



Robot Pololu 3Pi con varios sensores a distancia para evitar colisiones

Carlos Valdivieso A. Msc.¹, César Espinoza Estrada², Jhonny Barzola Iza³

Facultad de Ingeniería en Electricidad y Computación

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil-Ecuador

cvaldivi@espol.edu.ec¹, cesar.espinosa.estrada@gmail.com²

jhonny198722@gmail.com³

Resumen

La finalidad del proyecto es desarrollar un robot Pololu 3Pi con varios sensores ópticos implementados en este, con el fin de evitar colisiones en el medio, al momento que el robot esté circulando sobre alguna pista. Al presionar el botón de encendido del robot Pololu 3Pi, este mostrará mediante su pantalla LCD mensajes de bienvenida; tiempo durante el cual el robot se encuentra también preparándose para el recorrido. Una vez hecho esto, el robot mostrará nuevos mensajes indicando la distancia medida por cada sensor óptico así como la velocidad establecida para lograr esquivar los obstáculos y con esto evitar una colisión cuando esté recorriendo la pista. Cuando se encuentre con algún obstáculo en su camino que no ha logrado evadir o que fue ubicado directamente muy cerca, se detendrá completamente y realizará un giro hasta encontrar una salida por donde seguir su recorrido. Una de las principales aplicaciones de este proyecto es la medición de distancia entre dos objetos estáticos o en movimiento, así como la resolución de laberintos.

Las herramientas a utilizar para este proyecto son: el robot Pololu 3Pi cuyo microcontrolador marca ATMEL modelo atmega328p programaremos con el USB AVR programmer, los sensores ópticos Sharp para conocer la distancia. Para realizar el código utilizaremos el programa AVR Studio 4 con la previa instalación de la librería del Pololu 3Pi, que nos servirá de ayuda para aplicar funciones más directas. Se realizarán pruebas para establecer las velocidades y distancias a las cuales el robot Pololu 3Pi deberá realizar acciones de evasión y búsqueda de salida.

Palabras Claves: Pololu, 3Pi, sensores, distancia, evitar colisiones, evadir obstáculos, atmega328p.

Abstract

This project is to develop a Pololu 3Pi robot provided with optical sensors, to avoid collisions with surrounding objects. When the turn-on/off button of the Pololu 3Pi is pressed, the robot will show welcome messages on its LCD display; during this time the robot runs the process to start its function. After this, the Pololu 3Pi will display new messages showing the distance measured by each optical sensor and the established speed to avoid the obstacles, thru this process it avoids collisions when is running around. When the robot finds an object on its way that was not detected before or is too close to it, it will stop completely and it will immediately spin until it finds a way out where it can continue with the exploration. The main application of this project is the distance measurement between two statics or moving objects, but there are other applications such as the solving labyrinths application.

The tools needed in this project are: the Pololu 3Pi robot, ATMEL atmega328p microcontroller which is programmed with the USB AVR programmer, the Sharp optical sensors to calculate the distance and to develop the code, the AVR Studio 4 program will be used in conjunction with the Pololu 3Pi library, which will help us to apply functions in the code. Tests will be done to set the speed and the distances in which the Pololu 3Pi robot will take actions to overcome an obstacle by avoiding it and finding a way out.

Keywords: Pololu, 3Pi, sensors, distance, avoid collisions, avoid obstacles, atmega328p.

1. DESCRIPCIÓN DEL PROBLEMA

1.1. ANTECEDENTES

Desde hace mucho tiempo tenemos la necesidad de lidiar con el inconveniente de las conocidas colisiones ya sea entre personas en alguna puerta o corredor, así como el problema del tránsito vehicular y en los últimos años a maquinarias y robots móviles. Uno de los detectores que más comúnmente se instalan en un robot es algún tipo de parachoques. El parachoques de un robot puede ser la única capacidad sensorial. Pero resultan necesarios aún teniendo otros métodos de detección dependiendo la aplicación, pues constituyen algo así como una última línea de defensa.

1.2. FUNCIONAMIENTO BÁSICO

El sistema de sensores le indica al robot que hay un objeto en su recorrido, con forma y tamaño sin determinar. Ante el suceso, la lógica de control puede ordenar una maniobra que le permita salir de la situación o una señal indicadora de lo ocurrido. Aunque muchos robots experimentales básicos se construyen con una cantidad mínima de sensores en la parte delantera, y a veces uno en la parte trasera, lo que permite una navegación primitiva, una disposición así no será suficiente para lograr que un robot se mueva por los ambientes de una casa sin quedar atrapado en algún sitio.

