Instituto Politécnico Nacional

Escuela Superior de Cómputo

Departamento de Ciencias e Ingeniería de la Computación

Sistemas en Chip

Proyecto 01

"Robot"

Profesor: Fernando Aguilar Sánchez

Grupo: 6CM1

Equipo 3

Alumnos:

Ocampo Téllez Rodolfo

Patlani Mauricio Adriana

Ruvalcaba Flores Martha Catalina

Sandoval Hernández Eduardo

Fecha de entrega: 23/12/2022

Objetivo

Al término de la sesión, los integrantes del equipo contaran con la habilidad de realizar un contador sin rebotes de 00 a 99 mostrado en un par de displays activado con infrarrojo.

Introducción Teórica

El microcontrolador ATMEGA8535 fabricante ATMEL. En la figura 1 se muestra el microcontrolador ATMEGA8535 la cual maneja datos de 8 bits es decir su bus de datos de 8 bits. Aunque este contiene tres registros los cuales son el *x*, *y* y *z*, los cuales manejan datos de 16 bits.

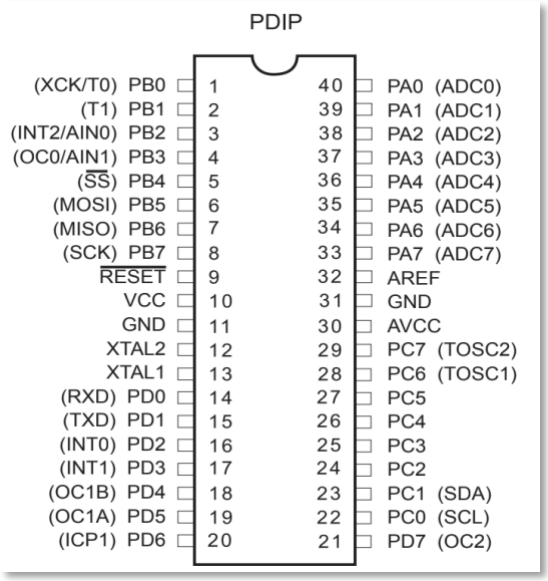

Figura 1. Configuración de pines ATmega8535

La comunicación interna del microcontrolador se categoriza en 4 puertos, en la figura 1 se puede analizar la etiquetación de los puertos PA0 al PA7, PB0 al PB7, PC0 al PC7 y PD0 al PD7. Cabe recordar que maneja datos de 8 bits.

Posee un oscilador interno de 1MHz, sin embargo, como es un oscilador RC, es susceptible a variar la frecuencia con respecto a las variaciones de temperatura. Por otro lado, puede conectarse un oscilador de 0 a 8 MHz o de 0 a 16 MHz.

Puente H

El puente H juega un papel muy importante en el área de la robótica y la automatización. Invertir el giro de un motor es posible gracias a dispositivos semiconductores y contactores eléctricos. El puente H es un mecanismo electrónico que se encarga de invertir el giro de un motor usando un elemento básico en electrónica como lo es el transistor.

Este versátil dispositivo tiene la habilidad de comportarse como un interruptor electrónico y como un amplificador. Cómo se hace necesario su uso para invertir el giro de un motor, solo para este caso se utilizará como un interruptor.

Su nombre proviene de su estructura de 4 interruptores en forma de letra H y un motor en el centro tal como se muestra en la figura 2.

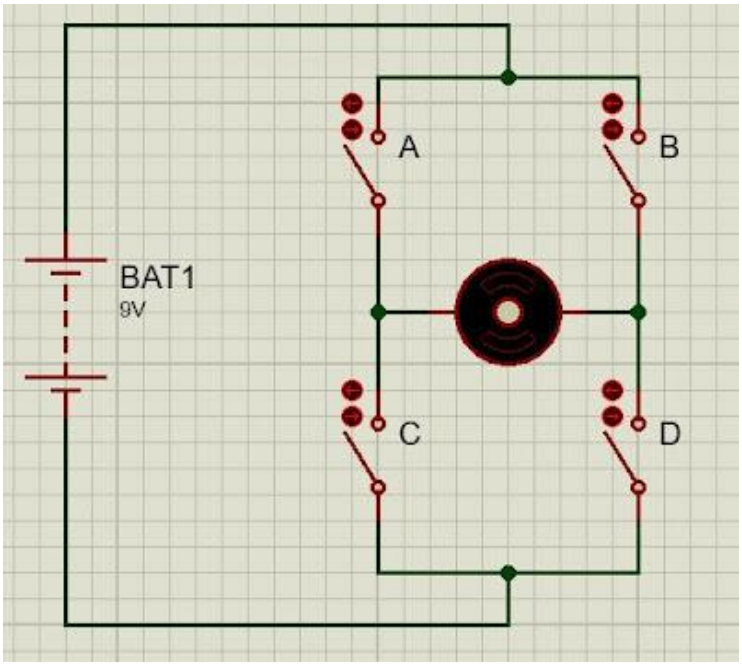

Figura 2. Esquema del Puente H.

