Yanru Yin

Arduino Robotic

**Final Project**

**Robot Description**

This robot has two motors, each one connect to a wheel. It also has a Pixy CMUcam5 Sensor and one 9V battery, everything are connect to Ardurino UNO.

The robot I create can remember multiple things and follow them. It has two parts, the first part is Pixy CMUcam5 Sensor, this sensor can remember things and position them, the second part is a little car with three wheels, it can move left, right, forward and backward. By connect the two parts together, the sensor will give the object”s position information to the car, then the car will change direction based on the position information and follow the object.

**Process**

1. **Let the Pixy CMUcam5 Sensor able to remember objects.**

I go to the website <http://www.cmucam.org/projects/cmucam5/wiki> Find the way to let the sensor connect the Arduino Uno and remember objects by download *Pixymon.*

Then I trying to learn some coding tutorial online, to make the robot able to follow object.

The first website I go to is <https://www.youtube.com/watch?v=Ne6wLCIa1xs> I understand most of the code and find out it is to complicated, the code is to find the distance between center of the camera and the center of the object then follow the object by reduce the distance.

**The code:**

//---------------------------------------

// Track blocks via the Pixy pan/tilt mech

// (based in part on Pixy CMUcam5 pantilt example)

//---------------------------------------

int TrackBlock(int blockCount)

{

int trackedBlock = 0;

long maxSize = 0;

Serial.print("blocks =");

Serial.println(blockCount);

for (int i = 0; i < blockCount; i++)

{

if ((oldSignature == 0) || (pixy.blocks[i].signature == oldSignature))

{

long newSize = pixy.blocks[i].height \* pixy.blocks[i].width;

if (newSize > maxSize)

{

trackedBlock = i;

maxSize = newSize;

}

}

}

int32\_t panError = X\_CENTER - pixy.blocks[trackedBlock].x;

int32\_t tiltError = pixy.blocks[trackedBlock].y - Y\_CENTER;

panLoop.update(panError);

tiltLoop.update(tiltError);

pixy.setServos(panLoop.m\_pos, tiltLoop.m\_pos);

oldX = pixy.blocks[trackedBlock].x;

oldY = pixy.blocks[trackedBlock].y;

oldSignature = pixy.blocks[trackedBlock].signature;

return trackedBlock;

}

The **second tutorial** I learned is this website <https://www.youtube.com/watch?v=Ne6wLCIa1xs>. This tutorial helped me. I download a code that can give the object’s position information.

**The Code:**

//

// begin license header

//

// This file is part of Pixy CMUcam5 or "Pixy" for short

//

// All Pixy source code is provided under the terms of the

// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).

// Those wishing to use Pixy source code, software and/or

// technologies under different licensing terms should contact us at

// cmucam@cs.cmu.edu. Such licensing terms are available for

// all portions of the Pixy codebase presented here.

//

// end license header

//

// This sketch is a good place to start if you're just getting started with

// Pixy and Arduino. This program simply prints the detected object blocks

// (including color codes) through the serial console. It uses the Arduino's

// ICSP port. For more information go here:

//

// http://cmucam.org/projects/cmucam5/wiki/Hooking\_up\_Pixy\_to\_a\_Microcontroller\_(like\_an\_Arduino)

//

// It prints the detected blocks once per second because printing all of the

// blocks for all 50 frames per second would overwhelm the Arduino's serial port.

//

#include <SPI.h>

#include <Pixy.h>

// This is the main Pixy object

Pixy pixy;

void setup()

{

Serial.begin(9600);

Serial.print("Starting...\n");

pixy.init();

}

void loop()

{

static int i = 0;

int j;

uint16\_t blocks;

char buf[32];

// grab blocks!

blocks = pixy.getBlocks();

// If there are detect blocks, print them!

if (blocks)

{

i++;

// do this (print) every 50 frames because printing every

// frame would bog down the Arduino

if (i%50==0)

{

sprintf(buf, "Detected %d:\n", blocks);

Serial.print(buf);

for (j=0; j<blocks; j++)

{

sprintf(buf, " block %d: ", j);

Serial.print(buf);

pixy.blocks[j].print();