1.3. LÓGICA CONVENCIONAL PARA EVITAR COLISIONES

En una implementación básica, lo más típico es que la lógica funcione con las siguientes reacciones:

- Si detecta un obstáculo a la izquierda, detener la marcha y girar un poco a la derecha, luego continuar;
- Si detecta un obstáculo a la derecha, detener la marcha y girar un poco a la izquierda, luego continuar.

Esto funciona, pero lamentablemente también es una buena fórmula para garantizar que un robot quede atrapado muy pronto en un rincón de una casa, o incluso en el recorrido más controlado de una zona de pruebas.

1.4. LA CANTIDAD DE SENSORES A UTILIZAR

Es obvio que agregar un anillo continuo de sensores alrededor de la periferia del robot puede ser algo excesivo, un desperdicio de recursos. Los experimentos en ambientes caseros normales han determinado que cinco sensores en el frente y uno en la parte posterior ofrecen un funcionamiento óptimo dentro de lo que es una red sensorial algo limitada.

Con el que se logra un campo de visión muy completo para satisfacer los diferentes ángulos en los que se detecte un obstáculo. La configuración de dos sensores en la parte delantera del robot ubicados ligeramente laterales para ampliar el campo de visión es muy usada para el propósito que requerimos ya que nos ofrece la capacidad de análisis hacia la izquierda y derecha para evitar las colisiones en su recorrido y por medio de programación la posibilidad de realizar rutinas de movimientos complejos para resolver las distintas circunstancias en la que podría llegar a encontrarse el robot.

1.5. PROYECTOS SIMILARES

1.5.1. Robot con sensores mecánicos de choques

El contacto en un parachoques le indica al robot que hay un objeto en su recorrido, con forma y tamaño sin determinar. Ante el suceso, la lógica de control puede ordenar una maniobra que le permita salir de la situación.



FIGURA 1.1 Sensores mecánicos de choque

Navegar con parachoques es como moverse palpando en la oscuridad: alcanzar un destino puede llevarle al robot un tiempo excesivamente largo, si es que alguna vez llega. Se entiende que es el método más básico para implementar la detección de obstáculos en un robot.

1.5.2. Seguidor de distancia

De la misma forma tenemos muchas posibilidades si variamos la distancia a la que ha detectado un obstáculo y a la que el robot ejecute una respuesta. Además del tipo de respuesta que emita, viene a ser desde un simple led indicador hasta una posible compleja resolución con muchas opciones de respuesta rápida o giros y almacenamientos de datos.

1.5.3. Seguidor de línea

Otra aplicación es el seguidor de línea, que se aplica bastante como ejercicio para aprendizaje de la programación para la cual se puede utilizar desde tiras de papel blanco hasta cinta de colores brillantes ó negra adhesiva como camino para que el robot siga.



FIGURA 1.3 Robot Pololu 3Pi

1.5.4. Resuelve laberintos

Uno de las aplicaciones más conocidas con el Pololu 3Pi es la resolución de laberintos, puesto que por medio de la programación logramos obtener una enorme cantidad de maneras de resolverlo. Como el de la posibilidad de memorizarlo para ir directamente hacia la llegada en un posible segundo recorrido.