Cuando se requiere que el motor que está ubicado en el centro de dicho circuito se mueva hacia la derecha es decir que tenga un giro horario normal, son los interruptores A y D los que se necesitan activar. Si asumimos que el motor tiene polaridad, de + a – y activamos los interruptores A y D la corriente circundante será de menos a más por la batería por lo cual circulará la corriente por el interruptor A, pasando por la polaridad positiva del motor, luego trasladándose hacia la polaridad negativa y por ende por el interruptor D habrá circulación de corriente por lo cual ese será el circuito o la malla circundante de corriente. al haber diferencia de potencial entre las terminales positiva y negativa habrá fuerza contra electromotriz por el motor de corriente directa DC.

Por el contrario, cuando se desea que el motor gire en sentido antihorario se deben activar los interruptores B y C. Desactivando los interruptores A y D y activando los interruptores B y C se puede observar que la corriente no circulará por el polo positivo del motor sino por el polo negativo del mencionado, por lo cual la corriente entrará por el interruptor B y saldrá por el interruptor C y de esta manera habrá fuerza electromotriz en sentido anti-horario.

A	B	C	D	MOTOR
OFF	OFF	OFF	OFF	APAGADO
ON	OFF	OFF	ON	GIRO HORARIO
OFF	ON	ON	OFF	GIRO ANTIHORARIO
ON	ON	ON	ON	FRENADO

Tabla 1. Estados de funcionamiento del puente H.

LED Infrarrojo

Este tipo de LED forma parte de la familia de semiconductores de unión p-n y, cuando lo unión es polarizada en directa, los electrones de dicha región n se recombinan con los huecos excedentes del material p.

Para su funcionamiento se requiere de un LED infrarrojo Emisor y un Fotodiodo Receptor. un receptor infrarrojo opera como un transistor normal y corriente que cumple con las siguientes características.

- Si recibe luz infrarroja entra en saturación
- Cuando la luz es interrumpida entra en corte

Bajo las dos consideraciones anteriores, cuando el receptor está en corto debido a una interrupción en la transmisión infrarroja, tenemos prácticamente un circuito abierto. Hay una prueba muy fácil de hacer para comprobar si un LED infrarrojo funciona y, es ver tu componente a través de una cámara como la de tu celular, solo enfócalo y veras como se ilumina, aunque este método únicamente funciona para el emisor.

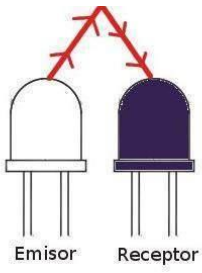

Figura 3. Par infrarrojo.

Desarrollo Experimental

1.- Diseñe un móvil controlado vía infrarrojo, solo controle movimiento como lo indican las flechas.

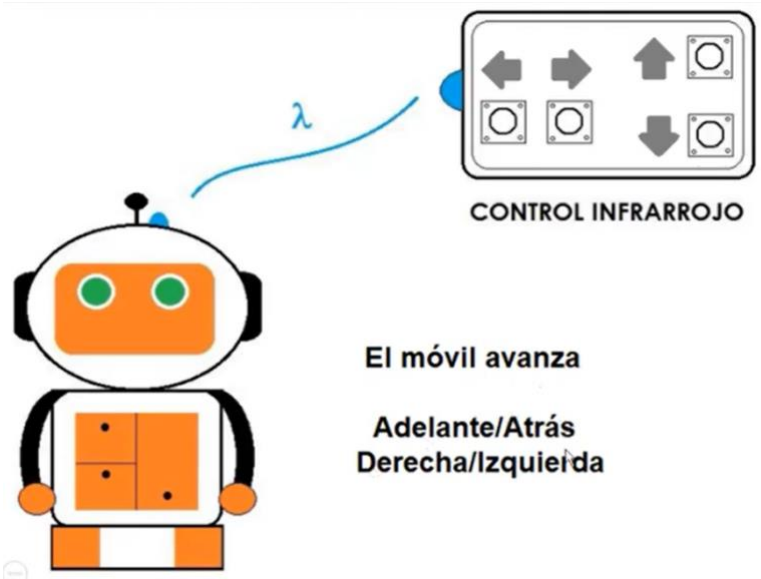

Figura 4. Esquema del proyecto.

Estructura del programa

Código de la configuración de los periféricos utilizados y código del programa principal en C, proporcionados por el IDE de CodeVision, correspondiente al **Atmega 1 (transmisor)** utilizado para el circuito.

```
#include <mega8535.h>

#include <delay.h>

#define Delante PINB.0

#define Reversa PINB.1

#define Izquierda PINB.2

#define Derecha PINB.3

#define Salida PORTA.0

void main(void)

{

// Input/Output Ports initialization

// Port A initialization

DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) |
(1<<DDA1) | (1<<DDA0);

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) |
(0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) |
(0<<DDB1) | (0<<DDB0);

PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) |
(1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);

// Port C initialization

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) |
(0<<DDC1) | (0<<DDC0);

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) |
(0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) |
(0<<DDD1) | (0<<DDD0);
```

```
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) |
(0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) |
(0<<CS00);

TCNT0=0x00;

OCR0=0x00;

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) |
(0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

ASSR=0<<AS2;

TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) |
(0<<CS20);

TCNT2=0x00;

OCR2=0x00;

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) |
(0<<OCIE0) | (0<<TOIE0);

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) |
(0<<RXB8) | (0<<TXB8);

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) |
(0<<ACIS0);
```

```
SFIOR=(0<<ACME);

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)
    {
    if(Delante == 0){
        Salida = 1;
        delay_ms(25);
        Salida = 0;
        delay_ms(475);
    } else if(Reversa == 0){
        Salida = 1;
        delay_ms(50);
        Salida = 0;
        delay_ms(450);
    } else if(Izquierda == 0){
        Salida = 1;
        delay_ms(75);
        Salida = 0;
        delay_ms(425);
    } else if(Derecha == 0){
        Salida = 1;
        delay_ms(100);
        Salida = 0;
        delay_ms(400); }
    }
```

}

Código de la configuración de los periféricos utilizados y código del programa principal en C, proporcionados por el IDE de CodeVision, correspondiente al **Atmega 2 (receptor)** utilizado para el circuito.

```
#include <mega8535.h>

#include <delay.h>

#define Delante 0x05

#define Reversa 0x0A

#define Izquierda 0x01

#define Derecha 0x04

#define Entrada PINB.0

int contador, i = 0;

void main(void)

{

// Input/Output Ports initialization

// Port A initialization

DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) |
(1<<DDA1) | (1<<DDA0);

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) |
(0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) |
(0<<DDB1) | (0<<DDB0);

PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) |
(1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);

// Port C initialization

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) |
(0<<DDC1) | (0<<DDC0);

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) |
(0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
```

```
// Port D initialization
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) |
(0<<DDD1) | (0<<DDD0);
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) |
(0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
…
…
…
while (1)
    {
    contador = 0;
    for(i = 0; i < 500; i++){
        if(Entrada == 1)
            contador++;
        delay_ms(1);
    }
    if(contador >= 15 && contador <= 35){
        PORTA = Delante;
    }else if(contador >= 40 && contador <= 60){
        PORTA = Reversa;
    }else if(contador >= 65 && contador <= 85){
        PORTA = Izquierda;
    }else if(contador >= 90 && contador <= 110){
        PORTA = Derecha;
    }else {
        PORTA = 0x00;
    }
    }
}
```

Simulaciones

La simulación se realizó en el software Proteus Design Suite, previamente cargado el programa en formato ".hex" obtenido del software CodeVision AVR, el cual es un software para el microcontrolador Microchip AVR y sus contrapartes XMEGA.

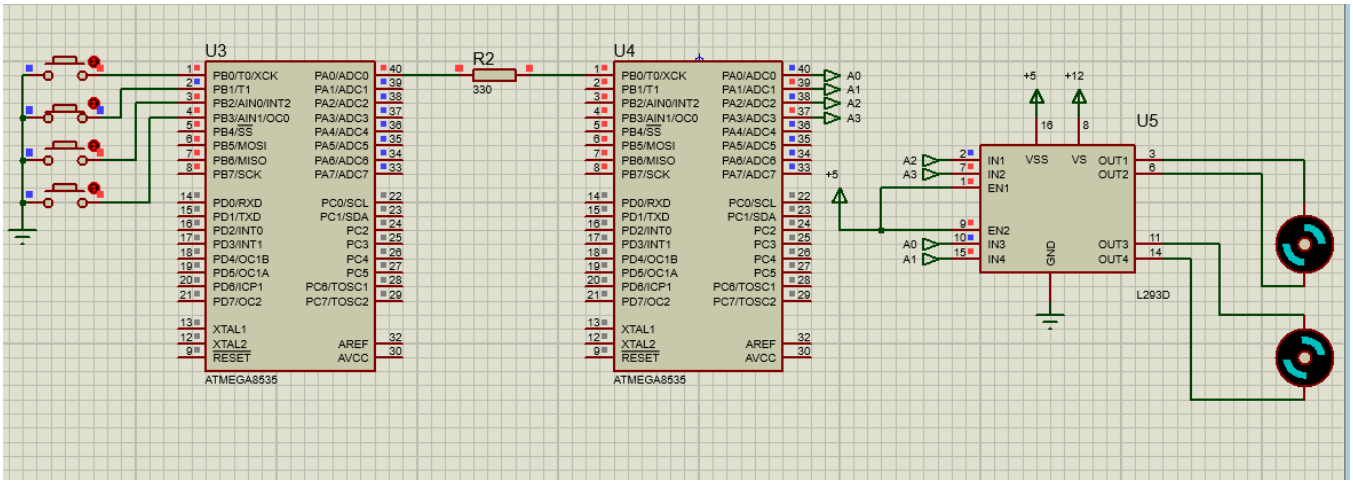

Figura 5. Simulación del proyecto 1.

Fotografía del circuito armado

A continuación, se muestra la figura 6 la evidencia sobre la realización y prueba del proyecto 1.

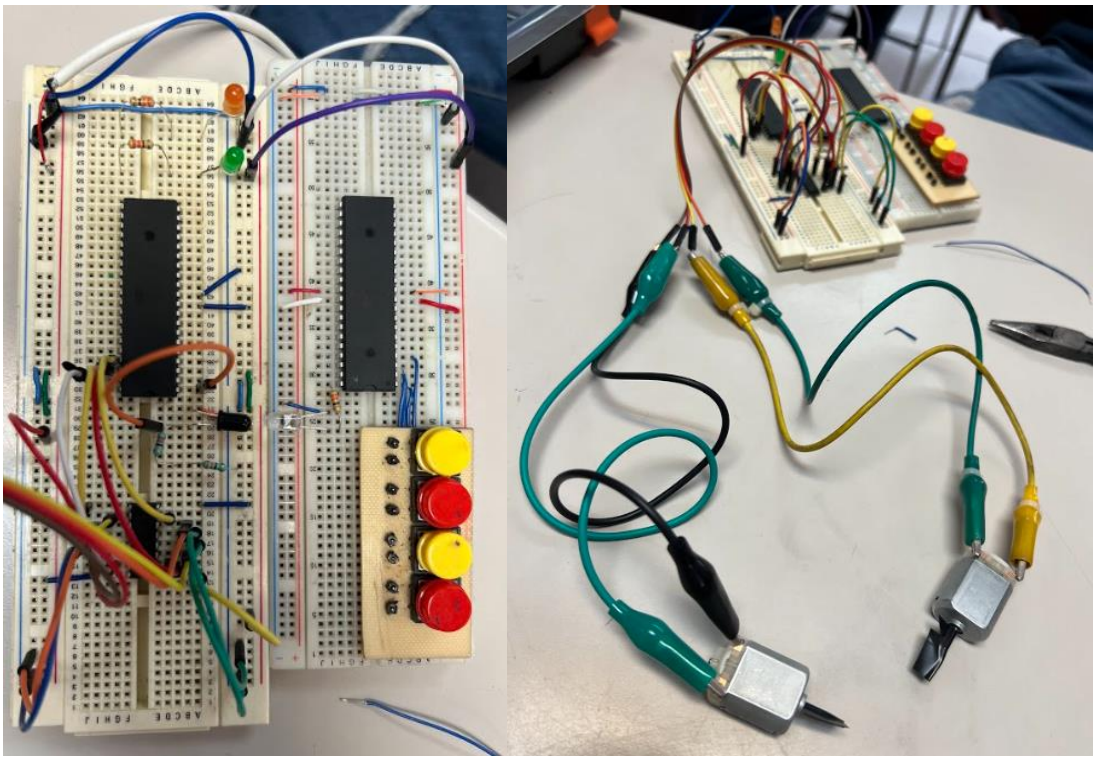

Figura 6. Circuito armado y funcionamiento del proyecto 1.

Observaciones y conclusiones Individuales

Ocampo Téllez Rodolfo

Para la realización de este proyecto se programaron de 2 microcontroladores, el primero fungiendo como transmisor (representando el control) y el segundo como receptor (representando el robot). Fue interesante que se debieron manejar diferentes frecuencias en las señales que emitía cada botón, ya que solo aplicando retrasos distintos en cada caso el receptor logra diferenciar cual botón ha sido pulsado y por ende que acción debe realizar. Para esto, del lado del receptor se implementó una ventana de pulsos mediante un ciclo for y de acuerdo con el retraso con el que viniera cada señal de entrada. Cabe destacar que se usaron los siguientes rangos de pulsos ya que no siempre se producían exactamente los mismos: entre 15 y 35 para avanzar, entre 40 y 60 para retroceder, entre 65 y 85 para ir a la izquierda, y entre 90 y 110 para ir a la izquierda.

El funcionamiento de la parte física en los circuitos armados en las protoboards tuvo cierto inconveniente, ya que, si bien las cuatro órdenes para el robot lograron ser ejecutadas de forma correcta, el componente de Puente H utilizado se calentaba demasiado, por lo que tuvimos que esperar algunos segundos entre órdenes para evitar que se dañara el circuito.

Patlani Mauricio Adriana

Este primer proyecto tiene la aplicación de 2 microcontroladores y el reto de usar la mayor parte de los conocimientos adquiridos de las prácticas anteriores en el cual uno actuará como receptor al recibir la señal el par infrarrojo y el otro como transmisor del cual se controlará el robot utilizando 4 frecuencias diferentes (adelante, atrás, izquierda, derecha) y por último un puente H que comunica las instrucciones a los motores del robot. Al inicio se tuvieron problemas al momento de implementar el proyecto, pues se notaba un aumento considerable de temperatura en el puente H y también un retraso visible al momento de captar las señales, pero se comprendió que dependía más de los componentes ya que al probarlo con leds, se comprobó su funcionamiento y así nos percatamos del retraso del circuito al captar las señales.

Ruvalcaba Flores Martha Catalina

El proyecto fue la aplicación de todos los conocimientos adquiridos en las practicas anteriores, puesto que este requería de la programación de dos microcontroladores y que estos se comunicaran por medio de un led y un sensor infrarrojo. Por lo que en un microcontrolador hace la función del control del robot y el otro será el receptor, con el par infrarrojo se reciben las señales. Se utilizaron 4 frecuencias para controlar dicho robot, entonces estas frecuencias son las que le llegan al par de infrarrojo, y con el uso de un puente H, se controlaron los motores. Al momento de probar el circuito, se observó un calentamiento en el puente H, provocando que no respondiera de forma rápida el proyecto, una forma de probar que estaba bien programado y armado fue utilizando los leds, ya que estos se activaban al momento de controlar el robot y no estábamos en la espera de respuesta de los motores.

Sandoval Hernández Eduardo

Este proyecto fue relativamente fácil de implementar una vez entendida la teoría por completo, lo interesante fue lograr hacer que se comunicaran los dos microcontroladores a través de un led y un sensor infrarrojos, así como lo hacen algunos dispositivos de la actualidad como las smart tv, proyectores, etc. La mayor complicación fue el que se elevara la temperatura del puente H lo cual hizo que no respondiera

bien el circuito y por ende creímos que probablemente habíamos implementado mal el control hasta que probamos utilizando leds. Tiene algunos puntos a mejorar como lo es la velocidad de respuesta ya que aún es lento para captar las señales que se envían del primer al segundo microcontrolador.

Bibliografía

• F. Aguilar. (2020, Noviembre). 09 Proyecto 01. Introducción a los Microcontroladores. [Online]. Disponible en: https://youtu.be/OcAB2RglvFc

• Atmel Corporation. (2006). 8-bit Microcontroller with 8K Bytes In-System Programmable Flash. [Online]. Disponible en: https://drive.google.com/file/d/1acpQaDlsyLHr3w3ReSgrlXRbshZ5Z-lb/view

• Sensoricx. (s.f). Puente H. [Online]. Disponible en: https://sensoricx.com/circuitos-para-armar/puente-h-funcionamiento-explicacion-detallada/

• Información Suelo Radiante. (Agosto, 2016) ¿Qué son y cómo funcionan los rayos infrarrojos?. [Online]. Disponible en: https://www.sueloradianteevoheat.com/que-son-y-como-funcionan-los-rayos-infrarrojos/

• UPM. (2011) Rayos infrarrojos?. [Online]. Disponible en: https://www.etsist.upm.es/estaticos/ingeniatic/index.php/tecnologias/item/571-rayos-infrarrojos.html