The Final Report

Team May1717

Client:Dan Stieler (Powerfilm Inc.)

Advisor:Gary Tuttle

Team Members:

Kebei Wang - Team leader

Kukjin Chung - Communication Leader

Trevor Brown - Key Concept Holder

Xiang Li - Webmaster

Yi Qiu - General Member

Team Website:

http://may1717.sd.ece.iastate.edu/index.html *Revised 04/22/2017*

Contents

1. Introduction

For doing this project, we are required to measure the low power consumption of a circuit system. The system should be remote sensing device, and be solar powered.

1.1. Project statement

This solar powered system needs to measure the temperature and humidity as well as translate the data through wireless transmission to mobile devices. The important thing is to provide power consumption data in different builds of Wireless solar temperature and humidity sensor circuit. The view of design and analysis is on how little power the system can consume and be useable. My team will try different methods of wireless transmissions and different sensor technologies to find out the lowest power consumption. We will also check the light power supply in different situations such as indoor and outdoor, or daytime and night.

1.2. Purpose

The temperature and humidity sensors are commonly used in the agriculture, pet shops, and weather forecasting. The wireless sensors make it much easier to obtain data anywhere. The wireless temperature and humidity sensor will be powered by solar panel and battery backup so that sensor can work for a long time without direct contact. The solar panel is beneficial in terms of self generating electricity system. The system should work with low power consumption effectively within 20 meters indoors and outdoors.

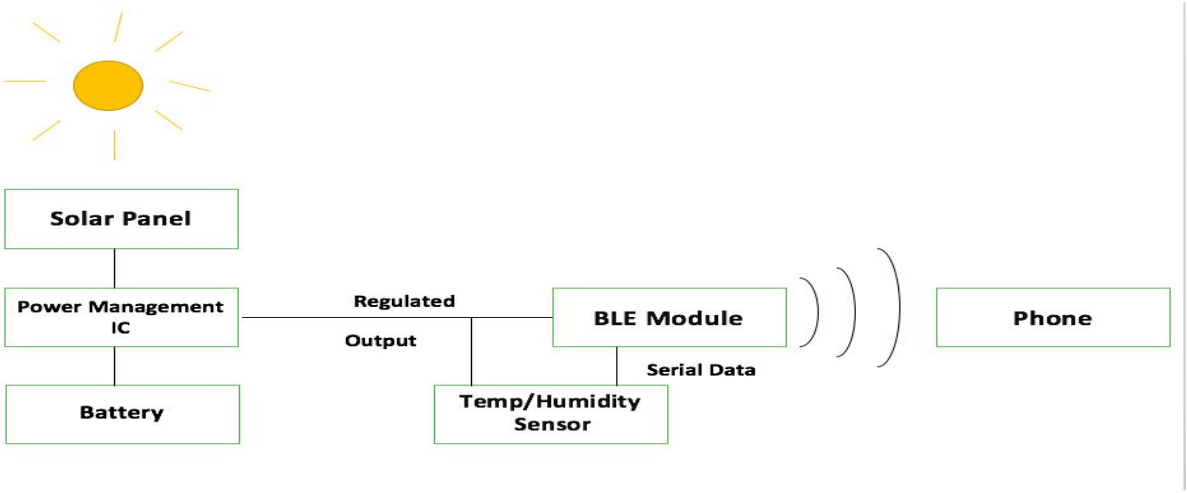
1.3. Goals

The goal of this project is to obtain the lowest system power consumption with using a BLE data transmission. Dan Stieler, the client (Powerfilm Company), wants the team to access many different ways to get the lowest power consumption sensor circuit. First of all, we will compare the power consumption in transmission between BLE(Bluetooth low energy) and Wifi. Secondly, we will compare temperature and humidity sensors of different brand and types. Then, we need to find a good sensor to meet our requirements. Lastly, we also need to check the possible solar energy indoor and outdoor. All the functions of sensor circuit is supposed to be executed

with the minimum light energy supply, and keep the system operate under 400 lux light level.

2 Design

The design of our project started with a review of the requirements of the project and and what would be the lowest power consumption for that part. The part that would consume the most power was the wireless transmission part. The wireless part has to connect to a smartphone so the choice was limited to cellular, bluetooth or WiFi. Cellular and WiFi use the most power so they were eliminated. Bluetooth was considered but using bluetooth low energy was ultimately decided upon. The sensing part of the project has to collect temperature and humidity data. Many different types of sensors were testing and the lowest power consumption and best accuracy was selected for use. The power side of the design consists of a solar panel and a battery. The solar panel was supplied by the client so we had to design around that part. The battery selection was based on the solar panel. Other types of storage were considered like supercapacitors but a lithium polymer battery was selected. The amount of light provided in the project requirement was very low, so selecting a small battery is important to keep it useable. If a too large of battery is selected, it will have a static discharge that will be hard to overcome. If the battery is too small it may be hard to provide enough current and stay cool. The interface between the solar panel, battery and power delivery was selected to be as efficient as possible. The client suggested the Texas Instrument energy harvesting circuit with regulated output as it is very flexible for testing.



