陽明交大
NYCU

物聯網裝置與平台
IoT Devices and Platforms

曾煜棋、吳昆儒

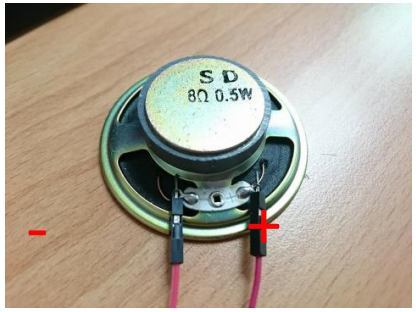
National Yang Ming Chiao Tung University

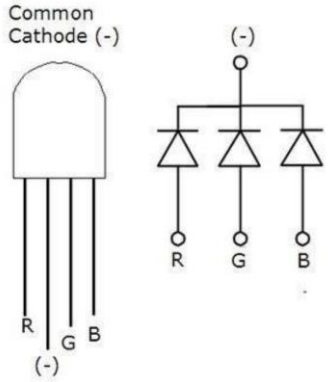
Ch 5, Sensors (2) - Summary

- Arduino IDE and how it interacts with the external world
  - Sensors
- Understanding this course:
  - Discussion: upload to e3
  - Quiz: show your code to TA
    - For remote-access, use discord to interact with TA

Labs (Last week)

1. **toneMelody: play a melody with a piezo speaker**
2. **tonePitchFollower: play a pitch on a piezo speaker depending on an analog input.**
3. **ReadASCIIString (RGB LED): Parse a comma-separated string of ints to fade an LED.**
4. **Ultrasonic sensor**
   - Measure distance





Ultrasonic

Labs (This week)

1. **DHT-11 (thermometer + hygrometer)**: read the values for the temperature and humidity
2. **ASCIITable**: Demonstrates the advanced serial printing functions

DHT-11

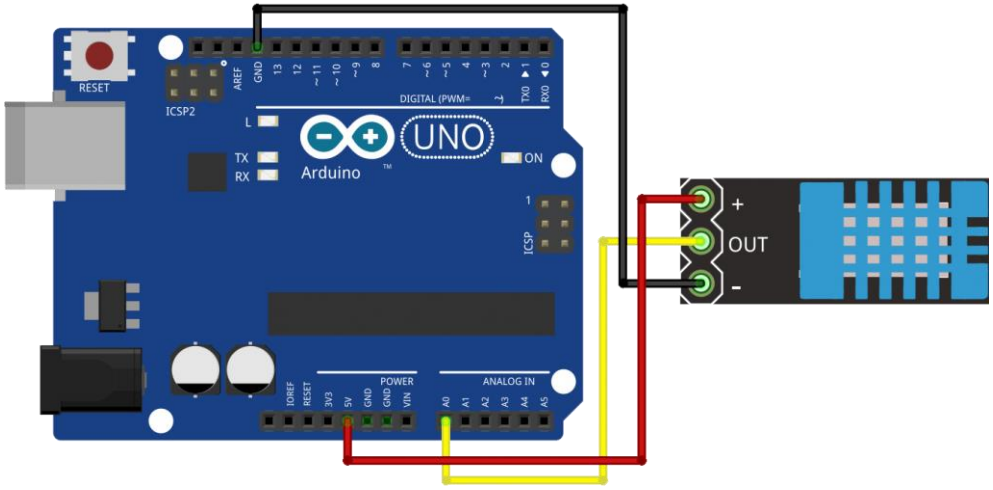
Thermometer + Hygrometer

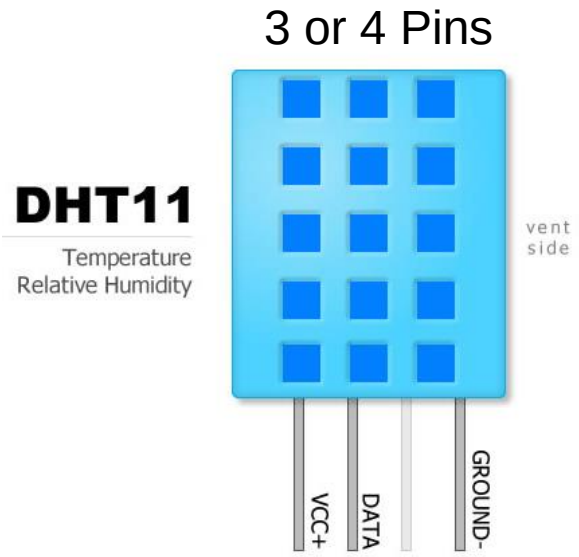
Lab1. DHT-11 (thermometer + hygrometer)