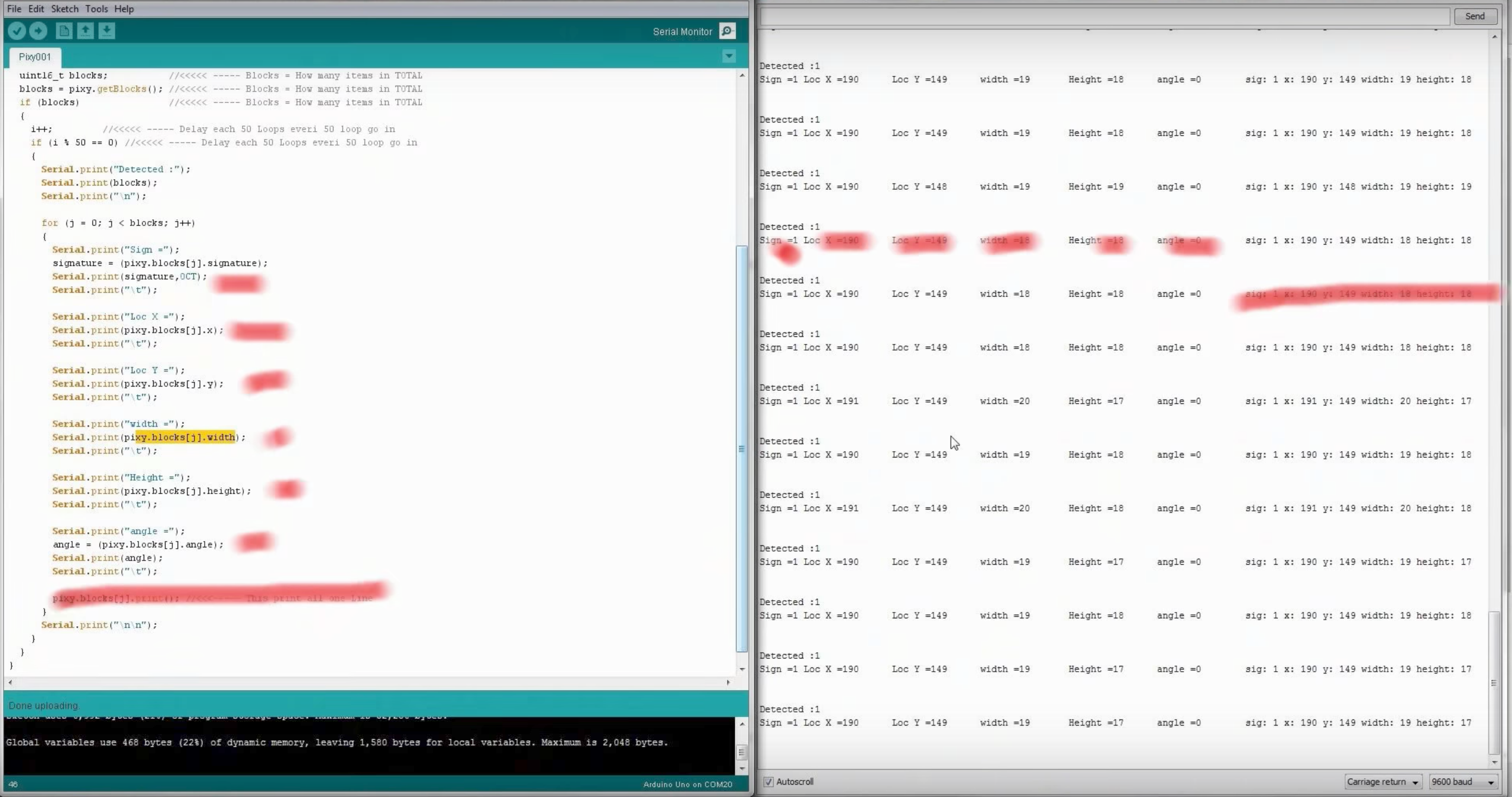
}

}

}

}

I learned how to read the information in serial printing.

After I understand these information, I will be able to simplify the coding to make the robot follow object based on X range.

At the **beginning** the robot can follow the object by turn to left and right, I decided the camera of the sensor in to half when the object appear on the left side the camera the car will turn left, and appear on the right side the car will turn right, but the motion of the car is not very smooth. It always turn around in circle, because the sensor can’t detect the object fast enough.

