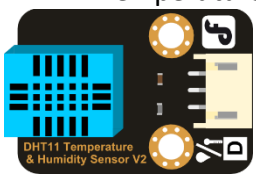
Pirate Ship – Temperature & Humidity Monitoring

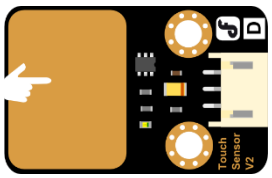
Pirate now already has basic skills of action, then we need to add new function with it – Environmental Monitoring, which let its "heart" more powerful. Temperature and humidity is closely linked with our life. We use it in a wide range of situations. DHT11 Sensor is often selected to use for temperature and humidity monitoring. Easy to use, Try it!

Hardware components

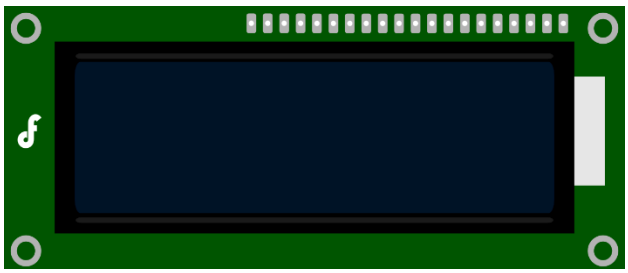
- DHT11 Temperature & Humidity Sensor x 1



- Digital Touch Sensor x 1



- I2C LCD1602 LCD Module x 1



- M3*6MM Nylon columns and tie-wraps

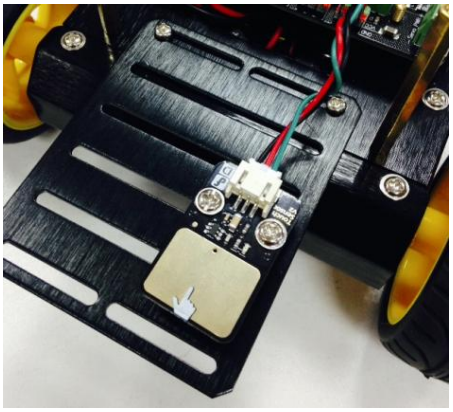
Assembly Directions

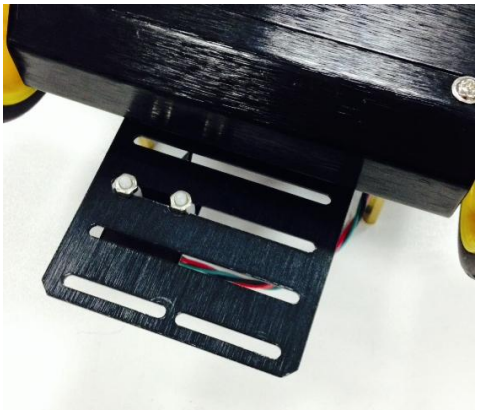
Now as we have all the components prepared, let's start assembling. It's quite easy. Just follow me and have fun.

STEP 1: Add the touch sensor to the car

There are two holes for the Nylon columns in the touch sensor. Fix the Nylon columns on the sensor. And I know some of you just cannot help twisting the column like an OCD. I totally get that. But I would still like to remind you not to over-twist those columns.

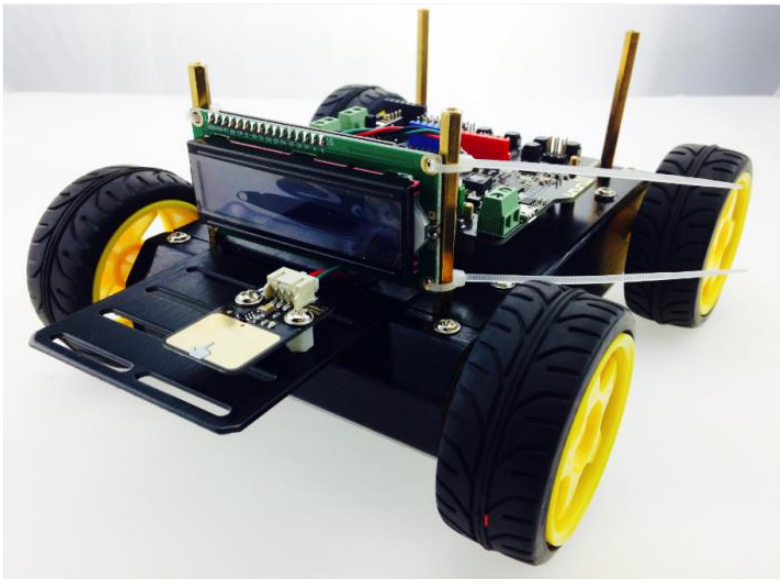
Once completed, attach the touch sensor to the sensor plate located at the fore end of the bodywork just as it is shown in the picture below.