read the values for the temperature and humidity

Lab1. DHT-11

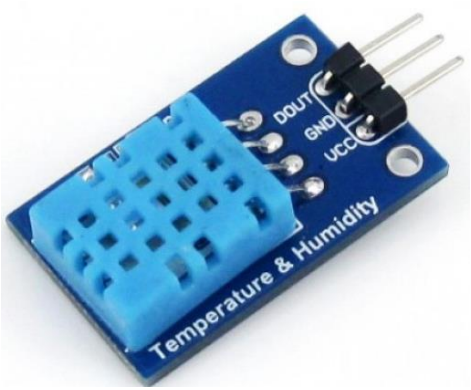
- Goal: Use sensor board with DHT-11 in a single chip to read the values for the temperature and humidity.
- Hardware Required
  - Arduino Board
  - DHT-11



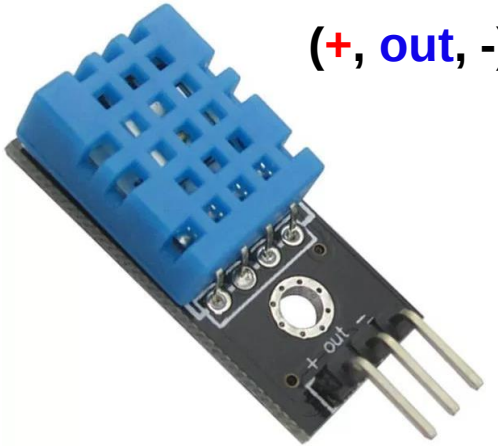


VCC, Data, GND
(+, out, -)

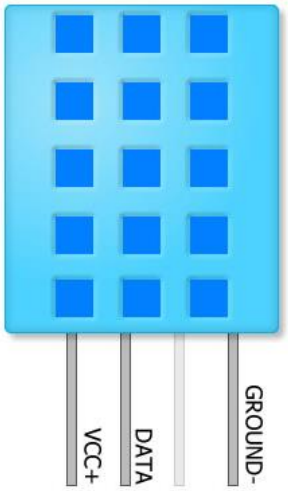
Lab1. Different kinds of DHT-11



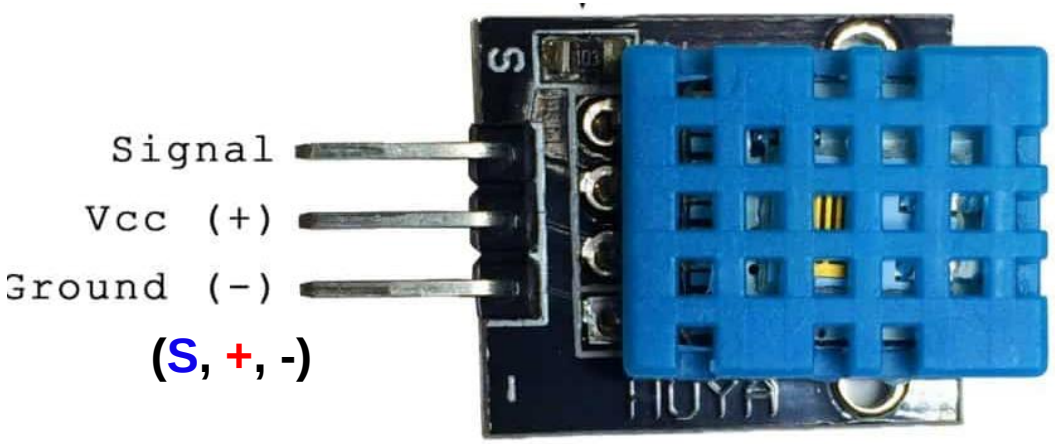
(DOUT, GND, VCC)



(+, out, -)



(VCC, Data, GND)



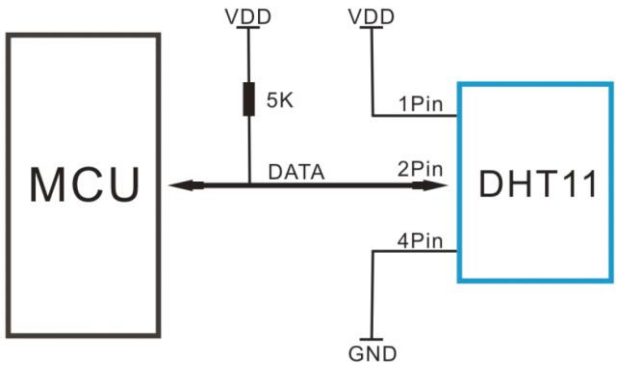
(S, +, -)

