Report

11 June – Monday – Day 1

- Briefing about the overall of the project.
- Introducing to all of the people involving in the project.
- Introducing to the Arduino board.

Arduino UNO Board

SainSmart MEGA Board 2560

- Learn how to solder pieces of wire together.

12 June – Tuesday – Day 2

- Arduino UNO & Arduino MEGA can work with other device.
    - Load Cell – Weighting

SainSmart MEGA Board 2560

        - Connection: needs to use 4 wire and connect it to ADC/Amplifier before connect it to the board.
            - Red – Power 5 V
            - Black – GND
            - Green – Digital output
            - White – System clock

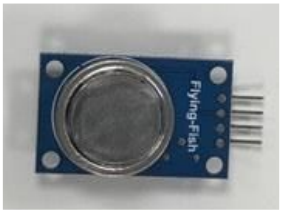
        - This device need additional library “HX711.h” in order to function properly.

    - Ultrasonic – Distance measurement

HC SR04

- Connection: needs 4 wire and doesn't need the ADC/Amplifier.
    - VCC – 5V Power
    - Trig – This pin control the device to send signal, HIGH for sending, and LOW for idle.
    - Echo – By using *pulseIn (echoPin, HIGH)* this pin will determine the duration of the sent signal. And we will use that value to calculate the distance by this equation *distance= duration*0.034/2.*

- Smoke Detection



Flying Fish MQ 135

- Connection: needs 4 wire to connect to a board and doesn't need ADC/Amplifier.
    - VCC – 5V Power
    - GND - Ground
    - DO – Digital Output, this pin either give 1, if there's smoke, or 0, if there's no smoke.
    - AO – Analog Output, this pin output the amount of smoke in the air.

- They can also be connected with multiple device through breadboard.

13 June – Wednesday – Day 3

- A7 Ai Thinker can be used to send SMS to another mobile number but not very reliable.
    - Connection: need to use 4 wires to connect from A7 to Arduino MEGA 2560 board.

A7 Ai Thinker

- URXD – TX3 of MEGA board
- UTXD – RX3 of MEGA board

- 5V – PWR ***we connect the A7 to power socket directly.**
- GND – GND of MEGA board

- The code is as follow

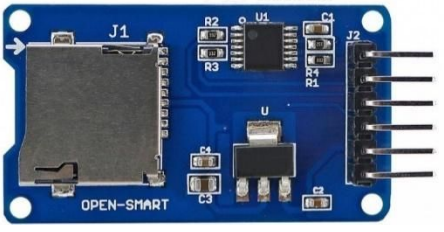
```
Serial.begin(115200);
Serial3.begin(115200);
Serial3.println("AT+CMGF=1");
delay(2000);
Serial3.print("AT+CMGS=\"");
Serial3.print(phone_no);
Serial3.write(0x22);
Serial3.write(0x0D);
Serial3.write(0x0A);
delay(2000);
Serial3.print("Testing A7 SMS");
delay(500);
Serial3.println(char(26));
delay(2000);
```

- Start with AT command to send SMS to a given phone number.
- If the SMS needs to be sent repeatedly, it might cause the board to not sync correctly, and cause the texts to be error.
  - Solved by only begin () the serial when needed to use, and end () when finished transmitting SMS.

- It can also send data collected by other devices connected to the board.
- LoRa32u4 can connect with Serial connection and other devices and communicate reliably.

14 June – Thursday – Day 4

- A7 Ai-Thinker's code is changed and optimized to send more reliable SMS.
- Arduino MEGA can now store data in micro SD card and send it via SMS from that same card at later time.
  - Connect MEGA board with the ultrasonic device and GSM as shown earlier.
  - Connect it with micro SD card reader.



Micro SD Card Catalex – Top

Micro SD Card Catalex - Bottom

- Connection: need 5 wires to connect to SainSmart MEGA
  - CS – Pin 53
  - SCK – Pin 52
  - MOSI – Pin 51
  - MISO – Pin 50
  - VCC – 5V
  - GND – GND of MEGA board

SPI	Uno	Mega
SS	10	53
MOSI	11	51
MISO	12	50
SCK	13	52

- In case of connecting to UNO board, the wire need to be like this

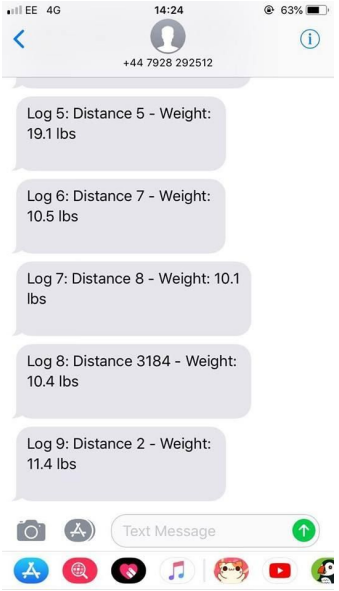
```
file = SD.open(fileNameRead, FILE_READ);
while(file.available()){
  char nChar = file.read();
  if (nChar == '\n') totalLog += '\n';
  else totalLog += nChar;
}
file.close();
Serial.println(totalLog);
```

- This code will take all data from the particular file in the micro SD card and store it in the string to be sent and/or display on the monitor later.

- LoRa32u4 doesn't work like yesterday, it failed to upload to the board.
  - The COM Port keep changing when we tried to upload new code to the board
  - It needs to be manually enter bootloader by double press the RST button after we hit the upload button.
  - New solution can work, but needs a lot of time

15 June – Friday – Day 5

- When setting pin for LoRa32u4, the setPins() the parameter needs to be 8, 9, 2, rather than 1, 9, 2 because the default setPins for LoRa is 1, 9, 2. But we're using other board that use 8, 9, 2 configuration. Now it can run and initialize.
- With 2 LoRa, it can now send data between 2 devices.
- Using A7 with MEGA board and it can continuously send SMS containing data read from other device for hours every 10 minutes without error.

