

請收到此模組後，在濕度大於80%環境下放置12小時，以確保精度恢復。

AHT10，新一代溫濕度感測器在尺寸與智慧方面建立了新的標準：它嵌入了適於回流焊的雙列扁平無引腳SMD封裝，底面4 x 5mm，高度1.6mm。感測器輸出經過標定的數位信號，標準I²C格式。

AHT10 配有一個全新設計的ASIC專用晶片、一個經過改進的MEMS半導體電容式濕度傳感元件和一個標準的片上溫度傳感元件，其性能已經大大提升甚至超出了前一代感測器的可靠性水準，新一代溫濕度感測器，經過改進使其在惡劣環境下的性能更穩定。

特性：

1. 模組尺寸： 16*11 mm
2. 介面類別型： I²C
3. 工作電壓： 1.8 - 6.0 V
4. 介面尺寸： 4*2.54mm間距
5. 濕度精度： 典型±2%
6. 濕度解析度： 0.024%
7. 溫度精度： 典型±0.3°C
8. 溫度解析度： 典型0.01°C
9. 工作溫度： -40°C--85°C



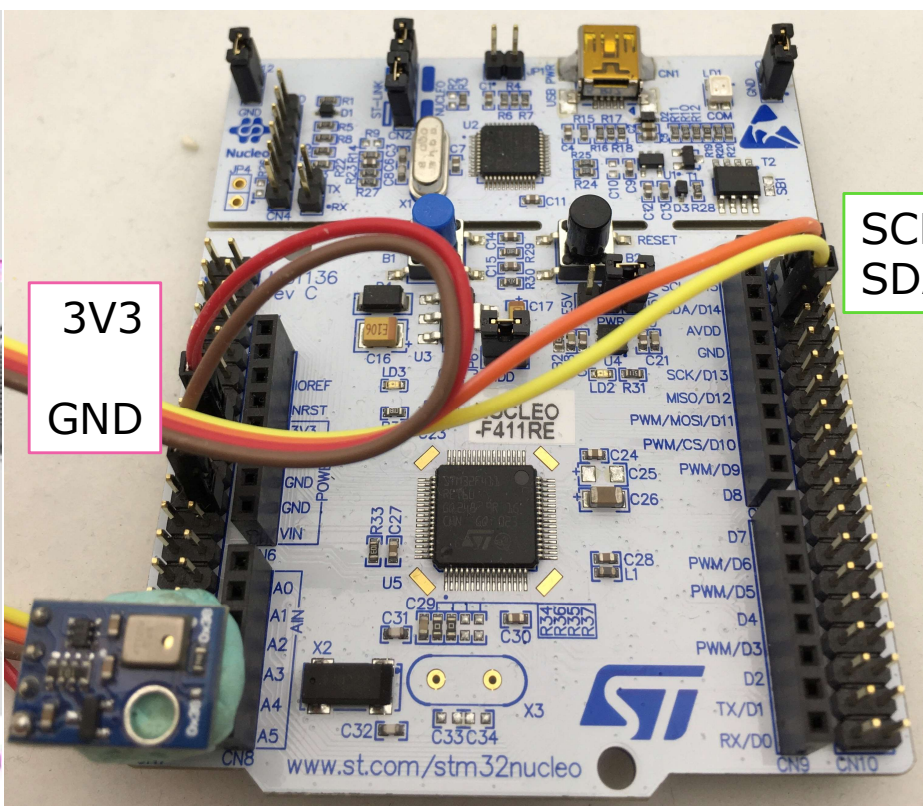


接線實體照片



CN7		CN6			
PC10	1	2		PC11	
PC12	3	4		PD2	
VDD	5	6		E5V	
BOOT0	7	8		GND	
NC	9	10	1	NC	NC
NC	11	12	2	IOREF	IOREF
PA13	13	14	3	RESET	RESET
PA14	15	16	4	+3V3	+3V3
PA15	17	18	5	+5V	+5V
GND	19	20	6	GND	GND
PB7	21	22	7	GND	GND
PC13	23	24	8	VIN	VIN
PC14	25	26		NC	
PC15	27	28	1	PA0	A0
PH0	29	30	2	PA1	A1
PH1	31	32	3	PA4	A2
VBAT	33	34	4	PB0	A3
PC2	35	36	5	PC1	A4
PC3	37	38	6	PC0	A5

