```
/*
  WXtrack Tracking

  Uses Mega2560 Board

  Modified by Pierre Marie GAYRAL F5XG 30 january 2015
  Used in conjunction with WXtrack from David Taylor :
 http://www.satsignal.eu/software/wxtrack.htm

For more details on FlexiTimer2 see:
http://www.arduino.cc/playground/Main/FlexiTimer2
https://github.com/wimleers/flexitimer2

 When new serial data arrives, this sketch adds it to a String.
 When a newline is received, the loop prints the string and
 clears it.

 Created 9 May 2011
 by Tom Igoe

 This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/SerialEvent

 */
#include <FlexiTimer2.h>        // library for software interrupt
#include <Servo.h>
#include<SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX


#define Laser_Pin 6             // Pin 6 for Laser diode

Servo Servo_AZ, Servo_EL;       // create servo(s) object to control a servo

String inputString = "";        // a string to hold incoming data
boolean stringComplete = false;  // whether the string is complete
int Az_0, Az_End;               // used for intermediate computations
int El_0, El_End;               // used for intermediate computations
int AZ_degree, EL_degree;        // value to be sent to servo
int AZ_Min = 70, AZ_Max = 290;   // limits angle to be converted later
int AZ_K = 39;                  // mechanical angle to align exactly
                                // the servo to right direction
int EL_K = 23;                  // mechanical angle to align exactly
                                // the servo to right direction


int Deg_Servo_Maxi = 180;       // limit max of servo NB: to be tested with your
```

```
                                  // own servo
int Parc_AZ = 0, Parc_EZ = 23;   // parking values

void setup() {
  // set pin 6 as output for laser
 pinMode(Laser_Pin, OUTPUT);
  // call function to blink Laser:
  FlexiTimer2::set(150, 1.0/1000, flash); // call every 500ms "ticks"
 FlexiTimer2::start();

  // initialize serial:
 Serial.begin(9600);
  Serial1.begin(9600);          // initialize COM1 in the Arduino board
  // reserve 200 bytes for the inputString:
 inputString.reserve(200);

  // initialize servos:
  Servo_EL.attach(8);          // attaches the servo EL on pin 8 to the servo
object
  Servo_AZ.attach(9);          // attaches the servo AZ on pin 9 to the servo
object
  Servo_AZ.write(Parc_AZ);     // sets the servo position park
  Servo_EL.write(Parc_EZ);     // sets the servo position park
}

void loop() {
  serialEvent();                // ? new line available
  //
  if (stringComplete) {         // if yes, then,
     Extract_Datas();           // go to function Extract_Datas

      AZ_degree = map(AZ_degree + AZ_K, AZ_Min, AZ_Max, Deg_Servo_Maxi, 0); //
scale it to use it with
                                                                  // the servo
(value between 0 and Deg_Servo_Maxi)
         if (AZ_degree >= Deg_Servo_Maxi) AZ_degree = Deg_Servo_Maxi;
           Servo_AZ.write(AZ_degree);                // sets the servo position
according to the scaled value
            delay(15);                               // waits for the servo to
get there

      EL_degree = map(EL_degree, 0, 180, 0, 180); // invert it to use it with
the servo
       Servo_EL.write(EL_degree + EL_K);          // sets the servo position
according to the scaled value
       delay(15);                                 // waits for the servo to get
there
  }
}
```

```
void serialEvent() {
/*
-------------------------------------------------------------
 SerialEvent occurs whenever a new data comes in the
 hardware serial RX.  This routine is run between each
 time loop() runs, so using delay inside loop can delay
 response.  Multiple bytes of data may be available.
-------------------------------------------------------------
 */
  while (Serial1.available()) {
    // get the new byte:
    char inChar = (char)Serial1.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if (inChar == '\r') {
      stringComplete = true;
    }
  }
}

void Extract_Datas(){
/* print the string when a newline arrives:
-------------------------------------------------------------------------------
    The string received has the followig form i.e. :
    AZ268.0 EL56.0
    We extract only the figures 268 and 56 from the above string
-------------------------------------------------------------------------------
*/
    Az_0 = inputString.indexOf('Z');      // location of Z
    Az_End = inputString.indexOf('.');    // 1st point in the String
    El_0 = inputString.indexOf('L');      // location of L

    //************************
    //*   String from COM1  *
    //************************
    Serial.println("-");
    Serial.print("String = " + inputString + "\r\n");

    //*************
    //*   Azimut  *
    //*************
    String Azimut = inputString.substring(Az_0+1,Az_End);// extract string Azimut
    Serial.print("Azimut = " + Azimut + "\r\n");

    //****************
```

```
    //*   Elevation  *
    //****************
    String Elevation = inputString.substring(El_0+1);// extract only string
Elevation
    El_0 = Elevation.indexOf('L');                  // position of character L
    El_End = Elevation.indexOf('.');                // 1st point in the String
Elevation
    String El = Elevation.substring(El_0+1,El_End); // format to get only figures
    Serial.print("Elevation = " + El + "\r\n");

    // clear the string:
    inputString = "";
    stringComplete = false;

    // converting strings to  integer
    AZ_degree = Azimut.toInt();
    Serial.print(AZ_degree + "\r\n");
    EL_degree = El.toInt();

  }

void flash()
{
static boolean output = HIGH;

  digitalWrite(Laser_Pin, output);
  output = !output;
}
```