



**UNIVERSIDAD YACAMBÚ
VICERRECTORADO ACADÉMICO
FACULTAD DE INGENIERÍA**

**VEHÍCULO PARA PERSONAS CON COMPROMISOS MOTRICES BAJO
EL MODELO DE PÉNDULO INVERTIDO.**

Autor: Sánchez O. Alexis A.

Tutor: Ing. Saul E. Moreno S.

Cabudare, Abril 2012



**UNIVERSIDAD YACAMBÚ
VICERRECTORADO ACADÉMICO
FACULTAD DE INGENIERÍA**

**VEHÍCULO PARA PERSONAS CON COMPROMISOS
MOTRICES BAJO EL MODELO DE PÉNDULO INVERTIDO.**

Trabajo de Grado presentado como requisito parcial para optar al grado de
Ingeniero Electrónico en Computación

Autor: Sánchez O. Alexis A.

Tutor: Ing. Saul E. Moreno S.

Cabudare, Abril 2012

DEDICATORIA

A Dios todopoderoso, sin el este logro habría sido imposible, ha estado presente en cada momento de mi vida, en cada alegría, cada tristeza, cada acierto y desacierto, sin ti Dios, esto habría sido imposible.

A mis Padres, Alí y Mariela, gracias por darme todo su apoyo durante este largo caminar, gracias por no dejar de creer en mí en todo este tiempo y siempre estar allí apoyándome en mis decisiones. Sé que el camino ha sido largo, tortuoso, y a veces se veía imposible, pero ustedes estuvieron allí en todo momento, Gracias. Ah y gracias por dejarme estar tantas horas frente al computador, y esas horas se ven hoy reflejadas, no solo era jugar.

A mi Abuela Sara, con su cariño, amor, ternura cuidados, sus palabras de aliento, sus preocupaciones, este logro es también de usted mi viejita, solo Dios sabe cuánto le agradezco y cuanto te amo.

A mi abuelo Rigoberto, que desde el cielo sé que me está observando, tú me enseñaste responsabilidad, respeto, y tantas cosas, me hubiese encantado verte sentado allí en mi defensa, pero aunque no te veo, se que estas allí junto a mí, te amo mi viejo.

A mi hermanita Saraly, sabes el gran apoyo que has sido en estos años, y tus palabras de aliento, siempre están presentes cuando más las necesitaba. Ahora estoy más cerca de llamarte “colega”.

Daniel y Maribel, hermanos que desde siempre están allí cuando se les necesita. Gracias por todo.

Jesús Daniel y Will vivo recuerdo diario de la inocencia y del amor, su sola presencia siempre alegras mis días.

A Katherine Nieto, mi novia y amor todos estos años, gracias por enseñarme tantas cosas y ser inspiración para ser mejor persona cada día. A ti te debo muchísimo, has estado conmigo durante todas las alegrías y tristezas de este caminar, has sufrido conmigo lo complicado de esta carrera, este logro también es tuyo.

Amigos y Amigas, como María Cote y Familia, Anyelic, Germán, Tigre, Atilio, Norelis, Mono, Jorge Kawaty, Agnew, Catherine, Chirgua, Pablo, Tatiana, Eleazar, Yenio, Regulo, Miguel, Karen, amigos que en un momento estuvieron allí cuando más los necesite durante mi carrera, sin lugar a dudas este logro también es de Ustedes.

A Génesis Rodríguez y Mamá Mercedes gracias por “jallerearnos” toda la carrera, muchas veces a la carrera.

A Roximar tu amistad incondicional que siempre ha estado allí desde hace tanto tiempo, siempre atenta a las necesidades de los demás.

A Fernando y Familia, cuantos momentos duros no vivimos a lo largo de esta carrera, y hoy vemos el fruto, peleas, alegrías, de todo un poco.

A Jesús PELUO, Amigo, compañero y en ocasiones hasta compañero de trabajo, poco a poco hemos demostrado del material que estamos hechos a pesar de las dificultades, gracias por creer en mis tantas veces.

Luis Anzola, Manuel, Jesús Pelón, su amistad incondicional no la cambiaría por nada, las alegrías vividas siempre superaran las adversidades.

Carlos Salazar, cuantos trasnochos, peleas y demás... pero hoy dan sus frutos, y aunque las ocupaciones nos alejan la amistad siempre será la misma.

Marvel Alberto y Familia, y creo que familia es realmente a TODA... la señora Ana, el Señor Marvel, tías, tíos, kathy, Juana, hermanos y demás... a todos les agradezco el apoyo que me brindaron durante todo este tiempo, las atenciones y hasta aguantarme, gracias por abrir las puertas de su casa para trabajar por esta meta, a toda la familia Prieto Muchísimas Gracias.

Y a todos aquellos que hoy no están junto a mí, pero han sido fuente de inspiración, de superación, y muchas veces sin saberlo han contribuido a este logro.

Y Finalmente a todos los que creyeron en mí y no creyeron en mí, de todos he sacado fuerzas para llegar a este punto. Ha sido un caminar muy largo lleno de tropiezos, cambios, amores y desamores, alegrías y tristezas, aciertos y desaciertos, pero todo eso me ha enseñado a valorar mucho mas esta meta, que no es más que el comienzo de una nueva etapa.

AGRADECIMIENTOS

En primer lugar agradecer a la Universidad Yacambú, por ser mi Alma Máter, la casa donde he adquirido tantos conocimientos que hoy me hacen ser la persona que soy, entre sus paredes he vivido momentos que jamás olvidare, y aprendí que una vida de éxito se construye con el lema “Aprender haciendo”.

A mis profesores que mas que profesores se han convertido en amigos, Carlos Zambrano, Rosa Leal, María Alejandra Mora, Mario Garrido, Rossana del Orco, Héctor Mendoza, personas que han creído en mí y me han brindado una gran confianza, es un orgullo haber sido más que solo un alumno y ser tratado por ustedes como su igual, su fe en mí ha sido fundamental, incluso en momentos en que yo no la tenía en mí, gracias por todo.

A Saúl Moreno, Profesor, Amigo, Tutor, por triplicado el agradecimiento, todas sus enseñanzas han marcado mucho quien soy, dentro y fuera el ambiente de clase, por enseñarme a ir más allá de lo que se nos puede dar en un salón de clases, gracias por enseñarme que la mejor virtud que puede tener un Ingeniero es la curiosidad y ganas de saber cada día más.

INDICE GENERAL

INDICE DE GRÁFICOS	x
INDICE DE CUADROS	xii
RESUMEN	xiii
INTRODUCCIÓN	1
ESTUDIOS PREVIOS	6
REVISIÓN DOCUMENTAL Y ELECTRONICA	7
Compromiso Motriz	7
Silla de Ruedas	8
Péndulo Invertido	10
Acelerómetro	12
Giróscopo	14
Bus I2C	16
Protocolo de comunicación del bus I2C	17
Definición de términos:	19
La comunicación en más detalle	21
Direccionamiento de dispositivos en el bus I2C	22
Protocolo de programación para el bus I2C	23
Lectura desde un dispositivo esclavo:	25
Filtro de Kalman	26
Arcotangente	27
Control PID	28
Funcionamiento	29
Proporcional	31
Integral	32
Derivativo	33
Motores Eléctricos	35
Principio de funcionamiento	37
Puente H	39
Arduino	40

Péndulo Invertido	41
Modelado Matemático del Péndulo Invertido	42
DESARROLLO DEL TRABAJO DE GRADO	46
Modulo de Adquisición de Datos	46
Sensor	49
Arcotangente	50
Interfaz de comunicación	52
Modulo de Control	53
Alimentación	54
Memoria	56
Entradas y Salidas	56
Comunicaciones	57
Reinicio Automático (Software)	58
Protección contra sobretensiones en USB	59
Características Físicas	60
Modulo de Potencia para Motores	60
L298 – Puente H Dual	61
DIAGRAMA DE BLOQUES Y/O DIAGRAMA DE PROCESOS	63
SOFTWARE DEL SISTEMA	65
Interfaz de Análisis Interactivo	65
Graficado USB	66
PRUEBAS Y RESULTADOS	68
Lecturas IMU3000&ADXL345	68
Aceleración Angular:	71
Señal PWM Motores DC	72
Prueba de Motores DC	72
CONCLUSIONES	74
RECOMENDACIONES	75
REFERENCIAS BIBLIOGRAFICAS	76
Anexos	77

Atmel 168	78
ADXL 345	79
L298	81
IMU 3000	82
EASY ROLLER	84

INDICE DE GRÁFICOS

Gráfico 1: Lesión en el Tobillo	8
Gráfico 2: Silla de Ruedas Antigua	9
Gráfico 3: Silla de Ruedas Convencional	10
Gráfico 4: Pendulo Invertido sobre un carro	11
Gráfico 5: Ejes de Aceleración	12
Gráfico 6: Ejes de Movimiento y polarización Giróscopo	15
Gráfico 7: Bus i2c	17
Gráfico 8: Protocolo i2c	18
Gráfico 9: Lectura i2c	26
Gráfico 10: Arcotangente	28
Gráfico 11: Diagrama en bloques de un control PID	29
Gráfico 12: Proporcional	32
Gráfico 13: Integral	33
Gráfico 14: Derivativo	35
Gráfico 15: Motores DC	36
Gráfico 16: Arduino	40
Gráfico 17: Diagrama de Módulos	46
Gráfico 18: Mediciones de acelerómetro inclinado.	47
Gráfico 19: IMU Fusion Board - ADXL345 & IMU3000	50
Gráfico 20: Triangulo Rectángulo	51
Gráfico 21: Modulo de conversión de Voltajes	52
Gráfico 22: Conexión IMU3000 y Conversión lógico	53
Gráfico 23: Arduino Duemilanove	54
Gráfico 24: L298 y electrónica necesaria	61
Gráfico 25: l298	62
Gráfico 26: Diagrama de Procesos	64
Gráfico 27: Diagrama de Flujo Bajo Nivel	65
Gráfico 28: Interfaz Interactiva	66

Gráfico 29: Graficador USB	67
Gráfico 30: IMU3000&ADXL345 0 Grados	68
Gráfico 31: IMU3000&ADXL345 90 Grados	68
Gráfico 32: IMU3000&ADXL345 180 Grados	69
Gráfico 33: IMU3000&ADXL345 270 Grados	69
Gráfico 34: IMU3000&ADXL345 -30 Grados Eje Y	70
Gráfico 35: IMU3000&ADXL345 +30 Grados Eje Y	70
Gráfico 36: Aceleraciones	71
Gráfico 37: PWM en relación al PID	71
Gráfico 38: PWM para ambos motores, inversamente proporcionales	72
Gráfico 39: Características ATML 168	78
Gráfico 40: Características ADXL345	79
Gráfico 41: Especificaciones Técnicas y Eléctricas ADXL345	80
Gráfico 42: Especificaciones l298	81
Gráfico 43: Características IMU3000	82
Gráfico 44: Mapa de Registros IMU3000	83
Gráfico 45: Características Easy Roller	84

INDICE DE CUADROS

Cuadro 1: Variables y Parámetros del Modelado Matemático.....	43
Cuadro 2: Características Arduino Duemilanove.....	53
Cuadro 3: Prueba Motores DC.....	72

UNIVERSIDAD YACAMBÚ
VICERRECTORADO ACADÉMICO
FACULTAD DE HUMANIDADES

**VEHÍCULO PARA PERSONAS CON COMPROMISOS
MOTRICES BAJO EL MODELO DE PÉNDULO INVERTIDO.**

Autor: Sánchez O. Alexis A.

Tutor: Ing. Saul E. Moreno S.

Fecha: Abril 2012

RESUMEN

El presente trabajo de investigación consta del desarrollo vehículo para personas con compromisos motrices bajo el modelo de péndulo invertido, se abordaran temas desde la historia de los actuales medios de asistencia a personas con problemas de movilidad, estudios matemáticos de sistemas de control que puedan ser implementados usando Arduino como tecnología de computo, extrayendo la información desde un Acelerómetro y un Giróscopo en este caso un IMU de 6 ejes de libertad, así como las ventajas de un modelo basado en el péndulo invertido para solventar problemas de esta naturaleza; logrando ser un instrumento de asistencia para las personas con compromisos motrices totales o parciales para lograr una movilidad con mayor fluidez y solventar parcialmente los problemas actuales que presentan las sillas de ruedas convencionales. Al finalizar se logro desarrollar un modelo a escala del vehículo logrando presentar un péndulo invertido que es capaz de mantener el equilibrio antes pequeñas perturbaciones externas.

Descriptores: PID, Filtro de Kalman, Arduino, Péndulo Invertido, i2c, IMU, acelerómetro, giróscopo.

INTRODUCCIÓN

En la actualidad existen infinidad de afecciones que impiden el desplazamiento autónomo de las personas. Éstas suelen depender de aparatos para asistir su movilidad como por ejemplo las sillas de ruedas, las cuales pueden generar un problema al usuario debido a que requieren de un área amplia de desplazamiento para poder maniobrar fluidamente. Es por ello que se hace necesario redistribuir el espacio que ocupa; muebles, divisiones estructurales y otros, a fin de garantizar la movilidad continua de la persona que usa este aparato, sin que deba abarcar un espacio tan amplio dentro de un área específica.

En retrospectiva, los aparatos y/o muebles para la asistencia del desplazamiento han evolucionado desde la época del Rey Felipe II de España, a quien se le diseñó la primera silla con este propósito documentada. Luego en 1869 se obtiene la primera patente de una silla de ruedas que no era más que un modelo bimanual impulsado por ruedas traseras, sin embargo al poco tiempo surgieron nuevos modelos de tres ruedas y con otras modificaciones. Es así como el primer modelo impulsado eléctricamente data de 1924.

La silla de ruedas, tal y como se conoce hoy, fue creada en 1932 por el ingeniero Harry Jennings para un amigo; juntos formaron la compañía Everest & Jennings y monopolizaron el mercado hasta la década de los 60. Hoy en día este modelo está estandarizado por la Organización de Estándares Internacionales (ISO por sus siglas en Ingles), bajo el estándar ISO 9999:2007.

De esta manera, con la evolución tecnológica, surgen nuevas

herramientas para la automatización y movimiento de estos aparatos, sin embargo su estructura física o modelo de desplazamiento ha sido constante desde hace más de 50 años. Dicho modelo, está basado en cuatro ruedas, dos traseras y dos delanteras, las primeras con un diámetro 5 veces mayor a las delanteras; todas adosadas de manera paralela a una silla.

Los intentos de mejorar la experiencia del uso de estos aparatos no han ido más allá de trabajos en los que, valiéndose de un joystick o de comandos de voz se induce el movimiento de la silla, generalmente en diferentes direcciones, pero sin modificar la estructura física o la organización de los elementos propios de la silla de ruedas.

Más allá del problema del espacio físico requerido para el desplazamiento de las sillas de ruedas convencionales, las personas que se ven forzadas a usar estos aparatos sufren una reducción en su rango de movimiento con respecto al volumen que pueden abarcar, lo cual limita el acceso a no más de 1m de radio aproximadamente.

En vista de toda la problemática esbozada se plantea el desarrollo de un vehículo que asista a la movilidad de las personas que tengan algún tipo de compromiso motriz venga a sustituir a las sillas de ruedas convencionales o a cualquier otro sistema de locomoción actual. Además, que el área de desplazamiento del referido vehículo sea menor, permita al usuario maximizar su espacio de trabajo, y sobre todo sea de alta integración con la mayoría de los entornos que se puede encontrar.

Por lo antes expuesto, se piensa en un modelo de simple, estable y amplia movilidad, y dando como resultado el estudio de modelo del Péndulo Invertido, un péndulo invertido es un péndulo que tiene su masa por encima

de su punto de giro, donde a menudo se implementa con el punto de giro montado en una plataforma móvil para que esta pueda lograr la corrección en caso necesario. Mientras que un péndulo normal es estable cuando cuelga hacia abajo, un péndulo invertido es inherentemente inestable, y debe ser activamente equilibrado con el fin de permanecer en posición vertical, ya sea mediante la aplicación de un esfuerzo de torsión en el punto de giro o moviendo éste en sentido horizontal, como parte de un sistema de retroalimentación.

El objetivo de estudio de este trabajo de investigación es desarrollar un vehículo para personas con compromisos motrices bajo el modelo de péndulo invertido ya que tradicionalmente las personas con este tipo de limitaciones presentan graves problemas para su desplazamiento en entornos de uso diario, siendo ésta una limitante para su desarrollo social que se presenta como consecuencia de la posición que adopta y espacio que ocupa la persona para moverse en este tipo de vehículos.

Para lograr este objetivo se ha planteado un desarrollo por etapas que constan de: estudiar el modelo matemático del Péndulo Invertido como solución a modelos de estabilidad para vehículos personales; desarrollar un Sistema de Control para un Péndulo Invertido que pueda ser implementado desde un Micro Controlador; implementar una etapa de monitoreo electrónico para los sensores ubicados en el vehículo; desarrollar los modelos de control de una etapa Potencia para un motor.

El desarrollo se presenta para dar una alternativa a las tecnologías a las tecnologías que actualmente existen para dar apoyo a la discapacidad, en el marco de un estudio de Control Automático de Motores Eléctricos.

Según las Líneas de investigación de la Universidad Yacambú, en este entorno se desea dar una solución a los problemas de movilidad y rango de movimiento a personas con compromisos motrices, parcial o total, según la lesión que presente el individuo.

De esta manera, se requiere de un medio de transporte que no limite al usuario dentro su entorno habitual y permita una integración más gradual con la sociedad; un dispositivo que permita a la persona simular estar de pie, (en caso de que su afección no le permita esto), garantizando una total autonomía del vehículo, y mayor comodidad del usuario.

De lo anteriormente expuesto, surge la investigación de un nuevo modelo de locomoción, que es el péndulo invertido, y nuevos sistemas de control embebidos e independientes, que servirán de soporte para proyectos en diferentes áreas de investigación de la ingeniería electrónica.

Esta investigación presentará un nuevo modelo de locomoción, de gran versatilidad y estabilidad en diferentes terrenos regulares, siendo este menos afectado por la inclinación del terreno, y requiriendo un mínimo o prácticamente inexistente esfuerzo de parte del usuario. Esto sustituiría el modelo tradicional de la silla de ruedas que lleva actualmente más de 50 años sin sufrir modificaciones de fondo, y se mejoraría la integración, movilidad y acoplamiento del usuario dentro su entorno.

También representará una guía para aquellos investigadores que desean estudiar modelos matemáticos y sistemas de control, dentro de un Micro Controlador, así como nuevos medios de transporte para personas con alguna discapacidad.

En este sentido, se mostrará en un modelo a escala, con todos los dispositivos electrónicos, sensores y actuadores necesarios para un modelo de tamaño real, siendo necesaria sólo la sustitución de los motores y su control de potencia para que pueda soportar a una persona promedio.

ESTUDIOS PREVIOS

Una base importante en el desarrollo de cualquier proyecto son los estudios previos. Se han encontrado trabajos orientados hacia el problema presentado y otros estrechamente relacionados con esta investigación los cuales suministran información que ayudará al desarrollo de este proyecto.