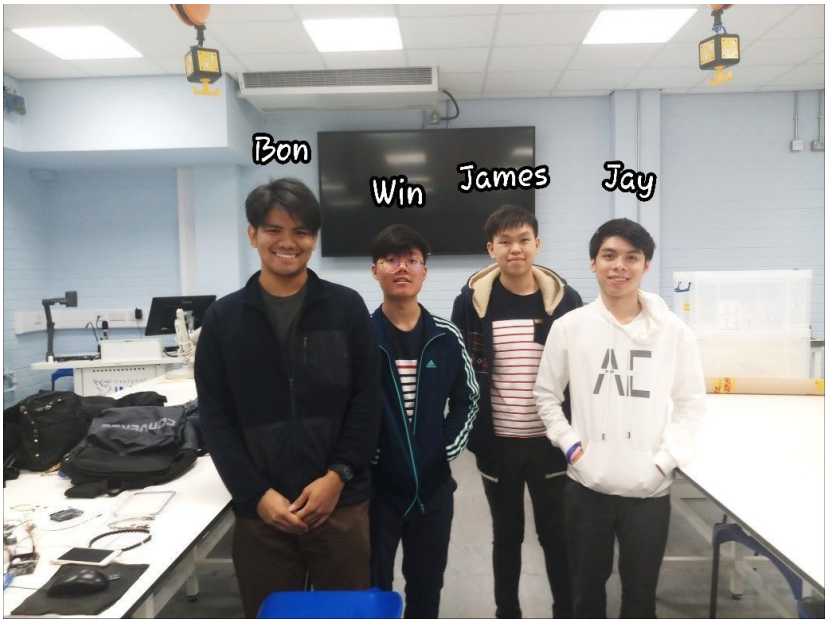


Received SMS

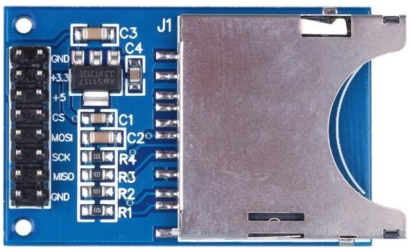
- We have a beer with David and his friends, Chris and Adam, at the Ship & Mitre. Next, we go to have a Thai food with David and talk about how things are in Thailand and UK. Then David take us for a tour around Liverpool and give us a history talk about the city.

18 June – Monday – Day 6

- When using the SD card to read or write to the card, the name of the file can't be longer than **"DISLOG10.txt"**, or the card won't create a new file.



- Hazel helps us took a picture.
- SD Card Slot Socket Reader Module



SD Card Slot Socket Reader Module

- Use the same connections similar to Micro SD card catalex with 5 V of power, but doesn't work. http://henrysbench.capnfatz.com/henrys-bench/arduino-output-devices/arduino-lc-studio-sd-card-tutorial/
- The result is the SD card is messed up and the content inside become corrupted.

- I2C soil moisture sensor
  - David will get us another sensor to implement into the whole system.
  - https://tindie.com/products/miceuz/i2c-soil-moisture-sensor/
- E32 TTL 100

E32-TTL-100

- Connection: needs 7 wire connections between this device and UNO board. According to this website https://github.com/Bob0505/E32-TTL-100

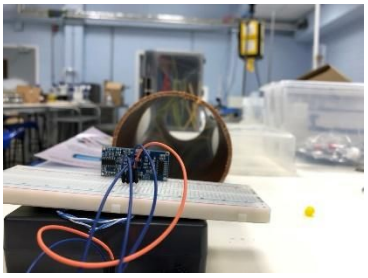
UNO/NANO(5V mode)		E32-TTL-100
D7	<---------->	M0
D8	<---------->	M1
A0	<---------->	AUX
D10(Rx)	<---> 4.7k Ohm <--->	Tx
D11(Tx)	<---> 4.7k Ohm <--->	Rx

- With 5V of power.
- We only have 2 antenna, so we can't test the sending side and receiving side at the same time.

19 June – Tuesday – Day 7

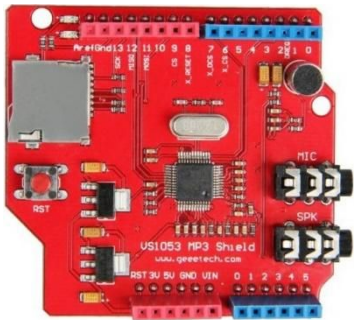
- We do the test of how accurate the Ultrasonic sensor can be by making a mark on a table with pencil on centimeters scale.

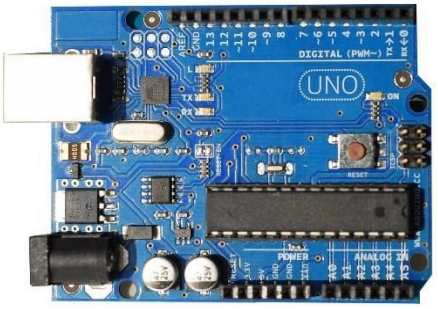
.



The testing

  - We try to use the ultrasonic sensor to measure distance through a tube and experience a problem that the object outside the tube can interfere with the sensor. To fix that, we lift the sensor higher and the problem is solved.
  - We try to push object inside the tube, and it works as intended.
  - Need to test the sensor with a longer tube in the future to make sure that it works as well.
- VS1053 MP3 Shield





VS1053 MP3 Shield *UNO Board*

  - This shield can play an audio and record audio to/from micro SD card with the socket that it has.
  - It will stay on top of the UNO board and doesn't require any additional cable.
  - The code for testing the shield is from https://elecrow.com/wiki/index.php?title=VS1053_MP3_Shield

```
#include <SPI.h>
#include <SdFat.h>
#include <SdFatUtil.h>
#include <SFEMP3Shield.h>
```

```
if(!sd.begin(SD_SEL, SPI_FULL_SPEED)) sd.initErrorHalt();
if(!sd.chdir("/")) sd.errorHalt("sd.chdir");
```

Header *Initialize SD Card*

  - All mp3 file in the SD card need to be named as "trackXXX", XXX being the number of the song. For example, "track001" or "track002".

