



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN**

**“Aplicación con inclinómetro y acelerómetro para
mantener control de movimiento de un robot”**

TESINA DE SEMINARIO

Previa a la obtención del título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Juan Fernando García Vargas

Xavier Enrique Sánchez Castro

GUAYAQUIL - ECUADOR
2011

AGRADECIMIENTO

A la Santísima Virgen y a Dios que me ha iluminado a lo largo de mi carrera universitaria y de mi vida.

Mi eterno agradecimiento y aprecio a mis padres y al Ing. Carlos Valdivieso, Director de Seminario de Graduación por su invaluable ayuda.

J.F.G.V.

Agradezco a Dios por darme la oportunidad de seguir en este mundo y a mis padres por el esfuerzo y el apoyo que siempre me han dado.

X.E.S.C.

DEDICATORIA

A ti Dios que me diste la oportunidad de vivir y de regalarme una familia maravillosa.

Con mucho cariño principalmente a mis padres Jorge Enrique Sánchez Naranjo y María Milene Castro Menéndez que me dieron la vida y han estado en todo momento junto a mí. A mis tías, tíos, abuelos, abuelas, primos, primas, amigos y amigas por apoyarme a cada instante, y brindarme su amor en cada etapa de mi vida.

Quiero agradecer especialmente a Ailén Covarrubia por el apoyo incondicional que me brindó. Te Amo. Sólo Dios sabe que habría sido sin ti.

A todos ustedes les quiero dedicar éste trabajo con mucho cariño como muestra de agradecimiento por todo lo que significan para mí.

X.E.S.C.

Dedico este trabajo a mi madre María Vargas, aquella que siempre me brindó su apoyo, cariño y comprensión en todo momento, por lo cual nunca baje los brazos ante cualquier problema que se suscitara a lo largo del camino que decidí seguir, a mi abuelita María Garcés, por sus continuos cuidados y consideraciones que me demostraron en todo momento que estaba ahí amándome y apoyándome incondicionalmente, las amo con todo mi corazón y les agradezco profundamente todo lo que hicieron por mí, todo lo que he logrado es gracias a ustedes.

A mi abuelito Juan Vargas, que aunque no se encuentre con nosotros físicamente, se cuanto le hubiera gustado poder compartir este momento tan importante en mi vida, siempre estuviste y seguirás estando a mi lado, te agradezco con todo el corazón.

A mi padre Ángel García, familia, amigos y amigas que me brindaron su apoyo y supieron darme un sano consejo en el momento que lo necesitaba.

A todos ustedes que formaron una parte importante en mi vida, les dedico éste trabajo como muestra del profundo agradecimiento y cariño que siento en mi corazón.

J.F.G.V.

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in black ink, appearing to read 'Carlos Valdivieso A.', written over a horizontal dashed line.

Ing. Carlos Valdivieso A.

PROFESOR DE SEMINARIO DE GRADUACIÓN

A handwritten signature in black ink, appearing to read 'Hugo Villavicencio', written over a horizontal dashed line.

Ing. Hugo Villavicencio

DELEGADO DEL DECANO

DECLARACION EXPRESA

“La responsabilidad del contenido de esta Tesina de Seminario, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”.

(Reglamento de Graduación de la ESPOL).


JUAN FERNANDO GARCÍA VARGAS


SÁNCHEZ CASTRO XAVIER ENRIQUE

RESUMEN

Esta tesina se divide en tres partes para una mejor comprensión.

Brevemente se presenta el sensor de movimiento MEMSIC 2125 y algunos conceptos del AVR STUDIO con el que se desarrolló la programación del microcontrolador que usa el Robot Pololu 3pi.

En el capítulo I se detallan ciertas generalidades respecto a los sensores de movimiento, historia, tipos de sensores, descripción del problema, aplicaciones de los sensores, así como las ventajas de haber elegido el MEMSIC 2125 sobre los demás sensores que existen en la actualidad.

El capítulo II consta aquellos fundamentos teóricos que consideramos importantes dar a conocer, acerca de los elementos que van a participar en el desarrollo del proyecto, como son el sensor MEMSIC 2125 en el cual detallamos principio de funcionamiento, características y configuración de pines, el robot Pololu 3pi, así como una breve descripción del AVR STUDIO el cual se ha utilizado para realizar las diferentes configuraciones del microcontrolador.

El capítulo III ilustra la forma como se desarrollo la implementación a nivel de Hardware y Software, se detalla el código que lleva el microcontrolador explicando rutinas utilizadas, así como funciones predefinidas en la librería del Pololu 3pi, las cuales han sido usadas para una mayor facilidad de implementación.

Además se realizan las pruebas pertinentes del funcionamiento del proyecto ensamblado así como gráficas con el osciloscopio que nos ayudan a demostrar los principios estipulados en el capítulo 2.

INTRODUCCIÓN

El proyecto que se presenta es una Aplicación con inclinómetro y acelerómetro para mantener control de movimiento de un robot. Para esto se desea obtener la medición de la inclinación y la aceleración a que se encuentra sometido cierto robot, y gracias a esta se logrará tomar una decisión en cuanto a estabilidad se refiere, ya sea incrementar o disminuir la velocidad a la cual se mueve con el fin de mantener una velocidad constante a lo largo de su trayectoria.

Para esto, se utilizarán sensores adecuados para la obtención de estos parámetros y el encargado de interpretar estas señales, será el microcontrolador que posee el robot Pololu 3Pi, el cual logrará el adecuado control de la velocidad al tomar la decisión de incrementarla o disminuirla según sea necesario.

Se detallará además, la forma como se logró acoplar los sensores al robot, así como la configuración usada y razones por las cuales se decidió usar el Robot Pololu 3pi y los sensores que se van a comentar.

INDICE GENERAL

	Pág.
RESUMEN.....	II
INTRODUCCIÓN.....	IV
INDICE GENERAL.....	V
INDICE DE FIGURAS.....	VIII
 I. GENERALIDADES	 1
1.1 Introducción	1
1.2 Historia.....	2
1.2.1 Sensores a través de la historia	2
1.3 Descripción del problema	6
1.3.1 Situación actual.....	6
1.3.1.1 Aplicaciones del acelerómetro.....	6
1.3.1.2 Aplicación del acelerómetro en la tecnología móvil de celulares.....	10
1.3.1.3 Aplicación del acelerómetro en el análisis de vibraciones en maquinas y estructuras	12
1.3.1.4 Aplicaciones del inclinómetro	13
1.3.1.5 Altímetro-Inclinómetro digital	14

II. FUNDAMENTACIÓN TEÓRICA	15
2.1 Descripción del Hardware y Software utilizado	15
2.1.1 Descripción del Robot Pololu 3pi	16
2.1.2 Descripción del microcontrolador ATMEGA328.....	18
2.1.3 Descripción del acelerómetro MEMSIC 2125.....	19
2.1.3.1 Características importantes del MX 2125.....	21
2.1.3.2 Configuración de pines del MX2125.....	22
2.1.3.3 Principio de funcionamiento del MX2125.....	22
2.1.3.4 Protocolo de comunicación del MX 2125.....	24
2.1.3.5 Cálculo del ángulo de inclinación.....	25
2.1.4 Descripción del programador AVR USB Pololu	26
2.1.4.1 Microcontroladores soportados.....	28
2.1.4.2 Sistemas operativos soportados	28
2.1.4.3 Puerto Serial TTL	29
2.1.4.4 Osciloscopio.....	29
2.1.5 Descripción del AVR STUDIO 4	30

III. DISEÑO DE SOFTWARE Y HARDWARE	31
3.1 Características del sistema.....	31
3.2 Descripción de los bloques funcionales	32
3.3 Diagrama de flujo de nuestro proyecto	32
3.4 Diagrama esquemático del Pololu 3pi	35
3.5 Programa desarrollado en el microcontrolador ATMEGA328.....	36
3.6 Gráficas tomadas de la salida del MEMSIC en el eje X.....	49
3.7 Implementación del Hardware	51

CONCLUSIONES

RECOMENDACIONES

ANEXOS

BIBLIOGRAFIA

INDICE DE FIGURAS

	Pág.
Figura 1.1. Acelerómetro de dos ejes	9
Figura 1.2. Acelerómetro de tres ejes	9
Figura 1.3. Acelerómetro de tres ejes, MMA7361L.....	9
Figura 1.4. Acelerómetro de tres ejes, LSM303DLH	9
Figura 1.5. Aplicación del acelerómetro en la Tecnología móvil	10
Figura 1.6. Aplicación del acelerómetro en el análisis de vibraciones.....	12
Figura 1.7. Acelerómetro Piezo Eléctrico	13
Figura 1.8. Altímetro-Inclinómetro digital	14
Figura 2.1. Robot Pololu 3pi	16
Figura 2.2. Configuración de pines del ATMEGA 328	19
Figura 2.3. Estructura del acelerómetro masa sísmica	20
Figura 2.4. Configuración de pines del MEMSIC 2125.....	22
Figura 2.5. Estructura del sensor MX2125.....	23
Figura 2.6. Gráfica que muestra la señal PWM de salida del sensor MX2125	24
Figura 2.7. Gráfica que muestra los ejes del sensor MEMSIC.....	25
Figura 2.8. Cálculo del ángulo de inclinación del sensor MX2125	26
Figura 2.9. Programador AVR USB Pololu	27
Figura 2.10. Entorno de trabajo del AVR STUDIO 4.....	30
Figura 3.1 Bloques funcionales del proyecto	32

Figura 3.2	Diagrama Esquemático.....	35
Figura 3.3	Gráfica de señal de salida Sensor en Reposo.....	49
Figura 3.4	Gráfica de señal de salida con inclinación máxima en el eje positivo	50
Figura 3.5	Gráfica de señal de salida con inclinación máxima en el eje negativo	50
Figura 3.6	Pendiente Ascendente, inclinación de 45°	51
Figura 3.7	Pendiente Descendente, inclinación de -13°	52
Figura 3.8	Superficie Horizontal, ángulo 0°	52
Figura 3.9	Robot 3pi siguiendo la línea.....	53
Figura 3.10	Pista diseñada para nuestro proyecto.....	53

CAPITULO 1

1.1 INTRODUCCIÓN

En este capítulo se proveerá la definición de acelerómetro e inclinómetro, se tratará la historia de los sensores, así como también su evolución a través de la historia. Además, haremos una descripción del problema que intentamos solucionar y la situación actual del mismo.

Más adelante en este capítulo, se enumeran algunas de las aplicaciones más importantes de los acelerómetros e inclinómetros en la actualidad, y se da una breve explicación de cada una de ellas.

1.2 HISTORIA

Se comienza por dar una apropiada definición acerca de acelerómetro e inclinómetro. Se define como acelerómetro a cualquier instrumento destinado a medir aceleraciones. Un acelerómetro también es usado para determinar la posición de un cuerpo, pues al conocerse su aceleración en todo momento, es posible calcular los desplazamientos que tuvo. Considerando que se conocen la posición y velocidad original del cuerpo bajo análisis, y sumando los desplazamientos medidos se determina la posición.

Un inclinómetro es un instrumento usado por topografía, por la aviación y por los navíos para medir la inclinación del plano con respecto de la horizontal (superficie terrestre).

1.2.1 Sensores a través de la historia

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una Tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación con lo que Puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo.

A menudo disfrutamos de las ventajas que nos aportan muchas tecnologías, que se muestran invisibles a nuestra percepción, y que muchas veces ni siquiera notamos su existencia, aunque es gracias a ellas que podemos desarrollar muchas tareas fundamentales. Esto se hace cada día más evidente con el

progreso digital imparable que introduce nuevos componentes a nuestros aparatos haciéndolos más eficientes, y a la vez complejos.

Sea como fuere, términos hoy ya bastante populares como son “acelerómetro” o “sensor inercial”, serían aún desconocidos si no fuera por su exitosa integración en los controles de la consola de la Nintendo Wii o en el Smartphone estrella de Apple, el iPhone. Y el concepto clave que manejan estas ingeniosas aplicaciones es el transformar una acción mecánica en una respuesta electrónica útil.

