



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Control PID de Velocidad de motor DC usando comunicación mediante
Señales de Radio Comunicación (RC)"

TESINA DE SEMINARIO

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentada por:

RAÚL XAVIER MATA SANTOS.

TITO JOSE DURAN DURAN.

GUAYAQUIL - ECUADOR

AÑO: 2011

AGRADECIMIENTO

A nuestro tutor, el Ing. Carlos Valdiviezo, así como a las demás personas que han formado parte del seminario, con quienes hemos compartido esta gran experiencia de aprendizaje e investigación.

También debemos agradecer a todos los que han aportado en ayudarnos a cumplir con nuestra meta: amigos, profesores, y autoridades de este prestigioso establecimiento educativo.

DEDICATORIA

Dedico este trabajo principalmente a Dios. A mis padres, a mis hermanos por ser quienes me apoyaron durante mi carrera universitaria. A todos mis amigos y personas queridas a los cuales guardaré siempre mi eterna gratitud.

Raúl Xavier Mata Santos.

A mis padres, a mis hermanos que siempre me apoyaron y creyeron en mí y especialmente a Dios y a todos los que me apoyaron en este camino y me motivaron para continuar en esta meta tan importante en mi vida.

Tito José Dura Duran.

TRIBUNAL DE SUSTENTACIÓN



Ing. Carlos Valdiviezo

PROFESOR DEL SEMINARIO



Ing. Hugo Villavicencio

PROFESOR DELEGADO

DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL).



Raúl Xavier Mata Santos.



Tito José Duran Duran.

RESUMEN

En el **primer capítulo** se realiza un análisis del problema, con el cual se intenta tener una mejor noción de la situación actual del proceso, y todos los factores que intervienen, así como cualquier tipo de perturbación existente.

En el **segundo capítulo** se analizan todas las herramientas necesarias para tener la capacidad de proponer una solución válida que tenga bases teóricas, se hará referencia a los conocimientos obtenidos durante el proceso de aprendizaje y que de alguna manera ayuden a entender un poco mejor la problemática.

En el **tercer capítulo** se presenta una etapa más práctica en la cual se realiza un diseño de la solución del problema que estamos analizando, esta solución como ya se lo había comentado debe tener bases teóricas para poder ser aplicado a la práctica.

En el **cuarto capítulo**, veremos la implementación de la solución con las herramientas escogidas y las consideraciones necesarias para poder llevar a cabo una buena comprobación de la solución diseñada, para corroborar que cumpla con las expectativas y además de tener una respuesta muy similar basado en las características de nuestro diseño mediante los software para el funcionamiento del controlador.

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL	IV
DECLARATORIA EXPRESA.....	V
RESUMEN	VI
INDICE GENERAL	VII
INDICE FIGURA	XI
INTRODUCCION	XIV

Capítulo 1

DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1 Antecedentes.....	1
1.2 Descripción del Proyecto.....	5
1.3 Limitaciones del Proyecto.....	6
1.4 Analisis de Soluciones Similares existente en el mercado.....	8
1.4.1 Pololu Jrk 12v2.....	8
1.4.2 Controlador de velocidad para motor DC máx. 60 000 rpm.....	9
1.4.3 Controlador de velocidad para motor DC 10-60 VDC RS 200.....	10
1.4.4 Dispositivo de velocidad digital de un motor.....	11

Capítulo 2

HERRAMIENTAS DEL SISTEMA	12
2.1 Antecedentes.....	12
2.1.1 Modulación PWM.....	13
2.2 Sistema de Control PID.....	14
2.3 Descripción de Componentes.....	15
2.3.1 Pololu JRK 21v3 USB Motor Controller with Feedback.....	15
2.3.1.1 Aspectos destacados acerca del Pololu Jrk 21v3 USB Motor Controller with Feedback.....	16
2.3.1.2 Diagrama de un sistema típico de retroalimentación.....	18
2.3.1.2.1 Señales de Entrada y Salida del Jrk.....	18
2.3.1.2.2 Input.....	18
2.3.1.2.3 Scaled Feedback.....	19
2.3.1.2.4 Current.....	19
2.3.1.2.5 Error.....	19
2.3.2 10:1 Micro Metal Gearmotor.....	20
2.3.2.1 Aspectos destacados del 10:1 Micro Metal Gearmotor.....	21
2.3.3 RC Snap Circuits Rover.....	22
2.3.4 Joystick.....	23
2.3.5 ATtyni 2313.....	24
2.3.6 Encoder para neumático Pololu 42x19mm.....	25
2.3.6.1 Aspectos destacados del Encoder para neumático Pololu 42x19m.....	26
2.3.7 Herramientas de Software.....	27
2.3.7.1 Jrk Configuration Utility.....	27
2.3.7.1.1 Pestaña de Input.....	28
2.3.7.1.2 Pestaña de Feedback.....	29
2.3.7.1.3 Pestaña del PID.....	30
2.3.7.1.4 Pestaña del Motor.....	32
2.3.7.1.5 Pestaña de Error.....	33
2.3.7.1.6 Pestaña de Plop de Window.....	34
2.3.7.2 Proteus 7.7.....	35
2.3.7.3 AVR Studio 4.....	36

Capítulo 3

DISEÑO DE LA SOLUCIÓN	37
3.1 Diseño Preliminar.....	37
3.2 Pruebas Preliminares.....	37
3.3 Comunicación entre Pc y Usuario.....	38
3.4 Diseño del Software y Hardware.....	46
3.4.1 Diagrama de Bloques General.....	46
3.4.2 Diagrama de Bloques Detallado.....	46
3.4.2.1 Bloques.....	47
3.4.2.1.1 Transición.....	47
3.4.2.1.2 Recepción.....	48
3.4.2.1.3 Comunicación.....	49
3.4.2.1.4 Sistema de Control PID.....	50
3.4.2.1.4.1 Usando el Interfaz Serial.....	52
3.4.2.1.4.1.1 UART.....	53
3.4.2.1.4.1.2 TTL Serial.....	54
3.4.2.1.4.2 Protocolo de Comandos.....	56
3.4.2.1.4.2.1 Protocolo Compacto.....	56
3.4.2.1.4.2.2 Protocolo del Pololu.....	57
3.4.2.1.4.3 Comando de control de los Motores.....	58
3.4.2.1.4.3.1 Motor Apagado.....	58
3.4.2.1.4.3.2 Set Target High Resolution.....	59
3.4.2.1.4.3.3 Set Target Low Resolution Forward.....	59
3.4.2.1.4.3.4 Set Target Low Resolution Reverse.....	60
3.5 Modelo 3D.....	61
3.5.1 Control PID de velocidad de motor DC usando comunicación mediante señales de Radio Comunicación (RC).....	61
3.5.2 Encoder con rueda.....	62
3.5.3 Pololu Jrk 21v3.....	62
3.5.4 Cable USB.....	63
3.5.5 R/C Receiver.....	63
3.5.6 Joystick.....	64
3.5.7 ATtyni 2313.....	64

Capítulo 4

SIMULACION/IMPLEMENTACION	65
4.1 Simulación del Proyecto.....	65
4.1.1 Ciclo de trabajo del sistema.....	65
4.1.2 La Señal de retroalimentación.....	68
4.1.3 La Señal Integral.....	69
4.1.3 La Señal de Error.....	70
4.3 Simulación del AVR con el ATMEL AVR Studio 4.....	71
4.3.1 Primeros Pasos.....	72
4.3.2 Selección de Dispositivos.....	73
4.3.3 IDE Windows.....	74
4.3.4 Opciones de simulación.....	75
4.4 Análisis del código Fuente en Lenguaje C en AVR Studio 4.....	76
4.6 Implementación del Proyecto.....	80

ÍNDICE DE FIGURAS

FIGURA 1.1A CLEPSYDRA.....	2
FIGURA 1.1B RELOJ MECÁNICO	2
FIGURA 1.1C MOLINO DE VIENTO	2
FIGURA 1.1D MAQUINA DE VAPOR	3
FIGURA 1.1E MOTOR DE CORRIENTE CONTINUA.....	4
FIGURA 1.1F MOTOR DE CORRIENTE ALTERNA.....	4
FIGURA 1.2A POLOLU JRK 21V3 R/C MOTOR CONTROLLER WITH FEEDBACK.....	6
FIGURA 1.4.1A POLOLU JRK 21V3.....	8
FIGURA 1.4.2A CONTROLADOR DE SERIE PI.....	9
FIGURA 1.4.3A CONTROLADOR ELECTRÓNICO RS200.....	10
FIGURA 1.4.4A GOVERNOR	11
FIGURA 2.1.1A UNA SEÑAL DE ONDA CUADRADA MOSTRANDO CICLO DE TRABAJO ...	13
FIGURA 2.2 DIAGRAMA DE BLOQUES DE UN CONTROL PID	14
FIGURA 2.3.1POLOLU JRK 21V3 USB MOTOR CONTROLLER WITH FEEDBACK.....	15
FIGURA 2.3.1.1A PINES DE COMUNICACIÓN SERIAL UART.....	17
FIGURA 2.3.1.2DIAGRAMA DE UN SISTEMA TÍPICO DE RETROALIMENTACIÓN	18
FIGURA 2.3.2 MICROMETAL GEARMOTOR 10:1	19
FIGURA 2.3.3A RC SNAP CIRCUITS ROVER	20
FIGURA 2.3.4A JOYSTICK	21
FIGURA 2.3.5A ATTYNI 2313.....	22
FIGURA 2.3.6A ENCODER POLOLU 42X19 MM.....	23
FIGURA 2.3.6B ENCODER(CODIFICADOR DE EJE O GENERADOR DE PULSÓS).....	24
FIGURA 2.3.7.1A WINDOWS VISTA DEVICE MANAGER SHOWING POLOLU JRK 21V3 ..	25
FIGURA 2.3.7.1.1A PESTAÑA INPUT.....	26
FIGURA 2.3.7.1.2A PESTAÑA FEEDBACK	27
FIGURA 2.3.7.1.3A PESTAÑA PID.....	28
FIGURA 2.3.7.1.4A PESTAÑA MOTOR.....	30
FIGURA 2.3.7.1.5A PESTAÑA ERROR.....	31

FIGURA 2.3.7.1.6A THE PLOTS WINDOWS	32
FIGURA 2.3.7.2A PROTEUS	33
FIGURA 2.3.7.3A AVR STUDIO 4.....	34
FIGURA 3.3A PAGINA DE DESCARGA	37
FIGURA 3.3B ÍTEM JRK 21V3 USB MOTOR CONTROLLER WITH FEEDBACK.....	38
FIGURA 3.3C POLOLU JRK 21V3 USB MOTOR CONTROLLER WITH FEEDBACK.....	38
FIGURA 3.3D LINK DE DESCARGA PDF DE DATASHEET DE ÍTEM JRK 21V3.....	39
FIGURA 3.3E LINK DE DESCARGA DE DRIVERS Y SOFTWARE.....	40
FIGURA 3.3F DESCARGA JRK_WINDOWS_091218.ZIP.....	41
FIGURA 3.3G CUADRO WINRAR CON CARPETA DE ARCHIVO EJECUTABLE	41
FIGURA 3.3H ARCHIVO EJECUTABLE.....	42
FIGURA 3.3I INICIO DE INSTALACIÓN	42
FIGURA 3.3J SELECCIÓN DE CARPETA DONDE IRÁ APLICACIÓN.....	43
FIGURA 3.3K CONFIRMACION DE INSTALACIÓN	43
FIGURA 3.3L INSTALANDO	44
FIGURA 3.3M INSTALACIÓN EXITOSA.....	44
FIGURA 3.4.1 DIAGRAMA DE BLOQUES GENERAL.....	45
FIGURA 3.4.2 DIAGRAMA DE BLOQUES DETALLADO.....	45
FIGURA 3.4.2.1.1A DIAGRAMA DEL TRANSMISOR	46
FIGURA 3.4.2.1.2A DIAGRAMA DE BLOQUES DEL RECEPTOR	47
FIGURA 3.4.2.1.2B R/C RECEIVER	48
FIGURA 3.4.2.1.3A ATTYNI 2313	48
FIGURA 3.4.2.1.4A DIAGRAMA DE BLOQUES DEL SISTEMA PID.....	49
FIGURA 3.4.2.1.4B INTERFAZ DEL JRK 21V3.....	51
FIGURA 3.4.2.1.4.1.1A UART SERIAL.....	53
FIGURA 3.4.2.1.4.1.2 TTL ASINCRÓNICA NO-INVERTIDA	54
FIGURA 3.5.1A CONTROL PID DE VELOCIDAD DE MOTOR DC USANDO COMUNICACIÓN MEDIANTE SEÑALES DE RADIO COMUNICACIÓN(RC).....	60
FIGURA 3.5.2A ENCODER CON RUEDA CON SU BASE DE PLASTICO	61
FIGURA 3.5.3A TARJETA POLOLU JRK 21V3	61
FIGURA 3.5.4 CABLE USB	62
FIGURA 3.5.5A R/C RECEIVER.....	62
FIGURA 3.5.6A JOYSTICK	63
FIGURA 3.5.7A ATTYNI 2313	63

FIGURA 4.1.1A CICLO DE TRABAJO (SISTEMA ACELERANDO).....	63
FIGURA 4.1.1B CICLO DE TRABAJO (SISTEMA DESACELERADO).....	64
FIGURA 4.1.1C CICLO DE TRABAJO (SISTEMA ACELERADO EN SENTIDO INVERTIDO) .	65
FIGURA 4.1.2A SEÑAL DE RETROALIMENTACIÓN	66
FIGURA 4.1.3A SEÑAL INTEGRAL.....	67
FIGURA 4.1.4A ERROR (MOTOR EN CONTRA DE LAS MANECILLAS DE RELOJ).....	68
FIGURA 4.1.4BERROR (MOTOR EN SENTIDO DE LAS MANECILLAS DE RELOJ)	68
FIGURA 4.3AVR STUDIO 4	69
FIGURA 4.3.1A AVR IDE.....	70
FIGURA 4.3.1 ABRIR ARCHIVO	71
FIGURA 4.3.2 DISPOSITIVO Y PLATAFORM DE DEPURACIÓN.....	71
FIGURA 4.3.3 IDE WINDOWS.....	72
FIGURA 4.3.4A DEBUG MENU	73
FIGURA 4.3.4B SIMULADOR DE FRECUENCIA DE RELOJ	73
FIGURA 4.5A SIMULACIÓN – TRANSMISIÓN UART CON EL ATTYNI 2313	77
FIGURA 4.5B TRANSMISIÓN COMPAC PROTOCOL AUMENTO DE VELOCIDAD HEXAECIMAL Y BINARIO.....	77
FIGURA 4.5B TRANSMISIÓN COMPAC PROTOCOL DISMINUCIÓNDE VELOCIDAD HEXAECIMAL Y BINARIO.....	77
FIGURA 4.6A PROYECTO IMPLEMENTADO	78

INTRODUCCIÓN

La gran mayoría de aplicaciones requiere algún tipo de control. El control automático es el mantenimiento de un valor deseado dentro de una cantidad o condición, midiendo el valor existente, comparándolo con el valor deseado, y utilizando la diferencia para poder reducirla. En consecuencia, el control automático exige un lazo cerrado de acción y reacción que funcione sin intervención humana.

