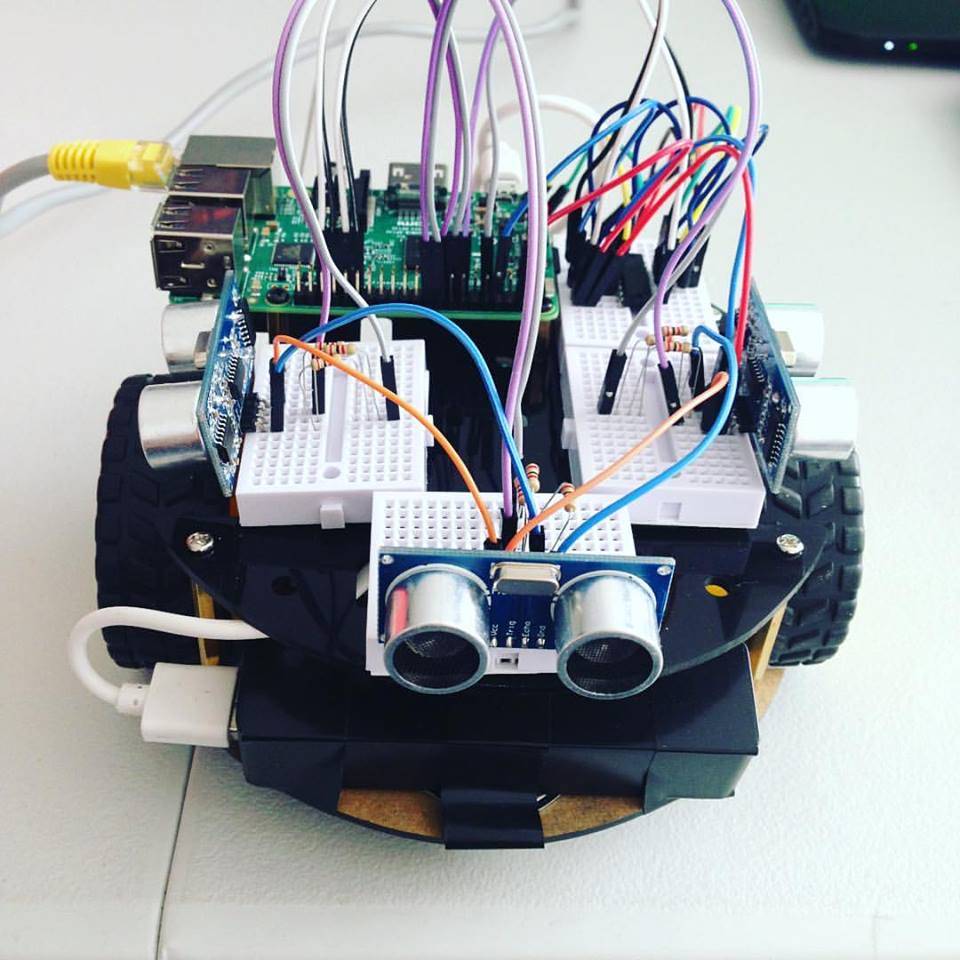


##### **Compiladores y Desarrollo de Librerías**

***Profesor:* Alí Piña**

**Alumno César Augusto Vargas Torres**

**Proyecto Final: Robot Evasor de Obstáculos**



**Robot Evasor de obstáculos**

**Objetivo**

Desarrollar un programa en lenguaje C para la tarjeta de desarrollo Raspberry Pi 3 que controle un robot evasor de obstáculos.

