```
/*

Copywrite (c) mgc Perth Western Australia 2020
Arduino UNO Servo point motor control.
M at the end of the command string designates Master transmission.
S at the end of the command string designates Slave transmission.

*/

#include <Servo.h>
String Rx = "";   // Read serial buffer
String Tx = "";   // Data to send to serial port
String Sig = "WP";// Signal designation prefix.
int Ti = 0;       // Used to set servo position
int Pwr = 7;      // Power relay control
int Grns = 9;     // Colour light green
int Grn = 8;      // Colour light red
int Red = 11;     // Colour light green
int Reds = 12;    // Colour light red
int adc0 = 0;     // Analog channel
int adc1 = 1;     // Analog channel
int Lmin = 190;     // Left minimum error value
int Rmax = 300;     // Right maximum error value
int Tmin = 1200;     // Minimum time out error value
int Tmax = 1400;     // Maximum time out error value
int MaxAmp = 650;    // Maximum Servo current limit
unsigned long dStart = 0;
unsigned long dFin = 0;
bool tFlag = false;
Servo myservo;    // create servo object to control a servo
                  // twelve servo objects can be created on most boards

void setup() {
  pinMode(Pwr, OUTPUT);
  pinMode(Red, OUTPUT);
  pinMode(Grn, OUTPUT);
  pinMode(Reds, OUTPUT);
  pinMode(Grns, OUTPUT);
  digitalWrite(Red, HIGH);
  digitalWrite(Grn, HIGH);
  digitalWrite(Grns, LOW);
  digitalWrite(Reds, LOW);
  myservo.attach(10);       // attaches the servo to pin 10 of the UNO
  delayMicroseconds(30000); // 18550 is 20ms - duration of the pusles
  myservo.write(40);
  analogReference(DEFAULT); // Set the analog referance to DEFAULT. 3V3 or 5V depending on
the board type.

  // initialize serial //
  Serial.begin(19200);
  Serial.setTimeout(25);
  delay(20);
```

```
  Serial.println("Serial Servo point control. V1.1");
}

void loop() {
  dFin = millis();

  if (tFlag && dFin - dStart >= Tmax) {
    myservo.detach();
    digitalWrite(Grn, LOW);
    digitalWrite(Grns, LOW);
    digitalWrite(Red, HIGH);
    digitalWrite(Reds, HIGH);
    Tx = Sig + "PT1S," + Tmax;
    delay(20);
    Serial.println(Tx);
    tFlag = false;
  }

  if (Serial.available()==0) {
  }

  Rx = Serial.readString();

  if(Rx.indexOf("SIGNOM") > -1){
    for (int i = 0; i < Rx.length(); i++) {
      if (Rx.substring(i, i+1) == ",") {
        Sig = Rx.substring(i+1);
      }
    }
    Sig = Sig.substring(0,Sig.length() - 2);
    delay(20);
    Serial.println (Sig);
  }

  if(Rx.indexOf("RESETM") > -1){
    digitalWrite(Pwr, LOW);
    delay(20);
    Serial.println ("RESETS");
  }

  if(Rx.indexOf("ESTOPM") > -1){
    myservo.write(40);
    digitalWrite(Pwr, HIGH);
    delay(20);
    Serial.println ("ESTOPS");
  }

  if(Rx.indexOf("TEST") > -1){
    adc0= analogRead(A0);
    delay(20);
    adc1= analogRead(A1);
    Serial.println (adc0);
```

```
  delay(20);
  Serial.println (adc1);
 }

 if(Rx.indexOf("VALUESM") > -1){
  delay(20);
  Tx = "VALUES," + String(Lmin) +"," + String(Rmax) +"," + String(Tmin) +"," + String(Tmax)
+"," + String(MaxAmp);
  Serial.println (Tx);
 }

 //  Get the minimum ERROR value //

 if(Rx.indexOf(Sig + "LE1M") > -1){

  for (int i = 0; i < Rx.length(); i++) {
   if (Rx.substring(i, i+1) == ",") {
    Ti = Rx.substring(i+1).toInt();
   }
  }
  Lmin = Ti;
  delay(20);
  Serial.println (Lmin);
 }

 // Get the maximum ERROR value  //

 if(Rx.indexOf(Sig + "RE1M") > -1){

  for (int i = 0; i < Rx.length(); i++) {