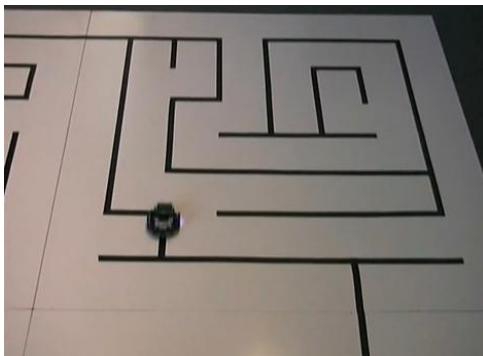


FIGURA 1.4 Robot Pololu 3Pi resolviendo laberinto con seguidor de línea negra

2. FUNDAMENTOS TEORICOS

2.1. POLOLU 3Pi

El nombre del robot viene del tamaño; el diámetro en centímetros es el valor de pi multiplicado 3 veces o 3Pi, en pulgadas es alrededor de 3,7 " de diámetro. El 3Pi es un robot autónomo con motores duales, cinco sensores de reflectancia, una pantalla LCD de 8×2 , un zumbador, 4 pilas AAA, y tres botones de usuario. Todos conectados a un microcontrolador ATmega328 programable. Capaz de alcanzar velocidades superiores a 3 pies por segundo, el 3Pi es un gran robot primero para los principiantes ambiciosos y segundo un robot perfecto para aquellos que buscan pasar de los robots para principiantes no programables o más lentos. El 3Pi está diseñado principalmente como un seguidor de línea, así que esto no es una gran preocupación.¹

El software libre y la cadena de herramientas GNU GCC proporcionan un compilador maduro, de gran alcance que puede hacer casi cualquier cosa que se necesite.

2.2. FUNCIONAMIENTO DEL POLOLU

2.2.1. Motores y engranajes.

Hay diferentes tipos de motores pero el más importante para robótica es el motor DC de escobillas y que usamos en el 3Pi.



FIGURA 2.5 Tipo de motor utilizado en el Pololu 3Pi

El típico motor DC contiene imanes permanentes en el exterior y bobinas electromagnéticas montadas en el eje del motor. Las escobillas son piezas deslizantes que suministran corriente desde una parte del bobinado a la otra produciendo una serie de pulsos magnéticos que permiten que el eje gire en la misma dirección. Cada motor tiene una velocidad máxima (sin resistencia aplicada) y un par máximo (cuando el motor está completamente parado). Llamamos a esto, funcionamiento a velocidad libre y al par de parada.²

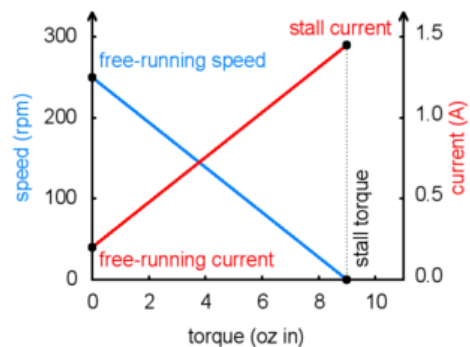


FIGURA 2.6 Curva de torque - velocidad de los motores del Pololu 3Pi

Naturalmente, el motor usa el mínimo de corriente cuando no se aplica una fuerza, y si la corriente que viene de la batería aumenta es por fuerzas de rodamiento o engranajes de modo que son parámetros importantes del motor. Según se muestra en el gráfico adjunto. La relación de engranaje es de 30:1 en el 3Pi, es decir 30 vueltas de motor, una vuelta de rueda. Estos parámetros están representados en la tabla siguiente:

TABLA 2.1 Características del funcionamiento de los motores del Pololu 3Pi

Par máximo:	oz·in
Velocidad libre:	700 rpm
Consumo min:	60 mA
Par máximo:	60 oz·in
Consumo máx.:	6 540 mA

2.3. COMPONENTES DEL DISEÑO

2.3.1. Sensor óptico SHARP GP2Y0A21YK0F

El GP2Y0A21YK0F es un sensor de medida de distancia compuesto por una combinación integrada de PSD (Detector Sensible a la Posición), IRED (Diodo Emisor Infrarrojo) y la señal del circuito en proceso. La variedad de la reflectividad del objeto, la temperatura del ambiente y la duración del funcionamiento no son influenciados fácilmente a la distancia de detección debido a la adopción del método de triangulación. Este dispositivo envía el voltaje correspondiente a la distancia de detección, así que este sensor también puede ser utilizado como un sensor de proximidad. Ejemplo de las características de las mediciones de distancia.³

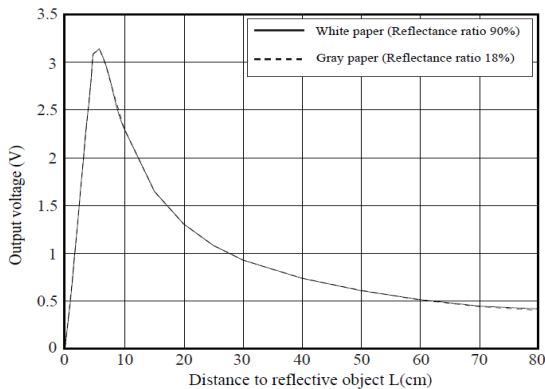


FIGURA 2.7 Curva de voltaje - distancia de los sensores ópticos

Se puede apreciar que hay una relación inversa entre el voltaje y la distancia del objeto encontrado. Mientras menor sea la distancia, mayor será el voltaje entregado por el sensor y viceversa.