Araujo (2007) en su trabajo “sistema de control para movimientos de una silla de ruedas mediante soplo /aspiración y movimientos del cuello” con una automatización de los movimientos de una silla de ruedas, se intenta facilitar el desplazamiento para personas con cuadriplejia, esto presenta una interesante semejanza al intentar facilitar la locomoción a personas con compromisos físicos que limitan su movimiento. De este trabajo se puede aprender sobre el manejo de motores dc, así como la correcta selección en cuanto al nivel de torque y velocidad para el desplazamiento humano,

También se puede encontrar a Sequera (2008) con el “controlador del péndulo invertido, usando lógica difusa” presenta un desarrollo muy extenso de sistema de control para un péndulo invertido, en este caso para conceptos de laboratorio, usando lógica difusa y controlado desde un ordenador, aunque el péndulo invertido controlado en este trabajo tiene un movimiento limitado, se puede encontrar, durante el desarrollo de este trabajo, con elementos fundamentales como las ecuaciones físicas y modelos matemáticos que rigen el modelo de estudio. A través del desarrollo de este trabajo se puede aprender sobre el análisis del modelo matemático necesario para el equilibrio de un péndulo invertido así como diferentes técnicas de control.

En este orden Imery (2010), con el trabajo “plataforma robótica de

transporte” presenta una solución a todo tipo de personas para la asistencia en la movilidad con una plataforma cuya superficie de movimiento es realmente pequeña, con un gran integración dentro de cualquier entorno, inclusive terreno irregular, su uso de servomotores para la locomoción de su robot poli-articulado, es de gran utilidad para el sistema de seguridad que llevara integrado el desarrollo este trabajo de grado.

Todos estos trabajos de investigación fueron el punto de inicio para la realización de esta investigación, en vista de que poseen lineamientos similares ya que se basan en los fundamentos teóricos y prácticos, así incluso materiales similares necesarios para la realización de esta.

REVISIÓN DOCUMENTAL Y ELECTRONICA

Para el desarrollo de esta investigación, es necesaria la revisión documental y electrónica de los temas referentes al trabajo por lo que se procede al entendimiento de las nociones básicas para lograr avanzar en el desarrollo de este proyecto.

Compromiso Motriz

Las discapacidades motoras se producen porque puede existir un trastorno de la función de cualquier punto del sistema nervioso (células del asta anterior, raíces nerviosas, plexos de las extremidades, nervios periféricos o uniones neuromusculares) o por factores externos que pueden lesionar el músculo o a nivel óseo (ver Gráfico 1).

Estos compromisos por su naturaleza limitan la movilidad de la persona, viéndose forzada a usar medios externos a los propios para asistirse, bastón,

andadera, silla de ruedas, entre otros.



Gráfico 1: Lesión en el Tobillo
(<http://www.diariodecuyo.com.ar>)

Silla de Ruedas

Traumas y enfermedades han creado una necesidad por sillas de ruedas que se extiende muchos años hacia atrás. Algunos comentaristas incluso han sugerido que como la evidencia más antigua de sillas y ruedas data de alrededor de 4000 AC, es posible que las dos fueran combinadas para hacer una silla con ruedas hasta seis mil años atrás. La primera prueba clara de una silla de ruedas, sin embargo, es de una imagen china grabada en 525 DC.

Cualquier historia subsiguiente de la silla de ruedas es difícil de documentar hasta 1595. Este fue el año en que un artista dibujó un borrador

del Rey español, Felipe II de España (1527 - 1598), sentado en una silla que tenía pequeñas ruedas montadas al final de cada pata. Los rasgos de la silla incluían una plataforma levantada para las piernas del Rey y un respaldo ajustable.



Gráfico 2: Silla de Ruedas Antigua
(<http://www.todomercado.com>)

El siguiente desarrollo fue una silla inventada en 1783 por John Dawson. Dawson trabajaba en Bath, Inglaterra, a donde muchos inválidos viajaban para beber y bañarse en las aguas del balneario. La silla "Bath" de Dawson, con su tercera rueda que el ocupante podía dirigir usando una manivela rígida agregada, fue un gran suceso. Hubo un número de versiones, algunas de ellas abiertas, otras con capuchas y frentes de vidrio, pero todas debían ser empujadas desde atrás o tiradas por un pequeño caballo o burro.

A comienzos del siglo veinte, las sillas de ruedas se habían desarrollado aún más y disponían de ruedas con rayos de alambre, respaldos ajustables, y apoyos móviles para brazos y pies. También había modelos livianos hechos de mimbre montados sobre marcos de metal.

En 1915, ingenieros británicos habían producido la primera silla de ruedas motorizada, aunque la mayoría de los usuarios continuaron en las versiones manuales, que se estaban volviendo mucho más baratas. A pesar de esto, las sillas aún eran rígidas y difíciles de guardar y transportar, particularmente

en autos. Pero en 1932, un ingeniero de Los Ángeles llamado Harry Jennings diseñó y construyó una silla plegable para su amigo, Herbert Everest. Los dos hombres inmediatamente vieron el potencial de este invento y establecieron una compañía para producir en masa las nuevas sillas portátiles. Estos fueron los precursores de las sillas de ruedas de uso común hoy en día.



Gráfico 3: Silla de Ruedas Convencional
(<http://www.localtel.com.ve>)

Péndulo Invertido

Dentro del estudio de los robots de piernas o extremidades, lo más importante ha sido el balanceo de estos, pues es la parte esencial para que no caiga el cuerpo del robot.

Las primeras máquinas que se balanceaban activamente fueron automáticamente controladas por péndulos invertidos. Es sabido por cada uno de nosotros que un humano puede balancear una escoba en su mano o en un dedo con relativa facilidad. ¿Por qué no utilizar control automático para construir una escoba (péndulo invertido) que se balancee por si misma?

Claude Shannon fue probablemente el primero en hacerlo. En 1951 usó las partes de un juego erector para construir una máquina que balanceara un péndulo invertido que estaba arriba de un pequeño carrito de poder (ver Gráfico 4). El manejador del carro iba en direcciones atrás y adelante, en respuesta a los tipos de movimiento del péndulo, éste era censado por un par de interruptores en su base. Para lograr moverlo de un lugar a otro, el carro primero tenía que manejarse fuera del destino para poder desbalancear al péndulo, para proceder en dirección del destino. Para balancear nuevamente hacia el destino, el carro se movía pasando el destino hasta que el péndulo se encontraría otra vez completamente perpendicular a la horizontal sin velocidad hacia delante, entonces se movía en sentido inverso del destino y así lo lograba balancear.

La importancia del balance activo para la locomoción no es un problema nuevo, pero el progreso en la construcción de sistemas con extremidades físicas que ocuparan estos principios fue retardado por la evidente dificultad del trabajo. No fue sino a finales de los años setenta que los experimentos en balance comenzaron a salir adelante.

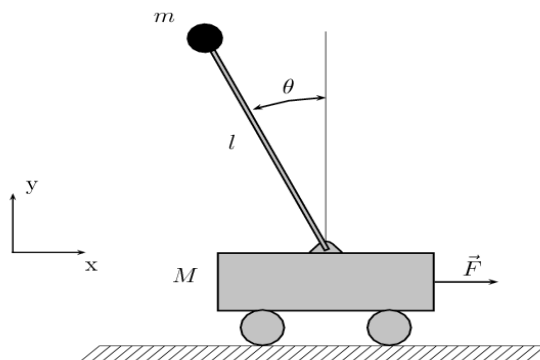


Gráfico 4: Pendulo Invertido sobre un carro

(<http://en.wikipedia.org>)

Acelerómetro

Es denominado acelerómetro a cualquier instrumento destinado a medir aceleraciones. Esto no es necesariamente la misma que la aceleración de coordenadas (cambio de la velocidad del dispositivo en el espacio), sino que es el tipo de aceleración asociadas con el fenómeno de peso experimentada por una masa de prueba que se encuentra en el marco de referencia del dispositivo. Un ejemplo en el que este tipo de aceleraciones son diferentes es cuando un acelerómetro medirá un valor sentado en el suelo, ya que las masas tienen un peso, a pesar de que no hay cambio de velocidad. Sin embargo, un acelerómetro en caída gravitacional libre hacia el centro de la Tierra medirá un valor de cero, ya que, a pesar de que su velocidad es cada vez mayor, está en un marco de referencia en el que no tiene peso.

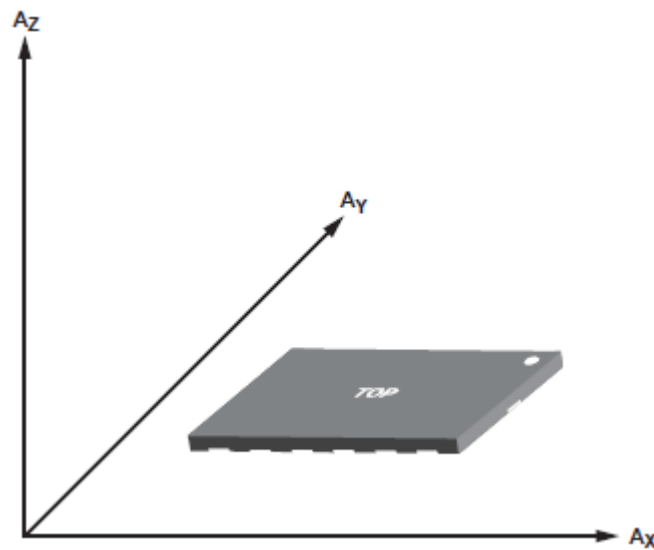


Gráfico 5: Ejes de Aceleración
(Pagina 22, Datasheet ADXL345)

Un acelerómetro mide la aceleración proporcional, que es la aceleración

que experimenta en relación con la caída libre y es la aceleración que sienten las personas y objetos. Dicho de otra manera, en cualquier punto del espacio-tiempo del principio de equivalencia garantiza la existencia de un punto de referencia inercial, y un acelerómetro mide la aceleración con respecto al sistema. Estas aceleraciones son popularmente medidas en términos de fuerza-g .

Un acelerómetro en reposo respecto a la superficie de la Tierra se indican aproximadamente $1g$ al alza, debido a que cualquier punto de la superficie de la Tierra se está acelerando hacia arriba en relación con el punto local inercial (el punto de un objeto en caída libre cerca de la superficie). Para obtener la aceleración debida a la moción con respecto a la Tierra, esta "gravedad offset" debe ser restada y las correcciones de los efectos causados por la rotación de la Tierra con respecto al sistema de referencia inercial.

La razón de la aparición de un desplazamiento gravitacional es principio de equivalencia de Einstein, que establece que los efectos de la gravedad sobre un objeto son indistinguibles de la aceleración. Cuando mantiene fijo en un campo gravitatorio, por ejemplo, aplicando una fuerza de reacción del suelo o un empuje hacia arriba equivalente, el punto de referencia para un acelerómetro, acelera hacia arriba con respecto a un punto de referencia en caída libre. Los efectos de esta aceleración son indistinguibles de cualquier aceleración otra experimentada por el instrumento, de modo que un acelerómetro no puede detectar la diferencia entre estar sentado en un cohete en la plataforma de lanzamiento, y estar en el mismo cohete en el espacio profundo, mientras que utiliza sus motores para acelerar la de $1g$. Por razones similares, un acelerómetro leerá cero durante cualquier tipo de caída libre. Esto incluye el uso de una nave espacial de cabotaje en el

espacio profundo lejos de cualquier masa, una nave espacial que orbita la Tierra, un avión de una parabólica "gravedad cero" de arco, o de cualquier caída libre en el vacío. Otro ejemplo es la caída libre a una altura suficientemente alta para que los efectos atmosféricos se puedan despreciar.

Sin embargo esto no incluye una caída (no libre) en el que la resistencia del aire produce fuerzas de arrastre que reducen la aceleración, hasta constante la velocidad terminal. A la velocidad terminal del acelerómetro indicará 1g de aceleración hacia arriba. Por la misma razón un paracaidista, al llegar a la velocidad terminal, no se siente como si él estuviera en "caída libre", sino que experimenta una sensación similar a la que se apoya (en 1 g) en una "cama" de aire uprushing.

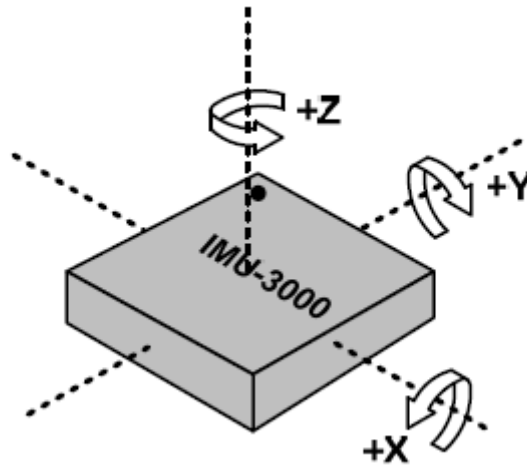
La aceleración se cuantifica en el Sistema Internacional de unidades en metros por segundo cuadrado (m/s^2), en el CGS unidad de galones (Gal), o comúnmente en forma de G-Force (g).

Para el propósito práctico de encontrar la aceleración de los objetos con respecto a la tierra, tales como para uso en un sistema de navegación inercial, se requiere un conocimiento de la gravedad local. Esto puede obtenerse bien mediante la calibración del dispositivo en reposo, o de un modelo conocido de la gravedad en la posición actual.

Giróscopo

El giróscopo o giroscopio es un dispositivo mecánico formado esencialmente por un cuerpo con simetría de rotación que gira alrededor de su eje de simetría. Cuando se somete el giróscopo a un momento de fuerza que tiende a cambiar la orientación del eje de rotación su comportamiento es

aparentemente paradójico ya que el eje de rotación, en lugar de cambiar de dirección como lo haría un cuerpo que no girase, cambia de orientación en una dirección perpendicular a la dirección «intuitiva».



*Gráfico 6: Ejes de Movimiento y polarización
Giróscopo*

(Pagina 14, Datasheet IMU3000)

Se podría suponer un giróscopo formado por un disco montado sobre un eje horizontal, alrededor del cual el disco gira libremente a gran velocidad. Un observador mantiene el eje del fondo con la mano izquierda y el eje de delante con la mano derecha. Si el observador trata de hacer girar el eje hacia la derecha (bajando la mano derecha y subiendo la mano izquierda) sentirá un comportamiento muy curioso, ya que el giróscopo empuja su mano derecha y tira de su mano izquierda. El observador acaba de sentir el efecto giroscópico. Es una sensación muy sorprendente porque da la impresión de que el giróscopo no se comporta como un objeto «normal».

Bus I2C

Abreviatura de Inter-IC (inter integrated circuits), un tipo de bus diseñado por Philips Semiconductors a principios de los 80s, que se utiliza para conectar circuitos integrados (ICs). El I2C es un bus con múltiples maestros, lo que significa que se pueden conectar varios chips al mismo bus y que todos ellos pueden actuar como maestro, sólo con iniciar la transferencia de datos. Este bus se utiliza en muchos dispositivos, en especial en equipos de vídeo como monitores de computadora, televisores y videocaseteras.

El bus I2C, un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa. Fue diseñado a este efecto por Philips y permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

Descripción de las señales

- SCL (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema.
- SDA (System Data) es la línea por la que se mueven los datos entre los dispositivos.
- GND (Masa) común de la interconexión entre todos los dispositivos

"enganchados" al bus.

Las líneas SDA y SCL son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistores "pull-up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

(<http://robots-argentina.com.a>)

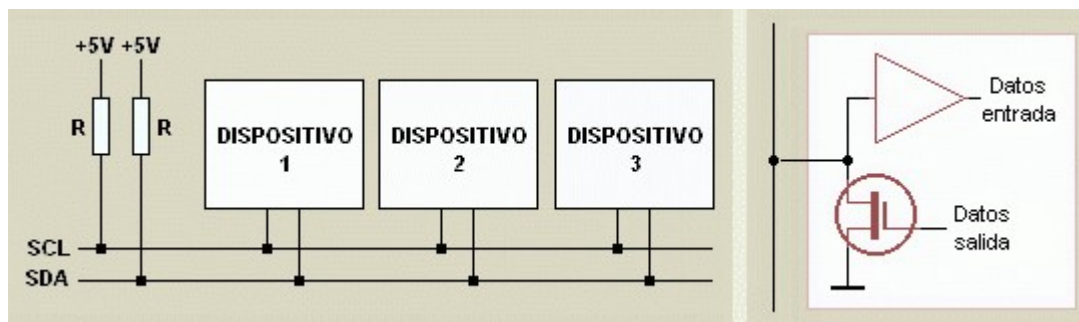


Gráfico 7: Bus i2c

El Gráfico 7 es suficientemente auto explicativo. Las dos líneas del bus están en un nivel lógico alto cuando están inactivas. En principio, el número de dispositivos que se puede conectar al bus no tiene límites, aunque hay que observar que la capacidad máxima sumada de todos los dispositivos no supere los 400 pF. El valor de los resistores de polarización no es muy crítico, y puede ir desde 1K8 (1.800 ohms) a 47K (47.000 ohms). Un valor menor de resistencia incrementa el consumo de los integrados pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales. Los valores más comunes en uso son entre 1K8 y 10K.

Protocolo de comunicación del bus I2C

Habiendo varios dispositivos conectados sobre el bus, es lógico que para

establecer una comunicación a través de él se deba respetar un protocolo. Digamos, en primer lugar, lo más importante: existen dispositivos maestros y dispositivos esclavos. Sólo los dispositivos maestros pueden iniciar una comunicación.

La condición inicial, de bus libre, es cuando ambas señales están en estado lógico alto. En este estado cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de inicio (start). Esta condición se presenta cuando un dispositivo maestro pone en estado bajo la línea de datos (SDA), pero dejando en alto la línea de reloj (SCL).

El primer byte que se transmite luego de la condición de inicio contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura).

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits (A0-A6) está presente en el bus, éste contesta con un bit en bajo, ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro. Este bit de reconocimiento (ACK) en bajo le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse. Aquí la comunicación se establece en firme y comienza el intercambio de información entre los dispositivos. (Ver Gráfico 8)

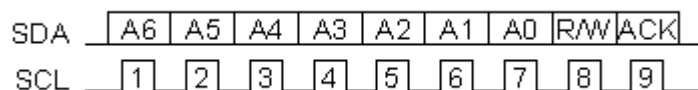


Gráfico 8: Protocolo i2c
(<http://robots-argentina.com.ar>)

Si el bit de lectura/escritura (R/W) fue puesto en esta comunicación a nivel

lógico bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo. Esto se mantiene mientras continúe recibiendo señales de reconocimiento, y el contacto concluye cuando se hayan transmitido todos los datos.

En el caso contrario, cuando el bit de lectura/escritura estaba a nivel lógico alto (lectura), el dispositivo maestro genera pulsos de reloj para que el dispositivo esclavo pueda enviar los datos. Luego de cada byte recibido el dispositivo maestro (quien está recibiendo los datos) genera un pulso de reconocimiento.

El dispositivo maestro puede dejar libre el bus generando una condición de parada (o detención; stop en inglés).

Si se desea seguir transmitiendo, el dispositivo maestro puede generar otra condición de inicio en lugar de una condición de parada. Esta nueva condición de inicio se denomina "inicio reiterado" y se puede emplear para direccionar un dispositivo esclavo diferente o para alterar el estado del bit de lectura/escritura.

Definición de términos:

- **Maestro (Master):** Dispositivo que determina los tiempos y la dirección del tráfico en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".