En la práctica existen cinco movimientos físicos fundamentales que puede sufrir un objeto y que una vez tratados pueden dar lugar a muchas aplicaciones: aceleración, vibración, golpe, inclinación y rotación. Si somos capaces de detectar los niveles precisos de dichos movimientos, podremos disponer de una información muy valiosa a la hora de interactuar con dichos objetos.

Las posibilidades que brinda son infinitas. Un ejemplo muy sencillo es en un sistema de ahorro de energía de una maquina. Si mediante sensores podemos detectar el movimiento que realiza ésta, sabremos también cuando está parada y cuando en un funcionamiento intenso. A partir de ahí, podemos enviarle fácilmente ordenes de que reduzca su consumo energético a mínimos (que se sitúe en modo “sleep”) cada vez que esté parada, ya que no requerirá un suministro tan grande desde las baterías, que en caso contrario irían a parar a sacos vacíos (a sub-sistemas que permanecerían activos aunque no se usasen, como la estructura de detección de obstáculos, muchos drivers motores, etc.). Así es como pasamos, poco a poco, de un simple aparato que cumple órdenes que nosotros le imponemos a algo más, un robot capaz de “autogestionarse” y

dar respuestas eficientes según las circunstancias de cada momento. Estamos hablando, claro está, de la entrada al mundo de la inteligencia artificial.

Así pues, un sistema inteligente deberá detectar en todo momento los cinco movimientos básicos antes mencionados para luego dominar plenamente un objeto e interpretarlo. En realidad y aunque nosotros no nos demos cuenta con los sentidos, todos ellos (aceleración, vibración, choque, inclinación y rotación) no son más que diferentes manifestaciones de uno mismo: una aceleración durante períodos temporales varios. Por este motivo, el sensor que detecta el movimiento es conocido como acelerómetro, y es capaz de darse cuenta electrónicamente de algo que a simple vista los seres humanos no percibimos.

La física, pero, conoce perfectamente que una operación de aceleración (medido en metros por segundo al cuadrado) mide la variación de velocidad por unidad de tiempo y que por tanto está relacionando espacio (con sus tres ejes básicos X, Y, Z) y tiempo de una forma directa y exponencial. Usando negativos a esas unidades estaríamos hablando de des-aceleración. En este sentido, un movimiento de vibración se podría considerar como una aceleración combinada con una des-aceleración constante y repetitiva. En la inclinación juegan parte, además de la aceleración, los movimientos gravitatorios que atraen los cuerpos al centro terrestre. En el choque, la aceleración de un objeto se confronta con la de otro, provocando como consecuencia una des-aceleración. En definitiva, vemos como estos cinco movimientos que detectan los sensores en realidad se basan en uno sólo.

Los acelerómetros que son más comúnmente utilizados hoy en día en el mercado se sirven de la fuerza gravitatoria para contrastar sus mediciones, y tras ello, convierten estos resultados en Voltios o en Bits (en el caso de dispositivos con salida digital). Esta información es trasladada a un microprocesador o microcontrolador que se encarga de interpretar los datos que han adquirido y presentarlos de modo audiovisual, o los usan para ejecutar otras instrucciones de programa.

De momento nos hemos referido a movimientos lineales (los que acontecen con la aceleración), pero en el caso de la rotación, entra en juego un nuevo movimiento: el angular.

Aquí es donde ya no tiene porqué observarse la aceleración para que tenga lugar, ya que los grados en los que se sitúa el objeto se deben medir con el uso de un giroscopio.

La evolución constante en la tecnología MEMS (**micro-electro-mechanical systems**), ha posibilitado que se fabriquen pequeños acelerómetros en rangos de detección muy bajas y altas unidades de gravedad, lo cual significa una ampliación de los anchos de banda y en definitiva, aumentando también los posibles campos de aplicación. [1]

1.3 DESCRIPCIÓN DEL PROBLEMA

1.3.1 Situación actual

El proyecto que se presenta es una Aplicación con inclinómetro y acelerómetro para mantener control de movimiento de un robot, sin embargo, la gama de aplicaciones

en las que se pueden utilizar dichos dispositivos, es muy amplia. Por lo cual, se detallará algunas de las más importantes y que pueden contribuir al desarrollo del presente proyecto.

1.3.1.1 Aplicaciones del Acelerómetro

En la actualidad, los acelerómetros han venido incrementando su aplicación de acuerdo a las necesidades industriales y tecnológicas, por ejemplo en el ámbito industrial, tal es el caso en la medición de las vibraciones en los ambientes hostiles en ámbito de la monitorización de máquinas industriales.

Se detallan los siguientes casos:

La versión de alta frecuencia es ideal para la monitorización de la vibración de cojinetes cilíndricos, cajas de engranajes, bombas.

La versión de media frecuencia se usa para la monitorización de la vibración de bombas, motores, ventiladores, transportadores.

El modelo a baja frecuencia, se usa en el sector petrolquímico, máquina herramienta, e industrias del papel para la monitorización de agitadores de baja velocidad, torres de refrigeración.

La versión alta temperatura, se usa donde es necesaria una mayor protección de temperatura, como por ejemplo la sección de una máquina para el papel.

También los acelerómetros han venido siendo incluidos en muchas aplicaciones tecnológicas, desde computadores personales hasta edificios.

- En computadores portátiles permiten detectar cuando hay una caída y detener el disco duro para prevenir daños.
- En vehículos son empleados para accionar *air-bags* en caso de choque.
- En edificios han sido incluidos para medir inclinaciones a causa de temblores.
- También han sido incluidos en mandos para el control de video juegos, en teléfonos móviles, etc.

Los acelerómetros se pueden utilizar para medir la vibración en los coches, las máquinas, los edificios, los sistemas de control de proceso y las instalaciones de seguridad.

Pueden también ser utilizados para medir actividad sísmica, la inclinación, la vibración de la máquina, la distancia dinámica y la velocidad con o sin la influencia de la gravedad.

Usos para los acelerómetros que miden gravedad, en donde un acelerómetro se configura específicamente para el uso adentro gravimetría, se llaman gravímetros.

Los acelerómetros se están incorporando cada vez más en los dispositivos electrónicos personales tales como jugadores de medios, los dispositivos del juego, o los contadores del paso. Smartphones y ayudantes digitales personales (por ejemplo Diamante del tacto de HTC, Apple iPhone, tacto del iPod y Nokia N95) contenga los acelerómetros para el control del interfaz utilizador, e.g., el cambiar entre el retrato y los modos del paisaje. Computadoras portátiles de Apple desde la característica 2005 un acelerómetro conocido como Sensor de movimiento repentino, en caso de que se utiliza proteger contra desplomes del disco duro choque.

En reguladores del juego tales como Telecontrol de Wii los acelerómetros pueden proporcionar control realista del juego.

Una más de las aplicaciones del campo común para los acelerómetros están adentro bolsa de aire sistemas del despliegue para los automóviles modernos. En este caso los acelerómetros se utilizan para detectar la aceleración negativa rápida del vehículo para determinarse cuando ha ocurrido una colisión y la severidad de la colisión. El uso extenso de acelerómetros en la industria del automóvil empujó su coste hacia abajo dramáticamente.

Existen comercialmente básicamente dos tipos de acelerómetros: de dos y de tres ejes. Igualmente, de acuerdo a su forma de operar, pueden ser de salida digital o analógica, tal como se muestra en la siguiente figura:

Memsic 2125 Dual-a:



Figura 1.1: Acelerómetro de dos ejes

MMA7455 3-Axis



Figura 1.2: Acelerómetro de Tres ejes

Los Acelerómetros que se dispone para el desarrollo del proyecto son los siguientes:

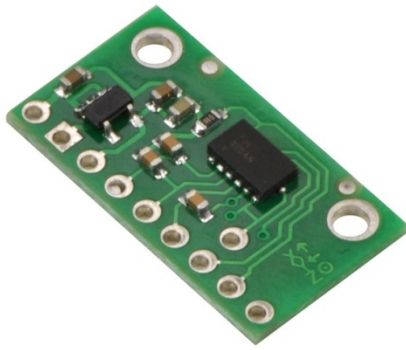


Figura 1.3: Acelerómetro de tres ejes, MMA7361L

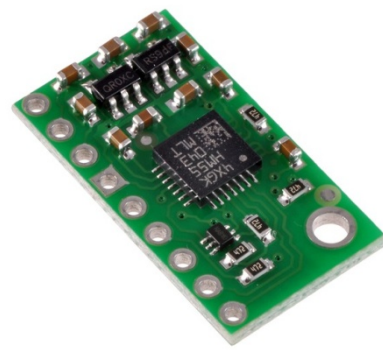


Figura 1.4: Acelerómetro de tres ejes, LSM303DLH

A continuación vamos a mostrar la aplicación del acelerómetro en la tecnología del celular, y el análisis de vibraciones en los equipos y estructuras. [1]

1.3.1.2 Aplicación del Acelerómetro en la Tecnología móvil de celulares.

Como se mencionó anteriormente, el campo de la telefonía móvil, es una de las aplicaciones tecnológicas que más utilizan los acelerómetros,

En telefonía móvil, se utiliza el Acelerómetro para detectar el ángulo de giro del terminal y así poder controlar funciones del móvil como por ejemplo para cambiar el ángulo de visión de una imagen o vídeo al girar el terminal, también en juegos, etc.

Pocket Inclínometer: Pocket Inclínometro es una aplicación capaz de medir la inclinación de una superficie plana utilizando el acelerómetro de tu teléfono celular.



Figura 1.5: Aplicación del Acelerómetro en la Tecnología Móvil de celulares

Rock Bobble Head: Aplicación basada en un objeto con figura de **Elvis** que se mueve en todas las direcciones al agitar tu teléfono celular

También, Apple inc. Utiliza LIS302DL acelerómetro en iPhone y tacto del iPod permitiendo que el dispositivo sepa cuándo se inclina en su lado.

La empresa Nokia, ofrece un acelerómetro 3D (Nokia 5500), se utiliza para el reconocimiento del gesto del golpecito en el interfaz utilizador. Los gestos del golpecito se pueden utilizar para controlar la música para el deportista, por ejemplo para cambiar a la canción siguiente golpeando ligeramente a través de la ropa cuando el dispositivo está en un bolsillo. Nokia N95 y Nokia N70 tienen acelerómetros encajados dentro de ellos. El N70 utiliza un sensor de inclinación para marcar la orientación de la etiqueta en las fotos tomadas con la cámara fotográfica incorporada.

Sony Ericsson W580 y Sony Ericsson W910 también tiene un acelerómetro construido dentro del teléfono que permite la conmutación de pista cuando estas escuchando música, esto es conocido por los usuarios como la característica de la coctelera, también hay otros modelos de celulares tales como el W910 junto con

K850 que puede utilizar la característica del sensor de movimiento en juego, la auto rotación del cuadro UI y muchos otros usos que requieran la característica

Una pequeña cantidad de acelerómetros modernos se utilizan en las computadoras tipo notebook, para alinear automáticamente la pantalla dependiendo de la dirección el dispositivo que la sostiene. Esta característica es solamente relevante dentro de un PC tipo tableta y algunos smartphones. Algunos cuadernos utilizan un acelerómetro para detectar gotas y para preparar la impulsión de disco duro para el impacto, para evitar pérdida de los datos.

1.3.1.3 Aplicación del Acelerómetro en el Análisis de Vibraciones en Máquinas y Estructuras.

Las máquinas tienen piezas en movimiento que producen fuerzas de inercia que excitan el conjunto. Cuando se analizan dichas vibraciones se puede obtener una gran cantidad de información del estado de la máquina en cuestión, pudiendo llegar a predecir el fallo de algunos componentes con la suficiente antelación para que los servicios de mantenimiento puedan realizar su sustitución antes de que dicho fallo sea catastrófico.