El uso de los sistemas de control se ha incrementado en los últimos años debido principalmente al éxito en distintas aplicaciones en el Campo de las Telecomunicaciones como en la parte Industrial. Nuestro Sistema de Control basado en el POLOLU Jrk 21v3 comprende una familia de algoritmos que previamente ha sido incorporado a su diseño mediante software como también explícitamente para modelar diversos procesos para determinar los parámetros necesarios de la variables de las cuales vamos a manipular en nuestras entradas reduciendo el error entre la variable manipulada y el punto de ajuste, en respuestas a cambios del mismo, de las perturbaciones del sistema o de las condiciones operacionales.

Para diseñar un sistema de control es necesario conocer su modelo. Una fase importante en el diseño es la identificación que tiene como objetivo que el modelo obtenido reproduzca con suficiente exactitud el comportamiento del proceso, en el presente trabajo se verá cómo de la exactitud del modelo obtenido dependerá posteriormente el buen comportamiento del controlador diseñado.

Existen dos métodos para la obtención de modelos: modelado teórico (método analítico) e identificación del sistema (método experimental). Para el modelado se requiere un conocimiento muy especializado sobre la tecnología del proceso, mientras que para la identificación del sistema (que es el método más directo) se requiere aplicar al proceso señales especiales como escalones, rampas, impulsos, sinusoides o señales pseudo aleatorias y registrar las salidas que se producen. Existen numerosos métodos de identificación tanto paramétricos como no paramétricos que ofrecen información variada sobre el sistema en estudio.

CAPÍTULO 1

1. DESCRIPCIÓN GENERAL DEL PROYECTO

1.1. Antecedentes

La aplicación del principio de retroalimentación tiene sus comienzos en máquinas e instrumentos muy sencillos, algunos de los cuales se remontan a 2000 años atrás.

El aparato más primitivo que emplea el principio de control por realimentación fue desarrollado por un griego llamado Ktsibios. En el siglo III aproximadamente 300 años A.C., en la Antigua Grecia, se trataba de un reloj de agua también conocido como Clepsydra (**Ver Figura 1.1a**), el cual medía el pasaje del tiempo por medio de un pequeño chorro de agua que fluía a velocidad constante dentro de un recipiente. El mismo poseía un flotante que subía a medida que el tiempo transcurría debido a las varias marcas en su interior que disponía el recipiente de tal manera que el nivel de agua indicaba los diferentes periodos, tanto diarios como nocturnos.

Ktsibios resolvió el problema del mantenimiento del caudal constante de agua inventando un aparato semejante usado en los carburadores de los motores modernos. Entre el suministro de agua y el tanque colector había una regulación de caudal del agua por medio de una válvula flotante que mantenía el nivel constante. Si el nivel se elevaba restringe el caudal del agua en el recipiente regulador hasta que el flotante volvía al nivel específico.

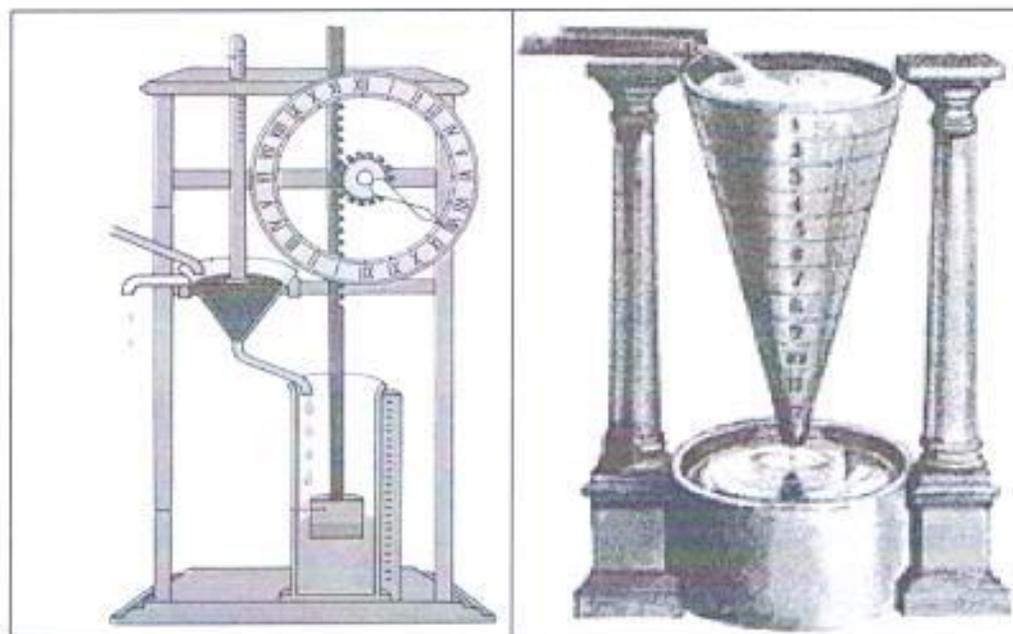


Figura 1.1a Clepsydra

En el siglo IX el regulador de nivel a flotante es reinventado en Arabia. En este caso se usaba para mantener el nivel constante en los bebedores de agua. En el siglo XVI, en Inglaterra se usaba el principio de realimentación para mantener automáticamente las paletas de los molinos de viento (Ver Figura 1.1b) como a su vez la invención del reloj mecánico (Ver Figura 1.1c) que hizo obsoleto el reloj de agua y sus sistemas de Feedback o retroalimentación. (El reloj mecánico no es un sistema de Feedback). El regulador Flotante no aparece otra vez hasta su uso en la Revolución Industrial alrededor de 1750

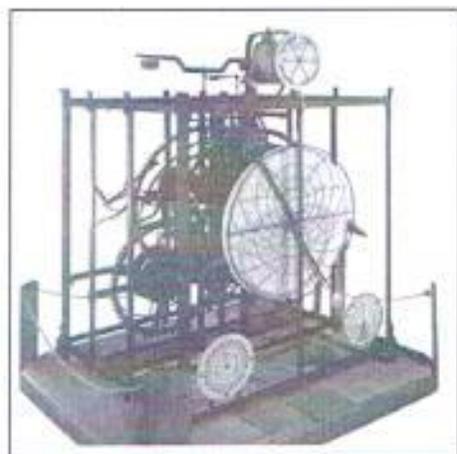


Figura 1.1b Reloj Mecánico



Figura 1.1c Molino de viento

El primer uso del control automático en la industria parece haber sido el regulador centrífugo de la máquina de vapor de Watt, en el año 1775. Este aparato fue utilizado para regular la velocidad del motor manipulando el caudal de vapor por medio de una válvula. Encontrándose presentes todos los elementos que abarca un sistema de control con realimentación (Ver Figura 1d). Fue el motor inicial de la Revolución Industrial que impulsa a la elaboración y construcción de motores en la actualidad. Desde fines del siglo XVIII en Inglaterra y desde casi mediados del siglo XIX, la construcción de motores y su respectivo uso aceleró asombrosamente el desarrollo económico de muchos de los principales países de la Europa Occidental y de los Estados Unidos.

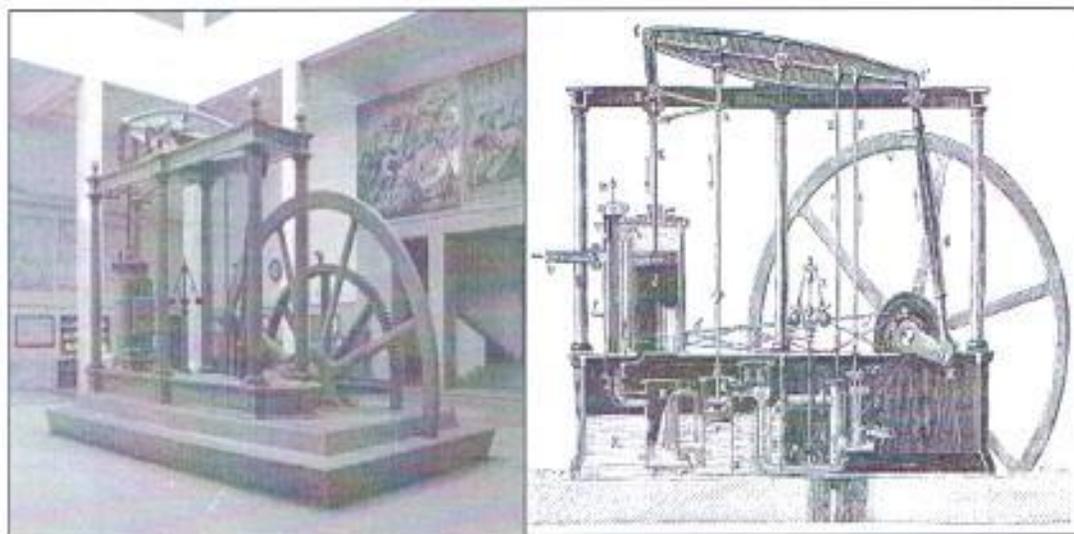


Figura 1.1d Máquina de Vapor

El primer análisis de Control Automático es la explicación matemática del regulador centrífugo por James clerk Maxwell en 1868, Mas tarde la técnica del regulador se adjudicó a otras maquinas y a principio del siglo XX comenzó la aplicación de reguladores y servomecanismos en reguladores de energía térmica en buques.

Con el transcurrir del tiempo la zona industrial comenzó a crecer enormemente en el aspecto tanto económico como tecnológico, creando así los motores de corriente continua (**Ver Figura 1.1e**). Estos motores son de lo más versátiles en la industria. Su fácil control de posición, paro y velocidad los han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos. Pero con la llegada de los motores de corriente alterna (**Ver Figura 1.1f**), del tipo asincrónico, 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, etc.)



Figura 1.1e Motor de Corriente Continua

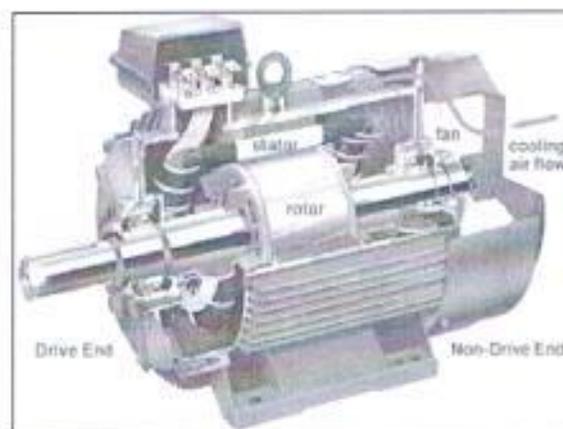


Figura 1.1e Motor de Corriente Alterna

Hoy en día, el control de sistemas es una de las áreas de estudio más interesantes y completas para el alumno de Ingeniería Electrónica, tanto por la complejidad que en sí mismo representa el control, como por la amplia gama de campos de conocimiento paralelos que se han de dominar a la hora de modelar el sistema (mecánica, teoría de fluidos, transmisión de calor... etc., dependiendo del tipo de sistema), diseñar el actuador que actuará sobre la planta del sistema (electrónica de potencia, funcionamiento de máquinas eléctricas...), o diseñar el circuito de acondicionamiento de medida de señales (electrónica analógica, instrumentación electrónica...).

1.2. Descripción del proyecto

Son muy amplias las aplicaciones que tiene el control de sistemas, por lo cual el presente proyecto trata sobre el Control PID de Velocidad de un Motor DC usando comunicación mediante Señales de Radio Comunicación (RC) que está controlado con un circuito electrónico el cual consta de un POLOLU JRK 21v3 (**Ver Figura 1.2a**), como principal componente, su entrada principal está formado por un receptor inalámbrico conectado a un Microcontrolador ATtiny 2313 que maneja las salidas del receptor y las envía al JRK mediante comunicación UART.

Actualmente existen en el mercado diferentes tipos de dispositivos electrónicos para controlar la velocidad de un motor que podrían compararse con nuestro proyecto, pero la diferencia es que nosotros tenemos la capacidad de comunicación serial y una fuente de energía de 4 pilas recargables AA, teniendo como diferencia la adquisición de energía para el funcionamiento del dispositivo.

El Pololu Jrk 21v3 USB Motor Controller with Feedback altamente configurable que soporta 4 tipos de modos de interface: USB, nivel lógico serial, voltaje analógico y control RC. Este controlador puede ser usado de 3 maneras distintas ya

sea como control de velocidad de lazo abierto, control de velocidad de lazo cerrado o control de posición.

La amplia operación de rango comprendida entre 5 a 28 Voltios y salida de corriente continua de 3 Amperios permite controlar la mayoría de motores pequeños DC. Voltajes Analógicos y Frecuencias en tacómetros de las opciones de retroalimentación permiten rápida implementación de sistemas de lazo cerrados, y también el software permite fáciles configuraciones y calibraciones por medio del puerto USB.

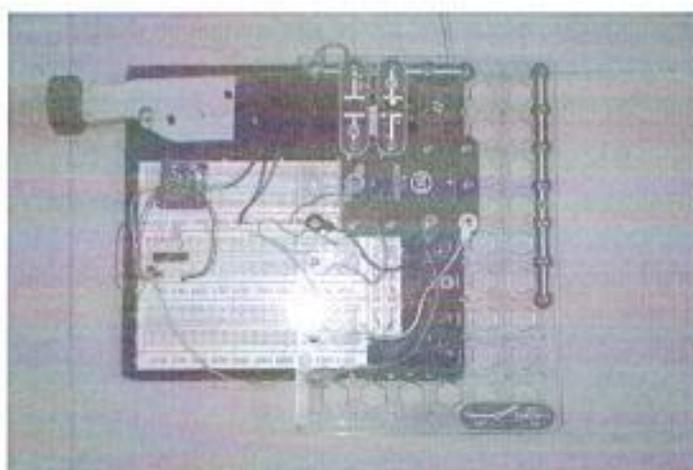


Figura 1.2a Pololu Jrk 21v3 R/C Motor Controller with Feedback

1.3. Limitaciones del Proyecto

Este proyecto contiene algunas limitantes que es recomendable tener en consideración al momento de implementar el mismo, a continuación presentamos algunas de estas limitaciones y sus respectivas consecuencias:

- **Sensores de movimiento.**- Los sensores tienen un gran campo de visión, los cuales visualizan casi toda el área alrededor de ellos para ejecutar ya sea una señal o un tipo de respuesta los cuales son muy precisos al momento del muestreo y rápidos en la captación de señal pero en el momento de divisar algún tipo de obstáculos en el lugar adjuntándole algún objeto siempre y cuando se

encuentre al nivel de este sensor no lo tomara en cuenta por lo que la retroalimentación que demandaría el motor a su entrada no son con las características deseada con lo que uno necesita que funcione.

- **Comunicación UART.-** Existen en la actualidad diferentes ejemplos de puertos que comunican información de manera serial (un bit a la vez). El conocido como "puerto serial" ha sido gradualmente reemplazado por el puerto USB (Universal Serial Bus) que permite mayor versatilidad en la conexión de múltiples dispositivos. Aunque en naturaleza serial, no suele referenciarse de esta manera ya que sigue sus propios estándares y no los establecidos por el RS-232.

El Microcontrolador ATtiny 2313 posee un módulo para comunicación serial, USART. Para utilizar este módulo, necesitamos configurar varios registros, para ver si se usará un bit de paridad, el número de bits de los datos, el número de stop bits entre otros. El módulo USART posee un generador de tasa de transferencia (baud rate), controlado por el registro UBRR. La fórmula para obtener el valor de UBRR depende de la frecuencia de operación del Microcontrolador y de los baudios que se quiera obtener. Al momento de realizar las configuraciones necesarias para usar la comunicación serial, los caracteres que enviábamos no reflejaban a las tramas deseadas al momento de simularlo por lo que el registro UDR guardaba otro tipo de trama y al enviarlo por el puerto RX y TX al POLOLU JRK 21v3 no respondía al caracter deseado. Es por esta razón que sea conveniente verificar aunque con que sólo exista un registro para la transmisión y recepción de datos a través del puerto serial es suficiente para conocer qué es lo que se está enviando .