STEP2: Add the LCD Screen

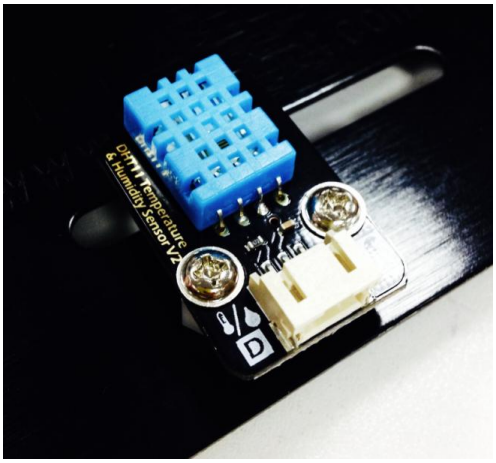
Slide the shores into the four holes in the LCD screen and fix them on the car with four tie-wraps. Then cut the remaining part of the tie-wraps.

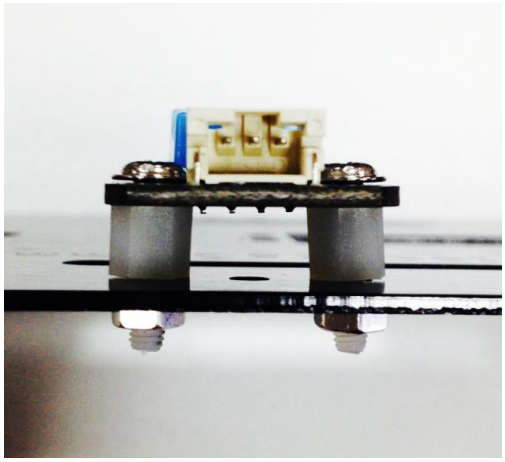


STEP3: Assemble the DHT11 Temperature & Humidity Sensor

Fix the Nylon columns on the DHT11 Temperature & Humidity Sensor just as it's shown in the picture below.

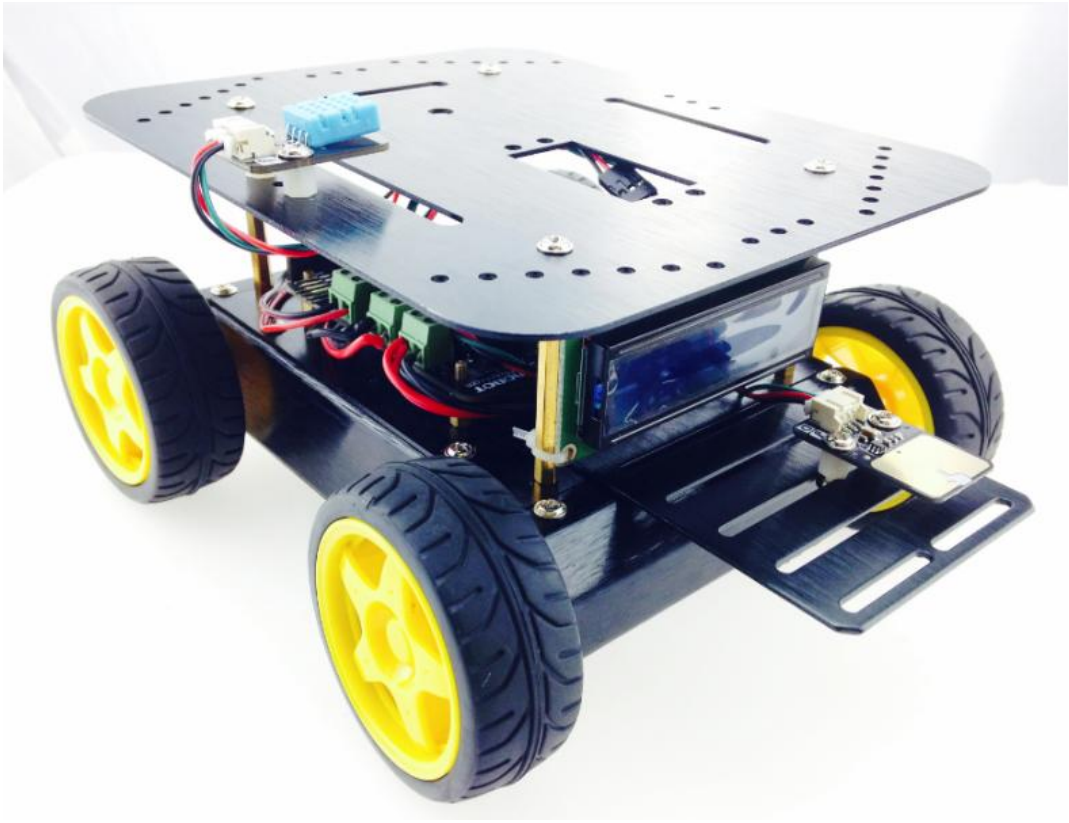
Attach the DHT11 Temperature & Humidity Sensor to the upper plate of the car by fastening the nuts for the tie-wraps.





