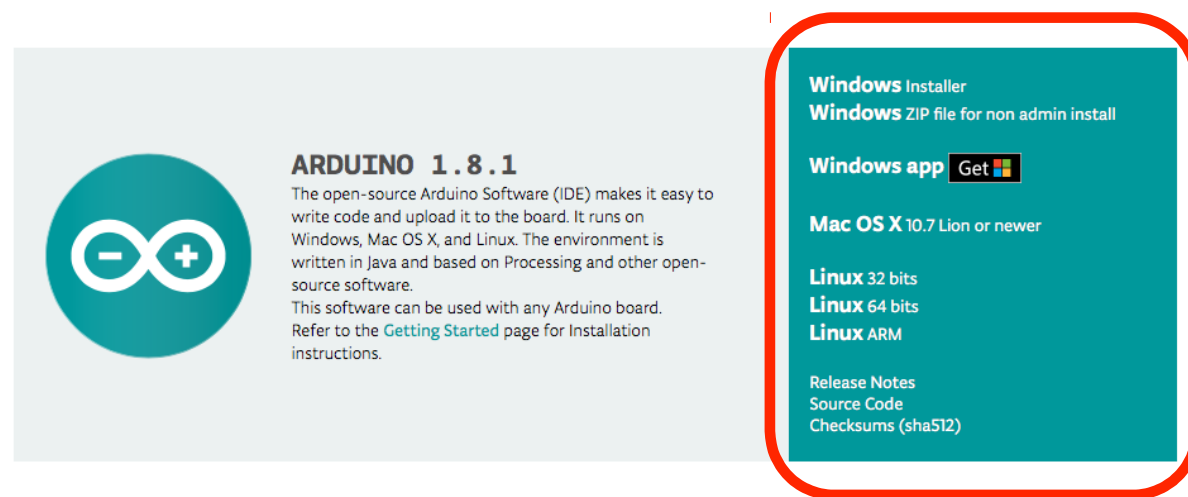


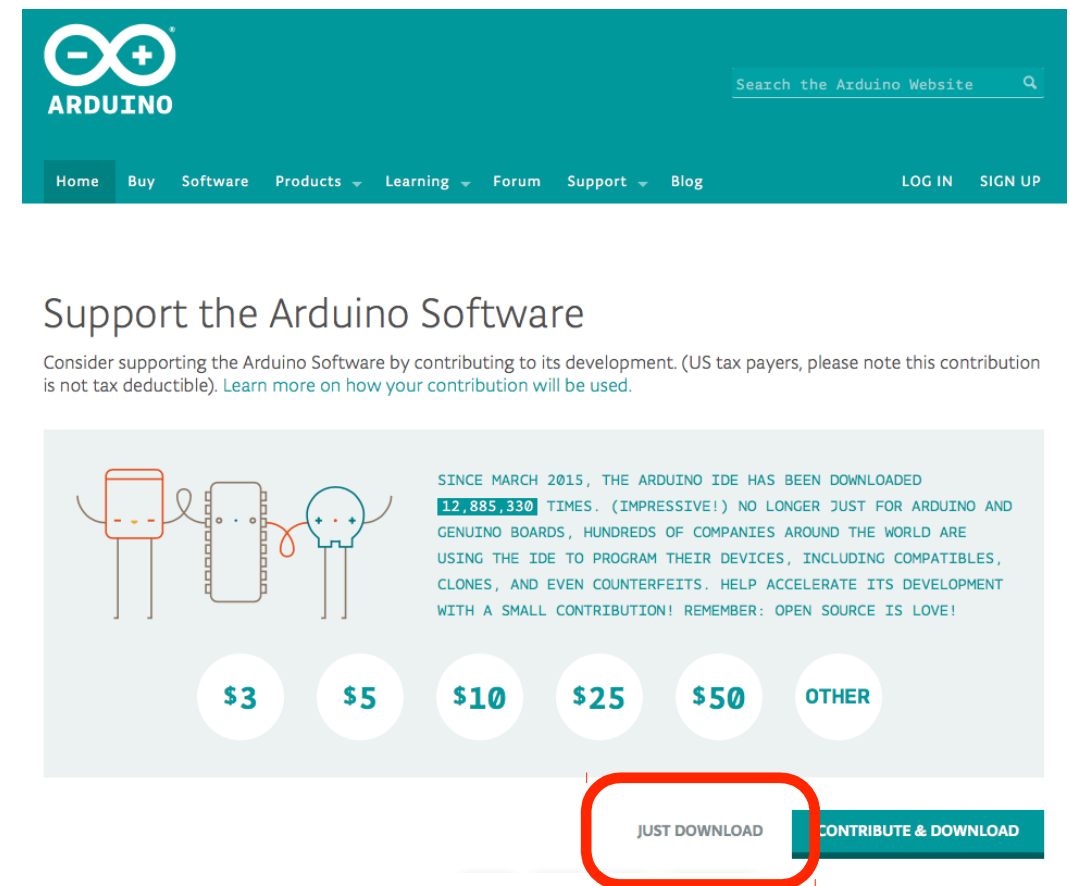
Arduino IDE のダウンロード

①<https://www.arduino.cc/en/Main/Software>