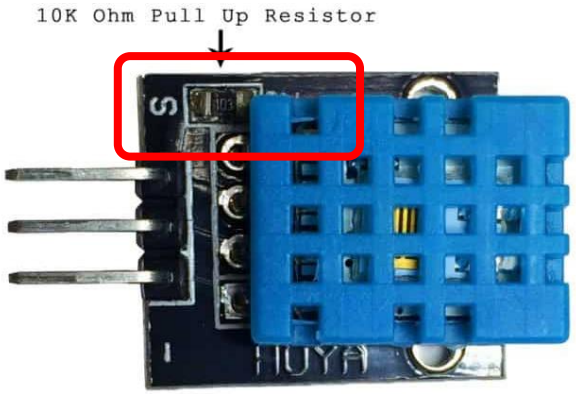
Lab1. DHT-11

What is DHT-11?

- Measuring both **temperature** and **relative humidity (RH)** and provide fully calibrated digital outputs.
- Pull-up resistor: $R_{PU} > 5k$

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row



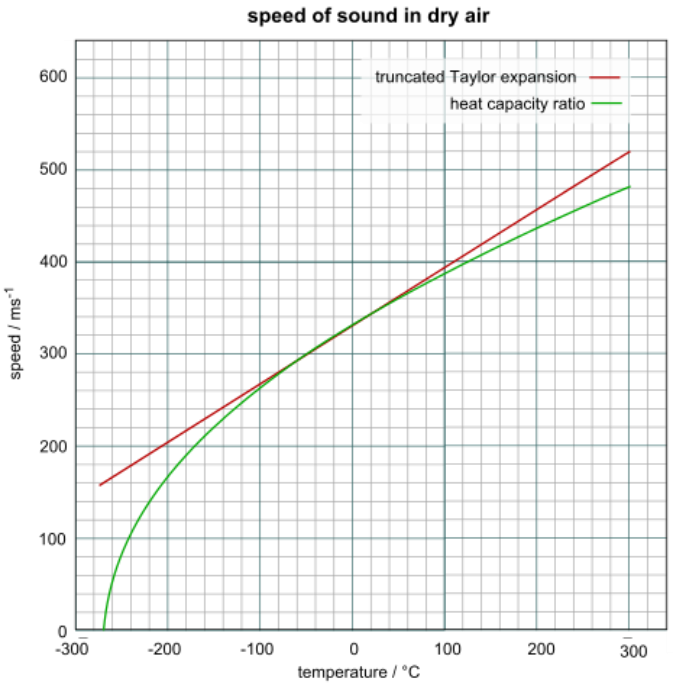


Lab1. DHT-11

- Speed of sound
  - At 20°C (68°F), the speed is 343 m/s.
  - The approximate speed of sound (c) can be calculated from:

$$\mathbf{c_{air} = (331.3 + 0.606 * \theta)\ (m/s)}$$

where θ is the temperature in degrees Celsius (°C).



$$Speed = \frac{Distance}{Time}$$

$$34300 = \frac{Distance}{Time/2}$$

$$17150 = \frac{Distance}{Time}$$

$$17150 \times Time = Distance$$

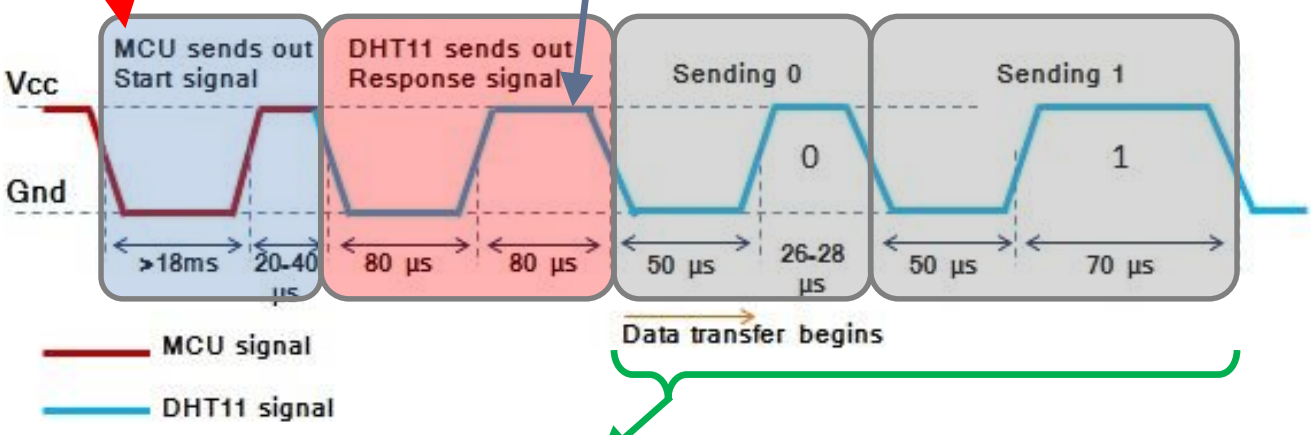
http://en.wikipedia.org/wiki/Speed_of_sound
https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

