<https://www.youtube.com/watch?v=UrqPxwsPWGk>

const int pressureInput = A0;

float pressureValue = 0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

pressureValue = analogRead(pressureInput);

Serial.println(pressureValue);

delay(2500);

}

#include <Arduino.h>

#include <SPI.h>

#if not defined (\_VARIANT\_ARDUINO\_DUE\_X\_) && not defined (\_VARIANT\_ARDUINO\_ZERO\_)

#include <SoftwareSerial.h>

#endif

#include "Adafruit\_BLE.h"

#include "Adafruit\_BluefruitLE\_SPI.h"

#include "Adafruit\_BluefruitLE\_UART.h"

#include "BluefruitConfig.h"

#include "sine.h"

float DENSITY;

float SPECIFICGRAVITY;

float units;

float pressureValue = 0;

const int pressureInput = 7;

#include <Arduino.h>

#include <SPI.h>

#include "Adafruit\_BLE.h"

#include "Adafruit\_BluefruitLE\_SPI.h"

#include "Adafruit\_BluefruitLE\_UART.h"

#include "BluefruitConfig.h"

#define BLUEFRUIT\_SWUART\_RXD\_PIN 9

#define BLUEFRUIT\_SWUART\_TXD\_PIN 10

#define BLUEFRUIT\_UART\_CTS\_PIN 11

#define BLUEFRUIT\_UART\_RTS\_PIN 8

#define BLUEFRUIT\_UART\_MODE\_PIN 12

#define GRAVITATIONALACCELERATION 9.8

#define HEIGHTFLUID 3030

#define DENSITYWATER 997

#define BUFSIZE 128 // Size of our read buffer

#if SOFTWARE\_SERIAL\_AVAILABLE

#include <SoftwareSerial.h>

#endif

#define FACTORYRESET\_ENABLE 1

#define MINIMUM\_FIRMWARE\_VERSION "0.6.6"

#define MODE\_LED\_BEHAVIOUR "MODE"

#define SEND\_SECOND\_PLOT 0

SoftwareSerial bluefruitSS = SoftwareSerial(BLUEFRUIT\_SWUART\_TXD\_PIN, BLUEFRUIT\_SWUART\_RXD\_PIN);

Adafruit\_BluefruitLE\_UART ble(bluefruitSS, BLUEFRUIT\_UART\_MODE\_PIN,

BLUEFRUIT\_UART\_CTS\_PIN, BLUEFRUIT\_UART\_RTS\_PIN);

void error(const \_\_FlashStringHelper\*err) {

Serial.println(err);

while (1);

}

void setup(void)

{

pinMode(12,OUTPUT);

while (!Serial);

delay(500);

Serial.begin(115200);

Serial.println(F("Adafruit Bluefruit Command <-> Data Mode Example"));

Serial.println(F("------------------------------------------------"));

/\* Initialise the module \*/

Serial.print(F("Initialising the Bluefruit LE module: "));

if ( !ble.begin(VERBOSE\_MODE) )

{

error(F("Couldn't find Bluefruit, make sure it's in CoMmanD mode & check wiring?"));

}

Serial.println( F("OK!") );

if ( FACTORYRESET\_ENABLE )

{

/\* Perform a factory reset to make sure everything is in a known state \*/

Serial.println(F("Performing a factory reset: "));

if ( ! ble.factoryReset() ){

error(F("Couldn't factory reset"));

}

}

/\* Disable command echo from Bluefruit \*/

ble.echo(false);

Serial.println("Requesting Bluefruit info:");

/\* Print Bluefruit information \*/

ble.info();

Serial.println(F("Please use Adafruit Bluefruit LE app to connect in UART mode"));

Serial.println(F("Then Enter characters to send to Bluefruit"));

Serial.println();

ble.verbose(false);

/\* Wait for connection \*/

while (! ble.isConnected()) {

delay(500);

}

Serial.println(F("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"));

// LED Activity command is only supported from 0.6.6

if ( ble.isVersionAtLeast(MINIMUM\_FIRMWARE\_VERSION) )

{

// Change Mode LED Activity

Serial.println(F("Change LED activity to " MODE\_LED\_BEHAVIOUR));

ble.sendCommandCheckOK("AT+HWModeLED=" MODE\_LED\_BEHAVIOUR);

}

// Set module to DATA mode

Serial.println( F("Switching to DATA mode!") );

ble.setMode(BLUEFRUIT\_MODE\_DATA);

Serial.println(F("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"));

}

void loop(void)

{

// Check for user input

char n, inputs[BUFSIZE+1];

if (Serial.available())

{

n = Serial.readBytes(inputs, BUFSIZE);

inputs[n] = 0;

Serial.print("Sending: ");

Serial.println(inputs);

ble.print(inputs);

}

while ( ble.available() )

{

int c = ble.read();

Serial.print((char)c);

Serial.print(" [0x");

if (c <= 0xF) Serial.print(F("0"));

Serial.print(c, HEX);

Serial.print("] ");

}

// specific gravity equation

pressureValue = analogRead(pressureInput);

if (pressureValue>0){

DENSITY = pressureValue/GRAVITATIONALACCELERATION\*HEIGHTFLUID;

SPECIFICGRAVITY = DENSITY/DENSITYWATER;

Serial.print("[Specific Gravity]");

Serial.println(SPECIFICGRAVITY);

delay(2500);

units = SPECIFICGRAVITY;

Serial.print(units,5);

ble.print (units, 5);

}

//print value on graph

delay(1000);

if (SEND\_SECOND\_PLOT) {

if (units > 0) units = SPECIFICGRAVITY;

ble.print(",");

ble.print(sine\_wave[sineIndex]);

sineIndex++;

}

ble.println();

}