Download the Arduino IDE



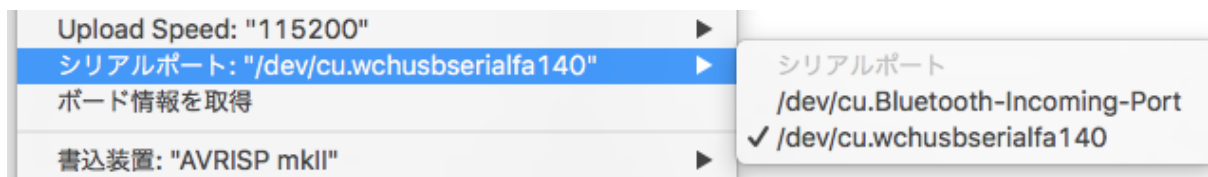
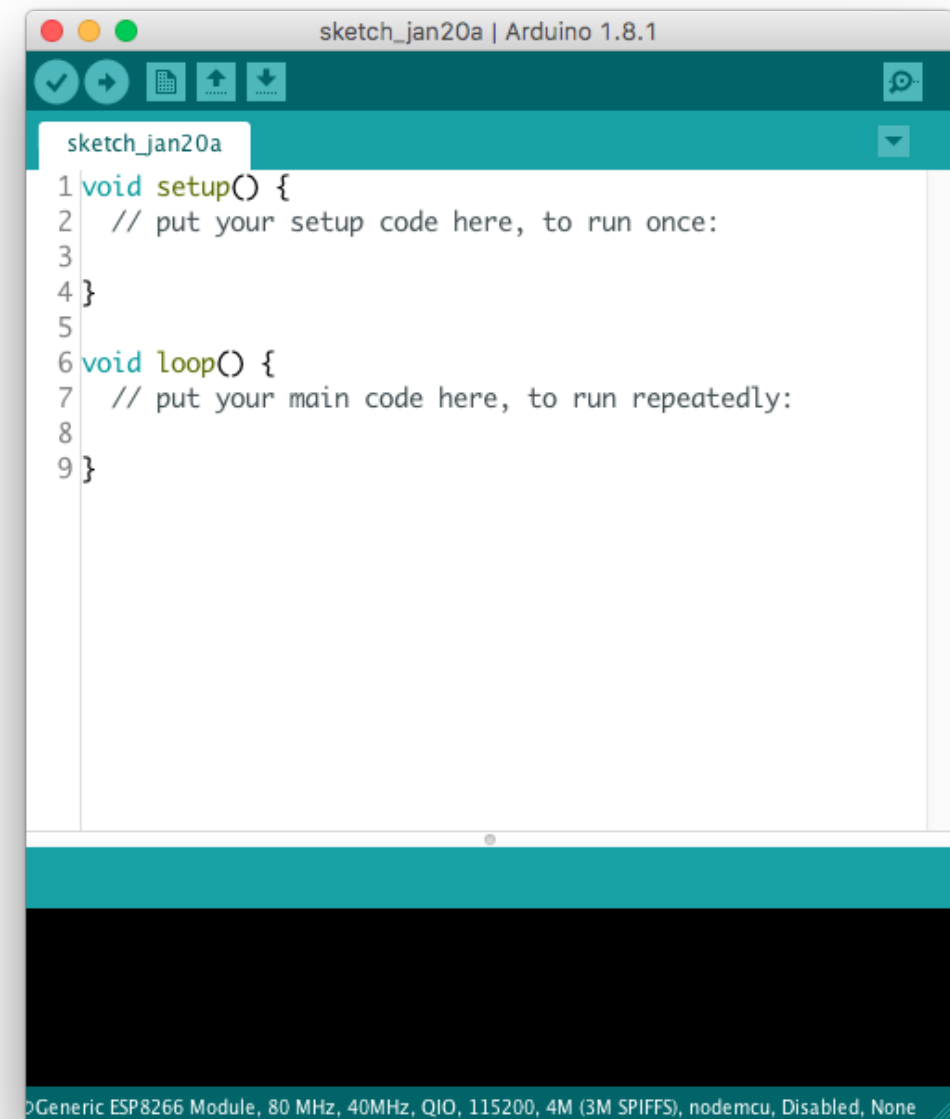
②お使いのOSを選択
(WindowsならInstallerを)