1.4 Análisis de soluciones similares existentes en el mercado

Actualmente en el mercado existe una gama de dispositivos y productos que pueden dar gran parte de la solución a los problemas planteados anteriormente ya que ofrecen optimización en nuestro proyecto, pero también incrementan sus costo, pero pueden mejorar la exactitud y rapidez para el control de velocidad del motor DC haciéndolo más eficiente

Las soluciones posibles a los problemas planteados son:

1.4.1. POLOLU JRK 12v2

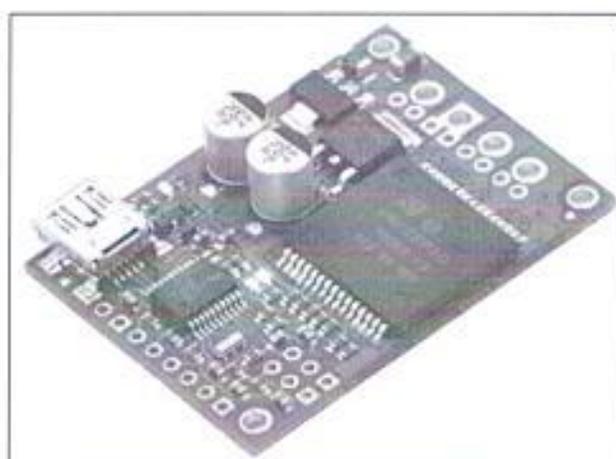


Figura 1.4.1a Pololu Jrk 12v12

El poderoso controlador Jrk 12v12 (Ver Figura 1.4.1a), altamente configurable es un controlador de motor de corriente continua que soporta cuatro modos de interfaz: USB, serie lógica de nivel, voltaje analógico, y el control de radio (RC). El controlador puede ser usado como control de velocidad de lazo abierto o de velocidad de lazo cerrado o control de posición. La salida de corriente continua es de aproximadamente 12 A (30 A máx) y el rango de operación de 16.6 V (con protección contra picos de hasta 40 V). Este dispositivo es similar al Pololu jrk

21v3 pero por su mayor rango de eficiencia y versatilidad a la hora de solucionar problemas lo hacen mucho más exacto y rápido.

1.4.2. Controlador de velocidad para motor DC máx. 60 000 rpm

El controlador de serie PI (Ver Figura 1.4.2a) viene con software configurable por lo que no necesita de Trimmers o switches DIP, este controlador puede ser operado fácilmente por el usuario y a su vez por el mismo, ya sea como motor de corriente DC o como unidad de conmutación electrónica, como con sensores integrados. Dependiendo del modo operativo, la velocidad puede ser controlada entre 50 y 60,000 RPM.

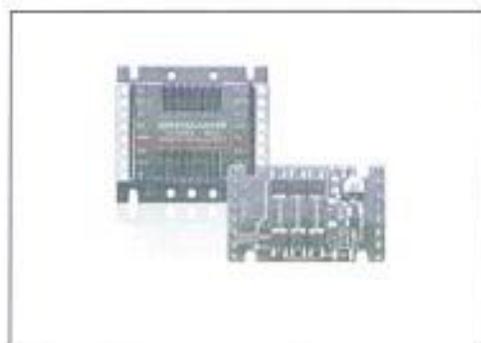


Figura 1.4.2a Controlador de Serie PI

Usuarios pueden manipular libremente los propósitos del controlador. Entre los últimos software 4.5 "Motion Manager" permite específicas familiares programaciones con esta herramienta fácil de usar. Límites de poder, parámetros de control, velocidades ajustables, resoluciones de encoder e insertar definición de valores puede ser configurado usando la señal PWM, con parámetros variables para obtener la máxima velocidad y operación de los motores DC y BL, ya sea con o sin sensores.

1.4.3 Controlador de velocidad para motor DC 10 - 60 VDC | RS 200

El controlador electrónico RS200 (Ver Figura 1.4.3a) es un transistor servo-controlador para motores DC. La electrónica del RS200 le permite al motor moverse y frenar en una dirección (operación en 2 cuadrantes). Los efectos de frenados son conseguidos gracias a un frenado a corto circuito. El servo-controlador RS200 es un regulador de velocidad analógico que opera sobre una base actual de regulación.

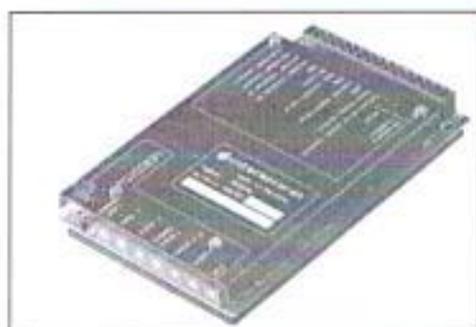


Figura 1.4.3a Controlador Electrónico RS200

La electrónica de ancho de pulso modulado tienen una alta eficiencia y son muy compactos. Para asegurarse el cumplimiento con los estándares EMC, es incluido un filtro, y el controlador tiene completa protección electromagnética. Estas características combinadas con el uso de cables blindados aseguran que se cumpla el requerimiento de EN55011 Cl. B. También hay un fusible en el circuito impreso que interrumpe el suministro de la red si existe sobrecarga.

1.4.4 Dispositivos de velocidad digital de un motor

El Governor (Ver Figura 1.4.4a) es un controlador de velocidad digital para motores mediante RC. Utiliza un algoritmo de control especial de prealimentación para regular la velocidad del motor. El JGovernor controla y regula la velocidad de giro del motor. Emplea un sistema de filtro de pulsos como salida en uno de sus Canales que limpia cualquier pico o perturbación de cualquier tipo de señal usados en los sistemas de FM, pero para un mejor resultado utiliza también un sistema PCM. Existen dos tipos de reguladores de velocidad del rotor; los que miden la velocidad del rotor en la corona principal y los que la miden en el ventilador. Los que lo hacen en la corona principal padecen la sobre rotación del rotor cada vez que este se descarga (debido al rodamiento "one-way"). Los reguladores que controlan desde el ventilador, es decir desde el eje del motor, en una situación de rotor descargado que mantienen al motor a velocidad constante. El Governor adapta continuamente la respuesta del motor a las demandas de la persona que la comanda con la finalidad de satisfacer necesidades a la hora de su funcionamiento.



Figura 1.4.4a Governor

CAPITULO 2

2. HERRAMIENTAS DEL SISTEMA

2.1 Antecedentes

El principal objetivo de nuestro proyecto es el control de velocidad de un motor DC; como se sabe el motor de corriente continua es una maquina que convierte la energía eléctrica en mecánica, principalmente mediante el movimiento rotatorio, debido a su fácil control de posición, paro y velocidad se ha convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos.

2.1.1 Modulación PWM

Una manera de controlar la velocidad del motor es por medio de la modulación por ancho de pulsos (PWM) de una señal o fuente de energía. Esta es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período (ver en la figura 2.1.1a).

Expresado matemáticamente:

$$D = \frac{\tau}{T}$$

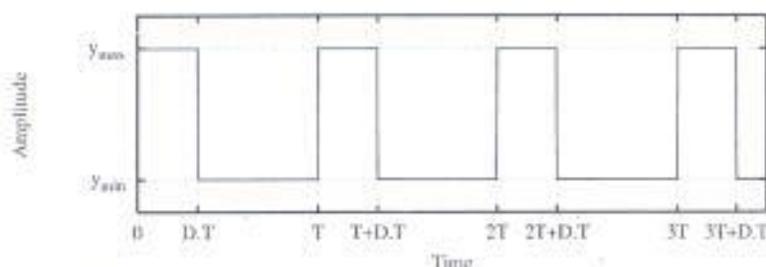


Fig. 2.1.1a Una señal de onda cuadrada mostrando ciclo de trabajo

D es el ciclo de trabajo

τ es el tiempo en que la función es positiva (ancho del pulso)

T es el periodo de la función

La construcción típica de un circuito PWM se lleva a cabo mediante un comparador con dos entradas y una salida. Una de las entradas se conecta a un oscilador de onda dientes de sierra, mientras que la otra queda disponible para la señal moduladora. En la salida la frecuencia es generalmente igual a la de la señal dientes de sierra, y el ciclo de trabajo está en función de la portadora.

La principal desventaja que presentan los circuitos PWM es la posibilidad de que haya interferencias generadas por radiofrecuencia.

Para minimizar estos tipos de errores se usará un control de sistema PID el cual mantendrá el motor a la velocidad deseada con un periodo del PID a la que se pueda ajustar con las dos correspondientes constante PID establecidos así la velocidad a la que el J_rk queremos controlar se ejecuta a través de todos sus cálculos y sea eficiente a su sistema.

2.2 Sistema de 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 (**Ver Figura 2.2**). El valor Proporcional determina la reacción del error actual. El Integral genera una corrección proporcional a la integral del error, esto nos 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.

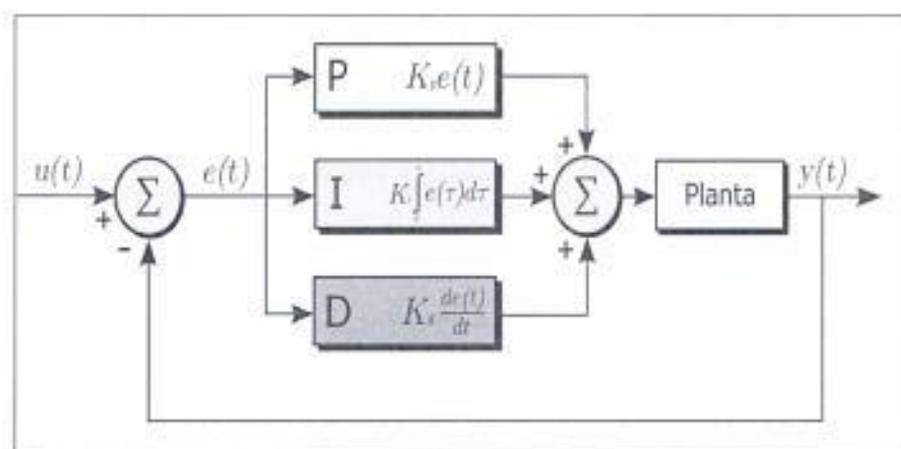


Fig. 2.2 Diagrama en bloques de un control PID

Hay que tener en cuenta que un mayor período de PID se traducirá en una más lenta evolución integral y un aumento en el derivativo, por lo que las dos correspondientes constantes PID posiblemente necesiten ser ajustado cada vez que el período de PID se cambia.

2.3 Descripción de Componentes

2.3.1 Pololu JRK 21v3 USB Motor Controller with Feedback

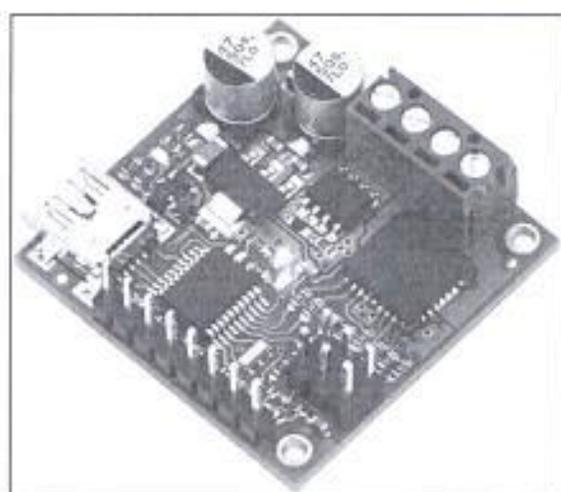


Figura 2.3.1a Pololu Jrk 21v3

El Pololu Jrk 21v3 (Ver Figura 2.3.1a) es un dispositivo altamente configurable que sirve para controlar motores de corriente continua, esta tarjeta soporta cuatro modos de interfaz con el usuario: USB, comunicación serial, voltaje analógico, y radio control (RC). El controlador se puede utilizar con retroalimentación para conformar un control a lazo cerrado de velocidad o posición, o puede ser utilizado sin realimentación como un control de velocidad de lazo abierto. La corriente de salida nominal es de aproximadamente 3 Amperios utilizando el rango de operación que oscila 5 - 28 V. Mediante el programa Jrk Configuration Utility permite al usuario una fácil calibración y configuración a través del puerto USB.

2.3.1.1 Aspectos destacados acerca del Pololu Jrk 21v3 USB Motor Controller with

Feedback

- Control bidireccional de un motor CC (Corriente Continua).
- Rango de voltaje de operación de 5 V a 28 V.
- Corriente máxima nominal de salida de 3 A (5 A pico).
- Cuatro opciones de comunicación o control:
 1. Interface USB para la conexión directa con la PC.
 2. Interface TTL serial asincrónica Full-dúplex para una conexión directa con micro controladores u otros dispositivos.
 3. Interface de radio control (RC) para una conexión directa con un receptor RC o con un RC servo controlador.
 4. Interface de voltaje analógico 0-5 V para una conexión directa con potenciómetros o un joystick analógico
- Dos opciones para conformar el control de lazo cerrado.
 - 2 Voltaje analógico de 0-5 V.
 - 3 Señal digital de entrada de hasta 2 MHz de frecuencia con 1 ms de periodo en el PID.
- Los parámetros configurables incluyen:
 1. El periodo y constantes del control PID Corriente máxima
 2. Ciclo de trabajo máximo
 3. Máxima aceleración
 4. Opción de detección de errores CRC

La Familia JRK soporta una variedad de interfaces, incluyendo la configuración del microcontrolador para la comunicación serial UART, esta configuración se la utilizará para la interacción entre el joystick inalámbrico y el JRK por medio de los pines TX y RX (Ver figura 2.3.1.1a), los cuales se conectarán a un microcontrolador que será la interface entre el receptor inalámbrico y el JRK.

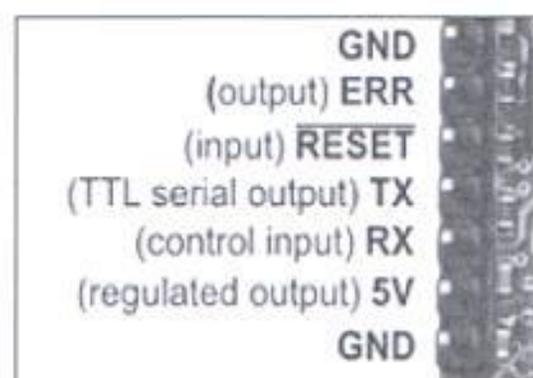


Fig. 2.3.1.1a Pines de Comunicación Serial UART

El controlador JRK 21v3 cuenta con un sistema PID con retroalimentación el cual está diseñado para ser parte de un sistema de control en el que la salida (en nuestro caso la velocidad del motor) se ajusta constantemente para que coincida con un valor de destino especificado. Para lograr esto, se mide constantemente con un tacómetro (frecuencia) el estado del sistema y se retroalimenta por el pin FB, llevando a cabo una rápida implementación de un lazo cerrado para nuestro sistema de control.