Lab1. DHT-11

- Overall Communication Process
  - "Start" and "Response" signals.
  - Data (40-bit) = Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp. + Checksum Byte.

1. Send “Start signal”

2. pull up voltage and keeps it for 80us and prepares for data transmission

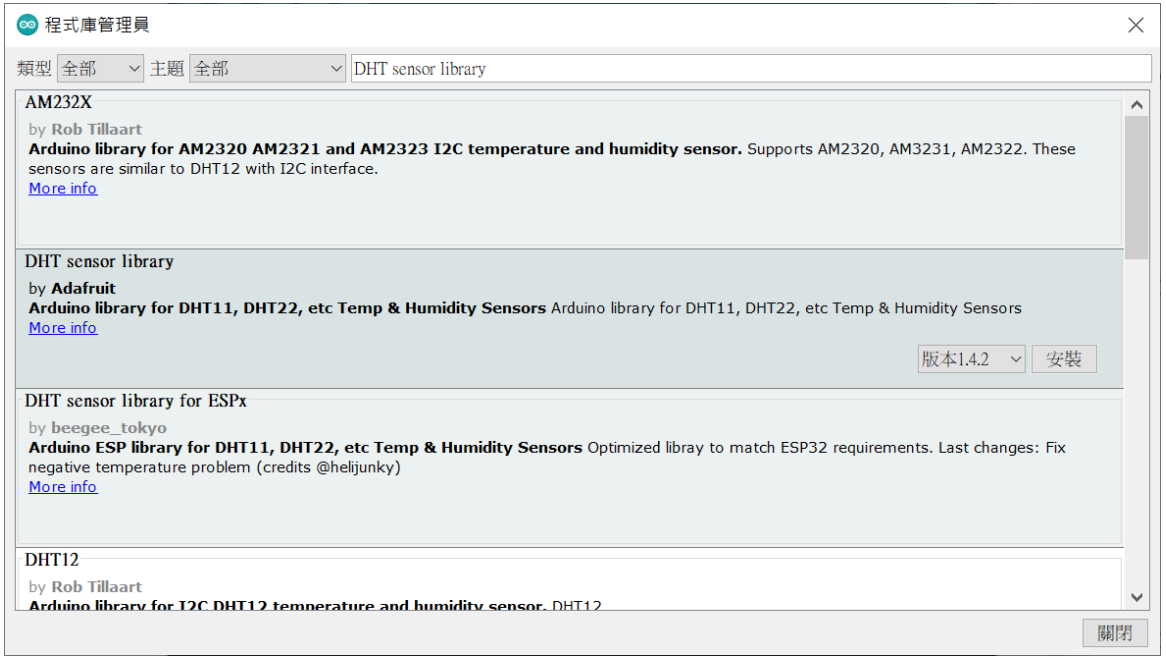


3. When DHT is sending data to MCU

- every bit of data begins with the 50us low-voltage-level
- **the length of the following high-voltage-level signal determines whether data bit is "0" or "1"**