③「JUST DOWNLOAD」
で無料ダウンロード
(隣は寄付用ボタン)

ArduinoをPCに認識させる

- Arduino IDE を起動し、ArduinoとPCをUSBケーブルで接続する。
- PCが正常にArduinoを認識していれば、ツール/シリアルポート 内に項目が増えています。



ポートを認識しない場合 (Mac)

- <https://tzapu.com/ch340-ch341-serial-adapters-macos-sierra/>
- 上記のページより「Winchiphead signed drivers for CH340-CH341, OS X 10.9 to macOS 10.12」をクリックし、ドライバーをダウンロードし、インストールを行う。

ESP開発環境の設定

- 環境設定を開き、追加のボードマネージャURLに下記アドレスを入力し、OKをクリック。
http://arduino.esp8266.com/stable/package_esp8266com_index.json
- ツール/ボード/ボードマネージャ で「esp8226」をインストール（espで検索すると良い）。
- ツール/ボード/Generic ESP8266 Module を選択。

ESP開発環境の設定

- ツール内のモード設定を右図のようにする。

ボード: "Generic ESP8266 Module"
Flash Mode: "QIO"
Flash Frequency: "40MHz"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Debug port: "Disabled"
Debug Level: "なし"
Reset Method: "nodemcu"
Upload Speed: "115200"

Lチカを実行する

- ファイル/スケッチ名/ESP8266/Blink を選択。
- スケッチ内の「LED_BUILTIN」に16を割り当てる。
- 書き込みを行い、Lチカを実行する。



