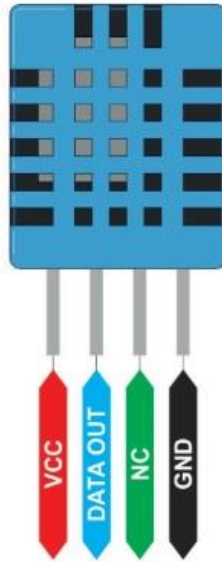


การทดลองที่ 4

การใช้งาน DHT11 Humidity and Temperature Sensor กับ NodeMCU โดยการใช้ Arduino IDE

DHT11 (DHT11 Humidity and Temperature Sensor) ที่ใช้วัดความชื้นและอุณหภูมิ

Sensor วัด อุณหภูมิ กับ ความชื้นใน Series นี้มีออกมาหลายรุ่นครับ เช่น DHT11 DHT 22 DHT21



ความหมายของขา

Pin 1 ต่อกับ VDD ที่ระดับแรงดัน 3 - 5.5 VDC

Pin 2 ต่อเป็นขา DATA

Pin 3 ไม่ได้ใช้

Pin 4 ลงกราวด์

Specification ของ DHT11

- ย่านวัดความชื้น 20-90% RH โดยมีค่าความแม่นยำ $\pm 5\%$ RH ความละเอียดในการวัด 1% แสดงผลแบบ 8 บิต
- ย่านวัดอุณหภูมิ 0 -50 องศาเซลเซียส โดยมีค่าความแม่นยำ ± 2 องศาเซลเซียส ความละเอียดในการวัด 1 องศาเซลเซียส แสดงผลแบบ 8 บิต
- มี PIN 4 ขารายละเอียดดังรูปด้านบน
- กินกระแส 0.5 - 2.5 mA (ขณะทำการวัดค่า) ที่ระดับแรงดัน 3 - 5.5 VDC
- อ่านค่าสัญญาณ (Sample Rate) ทุก 1 วินาที

รายละเอียดของ Spec อื่นๆ นอกเหนือจากนี้ดูในที่ Manual

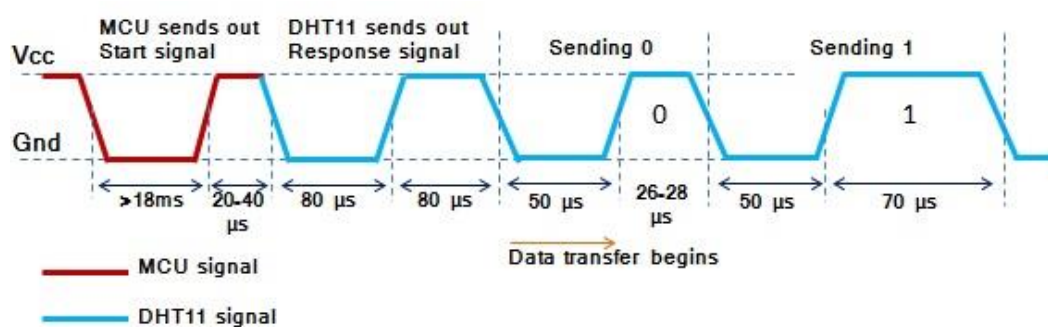
ในการการต่อวัดแบบปกติ คือ ระยะห่างระหว่าง Sensor กับตัว Arduino ห่างกันไม่เกิน 20 เมตร จะต้องใช้ Pull up resistor ขนาด 5kohm (ว่าง่ายๆ คือต่อ R 5k ไว้กับแหล่งจ่ายแรงดันและต่อเข้าไปที่ขา DATA ด้วย) โดยใช้แหล่งจ่ายแรงดัน VDD ขนาด 3-5.5 VDC ครับ ซึ่งข้อดีคือจะทำให้ DHT11 นี้สามารถใช้งานได้กับ Arduino และ ESPNODE MCU ได้หลายรุ่น ทั้งรุ่นที่มีแรงดัน 3.3 VDC

ที่นี้มาถึงวิธีการส่งข้อมูลของ DHT11 กันครับ อุปกรณ์ตัวนี้ใช้การสื่อสาร แบบสายเส้นเดียว 2 ทิศทาง (single-wire bi-directional)

ในที่นี้ จะเรียก MCU หรือ Arduino หรือ ESP NodeMCU ว่า Master และ DHT11 ว่า Slave ส่วนการใช้ pull up resistor นั้นต้องการให้เวลาทำงานปกติในขณะที่ไม่มีอุปกรณ์ที่ต่ออยู่กับสายสื่อสารของเรา ระดับแรงดันของสัญญาณในสาย Data คือแรงดันระดับ "สูง" และจะมีแรงดันในระดับต่ำเมื่อมีอุปกรณ์ (จะเป็น Master หรือ Slave ก็ได้) ดึงสัญญาณลงในระดับ "ต่ำ" ดังนั้นหากเรามี Scope วัดระดับสัญญาณได้เป็น "สูง" ตลอดเวลา ก็หมายความว่าอุปกรณ์ของเราอาจจะผิดปกติ

โปรโตคอลของการสื่อสารแบบ single-wire bi-directional อันดับแรก Master จะส่ง Start signal ที่เป็นแรงดันไฟฟ้าระดับต่ำอย่างน้อย 18 ไมโครวินาที ไปที่ Slave เพื่อให้ Slave เข้าใจว่าจะเริ่มส่งแล้วนะ แล้วรอไป 20-40 ไมโครวินาทีเพื่อรอ Slave ตอบกลับ