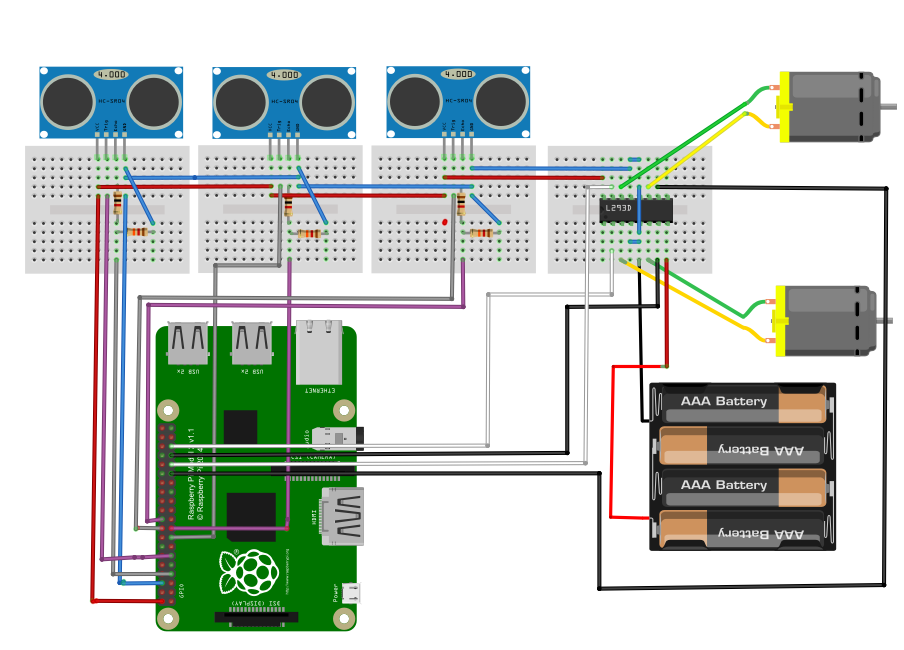
**Materiales**

* 2 Baterías 5V a 1A
* Cables Dupont
* 1 Puente H L293D
* 3 Sensores Ultrasónicos
* 3 Protos 170 puntos
* 9 Resistencias 2 k ohm
* 2 Rueda con reductor
* Juego de tornillos
* 2 Bases de acrílico

**Desarrollo**

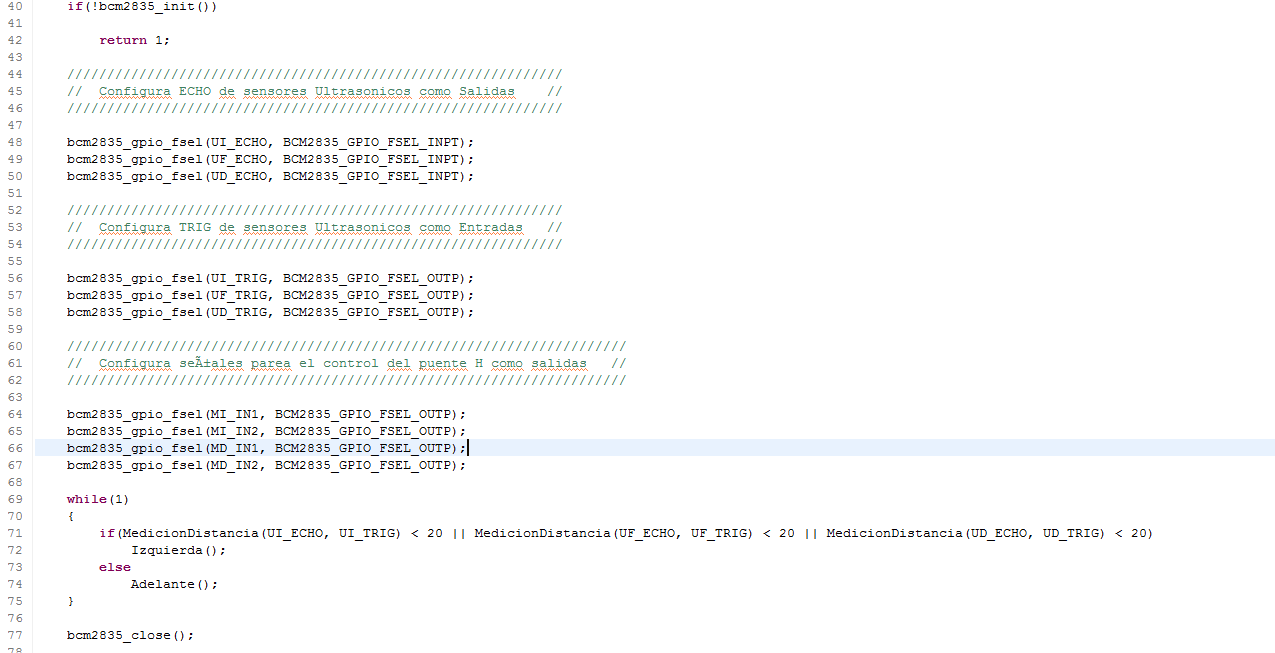
* Ensamblar chasis y mecanismos
* Ensamblar circuito eléctrico
* Escribir el código
* Pruebas y depuración