20 June – Wednesday – Day 8

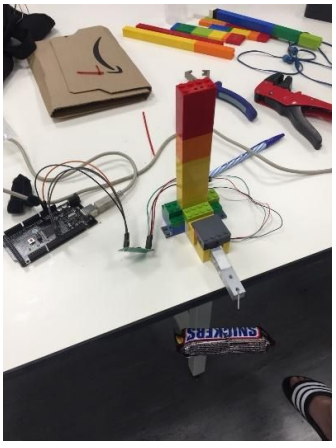
- Using LEGO to create a hand that will hold a LoadCell that attached weighting tube inside a glass with some water in it.
    - The weighting tube will be slightly above the bottom of the glass.
    - The current design can work, but may need to redesign in the future.
    - The LoadCell can't reliably output the weight.
    - Need to get another LoadCell in working condition.
- ET 016 General purpose Temperature Probe

ET 016 General purpose Temperature Probe

    - Connect the temperature dependent resistor to a Digital Multimeter and turn it to resistor mode to observe the change in resistance when temperature change on the metal end of the line.
    - The decrement of temperature cause the resistance to decrease when I put my hand around the metal head.
    - **Need another resistor** in order to read the change in resistance of the sensor when trying to read the temperature in the environment.

21 June – Thursday – Day 9

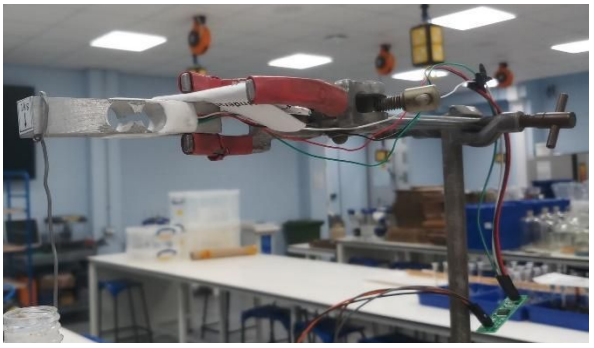
- Fix LoadCell by finding a LoadCell that can work and solder it with the proper wire to make the connection with the board.
- Continue redesign the design to test how to make the LoadCell accurate with LEGO.



LEGO set to hold LoadCell in place

Object with known weight (50 grams) – Snickers, consists of a nougat mixed with caramel and peanuts.

- To make the LoadCell more accurate, we need to put the object on the LoadCell and calibrate it. To do this, we need to use the function `scale.set_scale()` by put the new input inside the function. If the weight output is above the object's weight, we need to decrease the calibration factor until the weight output from the LoadCell is correct, and vice versa. In this case, the calibrator factor for our LoadCell with 50 grams is -2350.



LoadCell is firmly held into place

- Put LoadCell into spot that is firmly held and can withstand some magnitude of weights.
- Put glass and pipe underneath the LoadCell and take a small coil to tie the LoadCell and pipe together, so that the pipe will hover just above the bottom of the glass.
- Adjust the measurement tape to match the floor of the glass.
- After everything is in position fill in the water for 100 ml and measure the weight and height of water.
- Keep repeating this until the water is full.

Water (ml)	Weight (g)	Height (cm)
0	65.5	0
100	53.85	3
200	43.87	11.9
300	33.78	18.1
400	24.17	24.1
500	14.4	30

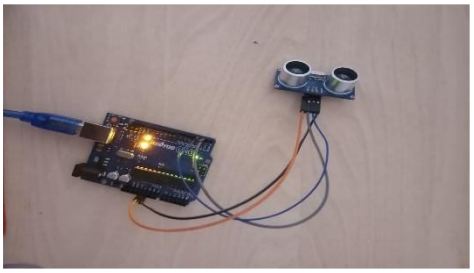
Result

- After we get all the result, we need to calculate an equation to translate weight into height.
  - We get this following equation.
  - y = -0.58*x + 38.45
- More information will be in Excel named waterEquation.

22 June – Friday – Day 10

- According to David Lamb's advice, he said about the next steps of our project that was related to the transmission and storage of data on server so we try to store data that we get from distance sensor (HC SR04) on database called MySQL and finally, we come up with accomplishment.

HC SR04



Connection between Arduino UNO and HC SR04

  - At first, we upload the code of distance measurement into the Arduino UNO.
  - Then, we access Xampp server to store our data. We set up the MySQL by creating new database, selecting database, and creating the table for our data on command line.

Xampp

```
Setting environment for using XAMPP for Windows.
USER@DESKTOP-RE4F4PA c:\xampp
# cd c:\xampp\mysql\bin

USER@DESKTOP-RE4F4PA c:\xampp\mysql\bin
# mysql.exe -u LJMU --password
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 59
Server version: 10.1.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Accessing MySQL on command line

  - After we have finished setting up MySQL, we use Processing 2.2.1 IDE as the intermediary between the Arduino and MySQL. We import BezierSQL library and use normal SQL language to program the Processing IDE.