Figura 1.6: Aplicación del Acelerómetro en el Análisis de Vibraciones

Por otra parte, las máquinas interactúan con las estructuras de soporte haciendo que estas vibren en ocasiones en sus puntos de resonancia, llegando a producirse roturas de soldaduras, pernos, fatiga del material, etc.

Ante estos problemas, el uso de los acelerómetros en este campo, puede ayudar a diagnosticar, corregir o diseñar de forma adecuada las máquinas, estructuras, soportes o elementos aislantes entre ambas para obtener un resultado óptimo.

Uno de los acelerómetros utilizados para este tipo de aplicación es el acelerómetro piezoeléctrico con sellado hermético, modelo 101.01 ha sido creado para la monitorización de vibraciones en ambientes industriales hostiles.



Modelo 101.01-A-2 cable compuesto termoplastico M12

Figura 1.7: Acelerómetro Piezo Eléctrico

Utiliza la técnica industrial estándar ©ICP con dos 2 cables, con alimentación eléctrica estándar y constante 4 mA. La tierra de señal, está aislada de la superficie de fijación y de la caja externa para prevenir bucles de masa. La pantalla de Faraday, limitará la sensibilidad del EMC hasta el mínimo.

El bajo ruido electrónico y un diseño de temperatura compensada, permite un resultado preciso sobre la completa gama de temperatura. [2]

1.3.1.4 Aplicaciones del Inclinómetro

Existen varios principios para construir un inclinómetro, algunos se basan en la deformación de una varilla de acero que experimenta torsión respecto de la inclinación a la que se le somete. Otros inclinómetros están basados en un lastre libre que actúa por la acción de la fuerza gravitacional, y otros se basan en el desplazamiento de una burbuja de aire en un medio líquido confinado (medidor de nivel).

1.3.1.5 Altímetro-Inclinómetro Digital

El Altímetro/Inclinómetro Digital es un instrumento que entrega el valor de la altitud (altura sobre el nivel del mar), y el grado de inclinación con respecto a dos ejes; además, incluye funciones de barómetro y termómetro, pudiendo entregar los valores de presión atmosférica y temperatura ambiente.

Para su funcionamiento, el Altímetro/Inclinómetro utiliza un sensor de presión absoluta, a través del cual es posible conocer la altitud mediante la utilización de la relación que vincula ambas magnitudes; y un acelerómetro, que permite obtener,

mediante la conversión de aceleración estática en grados, la inclinación del objeto en que se monte. La temperatura ambiente es medida a través de un termómetro digital.

El uso del Altímetro/Inclinómetro es recomendado en aplicaciones relacionadas con transporte aéreo de baja altura, o deportes relacionados con vuelo libre. [3]



Figura 1.8: Altímetro-Inclinómetro Digital

CAPITULO 2

2 FUNDAMENTACIÓN TEÓRICA

2.1 DESCRIPCIÓN DEL HARDWARE Y SOFTWARE UTILIZADO.

En este proyecto, para la aplicación mencionada en el capítulo anterior, utilizaremos hardware de la compañía POLOLU y la compañía PARALLAX, y el software con el cual compilaremos y programaremos nuestros integrados es por medio del AVR Studio 4. A continuación se detalla la lista de componentes y programas que utilizaremos para la implementación del mismo:

- Robot Pololu 3pi
- Microcontrolador ATmega328
- Acelerómetro MEMSIC 2125
- Avr Studio 4
- Programador Avr Usb Pololu

2.1.1 DESCRIPCIÓN DEL ROBOT POLOLU 3PI.

El robot 3pi Pololu (fig. 2.1) es una plataforma móvil completa, de alto desempeño que posee dos motores de engranes metálicos, cinco sensores IR, una pantalla LCD de 8x2 caracteres, una alarma y tres botones. Todo esto conectado a un microcontrolador programable en C ATmega168 ó ATmega328.

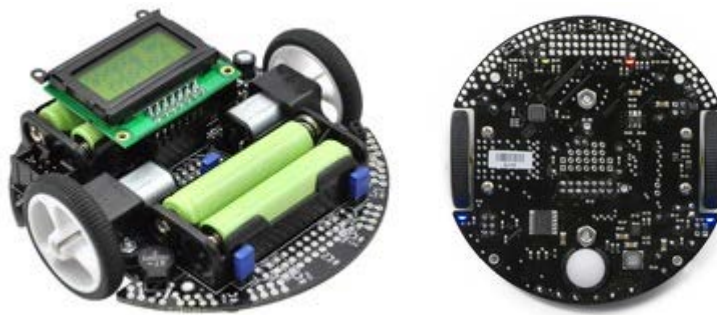


Fig. 2.1: Robot Pololu 3pi

La velocidad del robot es superior a los 3ft por segundo, el 3pi es el primer robot para principiantes con grandes aspiraciones y el segundo robot para quienes buscan pasar de robots no programables a una categoría superior.

El robot 3pi está diseñado para sobresalir en competencias en línea recta ó resolver laberintos.

Tiene un tamaño pequeño (9.5cm/3.7" de diámetro, y pesa 83g/2.9 oz sin baterías) y utiliza 4 pilas AAA (no incluidas).

Un sistema de poder único hace funcionar el motor a 9.25V de manera constante, independientemente del nivel de carga de las baterías.

El voltaje regulado permite al 3pi alcanzar velocidades superiores a los 100 cm/s

mientras da la vuelta de manera precisa sin que esto intervenga con el voltaje de la batería.

El robot 3pi es una gran plataforma para aquellas personas que cuentan con experiencia programando en C y desean aprender robótica, además es muy útil para principiantes ambiciosos que desean aprender a programar en C. En su corazón hay un microcontrolador Atmel que puede ser ATmega168 ó ATmega328P que corren a 20 MHz. Los robots 3pi con el microcontrolador ATmega168 funcionan con una memoria Flash de 16KB, 1KB de memoria en RAM y 512 bytes de memoria EEPROM. Los 3pi con el microcontrolador ATmega328 funcionan con una memoria flash de 32KB, 2KB de RAM y 1KB de memoria EEPROM. El popular compilador gratuito GNU C/C++ trabaja perfectamente con el 3pi, Atmel's AVR Studio provee un cómodo ambiente de desarrollo y el extenso set de librerías de Pololu hace verdaderamente fácil la interface con todo tipo de hardware.

2.1.2 DESCRIPCIÓN DEL MICROCONTROLADOR ATMEGA 328.

El robot Pololu 3pi tiene integrado un microcontralador ATmega 328, que será el microcontrolador que recibirá la señal PWM del acelerómetro, y la interpretará según el código cargado, luego decidirá que acción tomar: aumentar o disminuir la velocidad.

El ATmega 328 es un microcontrolador de 8 bits, CMOS de baja potencia basado en arquitectura RISC. Mediante la ejecución de instrucciones de gran alcance en un solo ciclo de reloj, el ATmega 328 logra tasas de transferencia que se acercan a 1 MIPS por MHz que permite al diseñador del sistema a optimizar el consumo de energía en comparación con la velocidad de procesamiento.

A continuación se menciona algunas de las características del ATmega 328:

Parámetro	Valor
Flash (Kbytes):	32
Pin Count:	28/32
Max. Operating Frequency:	20
QTouch Channels:	16
Hardware QTouch Aquisition:	no
Max I/O Pins:	23
Pb-Free Packages :	MLF (VQFN) 32, PDIP 28, TQFP 32

Como podemos observar en las características antes mencionadas, el empaquetado del ATmega 328 puede ser de tres maneras, MLF (VQFN) 32, PDIP 28 (fig. 2.2), TQFP32. A continuación se muestra una ilustración con la distrución de pines del mismo

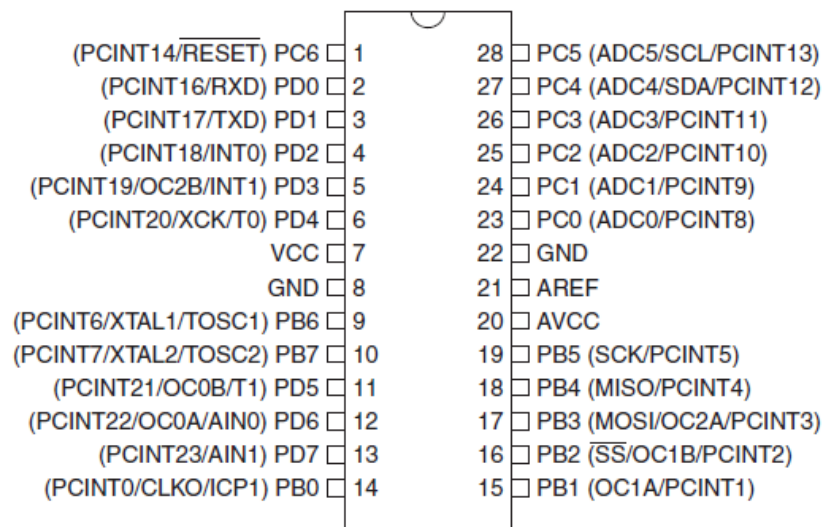


Figura 2.2 : Configuración de pines del ATmega 328

2.1.3 DESCRIPCIÓN DEL ACELERÓMETRO MEMSIC 2125

La aceleración es una magnitud vectorial que nos indica la tasa de cambio de la velocidad por unidad de tiempo. Como se intuye por su nombre, un Acelerómetro es un dispositivo que mide la aceleración y las fuerzas inducidas por la gravedad.

En nuestro proyecto vamos a utilizar un acelerómetro de la tecnología MEMSIC , Parallax, cuyo principio de operación de estos dispositivos se basan en el traspaso térmico, por convección natural.

El MEMSIC 2125 o también llamado MX 2125, es un acelerómetro térmico de doble eje y bajo costo capaz de medir la inclinación, la aceleración, rotación y vibración con un rango de ± 2 g. Para su integración en aplicaciones existentes, el MEMSIC 2125 es eléctricamente compatible con otros acelerómetros populares.

Estos dispositivos miden cambios internos, de la transferencia de calor causada por la aceleración, ofreciendo ventajas significativas sobre el empleo de una estructura tradicional sólida de masas de prueba. Ya que la masa de prueba en el diseño de los sensores MEMSIC (fig. 2.3) son moléculas de gas, las estructuras móviles mecánicas son eliminadas dentro del acelerómetro.

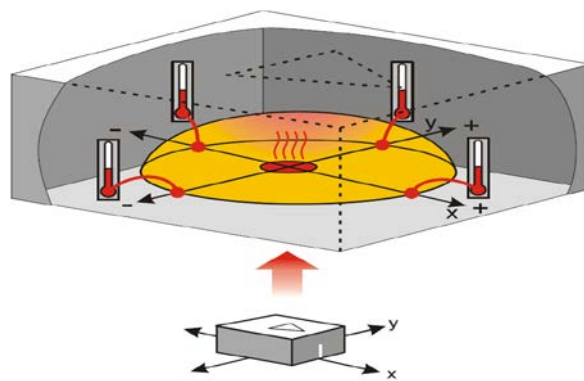


Figura 2.3: Estructura del Acelerómetro Masa Sísmica

Utilizaremos en este caso un acelerómetro de doble eje, es decir, que mide la aceleración en dos ejes perpendiculares y entrega el valor de la aceleración de cada eje en forma de dos señales PWM cuyo tiempo de estado activo (Duty Cycle) será mayor o menor según la aceleración experimentada en ese eje.