3. Detail description and testing

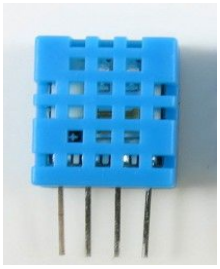
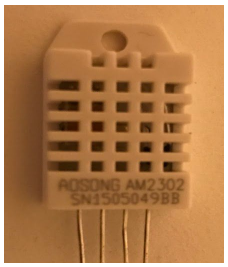
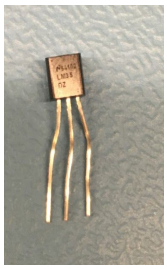
We have two parts of testing in this project. One is to test the hardware, and the other one is about software.

3.1. Hardware

The major hardware in this project is the transmission module and minor hardware components are microcontrollers, temperature and humidity sensors, battery, and solar panel. NRF8001 and NRF52 Bluetooth modules are used as the transmission modules. Based on the transmission modules, microcontrollers are selected. For the NRF8001, Arduino uno and nano are used to be tested. And the microcontroller NRF52 has BLEmodule itself so we tried to test the NRF52 microcontroller board. However, we had some problem with the nordic programing software uVsion5 to test NRF52 BLE function inside the microcontroller. So, we bought the NRF52 BLE breakout module and tested it. In this project, three temperature and humidity sensors(DHT11, DHT22, and HDC1080) and one temperature sensor(LM35) are tested. 3.7V Li batteries (LGABE11865 and dtp401525) are used and tested. One flexible solar panel is provided by the client Dan Stiler.

3.1.1. Sensors

We have three sensors: DHT11, DHT22,LM35, and HDC1080. The most important part for this is to compare the power consumption of the following sensor and pick the one with the lowest power consumption. On the other side, we also need to check other properties (like temperature range or voltage supply) to make sure that that the sensor we choose is good for our design.

DHT11	DHT22	HDC1080	LM35
			

	DHT22					
Voltage (V)	5	4	3.3	3	2.9	<2.8 not readable
Current During Read .5hz (ma)	1.61	1.58	1.56	1.55	1.04	
Power During Read (mW)	8.05	6.32	5.148	4.65	3.016	
Power On Time During Read 20ms						
	DHT11					
Voltage (V)	5	4	3.3	3	2.9	<2.8 not readable
Current During Read .5hz (ma)	2.4	1.68	1.23	1.03	0.9968	
Power During Read (mW)	12	6.72	4.059	3.09	2.89072	
Power On Time During Read 20ms						
	LM35 (needs ADC)					
Voltage (V)	5	4	3.3	<4 not recommended		
Current During Read (ua)	53.1	53	52.9			
Power During Read (uW)	265.5	212	174.57			
Power On Time During Read Always						

	HDC1080
Voltage (V)	3.3
Current During Read .5hz (mA)	0.041
Power During Read (mW)	0.1353
Power On Time During Read 20ms	
Idle current (mA)	0.037015

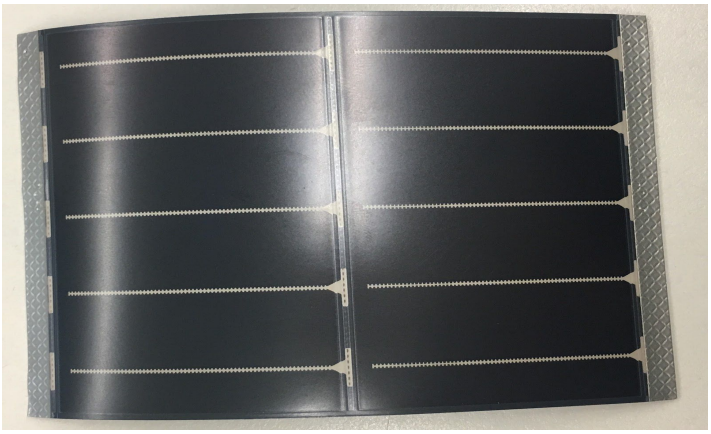
The above tables show the power, voltage and current results from the sensor testing. All of the sensors were tested in the same way with a ammeter recording data to simulink. Multiple voltages were tested to find the most efficient voltage.