El tratamiento de la información realizada por el JRK se describe en el siguiente diagrama (Ver Figura 2.3.1.2)

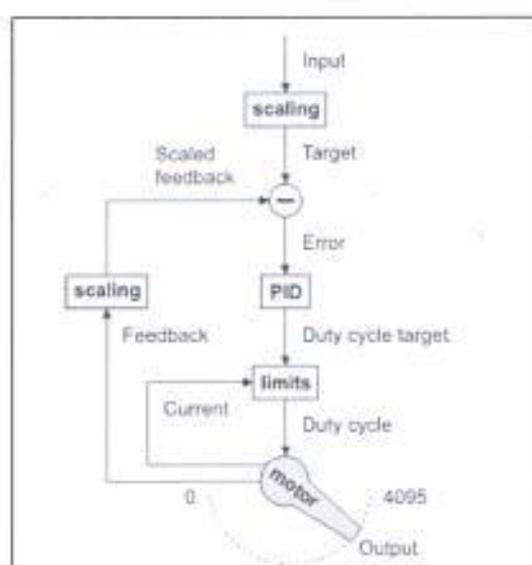


Fig. 2.3.1.2 Diagrama de un sistema típico de retroalimentación

2.3.1.2.1 Señales de entrada y salida del JRK

En este diagrama, cada flecha representa un número específico medido o calculado por el JRK, y las cajas azules representan los cálculos internos que se producen cada vez por cada periodo de PID. El periodo de PID se puede establecer en incrementos de 1 ms., el JRK utiliza las siguientes medidas para determinar la salida:

2.3.1.2.2 Input

Se mide como un valor entre 0 y 4095. En el modo de entrada analógica de tensión, lo que representa un nivel de voltaje de 0 a 5 V. En el modo RC, el número es un ancho de pulso en unidades de 3.2 ms. La entrada se ajusta de acuerdo a los parámetros de entrada de escala para determinar el destino, también un valor de 0 a 4095.

2.3.1.2.3 Scaled Feedback

Se mide como un valor entre 0 y 4095. En el modo analógico de retorno del voltaje, lo que representa un nivel de voltaje de 0 a 5 V.

En el modo digital, la frecuencia es una representación de la velocidad de salida. El JRK utiliza este valor para calcular la escala de retroalimentación, que es una representación de la salida del sistema de control completo. Una escala de retroalimentación de 0 debe representar la posición mínima del sistema, y 4095 deben representar la posición máxima.

2.3.1.2.4 Current

La corriente que pasa por el motor es medida como un número, desde 0 hasta 255. Un valor de calibración relaciona esta corriente con la corriente actual en amperios.

2.3.1.2.5 Error

El Error se calcula como la diferencia de la escala de la retroalimentación y el objetivo ($\text{error} = \text{comentarios escala} - \text{destino}$). La Opción de detección de errores CRC que elimina los errores de comunicación causados por defectos de ruido o de software es un parámetro establecido fijo con el que siempre va estar en funcionamiento dentro de su sistema

2.3.2 10:1 Micro Metal Gearmotor



Figura 2.3.2a Micro Metal Gearmotor

Este motor (Ver Figura 2.3.2a) pequeños y de alta calidad destinados a ser utilizados a 6 V, aunque en general este tipo de motores pueden funcionar a tensiones por encima y por debajo de este voltaje nominal, por lo que deben operar con comodidad en el rango 3-9 V (la rotación puede comenzar con tensiones tan bajas como 0,5 V). Los voltajes más bajos podrían no ser práctico y las tensiones más elevadas podrían empezar a afectar negativamente la vida útil del motor. Los micros de metal Gearmetal están disponibles en una amplia gama de relaciones de transmisión-de 5:1 hasta 1000:1 y ofrecen una elección entre dos motores diferentes: de alta potencia (HP) y estándar. Todos los motores con engranajes del GearMetal tiene las dimensiones física por lo que una versión puede ser fácilmente cambiado por otro si el diseño cambia.

Algunas versiones de los GearMotor se venden con un eje de salida adicional de 1 mm de diámetro que sobresale de la parte trasera del motor. Este eje trasero de 4,5 mm de largo gira a la misma velocidad que la entrada a la caja de cambios y ofrece una manera de agregar un codificador a la medida para proporcionar la velocidad del motor o de posición.

2.3.2.1 Aspectos destacados del 10:1 MICRO METAL GEARMOTOR

Tamaño:	24 x 10 x 12 mm
Peso:	0,34 oz
Diámetro del eje:	3 mm

Tabla 2.3.2.1a Dimensiones

Relación de engranaje:	10:1
Velocidad normal de funcionamiento @ 6V:	1300 rpm
Corriente normal de funcionamiento @ 6V:	40 mA
Corriente de motor bloqueado @ 6V:	360 mA
Torque de rotor bloqueado 6V:	2 oz·in

Tabla 2.3.2.1b Especificaciones Generales

2.3.3 RC Snap Circuits Rover

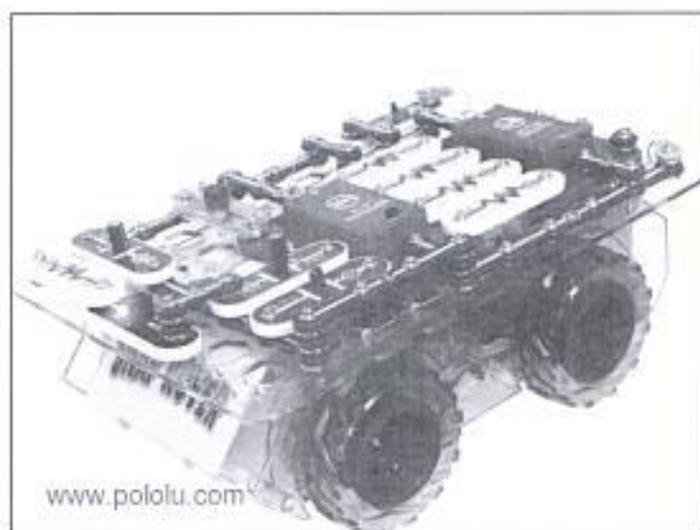


Figura 2.3.3a RC Snap Circuits Rover

El RC Snap Circuits Rover (Ver Figura 2.3.3a) es un complemento de radio-control conjunto sobre ruedas. Las cuatro ruedas del chasis deportivo 128:1 con dos trenes de engranajes, una linterna LED, un cuerno, y un transmisor (RC) de radio-control. Los motores izquierdo y derecho pueden controlarse de forma independiente, y el chasis claramente le permite ver los engranajes en la acción.

El RC Snap Circuits Rover viene con un manual impreso a todo color y con código de color, piezas de complemento juntos que hacen que la construcción del circuito sea fácil. El manual tiene 23 proyectos del circuito que cuentan con un esquema de color del circuito, un objetivo educativo, mostrándonos paso a paso las instrucciones de montaje, y una descripción de lo que el circuito montado debe hacer. No se requieren herramientas para el montaje del circuito. El robot necesita seis pilas AA y el transmisor necesita una batería de 9 voltios para funcionar.

2.3.4 Joystick

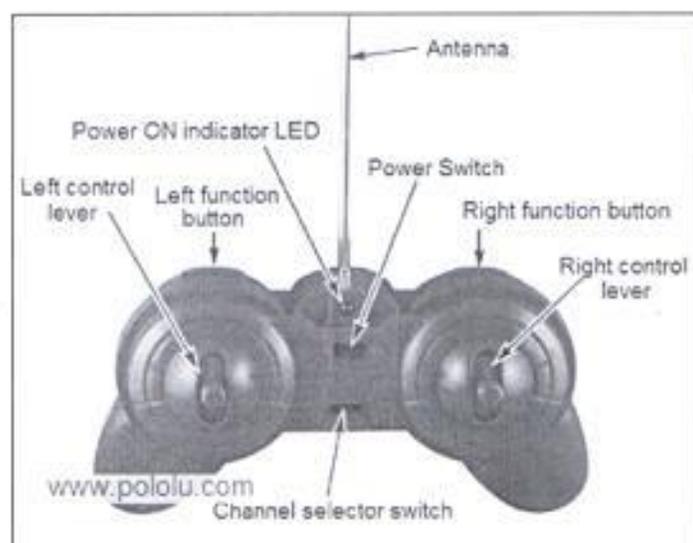


Fig. 2.3.4a Joystick

Los sistemas del RC Snap Circuits Rover (Ver Fig. 2.3.4a) Incluye un transmisor de RC por la cual consta de dos palancas y dos botones que nos permitirán controlar los motores e incluso otras partes del circuito mediante el movimiento de sus controles ejecutándose en la misma área sin interferir uno con el otro. Se suele diferenciar entre joysticks digitales (que leen cuatro interruptores encendido/apagado en cruceta situada en la base más sus combinaciones y los botones de acción) y joysticks analógicos (que usan potenciómetros para leer continuamente el estado de cada eje, y además de botones de acción pueden incorporar controles deslizantes), siendo estos últimos más precisos.

2.3.5 ATtiny 2313

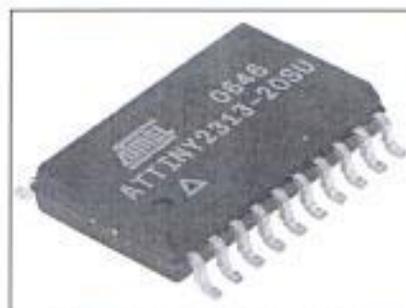


Figura 2.3.5a ATtiny2113

El ATtiny2313(Ver Fig. 2.3.5a) proporciona las siguientes características: 2K bytes de Entrada-Sistema de flash programable, 128 bytes EEPROM, 128 bytes de SRAM, 18 líneas I / O de propósito general, 32 registros de trabajo de propósito general de trabajo, un solo cable de interfaz para la comunicación con el chip, dos flexibles temporizador / Contadores con modos de comparación, interrupciones internas y externas, un interfaz serial USART programable, Un interfaz serial universal con la condición del arranque del detector, un temporizador de vigilancia programable con oscilador interno, y tres potencias seleccionable por software como modo de ahorro. El modo de apagado de energía guarda el contenido del registro, pero se congela el oscilador desactivando todas las funciones del chip hasta la siguiente interrupción.

Una de las ventajas del uso de los microcontroladores es su capacidad de comunicación serial estándar entre diferentes marcas. El Microcontrolador que hemos escogido se usara para la interacción entre el receptor inalámbrico y el JRK mediante la comunicación serial UART. La comunicación entre estos dos controladores será hecha por medio de sus respectivos puertos de comunicación TX y RX.

2.3.6 ENCODER PARA NEUMATICO POLOLU 42X19MM



Figura 2.3.6b Encoder Pololu 42X19 mm

Este Encoder de cuadratura (**Ver Figura 2.3.6a**) está diseñado para trabajar con micro motores de corriente continua usando dos sensores infrarrojos que están adentro del eje del neumático, los cuales miden el movimiento de los doce dientes a lo largo del aro de la rueda.

Los dos sensores están separados para proporcionar formas de onda aproximadamente con un desfase de 90 grados entre ellas, permitiendo determinar la dirección de rotación y proporcionando cuatro conteos por diente para una resolución de 48 conteos por cada rotación de la rueda.

Cada señal analógica del sensor es alimentada a un comparador con histéresis para proporcionar salidas digitales libres de saltos. El encoder está calibrado para un rango de operación de 4,5 V a 5,5 V, pero puede ser calibrado para operar a 3,3 V.

En el estudio de los Encoder sabemos que es un codificador rotatorio, también llamado codificador del eje o generador de pulsos, suele ser un dispositivo electromecánico usado para convertir la posición angular de un eje a un código digital, lo que lo convierte en una clase de transductor (**Ver Figura 2.3.6b**). Estos dispositivos se utilizan en robótica, en lentes fotográficas de última generación, en dispositivos de entrada de

ordenador (tales como el ratón y el Trackball), y en plataformas de radar rotatorias. Hay dos tipos principales: absoluto e incremental (relativo).

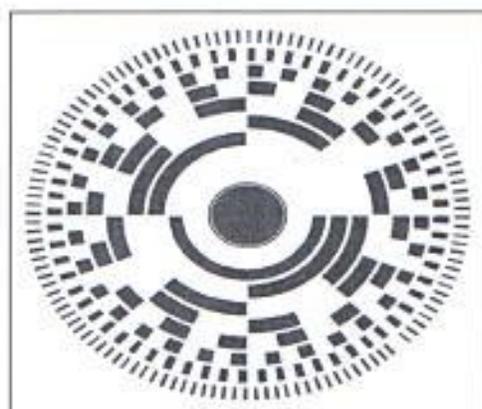


Figura 2.3.6b Encoder (Codificador de eje o generador de pulsos)

2.3.6.1 Aspectos destacados del Encoder para neumático Pololu 42x19mm.

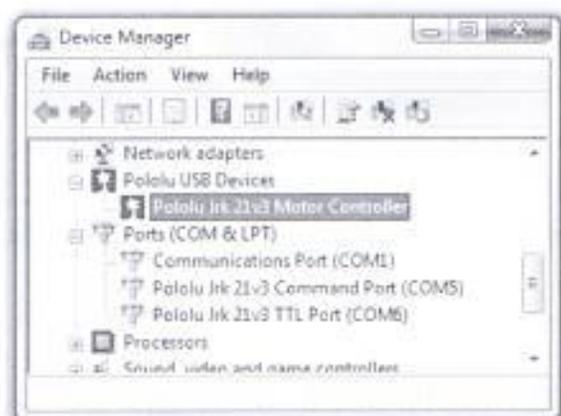
- Rango de operación de voltaje: 4,5 V a 5,5 V (el encoder puede ser modificado para trabajar a tensiones inferiores).
- Dos salidas digitales (cuadratura).
- 14 mA de consumo a 5,0 V
- 48 conteos por vuelta.
- Tamaño pequeño: encaja entre el motor y el chasis.
- Peso: 1,6 g (0,06 oz)

2.3.7 Herramientas de Software

El proyecto utilizó tres tipos de software, el primero tiene como fin la programación del ATtiny 2313 para la comunicación USART con nuestro dispositivo, el segundo tiene como objetivo la simulación del micro para nuestro proyecto para verificar los datos que se envían y por último el software del POLOLU JRK 21v3 para las configuraciones de los parámetros de seteo con la finalidad de eliminar los errores de alimentación.

2.3.7.1 Jrk Configuration Utility

El software que utilizamos como soporte para la realización de este proyecto es el **Jrk Configuration Utility**, el cual se encuentra de una forma gratuita en la dirección electrónica de la compañía Pololu con sus respectivos drivers como podemos observar en su instalación (Ver grafica 2.3.7.1a). Este programa nos ofrece una interfaz grafica con el usuario muy útil y sencilla al momento de configurar los parámetros deseados para la tarjeta Jrk 21v3 USB Controller with Feedback.



Gráfica 2.3.7.1a Windows Vista device manager showing
The Pololu Jrk 21v3 Motor Controller

A continuación se describirá cada pestaña que contiene el programa Jrk Configuration Utility:

2.3.7.1.1 Pestaña de INPUT (Ver Figura 2.3.7.1.1a)



Figura 2.3.7.1.1a Pestaña Input

La pestaña de INPUT del programa Jrk Configuration Utility contiene la configuración de cómo el sistema de realimentación (que consiste en el Jrk, un motor y un sensor de retroalimentación) es externamente controlado y vigilado. Lo más importante, hay tres modos de entrada:

Serial: Esta entrada nos sirve para configuración de puertos virtuales COM o para entradas TTL-seriales.