■ Arduino



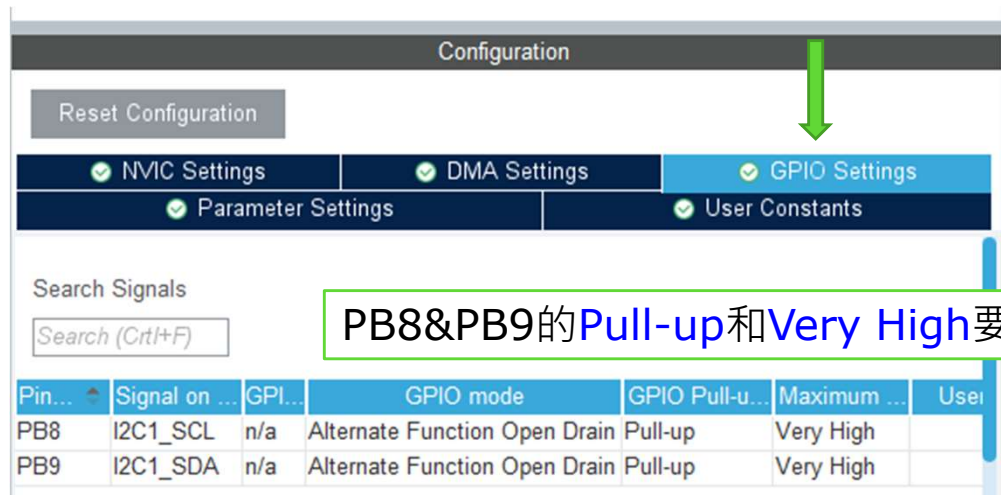
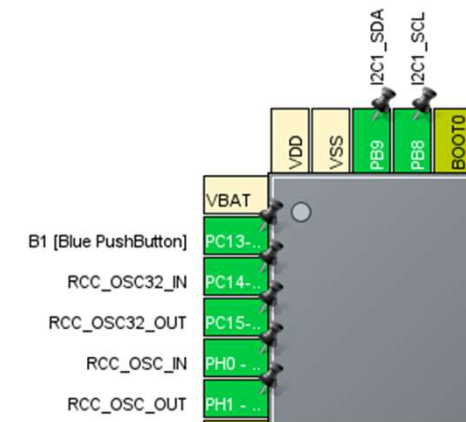
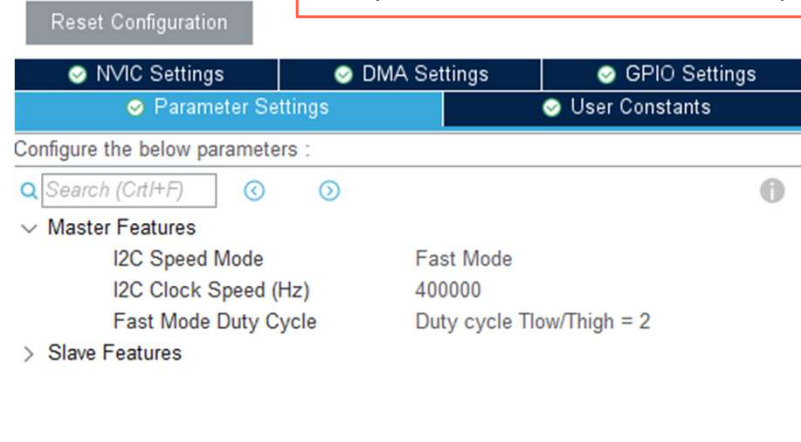
SCL
SDA

CN5		CN10			
PC9	1	2		PC8	
D15	3	4		PC6	
D14	5	6		PC5	
AVDD	7	8		U5V	
GND	9	10		NC	
D13	11	12		PA12	
D12	13	14		PA11	
D11	15	16		PB12	
D10	17	18		NC	
D9	19	20		GND	
D8	21	22		PB2	
D7	23	24		PB1	
D6	25	26		PB15	
D5	27	28		PB14	
D4	29	30		PB13	
D3	31	32		AGND	
D2	33	34		PC4	
D1	35	36		NC	
D0	37	38		NC	