**The code is:**

#include <SPI.h>

#include <Pixy.h>

// This is the main Pixy object

Pixy pixy;

int objectX;

int screenWidth = 320;

int inPin = 7; // choose the input pin (for a pushbutton)

int val = 0; // variable for reading the pin status

const int leftForward = 4;

const int leftBackward = 2;

const int rightForward = 5;

const int rightBackward = 3;

void setup()

{

Serial.begin(9600);

Serial.print("Starting...\n");

pixy.init();

// put your setup code here, to run once:

pinMode(2, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(3, OUTPUT);

pinMode(inPin, INPUT); // declare pushbutton as input

}

void loop()

{

static int i = 0;

int j;

uint16\_t blocks;

char buf[32];

// grab blocks!

blocks = pixy.getBlocks();

// If there are detect blocks, print them!

if (blocks >= 1)

{

i++;

// do this (print) every 50 frames because printing every

// frame would bog down the Arduino

if (i % 50 == 0)

{

sprintf(buf, "Detected %d:\n", blocks);

Serial.print(buf);

for (j = 0; j < blocks; j++)

{

sprintf(buf, " block %d: ", j);

Serial.print(buf);

Serial.print("J=");

Serial.println(j);

pixy.blocks[j].print();

objectX = pixy.blocks[j].x;

//determine direction, divide the camera in to half

//if the object appear in the right part go right, left part go left.

if (objectX > (screenWidth / 2)) {

turnRight();

} else {

turnLeft();

}

Serial.println(pixy.blocks[j].signature);

}

}

}

}

//car turn right

void turnRight() {

digitalWrite(leftBackward, LOW);

digitalWrite(leftForward, LOW);

digitalWrite(rightForward, HIGH);

digitalWrite(rightBackward, LOW);

delay(1);

Serial.println("right");

}

//car turn left

void turnLeft() {

digitalWrite(leftBackward, LOW);

digitalWrite(leftForward, HIGH);