Lab1. Library

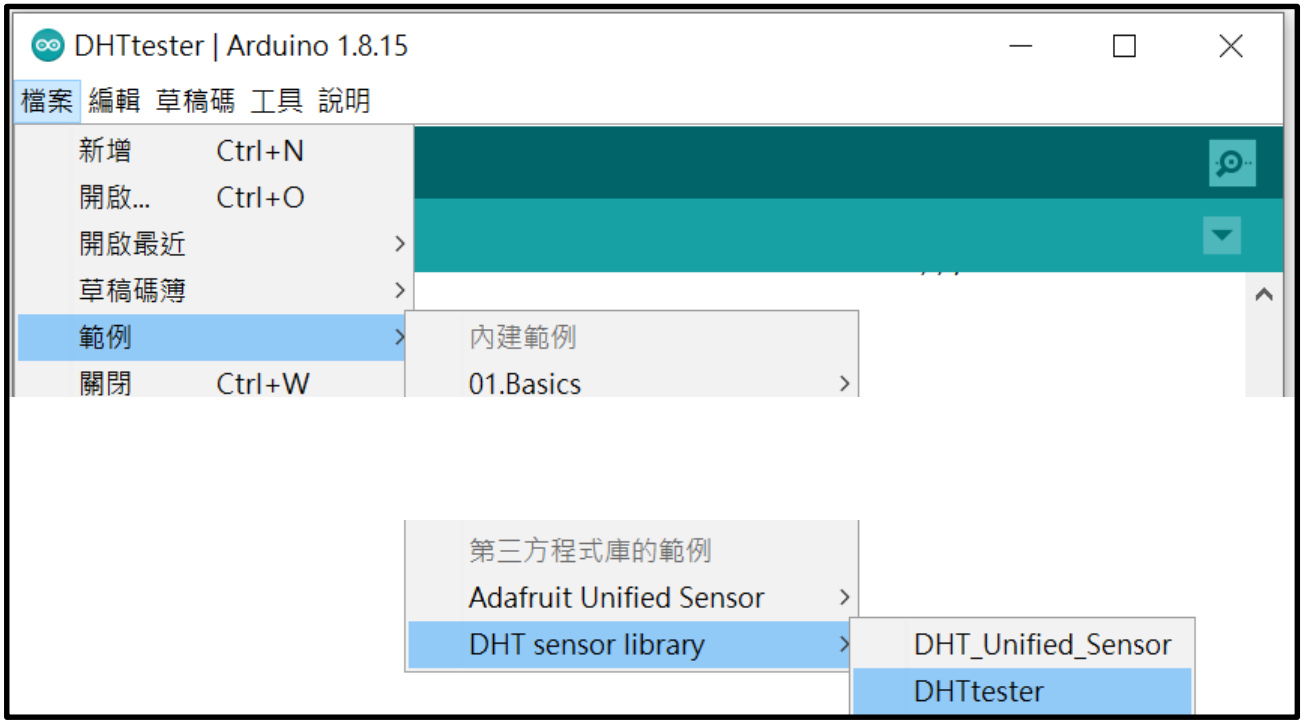
- Download and install DHT11, DHT21, DHT22 library
  - open the **Library Manager** in the Arduino IDE and install it from there
  - Library Manager = 管理函式庫
  - https://www.arduino.cc/reference/en/libraries/dht-sensor-library/



Lab1.

Arduino IDE

Open--->File--->Examples---> 第三方程式庫 --> DHT sensor library



DHTtester

```
#include "DHT.h"

#define DHTPIN 2     // Digital pin connected to the DHT sensor

// Uncomment whatever type you're using!
//#define DHTTYPE DHT11     // DHT 11
#define DHTTYPE DHT22       // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21     // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));

  dht.begin();
}
```

```
void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) ) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
```

```
// Compute heat index in Celsius (isFahreheit = false)
 float hic = dht.computeHeatIndex(t, h, false);

 Serial.print(F("Humidity: "));
 Serial.print(h);

 Serial.print(F("%  Temperature: "));
 Serial.print(t);
 Serial.print(F("°C "));

 Serial.print(F("°F  Heat index: "));
 Serial.print(hic);
 Serial.print(F("°C "));

}
```

Discussion 1

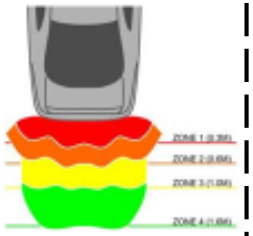
- 1. Please write down the temperature and humidity of the classroom.
- 2. Blow the DHT-11, then write down the temperature and humidity and check if there is difference.

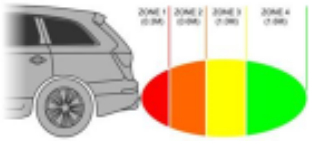
Quiz 1

- We designed a parking assist system in last week
  - **Please use the compensated sound speed to calculate distance**
    - DHT11 + HC-SR04

```
//Calculate the distance (in cm) based on the speed of sound.
distance = duration/58.2;
```

- Design a Parking Assist System (倒車雷達)
  - use ultrasonic sensor, RGB LED and speaker
  - Divide the detecting distance into three parts: a) Safe; b) Be careful; and c) Dangerous.
  - Use the speaker to reminder the driver.
    - Safe: green + no sound (> 1m)
    - Be careful: yellow + beeping(0.3 to 1m)
    - Dangerous: red + fast beeping (<0.3 m)
  - Please keep this quiz code, we will extend this in next week.