2.4. HERRAMIENTAS DEL DISEÑO

2.4.1. Pololu 3Pi USB AVR programmer PGM03A.

Este dispositivo es un programador de controladores AVR-base, tales como los controladores de robots Orangután y el robot 3Pi. El programador emula un AVRISP v2 en un puerto serie virtual, por lo que es compatible con el software estándar de

programación AVR. Dos características adicionales ayudan a generar y depurar proyectos: un puerto serie TTL-nivel para la comunicación de uso general y un SLO-ámbito de aplicación para el monitoreo de señales y los niveles de tensión. Un cable USB y el cable ISP están incluidos.⁴

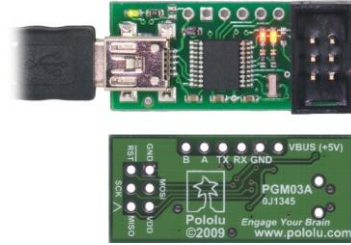


FIGURA 2.8 Programador Pololu 3Pi USB AVR

El programador AVR Pololu USB es un muy compacto, de bajo coste en el programador del sistema (ISP) para microcontroladores AVR de Atmel, lo que hace más versátil este dispositivo, es que es una solución de programación atractiva para los controladores AVR-base como los controladores de robots Pololu orangután. El programador AVR USB se conecta al puerto USB de su ordenador a través de un USB incluido a un cable mini-B y se comunica con el software de programación, tales como AVR Studio o AVRDUDE, a través de un puerto COM virtual utilizando el protocolo AVRISPv2/STK500.⁵

3. DISEÑO DE LA SOLUCIÓN

3.1. FUNCIONAMIENTO BÁSICO

El sistema de sensores le indica al robot que hay un objeto en su recorrido, con forma y tamaño sin determinar. Ante el suceso, la lógica de control puede ordenar una maniobra que le permita salir de la situación. Aunque muchos robots experimentales básicos se construyen con una cantidad mínima de sensores en la parte delantera, y a veces uno en la parte trasera, lo que permite una navegación primitiva, una disposición así no será suficiente para lograr que un robot se mueva por los ambientes de una casa sin quedar atrapado en algún sitio.

3.2. DIAGRAMA DE BLOQUES

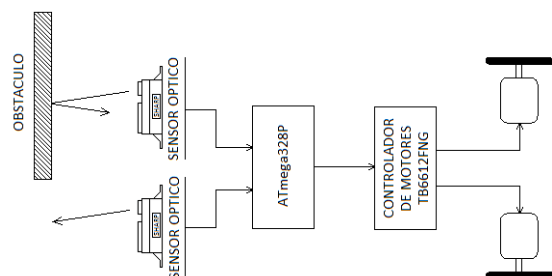


FIGURA 3.1 Diagrama de bloques del proyecto

3.3. TÉCNICAS PARA EVITAR COLISIONES

En una implementación básica, lo más típico es que la lógica funcione con las siguientes reacciones:

- Si detecta un obstáculo a la izquierda, detener la marcha y girar un poco a la derecha, luego continuar.
- Si detecta un obstáculo a la derecha, detener la marcha y girar un poco a la izquierda, luego continuar.

Esto funciona, pero lamentablemente también es una buena fórmula para garantizar que un robot quede atrapado muy pronto en un rincón de una casa, o incluso en el recorrido más controlado de una zona de pruebas.

3.3.1. PRIMERAS DIFICULTADES CON LOS SENSORES

En primer lugar, tan pronto como el robot encuentre una esquina con paredes en ángulo podrá quedar atrapado en un movimiento sin fin de ida y vuelta, hasta que se le acaben las baterías. A esto se le llama "encajonarse", y hay ciertas soluciones que se pueden utilizar para corregir la condición. En segundo lugar, en áreas estrechas o cerradas, el robot puede quedar atrapado entre dos paredes muy juntas, rebotando entre las paredes por horas, sin ir a ningún lado. Éste es el "problema del túnel".