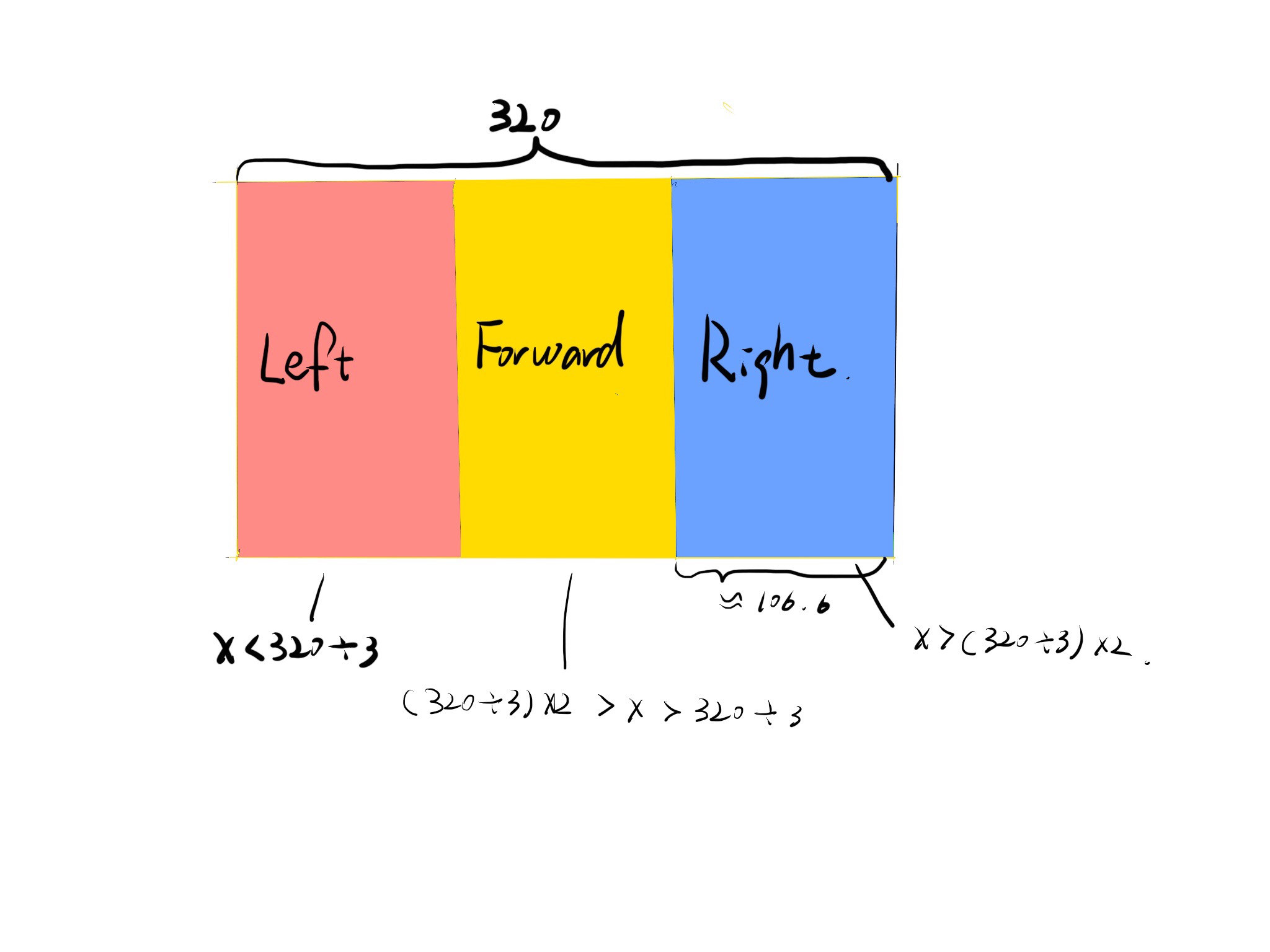
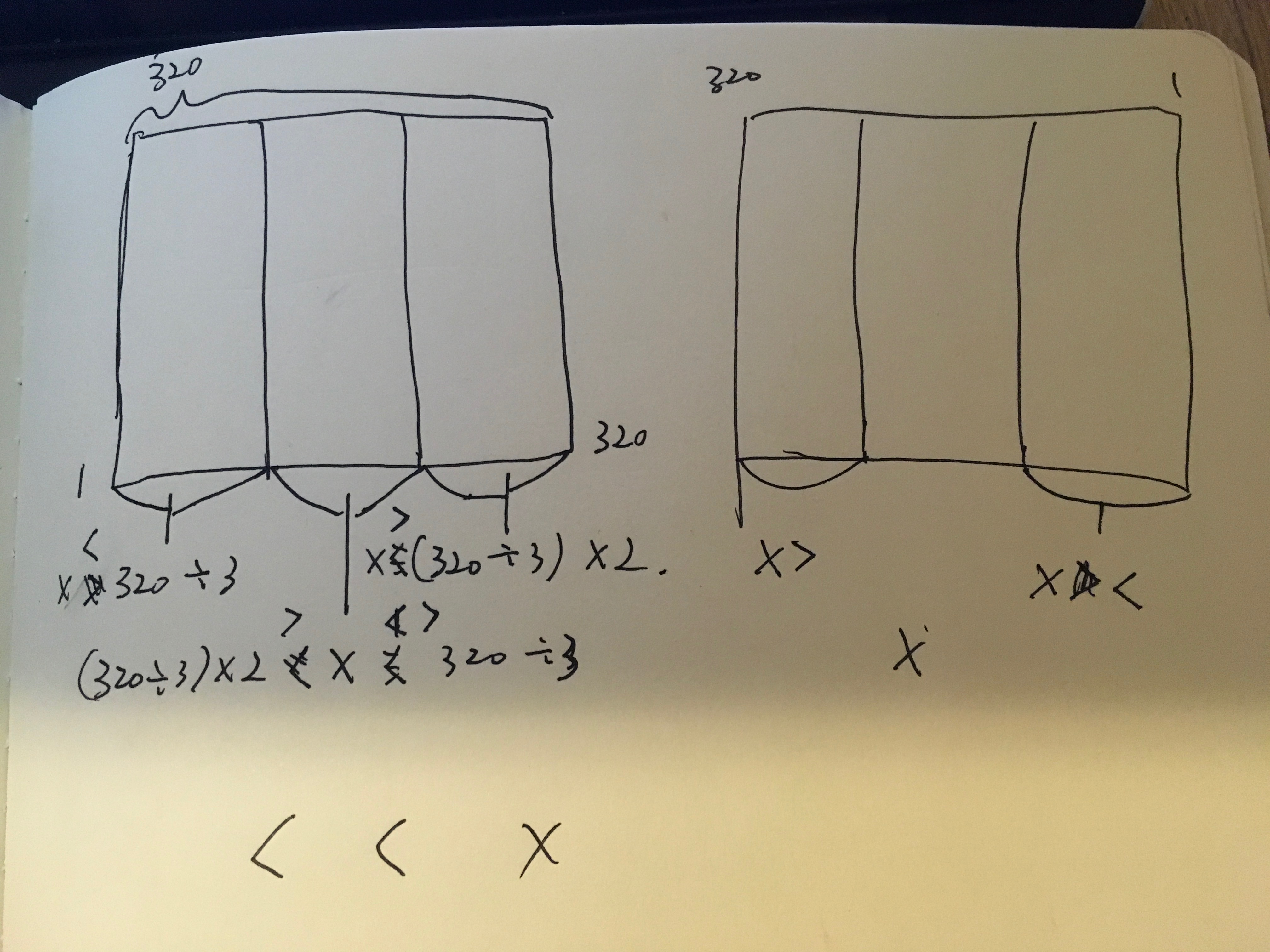
digitalWrite(rightForward, LOW);