2.1.3.1 CARACTERÍSTICAS IMPORTANTES DEL MX 2125

A continuación se listan las características, aplicaciones y especificaciones más importantes del MEMSIC:

Características:

- * Mide ± 2 g en cada eje
- * Simple salida de pulsos de fuerza-g para cada eje
- * Módulo DIP práctico de 6 pines y 0.1 " de espacio
- * Salida analógica para la temperatura (pin Tout)
- * Bajo consumo de corriente en 3,3 o 5 V operación: menos de 4 mA a 5 VCC

Aplicaciones de ejemplo:

- * Detección de inclinación de doble eje para aplicaciones de robótica autónoma
- * Detección de posición de rotación en un solo eje
- * Detección de movimiento / falta de movimiento para los sistemas de alarma
- * Proyectos R / C tales como pilotos automáticos

Especificaciones clave:

- * Requisitos de alimentación: 3,3 a 5 VDC
- * Comunicación: Señal de salida PWM de 100 Hz compatible con TTL / CMOS, con el ciclo de trabajo proporcional a la aceleración
- * Dimensiones: 0,42 x 0,42 x 0,45 (10,7 x 10,7 x 11,8 mm)
- * Rango de temperatura: 32-158 ° F (0 a +70 ° C)

2.1.3.2 CONFIGURACIÓN DE PINES DEL MX 2125

El MEMSIC posee 6 pines (fig. 2.4), el pin 1 es una salida de temperatura análoga, el pin 2 es la salida PWM del eje Y, el pin 3 y 4 son tierra, el pin 5 es la salida PWM del eje X, y el pin 6 es la alimentación. A continuación se muestra una ilustración acerca de la configuración de pines:

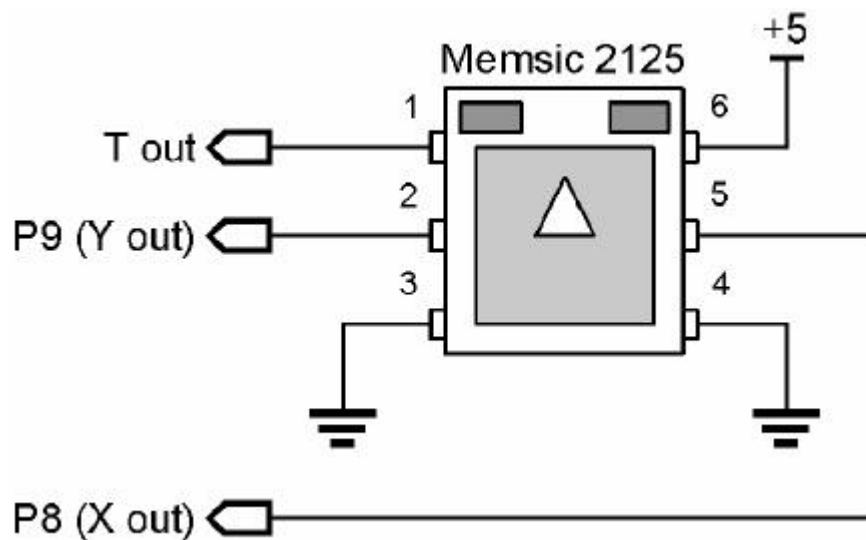


Figura 2.4: Configuración de pines del MEMSIC 2125

2.1.3.3 PRINCIPIO DE FUNCIONAMIENTO DEL MX 2125

El sensor MX2125 (fig. 2.5), consta de un receptáculo cuadrado con un elemento calorífico que calienta una burbuja de gas, y cuatro sensores de temperatura o termopilas a cada lado del receptáculo como se muestra en la siguiente figura.

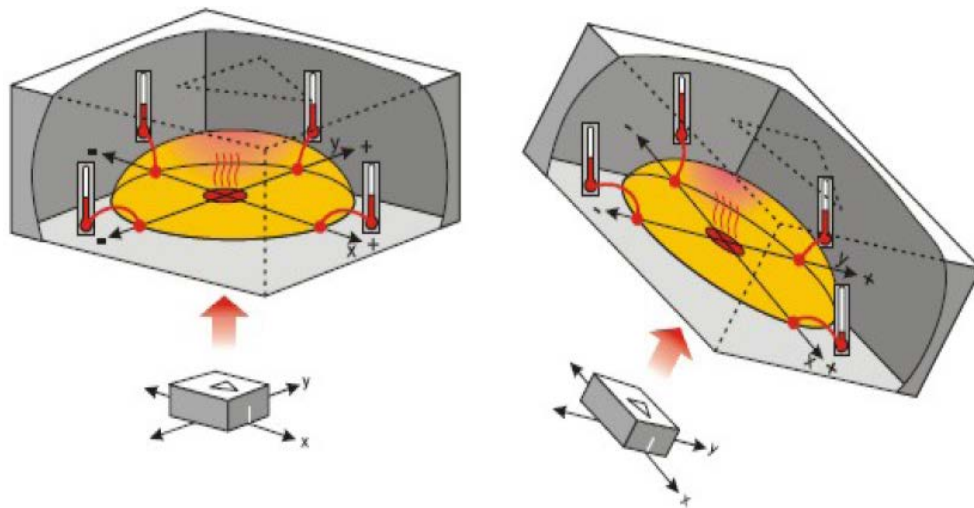


Figura 2.5: Estructura del sensor MX2125

Cuando el sensor se encuentra totalmente nivelado, la burbuja de gas caliente se desplaza centrada hacia arriba en el interior del receptáculo. Los cuatro sensores de temperatura o termopilas registran un mismo valor térmico.

Cuando el sensor sufre algún tipo de giro en cualquiera de sus ejes, la burbuja de gas se desplaza en el interior de la carcasa provocando un aumento de temperatura en algunas de las termopilas y reducción de temperatura en otras. Comparando esas temperaturas se detecta tanto la aceleración dinámica como estática (gravedad y giro).

La electrónica integrada en el sensor MX2125 convierte las medidas de temperatura en señales PWM fácilmente manejables por cualquier controlador (PIC, Atmel, Basica Stamp, Arduino, etc..).

2.1.3.4 PROTOCOLO DE COMUNICACIÓN DEL SENSOR MX2125

Cada eje proporciona una señal PWM (fig. 2.6), como la mostrada en la figura , con una frecuencia de 100Hz (periodo de 10000 μ S) cuya anchura es proporcional a la aceleración. Cualquier tipo de controlador puede medir la duración de la anchura del pulso y obtener así los valores de aceleración.

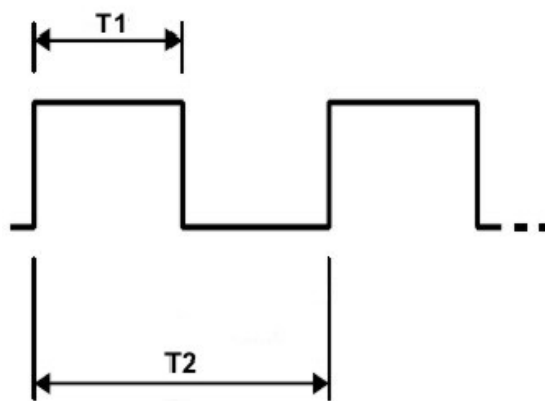


Figura 2.6: Gráfica que muestra la señal PWM de salida del sensor MX2125

Con una alimentación de +5V, una aceleración de 0g se corresponde con el 50% del ciclo útil aunque puede fluctuar entre 48.7% y 51.3%. Esto significa que si la señal PWM de salida tiene un periodo de 10000 μ S (100 Hz), una anchura del ciclo útil de 5000 μ S (50%) se corresponde con una aceleración de 0 g. La aceleración se calcula con la siguiente fórmula proporcionada por el fabricante:

$$A(g) = ((T1 / T2) - 0.5) / 12.5 \%$$

De cara a emplear un microcontrolador, la siguiente ecuación es equivalente y facilita los cálculos que hay que realizar:

$$A(g) = (((T1 / 10) - 500) * 8) / 1000$$

2.1.3.5 CÁLCULO DEL ÁNGULO DE INCLINACIÓN DEL SENSOR

MX2125

Mediante el acelerómetro obtenemos la señal proveniente del eje X, ésta señal PWM la capturamos y luego mediante funciones ya definidas por las librerías del 3pi son procesadas por el microcontrolador, luego para hallar el ángulo de inclinación hacemos uso de ciertos cálculos explicados con mayor detalle en este capítulo(fig. 2.7).

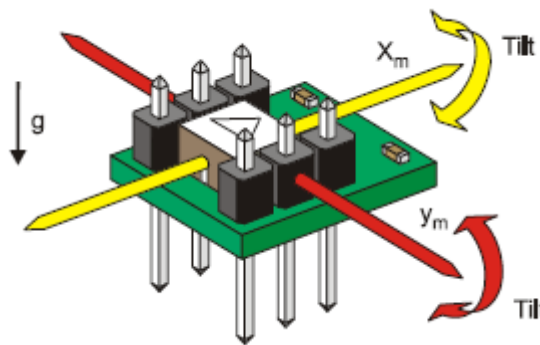


Figura 2.7: Gráfica que muestra los ejes del sensor MEMSIC

El seno de un ángulo (θ) en un triángulo rectángulo (fig. 2.8), es la relación entre el lado opuesto de una triángulo rectángulo (y) dividido por la hipotenusa (h). Si conocemos los valores de X , se puede utilizar el seno inverso o arco seno para hallar el ángulo θ .

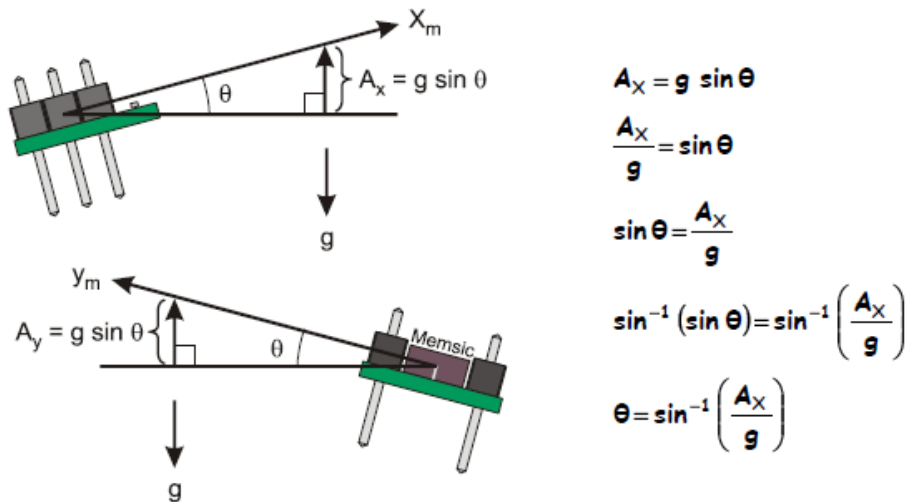


Figura 2.8: Cálculo del ángulo de inclinación del sensor MX2125

Si vemos el módulo del acelerómetro MEMSIC desde un costado podemos observar que la componente de la gravedad actúa sobre el eje de la componente de aceleración de X, que es $A_x = g \sin(\theta)$, por lo tanto, para hallar (θ) debemos despejar de la fórmula que finalmente quedaría:

$$(\theta) = \sin^{-1}(A_x/g)$$

La función arco seno se puede utilizar para determinar el ángulo de inclinación del acelerómetro utilizando la coordenada en X.

Debemos recordar que el ángulo (θ) , luego de hacer la operación del arco seno, está en radianes y para convertirlo a grados, se debe multiplicar por 180 y luego ser dividido para pi (3.14).

2.1.4 DESCRIPCIÓN DEL PROGRAMADOR AVR USB POLOLU

Del mismo modo que el controlador robótico Orangután y el robot 3pi, este dispositivo es un programador para controladores basados en AVR.

El programador emula un AVRISP v2 en un puerto serial, haciéndolo compatible con software de programación estándar AVR.

Dos funciones adicionales ayudan a construir y depurar errores en los proyectos: Un puerto serial TTL para propósitos de comunicación en general y un osciloscopio para monitorear señales y niveles de voltaje.

El programador USB AVR de Pololu (fig. 2.10), es un sistema ISP compacto y económico que sirve para programar microcontroladores Atmel AVR.