Analog Voltage: Se utiliza cuando la entrada es una fuente analógica, como un potenciómetro. El nivel bajo significa 0 [V] y nivel alto significa 5 [V].

Pulse width: Se utiliza cuando el sistema se controla por el ancho de pulsos digitales, como los de salida por un radio-control (RC) del receptor, medidas en la línea de RX del Jrk.

Para nuestro proyecto, la entrada será UART baud rate porque nuestro medio de comunicación con el dispositivo se realizará por medio de UART.

2.3.7.1.2 Pestaña de FEEDBACK (Ver Figura 2.3.7.1.2a)

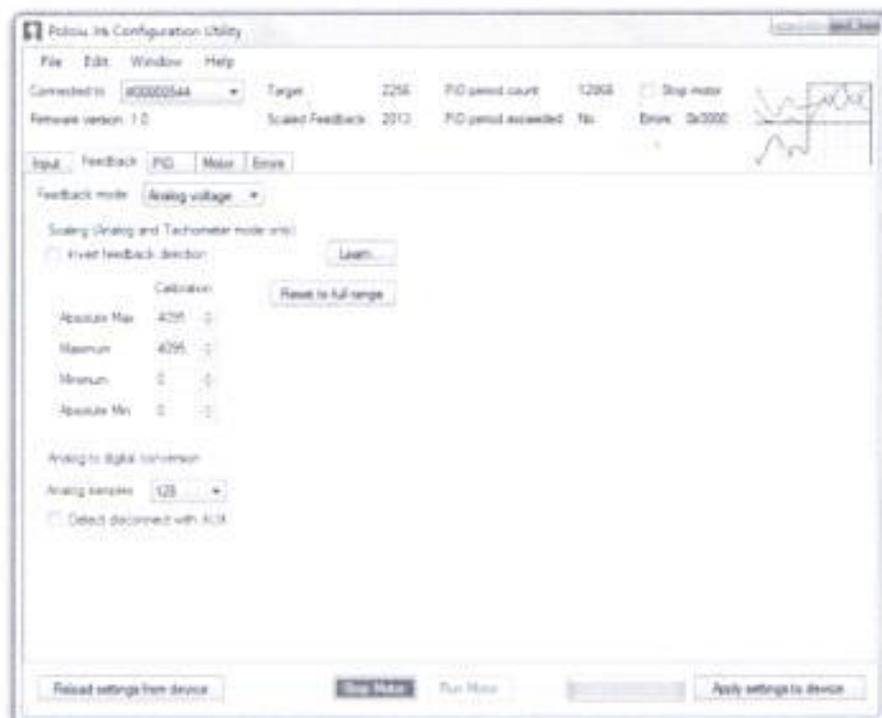


Figura 2.3.7.1.2a Pestaña Feedback.

La pestaña de Feedback de este software controla las mediciones de la salida del sistema de control. Están disponibles tres modos de realimentación:

None: Con esta opción activada, la realimentación y el control PID quedan deshabilitados. Es decir que el sistema trabajará con un control a lazo abierto.

Analog Voltage: Se utiliza cuando se conecta una fuente analógica, como un potenciómetro. El nivel bajo significa 0 [V] y nivel alto significa 5 [V].

Frecuency (digital): Se utiliza se utiliza con los dispositivos de medición de la velocidad que generan pulsos a un ritmo proporcional a la velocidad del eje de salida, como un

tacómetro. En nuestro caso utilizaremos un encoder, el cual nos servirá como realimentación para cerrar el lazo de control.

2.3.7.1.3 Pestaña PID (Ver figura 2.3.7.1.3a)



Figura 2.3.7.1.3a Pestaña PID

En estas opciones de PID encontramos que podemos configurar los valores que componen este sistema de control. 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.

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, en nuestro caso será

el motor y el encoder, integrando esta desviación en el tiempo y sumándola a la acción proporcional. Y la parte **derivativa** consiste en el error actual menos el error anterior.

El período de PID se puede ajustar aquí, lo que establece la velocidad a la que el Jrk se ejecuta a través de todos sus cálculos. Hay que tener en cuenta que un mayor período de PID se traducirá en una más lenta evolución integral y un aumento en el derivativo, por lo que las dos correspondientes constantes PID posiblemente necesiten ser ajustado cada vez que el período de PID se cambia.

2.3.7.1.4 Pestaña de MOTOR. (Ver Figura 2.3.7.1.4a)



Figura 2.3.7.1.4a Pestaña Motor

En esta pestaña podemos configurar los parámetros principales del motor, como el Máximo ciclo de trabajo, la máxima aceleración. El valor 600 significa el 100% de trabajo y a la vez el valor máximo que alcanza el motor.

También se puede configurar la frecuencia de la señal PWM que gobierna al motor. Utilizando una frecuencia de 20 KHz es de ultrasonidos y por lo tanto puede eliminar sonidos PWM incluidos el zumbido de motor, lo que hace que esta frecuencia conveniente para aplicaciones típicas. Para nuestro proyecto, estos parámetros nos servirán para regularizar la forma de trabajo del micro motor DC.

2.3.7.1.5 Pestaña de ERROR. (Ver Figura 2.3.7.1.5a)

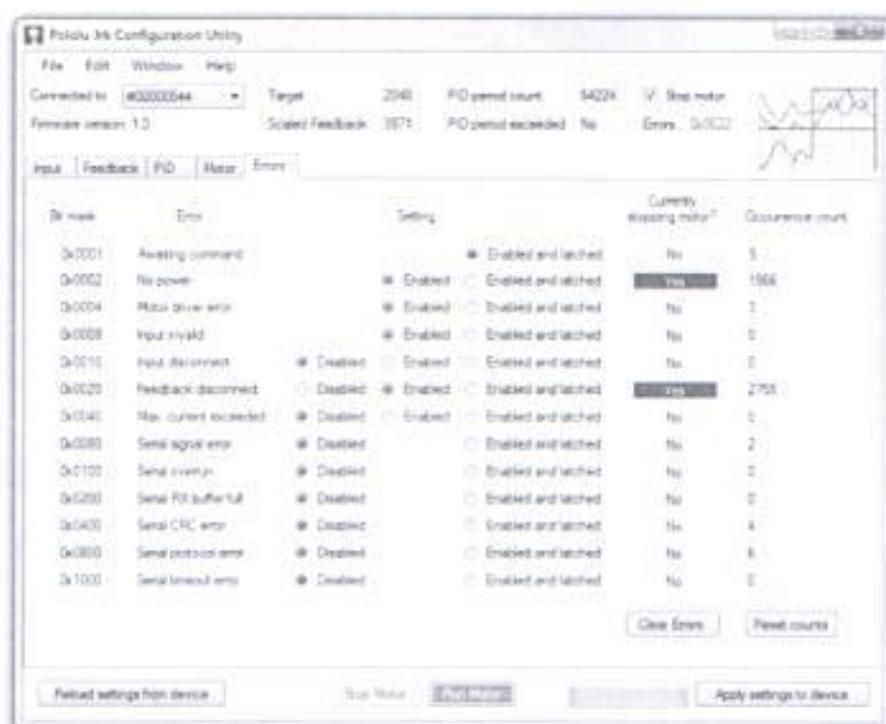


Figura 2.3.7.1.5a Pestaña Error

La respuesta del Jrk a los diferentes errores puede ser configurado. Cada error tiene hasta tres opciones disponibles.

- **Disabled:** El Jrk ignora cualquier error. Pero igual se puede determinar si el error se produce revisando la opción "Occurrence count".
- **Enabled:** Cuando esta opción se activa, el Jrk apague el motor. Cuando el error deja suceder, el motor puede arrancar.

- **Enabled and Latched:** Cuando este error ocurre, el Jrk se apague el motor y ajuste la espera de Comando de error de bit. El jrk no conducirá de nuevo el motor hasta que reciba una confirmación que el error se haya desaparecido.

2.3.7.1.6 The Plots Window. (Ver Figura 2.3.7.1.6a)

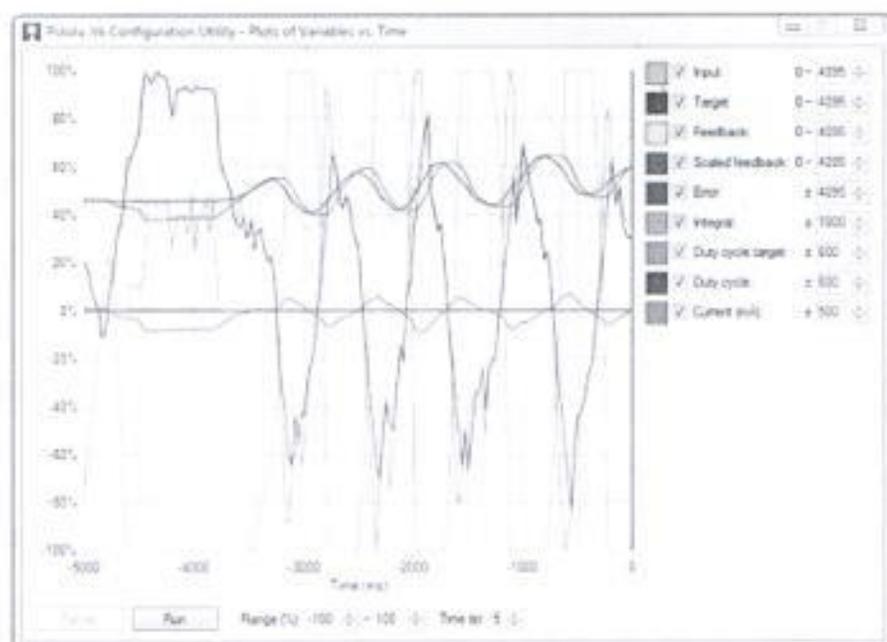


Figura 2.3.7.1.6a The Plots Windows

La opción Plots Windows del programa Jrk Configuration Utility muestra a tiempo real datos acerca del Jrk. Cada variable puede ser independiente a escala en un rango útil. Por ejemplo, el error puede ser -4095 hasta 4095, pero para los sistemas de retroalimentación bien afinado, por lo general tendrá un valor mucho menor, por tanto, establecemos el rango de ± 100 para su eficiencia.

El gráfico inicial muestra los datos de los últimos 5 segundos, con los valores más recientes sobre el derecho y los valores mayores a la izquierda. La escala de tiempo de la trama se puede reducir con el tiempo (s) de ajuste en la parte inferior de la ventana. El color de cada variable en el grafico se puede seleccionar haciendo uso de la opción del menú que está en la parte derecha de la ventana.

2.3.7.2 Proteus 7.7 (Ver Figura 2.3.7.2a)

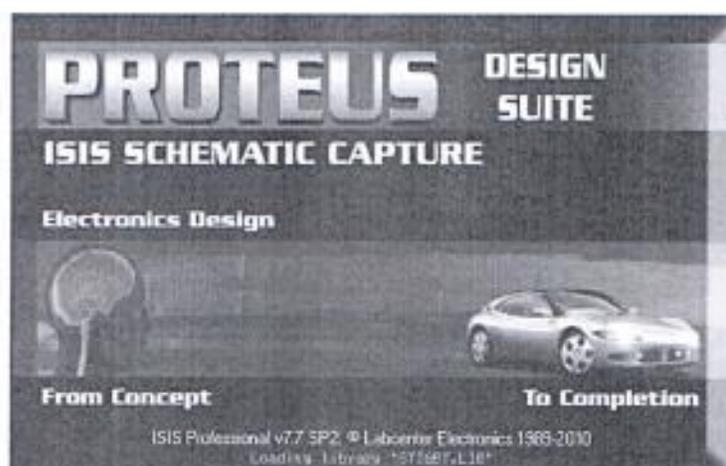


Figura 2.3.7.2a

Es un paquete de software para el diseño de circuitos electrónicos que incluye captura de los esquemas, simulación analógica y digital combinada, además posee una herramienta ARES que se utiliza para el diseño de circuitos impresos. Proteus es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción. El paquete está compuesto por dos programas: ISIS, para la captura y simulación de circuitos; y ARES, para el diseño de PCB's. También permite simular y depurar el funcionamiento de todo el sistema ejecutando el software paso a paso, insertando puntos de ruptura (breakpoints, que también pueden ser generados por el hardware), mirando el contenido de registros y posiciones de memoria, etc. y comprobando si la respuesta del hardware es la correcta. También se simulan herramientas electrónicas, como osciloscopios, analizadores lógicos, voltímetros, etc.

2.3.7.3 AVR Studio 4 (Ver Figura 2.3.7.3a)



Figura 2.3.7.3a

El AVR Studio 4 es un entorno de desarrollo integrado para la depuración de AVR software. El estudio del AVR permite la simulación de chips y la emulación en circuito para la familia de microcontroladores AVR. La interfaz de usuario está especialmente diseñado para ser fácil de usar y dar información general a la información completa tanto para la simulación y emulación.

Esta herramienta de desarrollo que nos permite realizar proyectos para microcontroladores. Proporciona una solución fácil para aplicaciones de sistemas embebidos, sin comprometer el rendimiento o el control, además desarrolla rápidamente y despliega aplicaciones complejas.

Originalmente concebido como una herramienta fácil de usar, BASIC se generalizó en microcomputadoras en la década de 1980, y sigue siendo popular hoy en día. BASIC permite un desarrollo mucho más rápido y fácil de las solicitudes de consentimiento fundamentado previo en comparación con el lenguaje del microchip Ensamblador MPASM. Al escribir el código de MCU, con frecuencia los programadores hacen frente a las mismas funciones, tales como la comunicación serial, la impresión en la pantalla

LCD, la generación de señales PWM, etc. Con el fin de facilitar la programación, se basa en una serie de rutinas built-in, destinados para resolver estos problemas.

En cuanto a la ejecución y el tamaño del programa en cuestión, MPASM tiene una pequeña ventaja en relación con BASIC. Por esta razón hay una opción de combinar BASIC y el código ensamblador. El ensamblador se utiliza comúnmente para las partes del programa en el que el tiempo de ejecución es crítico o que los comandos se ejecuten gran número de veces. Los microcontroladores modernos, como los PIC, ejecutan instrucciones en un solo ciclo de reloj. Si el reloj del microcontrolador es 4MHz, una instrucción en lenguaje ensamblador requiere $250\text{ns} \times 4 = 1\mu\text{s}$. Como cada comando básico es técnicamente una secuencia de instrucciones en ensamblador, el tiempo exacto necesario para su ejecución se puede calcular simplemente sumando los tiempos de ejecución de las instrucciones.

CAPITULO 3

3. DISEÑO DE LA SOLUCIÓN

3.1 Diseño Preliminar

Para el diseño de este proyecto se seleccionó la tarjeta controladora Pololu Jrk 21v3 con Realimentación (Feedback), que utiliza la vía USB como medio de comunicación con la PC con la finalidad de setear los errores de configuración y selección de su respectivo medio de comunicación en este caso el USART baud rate. Para cerrar el lazo cerrado de control utilizamos un Encoder de Cuadratura Pololu de 42x19 mm, extrayendo la señal que nos brinda el Encoder y conectándola en el Jrk 21v3 para obtener un sistema con realimentación.

