

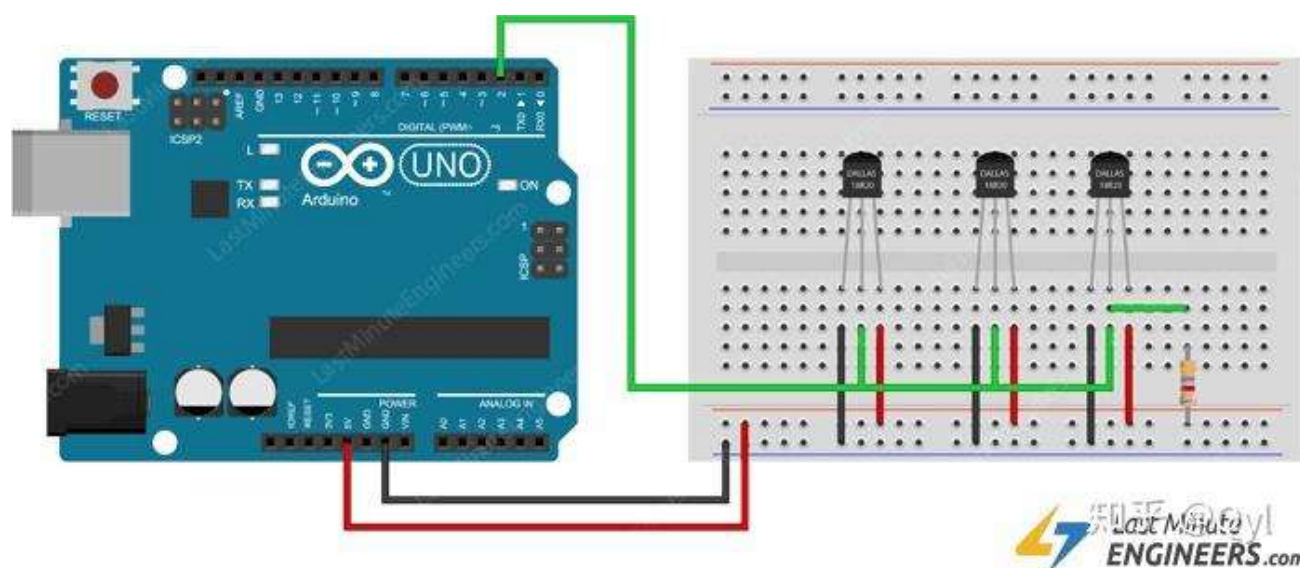
將多個 DS18B20 數位溫度感測器與 Arduino 接口

將多個 DS18B20 感測器連接到 Arduino

連接非常簡單。

首先將所有 DS18B20 並聯連接，即將所有 VCC 接腳，GND 接腳和訊號接腳共用。接著將 VCC 連接至 Arduino 上的 5V 輸出，將 GND 連接至 Arduino 接地，並將訊號接腳連接至 arduino 上的數位接腳 2。

接下來，您需要在訊號和電源引腳之間的整個總線上增加一個 **4.7k 上拉電阻**，以保持資料傳輸的穩定性。（arduino 上的內部引體管不起作用）



為 DS18B20 安裝庫

1. 下載 DallasTemperature 程式庫

<https://github.com/milesburton/Arduino-Temperature-Control-Library>

2. 下載 OneWire 程式庫

<https://github.com/PaulStoffregen/OneWire>

方法 1：透過索引讀取 DS18B20

在這種方法中，初始化達拉斯溫度庫時，它將檢測共享相同總線的所有感測器。它將整個總線視為感測器陣列，並為其分配一個索引。這樣我們就可以透過其索引和讀取溫度來精確定位每個感測器。

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into digital pin 2 on the Arduino
#define ONE_WIRE_BUS 33

// Setup a oneWire instance to communicate with any OneWire device
OneWire oneWire(ONE_WIRE_BUS);

// Pass oneWire reference to DallasTemperature library
DallasTemperature sensors(&oneWire);

int deviceCount = 0;
float tempC;

void setup(void)
{
  sensors.begin(); // Start up the library
  Serial.begin(115200);

  // locate devices on the bus
  Serial.print("Locating devices...");
  Serial.print("Found ");
  deviceCount = sensors.getDeviceCount();
  Serial.print(deviceCount, DEC);
  Serial.println(" devices.");
  Serial.println("");
}

void loop(void)
{
  // Send command to all the sensors for temperature conversion
  sensors.requestTemperatures();

  // Display temperature from each sensor
  for (int i = 0; i < deviceCount; i++)
  {
    Serial.print("Sensor ");
    Serial.print(i+1);
    Serial.print(" : ");
    tempC = sensors.getTempCByIndex(i);
    Serial.print(tempC);
    Serial.print((char)176); //shows degrees character
    Serial.print("C  |  ");
    Serial.print(DallasTemperature::toFahrenheit(tempC));
    Serial.print((char)176); //shows degrees character
    Serial.println("F");
  }
}
```