	LM35	HDC1080	DHT22	DHT11
Minimum Tested Voltage Supply (V)	4	2.7	3	3
Temp Range (°C)	-55 ~ 150	-40 ~ 125	-40 ~ 125	0 ~ 50
Temp Accuracy (°C)	0.5	0.2	0.5	2
Humidity Range (%)	NA	0 ~100	0 ~ 100	20 ~ 90
Humidity Accuracy (%)	NA	2	2	5
Idle Current	0.0529mA	0.100uA	40uA	150uA
Average Current During 1Hz Polling	0.0529mA	0.041mA	1.56mA	1.23mA

The above table shows the accuracy and observed power data. The HDC1080 sensor is the most accurate and consumes the least amount of power. We chose this sensor for the final design.

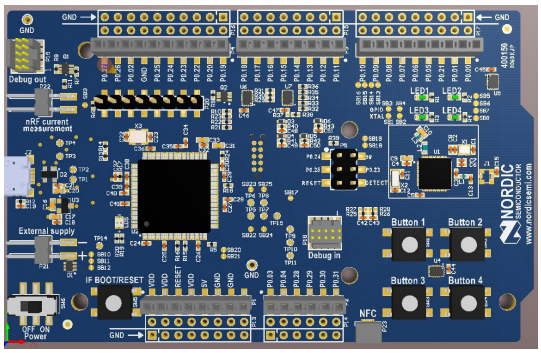
3.1.2. Solar Panel

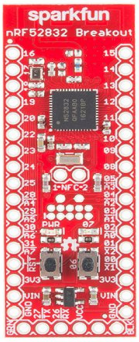
The solar panel was provided by PowerFilm. The operating voltage is 2V in our 400 lux test condition.



3.1.3. BLE Module

We have tested two BLE module, NRF52 and NRF8001. The supply voltage for NRF52 is from 1.7 V to 3.6V and the data rate is 1Mbps. On the other side, the supply voltage for NRF8001 is from 1.9V to 3.6V.

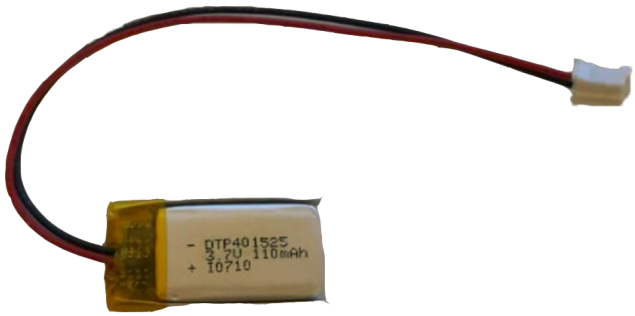
NRF52 Development Board

NRF 8001


Sparkfun NRF52832 Breakout


3.1.4. Battery

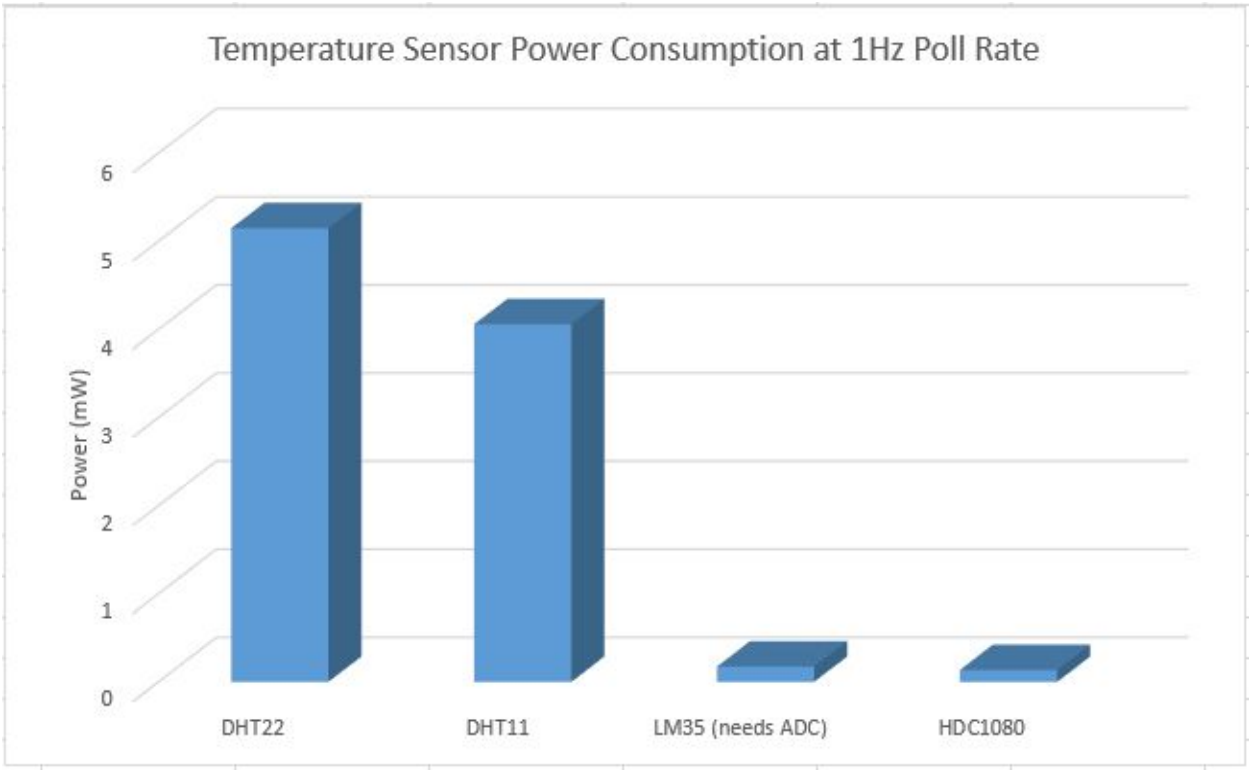
We are using the Lithium polymer battery 401525 for the battery part, the capacity for it is 110mAh and its operating voltage is 3.7V.