digitalWrite(rightBackward, LOW);

delay(1);

Serial.println("left");

}

**Later**, I add a new function that can make the robot go straight forward. By decided the camera in to three parts.

The X range of the camera on Pixy CMUcam5 Sensor is 320. I did some mathematic to calculate the range of each action, then change the code, make the robot be able to go forward when the object appear in the middle of the camera.

**The final code is:**

//

// begin license header

//

// This file is part of Pixy CMUcam5 or "Pixy" for short

//

// All Pixy source code is provided under the terms of the

// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).

// Those wishing to use Pixy source code, software and/or

// technologies under different licensing terms should contact us at

// cmucam@cs.cmu.edu. Such licensing terms are available for

// all portions of the Pixy codebase presented here.

//

// end license header

//

// This sketch is a good place to start if you're just getting started with

// Pixy and Arduino. This program simply prints the detected object blocks

// (including color codes) through the serial console. It uses the Arduino's

// ICSP port. For more information go here:

//

// http://cmucam.org/projects/cmucam5/wiki/Hooking\_up\_Pixy\_to\_a\_Microcontroller\_(like\_an\_Arduino)

//

// It prints the detected blocks once per second because printing all of the

// blocks for all 50 frames per second would overwhelm the Arduino's serial port.

//

#include <SPI.h>

#include <Pixy.h>

// This is the main Pixy object

Pixy pixy;

int objectX;

int screenWidth = 320;

int inPin = 7; // choose the input pin (for a pushbutton)

int val = 0; // variable for reading the pin status

const int leftForward = 4;

const int leftBackward = 2;

const int rightForward = 5;

const int rightBackward = 3;

void setup()

{

Serial.begin(9600);

Serial.print("Starting...\n");

pixy.init();

// put your setup code here, to run once:

pinMode(2, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(3, OUTPUT);

pinMode(inPin, INPUT); // declare pushbutton as input

}

void loop()

{

static int i = 0;

int j;

uint16\_t blocks;

char buf[32];

// grab blocks!

blocks = pixy.getBlocks();

// If there are detect blocks, print them!

if (blocks >= 1)

{

i++;

// do this (print) every 50 frames because printing every

// frame would bog down the Arduino

if (i % 50 == 0)

{

sprintf(buf, "Detected %d:\n", blocks);

Serial.print(buf);

for (j = 0; j < blocks; j++)

{

sprintf(buf, " block %d: ", j);

Serial.print(buf);

Serial.print("J=");

Serial.println(j);

pixy.blocks[j].print();

objectX = pixy.blocks[j].x;

//determine direction, divide the camera in to three parts

// Object appear in the right part turn right, left part turn left

// Object appear in the middle part go straight.

if (objectX > ((screenWidth / 3) \* 2)) {

turnRight();

} else if (objectX < ( screenWidth / 3)) {

turnLeft();

}

else if (objectX > ( screenWidth / 3) && (objectX < ((screenWidth / 3) \* 2))) {

forward();

}

Serial.println(pixy.blocks[j].signature);