書き込みボタン

Lチカスケッチで行われていること

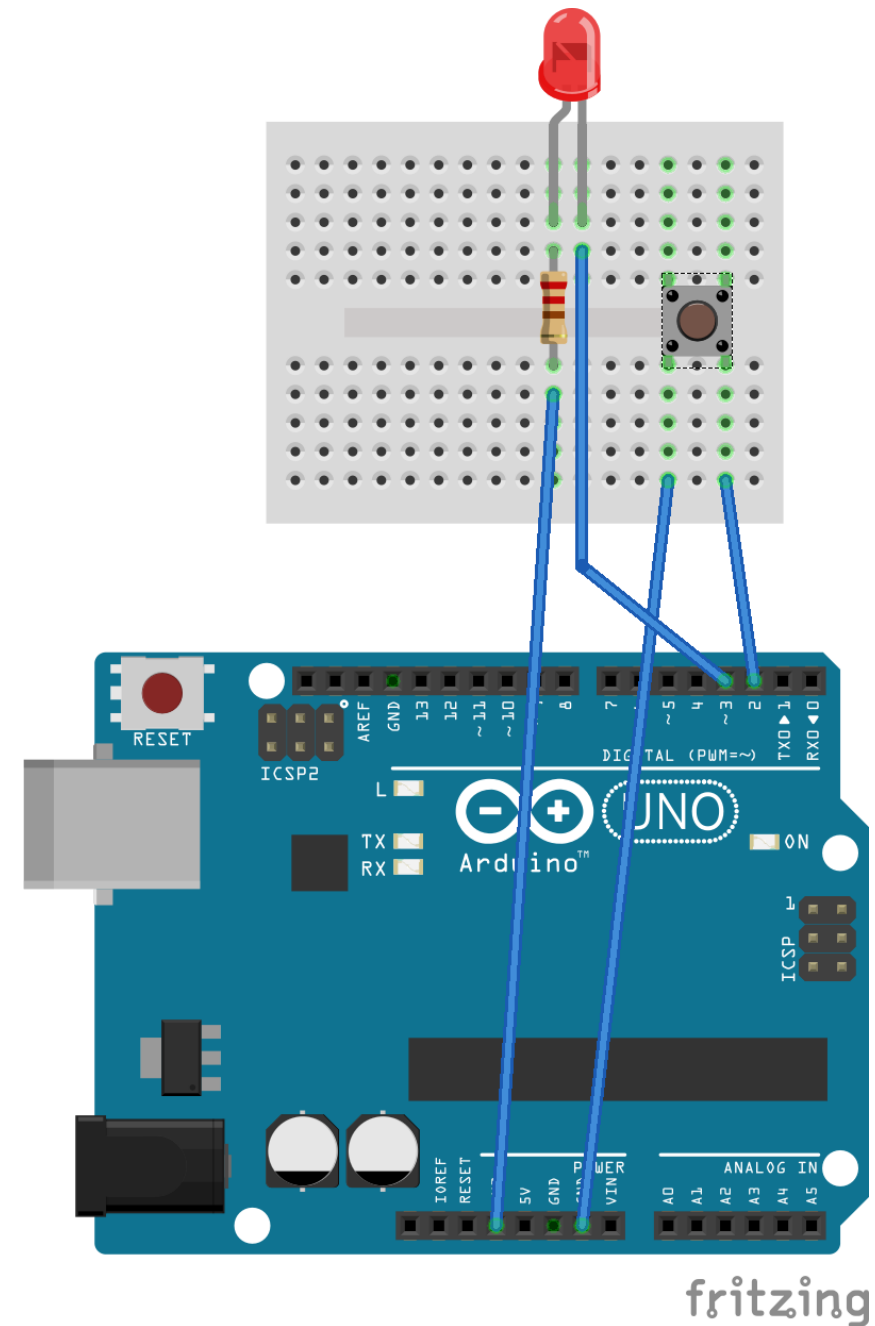
- 最初に「setup」の内容を一度実行する。ここでは準備が行われる。
- 16番ピンを出力モードに設定している。
- 次に「loop」の内容を繰り返し実行する。
- 1つめのdigitalWriteでLEDを消灯し、次の命令の前に1000ミリ秒間停止。
- 2つめのdigitalWriteでLEDを点灯し、2000ミリ秒間停止。
- 再びLED消灯から繰り返す。
- delayの数値を変えて書き込みしてみてください。

```
void setup() {  
    pinMode(16, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(16, LOW);  
    delay(1000);  
    digitalWrite(16, HIGH);  
    delay(2000);  
}
```