Lithium polymer battery 401525

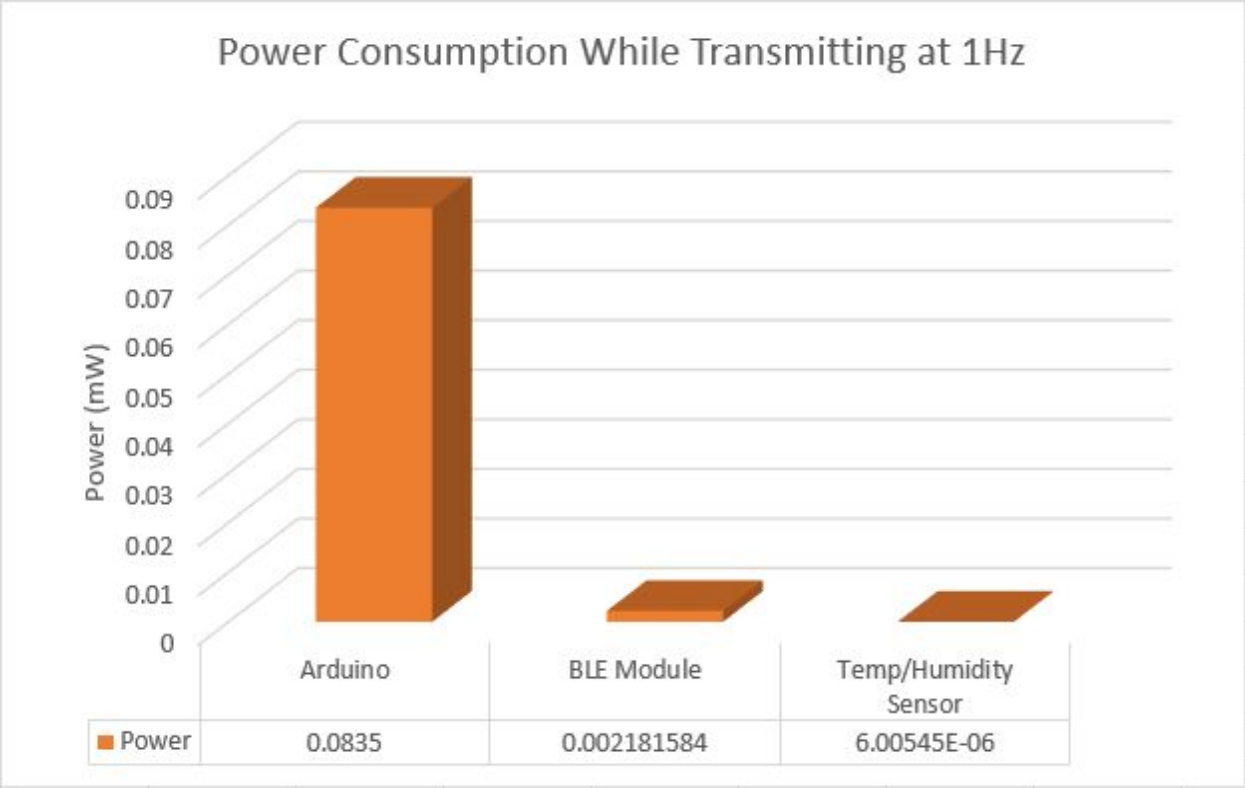


3.1.5 Power Consumption While Transmitting

We also tested the power consumption for different systems at a 1Hz poll rate.



First of all we compare the power consumption for different sensor and find that HDC1080 has the lowest power consumption, so we pick it to be the sensor for the final version of our design.



From the chart above, we can find that the power consumption for arduino is especially high compared to the BLE Module and sensors .