```
if ( msql.connect() )
  {
      msql.query( "insert into dataDistance(CurrentTime,Distance)values(\""+currentTime+"\",\""+serial+"\")" );
      count++;
      println("id: "+ count);
  }
```

Example of SQL code

- We set the sensor to measure and collect the data for every minute (delay(6000)).
- Finally, when we compile and execute the program, we get the expected result.

sno	CurrentTime	Distance
1	2:1:45	113.83 cm
2	2:1:47	114.81 cm
3	2:1:48	112.70 cm
4	2:1:49	97.34 cm
5	2:1:50	104.98 cm
6	2:1:51	5.96 cm
7	2:1:52	9.45 cm
8	2:1:53	19.11 cm
9	2:1:54	20.82 cm
10	2:1:55	31.20 cm

Result from command line

sno	CurrentTime	Distance
1	2:1:45	113.83 cm
2	2:1:47	114.81 cm
3	2:1:48	112.70 cm
4	2:1:49	97.34 cm
5	2:1:50	104.98 cm
6	2:1:51	5.96 cm
7	2:1:52	9.45 cm
8	2:1:53	19.11 cm
9	2:1:54	20.82 cm
10	2:1:55	31.20 cm

Data on our database

- Continue to work with the value mapping for the weight to height measurement by pouring water 25 ml water for each iteration to make it more precise.

25 - June - Monday - Day 11

- We talk with David via Skype about our progress and the problems that we face with.
- David Jordan suggests us to use a more formula way to turn weight into height method, so we need to come up with a new equation

$$Fb = (Weight\ Air) - (Weight\ Water)$$

  - With this equation, we can get the buoyant force by subtracting the weight together.

$$Fb = \rho V g$$

  - With this equation, we can find the volume of the sunken part of the hanging tube in the pipe, since we already know the buoyant force.

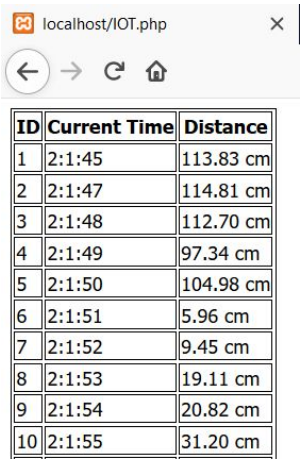
$$V = \pi r^2 H$$

  - With this equation, we can find the height of the sunken part of the hanging tube, since we already know the value of volume.
  - After we get the height of the sunken tube, we turn the value into cm and use that for further development.
- We also optimize the code by grouping code as a function, adding more comment, and adjusting the delay time.
- We also write the document and make the package for our system.

- Following the suggestion from David Jordan, instead of read a single value and send it, we use

26 - June - Tuesday - Day 12

- We have done the server side script to pull the data from MySQL and display it on website by PHP and SQL language.



ID	Current Time	Distance
1	2:1:45	113.83 cm
2	2:1:47	114.81 cm
3	2:1:48	112.70 cm
4	2:1:49	97.34 cm
5	2:1:50	104.98 cm
6	2:1:51	5.96 cm
7	2:1:52	9.45 cm
8	2:1:53	19.11 cm
9	2:1:54	20.82 cm
10	2:1:55	31.20 cm

Output on website

```
6     $conn = new mysqli($servername,$username,$password,$database);
7
8     if (!$conn){
9         die("Could not connect: ".$conn->connect_error);
10    }
11
12    $sql = "SELECT * from dataDistance";
```

Example of PHP code

- We also encode the data that we get from MySQL to JSON object. We think that this is very useful to apply with our project


