Le DS18B20 est un capteur de température du fabricant Dallas, il communique via un bus 1-Wire (du même fabricant) et possède une précision de 12 bits sur une plage de -55°C à +125°C.

Si vous voulez les détails voici le datasheet de la bête :
http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf

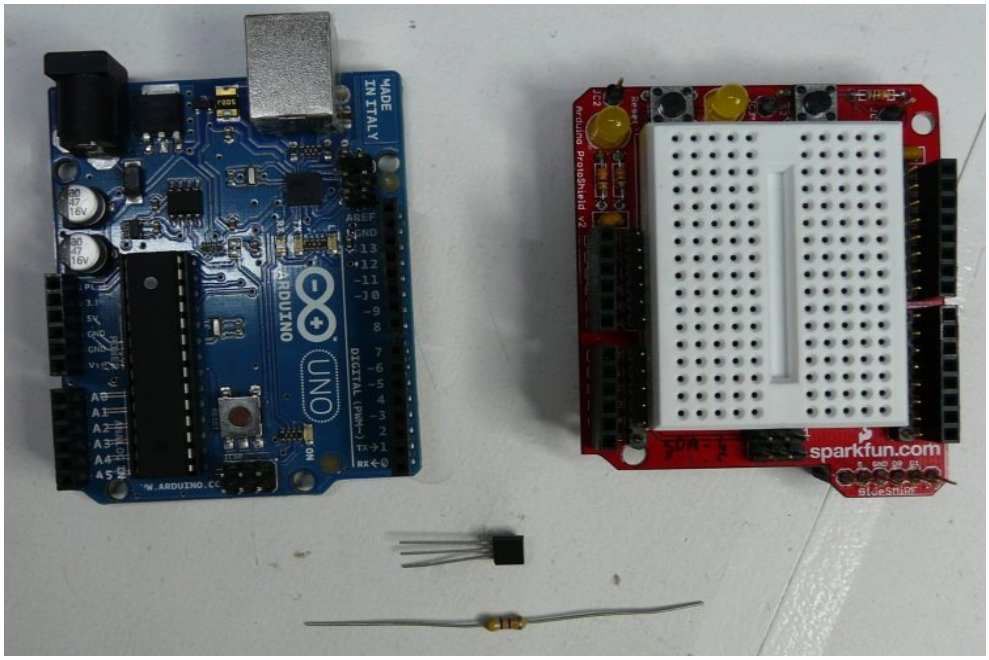
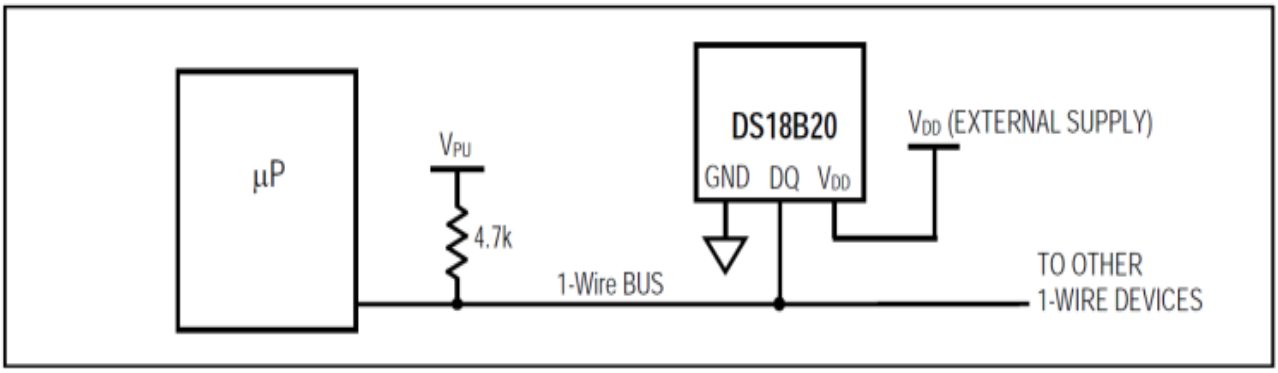
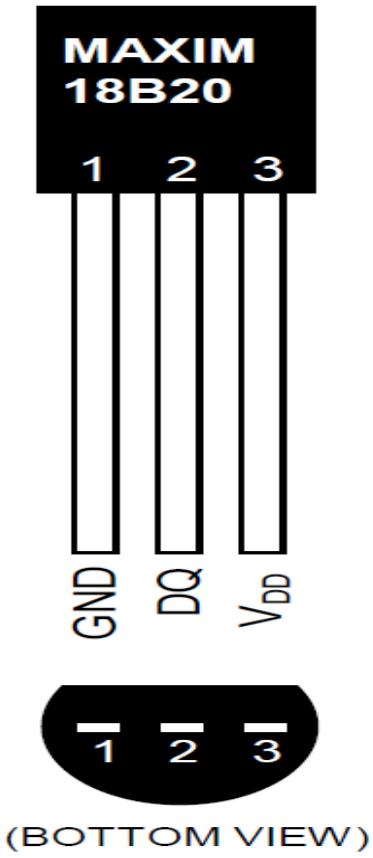
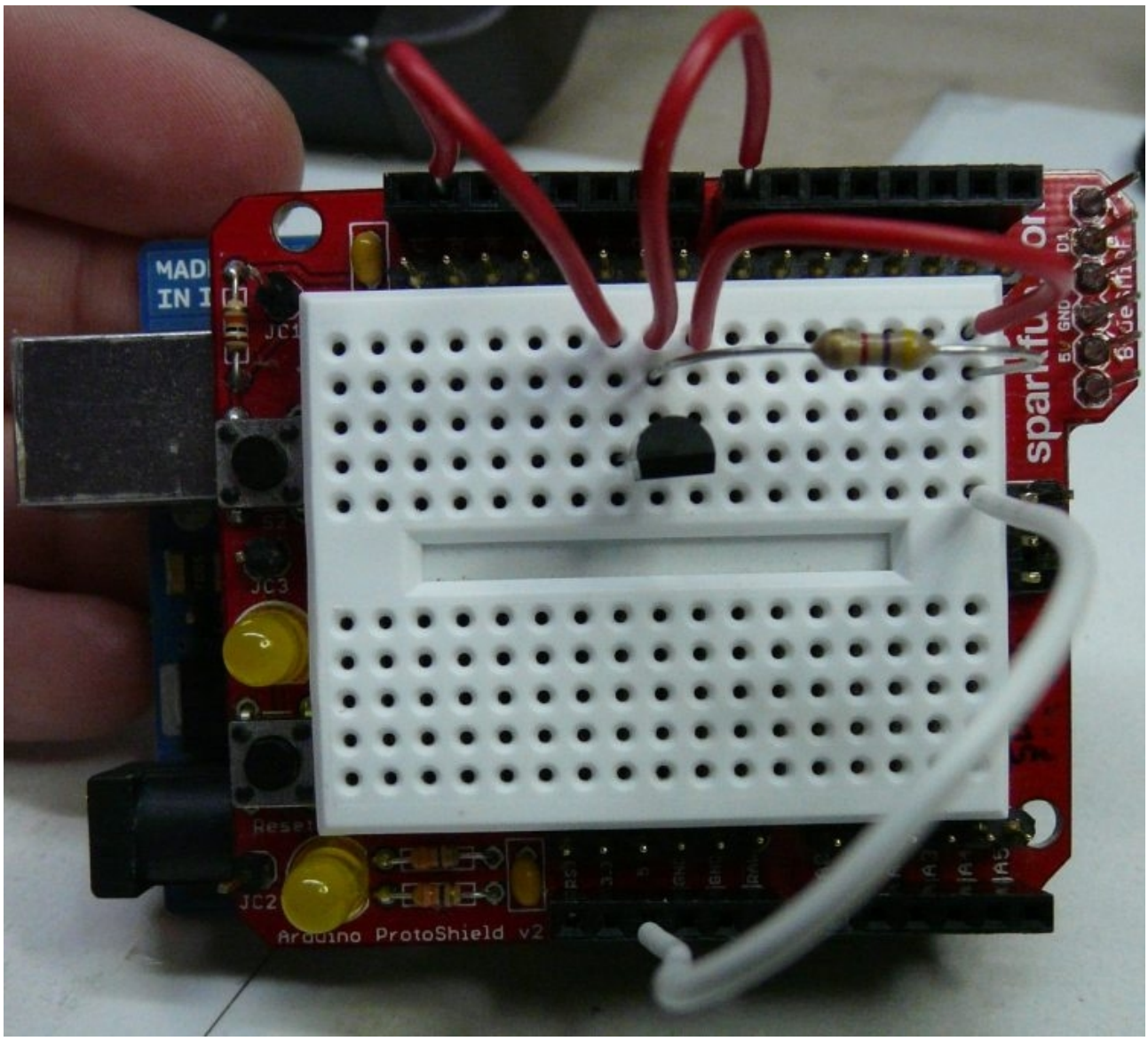