}

}

}

}

//turn right

void turnRight() {

digitalWrite(leftBackward, LOW);

digitalWrite(leftForward, LOW);

digitalWrite(rightForward, HIGH);

digitalWrite(rightBackward, LOW);

delay(1);

Serial.println("right");

}

// turn left

void turnLeft() {

digitalWrite(leftBackward, LOW);

digitalWrite(leftForward, HIGH);

digitalWrite(rightForward, LOW);

digitalWrite(rightBackward, LOW);

delay(1);

Serial.println("left");

}

// go straight!

void forward() {

digitalWrite(leftBackward, LOW);

digitalWrite(leftForward, HIGH);

digitalWrite(rightForward, HIGH);

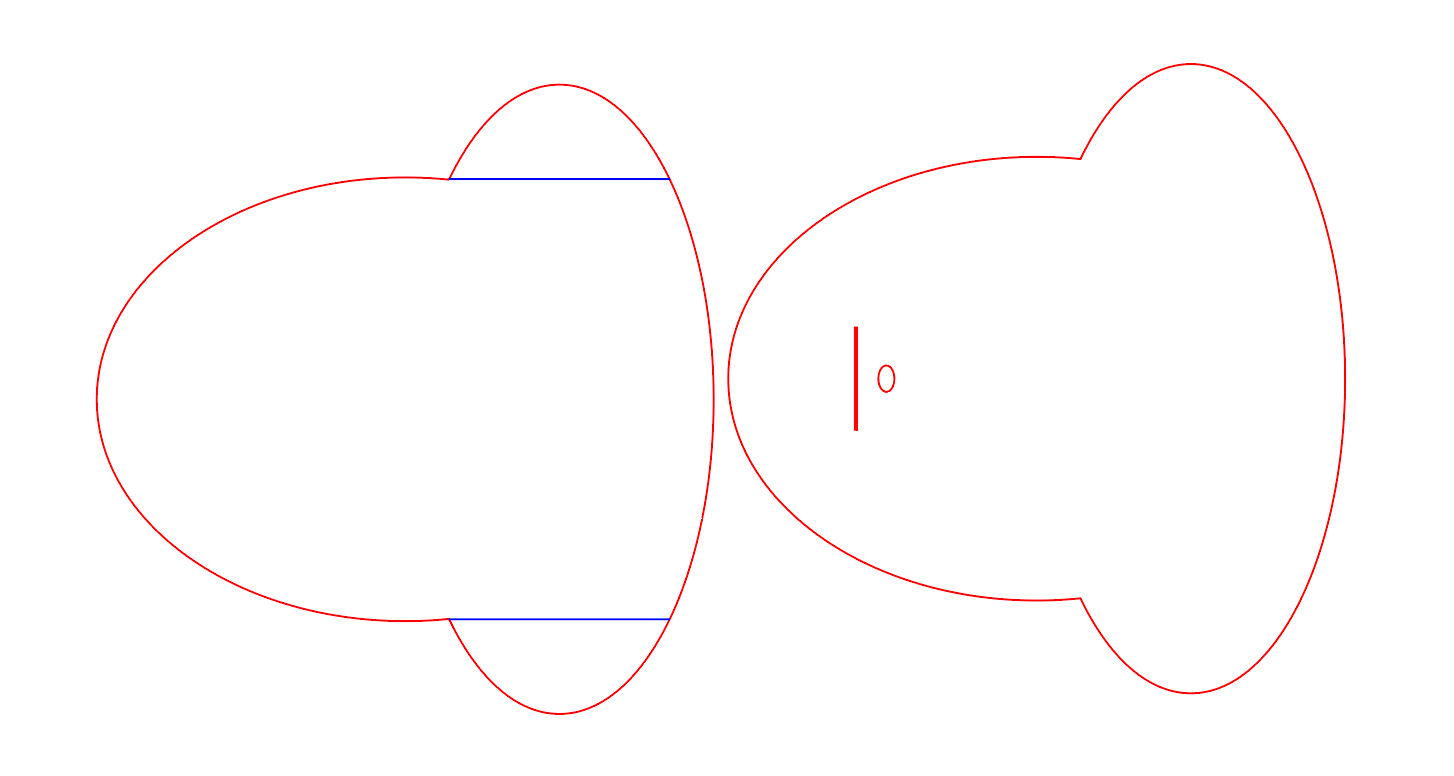
digitalWrite(rightBackward, LOW);