```
"Distance": [
    {
        "0": "1",
        "sno": "1",
        "1": "2:1:45",
        "CurrentTime": "2:1:45",
        "2": "113.83 cm\r\n",
        "Distance": "113.83 cm\r\n"
    },
    {
        "0": "2",
        "sno": "2",
        "1": "2:1:47",
        "CurrentTime": "2:1:47",
        "2": "114.81 cm\r\n",
        "Distance": "114.81 cm\r\n"
    },
    {
        "0": "3",
        "sno": "3",
        "1": "2:1:48",
        "CurrentTime": "2:1:48",
        "2": "112.70 cm\r\n",
        "Distance": "112.70 cm\r\n"
    },
```


JSON object

- We create blog by using blogger.com and we try to use the domain of cPanel that David gave its information to us via email but we face with the problem about DNS setting so we don't go on with this so far.

27 - June - Wednesday- Day 13

- We skype with David again in the topic of turning width of water to height method, new distance sensor, and also the problems that we face with such as about the blog.
    - David recommends us to try to continue with LoRa.
    - David says that he will buy the new domain for us.

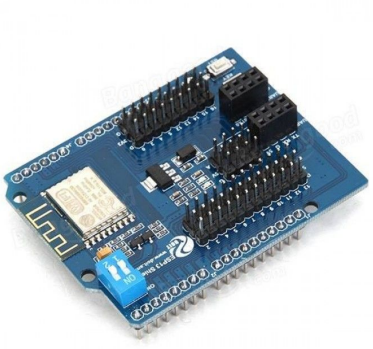
- LoRa32u4 seems more reliable than the past but we still can't use it with A7 Ai-Thinker to send SMS to mobile phone.
- We try to apply the new turning width of water to height method which David advices us.
- We try to turn weight into height from David's method.

Height	CalHeight	Deviation
3.8	5.09	-1.29
7.6	8.9	-1.3
11.5	12.5	-1
16.2	16.9	-0.7
25.5	25.4	0.1

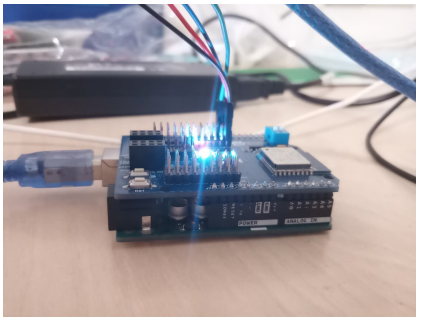
    - This is the result. It experiences high deviation at the beginning as we expected.
    - In order to do this, we need to find the ratio of the gram changed per centimeter of the tube. We find this by calculate the cross section of the tube. It has 0.7798 centimeters radius, so the area is $2\pi r^2 = 1.9094$. Once we know the area, we can find the volume of displaced water for every cm of the tube. For every 1.9094 grams changed in weight, that means the height of water has risen for a centimeter.

28 - June - Thursday - Day 14

- This is the first time we try to use ESP13 WiFi Shield



ESP8266 Serial WiFi Shield
(ESP13 Shied)

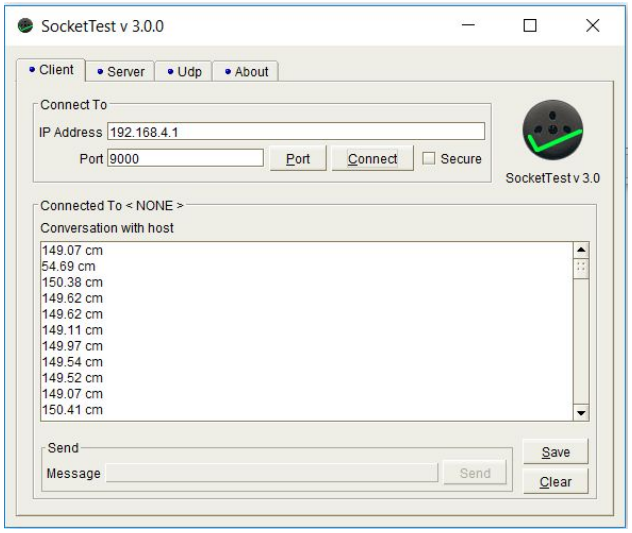


Arduino Uno with ESP13 WiFi shield

    - We connect ESP13 WiFi shield with Arduino Uno and we choose ultrasonic sensor (HC SR04) to test the wireless connection. We follow the steps that

http://wishidknownthat.blogspot.com/2016/12/getting-esp-13-arduino-wifi-shield-to.html stated.

- At first, we use Python program to receive the data but it doesn't work well as our expectation because of delay time. Therefore, we find another way that is better than the old one. Finally, we use SocketTest v 3.0.0 instead of the Python one and it seems to work very well.

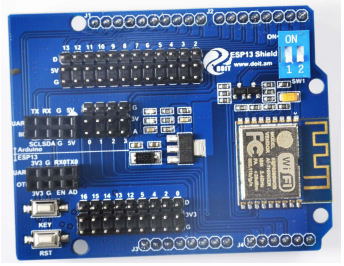


Displaying the data through SocketTest

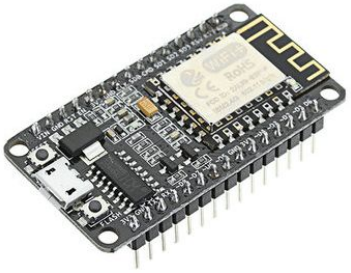
- After we have tried the wireless connection of WiFi shield, we learn more about how can SMSs on Android phone are sent and kept them on MySQL database. Currently, we just found only 1 solution to do this which is to use SMS Backup & Restore application to backup the file as XML file into SD card and use php language to store that file on MySQL manually so we think it is not a good idea to do that. Therefore, we will find and try the new way later.

29 - June - Friday - Day 15

- We learnt more about ESP13 Wifi shield and ESP8266 Module. We find just a few relating websites, then we try to apply it and its library with Arduino IDE but it is more complicated than we think because knowledge of network is required.
- Esp 13 Wifi



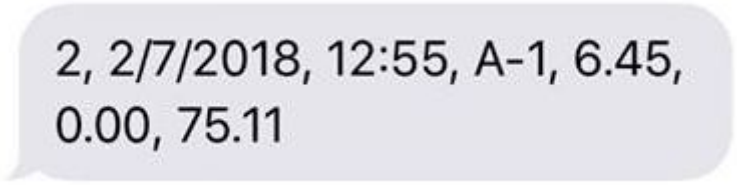
    - This UNO Shield can connect to UNO board and act like a Wifi to send and receive data
    - “SerialESP8266wifi” is the library that use for ESP 13 Wifi (src: https://github.com/ekstrand/ESP8266wifi)
- ESP8266



    - This ESP8266 board can operate on its own, given the correct driver for the board.
    - http://instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/ According to this link, it instruct us to install the correct driver and choose the correct one for the board.
    - This board can now operate like a normal board, but we still can’t get it to send data over Wifi.

2 - July - Monday - Day 16

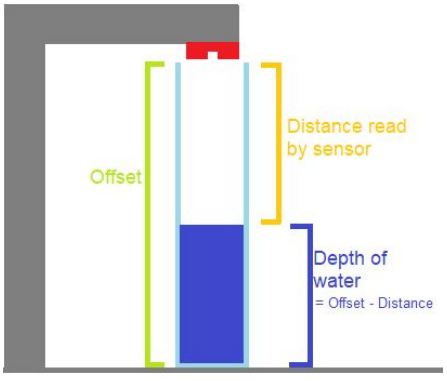
- We began writing blog in blogger.com. We have done from day 1 to day 9. "http://measurementwithiotdevices.blogspot.com/" is the temporary link of our blog.
- Change the format of the data we send in the SMS. Each section of the data being sent will be separated by comma to make it easier when use with webpage. It's now look like this.



  -
  - 1, 2/7/2018, 14:36, A-1, (data), (data), data)
  - The first value is the number of log it sent, start at 1.
  - The second value is the date when the data is observe.
  - The third value is the time.
  - The fourth is Code, it acts like a nickname for coordinator. The webpage will have more information to use it with google map API
  - The rest will be data read from sensor.
- Add the wake up and sleep cycle from David's email into the code in version 2.
  - This method will put the device into a low power mode, sleep, to save power between read, which can be 12 hours or a day apart from another. And back to normal state when it needs to read and send data.
  - This will also help the device to continue to work after a power loss happen by putting a code that will boot up after a power loss.

3 - July - Tuesday - Day 17

- We continue writing blog. Today, we have done from day 10 to day 14.
- Add the ability for the user to change or modify some data on the go.
  - 'r' - This button will set the variable 'weightAir' to the current value currently read by the LoadCell.
  - 'oXXX' - By typing 'o' and followed by number will set the offset value for the program. The offset is the number for the ultrasonic sensor to change distance to water surface to depth of water. For example, by typing 'o30', the program will set 30 cm as its floor.



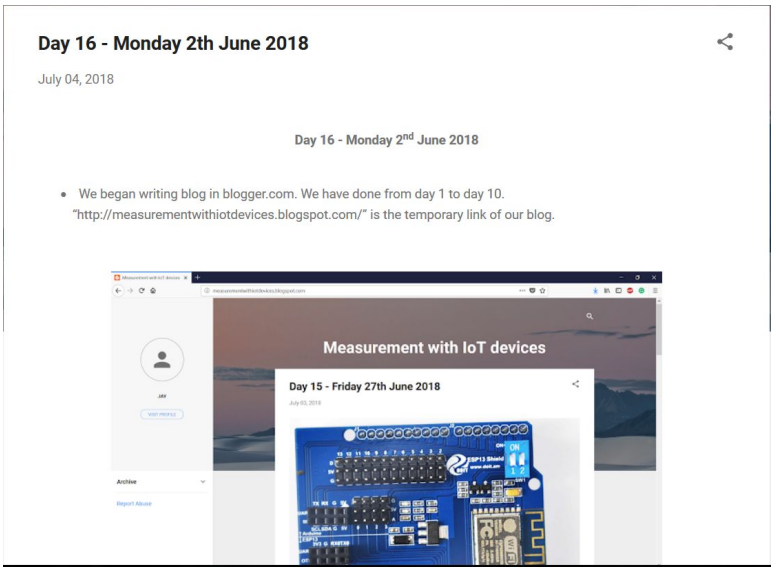
Visualization of Offset

