Intelligent Systems (AI I):
Lab 1: Using Robots for AI

Eduardo Larios Fernández (A00569364)
Manuel Alejandro Lopez Perez (A01208598)
José Ramón Romero Chávez (A01700318)

Tecnológico de Monterrey, Campus Querétaro
29th January 2019

Introduction

Computer programming is the process of designing and building an executable computer program for accomplishing a specific computing task. Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms in a chosen programming language (commonly referred to as coding).

In this report we will take a close look into the development and construction of a physical agent implementing very rudimentary algorithms for autonomous operation. This agent will be capable of drawing several simple shapes, and depending on its sensors input, be able to change between different draw modes.

Development:

```
//Declares components to be used by the robot
distancebk sensor (PORT4);
motorbk motor1 (PORT2);
motorbk motor2 (PORT6);
ledsbk light (PORT7);

// These values were calculated by measuring the robot’s movement
// speed
int degree90 = 918;
int degree60 = 1224;
int flag = -1;
int d = 2;
void draw(){
```

```
  // Moves on a straight line indefinitely
  motor1.set(LEFT);
  motor2.set(RIGHT);
  delay(2000);

  // The base value on the sensor draw a triangle
  if(flag){
      light.color(BLUE);
      motor1.set(LEFT);
      motor2.set(LEFT);
      delay(degree60);
  }

  // The other possible value draws a square
  else{
      light.color(RED);
      motor1.set(LEFT);
      motor2.set(LEFT);
      delay(degree90);
  }

  // If the sensor gets a reading (it detects something closer than
  // 16cm) it changes the operation mode
  if(sensor.read() >= 0 && sensor.read() <= 16){
      flag = flag * -1;
  }
}

code(){
  // Initializes the robot
  draw();
}
```

Video Result: https://youtu.be/R434LEh-S1s

Limitations

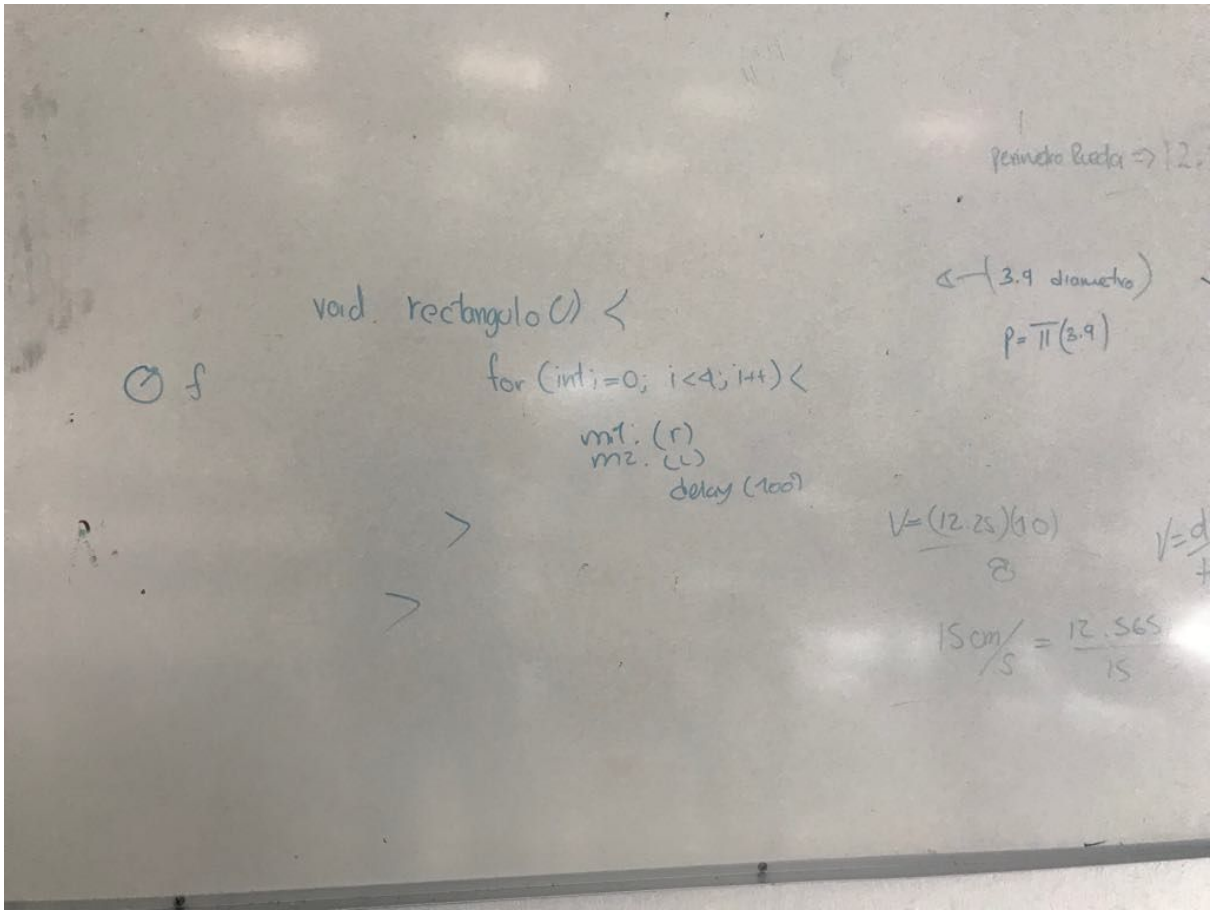
Our lab is tailored to the limitation we have through the development of this project. Our first obstacle was how to integrate multithreaded functions to the parallax. This approach was necessary because one of our main functions was an atomic function that needed to be stopped via another.

1. We had a faulty motor that was slower than the other; and there was no way to calibrate correctly with a faulty motor.

We decided to change to a Bricko and implement the main example of the lab description; the bricko also presented challenges that we resolved.

1. Version of the Bricko
    a. The version of the Bricko we had was limited to the most basic function of a bricko (no moving via distance or rotation) we only had how to set velocities for the two motors.
2. Uneven Tires
    a. Our calculation are tailored only to our setup of the Bricko since we have unevenness that we had to factor in.
3. Sensors Misfires
    a. Even Though this didn't affect that much the project; but we have to be aware that the sensor misfires sometimes.

Calculation for 90 degree rotations



Perimetro Rueda => 12.25 cm.

(3.9 diametro)

$P = \pi(3.9)$

16 cm diametro Robot

10 vueltas / 8 seg.

$V = \frac{(12.25)(10)}{8}$

$15\ cm/s = \frac{12.565}{15}$

$V = \frac{d}{t}$ $V = \frac{12.565}{x}$

> 50.26 cm / 4 =

12.565 cm de rotacion

diametro Robot

> 50.26 cm / 4 =

12.565 cm de rotacion

9/18 → 90°

Improvements/Modifications

For real applications and improvement about this and others applications, it is suggested to do changes especially in hardware and the software that is embedded in them.

Improvements in this area may allow the following features:

- Use of thread to manage parallel events and actions.
- Proper modulation of modern design patterns in code
- Improve the kind and quality of sensors used (i.e. sonic, HySpex cameras)

With improvements like this, it will be possible to develop any application, some of them may be in the manufacturing industries or more qualified areas and applications, such as robotics vision.