- Speed of sound
  - At 20°C (68°F), the speed is 343 m/s.
  - The approximate speed of sound (c) can be calculated from:

$$c_{air} = (331.3 + 0.606 * \theta) \text{ (m/s)}$$

where θ is the temperature in degrees Celsius (°C).

Quiz2-1

- **Dew point temperature** (露點) (反潮)
  - The temperature to which air must be cooled to become saturated with **water vapor**.
  - **Calculate the dew point** based on DHT11's report
- Simple approximation:
  - **T_{dp}: Dew point temperature**
  - T: temperature
  - RH: related humidity

$$\mathrm{T_{dp}} = T - \frac{100 - RH}{5}$$

Quiz2-2

- **Heat Index (HI) (酷熱指數)**
  - how hot it would feel
  - a human-perceived equivalent temperature
  - Calculate the **Heat Index (HI)** based on DHT11's report

$$HI = c_1 + c_2T + c_3RH + c_4T \times RH + c_5T^2 + c_6RH^2 + c_7T^2RH + c_8T \times RH^2 + c_9T^2 \times RH^2$$

c_1 = -42.379
c_2 = 2.04901523
c_3 = 10.14333127
c_4 = -0.22475541
c_5 = -6.83783×10^{-3}
c_6 = -5.481717×10^{-2}
c_7 = 1.22874×10^{-3}
c_8 = 8.5282×10^{-4}
c_9 = -1.99×10^{-6}

T: temperature
RH: related humidity

- Ref: Humidex (濕度指數) (In Canada)
  - Simple approximation :

$$Humidex = T + \frac{5}{9}(e - 10)$$

$$e = (6.122 \times 10^{\frac{7.5T}{237.7+T}} \times \frac{RH}{100})$$

Quiz2-3

- **Apparent temperature (AT, 體感溫度)** / Wind chill
  - the lowering of body temperature due to the passing-flow of lower-temperature air.
  - Calculate the **AT** based on DHT11's report
- Simple approximation
  - **AT: Apparent temperature(°C) = 1.04T + 0.2e – 0.65V – 2.7**
  - T: Temperature(°C)
  - e: water vapor pressure (hPa)
    - It can be calculated from the temperature and relative humidity
  - V: wind speed (m/s)
    - use 2.7 m/s. (the average in Hsinchu on March, 1992-2010)

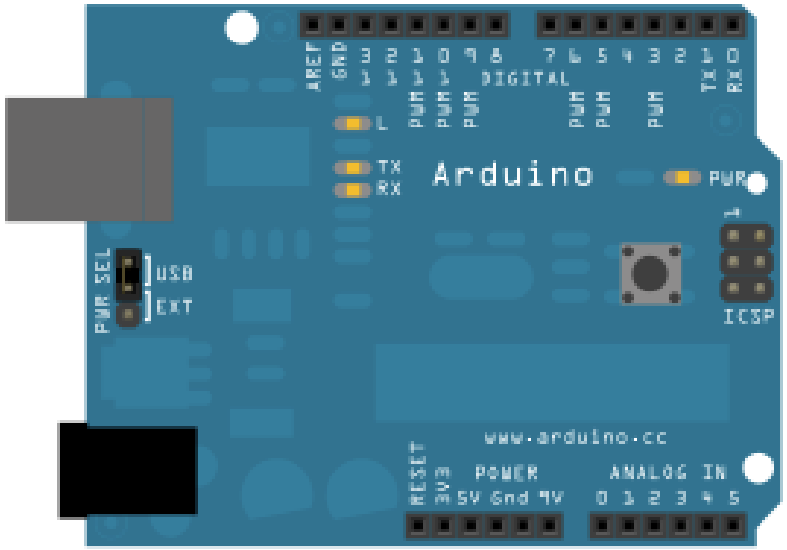
$$e = \frac{RH}{100} * 6.105 * \exp(\frac{17.27 * T}{237.7 + T})$$

