Blynk:

#define BLYNK\_PRINT SwSerial

#include <SoftwareSerial.h>

SoftwareSerial SwSerial(10, 11);

#include <BlynkSimpleStream.h>

char auth[] = "sGSrCJgH9hXgAo9WXe5xeaaG8B2nhruX";

BlynkTimer timer;

void sendPressure()

{

int sensorVal = analogRead(A1);

Blynk.virtualWrite(V0, sensorVal);

float voltage = (( sensorVal) \* 5.0) / 1023.0;

float pressure\_pascal = (3.0 \* ((float)voltage)) \* 1000000.0;

float pressure\_bar = pressure\_pascal / 10e5;

float pressure\_psi = pressure\_bar \* 14.5038;

Blynk.virtualWrite(V5, pressure\_bar);

Blynk.virtualWrite(V1, pressure\_psi);

}

void setup()

{

SwSerial.begin(9600);

Serial.begin(9600);

Blynk.begin(Serial, auth);

timer.setInterval(0, sendPressure);

}

void loop()

{

Blynk.run();

timer.run();

}

LCD AND DATALOGGER:

#include <LiquidCrystal\_I2C.h>

#include <Wire.h>

#include <RTClib.h>

#include <SD.h>

#define LCD\_ADDRESS 0x27

#define LCD\_ROWS 2

#define LCD\_COLUMNS 16

#define SCROLL\_DELAY 150

#define BACKLIGHT 255

RTC\_DS1307 RTC;

int n = 1;

float input\_voltage = 0.0;

int val;

#define LOG\_INTERVAL 1000

#define SYNC\_INTERVAL 10000

uint32\_t syncTime = 0;

const int chipSelect = 10;

File logfile;

void error(char \*str)

{

Serial.print(F("error: "));

Serial.println(str);

}

LiquidCrystal\_I2C lcd(0x27, 16, 2);

void setup(void)

{

Serial.begin(9600);

Serial.println(F("pressure reading off SKU237545 SENSOR"));

lcd.init();

lcd.backlight();

lcd.print(F("SKU237545 SENSOR" ));

lcd.setCursor(0, 1);

lcd.print(F(" bar and psi " ));

delay (2000);

Serial.print(F("Initializing SD card..."));

pinMode(10, OUTPUT);

if (!SD.begin(chipSelect)) {

error("Card failed, or not present");

}

Serial.println(F("card initialized."));

char filename[] = "LOGGER00.CSV";

for (uint8\_t i = 0; i < 100; i++) {

filename[6] = i / 10 + '0';

filename[7] = i % 10 + '0';

if (! SD.exists(filename)) {

logfile = SD.open(filename, FILE\_WRITE);

break;

}

}

if (! logfile) {

error("couldnt create file");

}

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(F("File No " ));

lcd.print(filename);

Serial.print(F("Logging to: "));

Serial.println(filename);

if (!RTC.begin()) {

logfile.println("RTC failed");

#if ECHO\_TO\_SERIAL

Serial.println(F("RTC failed"));

#endif

}

logfile.println("pressure ,bar ,reading ,psi,reading ");

delay(5000);

lcd.clear();

}

void loop(void)

{

int sensorVal = analogRead(A1);

float voltage = ((sensorVal) \* 5.0) / 1023.0;

Serial.print("Input volt:");

Serial.print(input\_voltage);

Serial.println();

Serial.print(sensorVal);

Serial.println();

Serial.print("sensor:");

float pressure\_pascal = (3.0 \* ((float)voltage)) \* 1000000.0;

float pressure\_bar = pressure\_pascal / 10e5;

float pressure\_psi = pressure\_bar \* 14.5038;

Serial.print("Pressure = ");

Serial.print(pressure\_bar);

Serial.print(" bars ");

Serial.print ("psi ");

Serial.print (pressure\_psi);

Serial.println();

DateTime now;

now = RTC.now();

delay((LOG\_INTERVAL - 1) - (millis() % LOG\_INTERVAL));

int H = now.hour();

int M = now.minute();

int S = now.second();

logfile.print(now.year(), DEC);

logfile.print("/");

logfile.print(now.month(), DEC);

logfile.print("/");

logfile.print(now.day(), DEC);

logfile.print(" ");

logfile.print(now.second(), DEC);

lcd.setCursor(0, 0);

lcd.print("Pressure=");

lcd.print(pressure\_bar);

lcd.print ("bar ");

lcd.setCursor(0, 1);

lcd.print("Pressure=");

lcd.print (pressure\_psi);

lcd.print ("psi ");

logfile.print(", ");

logfile.print(" Pressure = ");

logfile.print(",");

logfile.print(pressure\_bar);

logfile.print(",");

logfile.print(" bar ");

logfile.print(",");

logfile.print(pressure\_psi);

logfile.print(",");

logfile.print("psi");

logfile.println();

if ((millis() - syncTime) < SYNC\_INTERVAL) return;

syncTime = millis();

logfile.flush();

delay(10);

}