Figura 2.9: Programador Avr Usb Pololu

Este dispositivo es una solución atractiva para programar controladores basados en AVR, como el Controlador robótico Orangután. El programador USB AVR se conecta al puerto USB de tu computadora (incluye cable) mediante un puerto virtual COM utilizando el protocolo AVRISPV2/STK500. El software de programación puede ser AVRSTUDIO o AVRDUDE. El programador tiene incluido un cable de programación de 6-pins que se conecta al dispositivo destino (no soporta las viejas conexiones ISP de 10- pins, sin embargo es posible construir o comprar un adaptador ISP de 6 a 10-pins).

2.1.4.1 MICROCONTROLADORES SOPORTADOS

El programador funciona con todos los AVR's que pueden ser programados por un sistema AVR ISP, sin embargo, no ha sido probado en todos los dispositivos (ha sido probado con el controlador robótico Orangután y el robot 3pi). El programador posee un firmware que puede ser actualizado en el futuro para trabajar con nuevos dispositivos.

Actualmente NO funciona con la línea de microcontroladores Atmel XMega. Este programador utiliza un Bus USB de 5V, diseñado para funcionar con programadores AVR que trabajen con voltajes cercanos a los 5V (es importante resaltar que el programador no funciona como unidad de poder para el dispositivo destino). Puede ser usado para programar AVR's que trabajen con voltajes más bajos si se añaden divisores de voltaje para reducir el voltaje de los pines de salida del programador, a un nivel que sea seguro para el microcontrolador.

2.1.4.2 SISTEMAS OPERATIVOS SOPORTADOS

El programador ha sido probado en Windows XP, Windows Vista y Linux. No es compatible con ninguna versión de Mac OS o con versiones anteriores de Windows. Si usted necesita algo que funcione con Mac OS, o versiones anteriores de Windows, el Programador USB Orangután puede servir (Pololu no brinda soporte operaciones en Mac, sin embargo algunos usuarios lo han utilizado de manera exitosa).

2.1.4.3 FUNCIONES ADICIONALES: PUERTO SERIAL TTL.

Así como el Programador USB Orangután, el programador USB AVR funciona como un adaptador USB-puerto serial con unas cuantas mejorías. El programador se instala como dos puertos COM virtuales: Uno para comunicarse con el software de programación y el otro para el adaptador USB-serial. Esto significa que puedes programar el AVR y depurar los errores al mismo tiempo en el puerto TTL si necesitas de abrir y cerrar el programa terminal. Aunado a esto, mediante las líneas de transmisión (TX) y recepción (RX), el programador permite utilizar los pins A y B como líneas seriales handshaking. La herramienta de configuración del programador permite asignar una línea handshaking a un pin específico, para entrada y salida. El programador también brinda acceso a la unidad de poder del USB de 5V mediante el pin VBUS.

2.1.4.4 FUNCIONES ADICIONALES: OSCILOSCOPIO

El dispositivo puede actuar como un osciloscopio de dos canales con ciertas limitaciones. Funciona con una aplicación especial para Windows y utiliza los pins A y B como entradas análogas de voltaje.

El osciloscopio puede medir voltajes de entre 0 y ~5 V (el límite máximo está determinado por el voltaje del bus USB), si se utiliza un divisor de voltaje externo puede soportar un voltaje mayor.

El osciloscopio tiene 2 modos: 10khz probando entradas análogas de 8-bits ó 20khz probando una entrada análoga de 7-bits y una digital.

2.1.5 DESCRIPCIÓN DEL AVR STUDIO 4

AVR Studio es el entorno de desarrollo integrado gratuito para escribir y depurar aplicaciones en el entorno Windows 9x/NT/200/XP que incluye un ensamblador y un simulador.

El WinAVR es una plataforma software abierta que contiene herramientas de desarrollo para los microcontroladores RISC de Atmel para plataforma Windows. Construida por Eric Weddington incluye el compilador GNU GCC para C y C++ para los AVR.

En la Figura 2.11 aparece el entorno de trabajo de un programa AVR Studio, donde tenemos en el centro de la imagen el código que vamos redactando, y a la derecha aparecen los diferentes módulos sobre los cuales podemos modificar sus registros o visualizar como varían al depurar el programa.

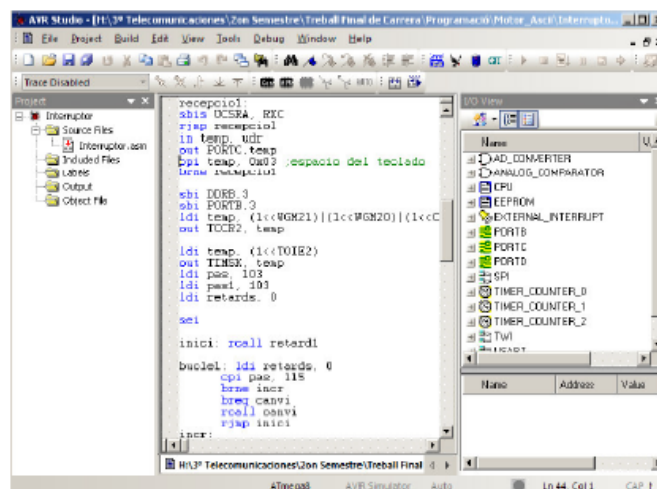


Figura 2.10: Entorno de trabajo del Avr Studio 4.

CAPITULO 3

3.- DISEÑO E IMPLEMENTACION

3.1 Características del Sistema.

El proyecto consiste en sensar inclinación del robot Pololu 3pi por medio del sensor MX2125, el cual envía señales PWM al atmega328 para ser procesadas y decidir si se debe mantener, reducir o aumentar la velocidad de los motores.

El software que controla las acciones del robot pololu 3pi, fue desarrollado en el programa AVR Studio 4, que fue utilizado también para programar el ATmega328.

3.2 Descripción de los Bloques Funcionales

El hardware está compuesto por 3 bloques funcionales (fig. 3.1), el microcontrolador ATmega328 que viene por defecto en el robot Pololu 3pi, el sensor MEMSIC 2125 que fue acoplado con el propósito de sensorar la inclinación, y el robot Pololu 3pi.

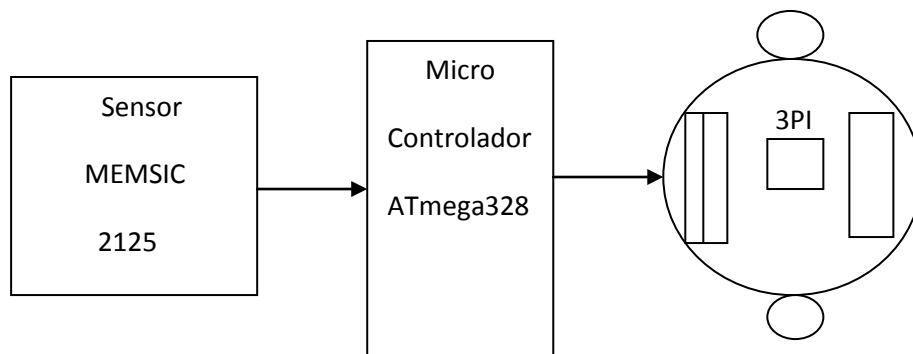
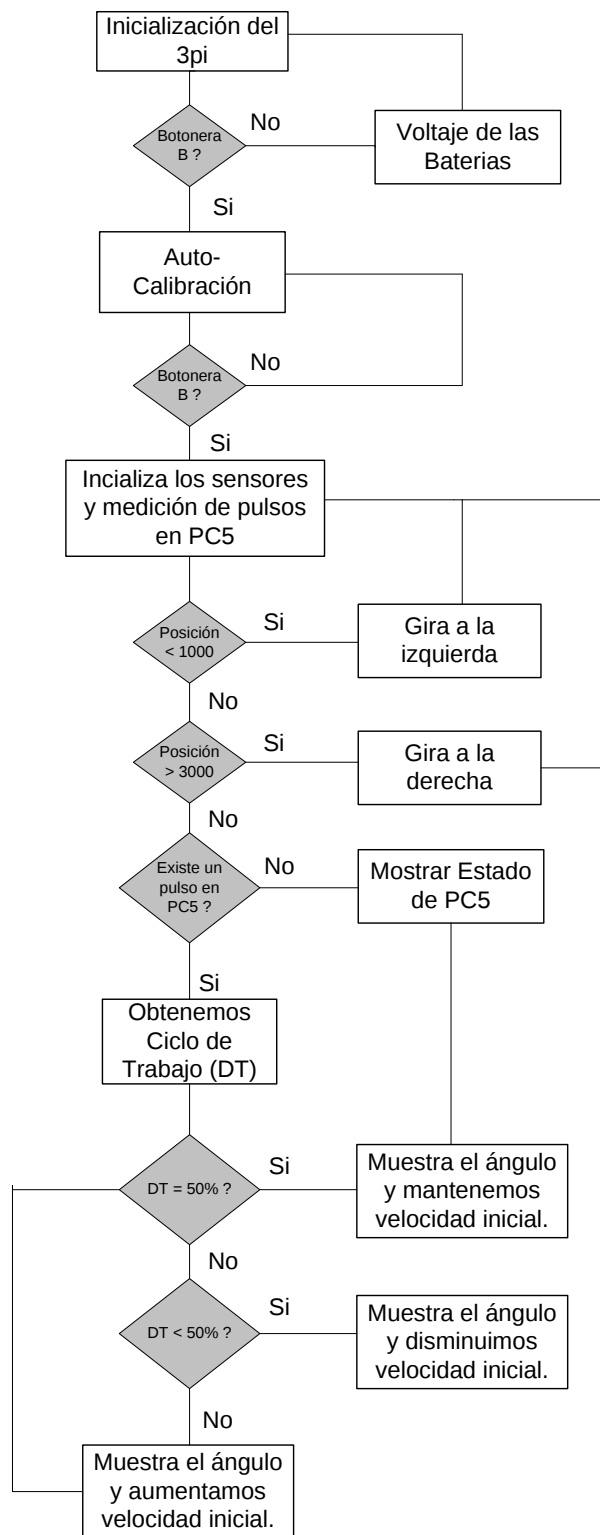


Figura 3.1: Bloques Funcionales del proyecto

3.3 Diagrama de Flujo de nuestro Proyecto.

El presente Diagrama de Flujo nos va a indicar el funcionamiento lógico del proyecto, así como las rutinas y preguntas que se va formulando a medida que va avanzando.



En el diagrama de flujo se puede observar que en primera instancia, se observa un mensaje de inicialización del 3pi, mostrando los nombres de los estudiantes, grupo de trabajo y el nombre de la materia de graduación.

Luego, preguntamos si la botonera “B” fue aplastada, en caso afirmativo, iniciará la rutina de Auto-calibración de los motores; en caso negativo, se mantendrá en la rutina que muestra el estado actual de las baterías.

A continuación se espera a que la botonera “B” sea aplastada de nuevo para activar los sensores y comenzar a medir los pulsos que llegan a PC5, caso contrario se mantendrá mostrando la rutina de calibración de los sensores.

Una vez activados los sensores, entraremos en un lazo infinito que representa que el Pololu 3pi se mantendrá en constante movimiento hasta que sea apagado.

A partir de ese momento, se preguntará por la posición que se encuentra con respecto a la línea que debe seguir, ya sea si está muy a la derecha, muy a la izquierda o si está en la posición correcta para seguir avanzando recto.

Cuando el Pololu 3pi se encuentra muy a la derecha, se recorre la rutina que indica a los motores que giren a la izquierda con el fin de volver a alcanzar la línea y mantenerse en ella. En cambio si se encuentra muy a la izquierda, se recorre la rutina que indica a los motores que giren a la derecha.