スイッチとLEDで 回路を作ろう

- 右図を参考にジャンパ線で回路を作ってください。



スイッチでLEDを 点灯するスケッチ



新規ファイルボタン

- 新規ファイルに右のスケッチを書いて、書き込みをしてください。
- スイッチを押してみてください。LEDは点灯しましたか？

```
1 void setup() {  
2     pinMode(2, OUTPUT);  
3     pinMode(0, INPUT_PULLUP);  
4 }  
5  
6 void loop() {  
7     if (digitalRead(0) == HIGH) {  
8         digitalWrite(2, HIGH);  
9     } else {  
10        digitalWrite(2, LOW);  
11    }  
12 }
```

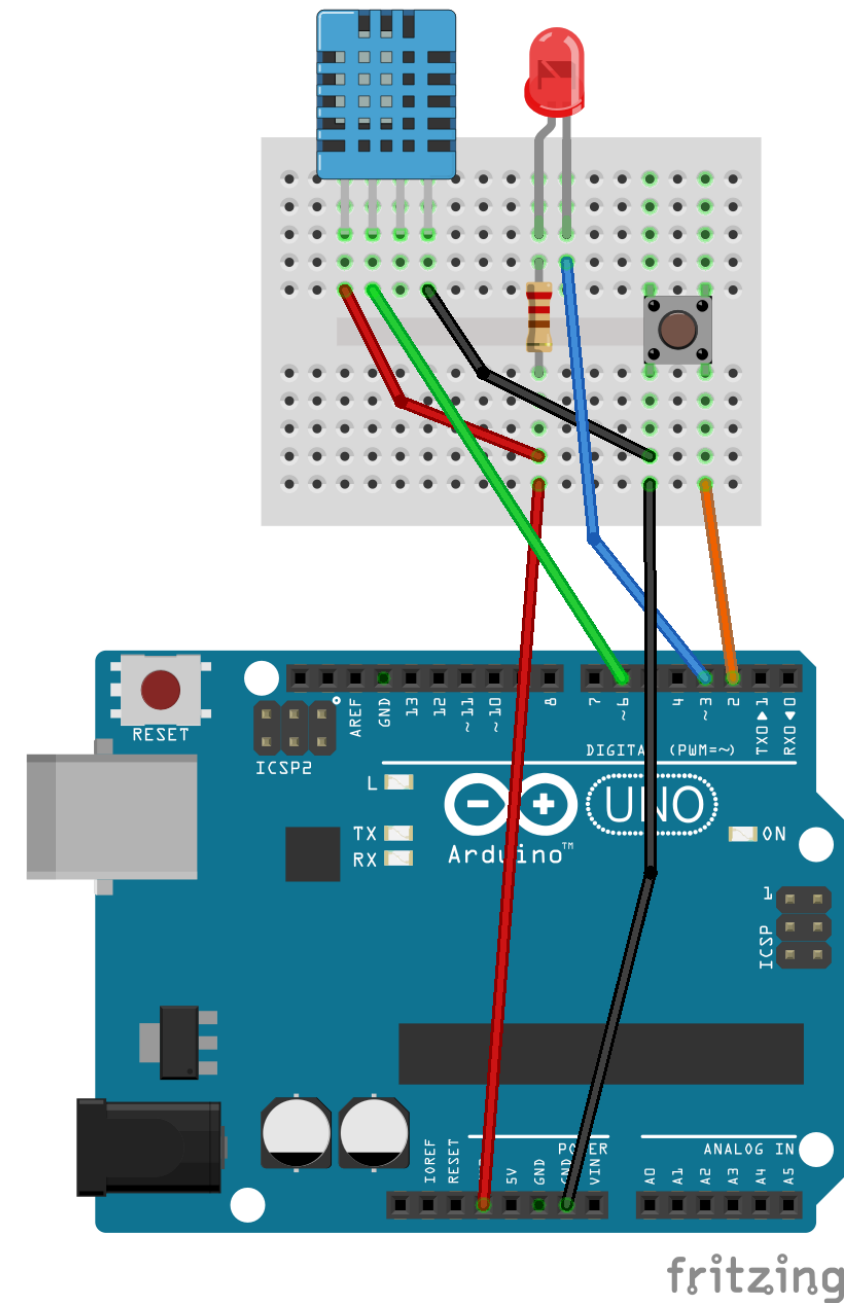
よりプログラムらしく

- 右図のようにスケッチを書き換えて、書き込みを行います。