```
char tmp = Serial.read();
if(tmp == 'o'){
  offset = Serial.readString().toFloat();
}
```

The code is as follow. The user can enter “o30.25” to set the offset distance, the distance from Load cell to the ground to 30.25 cm.

4 - July - Wednesday - Day 18

- We continue writing blog. Today, we have done from day 14 to day 16.



- We create a code for MEGA board to use with just Load cell and calibrate the weight into height. According to David’s advice and help.
    - From David’s advice about the description of code that is supposed to be a method calls “2 known weight” calibration.
    - This method needs to calibrate the Load cell at 2 points in the tube.
    - For example.
        - Fill the water and record the depth and value read by the Load cell and calls it “height1” and “weight1”.
        - Fill more water to record the second known weight, and calls it “height2” and “weight2”.
        - After this we have all the variable to calculate the height of water by knowing just the weight read by Load cell.
        - In order to get the currentHeight, we use the formula as follow.

$$currentHeight = Height1 - \left(\frac{Weight1 - currentWeight}{Weight1 - Weight2} \times (Height1 - Height2)\right)$$

- And then we tested this code with the hanging tube and glass. The result is improved from the previous method.

5 - July - Thursday - Day 19

- We combine the new calibration code into our main code that has the system which calibrates the load cell for measuring the height of water, measuring the distance by using ultrasonic sensor, stores them on the MicroSD card, and then sends them as an SMS and also monitors them on the serial monitor.
- Add the ability for the user to enter multiple variables to be used in the calculation. These include
    - Offset for the ultrasonic
    - First known height
    - Second known height
- To enter these value, the system will instruct the user to know which value to be entered first, making it easier for the user to keep track of the value.
- Because of all these changes, some part of the code that is no longer used is removed completely to make the code easier to read.
- We also include more comment and description of the pin connections between all devices in the header of our code.

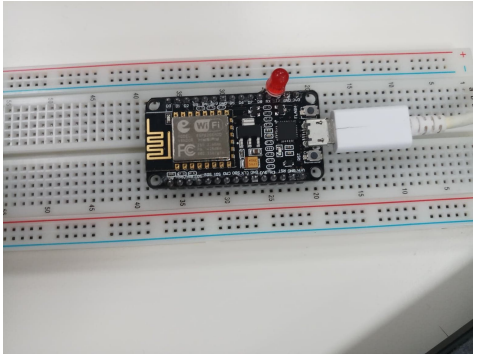
  For example:

```
Pin connection between Arduino Mega and HX711
 Arduino Mega pin 6 -> HX711 CLK
 Arduino Mega pin 5 -> HX711 DOUT
 Arduino Mega pin 5V -> HX711 VCC
 Arduino Mega pin GND -> HX711 GND

Pin connection between Arduino Mega and A7 Ai-Thinker
 Arduino Mega TX3 -> A7 Ai-Thinker RX
 Arduino Mega RX3 -> A7 Ai-Thinker TX
 A7 Ai-Thinker 5V -> A7 Ai-Thinker PWR
```

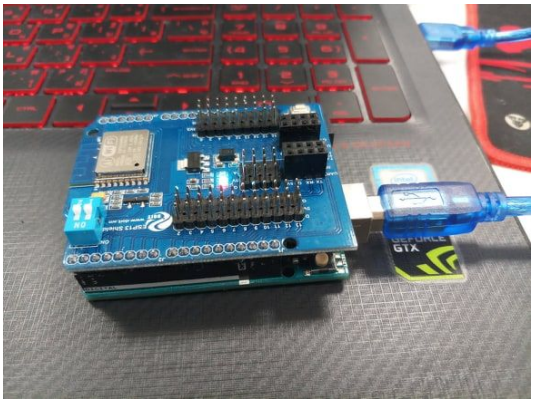
6 - July - Friday - Day 20

- Update description and document of our system and also newly manage package.
- Then, we do more research about Amica (Nodemcu) ESP32 board and ESP13 shield.
- We also continue trying to use Amica (Nodemcu) ESP32 board from the progress that we work on Monday, 29th June.



Amica (Nodemcu) ESP32 board

- We can control an LED from web browser.
  - We connect the Amica (Nodemcu) ESP32 board to the WiFi from ESP13 shield using SSID and password, where it connects to our wifi and create a web server, which can be accessed by looking through the serial Monitor of the Arduino window.



ESP13 shield

  - By making it easier, we use code from http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/.