SETP 4: What It Looks Like When Finished Assembling

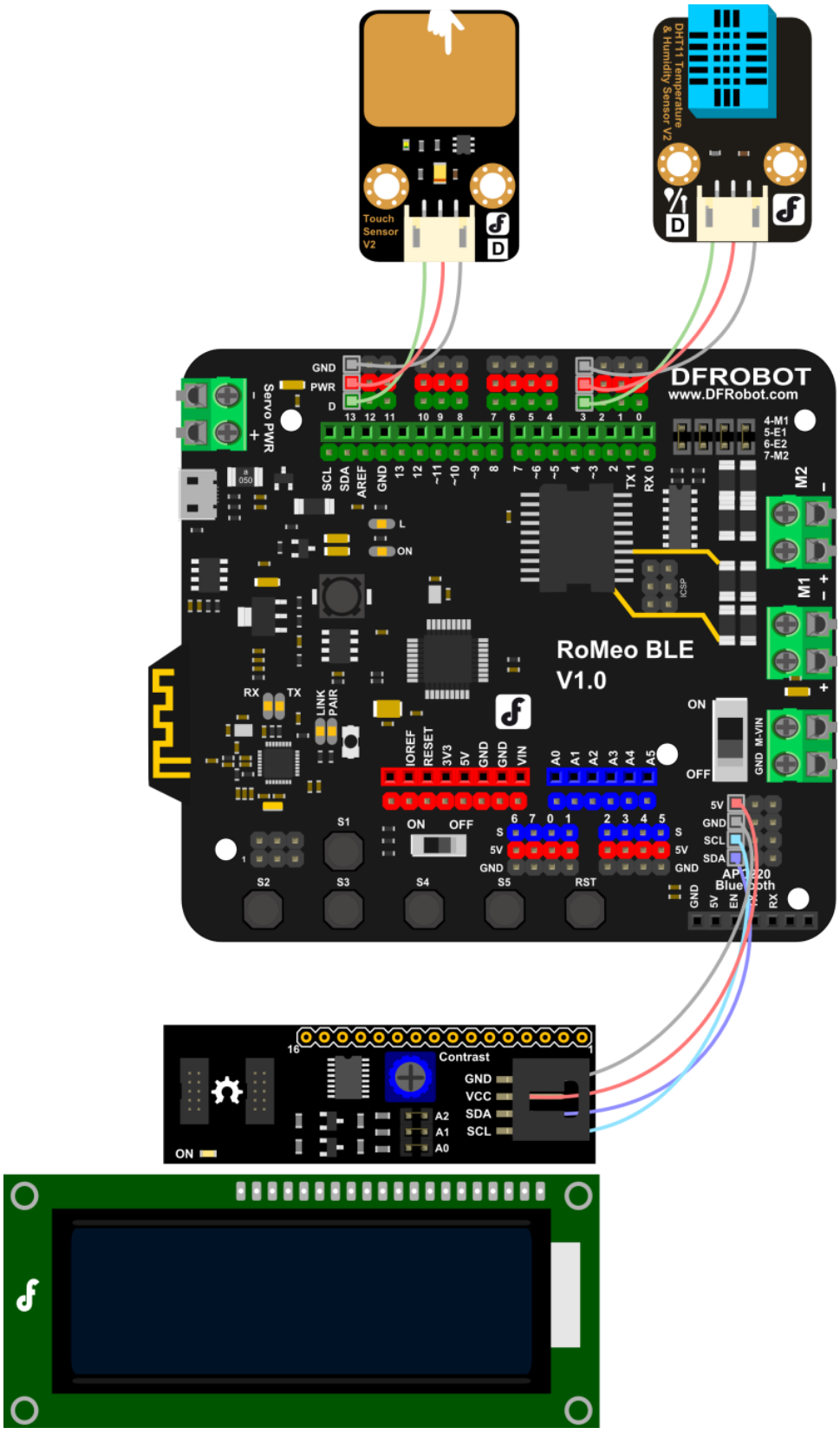
But now it's still too early for fixing the upper plate onto the car as we need to work on the electric circuit connection.