■ Morpho

*.ioc檔設定，容易出錯的地方

如果要使用和Arduino相容的腳位，I²C1在PB8 & PB9



PB8&PB9的Pull-up和Very High要設起來

注意這兩組設定
是在不同的選項之下

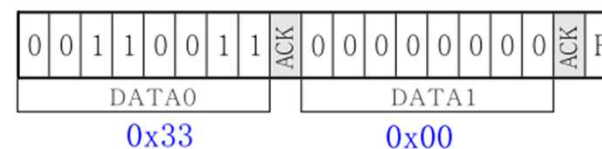
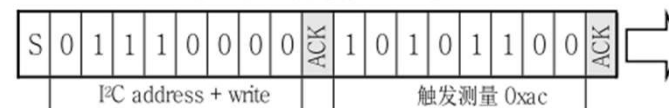
Datasheet遺漏的片段!

https://github.com/MoonFox2006/AHT10_Test/blob/master/src/AHT10.cpp

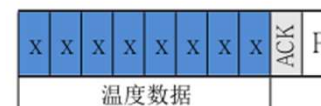
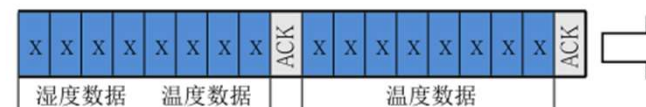
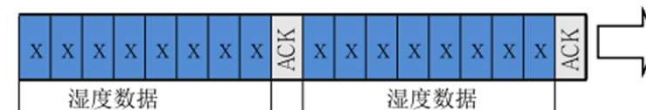
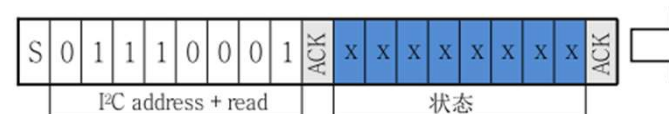
```
bool AHT10::begin() {
    Wire.beginTransmission(AHT10_ADDR);
    Wire.write(0xE1);
    Wire.write(0x08);
    Wire.write(0x00);
    return Wire.endTransmission() == 0;
}
```