```
4 const char* ssid = "DoitWiFi_Config";
5 const char* password = "";
```

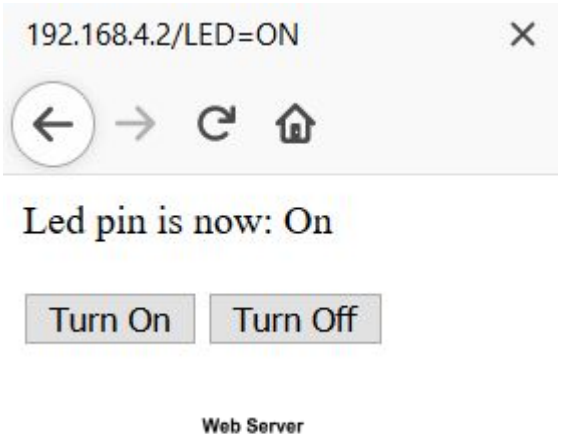
  - At first, we try to make this code work but it fails because of some library issues, then we include 2 libraries which are “ESP8266WiFi” and “WiFiConnector”. This can solve the above issues. And we change the SSID to “DoitWiFi_Config” and set password to NULL.

```
Connecting to DoitWiFi_Config
.....
WiFi connected
Server started
Use this URL to connect: http://192.168.4.3/
rl$□ʃ□ʃ#□ʃn♠□♠ʃ□lʃ□b|ʃʃ□ʃ□ʃ{ʃʃ□ʃ□"ʃp~ʃnʃ$noʃʃc♠#□pʃʃ$rl{lpʃNʃ□□□♠ʃ□$♠ʃʃ♠♠□b♠oʃ|□lʃ♠♠ʃb ʃʃNnʃ□$ʃʃ$`□ʃ□□oo□l □□□orʃʃʃo□♠B□ʃ□d□r'