Connect the hardware

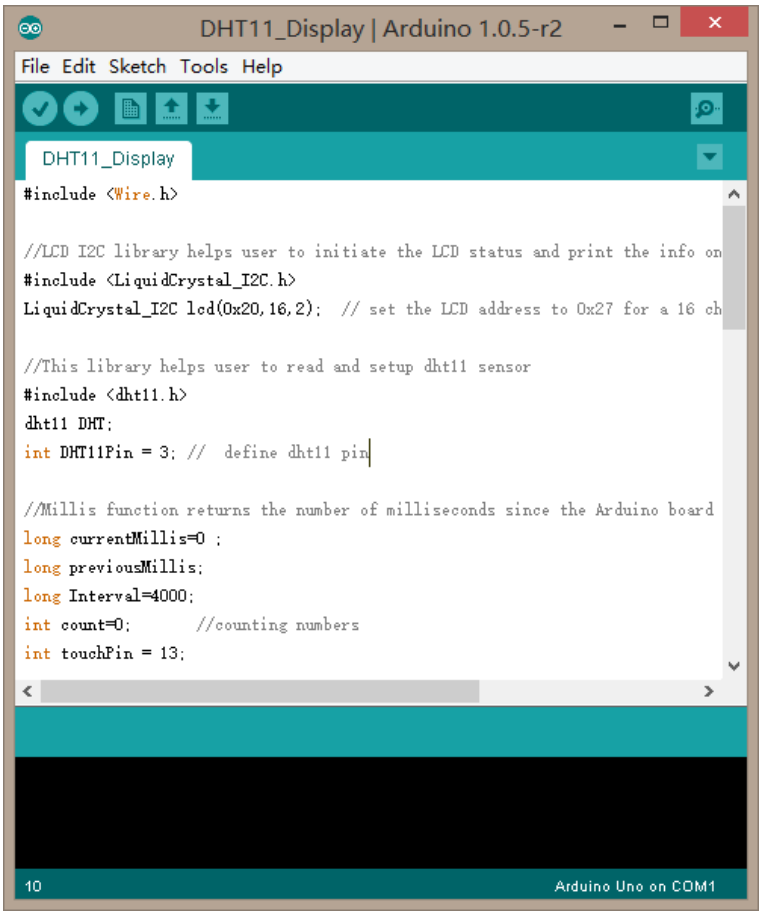
Now it's time for electric circus connection. All you have to do is to connect the 3pins on the sensor to the Romeo BLE.

Keeping cables in order is quite important. The green cable represents the signal one; the red one stands for VVC and the black one represents GND. The LCD monitor shall be connected to VCC, GND, SCL and SDA in that particular order.


Touch Sensor V2
DHT11 Temperature & Humidity Sensor V2
DFROBOT
www.DFRobot.com
RoMeo BLE V1.0
Servo PWR
4-M1
5-E1
6-E2
7-M2
M1
M2
M-VIN
ON
OFF
S1
S2
S3
S4
S5
RST
Contrast
GND
VCC
SDA
SCL
A2
A1
A0

Coding

Download the code first. The following picture only shows a part of the coding. You can find codes named *DHT11_Display.ino* and then download. And don't forget the library for *LiquidCrystal_I2C* and *dht11*.



If everything goes smoothly and you've successfully downloaded the coding, real-time temperature and humidity will be shown on the LCD screen. Hereby the touch sensor functions in two ways.

First, it functions switch of the LCD backlight lamp. If you don't touch the sensor for a long period, it will turn off itself automatically, which is energy-conservative.

Secondly, when you have several environment sensors, touch switch can help you quickly cut among different functions where you may see different environment monitoring data.

Code Synopsis

We need to call library at the beginning of the code. Library is quite important and software engineers have already written some functions to process complicated data in the library. It's unnecessary for you to understand such abstruse theory as long as you know how to use it. Thus, it won't be easy to understand code without library.

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x20,16,2);

#include <dht11.h>
dht11 DHT;
```

Here you need to know about *DHT11Pin*, a variable that is used to declare the sensor's pins.

```
int DHT11Pin = 3;
```

Namely, DHT11Pin represents Digital Pin3. That is to say, our DHT11 sensor is connected to Digital Pin3.

What if you want to control the sensor with the Digital Pin4, how would you modify the codes?