```

    }
    Serial.println("");
    delay(1000);
}

```

代碼說明：

草圖首先包括庫，聲明感測器總線連接到的引腳以及創建 DallasTemperature 庫的物件。

在程式碼的設定部分，我們首先呼叫 `begin()` 函數。它初始化總線並檢測總線上所有的 DS18B20。然後為每個感測器分配一個索引，並將位元解析度設為 12 位元。

接下來，我們呼叫 `getDeviceCount()` 函數以取得在總線上找到的裝置數量。

在程式碼的循環部分中，我們使用 `requestTemperatures()` 函數將命令發送到所有感測器以進行溫度轉換。

現在，使用一個簡單的 `for(int i = 0; i < deviceCount; i++)` 循環，我們可以遍歷感測器陣列，並 `i` 透過簡單地呼叫以下命令讀取 DS18B20 的溫度：
`getTempCByIndex(i)`

方法 2：透過位址讀取 DS18B20

我們知道，每個 DS18B20 都分配有一個唯一的 64 位元位址，以使它們彼此區分。在這種方法中，我們將找到該位址來相應地標記每個感測器。然後可以使用該位址分別讀取每個感測器。

在總線上尋找 DS18B20 的位址

下圖偵測了總線上所有的 DS18B20，並將其單線位址列印在串列監視器上。

您一次只能佈線一個感測器以找到其位址(或依序添加一個新感測器)，以便能夠透過其位址識別每個感測器。然後，您可以標記每個感測器。

```

#include <OneWire.h>
#include <DallasTemperature.h>

```

```

// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 33

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

// variable to hold device addresses
DeviceAddress Thermometer;

int deviceCount = 0;

void setup(void)
{
  // start serial port
  Serial.begin(115200);

  // Start up the library
  sensors.begin();

  // locate devices on the bus
  Serial.println("Locating devices...");
  Serial.print("Found ");
  deviceCount = sensors.getDeviceCount();
  Serial.print(deviceCount, DEC);
  Serial.println(" devices.");
  Serial.println("");

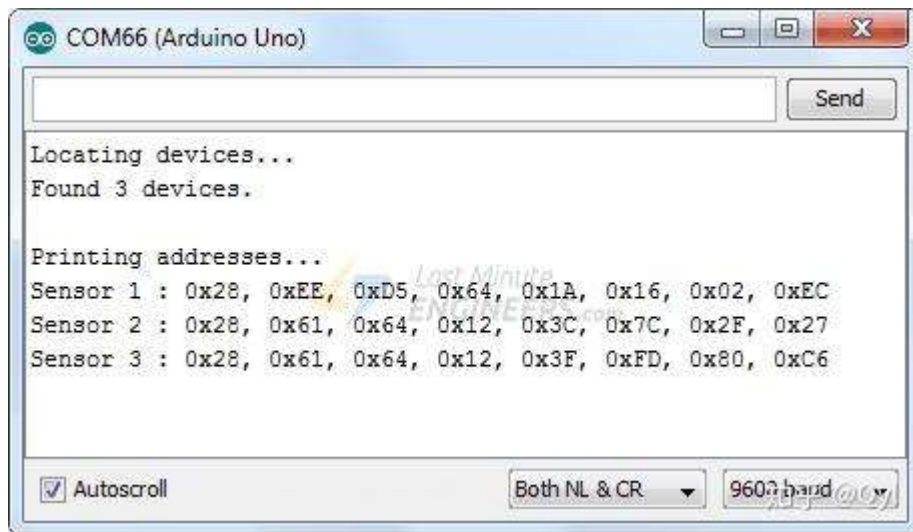
  Serial.println("Printing addresses...");
  for (int i = 0; i < deviceCount; i++)
  {
    Serial.print("Sensor ");
    Serial.print(i+1);
    Serial.print(" : ");
    sensors.getAddress(Thermometer, i);
    printAddress(Thermometer);
  }
}

void loop(void) {
  Serial.print("");
}

void printAddress(DeviceAddress deviceAddress)
{
  for (uint8_t i = 0; i < 8; i++)
  {
    Serial.print("0x");
    if (deviceAddress[i] < 0x10) Serial.print("0");
    Serial.print(deviceAddress[i], HEX);
    if (i < 7) Serial.print(" ");
  }
  Serial.println("");
}

```

現在，打開串行監視器。您應該獲得以下內容



在下一個草圖中根據需要複製所有位址。

按位址讀取 DS18B20

下圖透過其位址讀取 DS18B20 的溫度。在上傳草圖之前，您需要使用在先前的草圖中找到的位址來變更 DS18B20 的位址。