- **Esclavo (Slave):** Todo dispositivo conectado al bus que no tiene la capacidad de generar pulsos de reloj. Los dispositivos esclavos reciben señales de comando y de reloj generados desde el maestro.
- **Bus libre (Bus Free):** Estado en el que ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Es el único momento en que un dispositivo maestro puede comenzar a hacer uso del bus.
- **Comienzo (Start):** Se produce cuando un dispositivo maestro ocupa el bus, generando la condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.
- **Parada (Stop):** Un dispositivo maestro puede generar esta condición, dejando libre el bus. La línea de datos y la de reloj toman un estado lógico alto.
- **Dato válido (Valid Data):** Situación presente cuando un dato presente en la línea SDA es estable al tiempo que la línea SCL está a nivel lógico alto.
- **Formato de Datos (Data Format):** La transmisión de un dato a través de este bus consiste de 8 bits de dato (1 byte). A cada byte transmitido al bus le sigue un noveno pulso de reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento.
- **Reconocimiento (Acknowledge):** El pulso de reconocimiento, conocido como ACK (del inglés Acknowledge), se logra colocando la línea de datos a un nivel lógico bajo durante el transcurso del noveno pulso de reloj.
- **Dirección (Address):** Todo dispositivo diseñado para funcionar en este bus posee su propia y única dirección de acceso, preestablecida por el fabricante. Hay dispositivos que permiten definir externamente

parte de la dirección de acceso, lo que habilita que se pueda conectar en un mismo bus un conjunto de dispositivos del mismo tipo, sin problemas de identificación. La dirección 00 es la denominada "de acceso general"; a ésta responden todos los dispositivos conectados al bus.

- **Lectura/Escritura (Bit R/W):** Cada dispositivo tiene una dirección de 7 bits. El octavo bit (el menos significativo) que se envía durante la operación de direccionamiento, completando el byte, indica el tipo de operación a realizar. Si este bit es alto el dispositivo maestro lee información proveniente de un dispositivo esclavo. Si este bit es bajo, el dispositivo maestro escribe información en un dispositivo esclavo.

La comunicación en más detalle

Cuando el dispositivo maestro quiere comunicarse con un esclavo, produce una secuencia de inicio en el bus. La secuencia de inicio es una de las dos secuencias especiales que se han definido en el bus I2C; la otra es la secuencia de parada. Las secuencias de inicio y la de parada son especiales porque son los dos únicos casos en que se permite que la línea de datos (SDA) cambie cuando la línea de reloj (SCL) está alta. Cuando se están transmitiendo datos, la línea SDA debe permanecer estable, y jamás cambiar, mientras la línea SCL está alta. Las secuencias de inicio y de parada señalan el comienzo y el final de una transacción con los dispositivos esclavos.

Los datos se transfieren en secuencias de 8 bits. Estos bits se colocan en la línea SDA comenzando por el bit de más peso (o más significativo). Una vez puesto un bit en SDA, se lleva la línea SCL a alto. Debemos recordar que el chip no puede llevar la línea a un estado alto, en realidad, lo que hace es

"soltarla", y el que la pone en nivel lógico alto es el resistor de polarización. Por cada 8 bits que se transfieren, el dispositivo que recibe el dato envía de regreso un bit de reconocimiento, de modo que en realidad por cada byte de dato se producen 9 pulsos sobre la línea SCL (es decir, 9 pulsos de reloj por cada 8 bits de dato). Si el dispositivo que recibe envía un bit de reconocimiento bajo, indica que ha recibido el dato y que está listo para aceptar otro byte. Si retorna un alto, lo que indica es que no puede recibir más datos y el dispositivo maestro debería terminar la transferencia enviando una secuencia de parada.

Direccionamiento de dispositivos en el bus I2C

Lo más común en los dispositivos para el bus I2C es que utilicen direcciones de 7 bits, aunque existen dispositivos de 10 bits. Este último caso es raro.

Una dirección de 7 bits implica que se pueden poner hasta 128 dispositivos sobre un bus I2C, ya que un número de 7 bits puede ir desde 0 a 127. Cuando se envían las direcciones de 7 bit, de cualquier modo la transmisión es de 8 bits. El bit extra se utiliza para informarle al dispositivo esclavo si el dispositivo maestro va a escribir o va a leer datos desde él. Si el bit de lectura/escritura (R/W) es cero, el dispositivo maestro está escribiendo en el esclavo. Si el bit es 1 el maestro está leyendo desde el esclavo. La dirección de 7 bit se coloca en los 7 bits más significativos del byte y el bit de lectura/escritura es el bit menos significativo.

El hecho de colocar la dirección de 7 bits en los 7 bits más significativos del byte produce confusiones entre quienes comienzan a trabajar con este bus. Si, por ejemplo, se desea escribir en la dirección 21 (hexadecimal), en

realidad se debe enviar un 42, que es un 21 desplazado un bit hacia arriba. También se pueden tomar las direcciones del bus I2C como direcciones de 8 bit, en las que las pares son de sólo escritura y las impares son de sólo lectura. Para dar un ejemplo, el integrado de brújula magnética CMPS03 es fijado en fábrica en la dirección 0xC0 (\$C0). La dirección 0xC0 se utiliza para escribir en él y la dirección 0xC1 es para leer de él.

Protocolo de programación para el bus I2C

Lo primero que ocurre en un bus I2C es que el dispositivo maestro envía una secuencia de inicio. Esto alerta a los dispositivos esclavos, poniéndolos a la espera de una transacción. Éstos quedan atentos para ver si se trata de una solicitud para ellos. A continuación el dispositivo maestro envía la dirección de dispositivo. El dispositivo esclavo que posee esa dirección continuará con la transacción, y los otros ignorarán el resto de los intercambios, esperando la próxima secuencia de inicio.

Habiendo direccionado ya el dispositivo esclavo, lo que debe hacer ahora el maestro es enviar la ubicación interna o número de registro desde el que desea leer o al que va a escribir. La cantidad depende, obviamente, de qué dispositivo es y de cuántos registros internos posee. Algunos dispositivos muy simples no tienen ninguno, pero la mayoría sí los poseen.

Siguiendo con el ejemplo del CMPS03, éste posee 16 ubicaciones internas, numeradas desde el 0 al 15. Otro dispositivo, el medidor ultrasónico de distancia SRF08, tiene 36 registros.

Una vez que el maestro ha enviado la dirección del dispositivo en el bus I2C y la dirección del registro interno del dispositivo, puede enviar ahora el

byte o bytes de datos. El dispositivo maestro puede seguir enviando bytes al esclavo, que normalmente serán puestos en registros con direcciones sucesivas, ya que el esclavo incrementa automáticamente la dirección del registro interno después de recibir cada byte. Cuando el maestro ha terminado de escribir datos en el esclavo, envía una secuencia de parada que concluye la transacción.

Escritura en un dispositivo esclavo:

1. Enviar una secuencia de inicio.
2. Enviar la dirección de dispositivo con el bit de lectura/escritura en bajo
3. Enviar el número de registro interno en el que se desea escribir.
4. Enviar el byte de dato
5. [Opcionalmente, enviar más bytes de dato]
6. Enviar la secuencia de parada

Como ejemplo, veamos un SRF08, que tiene una dirección de bus fijada en fábrica de 0xE0. Para comenzar una medición de distancia con el SRF08 se debe escribir 0x51 en el registro de comando, ubicado en la dirección interna 0x00. La secuencia es la que sigue:

1. Enviar una secuencia de inicio
2. Enviar 0xE0 (La dirección de dispositivo del SRF08 con el bit de lectura/escritura en bajo)
3. Enviar 0x00 (dirección interna del registro de comando)
4. Enviar 0x51 (el comando para comenzar la medición del SRF08)

5. Enviar la secuencia de parada

Lectura desde un dispositivo esclavo:

Esta operación es algo más complicada, pero no demasiado. Antes de leer datos desde el dispositivo esclavo, primero se le debe informar desde cuál de sus direcciones internas se va a leer. De manera que una lectura desde un dispositivo esclavo en realidad comienza con una operación de escritura en él. Es igual a cuando se desea escribir en él: Se envía la secuencia de inicio, la dirección de dispositivo con el bit de lectura/escritura en bajo y el registro interno desde el que se desea leer. Ahora se envía otra secuencia de inicio nuevamente con la dirección de dispositivo, pero esta vez con el bit de lectura/escritura en alto. Luego se leen todos los bytes necesarios y se termina la transacción con una secuencia de parada.

Volviendo al ejemplo del módulo de brújula CMPS03, veamos cómo se lee el registro de ángulo:

1. Enviar una secuencia de inicio
2. Enviar 0xC0 (La dirección de dispositivo del CMPS03 con el bit de lectura/escritura en bajo)
3. Enviar 0x01 (dirección interna del registro de ángulo en valor 0-255)
4. Enviar una secuencia de inicio (inicio reiterado)
5. Enviar 0xC1 (La dirección de dispositivo del CMPS03 con el bit de lectura/escritura en alto)
6. Leer un byte de dato desde el CMPS03
7. Enviar la secuencia de parada

La secuencia se puede observar en el siguiente gráfico

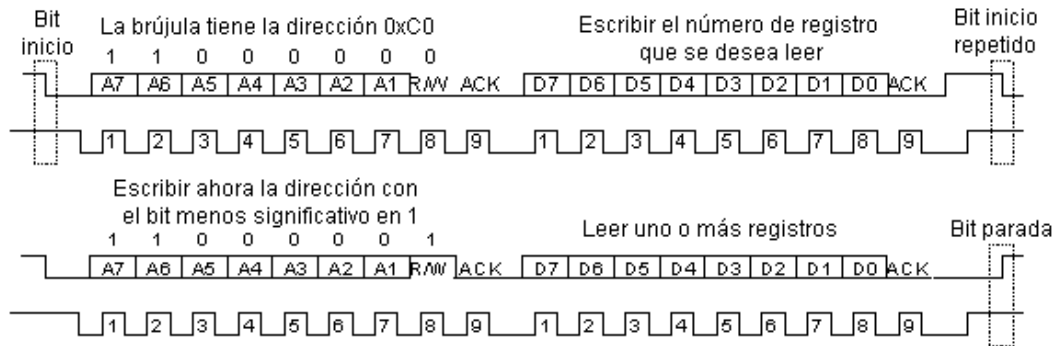


Gráfico 9: Lectura i2c

Filtro de Kalman

El filtro de Kalman es un algoritmo desarrollado por Rudolf E. Kalman en 1960 que sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal, al igual que el observador de Luenberger, pero sirve además cuando el sistema está sometido a ruido blanco aditivo. La diferencia entre ambos es que en el observador de Luenberger, la ganancia K de realimentación del error debe ser elegida "a mano", mientras que el filtro de Kalman es capaz de escogerla de forma óptima cuando se conocen las varianzas de los ruidos que afectan al sistema.

La ecuación general para un sistema es:

- $X_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}$
- $Z_k = C_k x_k + v_k$

Donde:

w_k es ruido blanco de valor promedio igual a cero y con varianza Q_k en

el instante k .

v_k Es ruido blanco de valor promedio igual a cero y con varianza R_k en el instante k .

El filtro de Kalman permite identificar el estado X_k a partir de las mediciones anteriores de $u_k, z_k, , R_k$ y las identificaciones anteriores de X_k .

Arcotangente

En trigonometría, la arcotangente está definida como la función recíproca de la tangente de un ángulo. Si tenemos: $\arctan \alpha$, su significado geométrico es el arco cuya tangente es α .

La función tangente no es biyectiva, por lo que no tiene recíproca. Es posible aplicarle una restricción del dominio de modo que se vuelva inyectiva y sobreyectiva. Por convención es preferible restringir el dominio de la

función tangente al intervalo abierto $-\frac{\pi}{2}, \frac{\pi}{2}$.

Además, el límite corresponde a: $\lim_{n \rightarrow \infty} \arctan(n) = \frac{\pi}{2}$

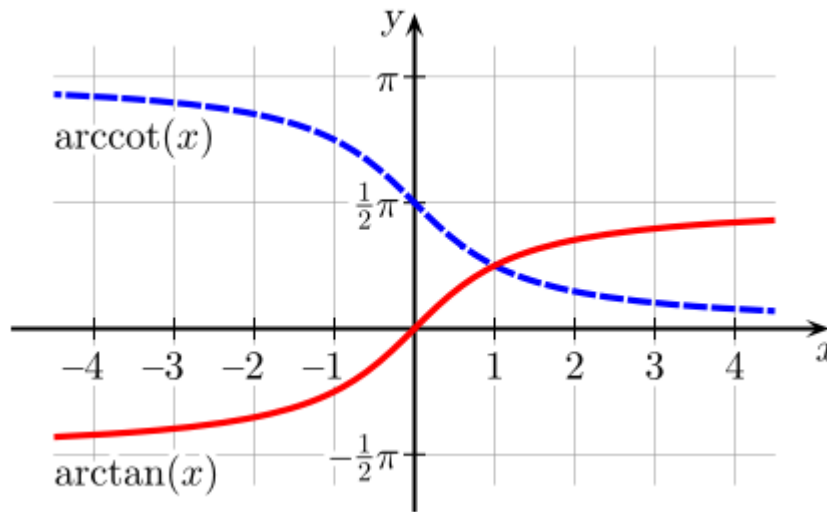


Gráfico 10: Arcotangente
(<http://es.wikipedia.org>)

Control PID

Un PID (Proporcional Integral Derivativo) es un mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor que se quiere obtener, para aplicar una acción correctora que ajuste el proceso. El algoritmo de cálculo del control PID se da en tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor Proporcional determina la reacción del error actual. El Integral genera una corrección proporcional a la integral del error, esto asegura que aplicando un esfuerzo de control suficiente, el error de seguimiento se reduce a cero. El Derivativo determina la reacción del tiempo en el que el error se produce. La suma de estas tres acciones es usada para ajustar al proceso vía un elemento de control como la posición de una válvula de control o la energía suministrada a un calentador, por ejemplo. Ajustando estas tres variables en el algoritmo de control del PID, el controlador puede proveer un control diseñado para lo que requiera el proceso a realizar. La respuesta del controlador puede ser

descrita en términos de respuesta del control ante un error, el grado el cual el controlador llega al "set point", y el grado de oscilación del sistema. Nótese que el uso del PID para control no garantiza control óptimo del sistema o la estabilidad del mismo. Algunas aplicaciones pueden solo requerir de uno o dos modos de los que provee este sistema de control. Un controlador PID puede ser llamado también PI, PD, P o I en la ausencia de las acciones de control respectivas. Los controladores PI son particularmente comunes, ya que la acción derivativa es muy sensible al ruido, y la ausencia del proceso integral puede evitar que se alcance al valor deseado debido a la acción de control.

(<http://es.wikipedia.org>)

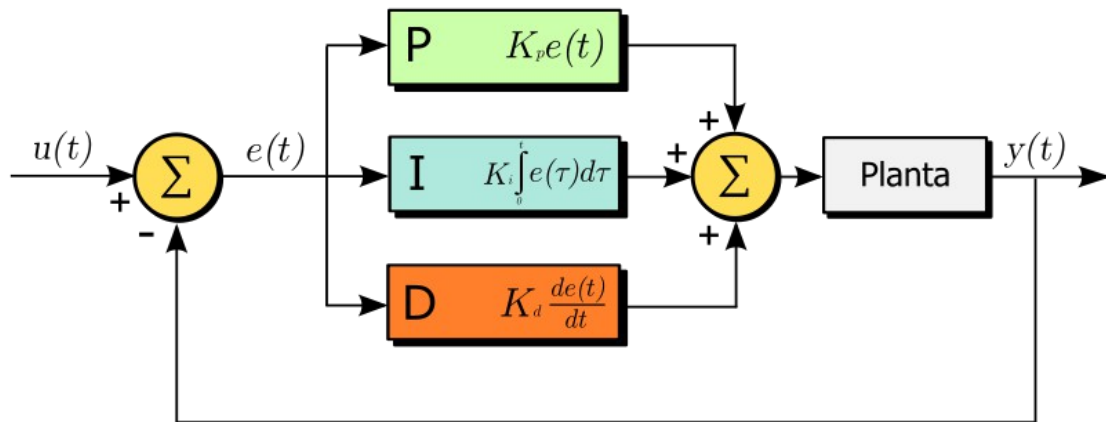


Gráfico 11: Diagrama en bloques de un control PID

Funcionamiento

Para el correcto funcionamiento de un controlador PID que regule un proceso o sistema se necesita, al menos:

- Un sensor, que determine el estado del sistema.

- Un controlador, que genere la señal que gobierna al actuador.
- Un actuador, que modifique al sistema de manera controlada (resistencia eléctrica, motor, válvula, bomba, etc.).

El sensor proporciona una señal analógica o digital al controlador, la cual representa el punto actual en el que se encuentra el proceso o sistema. La señal puede representar ese valor en tensión eléctrica, intensidad de corriente eléctrica o frecuencia. En este último caso la señal es de corriente alterna, a diferencia de los dos anteriores, que son con corriente continua.

El controlador lee una señal externa que representa el valor que se desea alcanzar. Esta señal recibe el nombre de punto de consigna (o punto de referencia), la cual es de la misma naturaleza y tiene el mismo rango de valores que la señal que proporciona el sensor. Para hacer posible esta compatibilidad y que, a su vez, la señal pueda ser entendida por un humano, habrá que establecer algún tipo de interfaz(HMI-Human Machine Interface), son pantallas de gran valor visual y fácil manejo que se usan para hacer más intuitivo el control de un proceso.

El controlador resta la señal de punto actual a la señal de punto de consigna, obteniendo así la señal de error, que determina en cada instante la diferencia que hay entre el valor deseado (consigna) y el valor medido. La señal de error es utilizada por cada uno de los 3 componentes del controlador PID. Las 3 señales sumadas, componen la señal de salida que el controlador va a utilizar para gobernar al actuador. La señal resultante de la suma de estas tres se llama variable manipulada y no se aplica directamente sobre el actuador, sino que debe ser transformada para ser compatible con el actuador utilizado.

Las tres componentes de un controlador PID son: parte Proporcional, acción Integral y acción Derivativa. El peso de la influencia que cada una de estas partes tiene en la suma final, viene dado por la constante proporcional, el tiempo integral y el tiempo derivativo, respectivamente. Se pretenderá lograr que el bucle de control corrija eficazmente y en el mínimo tiempo posible los efectos de las perturbaciones.

Proporcional

La parte proporcional consiste en el producto entre la señal de error y la constante proporcional como para que hagan que el error en estado estacionario sea casi nulo, pero en la mayoría de los casos, estos valores solo serán óptimos en una determinada porción del rango total de control, siendo distintos los valores óptimos para cada porción del rango. Sin embargo, existe también un valor límite en la constante proporcional a partir del cual, en algunos casos, el sistema alcanza valores superiores a los deseados. Este fenómeno se llama sobre oscilación y, por razones de seguridad, no debe sobrepasar el 30%, aunque es conveniente que la parte proporcional ni siquiera produzca sobre oscilación. Hay una relación lineal continua entre el valor de la variable controlada y la posición del elemento final de control (la válvula se mueve al mismo valor por unidad de desviación). La parte proporcional no considera el tiempo, por lo tanto, la mejor manera de solucionar el error permanente y hacer que el sistema contenga alguna componente que tenga en cuenta la variación respecto al tiempo, es incluyendo y configurando las acciones integral y derivativa.

La fórmula del proporcional está dada por: $P_{sal} = K_p e(t)$

El error, la banda proporcional y la posición inicial del elemento final de

control se expresan en tanto por uno. Indicará la posición que pasará a ocupar el elemento final de control.

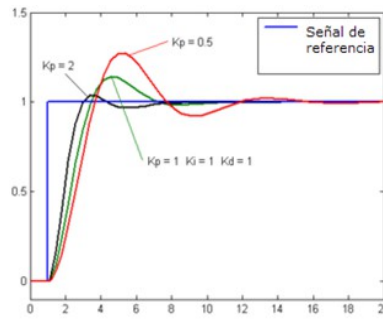


Gráfico 12: Proporcional

(<http://es.wikipedia.org>)

Integral

El modo de control Integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por el modo proporcional. El control integral actúa cuando hay una desviación entre la variable y el punto de consigna, integrando esta desviación en el tiempo y sumándola a la acción proporcional. El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un período determinado; Luego es multiplicado por una constante I. Posteriormente, la respuesta integral es adicionada al modo Proporcional para formar el control P + I con el propósito de obtener una respuesta estable del sistema sin error estacionario.