Si no se ha implementado una manera de saber si la parte posterior va a chocar con algo, el robot hará fuerza con sus motores para intentar retroceder.

3.4. SELECCIÓN DE SENSORES A UTILIZAR

Es obvio que agregar un anillo continuo de sensores alrededor de la periferia del robot puede ser algo excesivo, un desperdicio de recursos. Los experimentos en ambientes caseros normales han determinado que cinco sensores en el frente y uno en la parte posterior ofrecen un funcionamiento óptimo dentro de lo que es una red sensorial algo limitada.⁶

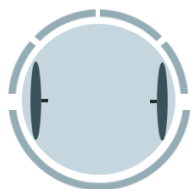


FIGURA 3.2 Configuración de varios sensores

Antes de que definamos qué comportamiento se debe producir cuando se activa cada sensor, recordemos que hay tres tipos de situaciones que debemos considerar. Primero, un comportamiento simplemente reactivo o de reflejos. Se trata de una

lógica simple de lecturas que activa un sistema preestablecido de maniobras de escape, que dependerán de sensor se haya activado. A estas maniobras se les llama "balísticas", porque una vez que comienzan, continúan hasta el final. En segundo lugar, y un poco más avanzado, se pueden analizar una serie de posibles choques que se repiten y están relacionados de manera cronológica. En tercer lugar, agregamos un factor de variación. En "inteligencia" robótica, agregar variación es, simplemente, introducir una pequeña cantidad de valores azarosos a la inteligencia estándar de reflejos de la máquina.

3.5. DIAGRAMA DE FLUJO

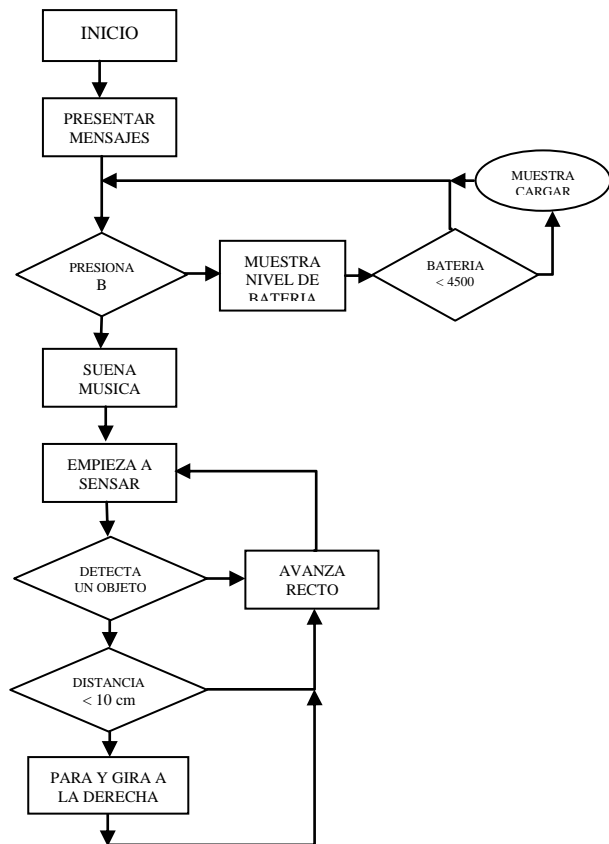


FIGURA 3.7 Diagrama de flujo

3.6. CÓDIGO

A continuación se muestra el código hecho en lenguaje C con el cual logramos configurar el robot Pololu para realizar lo antes dicho en el diagrama de flujo.⁷

/* Grupo # 6

Integrantes:

- César Espinoza

- Jhonny Barzola

* Descripción: Robot Pololu 3Pi evasor de obstaculos

* Este programa utiliza dos sensores SHARP GP2Y0A21YK0F conectados


```

* en ADC7 (izquierda) y ADC5 (derecha) y que le
permite esquivar obstáculos
*/
// Includes
#include <pololu/3pi.h> // especifico 3Pi
#include <avr/pgmspace.h> // variables en memoria
de programa
// Mensajes de introducción guardados en memoria de
programa
const char linea1[] PROGMEM = " GRUPO ";
const char linea2[] PROGMEM = " 6 ";
const char linea3[] PROGMEM = " CESAR ";
const char linea4[] PROGMEM = "ESPINOZA";
const char linea5[] PROGMEM = " JHONNY ";
const char linea6[] PROGMEM = "BARZOLA";
// Tonos musicales guardados en memoria de programa
const char tono1[] PROGMEM = ">g32>>c32";
const char tono2[] PROGMEM = "L16 cdegreg4";
// Variables generales
unsigned int sens_der; // sensor derecha
unsigned int sens_izq; // sensor izquierda
int motor_der; // motor derecha
int motor_izq; // motor izquierda
void iniciar(){
  emitters_off();
  //Muestra el nombre del grupo e integrantes
  print_from_program_space(linea1);
  lcd_goto_xy(0,1);
  print_from_program_space(linea2);
  play_from_program_space(tono1);
  delay_ms(1000);
  clear();
  print_from_program_space(linea3);
  lcd_goto_xy(0,1);
  print_from_program_space(linea4);
  delay_ms(1000);
  clear();
  print_from_program_space(linea5);
  lcd_goto_xy(0,1);
  print_from_program_space(linea6);
  delay_ms(1000);

  //Muestra el voltaje de la batería y espera botón
  while(!button_is_pressed(BUTTON_B))
  {
    clear();
    print_long(read_battery_millivolts()); // usa
ADC6 para sensar la batería del circuito
    print("mV");
    lcd_goto_xy(0,1);
    if (read_battery_millivolts()<4800){
      print (" !CARGAR¡");
      red_led(1);
    }
    else
      print("Pulsa B");
      delay_ms(100);
  }
}

```

```

// Espera que se presione el botón B para empezar a
moverse
wait_for_button_release(BUTTON_B);
clear();
// Toca música y espera a que termine para empezar
play_from_program_space(tono2);
while(is_playing());
}
void lee_sensores(){
  // Mira a la derecha si hay obstáculos
  if (!analog_is_converting())
  sens_der = analog_read(5);
  // Mira a la izquierda si hay obstáculos
  if (!analog_is_converting())
  sens_izq = analog_read(7);
}
void busca_salida(){
  clear();
  lcd_goto_xy(0,0);
  print("Buscando"); // Busca salida
  red_led(1); // Encienda leds rojo y verde
  green_led(1);
  while (sens_der>400 || sens_izq>400){
    play ("c32");
    set_motors(40,-40); // gira or -40,40
    delay_ms(200);
    lee_sensores();
    play ("g32");
  }
  red_led(0); // OK salida encontrada
  green_led(0); // Apaga leds rojo y verde
}

int main(){
  // inicializa el robot Pololu 3Pi
  iniciar();
  // Bucle principal
  while(1){
    lee_sensores();
    if (sens_der>400 || sens_izq>400){
      set_motors(0,0); // para los motores
      delay_ms(200);
      busca_salida();
    }
    if (sens_izq > 200 && sens_der < 200){
      // obstaculo izq entonces gira a la derecha ----->
      motor_izq=100-sens_der/10; // acelera
      motor_der=100-sens_izq/10; // reduce
    }
    if (sens_izq < 200 && sens_der > 200){
      // obstaculo der entonces gira a la izquierda <-----
      motor_izq=100-sens_der/10; // reduce
      motor_der=100-sens_izq/10; // acelera
    }
    if (sens_izq < 200 && sens_der < 200){
      motor_izq=100;
      motor_der=100;
    }
  }
}

```

```

if (sens_izq < 100 && sens_der < 100){
    motor_izq=127;
    motor_der=127;
}
set_motors(motor_izq, motor_der);
delay_ms(100);
clear(); // Muestra los valores
lcd_goto_xy(0,0);
print_long(sens_izq);
lcd_goto_xy(5,0); // valores de sensores
print_long(sens_der);
lcd_goto_xy(0,1);
print_long(motor_izq);
lcd_goto_xy(5,1); // valores de motores
print_long(motor_der);
}
}