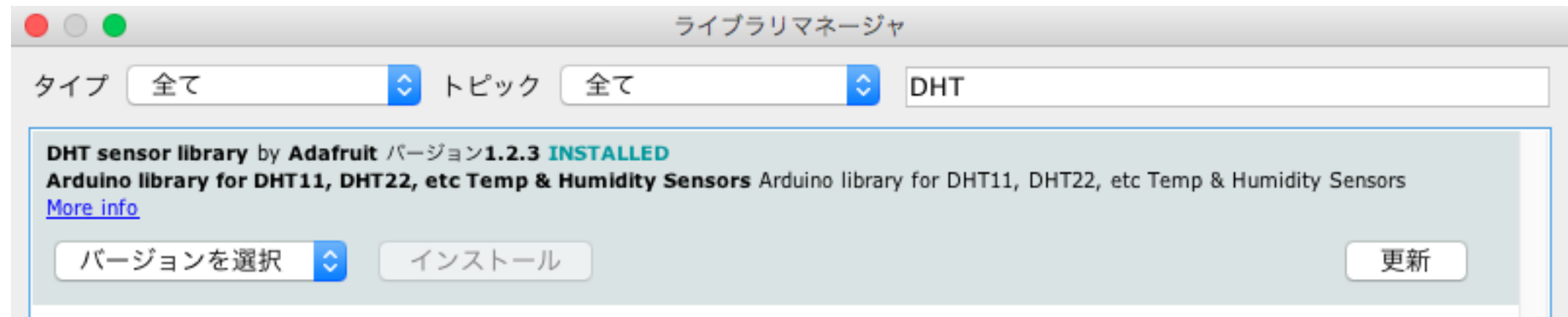
```
1 const int btn = 0;
2 const int led = 2;
3
4 void setup() {
5     pinMode(led, OUTPUT);
6     pinMode(btn, INPUT_PULLUP);
7 }
8
9 void loop() {
10     if (digitalRead(btn) == HIGH) {
11         digitalWrite(led, HIGH);
12     } else {
13         digitalWrite(led, LOW);
14     }
15 }
```

温度センサを使った 回路を作ろう

- 右図を参考にジャンパ線で回路を作ってください。



ライブラリの準備



- ライブラリマネージャを起動（スケッチ/ライブラリをインクルード/ライブラリを管理 を選択）。
- DHT sensor library by Adafruit を探す（DHTと検索すると良い）。
- バージョン1.2.3を選択してインストール。