El modo integral presenta un desfase en la respuesta de 90° que sumados a los 180° de la retroalimentación (negativa) acercan al proceso a tener un retraso de 270° , luego entonces solo será necesario que el tiempo muerto contribuya con 90° de retardo para provocar la oscilación del proceso. (La ganancia total del lazo de control debe ser menor a 1, y así inducir una atenuación en la salida del controlador para conducir el proceso a estabilidad

del mismo.) Se caracteriza por el tiempo de acción integral en minutos por repetición. Es el tiempo en que delante una señal en escalón, el elemento final de control repite el mismo movimiento correspondiente a la acción proporcional.

El control integral se utiliza para obviar el inconveniente del offset (desviación permanente de la variable con respecto al punto de consigna) de la banda proporcional.

La fórmula del integral está dada por:

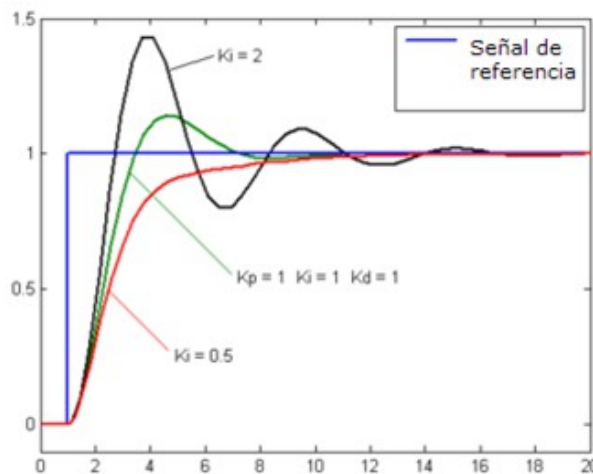
$$I_{sal} = K_i \int_0^t e(t) dt$$


Gráfico 13: Integral

(<http://es.wikipedia.org>)

Derivativo

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error; (si el error es constante, solamente actúan los modos proporcional e integral).

El error es la desviación existente entre el punto de medida y el valor consigna, o "Set Point".

La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente.

Se deriva con respecto al tiempo y se multiplica por una constante D y luego se suma a las señales anteriores (P+I). Es importante adaptar la respuesta de control a los cambios en el sistema ya que una mayor derivativa corresponde a un cambio más rápido y el controlador puede responder acordemente.

La fórmula del derivativo está dada por:

$$D_{sal} = K_d \frac{de}{dt}$$

El control derivativo se caracteriza por el tiempo de acción derivada en minutos de anticipo. La acción derivada es adecuada cuando hay retraso entre el movimiento de la válvula de control y su repercusión a la variable controlada.

Cuando el tiempo de acción derivada es grande, hay inestabilidad en el proceso. Cuando el tiempo de acción derivada es pequeño la variable oscila demasiado con relación al punto de consigna. Suele ser poco utilizada debido a la sensibilidad al ruido que manifiesta y a las complicaciones que ello conlleva.

La acción derivada puede ayudar a disminuir el rebosamiento de la variable durante el arranque del proceso. Puede emplearse en sistemas con

tiempo de retardo considerables, porque permite una repercusión rápida de la variable después de presentarse una perturbación en el proceso.

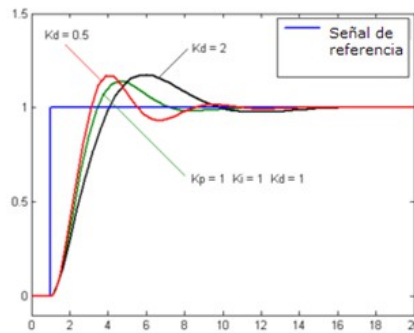


Gráfico 14: Derivativo

(<http://es.wikipedia.org>)

Motores Eléctricos

Es una máquina que convierte la energía eléctrica en mecánica, principalmente mediante el movimiento rotatorio. En la actualidad existen nuevas aplicaciones con motores eléctricos que no producen movimiento rotatorio, sino que con algunas modificaciones, ejercen tracción sobre un riel. Estos motores se conocen como motores lineales.

Esta máquina de corriente continua es una de las más versátiles en la industria. Su fácil control de posición, paro y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos. Pero con la llegada de la electrónica su uso ha disminuido en gran medida, pues los motores de corriente alterna, del tipo asíncrono, pueden ser controlados de igual forma a precios más accesibles para el consumidor medio de la industria. A pesar de esto los motores de corriente continua se siguen utilizando en muchas aplicaciones de potencia (trenes y tranvías) o de precisión (máquinas, micro motor, entre otros.).



Gráfico 15: Motores DC

La principal característica del motor de corriente continua es la posibilidad de regular la velocidad desde vacío a plena carga.

Su principal inconveniente, el mantenimiento, muy caro y laborioso.

Una máquina de corriente continua (generador o motor) se compone principalmente de dos partes, un estator que da soporte mecánico al aparato y tiene un hueco en el centro generalmente de forma cilíndrica. En el estator además se encuentran los polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro. El rotor es generalmente de forma cilíndrica, también devanado y con núcleo, al que llega la corriente mediante dos escobillas.

También se construyen motores de CC con el rotor de imanes permanentes para aplicaciones especiales.

Principio de funcionamiento

Según la Ley de Lorentz, cuando un conductor por el que pasa una corriente eléctrica se sumerge en un campo magnético, el conductor sufre una fuerza perpendicular al plano formado por el campo magnético y la corriente, siguiendo la regla de la mano derecha, con módulo.

$$F=B*i*l$$

- F: Fuerza en newtons
- i: Intensidad que recorre el conductor (Amperios)
- l: Longitud del conductor en metros lineales
- B: Densidad del Campo Magnético.

El rotor no solo tiene un conductor, sino varios repartidos por la periferia. A medida que gira, la corriente se activa en el conductor apropiado.

Normalmente se aplica una corriente con sentido contrario en el extremo opuesto del rotor, para compensar la fuerza neta y aumentar el momento.

La Fuerza contra electromotriz es la tensión que se crea en los conductores de un motor como consecuencia del corte de las líneas de fuerza, es el efecto generador de pines.

La polaridad de la tensión en los generadores es inversa a la aplicada en bornes del motor.

Las fuertes puntas de corriente de un motor en el arranque son debidas a que con la máquina parada no hay fuerza contra electromotriz y el bobinado se comporta como una resistencia pura del circuito.

Las escobillas deben poner en cortocircuito todas las bobinas situadas en la zona neutra. Si la máquina tiene dos polos, tenemos también dos zonas neutras. En consecuencia, el número total de escobillas ha de ser igual al número de polos de la máquina. En cuanto a su posición, será coincidente con las líneas neutras de los polos.

El sentido de giro de un motor de corriente continua depende del sentido relativo de las corrientes circulantes por los devanados inductor e inducido. La inversión del sentido de giro del motor de corriente continua se consigue invirtiendo el sentido del campo magnético o de la corriente del inducido. Si se permuta la polaridad en ambos bobinados, el eje del motor gira en el mismo sentido.

Los cambios de polaridad de los bobinados, tanto en el inductor como en el inducido se realizarán en la caja de bornes de la máquina, y además el ciclo combinado producido por el rotor produce la fmm (fuerza magneto motriz).

El sentido de giro se puede determinar con la regla de la mano derecha, la cual va a mostrar el sentido de la fuerza. La regla de la mano derecha es de la siguiente manera: el pulgar muestra hacia dónde va la corriente, el dedo índice apunta en la dirección en la cual se dirige el flujo del campo magnético, y el dedo medio hacia dónde va dirigida la fuerza resultante y por lo tanto el sentido de giro.

Los motores y los generadores de corriente continua están constituidos esencialmente por los mismos elementos, diferenciándose únicamente en la forma de utilización.

Por reversibilidad entre el motor y el generador se entiende que si se hace girar al rotor, se produce en el devanado inducido una fuerza electromotriz capaz de transformarse en energía en el circuito de carga.

En cambio, si se aplica una tensión continua al devanado inducido del generador a través del colector de delgas, el comportamiento de la máquina ahora es de motor, capaz de transformar la fuerza contra electromotriz en energía mecánica.

En ambos casos el inducido está sometido a la acción del campo inductor principal.

Puente H

En muchas aplicaciones en las que se utilizan motores DC es necesario que estos giren en ambos sentidos, para ello se emplean los puentes-H.

Un puente H muy popular es el L298, he aquí su descripción tomada textualmente de la hoja de datos:

“Es un driver de puente completo dual que acepta niveles lógicos TTL estándar y maneja cargas inductivas como relés, solenoides motores DC y de paso. Tiene dos entradas de activación (enable) para activar o desactivar los dispositivos independientemente de las señales de entrada. Los emisores de los transistores bajos de cada puente están conectados juntos y el correspondiente terminal externo puede ser usado para la conexión de una resistencia externa de medición. Se ha provisto de una entrada adicional para la fuente así que la lógica funciona a menor voltaje.”

Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader) que corre en la placa.

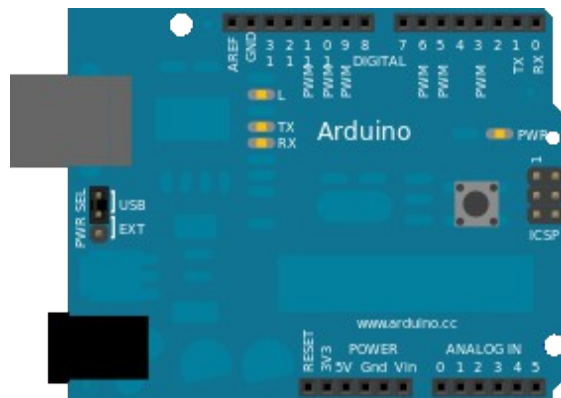


Gráfico 16: Arduino

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Al ser open-hardware, tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Consta de 14 entradas digitales configurables entrada i/o salidas que operan a 5 voltios. Cada pin puede proporcionar o recibir como máximo 40 mA. Los pines 3, 5, 6, 8, 10 y 11 pueden proporcionar una salida PWM (Pulse Width Modulación). Si se conecta cualquier cosa a los pines 0 y 1, eso interferirá con la comunicación USB. Diecimila también tiene 6 entradas analógicas que proporcionan una resolución de 10 bits. Por defecto miden de 0 voltios (masa) hasta 5 voltios, aunque es posible cambiar el nivel más alto, utilizando el pin Aref y algún código de bajo nivel.

Péndulo Invertido

Dentro del estudio de los robots de piernas o extremidades, lo más importante ha sido el balanceo de estos, pues es la parte esencial para que no caiga el cuerpo del robot.

Las primeras máquinas que se balanceaban activamente fueron automáticamente controladas por péndulos invertidos. Es sabido por cada uno de nosotros que un humano puede balancear una escoba en su mano o en un dedo con relativa facilidad. ¿Por qué no utilizar control automático para construir una escoba (péndulo invertido) que se balancee por si misma? Claude Shannon fue probablemente el primero en hacerlo. En 1951 usó las partes de un juego erector para construir una máquina que balanceara un péndulo invertido que estaba arriba de un pequeño carrito de poder. El manejador del carro iba en direcciones atrás y adelante, en respuesta a los tipos de movimiento del péndulo, éste era censado por un par de

interruptores en su base. Para lograr moverlo de un lugar a otro, el carro primero tenía que manejarse fuera del destino para poder des balancear al péndulo, para proceder en dirección del destino. Para balancear nuevamente hacia el destino, el carro se movía pasando el destino hasta que el péndulo se encontrará otra vez completamente perpendicular a la horizontal sin velocidad hacia delante, entonces se movía en sentido inverso del destino y así lo lograba balancear.

La importancia del balance activo para la locomoción no es un problema nuevo, pero el progreso en la construcción de sistemas con extremidades físicas que ocuparan estos principios fue retardado por la evidente dificultad del trabajo. No fue sino a finales de los años setenta que los experimentos en balance comenzaron a salir adelante.

Modelado Matemático del Péndulo Invertido

Para la implementación del vehículo en cuestión se requiere estimar ciertas variables físicas del sistema de forma de poder definir parámetros constructivos y de diseño, como posibles materiales, energía necesaria, tipos de sensores, entre otros.

Como se aprecia en el modelo simplificado de las mostradas, el vehículo es básicamente un péndulo invertido, con la diferencia de que el agente que induce el movimiento (el motor) tiene una acción directa tanto en la ecuación para el movimiento lineal como para el movimiento angular. En el caso del péndulo invertido clásico, la acción sobre el movimiento angular es sólo consecuencia de la dinámica de los cuerpos.

Las variables y parámetros que se considerarán se detallan en el

siguiente cuadro

Cuadro 1: Variables y Parámetros del Modelado Matemático

Nombre	Tipo	Significado
θ	Variable	Inclinación del vehículo respecto de la gravedad
$\dot{\theta}$	Variable	Velocidad de giro respecto del eje
x	Variable	Posición del centro de las ruedas respecto de la horizontal
\dot{x}	Variable	Velocidad del centro de las ruedas respecto de la horizontal
X_b	Variable	Posición del centro de masas del sistema respecto de la horizontal
\dot{X}_b	Variable	Velocidad del centro de masas del sistema respecto de la horizontal
Y_b	Variable	Posición del centro de masas del sistema respecto de la vertical
\dot{Y}_b	Variable	Velocidad del centro de masas del sistema respecto de la vertical
δ	Variable	Velocidad angular de giro sobre el plano
L	Parámetro	Distancia del eje de las ruedas al CM
m_b	Parámetro	Masa de la barra (estructura y pasajero)
$I_{b\theta}$	Parámetro	Momento de inercia de la barra respecto del CM en inclinaciones
$I_{b\delta}$	Parámetro	Momento de inercia de la barra respecto del CM en giros
D	Parámetro	Distancia entre centro del vehículo y centro de una rueda
R	Parámetro	Radio de la rueda
m_r	Parámetro	Masa de la rueda
I_r	Parámetro	Momento de inercia de la rueda respecto del eje de giro
τ_{izq}, τ_{der}	Acción	Acción Torques aplicados a la rueda izquierda y derecha

$$I_{b\theta}*\theta - L*m_b*x*\cos(\theta) + m_b*L^2*\theta = -\tau_{izq} - \tau_{der}$$

$$\left(m_b + 2m_r + 2\frac{I_r}{R^2}\right)*x - m_b*L*\theta^2*\sin(\theta) + m_b*L*\theta*\cos(\theta) = \frac{\tau_{izq} + \tau_{der}}{R}$$

$$I_{b\delta}*\delta = \frac{\tau_{izq} + \tau_{der}}{R} * D$$

De la observación de las ecuaciones, es fácil notar que existe una dependencia entre el movimiento lineal (representado por x) y la inclinación (representada por θ). Únicamente el giro del vehículo no tiene relación con el resto de las variables del sistema, siendo dependiente solo de la acción de los motores.

Al linealizar el sistema en torno a $\theta=0^\circ$, las ecuaciones quedan de la siguiente forma.

$$\theta = a*\theta + b*(\tau_{izq} + \tau_{der})$$

$$x = c*\theta + d*(\tau_{izq} + \tau_{der})$$

$$\delta = e*(\tau_{izq} - \tau_{der})$$

Donde a , b , c , d y e son constantes que dependen de los parámetros del vehículo. Visto de modo matricial, el sistema es el siguiente:

$$\frac{d}{dt} \begin{pmatrix} x \\ x \\ \theta \\ \theta \\ \delta \\ \delta \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} x \\ x \\ \theta \\ \theta \\ \delta \\ \delta \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ d & d \\ 0 & 0 \\ b & b \\ 0 & 0 \\ e & e \end{pmatrix} * \begin{pmatrix} \tau_{izq} \\ \tau_{der} \end{pmatrix}$$

Como se puede apreciar de las ecuaciones, la variable asociada a la inclinación (θ) dejó de depender de las otras variables del sistema (x, δ). Es decir, es posible diseñar un controlador simple para el ángulo θ y separadamente para el giro δ . No obstante, la dinámica del movimiento lineal sí dependerá de la inclinación. De este modo, se puede considerar el problema (en torno al punto de operación) como dos problemas menores e independientes, los que controlan la inclinación y el giro, mientras que el movimiento lineal queda dependiente de la dinámica de dichos sistemas.

Debido a la característica Múltiples entradas y múltiples salidas que presenta el sistema y el modo de funcionamiento del mismo, se requiere la medición o estimación en cada instante de las siguientes variables:

Inclinación / Velocidad de inclinación ($\theta, \dot{\theta}$). Este parámetro es la principal variable a ser controlada, ya que de ella depende directamente el equilibrio del sistema.

Velocidad lineal (\dot{x}). Es la velocidad de traslación del vehículo, combinación de la velocidad de ambas ruedas.

Referencia de giro (δ). Se requiere para establecer la dirección y velocidad de rotación deseada del vehículo. Básicamente representará una diferencia de velocidad entre cada una de las ruedas.

DESARROLLO DEL TRABAJO DE GRADO

Para la mejor comprensión y para facilitar el diseño, el sistema se dividió en módulos que, a continuación, se muestran en un diagrama de bloque que representa los módulos en los que estarán conformados el sistema a desarrollar.

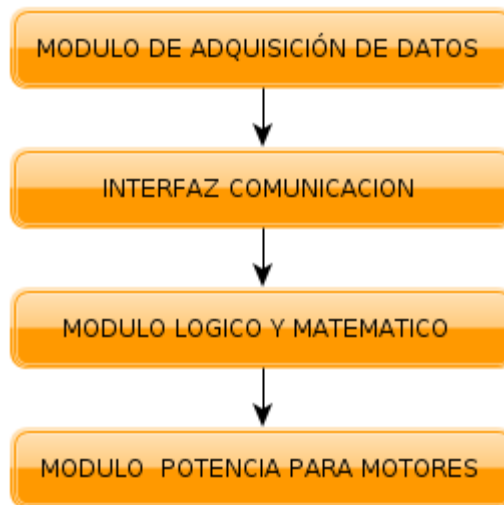
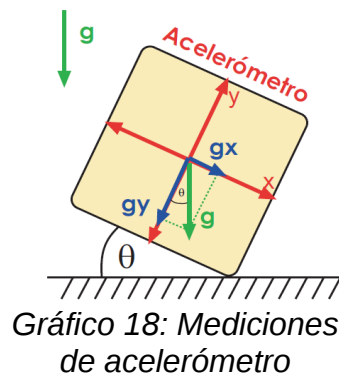


Gráfico 17: Diagrama de Módulos

Modulo de Adquisición de Datos

Esta medición tiene una gran complejidad, ya que requiere el uso de 2 tipos de sensores: acelerómetros y giróscopos. Un acelerómetro, como dice su nombre, es un sensor capaz de detectar y cuantificar aceleraciones externas, lo cual se hace generalmente referenciando el valor respecto de la gravedad. Así, un acelerómetro en reposo en la superficie terrestre indica una aceleración de $-1g$, mientras que aislado de cualquier fuente gravitacional, y manteniendo el estado de reposo, indicaría una medición

nula. Este tipo de sensores se utiliza para una estimación directa de la inclinación tomando como hipótesis que la única aceleración existente es la gravedad, la cual se busca descomponer en los ejes de medición según el ángulo de inclinación, como se esquematiza a continuación.