Si el Pololu 3pi se encuentra sobre la línea, entonces ordenamos a los motores mantener la misma velocidad en ambas ruedas y ahora se preguntara por el estado de la inclinación del 3pi. Para esto hacemos uso del Ciclo de trabajo de la señal que nos ofrece el MEMSIC, el cual, si es mayor al 50% nos indica que estamos en una pendiente ascendente, por lo tanto tomamos la decisión de aumentar la velocidad para que logre subirla sin problema alguno.

Si el Ciclo de trabajo es menor al 50%, entonces nos encontramos en una pendiente descendente, por lo cual disminuimos la velocidad del Pololu. Si el Ciclo de trabajo es 50% entonces nos encontramos en una superficie horizontal, por lo que mantenemos la velocidad constante.

3.4 Diagrama Esquemático del Pololu 3pi

Pololu 3pi Robot Simplified Schematic Diagram

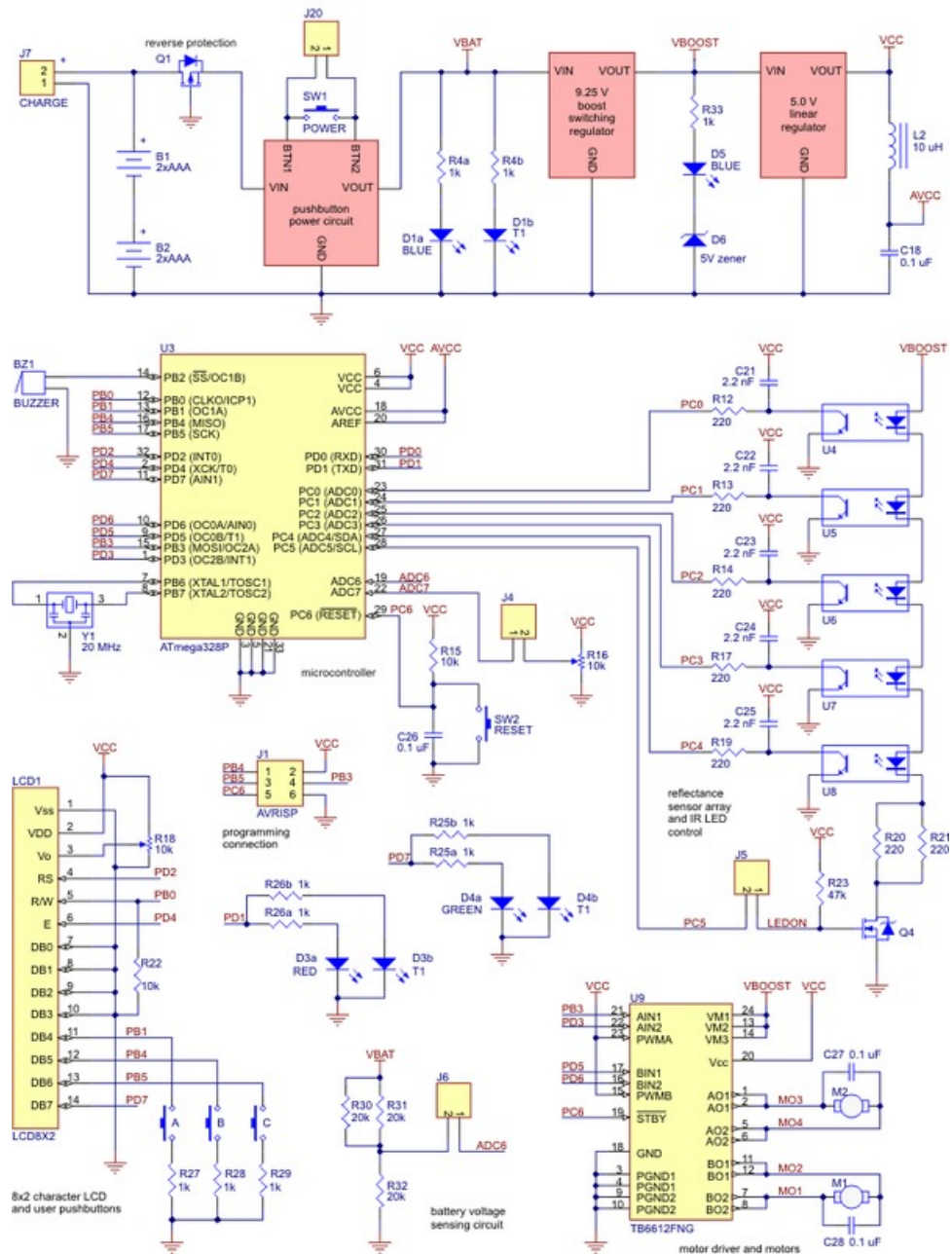


Figura 3.2: Diagrama Esquemático

3.5 Programa desarrollado en el microcontrolador ATmega328.

El MX2125 es un acelerómetro de doble eje que está optimizado para medir variaciones de $\pm 2G$ y entrega a su salida dos señales PWM, una por cada eje.

Para la lectura de la señal PWM se han utilizado ciertas funciones encontradas en la carpeta de ejemplos del Win-Avr, por citar algunas, la función:

- ***unsigned long get_last_high_pulse(unsigned char channel)***, esta función devuelve la longitud del último pulso en alto recibido por completo en el pin especificado en intervalos de tiempo (escala de 0.4us). Si el último pulso en alto fue mayor a 28.6 minutos, el valor mostrado por esta función estará desbordado y la lectura será incorrecta.
- ***void get_current_pulse_state(unsigned char channel, unsigned long* pulse_width, unsigned char* state)***, esta función nos da información acerca de lo que ocurre en el pulso que está sensando en el pin indicado. La información que provee es el ancho del pulso y el estado del mismo, si está en alto o bajo.
- ***unsigned long pulse_to_microseconds(unsigned long pulse_width)***, convierte la longitud del pulso que le llega como parámetro de intervalos sistemáticos a microsegundos, al multiplicar la longitud del pulso por 0.4us.

Una vez capturada la señal PWM, procedimos a convertirla en grados mediante el uso de una fórmula del fabricante:

$$A(g) = (((T1 / 10) - 500) * 8) / 1000$$

Con la cual obtuvimos la aceleración en el eje X y luego utilizando el arco seno obtuvimos el ángulo de inclinación mediante:

$$(\theta) = \sin^{-1} (Ax/g)$$

A continuación detallamos el código cargado en el microcontrolador del Pololu 3pi:

```
// El archivo include del 3pi debe ser una parte importante en el comienzo
// de cualquier programa que use las librerías del Pololu AVR y 3pi.

#include <pololu/3pi.h>

// Este archivo include permite los datos ser almacenados en el program space.
// El ATmega328 tiene 32k de program space, hay que ser cuidadosos cuando se
// colocan datos dentro de él, ya que podemos sobrepasar su capacidad.

#include <avr/pgmspace.h>

#include <math.h>

#define pi 3.141592 // Definimos como constante el valor de pi.

// Mensajes de introducción. El identificador "PROGMEM" nos ayuda a que los datos vayan
// dentro del program space.

const char welcome_line0[] PROGMEM = "Microcon";
const char welcome_line1[] PROGMEM = "Avanzados";
const char welcome_line2[] PROGMEM = "Grupo 4";
const char demo_name_line1[] PROGMEM = "Garcia V.";
const char demo_name_line2[] PROGMEM = "Sanchez C.";

// Este arreglo de pines se usa para inicializar las Rutinas de OrangutanPulseIn.
// Para medir pulsos en múltiples pines, se puede agregar más elementos al arreglo.
```

```

const unsigned char pulseInPins[] = { IO_D0 };

// Melodias simples almacenadas en el program space.

const char welcome[] PROGMEM = ">g32>>c32";
const char go[] PROGMEM = "L16 cdeg4";

// Datos para generar los caracteres usados en load_custom_characters y
// display_readings. De la misma forma se almacenan en el program space.
const char levels[] PROGMEM = {
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b11111
};

// Esta función carga los caracteres usados al LCD. Se puede cargar hasta
// 8 caracteres; los usamos para los 7 niveles del gráfico de barras.

```

```

void load_custom_characters()
{
    lcd_load_custom_character(levels+0,0); // no offset, e.g. one bar
    lcd_load_custom_character(levels+1,1); // two bars
    lcd_load_custom_character(levels+2,2); // etc...
    lcd_load_custom_character(levels+3,3);
    lcd_load_custom_character(levels+4,4);
    lcd_load_custom_character(levels+5,5);
    lcd_load_custom_character(levels+6,6);

    clear(); // El LCD debe estar vacia para que los caracteres tengan efecto.
}

// Esta función muestra la lectura del sensor usando un gráfico de barras.
void display_readings(const unsigned int *calibrated_values)
{
    unsigned char i;

    for(i=0;i<5;i++) {

        // Inicializa el arreglo de caracteres que usaremos para el gráfico de barras.
        // Usando el espacio, una copia extra de caracter de una barra, y el caracter
        // 255 (una caja negra), obtendremos los 10 caracteres en el arreglo.

        const char display_characters[10] = {' ',0,0,1,2,3,4,5,6,255};

        // La variable c tendrá valores de 0 a 9, dado que los valores calibrados están
        // en el rango de 0 a 1000, y 1000/101 es 9 usando solo valores enteros.

        char c = display_characters[calibrated_values[i]/101];
    }
}

```

```

        // Muestra el caracter de gráfico de barras.

        print_character(c);
    }
}

// Inicializa el 3pi, muestra un mensaje de bienvenida, lo calibra, y
// toca la melodía inicial.

void initialize()
{
    unsigned int counter; // Se usa como un simple temporizador
    unsigned int sensors[5]; // Un arreglo para mantener los valores de los sensores

    // Esto debe ser llamado en el comienzo del código del 3pi, para mantener
    // los sensores. Usamos un valor de 2000 para el tiemeout , el cual corresponde
    // a  $2000 \times 0.4 \text{ us} = 0.8 \text{ ms}$  en nuestro procesador de 20 Mhz.

    pololu_3pi_init(2000);
    load_custom_characters(); // Carga los caracteres usados

    // Toca la melodia inicial y muestra el mensaje

    print_from_program_space(welcome_line0);
    lcd_goto_xy(0,1);
    print_from_program_space(welcome_line1);
    play_from_program_space(welcome);
    delay_ms(1000);

    clear();

```

```
print_from_program_space(welcome_line2);

delay_ms(1000);


clear();

print_from_program_space(demo_name_line1);

lcd_goto_xy(0,1);

print_from_program_space(demo_name_line2);

delay_ms(1000);


// Muestra el voltaje de las baterías y espera hasta que se presione el botón


while(!button_is_pressed(BUTTON_B))
{
    int bat = read_battery_millivolts();


    clear();

    print_long(bat);

    print("mV");

    lcd_goto_xy(0,1);

    print("Press B");


    delay_ms(100);
}


// Siempre espera hasta que se suelte el botón, de tal forma que el 3pi
// no realice acción alguna que sea soltado.


wait_for_button_release(BUTTON_B);

delay_ms(1000);
```

```

// Auto-calibración: gira a la derecha y a la izquierda calibrando
// los sensores.

for(counter=0;counter<80;counter++)
{
    if(counter < 20 || counter >= 60)
        set_motors(40,-40);
    else
        set_motors(-40,40);

    // Esta función guarda un conjunto de lecturas del sensor y
    // mantiene monitoreando máximo y mínimo valor encontrado.
    // El argumento IR_EMITTERS_ON significa que el IR LEDs se
    // enciende durante la lectura.

    calibrate_line_sensors(IR_EMITTERS_ON);

    delay_ms(20);
}
set_motors(0,0);

// Muestra los valores calibrados en un gráfico de barras.

while(!button_is_pressed(BUTTON_B))
{
    // Lee los valores de los sensores y obtiene la medición de la
    // posición.

    unsigned int position = read_line(sensors,IR_EMITTERS_ON);