3.1.5 System Power Testing

The system power test was done with the 110mAh battery powering the Sparkfun NRF52832 breakout. The test consists of transmitting a value to a bluetooth devices 1 meter away once per second. The battery powered the module for 63120 packets sent over 17 hours and 32 minutes. The battery size is justified in the case it will not see daylight for extended periods of time. By comparison of the datasheets, the newer NRF52832 uses less power than the older NRF8001.

3.2. Software

The programming for this project occurred in two stages. The first stage was programming an arduino to interface with the HDC1080 temperature sensor and the Nordic BLE NRF8001 module. The second stage of

programming was with the Nordic NRF52 development kit interfacing with the HDC1080.

The first stage of programming is implemented by the Arduino IDE. There are two libraries used in the first stage; SPI.h, Adafruit_BLE_UART.h ,Wire.h and ClosedCube_HDC1080.h. SPI.h library supports the Serial Peripheral Interface(SPI) communication functions which is a serial data protocol used by Arduino microcontroller. Adafruit_BLE_UART.h library provides communication functions for NRF8001. Wire.h library is added to use the GPIO on the Arduino boards. ClosedCube_HDC1080.h provides its address configuration and functions to get the sensor data. Since the Arduino microcontroller is easy to program by Arduino IDE, we could easily apply the program code into the microcontroller.

The second stage of programming is with the NRF52832 Sparkfun breakout and the HDC1080 temperature sensor. In order to communicate with a computer, a FDTI usb to serial adapter is required. To program the breakout, it must be in bootloader mode. The breakout does not support auto-reset so you have to press reset and the pin 7 switch at the same time to enter bootloader mode. All programming was done in the Arduino IDE. This is possible due to the Arduino core being ported over to the NRF52. This means you can use Arduino functions and libraries to program a non Arduino microcontroller. The first important library we used to program the NRF52832 Sparkfun breakout was the BLE peripheral library. This library provides all the functions for communicating via BLE. The second important library we used was the HDC1080 temperature sensor library. This library was useful in programming the Arduino initially but the way it uses wire.h to communicate is not supported by the NRF52. So we had to code using functions that were supported by the NRF52. This was tricky as no one on the team was a computer or software engineering student.

3.2.1. First Stage Programming.

There are 4 libraries included in this code. REQ, RDY, and RST pins are defined to apply the NRF8001 BLE module. These pins are set up in this code.

In the setup, SPI, BLE and hdc1080 libraries functions are ready to set up.

```
/**************************************************************************/
/*!
    Constantly checks for new events on the nRF8001
*/
/**************************************************************************/
aci_evt_opcode_t laststatus = ACI_EVT_DISCONNECTED;
char b='p';
void loop()
{
  // Tell the nRF8001 to do whatever it should be working on.
  BTLEserial.pollACI();

  // Ask what is our current status
  aci_evt_opcode_t status = BTLEserial.getState();
  // If the status changed....
  if (status != laststatus) {
    // print it out!
    if (status == ACI_EVT_DEVICE_STARTED) {
        Serial.println(F("* Advertising started"));
    }
    if (status == ACI_EVT_CONNECTED) {
        Serial.println(F("* Connected!"));
    }
    if (status == ACI_EVT_DISCONNECTED) {
        Serial.println(F("* Disconnected or advertising timed out"));
    }
    // OK set the last status change to this one
    laststatus = status;
  }
```

The code above shows the beginning of the main. The main begins with BLE status check up codes. This part of code will print out the status of BLE module on the Serial window.

```
if (status == ACI_EVT_CONNECTED) {
  // Lets see if there's any data for us!
  if (BTLEserial.available()) {
    Serial.print("* "); Serial.print(BTLEserial.available()); Serial.println(F(" bytes available from BTLE"));
  }
  // OK while we still have something to read, get a character and print it out
  while (BTLEserial.available()) {
    char c = BTLEserial.read();
    b=c;
    Serial.print(c);
  }
```

The reading function codes are applied only if the BLEis in connected status. The BTLEserial.read function will transmit the data from the NRF8001 BLE module.

```
    // Next up, see if we have any data to get from the Serial console

      if (b =='x'){
      // Read a line from Serial
      Serial.setTimeout(100); // 100 millisecond timeout

      String s = String(hdc1080.readTemperature(), 3)+"C";
Serial.print(s);
  delay(100);
      // We need to convert the line to bytes, no more than 20 at this time
      uint8_t sendbuffer[20];
      s.getBytes(sendbuffer, 20);
      char sendbuffersize = min(20, s.length());

      Serial.print(F("₩n* Sending -> ₩"")); Serial.print((char *)sendbuffer); Serial.println("₩"");

      // write the data
      BTLEserial.write(sendbuffer, sendbuffersize+1);


      String s2 = String( hdc1080.readHumidity(), 3)+"%";

      s2.getBytes(sendbuffer, 20);
      Serial.print(F("₩n* Sending -> ₩"")); Serial.print((char *)sendbuffer); Serial.println("₩"");

      // write the data
      BTLEserial.write(sendbuffer, sendbuffersize+1);

       b = 'v';

       }


  }
```

Firstly, the code begins to read temperature data from hdc1080. And its data will be stored into String s and transfer to the parameter ‘sendbuffer’ to transmit to the receiver by Bluetooth. Next, the humidity data will be collected.