Usando el eje x del acelerómetro, cuya medición fue de g_x , se puede deducir la inclinación q de la igualdad $g * \sin(q) = g_x$, o bien usando el eje y, donde la igualdad sería $g * \cos(q) = g_y$. Es importante notar que dependiendo del cuadrante en que se encuentre el ángulo se debe usar una u otra ecuación, ya que solo una permitirá establecer el signo de q (en el ejemplo visto, la igualdad para el eje x permite determinar dicho signo). Cuando un acelerómetro se utiliza de este modo suele llamarse inclinómetro.

Esta estimación será tan exacta como lo sea el sensor respectivo siempre que la única aceleración existente sea la gravedad. De lo contrario, de existir aceleraciones anexas éstas aparecerán en las mediciones del sensor (ya que se trata de un sistema no inercial), dando así una solución distinta a la esperada de las igualdades ya vistas. El sistema en cuestión está expuesto a aceleraciones distintas a la gravedad, por su condición de vehículo en movimiento.

Otro modo de estimar la inclinación es mediante un giróscopo electrónico. Este sensor entrega una señal directa de la velocidad de inclinación ($\dot{\theta}$), es decir, de la derivada de la variable que se busca estimar. Tiene la gran ventaja respecto del acelerómetro que las aceleraciones externas no le afectan, por lo que suele usarse para estimar la inclinación mediante el uso de una condición inicial conocida e integración durante un período. Es decir, la inclinación se estima según $\theta_t = \theta_{t-1} + \dot{\theta} dt$, donde θ_{t-1} es la estimación anterior (o condición inicial) y dt corresponde al tiempo de muestreo utilizado. El problema más claro de esta solución es el error acumulado asociado a la integración numérica, que provoca que luego de transcurrido cierto tiempo de cómputo la estimación pierda credibilidad. En este tipo de aplicaciones este error es conocido como *drift*, ya que el valor asociado.

Luego, ambos sensores presentan ventajas y desventajas. Sin embargo, estas son complementarias. Es decir, donde tiene problemas un tipo sensor, el otro no los tiene. Así, pueden usarse ambos sensores al unísono, donde el acelerómetro tiene una mejor estimación de largo plazo (las aceleraciones son de corta duración y no tiene error acumulado), mientras el giróscopo posee un mejor comportamiento en el corto plazo (no es afectado por las aceleraciones transientes). Así, una posible estimación para la inclinación que use ambos sensores está dada por la ecuación.

$$\theta_t = (\theta_{t-1} + \dot{\theta} * dt) * \beta + \theta_{acc} * (1 - \beta)$$

Donde $\beta \in [0;1]$ determina el porcentaje de creencia que se le asigna a cada estimación (ya sea con giróscopo o con acelerómetro), y su valor se debe determinar según el comportamiento observado, θ_{acc} y $\dot{\theta}$ representan las mediciones del acelerómetro y el giróscopo, respectivamente. Este tipo

de estimación mediante acelerómetros y giróscopos es la base de una IMU, principal componente en los sistemas que buscan estimar su orientación e inclinación. Este tipo de unidades son utilizadas en una amplia variedad de aplicaciones, como lo son aviones, helicópteros, satélites, barcos e incluso misiles, entre otras posibles aplicaciones. La complejidad de la unidad IMU está dada principalmente por la cantidad de giróscopos y acelerómetros que esta posea, los cuales dependiendo de su orientación pueden determinar orientación en distintos ejes. Así, la placa de sensores que se debe diseñar tendrá (entre otros tipos de sensores) una unidad IMU básica, consistente en un acelerómetro y un giróscopo, permitiendo determinar orientación respecto de un solo eje (en este caso, respecto del eje de los motores).

Sensor

El IMU 3000 cuenta con un giróscopo de digital de 3 ejes con intervalos programables de 250 a 2000 grados por segundo. Además, tiene un puerto secundario I2C que interactúa con un acelerómetro digital (ADXL345) para proporcionar un completo procesamiento de movimientos en 6-ejes a la salida del sensor por su puerto I2C primario.

Por la naturaleza física de los componentes se adquirió una placa que integra ambos sensores, con los pines necesarios en el IMU 3000 y se conecta un acelerómetro ADXL345 a la entrada I2C de la IMU 3000. También posee una regulación de voltaje a 3.3v para mayor seguridad de los dispositivos.

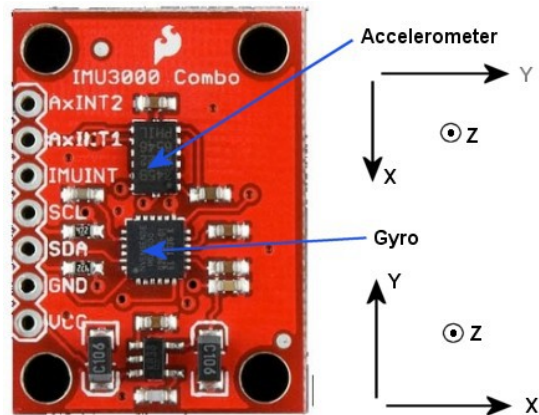


Gráfico 19: IMU Fusion Board - ADXL345 & IMU3000

Arcotangente

El cálculo de una arcotangente de 4 cuadrantes dentro de un Microcontrolador es constantemente objeto de muchas discusiones. Varias técnicas, tales como tabla de búsqueda y expansión polinomio son bien conocidos.

Para algunos métodos de normalización del número complejo debe ser necesario, la aplicación efectiva de un limitador fijo o función de amplitud invariante.

Naturalmente se calcula la arcotangente como:

$$\theta = \text{atan}\left(\frac{a}{b}\right)$$

Podemos ver quien es a y b en la el gráfico siguiente.

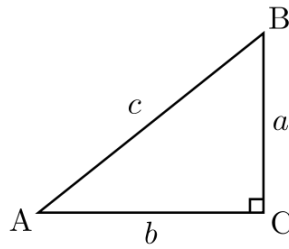


Gráfico 20: Triangulo Rectángulo

Para calcular una relación de auto-normalización depende del cuadrante donde resida el número complejo. Para un número complejo z , sea $x = \text{Re}(z)$, $y = \text{Im}(z)$.

$$r = \frac{x-y}{x+y} \quad (1)$$

Para obtener el ángulo de fase, calcular:

$$\theta_1 = \frac{\pi}{4} - \frac{\pi}{4 * r} \quad (2)$$

En este orden, si el numero complejo se encuentra en el II Cuadrante

$$r = \frac{x+y}{x-y} \quad (3)$$

Para obtener el Alguno en el Segundo cuadrante:

$$\theta_2 = \frac{3 * \pi}{4} - \frac{\pi}{4 * r} \quad (4)$$

El error máximo usando las ecuaciones (2) o (4) es un poco menos de 0,07 rads (sólo en algunos ángulos). La precisión del estimador es en realidad bastante buena teniendo en cuenta que se está utilizando un polinomio de primer orden para estimar el ángulo de fase.

Si utiliza un polinomio de grado superior, resulta que las potencias pares de la polinomio va a desaparecer (debido a la función impar), relajando así parte de la carga de computo.

Para obtener una mayor precisión (un error máximo de .01 rads), se puede sustituir las ecuación (2) con (4).

Interfaz de comunicación

Debido a la incompatibilidad de los niveles de tensión entre el Arduino y el IMU3000, se requirió una etapa de conversión de niveles de tensión, de 3.3voltios a 5voltios que sea bidireccional, esto también por el medio de comunicación i2c.

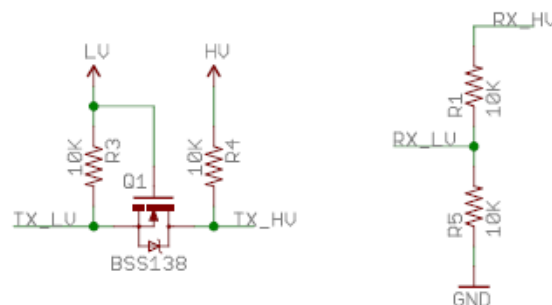


Gráfico 21: Modulo de conversión de Voltajes

Permitiendo así esta conexión para la comunicación exitosa y segura entre imu3000 y Arduino

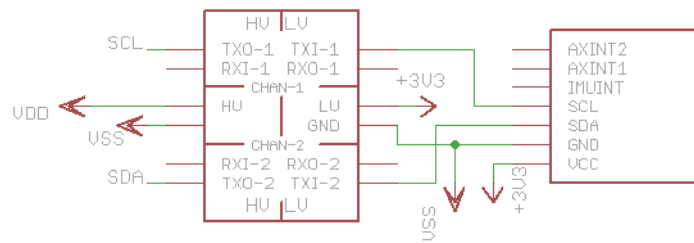


Gráfico 22: Conexión IMU3000 y Conversión lógico

Modulo de Control

El Arduino Duemilanove es una placa con microcontrolador basada en el ATmega168. Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset.

Cuadro 2: Características Arduino Duemilanove

Característica	Valor
Microcontrolador	ATmega168
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines E/S digitales	14 (6 proporcionan salida PWM)
Pines de entrada analógica	6
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	16 KB de las cuales 2 KB las usa el bootloader
SRAM	1 KB

Característica	Valor
EEPROM	512 bytes
Velocidad de reloj	16 MHz

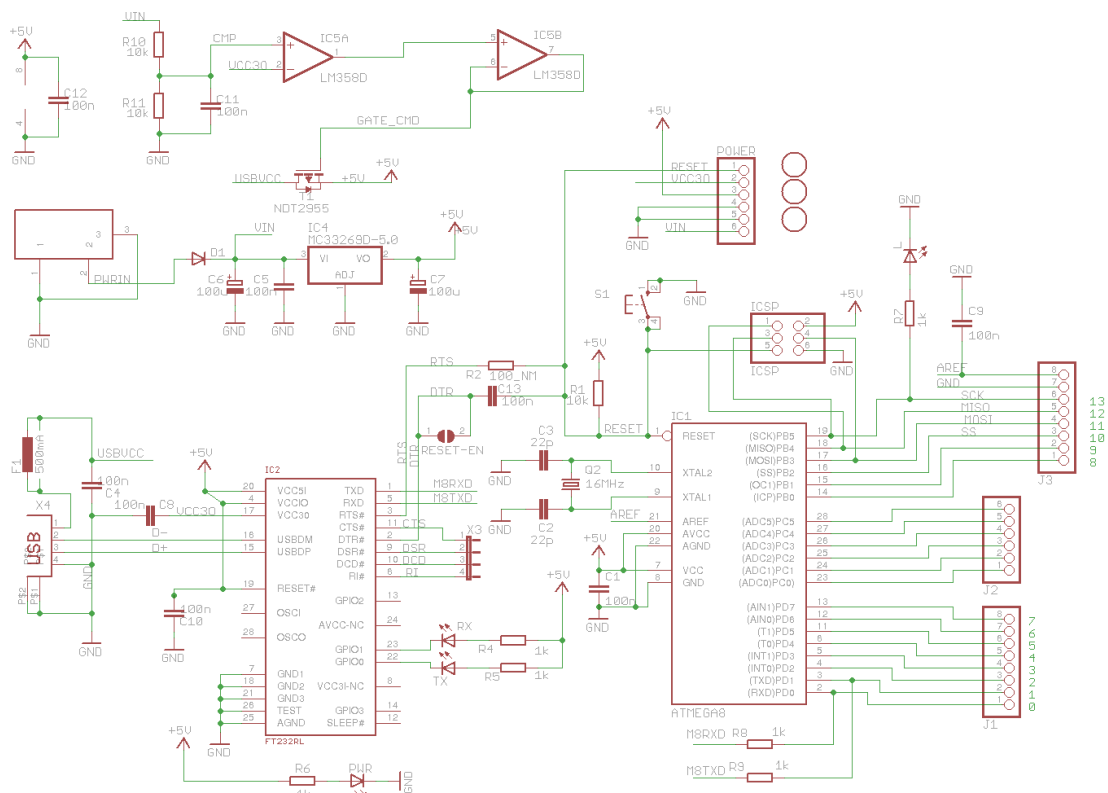


Gráfico 23: Arduino Duemilanove

Alimentación

El Arduino Duemilanove puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador o una batería. El transformador se puede conectar usando un conector macho de 2.1mm con centro positivo en el conector hembra de la placa. Los cables de la batería puede conectarse a los pines Gnd y Vin en los conectores de alimentación (POWER)

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable, si se usan mas de 12V los reguladores de voltaje se pueden sobre calentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- **VIN.** La entrada de voltaje a la placa Arduino cando se está usando una fuente externa de alimentación (en opuesto a los 5 voltios de la conexión USB). Se puede proporcionar voltaje a través de este pin, o, si se está alimentado a través de la conexión de 2.1mm , acceder a ella a través de este pin.
- **5V.** La fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.
- **3V3.** Una fuente de voltaje a 3.3 voltios generada en el chip FTDI integrado en la placa. La corriente máxima soportada 50mA.
- **GND.** Pines de toma de tierra.

Memoria

El ATmega328 tiene 32KB (el ATmega168 tiene 16 KB) de memoria flash para almacenar código (2KB son usados para el arranque del sistema(bootloader).El ATmega328 tiene 2 KB (Atmega168 1 KB) de memoria SRAM . El ATmega328 tiene 1KB (ATmega168 512 bytes) de EEPROM , que puede a la cual se puede acceder para leer o escribir con la Librería EEPROM.

Entradas y Salidas

Cada uno de los 14 pines digitales en el Duemilanove pueden utilizarse como entradas o como salidas usando las funciones pinMode(), digitalWrite(), y digitalRead() . Las E/S operan a 5 voltios. Cada pin puede proporcionar o recibir una intensidad máxima de 40mA y tiene una resistencia interna (desconectada por defecto)de 20-50kOhms. Además, algunos pines tienen funciones especializadas:

- **Serie: 0 (RX) y 1 (TX).** Usado para recibir (RX) transmitir (TX) datos a través de puerto serie TTL. Estos pins están conectados a los pines correspondientes del chip FTDI USB-to-TTL.
- **Interrupciones Externas:** 2 y 3. Estos pines se pueden configurar para lanzar una interrupción en un valor LOW(0V), en flancos de subida o bajada (cambio de LOW a HIGH(5V) o viceversa), o en cambios de valor. Ver la función attachInterrupt() para as detalles.
- **PWM:** 3, 5, 6, 9, 10, y 11. Proporciona una salida PWM (Pulse Wave Modulation, modulación de onda por pulsos) de 8 bits de resolución (valores de 0 a 255) a través de la función analogWrite().
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines

proporcionan comunicación SPI, que a pesar de que el hardware la proporcione actualmente no está incluido en el lenguaje Arduino.

- **LED:** 13. Hay un LED integrado en la placa conectado al pin digital 13, cuando este pin tiene un valor HIGH(5V) el LED se enciende y cuando este tiene un valor LOW(0V) este se apaga.

El Duemilanove tiene 6 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores). Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función `analogReference()`. Además algunos pines tienen funciones especializadas:

- **I2C:** 4 (SDA) y 5 (SCL). Soporte del protocolo de comunicaciones I2C (TWI) usando la librería Wire.

Hay unos otros pines en la placa:

- **AREF.** Voltaje de referencia para las entradas analógicas. Usado por `analogReference()`.
- **Reset.** Suministrar un valor LOW(0V) para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los shields que no dejan acceso a este botón en la placa.

Comunicaciones

EL Arduino Duemilanove facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores. El Atmega168 proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y 1(TX). Un chip FTDI FT232RL integrado en la

placa canaliza esta comunicación serie a través del USB y los drivers FTDI (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadearán cuando se detecte comunicación transmitida a través del chip FTDI y la conexión USB (no parpadearán si se usa la comunicación serie a través de los pines 0 y 1).

La librería SoftwareSerial permite comunicación serie por cualquier par de pines digitales del Duemilanove.

El ATmega168 soporta la comunicación I2C (TWI) y SPI. El software de Arduino incluye una librería Wire para simplificar el uso del bus I2C.

El ATmega168 en las placas Arduino Duemilanove viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original.

También es posible saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ISCP (In Circuit Serial Programming).

Reinicio Automático (Software)

En vez de necesitar reiniciar presionando físicamente el botón de reset antes de cargar, el Arduino Duemilanove está diseñado de manera que es posible reiniciar por software desde el ordenador donde esté conectado. Una de las líneas de control de flujo (DTR) del FT232RL está conectada a la línea de reinicio del ATmega328 o ATmega168 a través de un condensador de 100 nano faradios. Cuando la línea se pone a LOW(0V), la línea de reinicio

también se pone a LOW el tiempo suficiente para reiniciar el chip. El software de Arduino utiliza esta característica para permitir cargar los sketches con solo apretar un botón del entorno. Dado que el gestor de arranque tiene un lapso de tiempo para ello, la activación del DTR y la carga del sketch se coordinan perfectamente.

Esta configuración tiene otras implicaciones. Cuando el Duemilanove se conecta a un ordenador con Mac OS X o Linux, esto reinicia la placa cada vez que se realiza una conexión desde el software (vía USB). El medio segundo aproximadamente posterior, el gestor de arranque se está ejecutando. A pesar de estar programado para ignorar datos mal formateados (ej. cualquier cosa que la carga de un programa nuevo) intercepta los primeros bytes que se envían a la placa justo después de que se abra la conexión. Si un sketch ejecutándose en la placa recibe algún tipo de configuración inicial u otro tipo de información al inicio del programa, se debe asegurar que el software con el cual se comunica espera un segundo después de abrir la conexión antes de enviar los datos.

El Duemilanove contiene una pista que puede ser cortada para deshabilitar el auto-reset. Las terminaciones a cada lado pueden ser soldadas entre ellas para rehabilitarlo. Están etiquetadas con "RESET-EN".

Protección contra sobretensiones en USB

El Arduino Duemilanove tiene un multifusible reinicializable que protege la conexión USB de tu ordenador de cortocircuitos y sobretensiones. A parte que la mayoría de ordenadores proporcionan su propia protección interna, el fusible proporciona una capa extra de protección. Si mas de 500mA son detectados en el puerto USB, el fusible automáticamente corta la conexión

hasta que el cortocircuito o la sobretensión desaparece.

Características Físicas

La longitud y amplitud máxima de la placa Duemilanove es de 2.7 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. Ten en cuenta que la distancia entre los pines digitales 7 y 8 es 160 mil (0,16"), no es múltiple de la separación de 100 mil entre los otros pines.

Modulo de Potencia para Motores

Este modulo consta en 2 elementos, el primero son los motores DC que darán potencia al equipo, y el segundo el desarrollo de la electrónica de potencia capaz de controlarlos. Por facilidad en la implementación de este modulo, se opta por utilizar motores de corriente continua. Se opta un modelo de motor que presenta un fácil montaje y un torque considerable para soportar la estructura. El motor seleccionado es el modelo EasyRoller Spur Gear Head Motor, cuyas principales características se muestran en la siguiente tabla.

Parámetro	Valor
Voltaje	12V
RPM	200
Torque	3.6Kg
Reducción	30:1

La elección del motor permite determinar parámetros generales de diseño en términos estructurales y de electrónica de potencia. El control de un motor de este tipo se realizará mediante un circuito tipo puente H integrado L298 que soporta los voltajes y corrientes necesarios.

El control del voltaje entregado al motor, será mediante señales PWM que excitan el semiconductor, de modo de regular la potencia traspasada al motor según el ciclo de trabajo asignado a dichas señales.

L298 – Puente H Dual

El circuito le permitirá de manera fácil e independiente el control de dos motores DC de hasta 2A cada uno en ambas direcciones.

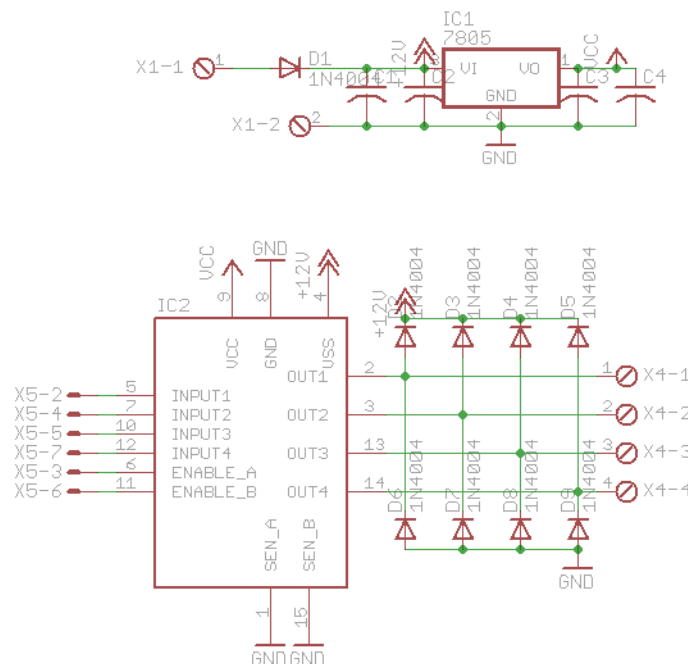


Gráfico 24: L298 y electrónica necesaria

Permitiendo también interconectar directamente con el Arduino, mediante sus pines de entrada. El circuito incorpora 4 LEDs de dirección (2 por motor), un disipador de calor, terminales de tornillo, así como ocho diodos Schottky de protección de los CEM, y dos resistencias de alta potencia, que permiten el seguimiento de la corriente consumida en cada motor a través de los canales analógicos del Arduino.

(Solarrobotics.com)

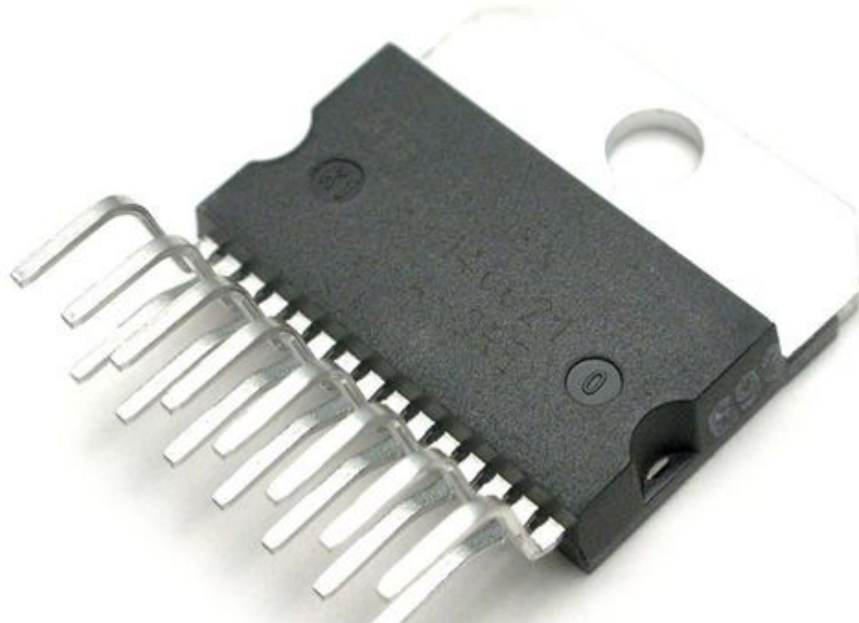


Gráfico 25: l298

Como el l298 es dual bien es posible manejar 2 motores distintos sin sobrepasar la corriente máxima de 2 amperios para cada puente o bien podemos conectar ambos puentes-H en paralelo y obtener un solo puente-H en teoría con el doble de capacidad de corriente.

La resistencia sensora es una que está en serie a la bobina de armadura

del motor con el fin de medir la corriente que pasa a través de este. Esto puede ser muy útil: se puede implementar un control por software software que proteja al motor en caso que se supere la corriente máxima nominal estipulada por el fabricante del motor. Recordemos que en un motor DC el torque es proporcional a la corriente de armadura, esto puede ser muy interesante desde el punto de vista de control.

El valor de la resistencia sensora depende de su motor, de las corrientes que maneja, y lo que se necesite. Recuerda que en realidad medirá tensión sobre la resistencia sensora y no la corriente directamente, por lo tanto el ruido y las tensiones máximas de trabajo son un factor a tener en cuenta.

DIAGRAMA DE BLOQUES Y/O DIAGRAMA DE PROCESOS

Se presenta un único diagrama de proceso ya que el sistema así lo requiere.

Se puede ver todo como un lazo abierto de control, que está en constante ejecución comenzando por la lectura de la señales del IMU3000 y el ADXL345, pasando estas por un proceso de promediar cada lectura a razón de evitar lecturas no deseadas, seguido se puede obtener la razón de aceleración angular, obteniendo el arcotangente entre las señales del eje Z y X del acelerómetro.

Esta señal será procesada por un filtro de Kalman, el cual ayuda a evitar vibraciones y ruido en la señal.

Seguido se debe analizar las señales finalmente por un control PID, dentro del Microcontrolador, como resultado de esto se obtendrá el valor que

será asignado a la señal de PWM que se asignara a los motores DC.

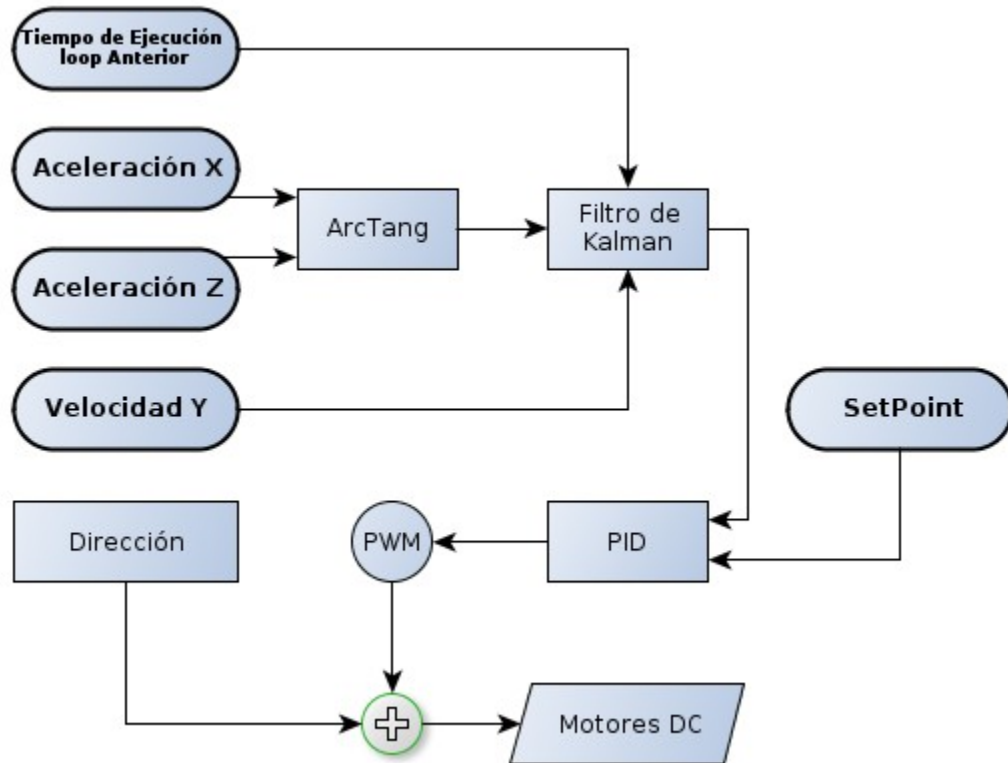


Gráfico 26: Diagrama de Procesos

SOFTWARE DEL SISTEMA

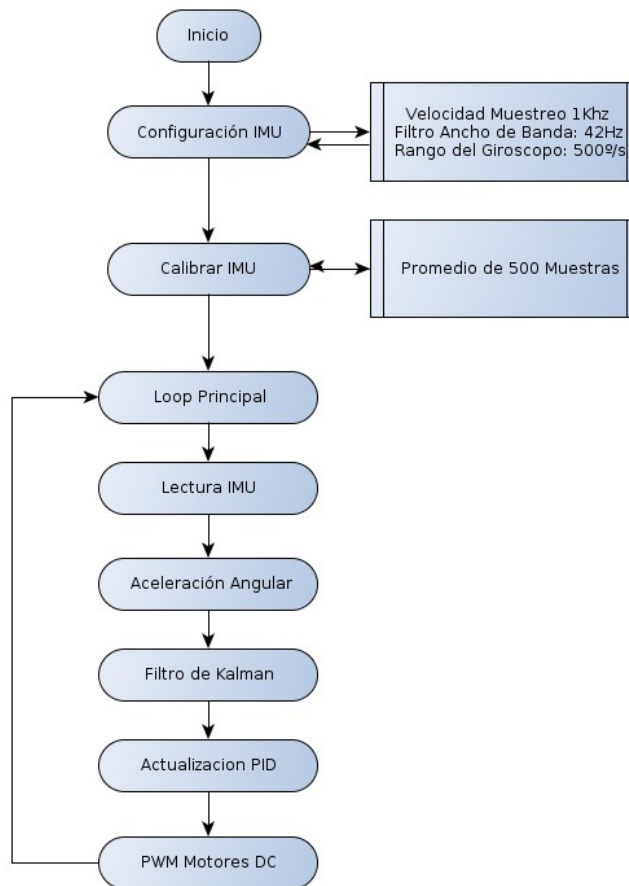


Gráfico 27: Diagrama de Flujo Bajo Nivel

Interfaz de Análisis Interactivo

Se desarrollo una interfaz de análisis interactivo para estudiar todas las variables involucradas en los procesos de compensación de ángulo de caída vertical, esta interfaz fue desarrollada en el lenguaje Processing.

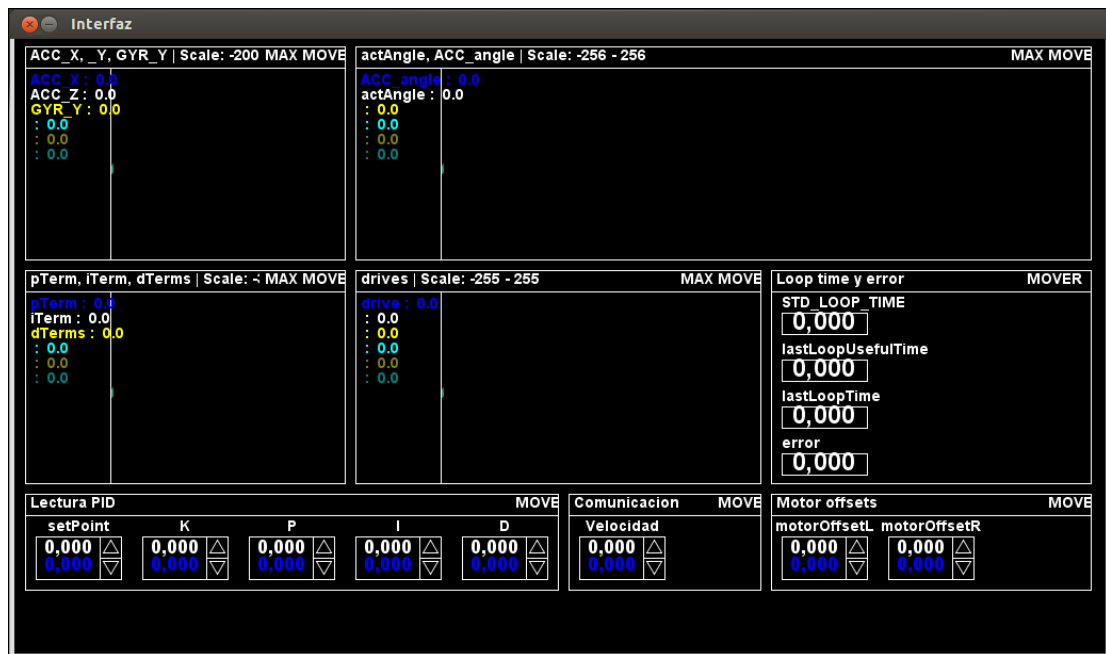


Gráfico 28: Interfaz Interactiva

Esta interfaz permite leer y manipular todas las variables del sistema para así obtener una respuesta deseada, principalmente está diseñada manipular la respuesta de algoritmo de control PID. Esta pantalla muestra en todo momento todas las variables que intervienen en el desarrollo del prototipo.

Graficado USB

Para un análisis puro y en tiempo real todas las variables de los procesos de lectura y procesamiento de señales, fue diseñado un software en Python que se encarga de leer estas variables desde el Puerto USB y graficar directamente, sin ningún análisis.

Se uso el modulo de interfaces gráficas Wxpython para generar las ventanas.