3.2 Pruebas Preliminares

Para comprobar el funcionamiento correcto de la tarjeta Pololu Jrk 21v3 con Realimentación (Feedback) nosotros realizamos una sencilla configuración que se encuentra en el Manual de Usuario que viene incorporado con esta tarjeta. Al terminar esta prueba, se logro observar el trabajo adecuado de la tarjeta conectada con el Motor DC y el control que tiene el usuario a lazo abierto sobre la velocidad del motor. Después, al incorporar el Encoder de cuadratura a nuestro sistema, tuvimos que alimentarlo con 5 [V] y después conectar el Output A del Encoder al Pin de Realimentación (Fb) de la tarjeta controladora, para así poder tener un control a lazo cerrado.

3.3 Comunicación entre PC y Usuario

Inicialmente, necesitamos ingresar la dirección de correo electrónico de la pagina donde vamos a descargar el software, www.pololu.com. (Ver Figura 3.3a)

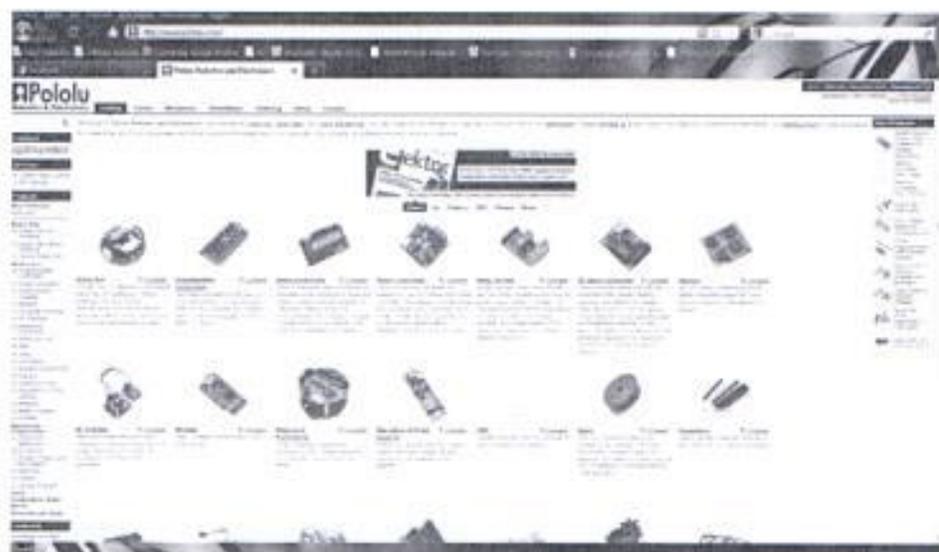


Figura 3.3a Pagina de Descarga

Luego se inicia la búsqueda del Ítem Pololu Jrk 21v3 USB Motor Controller with Feedback. (Ver Figura 3.3b).



Figura 3.3b Búsqueda de Ítem Jrk 21v3 USB Motor Controller with Feedback

Una vez seleccionado el ítem encontraremos su precio, acompañado con diferentes pestañas como Description, Specification, Pictures, Resources y FAQs. (Ver Figura 3.3c)



Figura 3.3c Pololu Jrk 21v3 USB Motor Controller with Feedback

Se selecciona la pestaña Resources y aparecerá el link para la descarga del Datasheet del Ítem que vamos a utilizar, a continuación damos click en `jrk_motor_controller.pdf`. (Ver Figura 3.3d)



Figura 3.3d Link de Descarga PDF de Datasheet de Ítem Jrk 21v3.

Una vez leída toda la documentación acerca de nuestro controlador de velocidad nos dirigimos al capítulo 3. En este Capítulo podemos observar los requisitos que se necesitan para instalar la aplicación que se usará para la comunicación entre la PC y la tarjeta Jrk 21v3 por medio de USB. Si se tiene como sistema operativo Windows XP se debe por medio del 1er link descargar el Service Pack 3 antes de instalar los drivers de lo contrario no funcionará. Una vez descargado, se procede a descargar los Drivers y Software de configuración por medio del Segundo Link. (Ver Figura 3.3e).

3. Configuring the Motor Controller

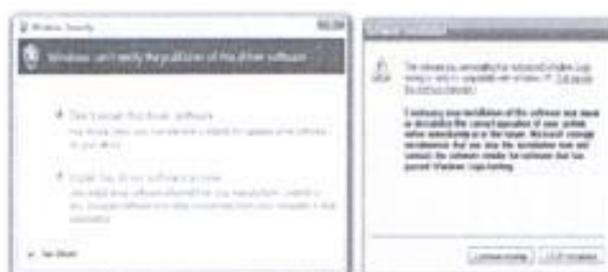
3.1. Installing Windows Drivers and the Configuration Utility



If you use Windows XP, you will need to have **Service Pack 3** (<http://www.microsoft.com/downloads/details.aspx?FamilyId=6C9DAD-BC34-48E2-8D96-4E8B479F71D8>) installed before installing the drivers for the mk. See below for details.

Before you connect your Pololu mk USB motor controller to a computer running Microsoft Windows, you must install its driver:

1. Download the mk driver and configuration software (http://www.pololu.com/files/downloads/mk_windows_driver.zip) (519Kb .zip).
2. Open the ZIP archive and run *setup.exe*. If the installer fails, you may have to extract all the files to a temporary directory, right-click *setup.exe*, and select "Run as administrator". The installer will guide you through the steps required to install the Pololu mk Configuration Utility and the mk driver on your computer.
3. During the installation, Windows will warn you that the driver has not been tested by Microsoft and recommend that you stop the installation. Click "Install this driver software anyway" (Windows 7 and Vista) or "Continue Anyway" (Windows XP).



4. After the installation is finished, your start menu will have a shortcut to the Mk Configuration Utility (in the Pololu folder). This is a Windows application that allows you to change all of the settings of your motor controller, as well as see real-time information about its state.

Windows 7 and Vista users: Your computer should now automatically install the necessary drivers when you connect a mk. No further action from you is required.

Figura 3.3e Link de Descarga Drivers y Software Configurable.

A continuación se abrirá un cuadro especificando que el archivo que se va a descargar es uno comprimido WINRAR de nombre `jrk_windows_091218.zip`. (Ver Figura 3.3f).

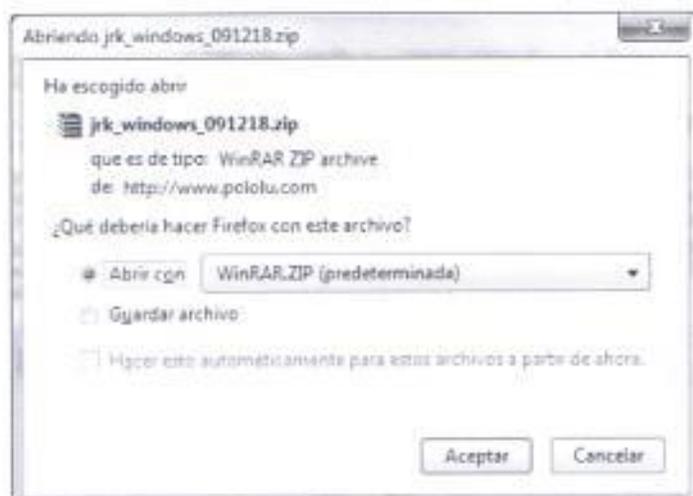


Figura 3.3f Descarga `jrk_windows_091218.zip`

Se abrirá un cuadro del programa WINRAR mostrando la carpeta con el archivo ejecutable; se da click a esta carpeta y aparecerá el archivo deseado Setup. (Ver Figuras 3.3g y 3.3h)

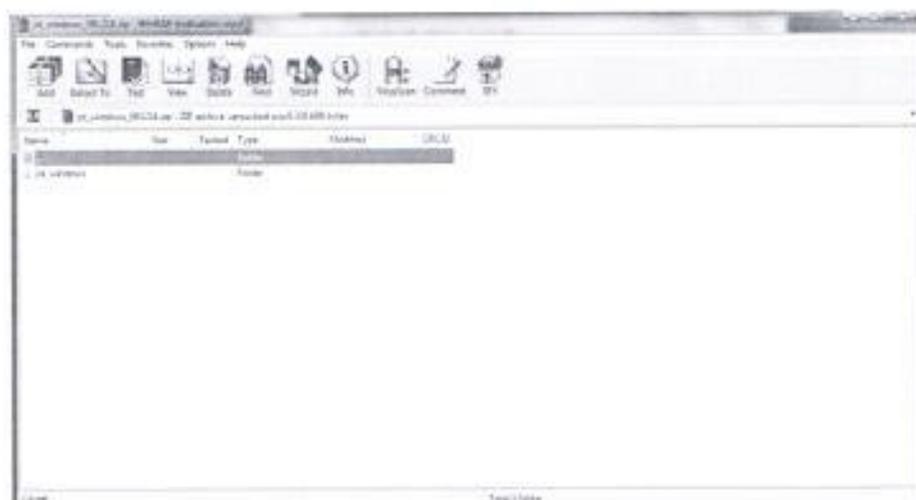


Figura 3.3g Cuadro WINRAR con carpeta de archivo ejecutable

Se selecciona la carpeta donde irá la aplicación, luego NEXT.(Ver Figura 3.3j)

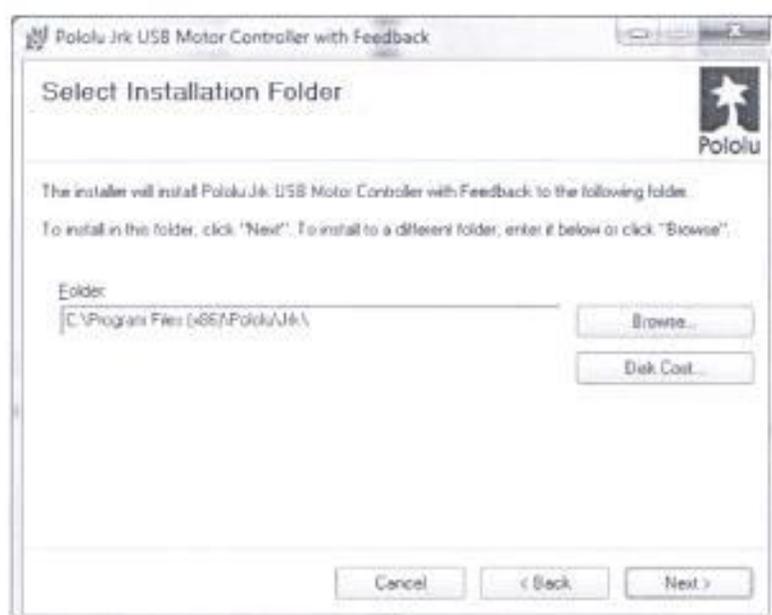


Figura 3.3j Selección de Carpeta donde irá aplicación

Confirmación de Instalación, luego NEXT. (Ver Figura 3.3k)



Figura 3.3k Confirmación de Instalación.

Instalando. (Ver Figura 3.3l)



Figura 3.3l Instalando.

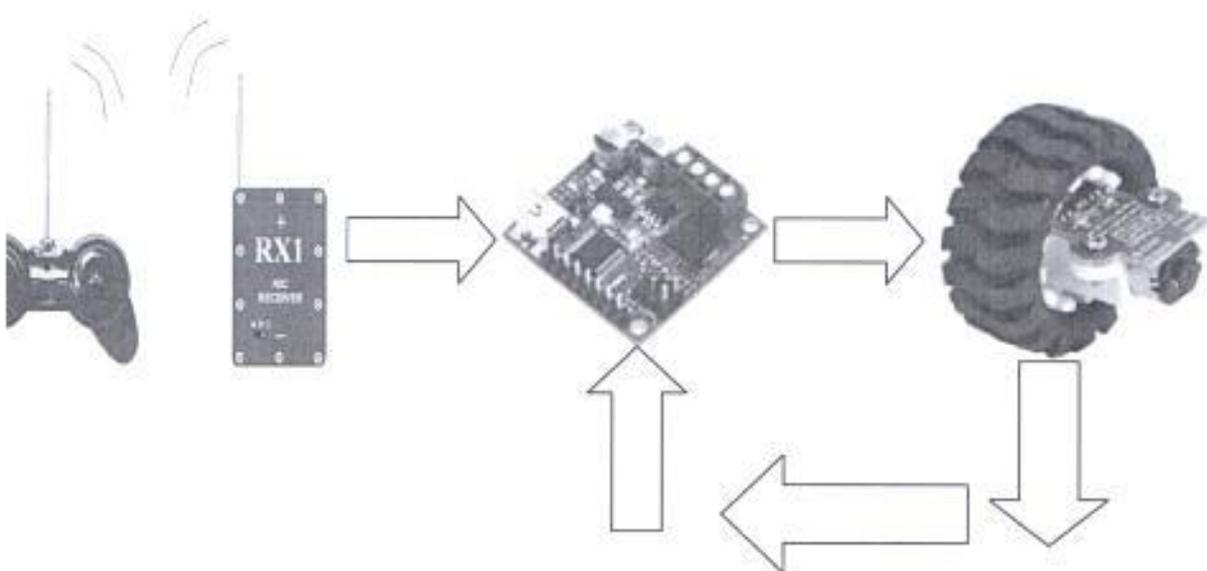
Una vez acabado el proceso el instalador mostrara el mensaje de instalación exitosa, presiona CLOSE para salir. (Ver Figura 3.3m)



Figura 3.3m Instalación Exitosa.

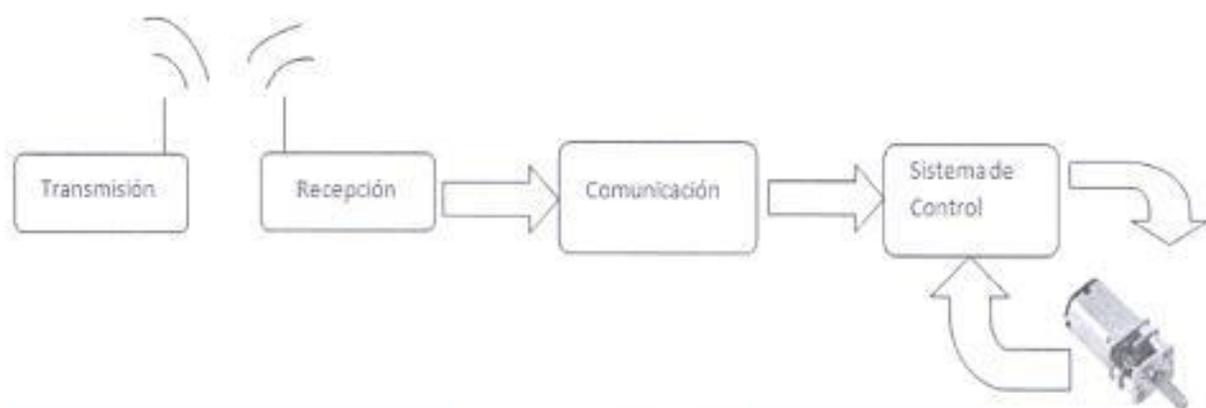
3.4 Diseño del software y hardware

3.4.1 Diagrama de Bloques General



3.4.2 Diagrama de Bloques detallado

Nuestro diseño de proyecto se basa en 4 bloques principales, transmisión, recepción, comunicación y sistema de control.



3.4.2.1 Bloques:

3.4.2.1.1 Transición

La Transmisión de datos se hace por medio de comunicación de radio frecuencia, gracias a esto se realizara un enlace inalámbrico con el cual se podrá comandar de forma remota la velocidad del motor, para esto se utilizar el Joystick RC Snap Circuits Rover que ya viene con 3 tipos de canales predeterminados. Este transmisor codificara las entradas y las enviara en un carrier de 27 MHz (Ver Fig. 3.4.2.1.1a).