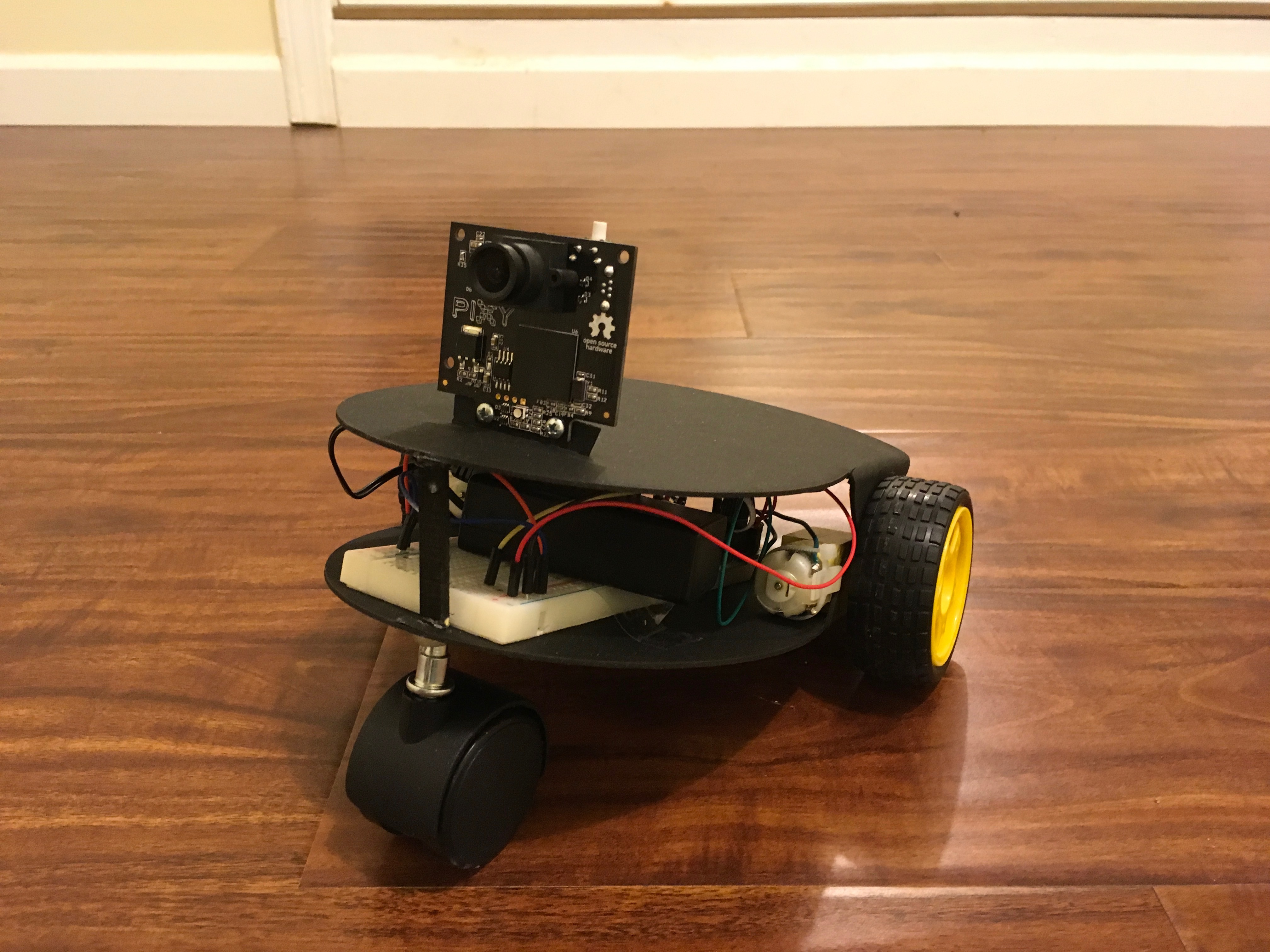
delay(1);

Serial.println("forward");

}

Then I use cardboard to laser cut the robot’s body, put the camera on the top, make it stable, and add the third wheel in the front.



Finally, the robot looks like this:

