

```

/*
Igniter Data Acquisition and control code

Designed by Lloyd Droppers & Jasmine Cashbaugh Jan 2015

This code is available for reuse and adaption under
Creative Commons Attribution-ShareAlike 3.0
*/

// initilize variables
int aic1=0;
int aic2=1;
int aic3=2;
int      THERMISTORNOMINAL=10000;
int      TEMPERATURENOMINAL=25;
int BCOEFFICIENT=3984;
int SERIESRESISTOR=10000;
int nsamp = 3;
int sample1;
int sample2;
int sample3;
float temp1=0;
float pres1=0;
float pres2=0;
int doc1=12;
int doc2=11;
int doc3=10;
int doc4=6;
int do1=0;
int do2=0;
int do3=0;
int do4=0;
int arm=0;
int autoseq=0;
int lastin = 0;
int ser=0;
long time;
long timecheck=0;
long autostart;

void setup() {
  // initialize serial com at 57600 baud
  Serial.begin(57600);
  while(!Serial);
  // Define the digital output channels
  pinMode(doc1, OUTPUT);

```

```

pinMode(doc2, OUTPUT);
pinMode(doc3, OUTPUT);
pinMode(doc4, OUTPUT);

// Initilize the Channel (High= off!)
digitalWrite(doc1,HIGH);
digitalWrite(doc2,HIGH);
digitalWrite(doc3,HIGH);
digitalWrite(doc4,HIGH);
}

void loop() {

    time=millis();

// Do some serial Communication
    if (Serial.available()) {

        //read serial as ascii character
        int ser =Serial.read();
        lastin=ser;

        if(ser >= 48 && ser <= 122){
            //The ascii equivilent of numbers 0 - 9 are 48 - 57
            // so subtracting 46 from the ascii gives us 2 - 12 (the pins we want to u
            int usePin = ser;
            triggerPin(usePin);
        }

    }

//Read the Temp and Pressure
    readanalogchans();

//Print out the current state to serial
    Serial.print(time);
    Serial.print(",");
    Serial.print(lastin);
    Serial.print(",");
    Serial.print(do1);
    Serial.print(do2);
    Serial.print(do3);
    Serial.print(do4);
    Serial.print(",");
    Serial.print(temp1);
    Serial.print(",");

```

```

Serial.print(pres1);
Serial.print(",");
Serial.print(pres2);
Serial.print(",");
Serial.println(arm);

//Chill out till 20 ms have ellapsed from the start of the loop
while (20>(time-timecheck)){
time=millis();
//Serial.print(dol);
delay(1);}
timecheck=time;

}

//Takes inputs and writes as channels
// 1 = ascii 49 = D01 on
// 2 = ascii 50 = D02 on
// 3 = ascii 51 = D03 on
// 4 = ascii 52 = D04 on
// q = ascii 113 = D01 off
// w = ascii 119 = D02 off
// e = ascii 101 = D03 off
// r = ascii 114 = D04 off
// a = ascii 97 = arms autosequence
// s = ascii 115 = starts autosequence
// z = ascii 122 = aborts autosequence

void triggerPin(int pin){
  if (pin==49){
    digitalWrite(doc1,LOW);
    dol=1; }
  if (pin==113){
    digitalWrite(doc1,HIGH);
    dol=0;}
  if (pin==50){
    digitalWrite(doc2,LOW);
    do2=1;}
  if (pin==119){
    digitalWrite(doc2,HIGH);
    do2=0;}
  if (pin==51){
    digitalWrite(doc3,LOW);
    do3=1; }
  if (pin==101){
    digitalWrite(doc3,HIGH);

