REPORTE PROYECTO:

"CARRO LABERINTO"

Omar I. Vizcaino Alarcon 24000207

Axel Arriola Fonseca 5802023

Karime García Xalteno 24100349

Microprocesadores

Puebla, Pue. A 21 de mayo de 2019

Introducción:

En este proyecto se realizó un carrito de 3 ruedas con 3 sensores ultrasónicos y una cámara Web. Con la finalidad de que resolviera un laberinto por sí solo, sin embargo por falta de tiempo se configuró para poderlo controlar remotamente, sin embargo los sensores deben marcar su funcionamiento y la cámara Web debía mostrar la imagen en tiempo real.

Material:

- Raspberry Pi 3 (cables)
- Computadora o algún visualizador para la Raspberry
- 2 motores
- Carcasa para carrito
- 3 ruedas
- 3 sensores ultrasónicos
- Camara Web
- Puente H

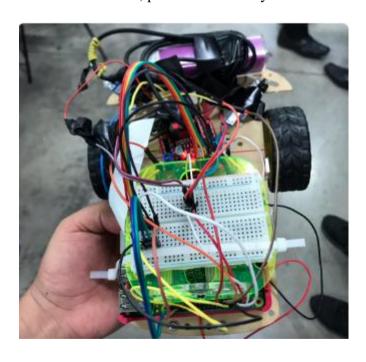
Procedimiento:

- Primero se creó el programa Glade, introduciendo primero una etiqueta "Laberinto", luego 7 botones, "izquierda" "derecha" "adelante" "atrás" "detener" "sensores" y "cámara", finalmente se agregaron 3 etiquetas mas "sensor izquierda" "sensor delantero" "sensor izquierda".
- Después se armó el carrito, conectando los motores al puente de pilas, conectando el puente H con los sensores y los motores y conectando la camara web.
- Se utilizaron los pines basicos de GPIO y Wiring.Pi.
- Posteriormente se creó el programa en Geany, primero asimilando los botones de movimiento para los sensores, poniendo condiciones para que cuando se pulsara el botón izquierda, los motores generan una rotación a la izquierda y así con los demás botones hasta el botón de "detener".

- Para el botón de "sensores" de igual modo se establecieron condiciones para que cuando el sensor derecho fuera activado, en la pantalla de Glade arroja una baja, de este modo en la pantalla se podía ver hacia qué dirección el carrito se podía mover, es decir, si la pared del laberinto tiene libre hacia el frente, en la pantalla de Glade se vería que los sensores derecho e izquierdo están activados, por lo que el delantero esta desactivado y así el carrito solo se puede mover hacia al frente o tal vez hacia atrás.
- Finalmente se vinculó la cámara mediante VLC y el protocolo para poder hacer streaming desde la aplicación y con solo oprimir el botón en la pantalla Glade, se podía ver en la computadora o en el celular.

Resultados:

Para llegar a los resultados correcto tuvimos que pasar por varias pruebas y problemas, sin embargo el programa funcionó. El carrito se iba un poco hacia la derecha pero se soluciono. Sin embargo tuvimos un problema con la Raspberry y no pudimos configurar de ningún modo la camara Web, pero los sensores y el movimiento del carrito no tuvo problemas.



Conclusión:

Para este proyecto final, se aplicaron todos los conocimientos vistos a lo largo del curso, desde un gráfico básico de Glade, hasta el movimiento mediante Wiring.Pi con GPIO para los motores y condiciones para los sensores, lo nuevo en este caso fue la instalación de la cámara Web. Fue muy útil hacer este proyecto que engloba todo lo visto en el curso.

```
Anexos:
#include <stdio.h>
#include <stdlib.h>
#include <gtk/gtk.h>
#include <wiringPi.h>
GtkWidget *g bt left;
GtkWidget *g bt backward;
GtkWidget *g bt forward;
GtkWidget *g bt right;
GtkWidget *g bt stop;
GtkWidget *g_lbl_front;
GtkWidget *g lbl left;
GtkWidget *g_lbl_right;
int main(int argc, char *argv[])
{
  GtkBuilder
                *builder;
```

```
GtkWidget
              *window;
gtk init(&argc, &argv);
builder = gtk builder new();
gtk builder add from file(builder, "Control.glade", NULL);
window = GTK WIDGET(gtk builder get object(builder, "Ventana"));
gtk builder connect signals(builder, NULL);
g bt left = GTK WIDGET(gtk builder get object(builder, "bt left"));
g bt backward = GTK WIDGET(gtk builder get object(builder, "bt backward"));
g_bt_forward = GTK_WIDGET(gtk_builder_get_object(builder, "bt_forward"));
g_bt_right = GTK_WIDGET(gtk_builder_get_object(builder, "bt_right"));
g bt stop = GTK WIDGET(gtk builder get object(builder, "bt stop"));
g lbl front = GTK WIDGET(gtk builder get object(builder, "lbl front"));
g_lbl_right = GTK_WIDGET(gtk_builder_get_object(builder, "lbl_right"));
g lbl left = GTK WIDGET(gtk builder get object(builder, "lbl left"));
g object unref(builder);
gtk_widget_show(window);
gtk main();
```

```
return 0;
}
void on_bt_left_clicked()
{
  if(wiringPiSetup() == -1)
  {
      return 1;
  }
  /Motor A/
  pinMode(15, OUTPUT);
  pinMode(16, OUTPUT);
  pinMode(1, OUTPUT);
  /Motor B/
  pinMode(27, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(29, OUTPUT);
  digitalWrite(15, HIGH);
  digitalWrite(16, LOW);
  digitalWrite(1, HIGH);
```

```
delay(50);
  digitalWrite(27, HIGH);
  digitalWrite(28, HIGH);
  digitalWrite(29, LOW);
  delay(50);
}
void on_bt_backward_clicked()
{
  if(wiringPiSetup() == -1)
  {
      return 1;
  }
  /Motor A/
  pinMode(15, OUTPUT);
  pinMode(16, OUTPUT);
  pinMode(1, OUTPUT);
  /Motor B/
  pinMode(27, OUTPUT);
  pinMode(28, OUTPUT);
```

```
pinMode(29, OUTPUT);
  digitalWrite(15, HIGH);
  digitalWrite(16, LOW);
  digitalWrite(1, HIGH);
  delay(50);
  digitalWrite(27, HIGH);
  digitalWrite(28, LOW);
  digitalWrite(29, HIGH);
  delay(50);
void on_bt_forward_clicked()
  if(wiringPiSetup() == -1)
       {
      return 1;
       }
```

}

{

```
/Motor A/
      pinMode(15, OUTPUT);
      pinMode(16, OUTPUT);
      pinMode(1, OUTPUT);
      /Motor B/
      pinMode(27, OUTPUT);
      pinMode(28, OUTPUT);
      pinMode(29, OUTPUT);
      digitalWrite(15, HIGH);
      digitalWrite(16, HIGH);
      digitalWrite(1, LOW);
      delay(50);
      digitalWrite(27, HIGH);
      digitalWrite(28, HIGH);
      digitalWrite(29, LOW);
      delay(50);
void on_bt_right_clicked()
```