触发测量数据

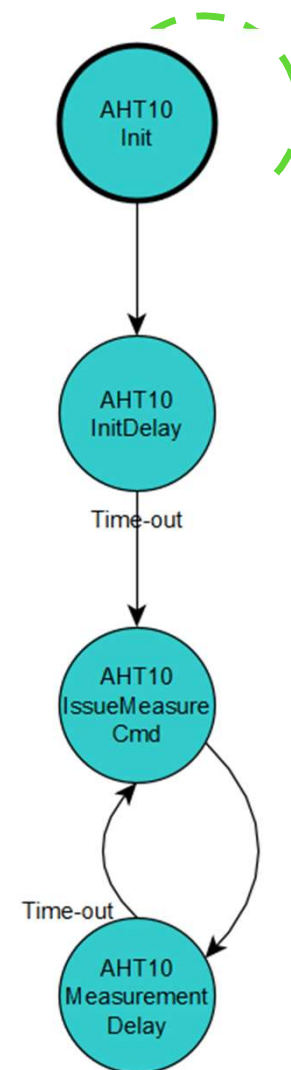
$$0x70 = 0x38 \ll 1$$



读取温湿度数据

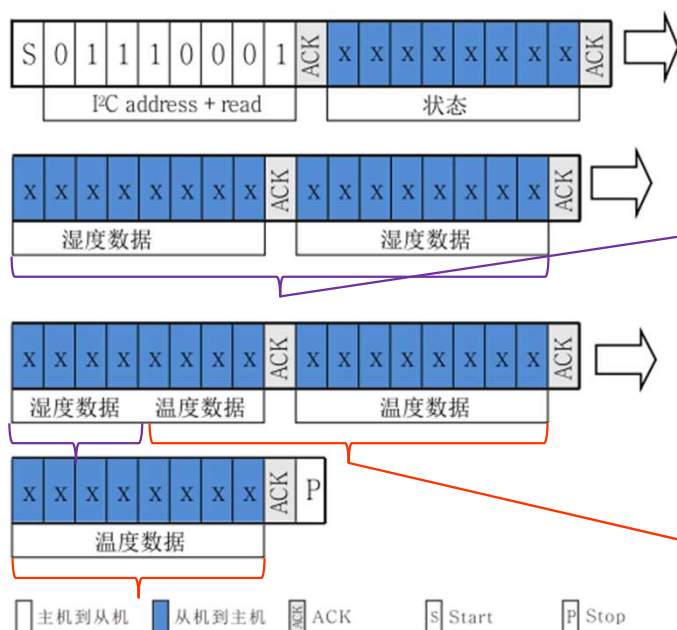


□ 主机到从机 ■ 从机到主机 ACK ACK S Start P Stop



數據換算公式

读取温湿度数据



6.1 相对湿度转换

相对湿度 RH 都可以根据 SDA 输出的相对湿度信号 S_{RH} 通过如下公式计算获得 (结果以 %RH 表示)。

$$RH[\%] = \left(\frac{S_{RH}}{2^{20}} \right) * 100\%$$

$S_{RH} * 0.000095$

6.2 温度转换

温度 T 都可以通过将温度输出信号 S_T 代入到下面的公式计算得到 (结果以温度 $^{\circ}\text{C}$ 表示)：

$$T(^{\circ}\text{C}) = \left(\frac{S_T}{2^{20}} \right) * 200 - 50$$

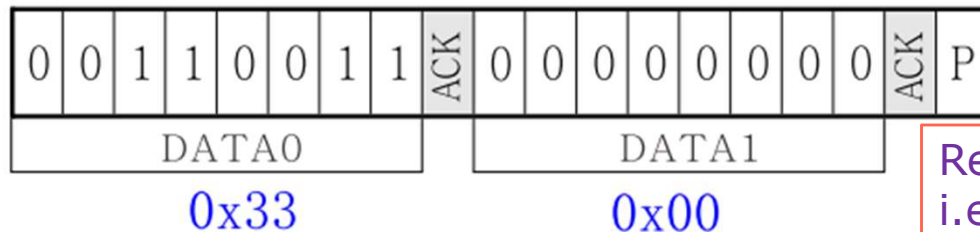
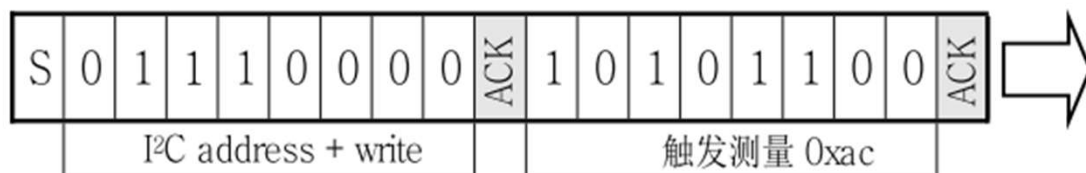
$S_T * 0.000191 - 50$

AHT10 I²C Address

触发测量数据

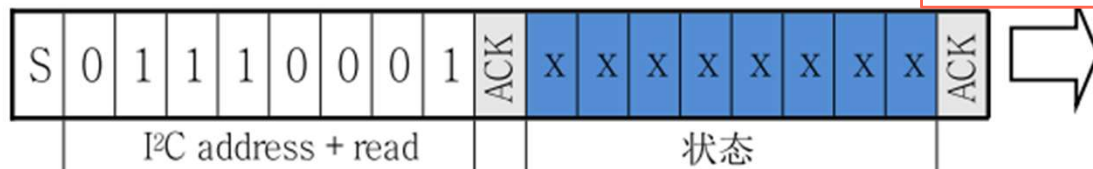
$0x70 = 0x38 \ll 1$

模組上標示的0x38，和Datasheet中是不一致的



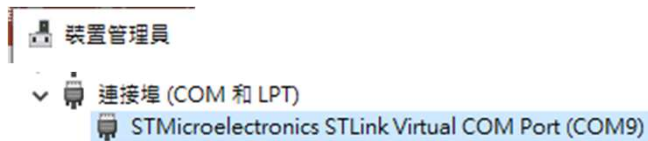
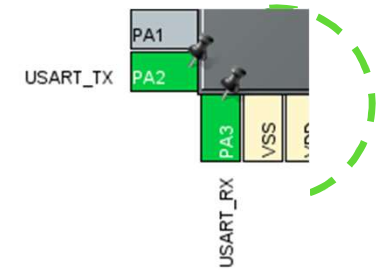
读取温湿度数据