```
#include <OneWire.h> // 引用 OneWire 程式庫
#include <DallasTemperature.h> // 引用 DallasTemperature 程式庫
uint8_t sensor1[8] = { 0x28, 0x4A, 0x07, 0x49, 0xF6, 0x4F, 0x3C, 0x44 };
uint8_t sensor2[8] = { 0x28, 0x8B, 0xE7, 0x49, 0xF6, 0xF2, 0x3C, 0xBD };
uint8_t sensor3[8] = { 0x28, 0x20, 0x6C, 0x49, 0xF6, 0x89, 0x3C, 0x37 };
#define ONE_WIRE_BUS 33 // 定義 DS18B20 資料腳位(數位腳 D6)((所有)溫度感測器都接在同一個腳位)
OneWire oneWire(ONE_WIRE_BUS); // 定義 oneWire 物件類型
DallasTemperature sensors(&oneWire); // 定義溫度感測器物件

float Tb1;
float Tb2;
float Tb3;

void setup() {
  Serial.begin(115200);
  // put your setup code here, to run once:
  sensors.begin();
}

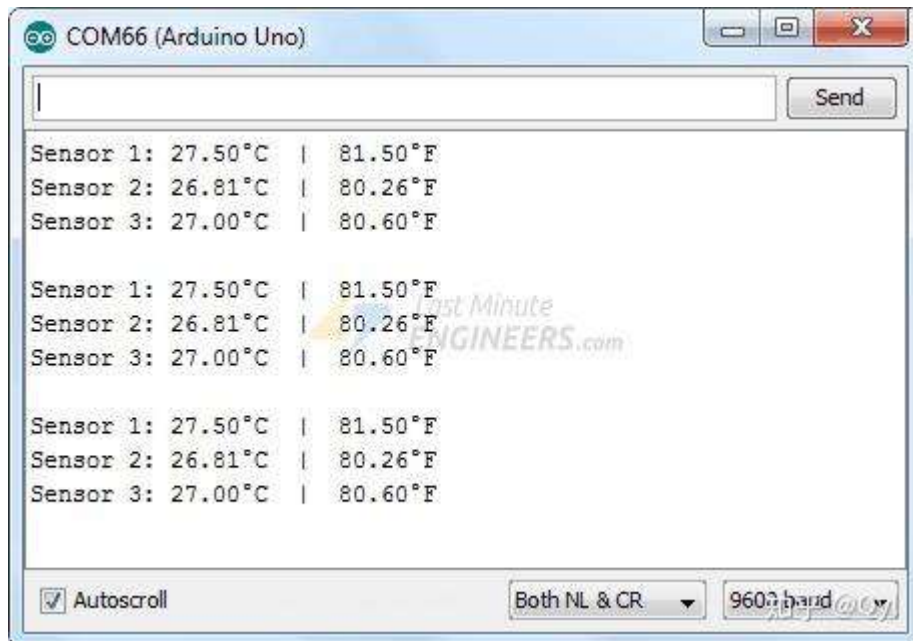
void loop() {
  // put your main code here, to run repeatedly:
```

```

    sensors.requestTemperatures();    // 下指令要求取得(所有)溫度感測器
    的溫度值
    float Tb1 = sensors.getTempC(sensor1);
    float Tb2 = sensors.getTempC(sensor2);
    float Tb3 = sensors.getTempC(sensor3);
    Serial.print("Temperature(主機): ");
    Serial.print(Tb1);
    Serial.print(" *C\t");
    Serial.print("Temperature(電池): ");
    Serial.print(Tb2);
    Serial.print(" *C\t");
    Serial.print("Temperature(環境): ");
    Serial.print(Tb3);
    Serial.println(" *C ");
}

```

上面的草圖的輸出看起來像



代碼說明：

通常，草圖包括庫，聲明感測器總線連接到的引腳以及創建 DallasTemperature 庫的物件。

接下來，我們輸入先前為每個溫度感測器找到的位址。在我們的情況下，我們有以下幾點。

```
uint8_t sensor1[8] = { 0x28, 0x4A, 0x07, 0x49, 0xF6, 0x4F, 0x3C, 0x44 };  
uint8_t sensor2[8] = { 0x28, 0x8B, 0xE7, 0x49, 0xF6, 0xF2, 0x3C, 0xBD };  
uint8_t sensor3[8] = { 0x28, 0x20, 0x6C, 0x49, 0xF6, 0x89, 0x3C, 0x37 };
```

在設定部分，我們透過呼叫 `begin()` 函數初始化函式庫並初始化與 PC 的串列通訊。

在循環部分中，我們只需將命令發送到所有感測器即可使用 `requestTemperatures()` 功能進行溫度轉換。

參考資料：

https://zhuanlan.zhihu.com/p/341234282?utm_id=0