}

{

```
if(wiringPiSetup() == -1)
{
    return 1;
}
/Motor A/
pinMode(15, OUTPUT);
pinMode(16, OUTPUT);
pinMode(1, OUTPUT);
/Motor B/
pinMode(27, OUTPUT);
pinMode(28, OUTPUT);
pinMode(29, OUTPUT);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
digitalWrite(1, LOW);
delay(50);
digitalWrite(27, HIGH);
digitalWrite(28, LOW);
digitalWrite(29, HIGH);
```

```
delay(50);
}
void on_bt_stop_clicked()
{
  if(wiringPiSetup() == -1)
      return 1;
      }
      /Motor A/
      pinMode(15, OUTPUT);
      pinMode(16, OUTPUT);
      pinMode(1, OUTPUT);
      /Motor B/
      pinMode(27, OUTPUT);
      pinMode(28, OUTPUT);
      pinMode(29, OUTPUT);
      digitalWrite(15, LOW);
      delay(5);
```

```
digitalWrite(16, LOW);
       delay(5);
       digitalWrite(1, LOW);
       delay(5);
       digitalWrite(27, LOW);
       delay(5);
       digitalWrite(28, LOW);
       delay(5);
       digitalWrite(29, LOW);
       delay(5);
}
int on_bt_auto_clicked()
{
  if(wiringPiSetup() == -1)
       {
       return 1;
       int loop = 0;
       int sI = 0;
```

```
int sF = 0;
int sD = 0;
/Sensores/
pinMode(0, INPUT);
pinMode(2, INPUT);
pinMode(3, INPUT);
/Motores/
/Motor A/
pinMode(15, OUTPUT);
pinMode(16, OUTPUT);
pinMode(1, OUTPUT);
/Motor B/
pinMode(27, OUTPUT);
pinMode(28, OUTPUT);
pinMode(29, OUTPUT);
do
for(;;)
```

```
if(digitalRead(0) == LOW)
                                          {
                                                 printf("No Funcionando Izquierda\n");
                                                 sI = 0;
                                                 if(digitalRead(2) == LOW)
                                                 {
                                                        printf("No
                                                                           Funcionando
Frente\n");
                                                        sF = 0;
                                                        if(digitalRead(3) == LOW)
                                                        {
                                                               printf("No
                                                                           Funcionando
Derecha\n");
                                                               sD = 0;
                                                               break;
                                                        }
                                                 }
                                          }
                                          if(digitalRead(0) == LOW)
```

```
{
                                                 printf("No Funcionando Izquierda\n");
                                                sI = 0;
                                                 if(digitalRead(2) == HIGH)
                                                 {
                                                        printf("No
                                                                           Funcionando
Frente\n");
                                                        sF = 1;
                                                        if(digitalRead(3) == LOW)
                                                        {
                                                               printf("No Funcionando
Derecha\n");
                                                               sD = 0;
                                                               break;
                                                        }
                                                 }
                                          }
                                         if(digitalRead(0) == LOW)
                                          {
```

```
printf("No Funcionando Izquierda\n");
                                                sI = 0;
                                                 if(digitalRead(2) == LOW)
                                                 {
                                                        printf("No
                                                                           Funcionando
Frente\n");
                                                        sF = 0;
                                                        if(digitalRead(3) == HIGH)
                                                        {
                                                              printf("No Funcionando
Derecha\n");
                                                              sD = 1;
                                                              break;
                                                        }
                                                }
                                          }
                                         if(digitalRead(0) == LOW)
                                          {
```

```
printf("No Funcionando Izquierda\n");
                                                 sI = 0;
                                                 if(digitalRead(2) == HIGH)
                                                 {
                                                        printf("No
                                                                            Funcionando
Frente\n");
                                                        sF = 1;
                                                        if(digitalRead(3) == HIGH)
                                                        {
                                                               printf("No Funcionando
Derecha\n");
                                                               sD = 1;
                                                               break;
                                                        }
                                                 }
                                          }
                                          if(digitalRead(0) == HIGH)
                                          {
```

```
printf("Funcionando Izquierda\n");
                                                  sI = 1;
                                                  if(digitalRead(2) == LOW)
                                                  {
                                                         printf("Funcionando Frente\n");
                                                         sF = 0;
                                                         if(digitalRead(3) == LOW)
                                                         {
                                                                printf("Funcionando
Derecha\n");
                                                                sD = 0;
                                                                break;
                                                         }
                                                  }
                                           }
                                           if(digitalRead(0) == HIGH)
                                           {
                                                  printf(...
```