Read/write bit和ACK bit，HAL底層都會處理
i.e. 0x70和0x71的I²C Address是一致的，LSB都會被忽略！
HAL_I2C_Master_Receive(&hi2c1, **0x70**, buf_I2C_Rx, 6, HAL_MAX_DELAY);
等於
HAL_I2C_Master_Receive(&hi2c1, **0x71**, buf_I2C_Rx, 6, HAL_MAX_DELAY);

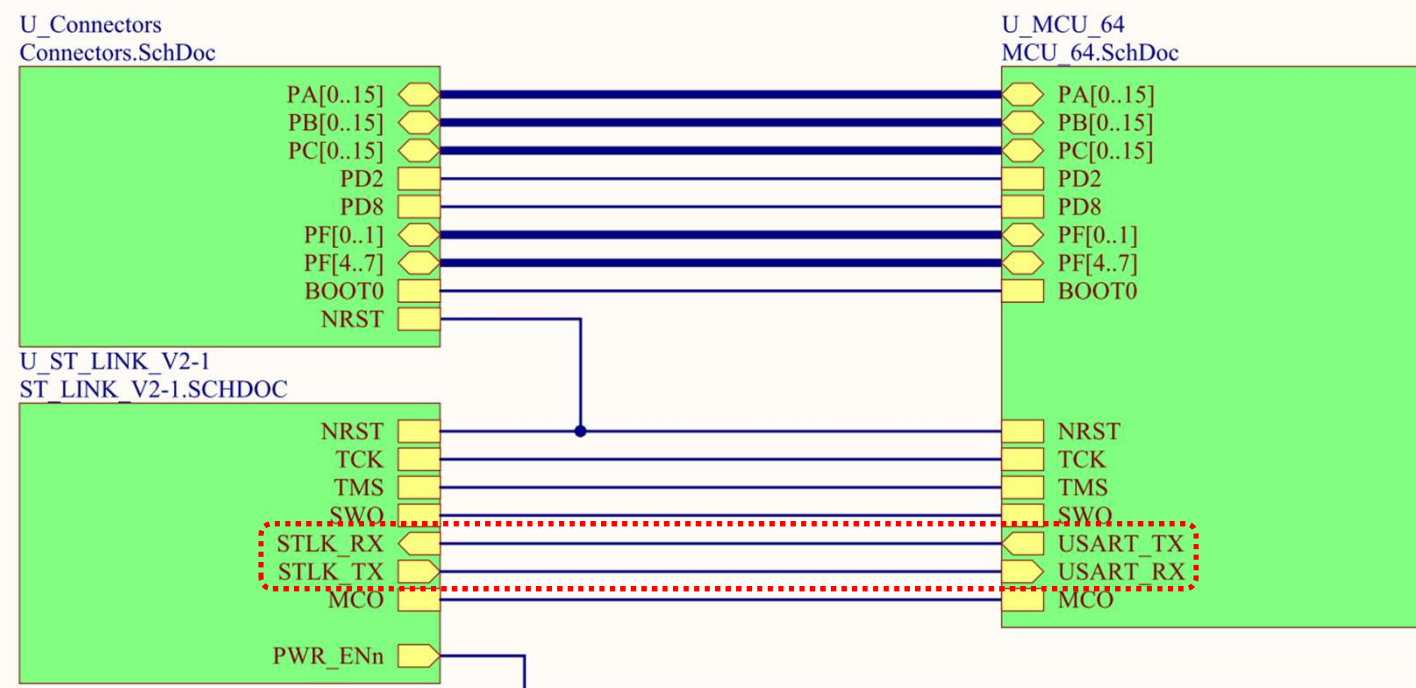


Virtual COM Port Default@UART2

Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum o...	User Label
PA2	USART2_TX	n/a	Alternate Fu...	No pull-up a...	Very High	USART_TX
PA3	USART2_RX	n/a	Alternate Fu...	No pull-up a...	Very High	USART_RX