3.2.2. Second State Programming

```
int readData(int what){
  Wire.beginTransmission(0x04);
  Wire.write(what);
  Wire.endTransmission();

  delay(9);
  Wire.requestFrom(0x04, y );

  byte msb = Wire.read();
  byte lsb = Wire.read();

  return msb << 8 | lsb;

}
```

The above code shows the first implemented function. This function takes the raw data from the HDC1080 temperature sensor.

```
float readTemperature()  {

  int rawT = readData(0x00);
  return (rawT / pow(2, 16)) * 165 - 40;

}
```

The next bit of code is the temperature function. This takes the raw data and converts it to a useable temperature in ℃.

```
float readHumidity() {

  int rawH = readData(0x01);
  return (rawH / pow(2, 16)) * 100;
```

The last bit of code takes the raw humidity data and converts it to RH%.

```
// periodically sent temp and hum
void sendtemp() {
  if (BLESerial) {
    temperature = readTemperature();
    humidity = readHumidity();

    BLESerial.print("Temperature=");
    BLESerial.print(temperature);
    BLESerial.print("C ");
    BLESerial.print("Humidity=");
    BLESerial.print(humidity);
    BLESerial.println("%");

    delay(1000);



  }
```

The above code shows the overall function that sends the acquired data over the serial BLE connection.

```
/* TWI instance ID. */
#define TWI_INSTANCE_ID     0

 /* Number of possible TWI addresses. */
 #define TWI_ADDRESSES      127
 #define temp_address       0x00
 #define humid_address      0x01
 #define sensor_address     0x40
 #define config             0x02
 #define NORMAL_MODE 0U

static volatile bool m_xfer_done = false;
/* TWI instance. */
static const nrf_drv_twi_t m_twi = NRF_DRV_TWI_INSTANCE(TWI_INSTANCE_ID);


/**
 * @brief TWI initialization.
 */
 static uint8_t m_sample;
void twi_init (void)
{
    ret_code_t err_code;

    const nrf_drv_twi_config_t twi_config = {
       .scl                = ARDUINO_SCL_PIN,
       .sda                = ARDUINO_SDA_PIN,
       .frequency          = NRF_TWI_FREQ_100K,
       .interrupt_priority = APP_IRQ_PRIORITY_HIGH,
       .clear_bus_init     = false
    };

    err_code = nrf_drv_twi_init(&m_twi, &twi_config, NULL, NULL);
    APP_ERROR_CHECK(err_code);

    nrf_drv_twi_enable(&m_twi);
}
```

In NRF52 dk we try to use TWI.h replaced wire.h in Arduino code. we keep all the function logic the same as the Arduino code but the final code not work well. The code above defined the address of sensor. it also initialized the twi environment which allowed the scl and sda pins to exchange data from sensor. The code below is used to request data from different sensor address which will return temperature or humidity.

```
void read_temp(void)
{

    ret_code_t err_code;

    /* Writing to HDC1080_REG_CONF "0" set temperature sensor in NORMAL mode. */
    uint8_t reg[2] = {config, NORMAL_MODE};
    err_code = nrf_drv_twi_tx(&m_twi, sensor_address, reg, sizeof(reg), false);
    APP_ERROR_CHECK(err_code);


    /* Writing to pointer byte. */
    while (m_xfer_done == false);
    m_xfer_done = false;
          reg[0] = temp_address;

    err_code = nrf_drv_twi_tx(&m_twi, sensor_address, reg, 1, false);
    APP_ERROR_CHECK(err_code);
    while (m_xfer_done == false);

}
void read_humid(void)
{
    ret_code_t err_code;

    /* Writing to HDC1080_REG_CONF "0" set Humidity sensor in NORMAL mode. */
    uint8_t reg[2] = {config, NORMAL_MODE};
    err_code = nrf_drv_twi_tx(&m_twi, sensor_address, reg, sizeof(reg), false);
    APP_ERROR_CHECK(err_code);
    while (m_xfer_done == false);

    /* Writing to pointer byte. */
    reg[0] = humid_address;
    m_xfer_done = false;
    err_code = nrf_drv_twi_tx(&m_twi, sensor_address, reg, 1, false);
    APP_ERROR_CHECK(err_code);
    while (m_xfer_done == false);
}
```