   if (Rx.substring(i, i+1) == ",") {
    Ti = Rx.substring(i+1).toInt();
   }
  }
  Rmax = Ti;
  delay(20);
  Serial.println (Rmax);
 }

 // Get the minimum ERROR time out value  //

 if(Rx.indexOf(Sig + "IT1M") > -1){

  for (int i = 0; i < Rx.length(); i++) {
   if (Rx.substring(i, i+1) == ",") {
    Ti = Rx.substring(i+1).toInt();
   }
  }
  Tmin = Ti;
  delay(20);
  Serial.println (Tmin);
 }
```

```
// Get the maximum ERROR time out value  //

if(Rx.indexOf(Sig + "AT1M") > -1){

  for (int i = 0; i < Rx.length(); i++) {
    if (Rx.substring(i, i+1) == ",") {
      Ti = Rx.substring(i+1).toInt();
    }
  }
  Tmax = Ti;
  delay(20);
  Serial.println (Tmax);
}

// Get the maximum Servo current  //

if(Rx.indexOf(Sig + "AMPM") > -1){

  for (int i = 0; i < Rx.length(); i++) {
    if (Rx.substring(i, i+1) == ",") {
      Ti = Rx.substring(i+1).toInt();
    }
  }
  MaxAmp = Ti;
  delay(20);
  Serial.println (MaxAmp);
}

// Do only if WPLP1M found in the string eg. WPLP1M,30  //
// Eg. for Wilson Park Station top points use string WPLP1M,30. //

if(Rx.indexOf(Sig + "LP1M") > -1){

  //  Extract 30 as an integer from commar delimited string (file type is CSV) WPLP1M,30 //
  //  NOTE max. 16 bit interger is -32768 to 32,768 use long for 32 bit integers  //

  for (int i = 0; i < Rx.length(); i++) {
    if (Rx.substring(i, i+1) == ",") {
      Ti = Rx.substring(i+1).toInt();
    }
  }
  myservo.attach(10);
  myservo.write(Ti);
  digitalWrite(Red, LOW);
  digitalWrite(Grn, LOW);
  digitalWrite(Reds, HIGH);
  digitalWrite(Grns, HIGH);
  delay(25);
  // Reads the servos internal pot //
  adc0= analogRead(A0);
  delay(20);
```

```
  Tx = Sig + "LP1S," + String(adc0);
  Serial.println (Tx);
  dStart = millis();
  tFlag = true;

  // Cancell the ERROR timer if the servo made it fo the minimum value  //

  delay(Tmin);
  adc0= analogRead(A0);

  if(adc0 <= Lmin){
    tFlag = false;
  }
}

// Do only if WPRP1M found in the string, eg. WPRP1M,90  //
// Eg. for Wilson Park Station top points use string WPRP1M,90. //

if(Rx.indexOf(Sig + "RP1M") > -1){

  //  Extract 90 as an integer from commar delimited string (file type is CSV) WPRP1,90 //
  //  NOTE 90 max for most points.
  //  max. 16 bit interger is -32768 to 32,768 use long for 32 bit integers  //

  for (int i = 0; i < Rx.length(); i++) {
    if (Rx.substring(i, i+1) == ",") {
      Ti = Rx.substring(i+1).toInt();
    }
  }
  myservo.attach(10);
  myservo.write(Ti);
  digitalWrite(Red, HIGH);
  digitalWrite(Grn, HIGH);
  digitalWrite(Reds, LOW);
  digitalWrite(Grns, LOW);
  // Reads the servos internal pot //
  adc0= analogRead(A0);
  Tx = Sig + "RP1S," + String(adc0);
  delay(20);
  Serial.println (Tx);
  dStart = millis();
  tFlag = true;

  // Cancell the ERROR timer if the servo made it fo the maximum value  //

  delay(Tmin);
  adc0= analogRead(A0);

  if(adc0 >= Rmax){
    tFlag = false;
  }
```

```
  }

  // If points are jammed or split set signal to DANGER red. //

  adc1= analogRead(A1);

  if(MaxAmp <= 200){
    MaxAmp = 600;
  }

  if(adc1 >= MaxAmp){
    myservo.write(40);
    digitalWrite(Pwr, HIGH);
    delay(20);
    Tx = "MAXAMPS," + String(adc1) +"," + String(MaxAmp);
    Serial.println (Tx);
  }

  if(Rx.indexOf(Sig + "PJ1M") > -1){
    delay(20);
    Serial.println(Sig + "PJ1S");
    digitalWrite(Grn, LOW);
    digitalWrite(Grns, LOW);
    digitalWrite(Red, HIGH);
    digitalWrite(Reds, HIGH);
  }

  Rx = "";
  Serial.flush();
}
```