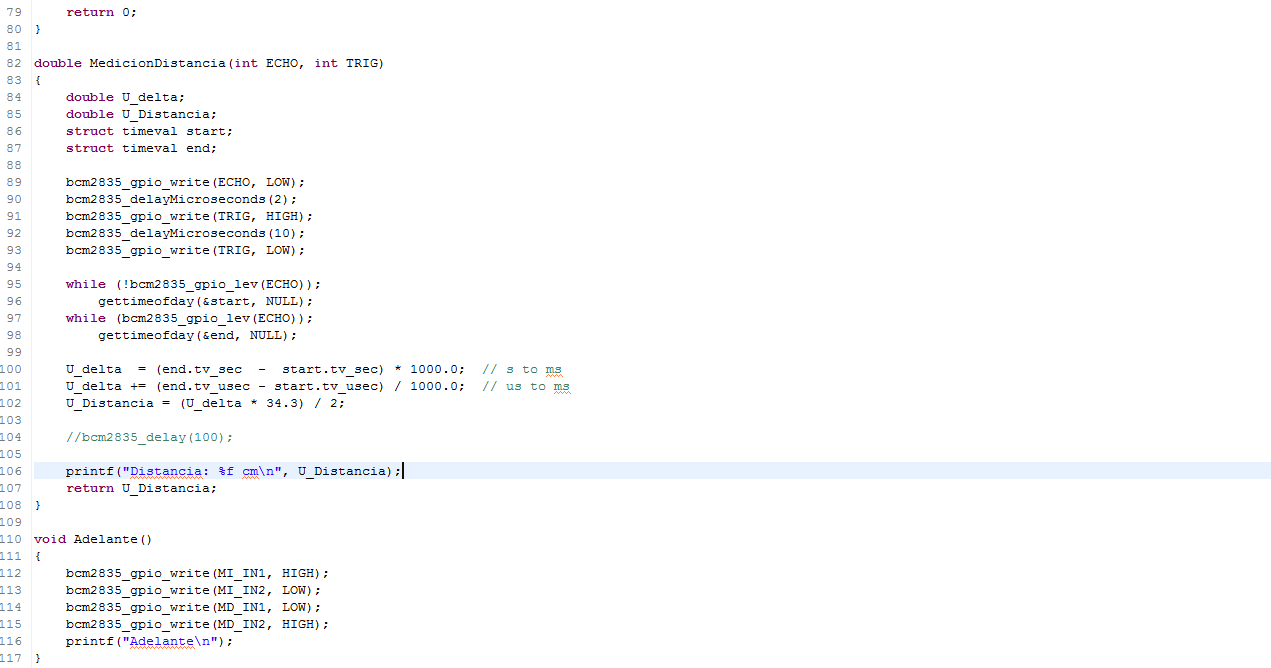
**Diagrama Eléctrico**

****

**Código**

****

****

****

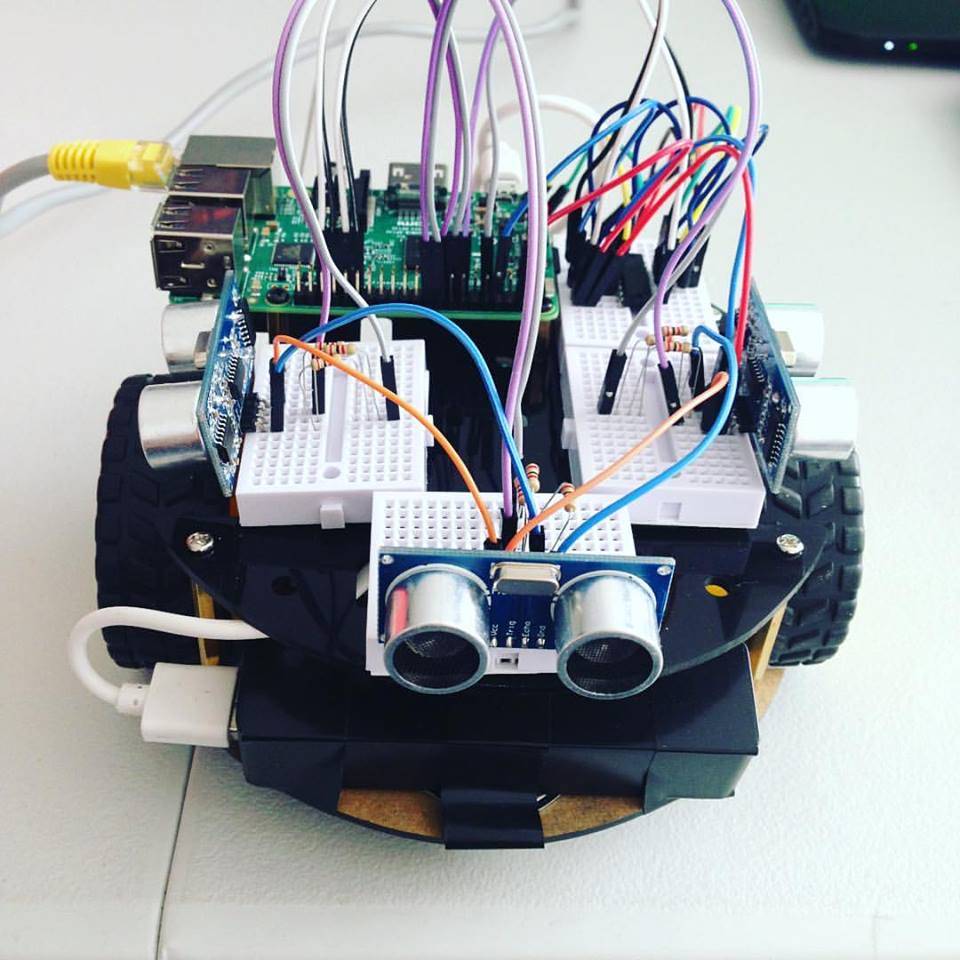
****

**Conclusiones**

Fue bastante sencillo hacer el control de los sensores y los actuadores usando la librería de bcm2835, no cabe duda que el lenguaje C es bastante útil para este tipo de aplicaciones. El algoritmo que se utilizo es bastante sencillo, pero aprovechando la plataforma sería muy sencillo implementar alguno mas robusto.

**Anexos**

**Imágenes**



**Código**

/\*

\* main.c

\*

\* Created on: Jun 10, 2016

\* Author: cesar

\*/

**# include** <bcm2835.h>

**# include** <stdio.h>

**# include** <time.h>

**# include** <stdlib.h>

//////////////////////////////////////

// SeÃ±ales Sensores Ultrasonicos //

//////////////////////////////////////

**#define** UI\_ECHO 4

**#define** UI\_TRIG 17

**#define** UF\_ECHO 23

**#define** UF\_TRIG 24

**#define** UD\_ECHO 27

**#define** UD\_TRIG 22

//////////////////////////

// SeÃ±ales Motores //

//////////////////////////

**#define** MI\_IN1 5

**#define** MI\_IN2 6

**#define** MD\_IN1 13

**#define** MD\_IN2 19

**double** MedicionDistancia(**int** Echo, **int** Trig);

**void** Adelante();

**void** Izquierda();

**int** main(**int** argc, **char** \*\*argv)

{

**if**(!bcm2835\_init())

**return** 1;

//////////////////////////////////////////////////////////////

// Configura ECHO de sensores Ultrasonicos como Salidas //

//////////////////////////////////////////////////////////////

bcm2835\_gpio\_fsel(UI\_ECHO, BCM2835\_GPIO\_FSEL\_INPT);