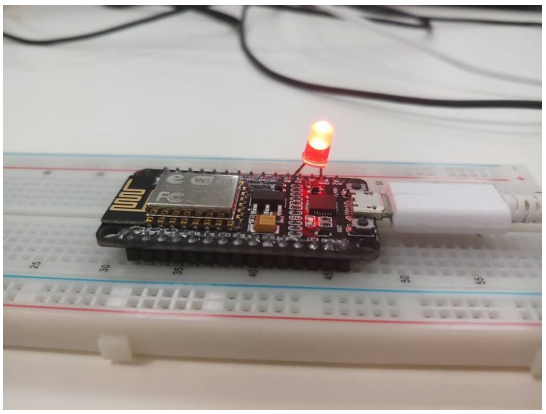
Connecting to DoitWiFi_Config
.
WiFi connected
Server started
Use this URL to connect: http://192.168.4.3/
```

Output on Serial Monitor

  - Finally, we can control LED from web server by pressing turn on and turn off button on 192.168.4.2. Now, we know that we can use this board to connect with our wifi and we think that it is very useful to apply with our system that we are doing right now in the field of communication and networking.



Web Server



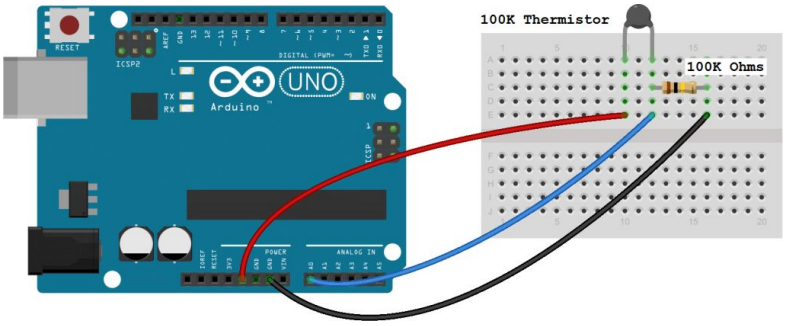
Light on

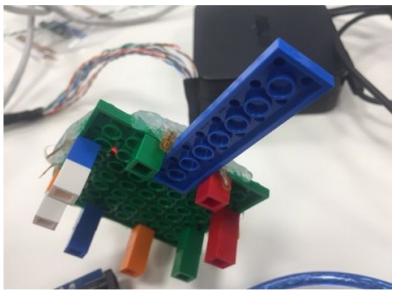
9 - July - Monday - Day 21

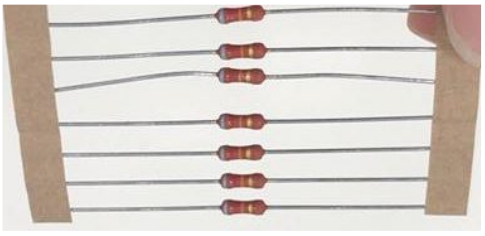
- We have soil moisture sensor now so we begin doing the research about the pin connection of it and Arduino board.

soil moisture sensor

- ET 016 General purpose Temperature Probe
    - We connect this temperature probe to MEGA board and get them to display the temperature by following the instruction of this website.
    - http://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/

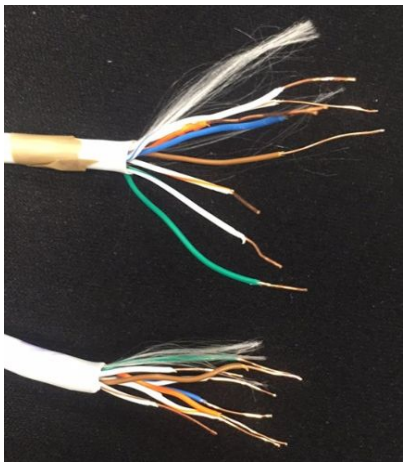


- 

    - By connecting the board and the probe to this configuration, this can be used to determine the resistance of the temperature probe when it changes according to the temperature outside.
    - We use 1 known resistor with the resistance of 82 kOhm. With 5 band color pattern of Gray - Red - Orange - Gold. That means the variable ‘R1’ will be 82000.
    - 

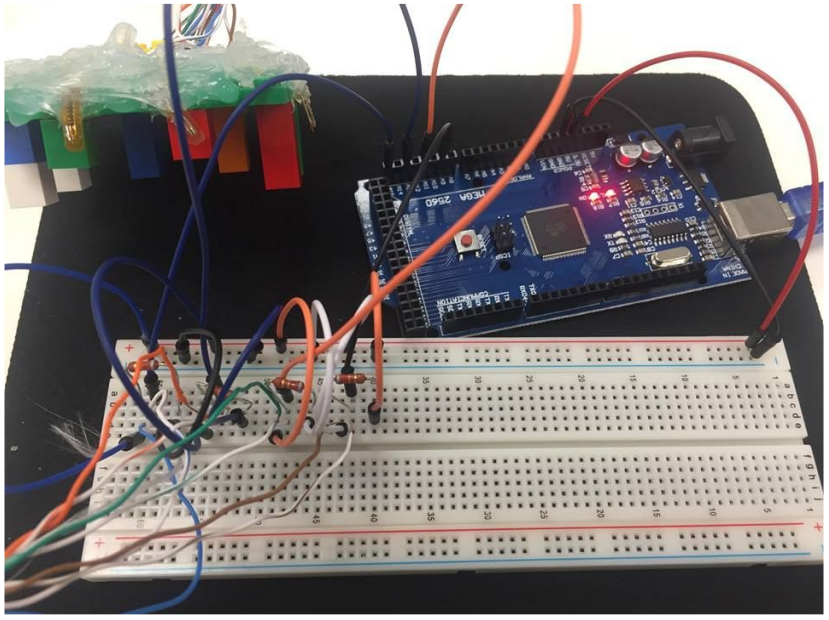
10 - July - Tuesday - Day 22

- Water current detector
    - 
    - This device has 4 temperature probes attached to its surrounding to read the water temperature.
    - The idea is that in the middle, another sensor that can emit heat will be put there and will only emit heat when the user needs to read the direction of the water. This works by reading the temperature on each temperature probes and finding the direction by observing the difference of the temperature.

- 
- The wire with a duct tape will be notated with double layer LEGO.

- The first step is to identify which probes corresponding to which pairs of wire. This can be done by manually connect each wire to the board and read the temperature, pressing it between fingers to see if it reading is increase.

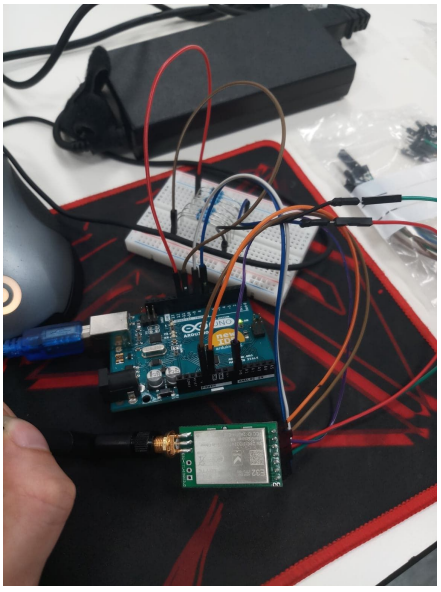
11 - July - Wednesday - Day 23

- Continue to work on the current detector device.
  - 
  - To read from all the probes, we need to use a long breadboard to help and connect like the picture.
  - After all the probes can read the correct temperature and work as intended, we begin to work on the code to ture this data into water direction.
  - We use blue LEGO to mark that direction as North.
  - There're two probes for each direction. ie, north, east, south, west. We take the same direction to find the mean and keep that value to find the direction.
  - ```
    // North = gre2 + red2
    no = (gre2 + bro2)/2;
    ```
    ```

- We then create a flag for either the water is flowing north to south or south, and either west to east or east to west for a total of 4 flag.
- If only one flag is toggled, that's mean the water is flowing to North or East, but if there're more than one flag being toggled, that means the water is flowing Southeast or Northwest for example.

12 - July - Thursday - Day 24

- David returns.
  - 55
  - 55
  - 55
- Test car rear sensor.
  - 55
  - 55
  - 55
- After that we turn to pay attention with E32-TTL-100 or E32-433T20D. According to https://github.com/Bob0505/E32-TTL-100, it needs series a 4.7k Ohm resistor between UNO's Tx, E32-TTL-100's Rx and UNO's Rx, E32-TTL-100's Tx but we only have 17.7k Ohm resistor so we try to make it as close as possible by making the parallel circuit with 17.7k Ohm to decrease the total of resistance as shown in below figure. Finally, we have 4.4k Ohm resistor to connect between Tx-Rx of 2 devices.



  - Moreover, we include SoftwareSerial library to allow serial communication on other digital pins of the Arduino to make it easy for testing whether it works or not.
  - At last, It doesn't work as we think therefore we are going to try this again later.