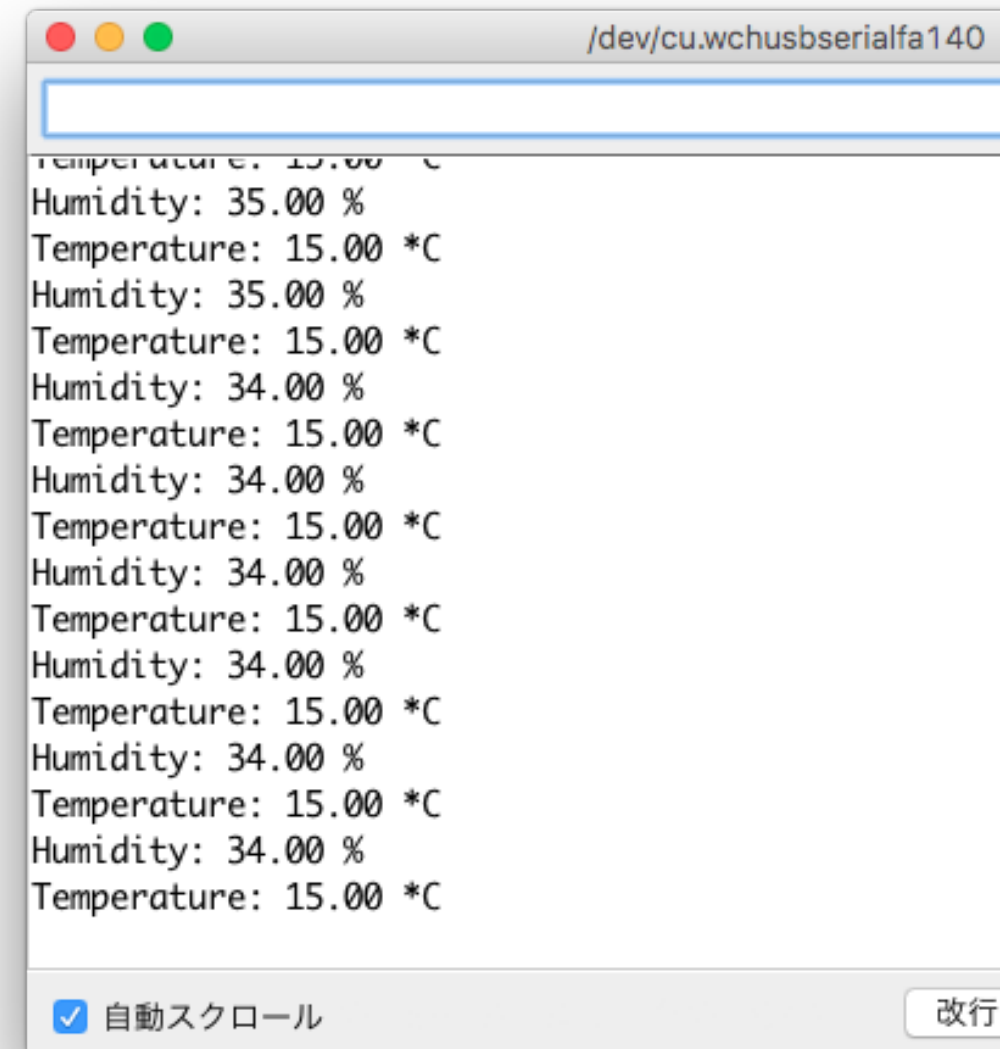
温度、湿度を 表示するスケッチ

```
1 #include <DHT.h>
2
3 #define DHTPIN 12
4 #define DHTTYPE DHT11
5
6 DHT dht(DHTPIN, DHTTYPE);
7
8 void setup() {
9   pinMode(2, OUTPUT);
10  pinMode(0, INPUT_PULLUP);
11
12  Serial.begin(9600);
13  Serial.println("DHT11 test!");
14  dht.begin();
15 }
16
17 void loop() {
18   float h = dht.readHumidity();
19   float t = dht.readTemperature();
20
```

```
21   if (isnan(h) || isnan(t)){
22     Serial.println("Failed to read from DHT sensor!");
23     return;
24   }
25   Serial.print("Humidity: ");
26   Serial.print(h);
27   Serial.println(" %\t");
28   Serial.print("Temperature: ");
29   Serial.print(t);
30   Serial.println(" *C ");
31
32   if (digitalRead(0) == HIGH) {
33     digitalWrite(2, HIGH);
34   } else {
35     digitalWrite(2, LOW);
36   }
37
38   delay(1000);
39 }
```

動作確認

- シリアルモニタを起動（ツール/シリアルモニタ と選択）。
- 1秒（1000ミリ秒）ごとに、湿度と温度の情報（print、printlnで書いた内容）が追記される。
- センサを手指や吐息で温めて、表示温度が変わることを確かめてください。



The screenshot shows a serial monitor window titled "/dev/cu.wchusbserialfa140". The window displays a stream of text output from a program, showing temperature and humidity readings. The data is as follows:

Temperature	Humidity
15.00 °C	35.00 %
15.00 °C	35.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %
15.00 °C	34.00 %

At the bottom of the window, there is a checkbox labeled "自動スクロール" (Auto Scroll) which is checked, and a button labeled "改行" (New Line).