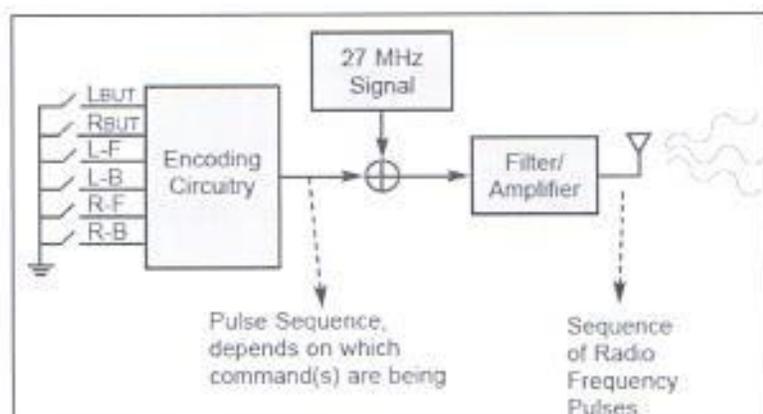


Fig. 3.4.2.1.1a Diagrama de Bloque del Transmisor

Se hará uso de tres botones para el manejo de la velocidad del motor, el primero aumentará la velocidad del motor, el segundo disminuirá su velocidad y el tercero hará que el motor se detenga de forma paulatina.

3.4.2.1.2 Recepción

Gracias a la variedad de dispositivos de transmisión y recepción RF se tiene un paquete completo de comunicación inalámbrica en el cual el bloque de recepción ya viene compactada para la obtención de señales RF con el mismo número de canales de comunicación de igual frecuencia que el transmisor

El receptor es un módulo complejo que contiene un circuito receptor de radio, un circuito decodificador integrado de radio especializados, y otros componentes de apoyo. Incluye resistencias, condensadores, inductores, transistores y que siempre se necesitan entre sí (Ver Fig.3.4.2.1.2a). un receptor de radio consiste en un circuito eléctrico, diseñado de tal forma que permite filtrar o separar una corriente pequeñísima, que se genera en la antena, por efecto de las ondas electromagnéticas (el fenómeno se llama inducción electromagnética) que llegan por el aire normalmente (aunque viajan por cualquier medio, inclusive el vacío) y luego amplificarla selectivamente, miles de veces, para enviarla hacia un elemento con un electroimán, que es el altavoz (o parlante), donde se transforman las ondas eléctricas en sonido.

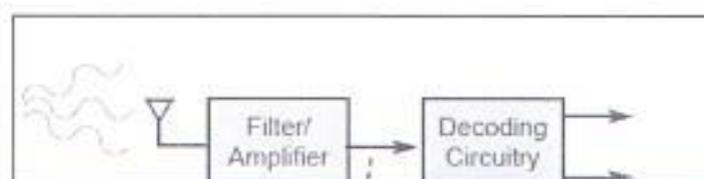


Fig. 3.4.2.1.2a Diagrama de Bloque del Receptor

Las entradas y salidas se pueden ver en la siguiente figura compactada del receptor (Ver Fig. 3.4.2.1.2b).

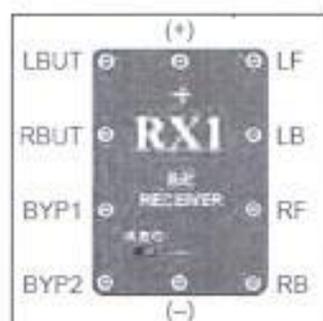


Fig. 3.4.2.1.2b R/C Receiver

Las salidas LF, LB, y RF se enviarán hacia el Microcontrolador para que estas las maneje según el requerimiento que necesitemos.

3.4.2.1.3 Comunicación

Este bloque está conformado por el Microcontrolador ATtiny 2313 (Ver Fig. 3.4.2.1.3a) el cual capta las señales de salida del receptor y las envía al sistema de control por medio de comunicación Serial UART. Esta comunicación se lleva a cabo debido a que el sistema de control PID solo necesita una señal de entrada. Estas tres señales se programarán según las

especificaciones del sistema de control PID, el cual propone un protocolo para el manejo de velocidad del motor.

Fig. 3.4.2.1.3a ATtyni 2313

3.4.2.1.4 Sistema de Control PID

Cuando se hace necesario conocer el comportamiento de un sistema en unas determinadas condiciones y ante unas determinadas entradas, se puede recurrir a la experimentación sobre dicho sistema y a la observación de sus salidas.

El controlador que vamos a usar el PID está incluido dentro de la categoría de controladores de lazo cerrado (Ver Fig. 3.4.2.1.4a), pues resulta imposible establecer una relación fiable entre la señal aplicada y la posición final, es decir, los motores de corriente continua no pueden funcionar en lazo abierto.

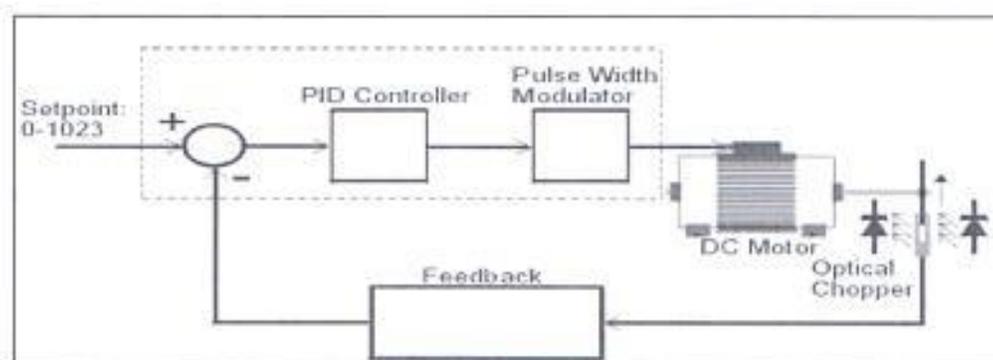


Fig. 3.4.2.1.4a Diagrama de Bloques del Sistema PID

Para ello, se realimenta la variable de salida (en este caso la velocidad) para generar a partir de esta y de la señal de control una función de error que el controlador tratará de minimizar. Aunque esto implica un mayor costo en hardware, es más adecuado para aplicaciones en las que existen perturbaciones externas que pueden variar el funcionamiento del sistema. Para el correcto funcionamiento de estos motores, se necesita incorporar un detector de posicionamiento final del eje.

Hay multitud de tipos de detectores de posición, como los resistivos (potenciómetros acoplados al eje), magnéticos, capacitivos, etc., pero los más prácticos y sencillos son los de tipo óptico. Los detectores de tipo óptico constan de un emisor y un receptor de luz, entre los que gira un disco acoplado al eje del motor. El disco tiene zonas opacas y transparentes (ranuras), que cortan o dejan atravesar el haz luminoso desde el emisor al receptor, provocando impulsos eléctricos. Este disco ranurado proporciona suficiente información para determinar el ángulo girado y el sentido de giro. Cuantas más ranuras disponga, mayor será la precisión en la detección del movimiento.

Para controlar el posicionamiento del motor de CD hay que tener en cuenta que la velocidad que desarrolla el motor es directamente proporcional a la tensión aplicada en el inducido y al flujo producido. Para el control de la velocidad se manipulan los dos factores antes mencionados dependiendo de las características requeridas en cada caso particular.

Debido a que estos sistemas de control son complejos se usará el Pololu JRK 23v3, este es un dispositivo de control de motor con sistema de control PID viene con una interfaz para

que el usuario configure la diferentes propiedades para el manejo de motor (Ver Fig. 3.4.2.1.4b)

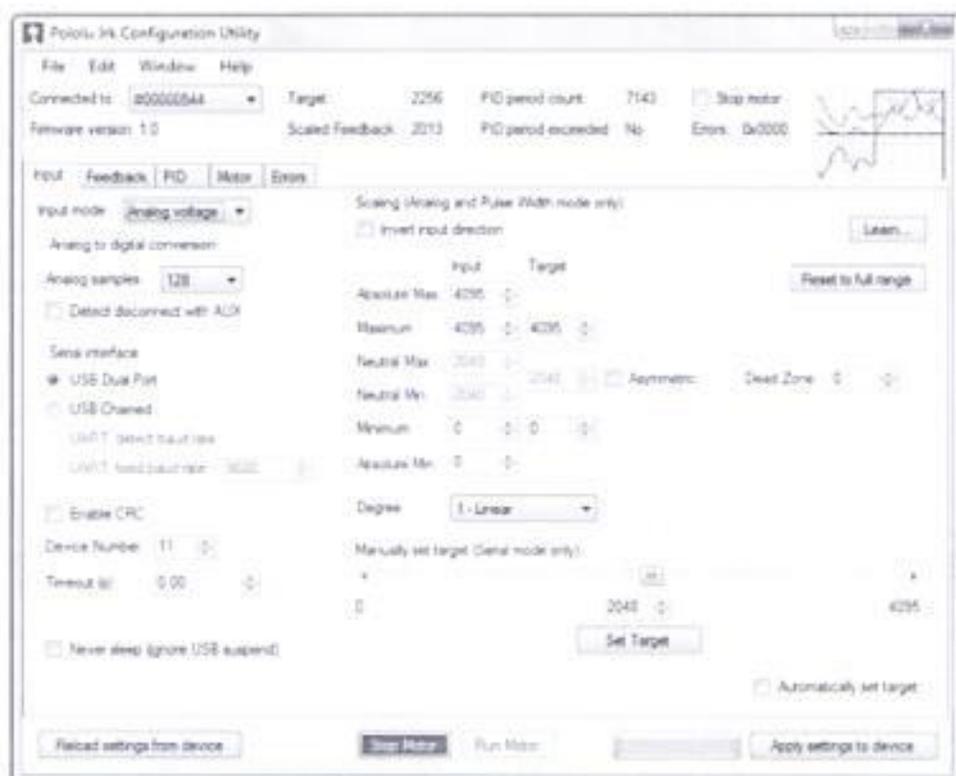


Fig. 3.4.2.1.4b Interfaz del Jrk 23v3

La pestaña de entrada de la utilidad de configuración jrkl contiene la configuración de retroalimentación del sistema (consiste de un motor DC y un sensor de retroalimentación) es externamente controlado y vigilado. Lo más importante, hay tres modos de entrada:

3.4.2.1.4.1 Usando el interfaz Serial

El jrkl tiene tres interfaces de serie diferente. En primer lugar, tiene las líneas RX y TX. El jrkl puede enviar bytes en la línea TX. Si el jrkl está en modo de entrada serial, la línea RX puede ser usado para recibir, TTL (0 - 5 V) bytes de serie No-invertida. Si el jrkl no está en modo de entrada Serial, no puede recibir bytes en RX porque la línea se utiliza para la tensión analógica o medida de ancho de pulso. En segundo lugar, el jrkl se muestra como dos puertos seriales virtuales en mi equipo si está conectado a través de USB. Uno de estos puertos se llama el puerto de mando y el otro se llama el puerto TTL. Usted puede determinar los números de puerto COM de estos puertos mirando en el Administrador de dispositivos de su equipo.

El Jrkl puede ser configurado para estar en uno de estos tres básicos modos serial como se ha detallado en los capítulos anteriores pero en nuestro presente proyecto vamos a especificar a continuación el modo de comunicación Serial que utiliza nuestro sistema para controlar la velocidad del motor que nos brindara una mayor eficiencia y confiabilidad en la hora de poder transmitir datos

3.4.2.1.4.1.1 UART

En este modo, Las Líneas TX y RX (Ver Fig. 3.4.2.1.4.1.1a UART Serial) puede ser usado para enviar comandos al Jrk y recibir respuestas de él. Cualquier byte recibido en RX se enviará al puerto de comandos, pero bytes enviados desde el puerto de comandos serán ignoradas.

El Puerto TTL no se utiliza. La velocidad de transmisión en TX y RX pueden ser detectados automáticamente por el Jrk cuando un byte "0xAA" es recibido en RX, o puede ser fijado a un valor fijo de antemano. Este modo sólo está disponible cuando el modo de entrada es serial. Este modo te permite controlar el Jrk (y enviar bytes a un programa de modo serial al equipo) utilizando un microcontrolador u otro dispositivo de serie TTL.

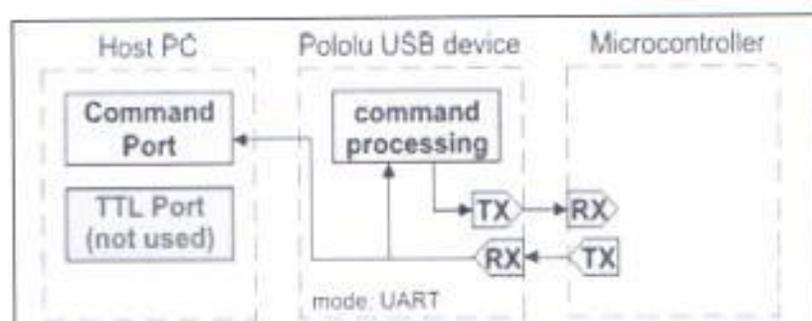


Fig. 3.4.2.1.4.1.1a UART Serial

3.4.2.1.4.1.2 TTL SERIAL

Si el Jrk en su entrada está en modo de interfaz serial, entonces su línea Serial de recepción, RX, puede recibir bytes cuando se conecta a un nivel Lógico (de 0 a 4.0 a 5 V, o "TTL"), de señal Serial no invertida. Los bytes enviados al Jrk en RX puede ser comandos del jrk o

una secuencia arbitraria de datos que el Jrk pasa a un ordenador mediante el puerto USB, dependiendo de qué modo serial se encuentra el Jrk. La tensión en el pin RX no debe ir por debajo de 0 V y no debe exceder de 5 V.

El Jrk proporciona un nivel lógico (0-5 V) de salida en su línea de transmisión Serial, TX. Los bytes enviados por el Jrk en TX pueden ser respuestas a comandos que requieren información o una corriente arbitraria de datos que el Jrk está recibiendo desde un ordenador por medio del puerto USB, dependiendo del modo serial que el Jrk se encuentre. Si no deseo recibir bytes TTL de manera serial desde el Jrk, puede dejar la línea TX desconectado.

La interfaz serial es **Asincrónica**, lo que significa que el emisor y el receptor cada vez de forma independiente los bits los comanda en modo serial. El modo TTL Serial Asincrono está disponible en forma de módulos de hardware llamado "UART" en muchos microcontroladores.

Salida serial asincrona también pueden ser "bit-banged" de una línea de salida digital estándar en el software de control.

El formato de datos es de 8 bits, un stop bit, sin paridad, que se expresa a menudo como el 8-N-1. El siguiente diagrama muestra la interfaz serial TTL asincrónica de bytes no invertida (Ver fig. 3.4.2.1.4.1.2a)



Fig. 3.4.2.1.4.1.2 TTL bytes asincrónica No-invertida

Un byte de transmisión se inicia con un "bit de inicio", seguido por los bits del byte, el bit menos significativo (LSB) en primer lugar. Los Unos lógicos se transmiten como alto (V_{cc}) y los ceros lógicos se transmiten como bajo (0 V), por lo que este formato se conoce como modo serial "no-invertida". El byte es terminado por un "bit de parada". Debido a que cada byte requiere un bit de inicio, 8 bits de datos, y un bit de parada, cada byte tiene 10 veces más bits a transmitir, por lo que la tasa más rápida posible de datos en bytes por segundo es la velocidad dividido por diez. La velocidad de transmisión máxima del Jrk es de 115.200 bits por segundo, que es la tasa máxima realizable de datos.