Figure 5. Powering the DS18B20 with an External Supply







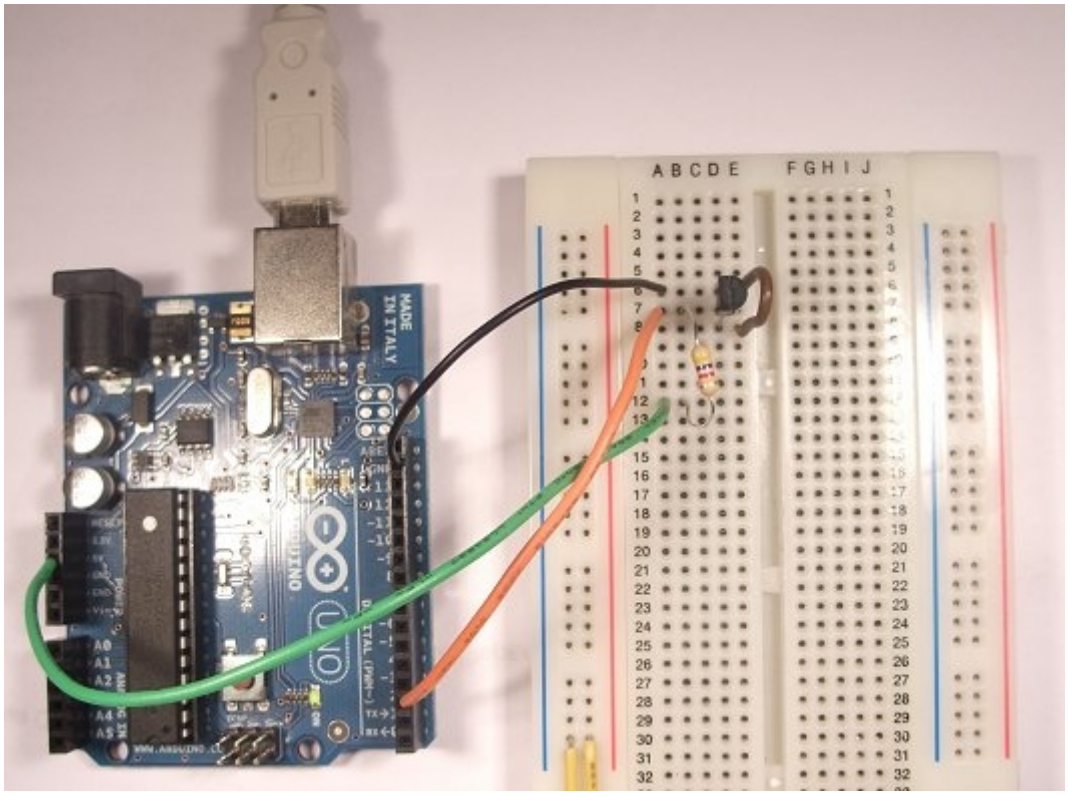

```
Arduino -> DS18B20
D7 -> Data
VCC -> VCC
GND -> GND

Data -> résistance de 4K7 -> VCC
```

—

Now, before we get to the programming part, lets wire up our temperature sensor. The DS18B20 can be powered by between 3.0V and 5.5V so you can simply connect its GND pin to 0V and the VDD pin to +5V from the Arduino. However, the DS18B20 can also extract its power from the data line which means we only effectively need two wires to connect it up. This makes it great for use as an external sensor.

So we will use the 2 wire method. Simply connect both the GND and VDD pins to 0V (yes both). Then connect the DQ pin to pin 2 on the Arduino board (can be any Arduino digital pin)

A 4K7 ohm pullup resistor is required on the DQ pin to pull it up to 5V



Le fonctionnement interne du DS18B20 :

Comme tout module Dallas 1-Wire, le DS18B20 contient un « scratchpad » qui est une sorte de mémoire tampon sécurisé ou l'on peut venir lire et/ou écrire.

Voici comment se structure le scratchpad du DS18B20 :

Figure 7. DS18B20 Memory Map

SCRATCHPAD
(POWER-UP STATE)

Byte 0	Temperature LSB (50h)	(85°C)
Byte 1	Temperature MSB (05h)	
Byte 2	T_H Register or User Byte 1*	
Byte 3	T_L Register or User Byte 2*	
Byte 4	Configuration Register*	
Byte 5	Reserved (FFh)	
Byte 6	Reserved	
Byte 7	Reserved (10h)	
Byte 8	CRC*	

**Power-up state depends on value(s) stored in EEPROM.*

On peut voir que les deux premiers octets contiennent la température mesurée, c'est sur ces deux octets que nous allons venir lire avec notre programme.

Le détail de ces deux registres :

Figure 2. Temperature Register Format

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

**The power-on reset value of the temperature register is +85°C.*

Pas facile à comprendre hein ;)
La température étant une valeur à virgule celle ci est stockée sous la forme d'un flottant avec un exposant, une mantisse et un signe.

Pour avoir la température en degré Celsius il faut appliquer la formule suivante :

```
temp = ((MSB << 8) + LSB) * 0.0625;
```

Mais cela ne fera pas tout !

Pour pouvoir lire la température il faut tout d'abord lancer une mesure de température (logique). Pour ce faire il faut envoyer la commande 0x44, puis faire une demande de lecture du scratchpad (0xBE).

Sauf que …

Table 2. Thermometer Resolution Configuration

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	(t_{CONV}/8)
0	1	10	187.5ms	(t_{CONV}/4)
1	0	11	375ms	(t_{CONV}/2)
1	1	12	750ms	(t_{CONV})

La conversion analogique -> numérique n’est pas instantanée !
Elle prend même beaucoup de temps ! Presque une seconde !

—