Lab 2. ASCIITable

Demonstrates the advanced serial printing functions

ASCIITable

- Goal: Demonstrates the **advanced serial printing functions** by generating a table of characters and their ASCII values in decimal, hexadecimal, octal, and binary.
- Hardware Required
  - Arduino Board



ASCII Table and Description

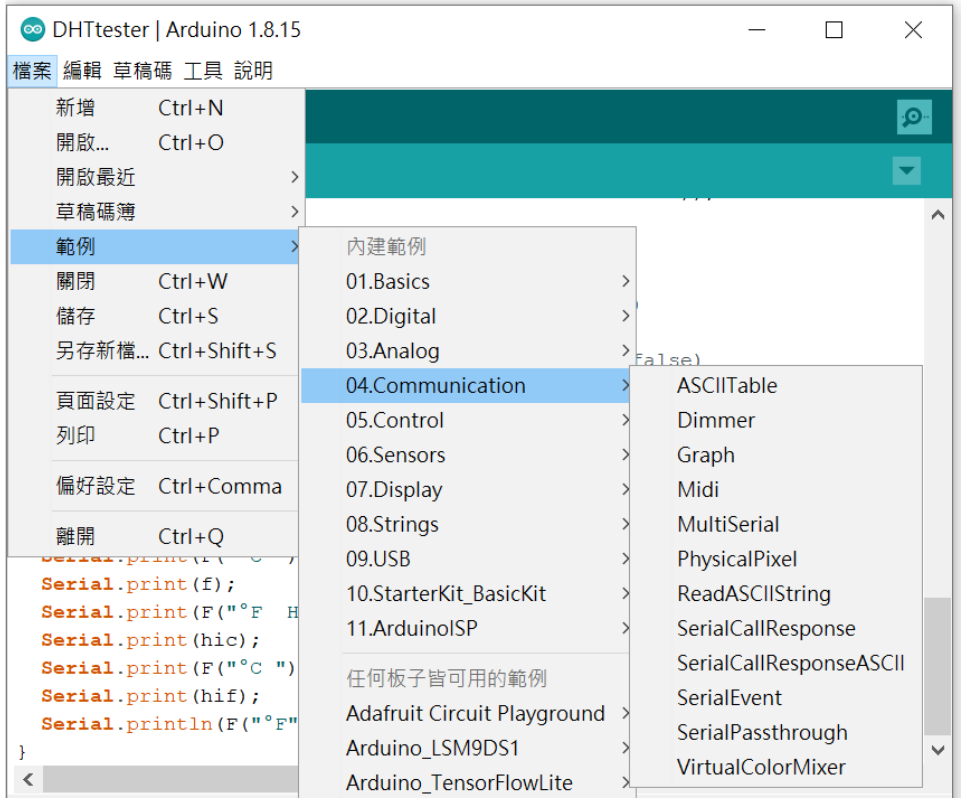
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Lab2. ASCIITable

Arduino IDE

Open--->File--->Examples---> **04. Communication--->ASCIITable**



Sample code

```
// first visible ASCIIcharacter '!' is number 33:
int thisByte = 33;
// you can also write ASCII characters in single quotes.
// for example, '!' is the same as 33, so you could also use this:
// int thisByte = '!';

void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // prints title with ending line break
  Serial.println("ASCII Table ~ Character Map");
}
```

```
void loop() {
  // prints value unaltered, i.e. the raw binary version of the byte.
  // The Serial Monitor interprets all bytes as ASCII, so 33, the first number,
  // will show up as '!'
  Serial.write(thisByte);

  Serial.print(", dec: ");
  // prints value as string as an ASCII-encoded decimal (base 10).
  // Decimal is the default format for Serial.print() and Serial.println(),
  // so no modifier is needed:
  Serial.print(thisByte);
  // But you can declare the modifier for decimal if you want to.
  // this also works if you uncomment it:

  // Serial.print(thisByte, DEC);
```

```
 Serial.print(", hex: ");
  // prints value as string in hexadecimal (base 16):
  Serial.print(thisByte, HEX);

  Serial.print(", oct: ");
  // prints value as string in octal (base 8);
  Serial.print(thisByte, OCT);

  Serial.print(", bin: ");
  // prints value as string in binary (base 2) also prints ending line break:
  Serial.println(thisByte, BIN);

  // if printed last visible character '~' or 126, stop:
  if (thisByte == 126) {    // you could also use if (thisByte == '~') {
    // This loop loops forever and does nothing
    while (true) {
      continue;
    }
  }
  // go on to the next character
  thisByte++;
}
```