```
int DHT11Pin = 4;
```

It's quite easy. You just need to replace 3 with 4.

The followings are some declarations for the time variables. TouchPin represents touch sensor while 13 stands for digital pin.

```
long currentMillis=0;
long previousMillis;
long Interval=4000;
int count=0;      //counting numbers
int touchPin = 13;
```

Bring in the function of setup(), which is a setting-up for the initation.

```
pinMode(touchPin,INPUT);
```

Then keep the touch sensor with a type-in mode. For specific information, you may check the Arduino Reference in Arduino Website(www.arduino.cc), which has introductof the function of pinMode().

Next, you need to initialize the LCD screen and turn on the LCD light, which show that LCD screen is ready.

```
lcd.init();
lcd.backlight();
delay(100);
lcd.setBacklight(0);
```

Now it's turn for the function of loop(). First we need to read the value from the touch sensor and then store those data with one variable touchState.

```
int touchState = digitalRead(touchPin);
```

Then check if the controller will receive a signal of HIGH once you touch the touch sensor with you fingers, 1 shall be added to the *count*.

```
if (touchState == HIGH){
    count++;
    previousMillis= millis();
}
```

Hereby *count* means how many times you have touched the screen. But if you only touch the sensor one time, then the amount of time for each touch will be included in the function of millis().

We change the length of touch time with a sub sentence initiating with if. Interval here means the period for touch we set up. Thus, we know what action shall be taken within four-second of touch and more than four seconds' touch respectively.

```
if(currentMillis - previousMillis < Interval) {
     //do something in 4 second
else{
     //do something more than 4 seconds
}
```

What action shall be taken when we touch the sensor for more than four seconds

```
lcd.setBacklight(0);
```

The function of setBacklight() is used to turn off the LCD backlight lamp.

If we touch the sensor for more than four seconds, we know that the LCD backlight lamp can be turned off.

what action shall be taken within four-second of touch.

```
if (count==1){
   // One touch, the LCD screen won't show any difference
}
else if (count==2){
   // Touch twice, value will be shown on the LCD screen
}
```

Press the touch sensor one more time within four seconds; the screen would still be off. Only if you touches it twice at the same time, will the LCD backlight be on and figures of temperature and humidity be shown.

Please remember to keep the count as zero after you touch the sensor for the last time.

```
count=0;
```

Thus the complete code shall be:

```
if (count==1){
   lcd.setBacklight(0);
}
else if (count==2){
   lcd.backlight();
   DHT11Show();
   count=0;
}
```

Then we need to keep a track of the current time as we can compare it with *previousMillis.* This point is very important.

```
 currentMillis = millis();
```

Once more function involved: DHT11Show(). It's not hard to guess that it's used to show the figure for the DHT11 sensor.

```
int value = DHT.read(DHT11Pin);
```

The function of DHT.read(pin) is used to read data read from the pins of DHT11 sensor.
Here is how we would used the function related to LCD screen.

```
lcd.setCursor(0,0);
lcd.setCursor(0,1);
```

The function of setCursor(column,row) is used to demonstrate which column and row the cursor is shown, starting from zero within the brackets.

```
lcd.print(DHT.humidity,1);   // It shows humidity
lcd.print(DHT.temperature,1);   // It shows temperature
```

print() means this figure can be shown on the screen directly.

```
lcd.print("      ");
```

lcd.print(" ") means blank space shown on the screen. It's used to clear the screen.

A Combination of Multiple Sensors

How can you combine multiple environmental sensors once you have bought some kind of sensors?

Don't worry. We will offer you guys a coding template for the testing of multiple sensors. You can make adjustments of the combination by referring to the mentioned template. In fact, the theory is the same as single sensor except that there are for steps for the changes of LCD screen.

The coding in red below needs to be modified. We mentioned before that count refers to how many times fingers touch the sensor. Thus, count=2 means that we have pressed twice and it shows the figures for the first sensor. Keep going! Please bear in your mind that you shall keep the count zero again.

Sample Code:

```
if(currentMillis - previousMillis < Interval) {
    if (count==1){
      lcd.setBacklight(0);
    }
    else if (count==2){     //No.1 Sensor
      Sensor1Show();
      lcd.backlight();
      else if(count==3){      //No.2 Sensor
        Sensor2Show();
        lcd.backlight();
        count = 0;
      }
  }
```

Of course, initiation set-up, declaration of variables at the beginning, for the sensor is important.

You can check the sample code named *WeatherStation.ino* for reference if you still have no idea how to modify your codes.