4. Appendix

4.1 Operation Manual

Setup the system:

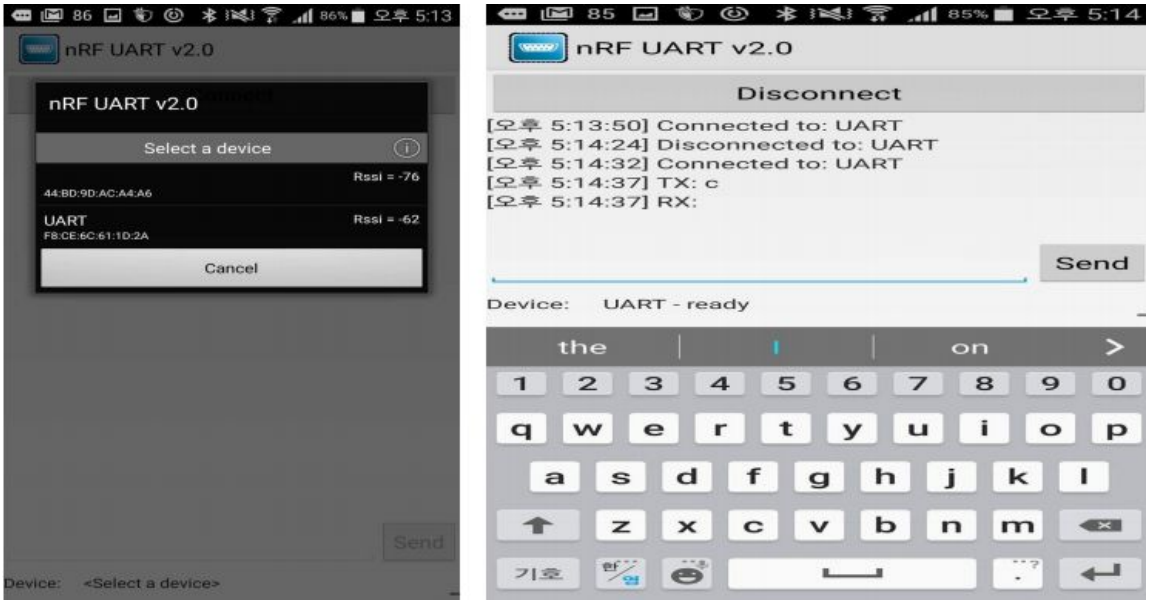
1. Connect the HDC1080 sensor with BLE module
2. Connect the BLE module with Microcontroller board
3. Connect the board with computer

Setup for the demo:

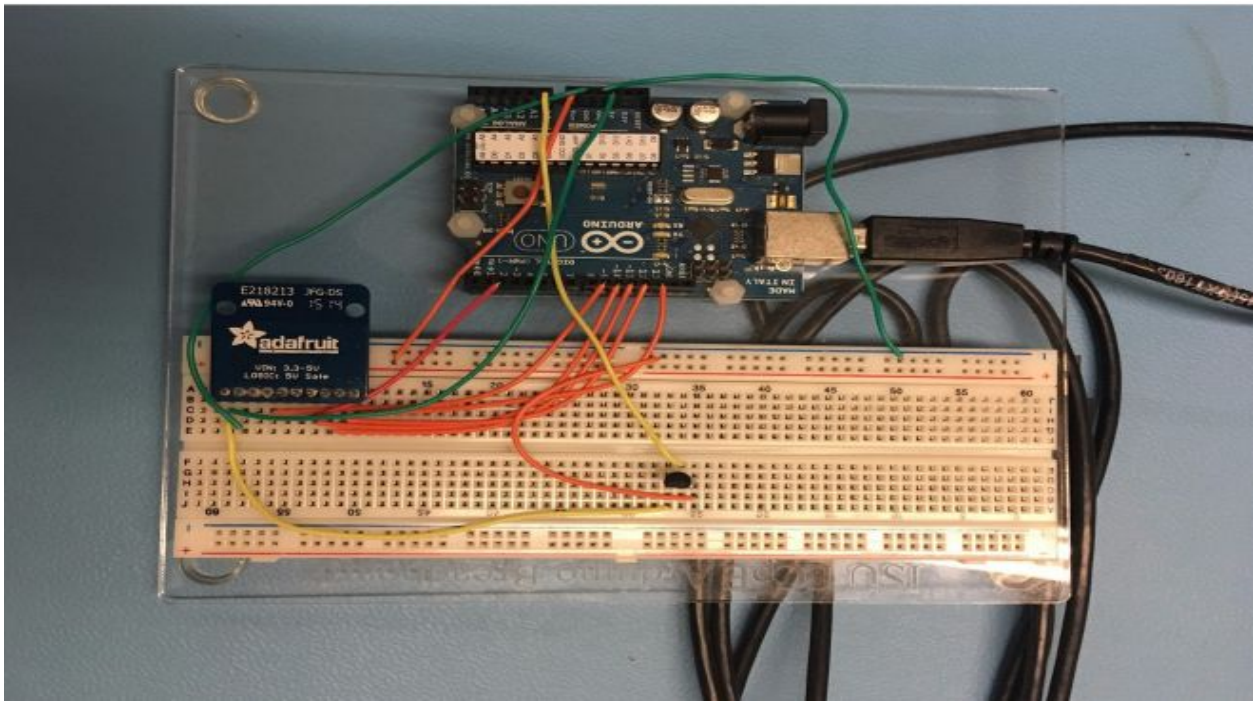
4. Open the Arduino IDE and open the code for test
5. compile the code into the circuit
6. remove the connecting with computer
7. connect the circuit with battery system
8. Turn on the power of the circuit
9. Turn on the bluetooth of your mobile device