```

```

// Muestra la medición de la posición, la cual va de 0 (cuando
// el sensor de mas a la izquierda está sobre la línea) a
// 4000 (cuando el sensor de mas a la derecha está sobre la línea)
// en el 3pi, además del gráfico de barras de las lecturas del
// sensor. Esto nos permite saber que el robot está listo.

clear();

print_long(position);

lcd_goto_xy(0,1);

display_readings(sensors);

delay_ms(100);

}

wait_for_button_release(BUTTON_B);

clear();

print("Go!");

// Toca una melodía y espera que termine antes de empezar a manejar.

play_from_program_space(go);

while(is_playing());

}

// Esta es la función principal, donde el código comienza.

int main()

{

```

```

unsigned int sensors[5]; // Un arreglo para mantener los valores de los sensores

// Inicializa el 3pi

initialize();

pulse_in_start(pulseInPins, 1);    // Comienza la medición de pulsos en PD0

// Este es el lazo principal, el cuál correrá por siempre.

while(1)
{

    // Obtiene la posición de la línea.

    unsigned int position = read_line(sensors,IR_EMITTERS_ON);

    if(position < 1000)
    {

        // Estamos muy a la derecha de la línea: Gira a la izquierda!.

        // Setea el motor derecho a 100 y el izquierdo a 0, para
        // hacer un giro a la izquierda.

        set_motors(0,100);

        // indicamos la dirección a la cual estamos girando en los LEDs.

        left_led(1);

        right_led(0);

```

```

    }

    else if(position < 3000)

    {

        // Estamos cerca de centrarnos en la línea: Avanza Recto!.

        unsigned long curPulse; // Longitud del pulso actual en ticks (0.4 us)

        unsigned char state; // Estado actual de la entrada (1 si es alto, 0 si es bajo)

        get_current_pulse_state(0, &curPulse, &state); // Pasa los argumentos como punteros

        // Si mas de 300 ms han pasado desde que el ultimo pin cambió en este canal,

        // indicamos que los pulsos han parado.

        if (pulse_to_microseconds(curPulse) >= 300000UL)

        {

            if (state == HIGH) // Si la línea está actualmente en alto

            {

                lcd_goto_xy(0, 0); // Comienzo de la primera fila del LCD

                print("Pin HIGH ");

                lcd_goto_xy(0, 1); // Comienzo de la segunda fila del LCD

                print(" "); //Limpia la fila con espacios en blanco.

            }

            else

            {

                lcd_goto_xy(0, 0); //Comienzo de la primera fila del LCD

                print(" "); //Limpia la fila con espacios en blanco.

                lcd_goto_xy(0, 1); // Comienzo de la segunda fila del LCD

                print("Pin LOW ");

            }

        }

    }

```

```

    }
}

else if (new_high_pulse(0) && new_low_pulse(0)) // pulsos altos y bajos

{

    float motores = 100; // velocidad inicial de 100 para los motores.

    unsigned long high_pulse = get_last_high_pulse(0); // pulso en alto

    unsigned long period_in_ticks = high_pulse + get_last_low_pulse(0);

    unsigned long T = 0.4 * high_pulse;

    float A, angle; // Variables reales


    //Ciclo de Trabajo (Duty Cycle) = pulso alto / (pulso alto + pulso bajo)

    // lo multiplicaremos por 100 para convertirlo en un porcentaje y le

sumamos la mitad

    // del denominador para el numerador, con el fin de obtener un

resultado redondeado.


    unsigned long duty_cycle_percent = (100 * high_pulse +

period_in_ticks/2) / period_in_ticks;

    if (duty_cycle_percent == 50)

    {

        // Mientras el ciclo de trabajo sea 50%, nos indica que estamos en una

        // superficie horizontal.


        float angle=0;

        clear();

```

```

        set_motors(100,100);    // Motores a velocidad normal

        lcd_goto_xy(0, 0);

        print("Fordward");

        lcd_goto_xy(0, 1);

        print("Tilt:");

        print_unsigned_long(angle);    // Mostramos el ángulo

        left_led(1);

        right_led(1);

    }

    else if (duty_cycle_percent >50)

    {

        // Si es ciclo de trabajo es mayor a 50%, nos indica que estamos en

        // presencia de una inclinación ascendente.

        float A = (((float)T / 10.0) - 500) * 8) / 1000;

        float angle = ((asin((float)A)) * 360.0) / (2 * pi);

        clear();

        set_motors(motores  +  angle,motores  +  angle ); //

Aumentamos la velocidad

        lcd_goto_xy(0, 0);

        print("Up");

        lcd_goto_xy(0, 1);

        print("Tilt:");

        print_unsigned_long(angle);    //Mostramos el ángulo

        left_led(1);

        right_led(1);

    }

```

```

else if (duty_cycle_percent < 50)
{
//Si el ciclo de trabajo es menor al 50%, nos indica que estamos en
// presencia de una inclinación descendente.

float A = (((float)T / 10.0) - 500) * 8) / 1000;

float angle = ((asin((float)A) * -1) * 360.0) / (2 * pi);

set_motors(motores - angle ,motores - angle); // Disminuimos
la velocidad

clear();

lcd_goto_xy(0, 0);

print("Down");

lcd_goto_xy(0, 1);

print("Tilt:-");

print_unsigned_long(angle);      //Mostramos el ángulo

left_led(1);

right_led(1);

}

}

}

else
{

// Estamos muy a la izquierda de la línea: Gira a la derecha!.

set_motors(100,0);

left_led(0);

right_led(1);

}

}

}

```

3.6 Gráficas tomadas de la salida del MEMSIC en el eje X.

Haciendo uso del osciloscopio, conectamos el MEMSIC a un protoboard, y medimos la señal de salida del pin 5, que corresponde a la salida PWM del eje X, luego inclinamos el protoboard y capturamos las señales obtenidas del osciloscopio. A continuación mostramos los resultados obtenidos al ir variando la inclinación en el sensor y vamos a poder apreciar como el ciclo de trabajo (Duty Cycle) de la señal va cambiando.

Grafica cuando el sensor MX2521 está en forma horizontal

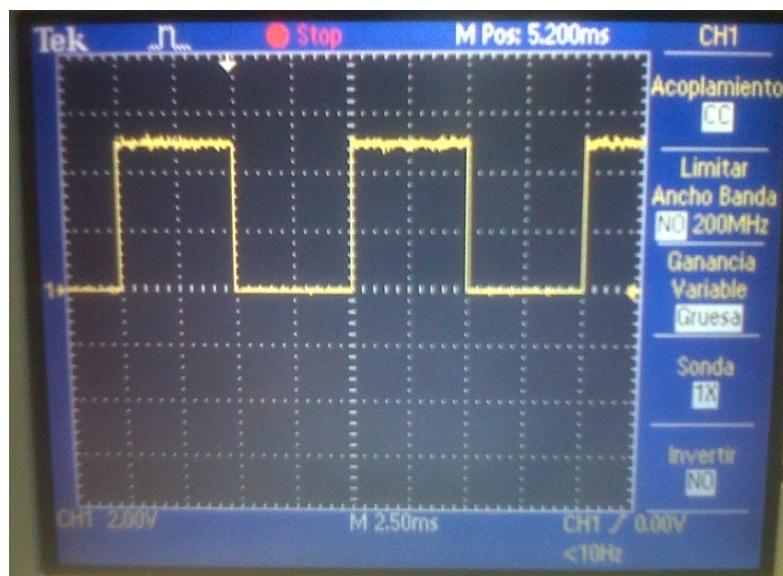


Figura 3.3: Gráfica de señal de salida Sensor en Reposo

Grafica cuando el sensor MX2521 tiene una inclinación en el eje X y obtenemos un valor máximo

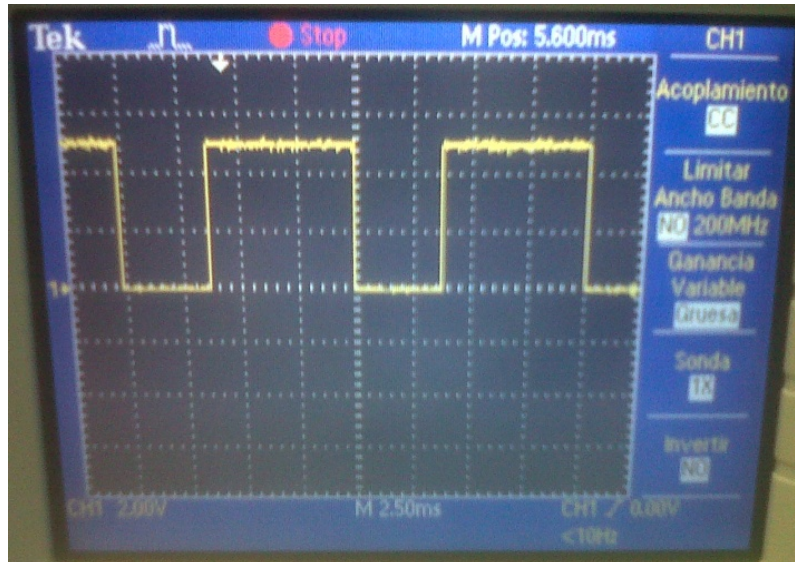


Figura 3.4: Gráfica de señal de salida con inclinación máxima en el eje positivo

Grafica cuando el sensor MX2521 tiene una inclinación en el eje X y obtenemos un valor mínimo

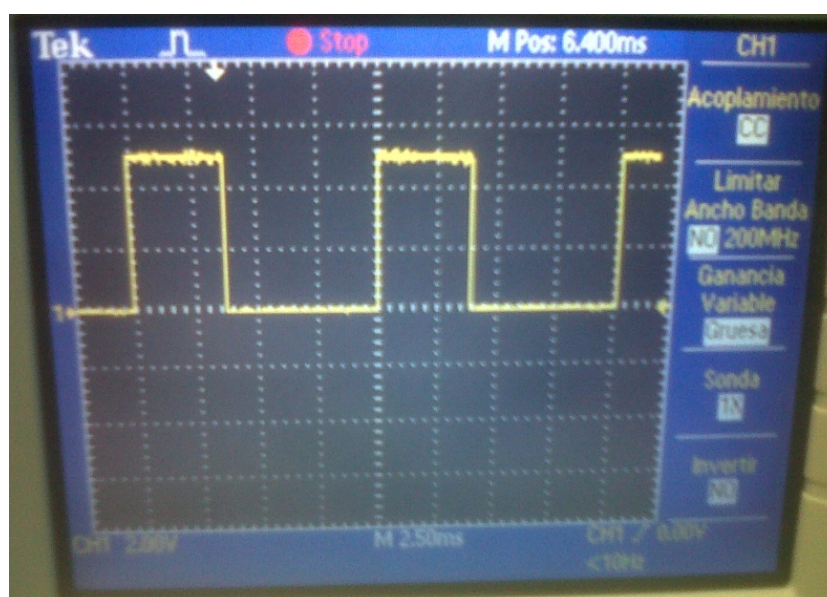


Figura 3.5: Gráfica de señal de salida con inclinación máxima en el eje negativo

3.7 Implementación del Hardware.

Se muestran los resultados de las pruebas realizadas cuando el Pololu 3pi se encuentra en los 3 escenarios propuestos: Pendiente Ascendente, Pendiente Descendente y Superficie horizontal.