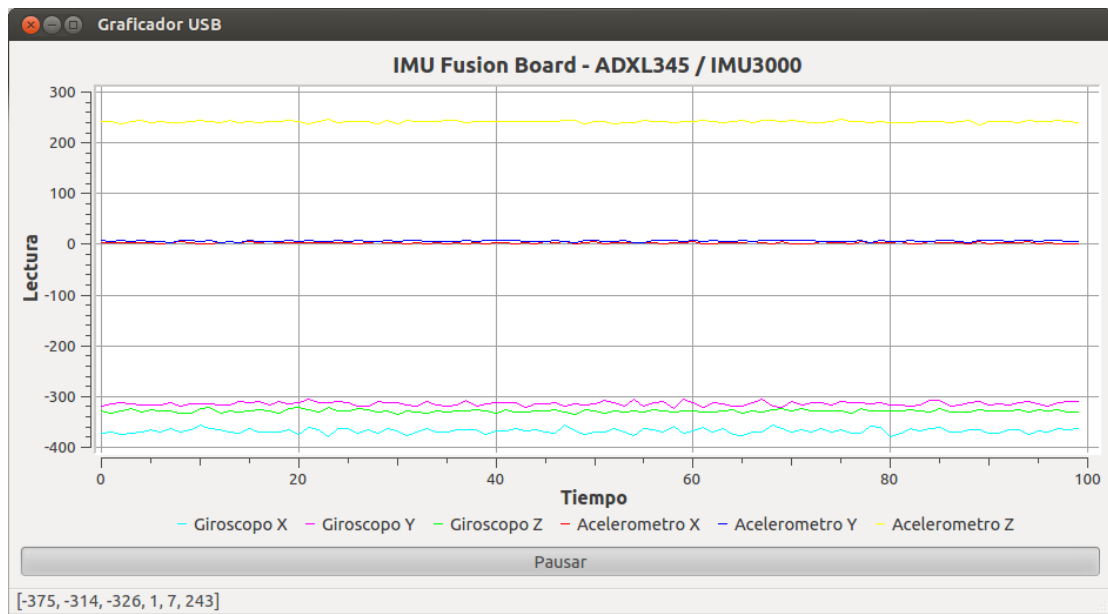


Gráfico 29: Graficador USB

PRUEBAS Y RESULTADOS

Lecturas IMU3000&ADXL345

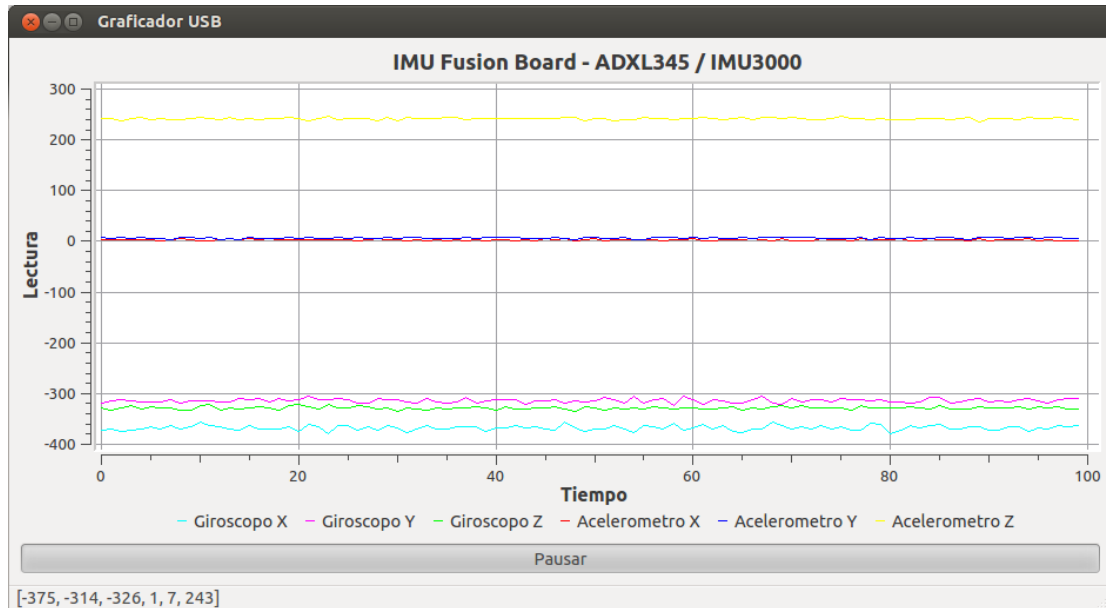


Gráfico 30: IMU3000&ADXL345 0 Grados

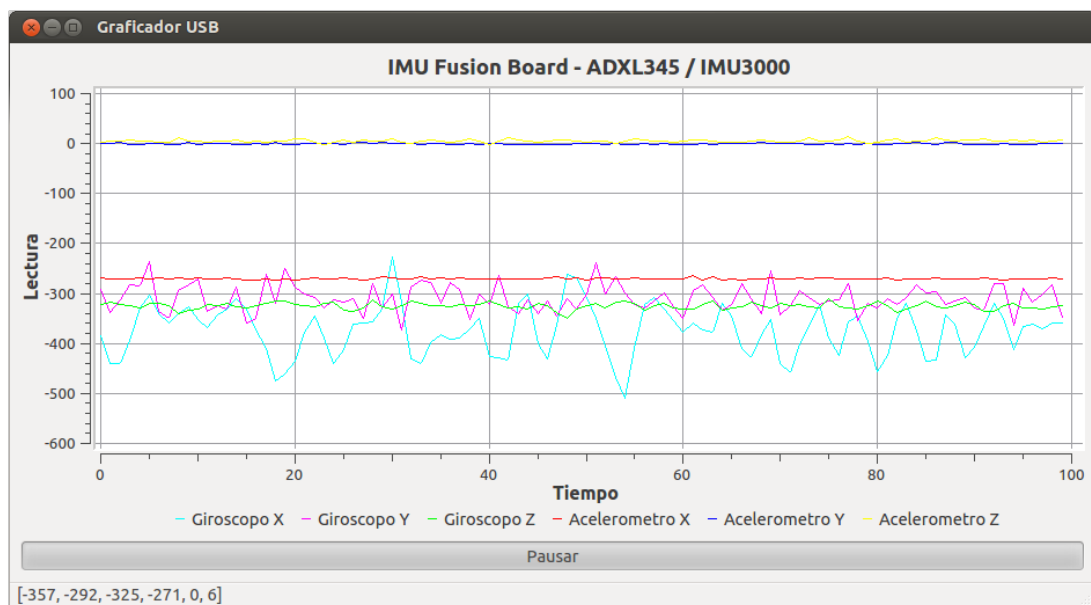
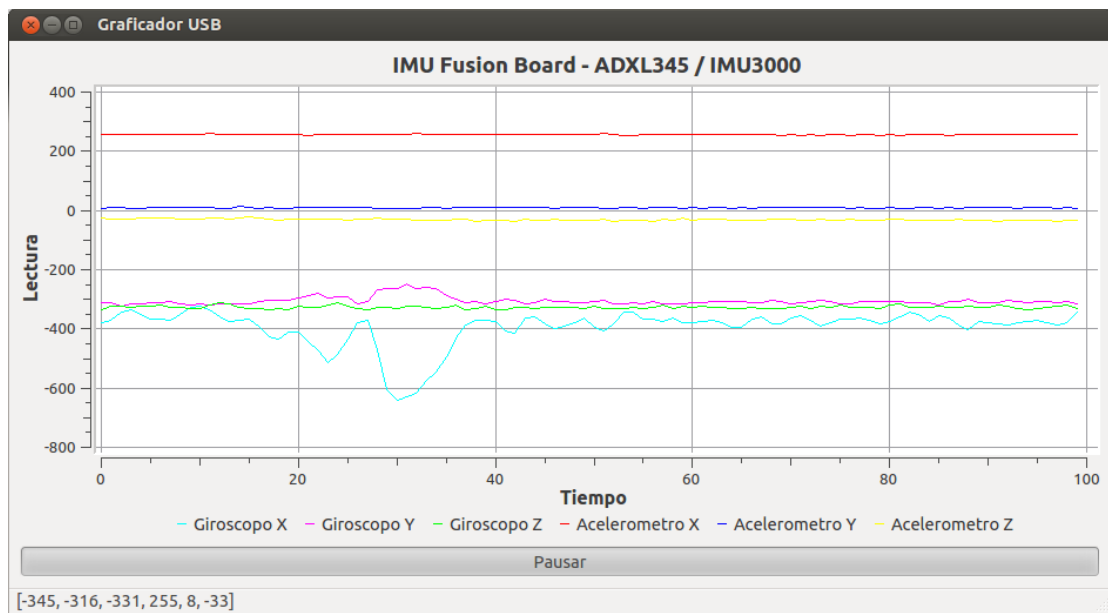
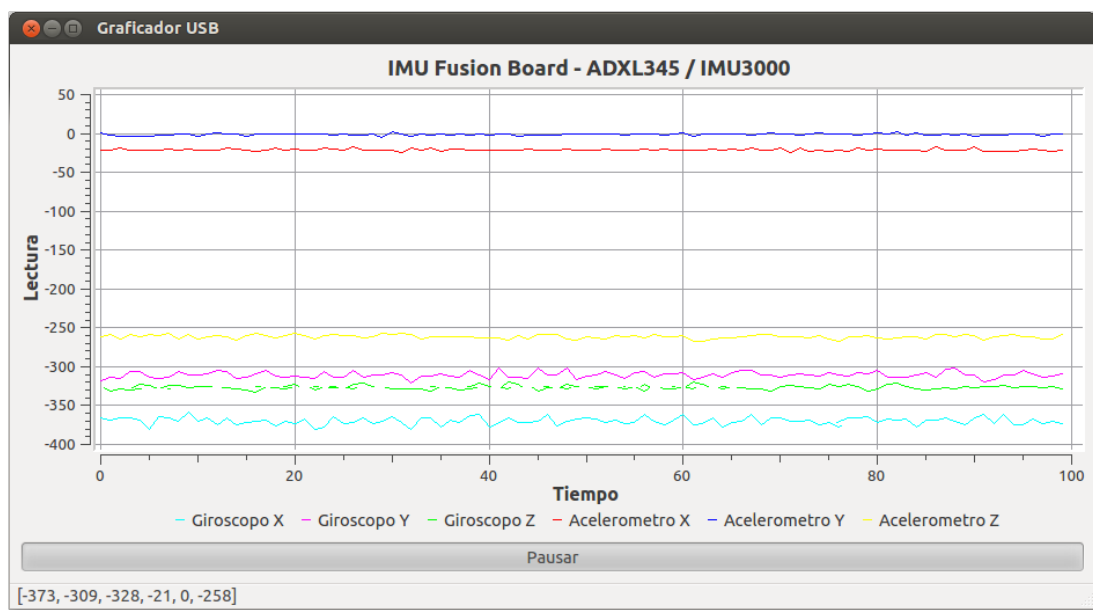


Gráfico 31: IMU3000&ADXL345 90 Grados



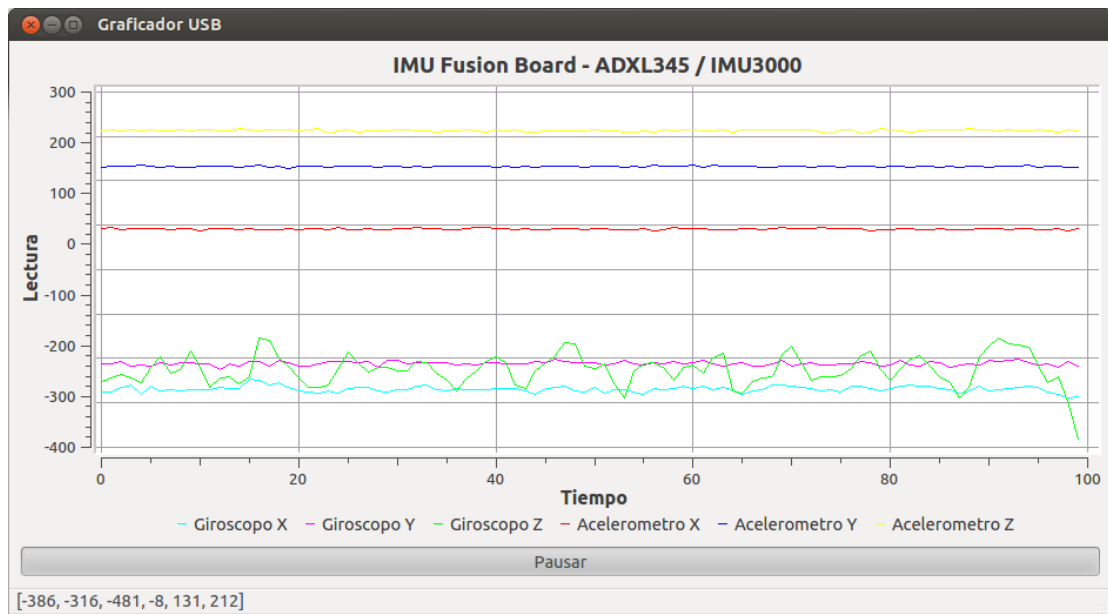


Gráfico 34: IMU3000&ADXL345 -30 Grados Eje Y

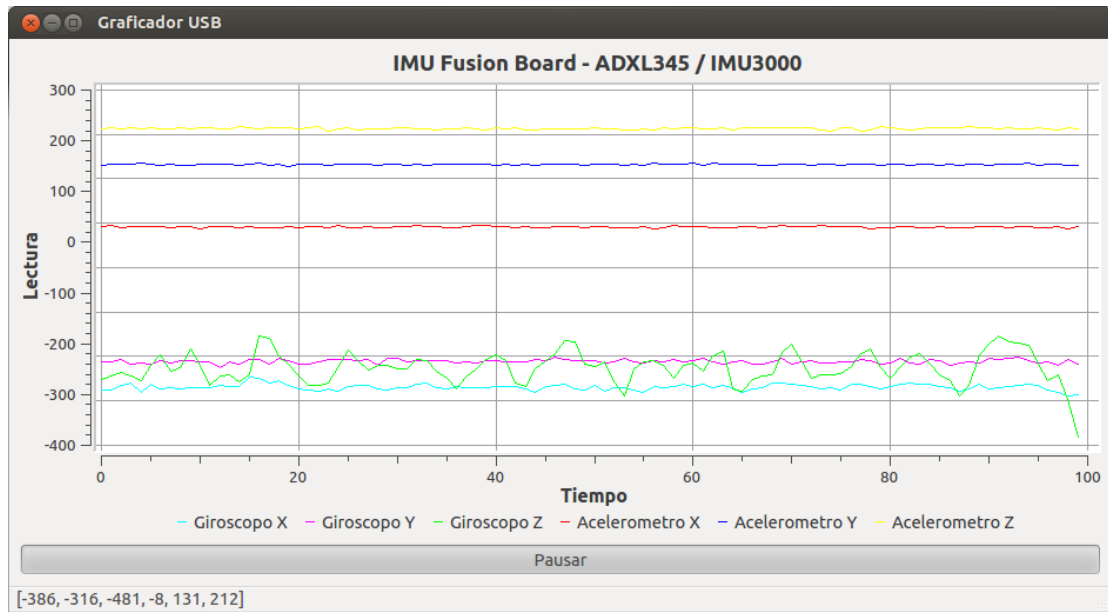


Gráfico 35: IMU3000&ADXL345 +30 Grados Eje Y

Aceleración Angular:

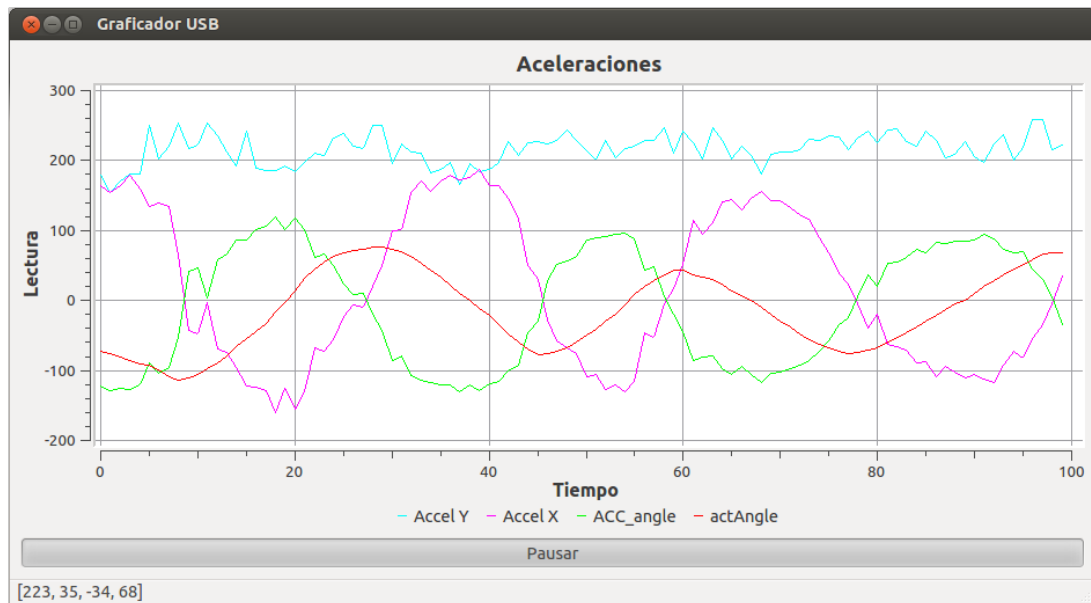


Gráfico 36: Aceleraciones

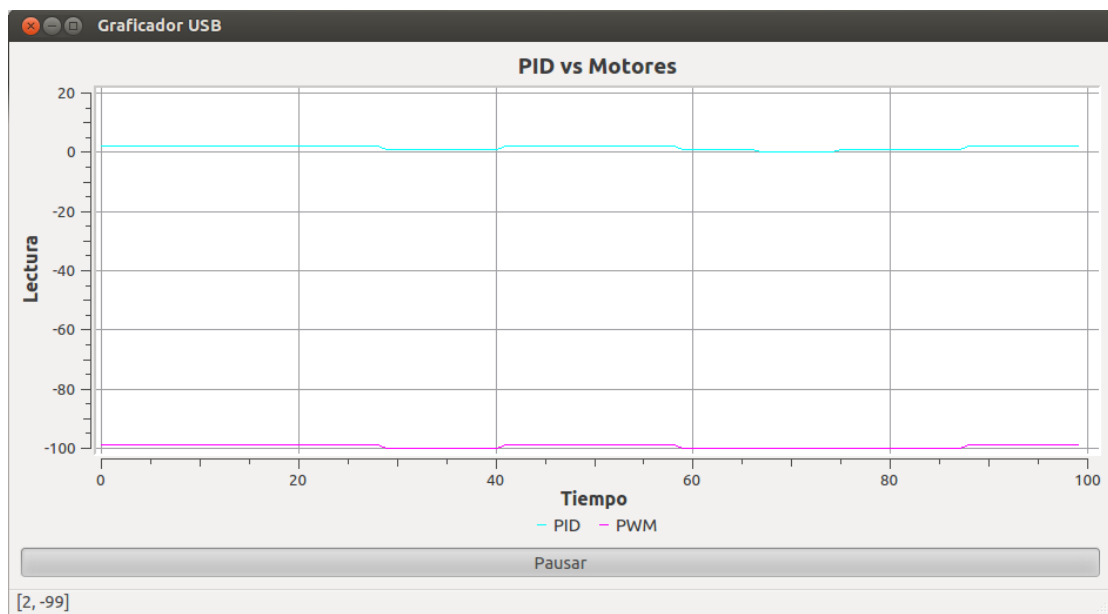


Gráfico 37: PWM en relación al PID

Señal PWM Motores DC

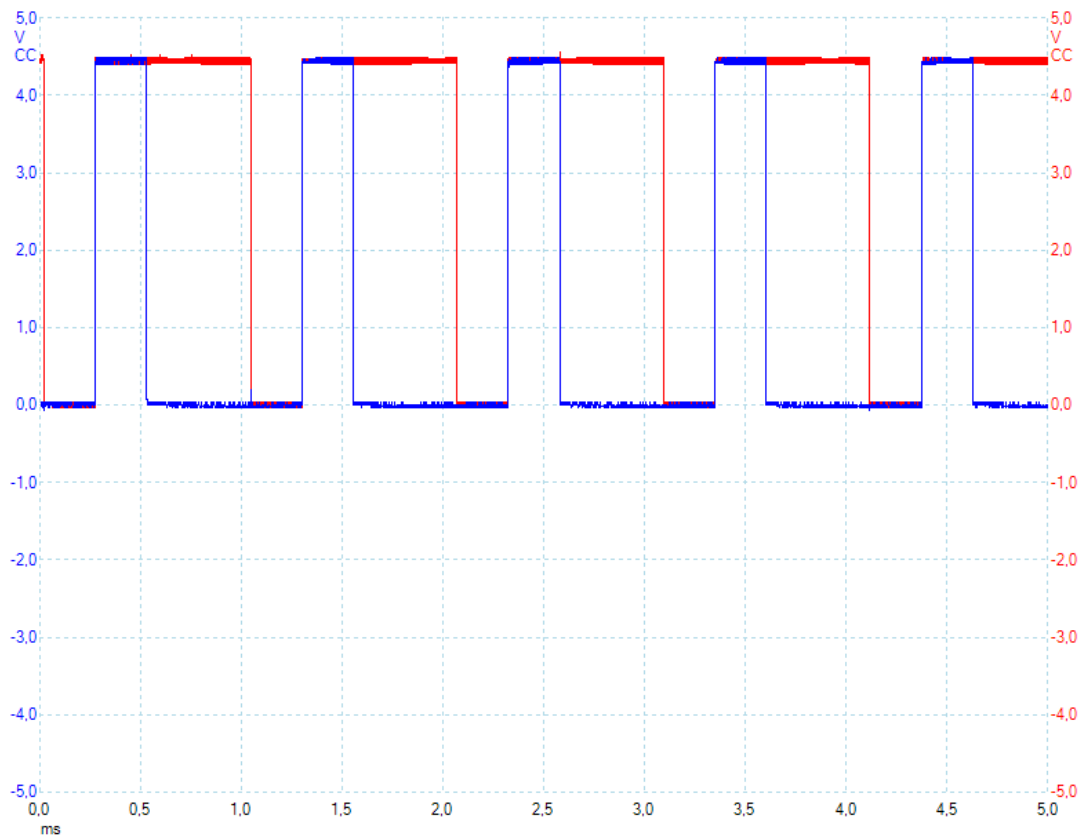


Gráfico 38: PWM para ambos motores, inversamente proporcionales

Prueba de Motores DC

Cuadro 3: Prueba Motores DC

Prueba	Resultado
Valor mínimo Asignado a PWM para generar movimiento:	<ul style="list-style-type: none"> • 96 Motor Derecho, • 110 Motor Izquierdo
Tiempo Mínimo de Respuesta para 45-50ms movimiento:	

Prueba	Resultado
Consumo de Amperaje sin Carga:	120mA
Consumo con Carga:	1.2A

CONCLUSIONES

Considerando los objetivos específicos planteados se diseñó un dispositivo móvil basado en el modelo péndulo invertido capaz de mantener el equilibrio ante pequeñas perturbaciones de oscilación física. La necesidad de desarrollar un completo sistema de control dentro de un microcontrolador de 8 bits se logro demostrar que si es posible, con resultados exitosos, dando paso a una mayor autonomía de los equipos que antes dependían de un computador para realizar este tipo de cálculos. En el trabajo de investigación se realizaron varios avances en el modelado matemático dentro de un microcontrolador en este caso Arduino, se pudo determinar la importancia de un estudio de la velocidad de respuesta de los motores dc para aplicaciones específicas como estas, y se sitúa dentro de un nuevo ecosistema de desarrollo electrónico “Stand Alone” logrando independencia de agentes y módulos externos para sistemas de control de alta velocidad.

A lo largo del desarrollo del proyecto se demostró que es capaz realizar la lectura, filtrado digital y análisis de las señales que provee un giróscopo y un acelerómetro para la corrección del equilibrio y posición de un dispositivo móvil ante ligeras perturbaciones de oscilación.

RECOMENDACIONES

La velocidad de respuesta y métodos de construcción del embobinado de un motor dc son valores realmente importantes a la hora de diseñar dispositivos de este tipo, recomienda usar motores y drivers especializados que garanticen una velocidad de respuesta menor a 20ms, así como motores dc que tengan incluidos dentro de su armadura un sistema de encoders, para así poder leer en tiempo real y precisa la distancia recorrida por cada motor, logrando así un mayor control del movimiento del equipo, esta variable puede ser de gran ayuda dentro del control pid para la retroalimentación

Como recomendaciones de sensores, es posible utilizar sensores IMU de señal analógica y esto podría mejorar la velocidad de cómputo y análisis de las señales.

En relación a los componentes eléctricos, se recomienda cambiar baterías LiPO por Baterías de NiCad debido a su resistencia bajos niveles de voltaje.

REFERENCIAS BIBLIOGRAFICAS

Control de motores de CC, Puente H
http://robots-argentina.com.ar/MotorCC_PuenteH.htm [Consultado 2012 Abril 10]

Analog Devices Inc. (2010) ADXL345 Digital Accelerometer
Disponible <http://www.analog.com>

Atmel Corporation (2011) Atmel 168
Disponible <http://www.atmel.com>

STMicroelectronics (2000) L298 - DUAL FULL-BRIDGE DRIVER
<http://www.st.com>

InvenSense (2010) IMU-3000 Motion Processing Unit
<http://www.invensense.com>

Solutions Cubed (2011) Easy Roller Module
<http://www.solutions-cubed.com>

Kalman filtering of IMU data. Disponible
<http://tom.pycke.be/mav/71/kalman-filtering-of-imu-data> [Consulta 2012 Marzo 5]

Maybeck, Peter S. Stochastic models, estimation, and control. Vol. 141 Año 1979.

H. W. Sorenson. IEEE Spectrum Least-squares estimation: from Gauss to Kalman Vol 7. 1970

Arcotangente (2012) Disponible <http://es.wikipedia.org/wiki/Arcotangente> [Consulta 2012 Mayo 20].