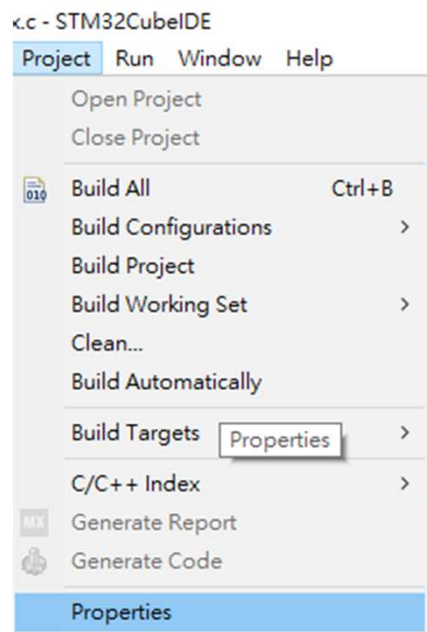


使用時注意它和Debug Mode是使用同一個Virtual COM Port !

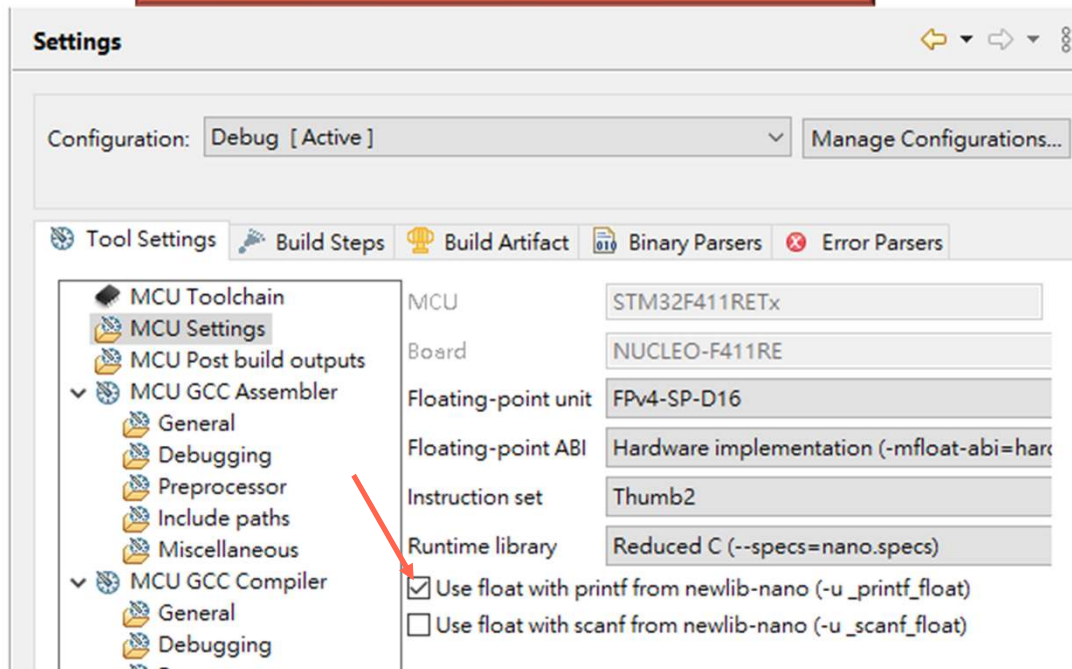


使用sprintf時，IDE會提示打開newlib的選項

每新增加一個project，需要再設一次



IDE Properties for I2C_AHT10



```
sprintf((char*)buf_UART, "%.2f", tmp);
```

The float formatting support is not enabled, check your MCU Settings from "Project Properties > C/C++ Build > Settings > Tool Settings", or add manually "-u _printf_float" in linker flags.