**Tape Follower Code**

/\*

This code is used to drive a robot to follow a reflective line of tape.

This code runs servomotors at the same speed. The red wire of each servo goes

to power and the black wire goes to ground. The white wire of each servo goes

to one of the digital pins on the Arduino (you pick the pins). The IR sensors connect to

analog pins 0 and 1.

\*/

// define global variables for the IR sensor on the left side of robot

//as if you were sitting in the driver’s seat

int analogPin0 = 0; // define a variable for the left IR senor pin (pin 3, output pin, from the left IR sensor

 // will go to pin 0 on the Arduino

int sensorValL = 0; // variable to store the value coming from the sensor, initialize it to 0

// define global variables use for the IR sensor on the right side of robot

int analogPin1 = 1; //define a variable for the right IR senor pin (pin 3, output pin, from the right IR sensor

// will go to pin 1 on the Arduino)

int sensorValR = 0; // variable to store the value coming from the sensor, initialize it to 0

// define global variables use for servomotors

int leftMotorPin = 6; //select a digital pin to drive the left motor

int rightMotorPin =5; //select a digital pin to drive the right motor

int leftMotorPulseWidth = 100; //choose a pulse width in milliseconds between 10 and 200, larger pulse widths result in faster servo speeds

int rightMotorPulseWidth = 100; //choose a pulse width in milliseconds between 10 and 200, larger pulse width result in faster servo speeds

int level = 600; //this is value determined by measuring reflections off a piece of reflective tape

 // Students determine this value

//SETUP

void setup()

{

 pinMode(leftMotorPin, OUTPUT); // configure the pin called lefMotorPin as an output

 pinMode(rightMotorPin, OUTPUT); // configure the pin called rightMotorPin as an output

}

//LOOP

void loop()

{

//read both IR sensor values

sensorValL = analogRead(analogPin0); // left IR sensor

sensorValR = analogRead(analogPin1); //right IR sensor

 //Create appropriate pulse for the left motor to turn counter-clockwise

//pulse widths of 1500 microseconds stop the motor, pulse widths greater

// than 1500 microseconds cause the motor to turn counter-clockwise

//pulse widths less than 1500 microseconds cause the motor to turn clockwise

 digitalWrite(leftMotorPin, HIGH);

 delayMicroseconds(1500 + leftMotorPulseWidth);

 digitalWrite(leftMotorPin, LOW);

//low state of pulse needs to be on the order of 20 ms

delayMicroseconds(20000);

//Create appropriate pulse for the right motor to turn at the same speed as the other except clockwise

 digitalWrite(rightMotorPin, HIGH);

 delayMicroseconds(1500 - rightMotorPulseWidth);

 digitalWrite(rightMotorPin, LOW);

//low state of pulse needs to be on the order of 20 ms

delayMicroseconds(20000);

// decide if the left front side of the robot is going off the reflective tape (IR sensor values greater than //600

// and stop the right motor to steer the robot back on course

if ( sensorValL > level) rightMotorPulseWidth = 0;

// if the robot is on course (following the tape), keep the right motor running

if ( sensorValL <= level) rightMotorPulseWidth = 100;

// decide if the right front side of the robot is going off the reflective tape (IR sensor values greater than //600

// and stop the left motor to steer the robot back on course

if ( sensorValR > level) leftMotorPulseWidth = 0;

// if the robot is on course (following the tape), keep the left motor running

if ( sensorValR <= level) leftMotorPulseWidth = 100;

//stop the robot at a piece of black electrical tape placed over the reflective tape

if ( sensorValL > level) leftMotorPulseWidth = 0;

if ( sensorValR > level) rightMotorPulseWidth = 0;

}