ที่นี้เพื่อให้ Master รู้ว่า Slave ก็พร้อม Slave จะส่งแรงดันระดับต่ำกลับไปบ้าง การส่งแรงดันจาก Slave กลับไปจะนาน 80 ไมโครวินาที จากนั้นจะรออีก 80 ไมโครวินาที ก่อนที่จะส่งข้อมูลบิตแรก มาถึงตรงนี้ ข้อมูลยังไม่ถูกส่งเลยนะครับ แต่พร้อมจะส่งแล้ว



คราวนี้มาดูครับว่าจะส่งบิต "0" กับ บิต "1" อย่างไร สำหรับการส่งบิตเป็น "0" ตัว Slave จะดึงระดับแรงดันลงต่ำนาน 50 ไมโครวินาที และปล่อยเป็นระดับ "สูง" นาน 26-28 ไมโครวินาที ดังรูปด้านด้านบนครับ (ดูช่วง Sending 0)

แต่ถ้าเป็นการส่งข้อมูลเป็น "1" ตัวส่งจะดึงสายสัญญาณลงระดับต่ำ 50 ไมโครวินาที และปล่อยให้เป็นระดับสูงนาน 70 ไมโครวินาที (ดูช่วง Sending 1) เอาละครับ ทีนี้รู้ว่าแต่ละบิต DH11 ส่งมาเป็นบิต "0" หรือ "1" มา และส่งมาจนครบข้อมูลหนึ่งชุดครับ

ในแต่ละชุดของข้อมูลที่ส่งมาจาก DH11 ตัว MCU รับข้อมูลแล้วจะต้องเอามาแปลงก่อนจะรับ ว่าข้อมูลที่ส่งมามันแปลว่าอะไร แต่ละชุดข้อมูลจะยาว 40 บิต และใช้เวลาส่งประมาณ 40 มิลลิวินาทีครับ

ใน 40 บิตที่ส่งมา ประกอบด้วย " 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum"

ตัวอย่างที่ 1: เมื่อการรับข้อมูล 40 บิตถูกต้อง :

0011 0101	0000 0000	0001 1000	0000 0000	0100 1101
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

การคำนวณ :

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101$$

Received data is correct :

Humidity : 0011 0101 = 35H = 53%RH

Temperature : 0001 1000 = 18H = 24°C

ตัวอย่าง 2: เมื่อการรับข้อมูล 40 บิตผิดพลาด :

0011 0101	0000 0000	0001 1000	0000 0000	0100 1001
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

การคำนวณ :

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101$$

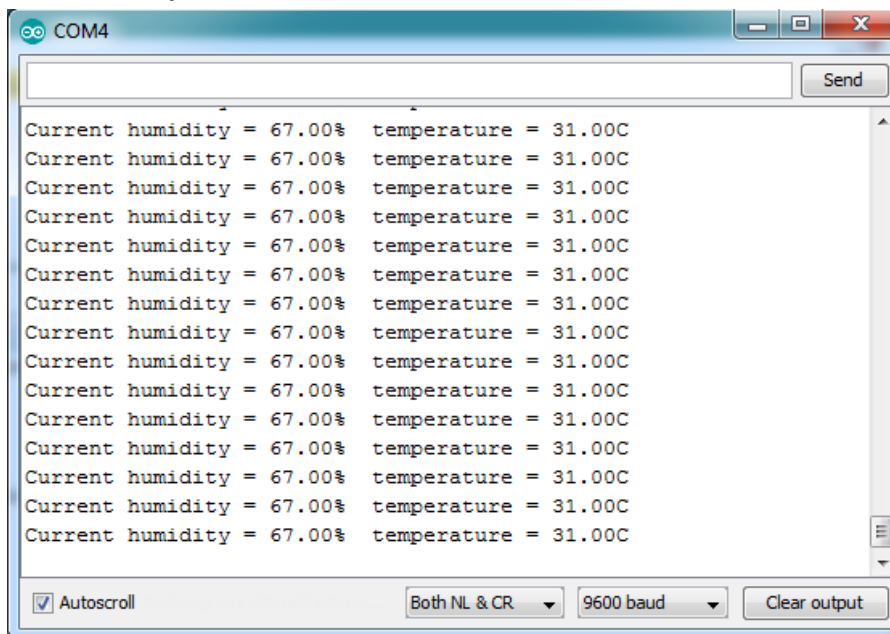
$$01001101 \neq 0100\ 1001$$

สำหรับ library ของ DHT11 ดูได้จากในเว็บ

โปรแกรมทดสอบ

```
#include "DHT.h" // including the library of DHT11 temperature and humidity sensor
#define DHTTYPE DHT11 // DHT 11
#define dht_dpin 0
DHT dht(dht_dpin, DHTTYPE);
void setup(void)
{
  dht.begin();
  Serial.begin(9600);
  Serial.println("Humidity and temperature\n\n");
  delay(700);
}
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print("Current humidity = ");
  Serial.print(h);
  Serial.print("% ");
  Serial.print("temperature = ");
  Serial.print(t);
  Serial.println("C ");
  delay(800);
}
```

ผลการทำงานดูได้จาก Serial monitor



อ้างอิง

1. <https://www.arduitronics.com/article/13/การใช้งาน-dht11-humitdity-and-temperature-sensor-กับ-บอร์ด-arduino>
2. <https://roboindia.com/tutorials/DHT11-NodeMCU-arduino>
3. DHT11.pdf