```

```

do3=0;}
if (pin==52){
    digitalWrite(doc4,LOW);
    do4=1;}
if (pin==114){
    digitalWrite(doc4,HIGH);
    do4=0;}
if (pin==97){
    arm=1;}
if (pin==122){
    arm=0;}
if (pin==115 && arm==1){
    Serial.println("Autosequence Begin");
    autosequence();}
}

//Run a Autosequence assigned below
void autosequence(){
    // Initiallizes all valves off at start of autosequence
    digitalWrite(doc1,HIGH);
    digitalWrite(doc2,HIGH);
    digitalWrite(doc3,HIGH);
    digitalWrite(doc4,HIGH);
    int stepstate=0;
    autostart=millis();

    while (stepstate>(-1)){
        //If z is a serial command abort autosequence
        int ser =Serial.read();
        if (ser==122){
            arm=0;
            Serial.println("abort");
            digitalWrite(doc1,HIGH);
            digitalWrite(doc2,HIGH);
            digitalWrite(doc3,HIGH);
            digitalWrite(doc4,HIGH);
            break;
        }
    }
    int buttonstate =digitalRead(doc1);

    //Walks though autosequence changing stepstate after time ellapses
    if (stepstate==0 && 20<(time-autostart)){
        digitalWrite(doc3,LOW);
        do3=1;
        stepstate=1; }
    if (stepstate==1 && 800<(time-autostart)){

```

```

    digitalWrite(doc1,LOW);
    do1=1;
    stepstate=2; }
if (stepstate==2 && 850<(time-autostart)){
    digitalWrite(doc2,LOW);
    do2=1;
    stepstate=3; }
if (stepstate==3 && 1200<(time-autostart)){
    digitalWrite(doc3,HIGH);
    do3=0;
    stepstate=4; }
if (stepstate==4 && 3000<(time-autostart)){
    digitalWrite(doc1,HIGH);
    digitalWrite(doc2,HIGH);
    do1=0;
    do2=0;
    stepstate=5; }
if (stepstate==5 && 3500<(time-autostart)){
    stepstate=-2; }

readanalogchans();

//Serial Com valve state during autosequence
//Make sure this jives with Serial Com above
Serial.print(time-autostart);
Serial.print(",");
Serial.print(stepstate);
Serial.print(",");
Serial.print(do1);
Serial.print(do2);
Serial.print(do3);
Serial.print(do4);
Serial.print(",");
Serial.print(temp1);
Serial.print(",");
Serial.print(pres1);
Serial.print(",");
Serial.print(pres2);
Serial.print(",");
Serial.println(arm);

while (20>(time-timecheck)){
time=millis();

delay(1);}
timecheck=time;

```

```

}

Serial.println("Autosequence Complete");
arm=0;
}

void readanalogchans(){

uint8_t i;
float averagel = 0;
float average2 = 0;
float average3 = 0;

// take nsamp samples in a row, with a 1ms delay between sample
// Then sums the samples

for (i=0; i< nsamp; i++) {
    sample1 =analogRead(aic1);
    averagel = averagel+sample1;
    sample2 =analogRead(aic2);
    average2 = average2+sample2;
    sample3 =analogRead(aic3);
    average3 = average3+sample3;
    delay(1);
}

// averages the samples
averagel /= nsamp;
average2 /= nsamp;
average3 /= nsamp;

// convert the values to resistance
// from Adafruit at learn.adafruit.com/thermistor/using-a-thermistor
averagel = 1023 / averagel - 1;
averagel = SERIESRESISTOR / averagel;

float steinhart;
    steinhart = averagel / THERMISTORNOMINAL;// (R/Ro)
    steinhart =log(steinhart);                // ln(R/Ro)
        steinhart      /=      BCOEFFICIENT;// 1/B * ln(R/Ro)
    steinhart += 1.0 / (TEMPERATURENOMINAL + 273.15);// + (1/To)
        steinhart      =      1.0      /      steinhart;// Invert
            steinhart      -=      273.15;// convert to C
templ=steinhart;

//convert to pressures using calibration derived elsewhere

```

```
pres1 = average2 / 1.239 + 1;  
pres2 = average3 / 1.287 + 2.3;  
}
```