bcm2835\_gpio\_fsel(UF\_ECHO, BCM2835\_GPIO\_FSEL\_INPT);

bcm2835\_gpio\_fsel(UD\_ECHO, BCM2835\_GPIO\_FSEL\_INPT);

//////////////////////////////////////////////////////////////

// Configura TRIG de sensores Ultrasonicos como Entradas //

//////////////////////////////////////////////////////////////

bcm2835\_gpio\_fsel(UI\_TRIG, BCM2835\_GPIO\_FSEL\_OUTP);

bcm2835\_gpio\_fsel(UF\_TRIG, BCM2835\_GPIO\_FSEL\_OUTP);

bcm2835\_gpio\_fsel(UD\_TRIG, BCM2835\_GPIO\_FSEL\_OUTP);

//////////////////////////////////////////////////////////////////////

// Configura seÃ±ales parea el control del puente H como salidas //

//////////////////////////////////////////////////////////////////////

bcm2835\_gpio\_fsel(MI\_IN1, BCM2835\_GPIO\_FSEL\_OUTP);

bcm2835\_gpio\_fsel(MI\_IN2, BCM2835\_GPIO\_FSEL\_OUTP);

bcm2835\_gpio\_fsel(MD\_IN1, BCM2835\_GPIO\_FSEL\_OUTP);

bcm2835\_gpio\_fsel(MD\_IN2, BCM2835\_GPIO\_FSEL\_OUTP);

**while**(1)

{

**if**(MedicionDistancia(UI\_ECHO, UI\_TRIG) < 20 || MedicionDistancia(UF\_ECHO, UF\_TRIG) < 20 || MedicionDistancia(UD\_ECHO, UD\_TRIG) < 20)

Izquierda();

**else**

Adelante();

}

bcm2835\_close();

**return** 0;

}

**double** MedicionDistancia(**int** ECHO, **int** TRIG)

{

**double** U\_delta;

**double** U\_Distancia;

**struct** timeval start;

**struct** timeval end;

bcm2835\_gpio\_write(ECHO, LOW);

bcm2835\_delayMicroseconds(2);

bcm2835\_gpio\_write(TRIG, HIGH);

bcm2835\_delayMicroseconds(10);

bcm2835\_gpio\_write(TRIG, LOW);

**while** (!bcm2835\_gpio\_lev(ECHO));

gettimeofday(&start, NULL);

**while** (bcm2835\_gpio\_lev(ECHO));

gettimeofday(&end, NULL);

U\_delta = (end.tv\_sec - start.tv\_sec) \* 1000.0; // s to ms

U\_delta += (end.tv\_usec - start.tv\_usec) / 1000.0; // us to ms

U\_Distancia = (U\_delta \* 34.3) / 2;

//bcm2835\_delay(100);

printf("Distancia: %f cm\n", U\_Distancia);

**return** U\_Distancia;

}

**void** Adelante()

{

bcm2835\_gpio\_write(MI\_IN1, HIGH);

bcm2835\_gpio\_write(MI\_IN2, LOW);

bcm2835\_gpio\_write(MD\_IN1, LOW);

bcm2835\_gpio\_write(MD\_IN2, HIGH);

printf("Adelante\n");

}

**void** Izquierda()

{

bcm2835\_gpio\_write(MI\_IN1, HIGH);

bcm2835\_gpio\_write(MI\_IN2, LOW);

bcm2835\_gpio\_write(MD\_IN1, HIGH);

bcm2835\_gpio\_write(MD\_IN2, LOW);

printf("Izquierda\n");

}