Lab2. Syntax

- Syntax
  - Serial.write(val) // a value to send as a single byte
  - Serial.write(str) // a string to send as a series of bytes
  - Serial.write(buf, len) // an array to send as a series of bytes
- Description
  - Write binary data to the serial port.
  - This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.
- Returns
  - return the number of bytes written
- Example
  - Serial.write(45); // send a byte with the value 45

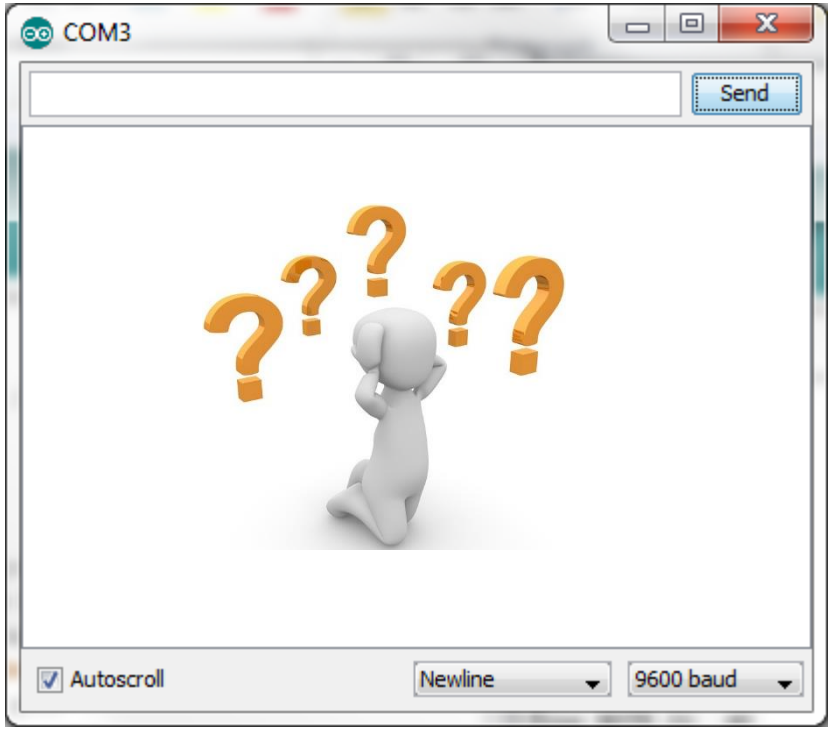
Lab2. Syntax

- Serial.print()
  - Description
    - Print data to the serial port as human-readable ASCII text.

- Serial.println()
  - Description
    - Print data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n').
    - This command takes the same forms as Serial.print().

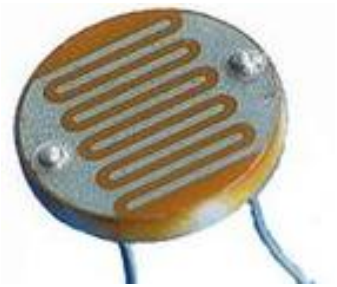
Discussion 2

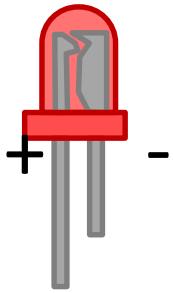
- Check the ASCII Table printed in Lab 2, how to print symbol "**?**" in Serial monitor by its ASCII value?
- Just show "**?**" in your serial monitor

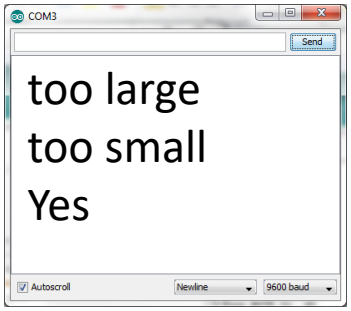


Quiz3 - Guess what

1. Design a circuit using a photocell as input.
2. When you press a button, the input of the photocell is recorded as your password. But you have no idea what it is.
3. Try to guess the password (photocell value). Input your guessing in the Serial Monitor and send it back to Arduino.
4. Arduino will send back “too large” or “too small” in the Serial Monitor until you are correct.
5. Use a LED to show if you made right guessing.







Summary

Summary

- **Practice Labs by yourself**
- **Write Answers for Discussion 1 to 3**
  - Upload to e3 before next class
- **Quiz: Write code for quiz, then demonstrate to TAs**
  - Parking assist system plus
  - Show: Dew point temperature/Heat Index/Apparent temperature
  - Guess what