En la fotografía 3.6 se puede observar el momento en el que el Pololu 3pi se encuentra sometido a una inclinación ascendente de 45° con respecto a la horizontal:

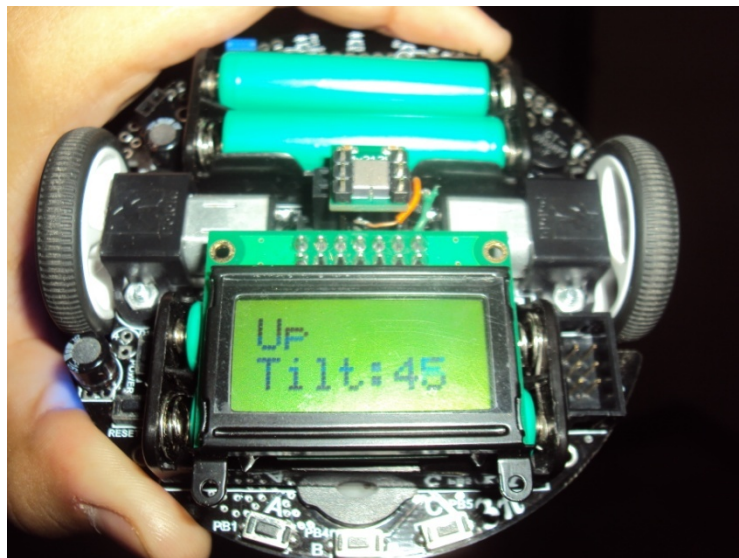


Figura 3.6: Pendiente Ascendente, inclinación de 45°

En la fotografía 3.7 se puede observar el momento en el que el Pololu 3pi se encuentra sometido a una inclinación descendente de 13° con respecto a la horizontal:

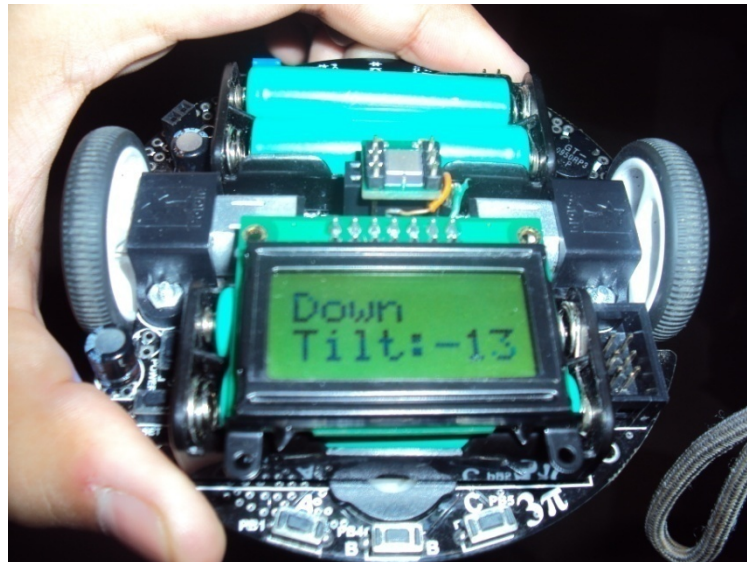


Figura 3.7: Pendiente Descendente, inclinación de -13°

En la fotografía 3.8 se observa el momento en el cual el Pololu 3pi se encuentra sobre una superficie horizontal, por lo cual el ángulo mostrado es de 0° :

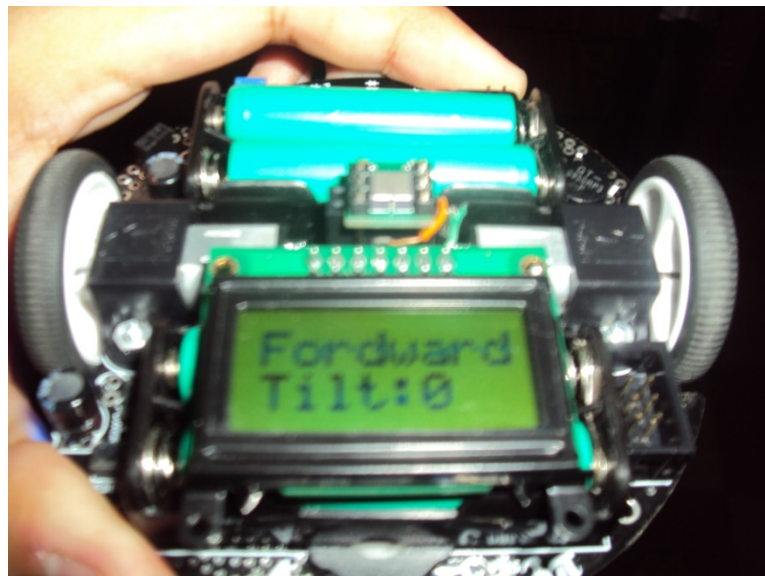


Figura 3.8: Superficie Horizontal, ángulo 0°

En la figura 3.9 se muestra el 3pi sobre la superficie, siguiendo la pista elaborada por medio de la cinta.



Figura 3.9: robot 3pi siguiendo la línea

En la figura 3.10 tenemos otra foto vista desde arriba de la pista diseñada para el seguimiento de línea del robot 3pi, la cual inclinamos cierto ángulo para demostrar la aplicación del MEMSIC, al sensor la inclinación y aumentar o disminuir la velocidad.

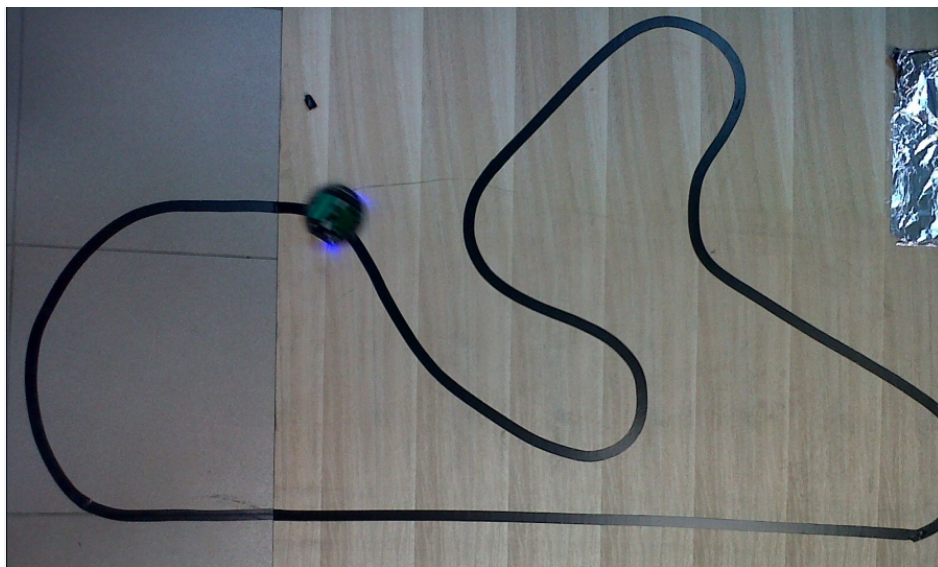


Figura 3.10: Pista diseñada para nuestro proyecto

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. El MEMSIC 2125 es un sensor bastante sensible a los cambios de inclinación en los ejes X y Y. Da mediciones muy precisas por medio de sus señales PWM, en las cuales por medio del ancho de los pulsos o Duty Cycle podemos obtener diversos datos, como por ejemplo: aceleración, inclinación, posición (si conocemos la posición inicial).
2. Sólo se requirió trabajar con el eje X del sensor, debido a que en nuestro proyecto solo nos interesaba medir la inclinación en un eje. El cual conectamos al pin PC5 del robot Pololu 3pi, luego para medir la señal PWM que entraba en ese pin se hizo uso de funciones implementadas en la carpeta de ejemplos del programa AVR Studio 4, con las cuales obtuvimos el ancho de pulso y el estado del pulso en el momento requerido.
3. Con el Ancho de Pulso y el estado del pulso, hicimos uso de la fórmula $A(g) = (((T1 / 10) - 500) * 8) / 1000$, para obtener la aceleración en el eje X. Luego con este dato utilizamos la siguiente fórmula $(\theta) = \sin^{-1} (Ax/g)$, para obtener el ángulo de inclinación. Cabe mencionar que (θ) nos da un ángulo en radianes así que tenemos que transformar ese resultado a grados.

4. Al medir con el osciloscopio la salida PWM del eje X, observamos que el ciclo de trabajo (Duty Cycle), va desde 38% para ángulos cercanos a -90° y llega hasta 63% para ángulos cercanos a $+90^\circ$. Ésta asimetría produce que por cada 1% para los ángulos positivos se aumente 6.92° , mientras que para 1% de los ángulos negativos se aumente 7.5° . Debido a esto, cuando sensamos ángulos negativos hay una pequeña inexactitud comparada a los resultados que obtenemos cuando medimos ángulos positivos.

5. Una vez obtenido el ángulo, hicimos uso de la función `set_motors` para definir la velocidad del robot pololu 3pi, haciendo uso de varios IFs y con esta función para ángulos positivos `set_motors(motores + angle, motores + angle)` para que la velocidad del robot suba gradualmente a medida que sensa un cambio en su inclinación; y con ésta función para ángulos negativos `set_motors(motores - angle, motores - angle)`.

Recomendaciones

1. El sensor puede trabajar en ambientes de hasta 70° C sin inconvenientes, es decir sin variar las señales de salida de PWM, ya que internamente están encapsulados los 4 sensores de temperatura, luego de superar la temperatura máxima el sensor no envía señales fiables.
2. Al momento de implementar el código en el microcontrolador tuvimos ciertos inconvenientes debido a que manejábamos variables que contenían decimales, y estas variables sólo almacenaban el valor entero, para resolver este problema tuvimos que utilizar type cast, que consiste en convertir una expresión de un tipo dado en otro tipo de variable.
3. Para poder conectar la entrada del sensor MEMSIC 2125 al microcontralador ATmega328, fue necesario remover el jumper alojado en el pin PC5, para que pueda ser utilizado como entrada para nuestro proyecto. Dejando de lado la función de medir el voltaje de las baterías.
4. Al utilizar la fórmula para obtener el ángulo, recordar que ese ángulo está dado en radianes y debe ser transformado a grados multiplicando por 180 y dividiendo para pi.
5. En el momento de utilizar la función matemática arco seno para los ángulos negativos nos daba un resultado inconsistente con los valores que esperábamos, para resolver este problema en la fórmula de aceleración multiplicamos el ángulo por -1, y luego agregamos el menos (-) como parte del mensaje.

BIBLIOGRAFÍA

[1] Parallax; Smart Sensors and Applications;

<http://www.parallax.com/Education/TutorialsTranslations/tabid/535/Default.aspx>

[Consulta 3 Mayo 2011]

[2] ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA. *Aplicaciones de un acelerómetro para la medición de inclinaciones horizontales y verticales utilizando microcontroladores avanzados y comunicación serial datalogger e interfaz gráfica.*

Fuente de energía 4 pilas recargables. Nelson Quizhpe; Luis Brito; Carlos Valdivieso

[Consulta 23 abril 2011]. Capítulo 2.

[3] Fuentes Sacha, El Acelerómetro en los móviles,

<http://www.xatakamovil.com/varios/el-acelerometro-en-los-moviles>, fecha de consulta

23 abril 2011

[4] Electrónica Estudio, Pololu Robotic & Electronics,

http://www.electronicaestudio.com/pololu_productos.htm, fecha de consulta 4 Mayo

2011

[5] Atmel, Atmega 328,

http://www.atmel.com/dyn/products/product_parameters.asp?category_id=163&family_id=607&subfamily_id=760&part_id=4720&ListAllAttributes=1, fecha de consulta 1 de Mayo 2011

[6] Pololu Robotics & Electronics, Orangutan Pulse/PWM Inputs,

<http://www.pololu.com/docs/0j18/8>, fecha de consulta 9 de Mayo 2011.

[7] Fisher Christopher Analog Devices, Using an Accelerometer for Inclination

Sensing, http://www.analog.com/static/imported-files/application_notes/AN-1057.pdf, fecha de consulta 9 de Mayo 2011.

[8] Parallax, Memsic 2125 Accelerometer Demo Kit,

<http://www.parallax.com/dl/docs/prod/acc/memsickit.pdf>, fecha de consulta 30 Abril 2011.

[9] Arduino, Lectura de un Acelerómetro MEMSIC,

http://arduino.cc/es_old/Tutoriales/AcelerometroMemsic2125, fecha de consulta 5 Mayo 2011.