```

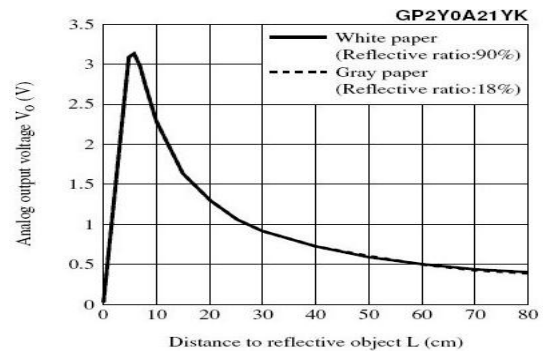


FIGURA 4.3 Curva de voltaje - distancia de los sensores Sharp

4. ANÁLISIS DE LOS RESULTADOS, VALIDACIÓN Y PRUEBAS

4.1. SIMULACIÓN EN PROTEUS

Gracias a la ayuda de la herramienta de simulación Proteus, se logró realizar la mayor parte de este proyecto.

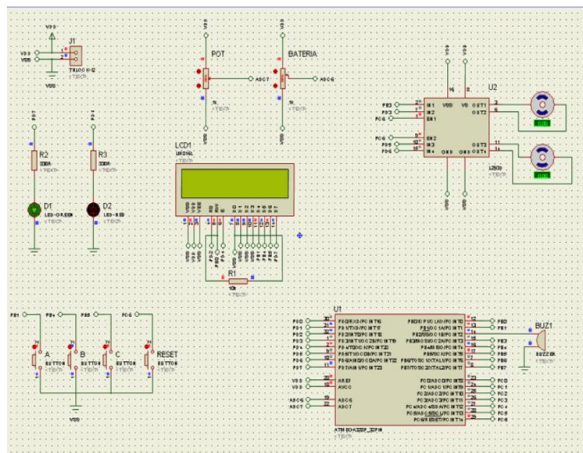


FIGURA 4.1 Diagrama en Proteus del Pololu 3Pi

La simulación de nuestros sensores Sharp lo hemos implementado con el uso de unos potenciómetros conectados a una fuente y variamos el potenciómetro para simular la señal que emite al detectar un obstáculo a una distancia que emiten los sensores Sharp.

El problema de la simulación de los sensores Sharp es por su curva característica la cual nos representa un rango de valores de voltaje, los cuales pueden ser mal interpretados en distancias entre 10cm y 80 cm debido a que son valores de voltaje de similares magnitudes para las respectivas distancias.

4.2. IMÁGENES DEL PROYECTO FUNCIONAL

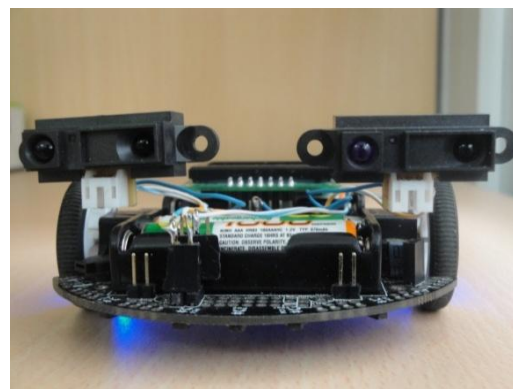


FIGURA 4.4 Robot Pololu 3Pi con los sensores acoplados

El inicio del proyecto empieza mostrando en la pantalla LCD los nombres de los integrantes del proyecto. Además nos pide presionar la tecla B del POLOLU para que los motores puedan arrancar progresivamente desde el reposo hasta alcanzar una velocidad constante y adecuada para que permita que los sensores puedan estimar correctamente las distancias de los obstáculos que se presentan en la trayectoria del robot.



FIGURA 4.8 Nivel actual del voltaje mostrado en la LCD funcionando

Mientras espera por el botón B, se muestra en la LCD el valor del voltaje de las baterías expresada en mili voltios; si el valor es menor a 4500 mV se presentara el mensaje “CARGAR”.