K. Ogata, Modern Control Engineering Edición 4, Capitulo 10.

Proporcional integral derivativo Disponible
http://es.wikipedia.org/wiki/Proporcional_integral_derivativo [Consulta 2012 Abril 1]

ANEXOS

Atmel 168

Features

- High performance, low power Atmel® AVR® 8-bit microcontroller
- Advanced RISC architecture
 - 131 powerful instructions – most single clock cycle execution
 - 32 × 8 general purpose working registers
 - Fully static operation
 - Up to 20 MIPS throughput at 20MHz
 - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
 - 4/8/16 Kbytes of in-system self-programmable flash program memory
 - 256/512/512 bytes EEPROM
 - 512/1K/1Kbytes internal SRAM
 - Write/erase cycles: 10,000 flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C¹⁾
 - Optional boot code section with independent lock bits
 - In-system programming by on-chip boot program
 - True read-while-write operation
 - Programming lock for software security
- QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- Peripheral features
 - Two 8-bit timer/counters with separate prescaler and compare mode
 - One 16-bit timer/counter with separate prescaler, compare mode, and capture mode
 - Real time counter with separate oscillator
 - Six PWM channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - 6-channel 10-bit ADC in PDIP Package
 - Programmable serial USART
 - Master/slave SPI serial interface
 - Byte-oriented 2-wire serial interface (Philips I²C compatible)
 - Programmable watchdog timer with separate on-chip oscillator
 - On-chip analog comparator
 - Interrupt and wake-up on pin change
- Special microcontroller features
 - DebugWIRE on-chip debug system
 - Power-on reset and programmable brown-out detection
 - Internal calibrated oscillator
 - External and internal interrupt sources
 - Five sleep modes: Idle, ADC noise reduction, power-save, power-down, and standby
- I/O and packages
 - 23 programmable I/O lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating voltage:
 - 1.8V - 5.5V for Atmel ATmega48V/88V/168V
 - 2.7V - 5.5V for Atmel ATmega48/88/168
- Temperature range:
 - -40°C to 85°C
- Speed grade:
 - ATmega48V/88V/168V: 0 - 4MHz @ 1.8V - 5.5V, 0 - 10MHz @ 2.7V - 5.5V
 - ATmega48/88/168: 0 - 10MHz @ 2.7V - 5.5V, 0 - 20MHz @ 4.5V - 5.5V
- Low power consumption
 - Active mode:
 - 250µA at 1MHz, 1.8V
 - 15µA at 32kHz, 1.8V (including oscillator)
 - Power-down mode:
 - 0.1µA at 1.8V

Note: 1. See "Data retention" on page 8 for details.



8-bit Atmel
Microcontroller
with 4/8/16K
Bytes In-System
Programmable
Flash

ATmega48/V
ATmega88/V
ATmega168/V

Rev. 2545T-AVR-05/11



Gráfico 39: Características ATML 168

ADXL 345



3-Axis, $\pm 2\text{ g}$ / $\pm 4\text{ g}$ / $\pm 8\text{ g}$ / $\pm 16\text{ g}$
Digital Accelerometer

ADXL345

FEATURES

- Ultralow power: as low as 40 μA in measurement mode and 0.1 μA in standby mode at $V_s = 2.5\text{ V}$ (typical)
- Power consumption scales automatically with bandwidth
- User-selectable resolution
- Fixed 10-bit resolution
- Full resolution, where resolution increases with g range, up to 13-bit resolution at $\pm 16\text{ g}$ (maintaining 4 mg/LSB scale factor in all g ranges)
- Embedded, patent pending FIFO technology minimizes host processor load
- Tap/double tap detection
- Activity/inactivity monitoring
- Free-fall detection
- Supply voltage range: 2.0 V to 3.6 V
- I/O voltage range: 1.7 V to V_s
- SPI (3- and 4-wire) and I²C digital interfaces
- Flexible interrupt modes mappable to either interrupt pin
- Measurement ranges selectable via serial command
- Bandwidth selectable via serial command
- Wide temperature range (-40°C to $+85^\circ\text{C}$)
- 10,000 g shock survival
- Pb free/RoHS compliant
- Small and thin: 3 mm \times 5 mm \times 1 mm LGA package

APPLICATIONS

- Handsets
- Medical instrumentation
- Gaming and pointing devices
- Industrial instrumentation
- Personal navigation devices
- Hard disk drive (HDD) protection
- Fitness equipment

GENERAL DESCRIPTION

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to $\pm 16\text{ g}$. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I²C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0° .

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins. An integrated, patent pending 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor intervention.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm \times 5 mm \times 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

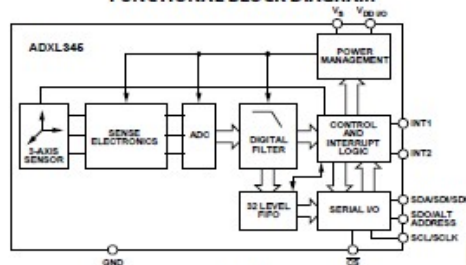


Figure 1.

Rev. 0
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners. See the last page for disclaimers.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2009 Analog Devices, Inc. All rights reserved.

G

ráfico 40: Características ADXL345

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 2.5\text{ V}$, $V_{DDIO} = 1.8\text{ V}$, acceleration = 0 g, $C_S = 1\text{ }\mu\text{F}$ tantalum, $C_{IO} = 0.1\text{ }\mu\text{F}$, unless otherwise noted.

Table 1. Specifications¹

Parameter	Test Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis User selectable		$\pm 2, \pm 4, \pm 8, \pm 16$		g
Nonlinearity	Percentage of full scale		± 0.5		%
Inter-Axis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ²			± 1		%
OUTPUT RESOLUTION					
All g Ranges	Each axis 10-bit resolution		10		Bits
$\pm 2\text{ g}$ Range	Full resolution		10		Bits
$\pm 4\text{ g}$ Range	Full resolution		11		Bits
$\pm 8\text{ g}$ Range	Full resolution		12		Bits
$\pm 16\text{ g}$ Range	Full resolution		13		Bits
SENSITIVITY					
Sensitivity at $X_{OUT}, Y_{OUT}, Z_{OUT}$	Each axis $\pm 2\text{ g}$, 10-bit or full resolution	232	256	286	LSB/g
Scale Factor at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 2\text{ g}$, 10-bit or full resolution	3.5	3.9	4.3	mg/LSB
Sensitivity at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 4\text{ g}$, 10-bit resolution	116	128	143	LSB/g
Scale Factor at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 4\text{ g}$, 10-bit resolution	7.0	7.8	8.6	mg/LSB
Sensitivity at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 8\text{ g}$, 10-bit resolution	58	64	71	LSB/g
Scale Factor at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 8\text{ g}$, 10-bit resolution	14.0	15.6	17.2	mg/LSB
Sensitivity at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 16\text{ g}$, 10-bit resolution	29	32	36	LSB/g
Scale Factor at $X_{OUT}, Y_{OUT}, Z_{OUT}$	$\pm 16\text{ g}$, 10-bit resolution	28.1	31.2	34.3	mg/LSB
Sensitivity Change Due to Temperature			± 0.01		%/ $^\circ\text{C}$
0 g BIAS LEVEL					
0 g Output for X_{OUT}, Y_{OUT}	Each axis	-150	± 40	+150	mg
0 g Output for Z_{OUT}		-250	± 80	+250	mg
0 g Offset vs. Temperature for x-, y-Axes			± 0.8		mg/ $^\circ\text{C}$
0 g Offset vs. Temperature for z-Axis			± 4.5		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise (x-, y-Axes)	Data rate = 100 Hz for $\pm 2\text{ g}$, 10-bit or full resolution		<1.0		LSB rms
Noise (z-Axis)	Data rate = 100 Hz for $\pm 2\text{ g}$, 10-bit or full resolution		<1.5		LSB rms
OUTPUT DATA RATE AND BANDWIDTH					
Measurement Rate ³	User selectable	6.25		3200	Hz
SELF-TEST⁴	Data rate $\geq 100\text{ Hz}$, $2.0\text{ V} \leq V_S \leq 3.6\text{ V}$				
Output Change in x-Axis		0.20		2.10	g
Output Change in y-Axis		-2.10		-0.20	g
Output Change in z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V_S)	$V_S \leq 2.5\text{ V}$	2.0	2.5	3.6	V
Interface Voltage Range (V_{DDIO})	$V_S \geq 2.5\text{ V}$	1.7	1.8	V_S	V
Supply Current	Data rate $> 100\text{ Hz}$		145		μA
	Data rate $< 10\text{ Hz}$		40		μA
Standby Mode Leakage Current			0.1	2	μA
Turn-On Time ⁵	Data rate = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	$^\circ\text{C}$
WEIGHT					
Device Weight			20		mg

¹ All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

² Cross-axis sensitivity is defined as coupling between any two axes.

³ Bandwidth is half the output data rate.

⁴ Self-test change is defined as the output (g) when the SELF_TEST bit = 1 (in the DATA_FORMAT register) minus the output (g) when the SELF_TEST bit = 0 (in the DATA_FORMAT register). Due to device filtering, the output reaches its final value after $4 \times \tau$ when enabling or disabling self-test, where $\tau = 1/(\text{data rate})$.

⁵ Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately $\tau + 1.1$ in milliseconds, where $\tau = 1/(\text{data rate})$.

Gráfico 41: Especificaciones Técnicas y Eléctricas ADXL345

L298



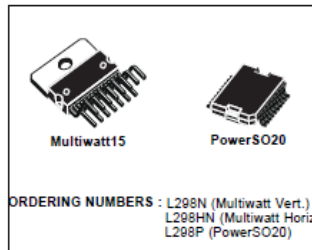
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 48 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

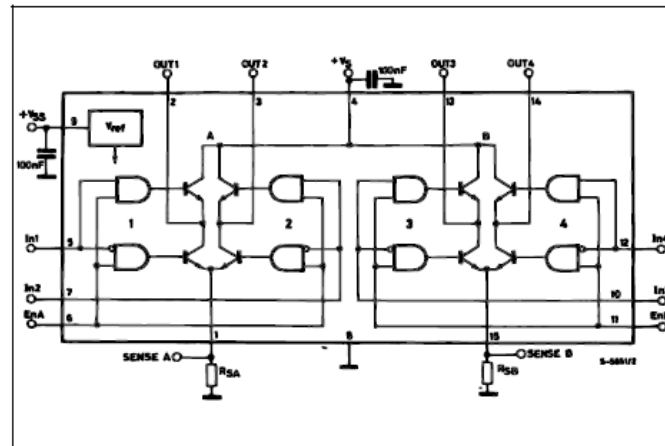
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM




Document 00000

4/40

Gráfico 42: Especificaciones l298

IMU 3000

	IMU-3000 Product Specification	Document Number: PS-IMU-3000A-00-01.1 Revision: 1.1 Release Date: 08/19/2010
---	--------------------------------	--

2 Features

The IMU-3000 Motion Processing Unit includes a wide range of features:

2.1 Sensors

- X-, Y-, Z-Axis angular rate sensors (gyros) on one integrated circuit
- Digital-output temperature sensor
- 6-axis motion processing capability using secondary I²C interface to directly connect to a digital 3-axis third-party accelerometer
- Factory-calibrated scale factor
- High cross-axis isolation via proprietary MEMS design
- 10,000g shock tolerant

2.2 Digital Output

- Fast Mode (400kHz) I²C serial interface
- 16-bit ADCs for digitizing sensor outputs
- Angular rate sensors (gyros) with applications-programmable full-scale-range of $\pm 250^\circ/\text{sec}$, $\pm 500^\circ/\text{sec}$, $\pm 1000^\circ/\text{sec}$, or $\pm 2000^\circ/\text{sec}$.

2.3 Motion Processing

- Embedded Digital Motion Processing™ (DMP™) engine supports 3D motion processing. When used together with a digital 3-axis third party accelerometer, the IMU-3000 collects the accelerometer data via a dedicated interface, while synchronizing data sampling at a user defined rate. The total data set obtained by the IMU-3000 includes 3-axis gyroscope data, 3-axis accelerometer data, and temperature data.
- FIFO buffers complete data set, reducing timing requirements on the system processor and saving power by letting the processor burst read the FIFO data, and then go into a low-power sleep mode while the IMU collects more data.
- Data collection polled or interrupt driven with on-chip programmable interrupt functionality
- Programmable low-pass filters

2.4 Clocking

- On-chip timing generator clock frequency $\pm 2\%$ variation over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz
- 1MHz clock output to synchronize with digital 3-axis accelerometer


2.5 Power

- VDD analog supply voltage range of 2.1V to 3.6V
- Flexible VLOGIC reference voltage allows for I²C interface voltages from 1.71V to VDD
- Power consumption with all three axis active: 6.1mA
- Sleep mode: 5 μ A
- Each axis can be individually powered down

2.6 Package

- 4x4x0.9mm QFN plastic package
- MEMS structure hermetically sealed and bonded at wafer level
- RoHS and Green compliant

Gráfico 43: Características IMU3000

	IMU-3000 Product Specification	Document Number: PS-IMU-3000A-00-01.1 Revision: 1.1 Release Date: 08/19/2010
---	--------------------------------	--

9 Register Map

Addr (Hex)	Addr (Decimal)	Register Name	R/W	Bit7	Bits	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	WHO_AM_I	R/W	0	ID						-
C	12	X_OFFS_USRH	R/W	X_OFF_H							
D	13	X_OFFS_USRL	R/W	X_OFF_L							
E	14	Y_OFFS_USRH	R/W	Y_OFFS_H							
F	15	Y_OFFS_USRL	R/W	Y_OFFS_L							
10	16	Z_OFFS_USRH	R/W	Z_OFFS_H							
11	17	Z_OFFS_USRL	R/W	Z_OFFS_L							
12	18	FIFO_EN	R/W	TEMP_OUT	GYRO_XOUT	GYRO_YOUT	GYRO_ZOUT	AUX_XOUT	AUX_YOUT	AUX_ZOUT	FIFO_FOOTER
13	19	AUX_VDDIO	R/W	0	0	0	0	0	AUX_VDDIO	0	0
14	20	AUX_SLV_ADDR	R/W	CLKOUT_EN	AUX_ID						
15	21	SMPLRT_DIV	R/W	SMPLRT_DIV							
16	22	DLPF_FS	R/W	0	0	0	FS_SEL		DLPF_CFG		
17	23	INT_CFG	R/W	ACTL	OPEN	LATCH_INT_EN	INT_ANYRD_2CLEAR	I2C_MST_ERR_EN	IMU_RDY_EN	DMP_DONE_EN	RAW_RDY_EN
18	24	AUX_BURST_ADDR	R/W	BURST_ADDR							
1A	26	INT_STATUS	R	FIFO_FULL	-	-		I2C_MST_ERR	IMU_RDY	DMP_DONE	RAW_DATA_RDY
1B	27	TEMP_OUT_H	R	TEMP_OUT_H							
1C	28	TEMP_OUT_L	R	TEMP_OUT_L							
1D	29	GYRO_XOUT_H	R	GYRO_XOUT_H							
1E	30	GYRO_XOUT_L	R	GYRO_XOUT_L							
1F	31	GYRO_YOUT_H	R	GYRO_YOUT_H							
20	32	GYRO_YOUT_L	R	GYRO_YOUT_L							
21	33	GYRO_ZOUT_H	R	GYRO_ZOUT_H							
22	34	GYRO_ZOUT_L	R	GYRO_ZOUT_L							
23	35	AUX_XOUT_H	R	AUX_XOUT_H							
24	36	AUX_XOUT_L	R	AUX_XOUT_L							
25	37	AUX_YOUT_H	R	AUX_YOUT_H							
26	38	AUX_YOUT_L	R	AUX_YOUT_L							
27	39	AUX_ZOUT_H	R	AUX_ZOUT_H							
28	40	AUX_ZOUT_L	R	AUX_ZOUT_L							
35	53	DMP_REG1	R/W	RESERVED1							
36	54	DMP_REG2	R/W	RESERVED2							
37	55	DMP_REG3	R/W	RESERVED3							
38	56	DMP_REG4	R/W	RESERVED4							
39	57	DMP_REG5	R/W	RESERVED5							
3A	58	FIFO_COUNTH	R	-	-	-	-	-	-	FIFO_COUNT_H	
3B	59	FIFO_COUNTL	R	FIFO_COUNT_L							
3C	60	FIFO_R	R	FIFO_DATA							
3D	61	USER_CTRL	R/W	DMP_EN	FIFO_EN	AUX_IF_EN	-	AUX_IF_RST	DMP_RST	FIFO_RST	GYRO_RST
3E	62	PWR_MGM	R/W	H_RESET	SLEEP	STBY_XG	STBY_YG	STBY_ZG	CLK_SEL		

Gráfico 44: Mapa de Registros IMU3000

EASY ROLLER

2. Easy Roller Specifications and Assembly

2.1 DC Gearhead Motor Characteristics

Characteristic	Value	Unit
M12V200		
Operating voltage	4.5-12	V
No load current	111	mA
No load speed	200	RPM
Startup torque	3.6	KG•CM
Gear ratio	30:1	

2.2 Neoprene Wheel Dimensions

Characteristic	Value	Unit
WHL2		
Wheel width	0.688	Inch
Wheel diameter	2.125	Inch
Color	White	
Material	Plastic/O-ring	
Set screw	6-32	

2.3 Mechanical Dimensions

All dimensions listed below are in mils (0.001 inches).

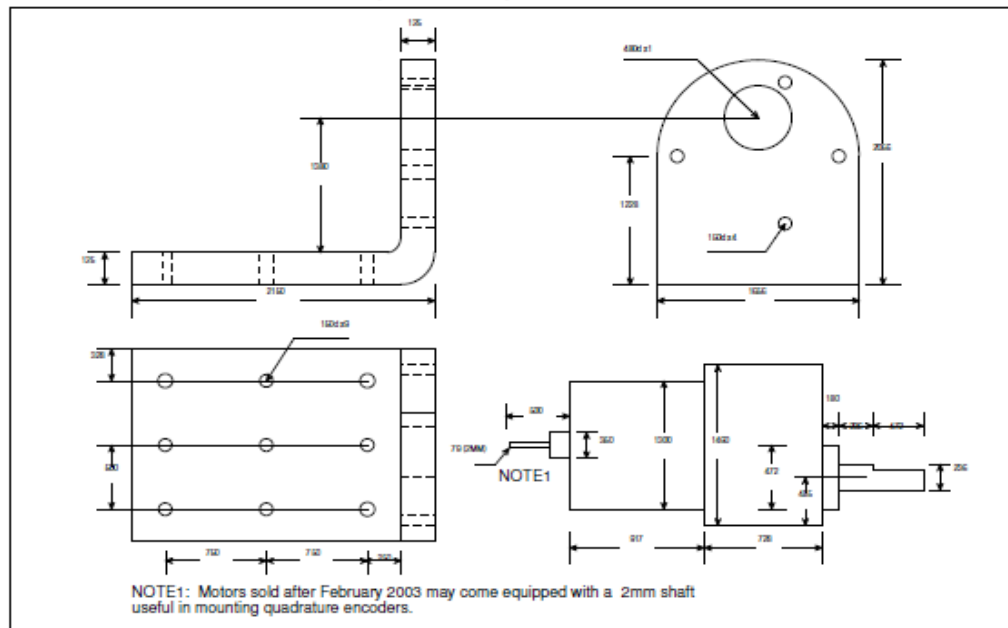


Gráfico 45: Características Easy Roller