Test the system:

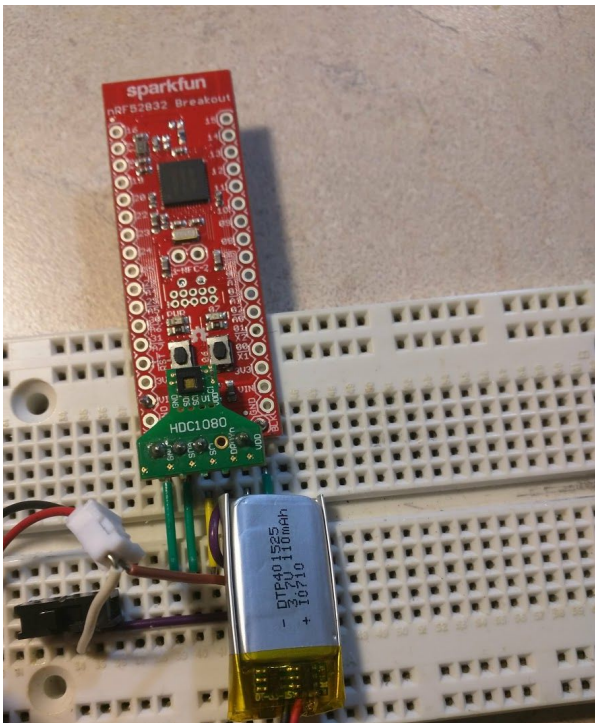
10. Open “nRF UART” app
11. Click “Connect” button
12. After it shows “connected”, type “t” and it will return temperature and humidity data in decimal number on the screen.



4.2 Alternative initial version



In the previous version we use LM35 sensor and Arduino Uno for the microcontroller. The restriction is that the Voltage supply level is higher than our requirement, and then we did some change with it.



The photo above shows the battery test configuration.

4.3 Other Considerations

The senior design is a good experience for us to work with a team. We started our senior design was not very easy. At the beginning, we couldn't contact with our client and don't have an advisor. Based on this situation, some of our team members searched our client's contact information and tried to contacted with him, and other members tried to contact with professors who can be our advisor. Finally, we figured out the problem. Then, there was a team member who is the only one has software engineering background wanted to change project. We respected his choice, and then some of us have to take some responsibility to do some research regarding with the programming. Within these two

semesters, we tried our best to take our responsibility clearly for each group member, and helped each others to finish some tasks. We also learned how to have a good communication with group members, client and our advisor so that we can solve the problems easily.

4.4 Bill of components

Wireless Solar Temperature and Humidity Sensor Circuit

Bill of components

Item	Qty.	Cost($)	part Description	Supplier	Supplier#	Sub Total($)
1	1	1.86	SENSOR TEMP ANLG VOLT TO-92-3	digikey.com	LM35DZ/NOPB-	1.86
2	1	5	DHT11 basic temperature-humidity sensor + extras	adafruit.com	386	5
3	1	9.95	DHT22 temperature-humidity sensor + extras	adafruit.com	385	9.95
4	1	29.09	LOW POWER HUMIDITY AND TEMP EVAL	digikey.com	296-43865-ND	29.09
5	1	21.49	ARDUINO NANO BOARD	digikey.com	1050-1001-ND	21.49
6	1	23.38	ATmega328P Arduino Uno AVR® ATmega MCU 8-Bit AVR Embedded Evaluation Board	digikey.com	1050-1024-ND	23.38
7	1	19.95	NRF52832 BREAKOUT	digikey.com	1568-1449-ND	19.95
8	1	19.95	BREAKOUT LE BT BLE4.0 NRF8001 V1	digikey.com	1528-1199-ND	19.95
9	1	1.5	rechargeable wholesale 3.7V small lithium battery 401525 100mAh	alibaba.com		1.5
Total						132.17