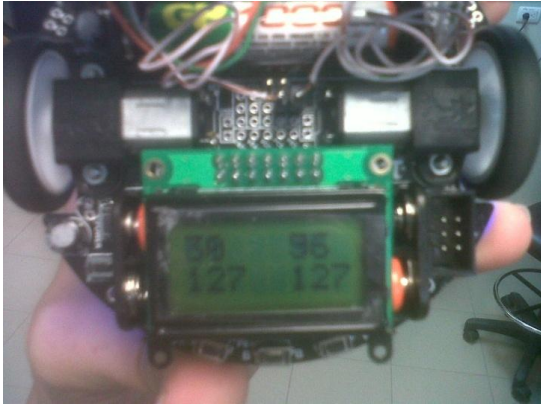


FIGURA 4.9 Nivel de velocidad de cada motor y distancia respectiva de cada sensor mostrados en la LCD

Cuando el robot se encuentra en movimiento, los mensajes que se presenta en la LCD son los datos correspondientes a las velocidades de los motores y sus respectivas distancias si se presenta algún obstáculo en la trayectoria. Además cuando el robot encuentra algún objeto, describe un pequeño giro previniendo una posible colisión pero en el caso que el obstáculo aparece de improvisto, de que el robot no lo haya detectado o que el obstáculo no estuvo en una posición en la cual los sensores hayan logrado detectar el objeto. El robot se detiene inmediatamente y gira en busca de una salida y presenta un mensaje que dice “BUSCANDO”.

5. CONCLUSIONES

1. Con una configuración de dos sensores tenemos la posibilidad de resolver una gran cantidad de limitaciones respecto a la detección de los obstáculos, sin embargo debemos tomar en cuenta la necesidad de obtener más información del medio (mas sensores) al quedar sujeto a condiciones durante el recorrido del robot Pololu 3Pi en una pista o una sala, ya que es una limitación muy difícil de enfrentar.
2. Este proyecto nos brinda la oportunidad de resolver problemas lógicos de la vida real aplicados a la robótica por medio del ingenio en la programación; logrando demostrar que con una reducida cantidad de información, se puede conseguir resolver una gran cantidad de limitaciones de desplazamiento al momento de evadir obstáculos mediante mediciones periódicas de localización.
3. Los sensores ópticos son una manera muy acertada de obtener información de distancia para el Atmega328

del Robot Pololu para la detección de objetos. Con estas señales se hizo más sencillo relacionar constantemente la ubicación de los obstáculos en el medio, así como la capacidad de realizar una respuesta inmediata y eficaz.

6. AGRADECIMIENTOS

A Dios y a la Santísima Virgen María por haberme permitido seguir en este mundo y poder ver realizada esta anhelada meta.

A mis padres por haberme brindado su apoyo incondicional durante toda mi vida; en especial en mi carrera académica, por darme la oportunidad de terminar mi formación profesional y por creer en mí en todo momento. Agradecimiento especial a Msc. Carlos Valdivieso, Director de Seminario de Graduación por su orientación e invaluable ayuda.

César Andrés Espinoza Estrada

A Dios, a quien mantengo firmemente en mis pensamientos y en mi corazón. Al ser más importante que tengo y le dedico cada momento. Al cual obsequio palabras de amor con prudencia, respeto y temor. El que me ha permitido una vida llena de salud, gracia y al que debo todo lo que tengo por su gran misericordia. A mi familia de todo corazón, que Dios los bendiga en cada momento y los mantenga en su gracia.

A los Ingenieros, maestros, compañeros y amigos.

Jhonny Javier Barzola Iza

7. REFERENCIAS

- [1] Catálogo de características del Robot Pololu.
<http://www.pololu.com/catalog/product/136>
Fecha de consulta: 20 de abril del 2011.
- [2] Manual de guía de usuario del robot Pololu.
[http://www.pololu.com/file/0J137/Pololu3PiRobotGui aUsuario.pdf](http://www.pololu.com/file/0J137/Pololu3PiRobotGui%20aUsuario.pdf)
Fecha de consulta: 27 de abril del 2011.
- [3] Catálogo del Sensor óptico GP2Y0A21.
<http://www.pololu.com/catalog/product/975>
Fecha de consulta: 27 de abril del 2011.
- [4] Herramientas para su programación.
<http://robots.net/article/2866.html>
Fecha de consulta: 27 de Abril 2011.
- [5] Descripción del Pololu AVR USB programmer.
<http://www.pololu.com/catalog/product/1300>
Fecha de consulta: 27 de Abril 2011.
- [6] Colisiones y su funcionamiento.
http://robots-argentina.com.ar/Sensores_parachoques.htm
Fecha de consulta: 1 de mayo del 2011.
- [7] Documentación de la Librería para robots Pololu.
<http://www.pololu.com/docs/0J21/6.a>
Fecha de consulta: 27 de abril del 2011.