3.4.2.1.4.2 Protocolos de Comando

Usted puede controlar el Jrk mediante los comandos serial. Si el modo del jrk en su entrada es serial y su modo Serial es "UART, detector de velocidad de transmisión", primero debemos enviar el byte 0xAA (170 en decimal) en la línea de RX (por lo que puede detectar la velocidad de transmisión) antes de enviarle cualquier otro comando.

El protocolo de comando serial del Jrk es bastante sencillo. La comunicación se realiza mediante el envío de paquetes de mando integrada por un solo byte de comando seguido de cualquier bytes de datos que requiere comandos. Los bytes de Comando siempre tienen sus bits más significativos de ajuste (es decir, rango 128 a 255, o 0x80, 0xFF en hexadecimal).

mientras que los bytes de datos siempre tienen sus bits más significativos despejados (es decir, rango 0 a 127, o 0x00-0x7F en hexadecimal). Esto significa que cada byte de datos sólo puede transmitir siete bits de información.

El Jrk responde a dos sub-protocolos:

3.4.2.1.4.2.1 Protocolo Compacto

Este es el más simple y más compacto de los dos protocolos, sino que es el protocolo que debe utilizar si su Jrk es el único dispositivo conectado a la línea serial. El paquete compacto de comando de protocolos del Jrk es simplemente:

byte de comando (con un set MSB), Cualquier datos de bytes necesarios.

Por ejemplo, si queremos establecer la meta de 4080 (el valor más alto alcanzable mediante el conjunto de baja resolución de comandos de destino), que podría enviar la secuencia de bytes es la siguiente:

En hexadecimal: **0xE1, 0x7F**

En decimal: **225, 127**

El byte 0xE1 es el set target de baja resolución hacia delante, y el byte de datos contiene la velocidad del motor.

3.4.2.1.4.2.2 Protocolo del Pololu

Este protocolo es compatible con el protocolo serial ya que puede cambiar la línea de datos en modo serial con otros controladores seriales que son adicionales usado por otros motores

Seriales y controladores de servos. Otros Pololu ignoran los protocolos de comandos. Los bytes restantes en el paquete de comandos son los mismos que el paquete compacto del protocolo de comando que se envía, con una diferencia clave: el byte del paquete compacto del protocolo de comando es ahora un byte de datos para el comando 0xAA y por lo tanto, debe tener su bit más significativo despejado.

Por lo tanto, el paquete de comandos es la siguiente:

0xAA, byte número de dispositivo, byte de comando con MSB despejado, los bytes de datos necesarios

Por ejemplo, si queremos establecer la meta de 4080 para un Jrk enviamos la secuencia de bytes siguientes:

En hexadecimal: **0xAA, 0x0B, 0x61, 0x7F**

En decimal: **170, 11, 97, 127**

Tenga en cuenta que es el 0x61 es el comando 0xE1 con su bit más significativo.

3.4.2.1.4.3 Comandos de Control de Motores

El Jrk tiene varios comandos serial para activar el motor para su encendido y apagado. Estos comandos están principalmente destinados a ser utilizados en su entrada de modo serial, pero se puede utilizar en cualquier modo de entrada para encender o apagar el motor desde un ordenador.

3.4.2.1.4.3.1 Motor apagado

Protocolo compacto: **0xFF**

Protocolo Pololu: **0xAA**, número de dispositivo, **0x7F**

Este comando apaga al motor mediante el seteo del comando de error del bit. El Jrk no reiniciará al motor hasta que reciba un comando Set Target. El Jrk puede ser configurado para cualquier freno o error, mientras el motor está apagado.

3.4.2.1.4.3.2 Set Target High Resolution

Protocolo compacto, binario: **110LLLLL, 0HHHHHHH**

Protocolo compacto, hexagonal: **0xC0 + target low de 5 bits, target high 7 bits**

Protocolo Pololu, binaria: **10101010, device number, 010LLLLL, 0HHHHHHH**

Protocolo Pololu, hexagonal: **0xAA, device number, 0x40 + target low 5 bits, target high 7 bits (donde target es el número de 12 bits HHHHHHLLLLL)**

Este comando borra los errores de espera de un bit (si su entrada es de modo serial) le permite establecer la meta de 12 bits a cualquiera de sus valores permitidos (0-4.095) lo que significa que el target depende sobre que realimentación el Jrk se encuentre. Los 5

bits inferiores de los bytes de comando representan los 5 bits inferiores del target, mientras que el menor de 7 bits del byte de datos representan la parte superior de 7 bits del target.

3.4.2.1.4.3.3 Set Target Low Resolution Forward

Protocolo compacto: **0xE1, magnitud**

Protocolo Pololu: **0xAA, device number, 0x61, magnitud**

Este comando borra los errores de espera de un bit (si su entrada es de modo serial) y establece el objetivo a un valor de 2048 o mayor determinado por la magnitud (0-127).

- Si el modo de votos es analógica o tacómetro, entonces la fórmula es

$$\text{Target} = 2048 + 16 \times \text{magnitud.}$$

- Si el modo de votos es Ninguno (modo de control de velocidad), entonces la fórmula es

$$\text{Target} = 2048 + (600/127) \times \text{magnitud.}$$

Esto significa que una magnitud de 127 fijó el objetivo del ciclo de trabajo que transmita la máxima velocidad(600), mientras que una magnitud de cero hará que el motor se detenga

3.4.2.1.4.3.4 Set Target Low Resolution Reverse

Protocolo compacto: **0xE0, magnitud**

Protocolo Pololu: **0xAA, device number, 0x61, magnitud**

Si la magnitud es cero, entonces este comando es equivalente para el apagado del motor: la espera poco de comandos de error se establecerá, por lo que el jrk se apague el motor hasta que otro comando Set de destino se recibe.

De lo contrario, este comando borra la espera poco de comandos de error y (si es del modo de entrada es de serie) establece el objetivo a un valor inferior a 2048 que es determinado por la magnitud (0-127).

- Si el modo de votos es analógica o tacómetro, entonces la fórmula es
$$\text{Target} = 2,048 + 16 \times \text{magnitud}.$$
- Si el modo de votos es Ninguno (modo de control de velocidad), entonces la fórmula es
$$\text{Target} = 2048 - (600/127) \times \text{magnitud}.$$

Esto significa que una magnitud de 127 se estableció el objetivo de revertir el ciclo de trabajo de alta velocidad (-600).

3.5 Modelo 3D

3.5.1 Control PID de velocidad de motor DC usando comunicación mediante señales de Radio Comunicación (RC)

A continuación se muestra la visualización en 3D del proyecto armado en Hardware, donde se observa la tarjeta Jrk 21v3 USB Controller with Feedback, el micro metal Gearmotor, el Encoder armado en conjunto con la rueda de caucho puesta sobre una base de plástico

como soporte para que mantenga el dispositivo para no estar cogiéndolo con las manos y no presente algún percance si es que se nos llegara a caer y por último el circuito electrónico formado por el receptor y su transmisor el joystick. (Ver Figura 3.5.1a).

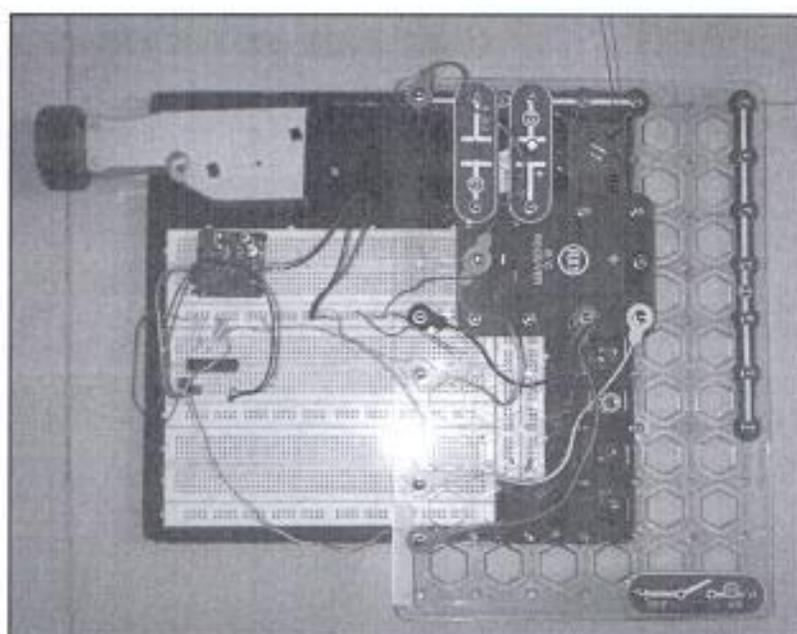


Figura 3.5.1a Control PID de velocidad de motor DC usando comunicación mediante señales de Radio Comunicación (RC)

3.5.2 Encoder con rueda. (Ver Figura 3.5.2a).

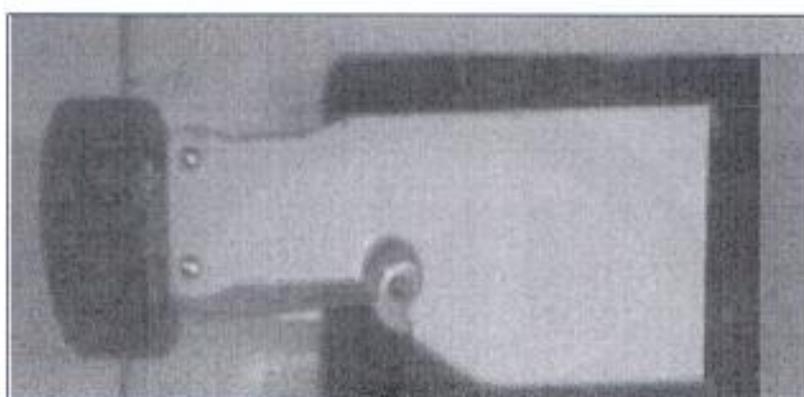


Figura 3.5.2a Encoder con Rueda con su base de plastica

3.5.3 Tarjeta Pololu JRK 21v3 (Ver Figura 3.5.3a)

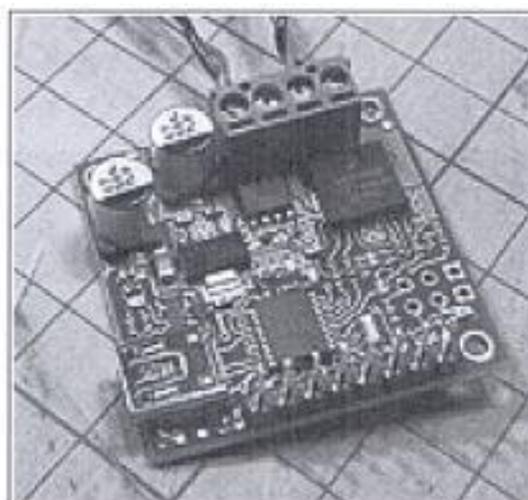


Figura 3.5.3a Tarjeta Pololu JRK 21v3.

3.5.4 Cable USB (Ver Figura 3.5.4a)



Figura 3.5.4a Cable USB

3.5.5 R/C Receiver (Ver Figura 3.5.5a)

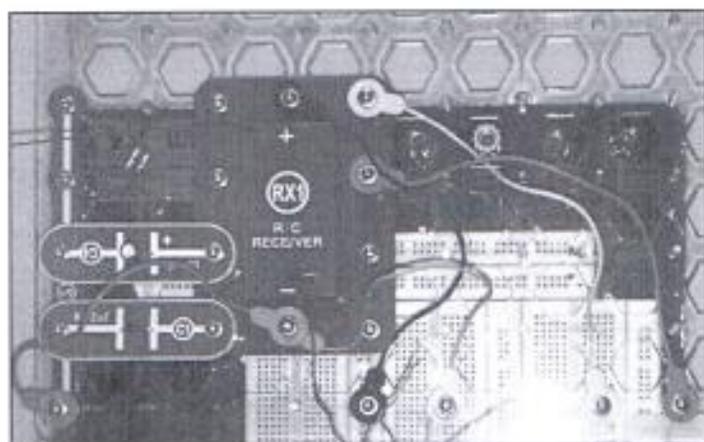


Fig. 3.5.5a R/C Receiver

3.5.6 Joystick (Ver Figura 3.5.6a)

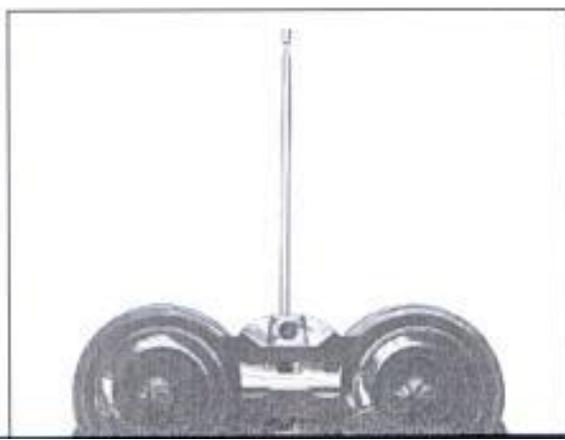


Figura 3.5.6a Joystick

3.5.7 ATtyni 2313 (Ver Figura 3.5.7a)

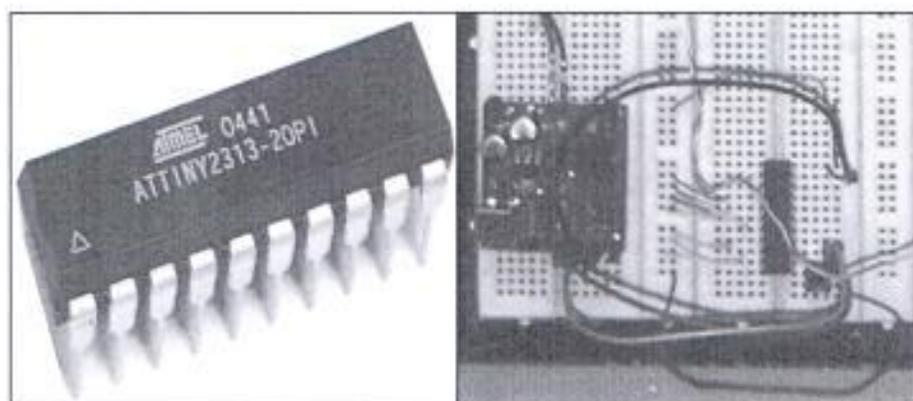


Figura 3.5.7a ATtyni 2313

Capítulo 4

4. Simulación/Implementación

4.1 Simulación del Proyecto

En esta sección mostraremos gráficamente el comportamiento de las diferentes variables del sistema cuando la señal de entrada es modificada, esto quiere decir, cuando aceleramos hacia una dirección, desaceleramos y cuando aceleramos en dirección contraria.

4.1.1 Ciclo de Trabajo del Sistema



Figura 4.1.1a Ciclo de Trabajo (Sistema Acelerando)

En este gráfico podemos observar la variable Entrada (color Turquesa) y la variable Ciclo de Trabajo (Color Verde Oscuro), las cuales van modificando su valor a medida que el usuario aumenta la Entrada a través del software usado. Observamos que el Ciclo de Trabajo llega a su máximo nivel cuando la rueda esta totalmente acelerada.

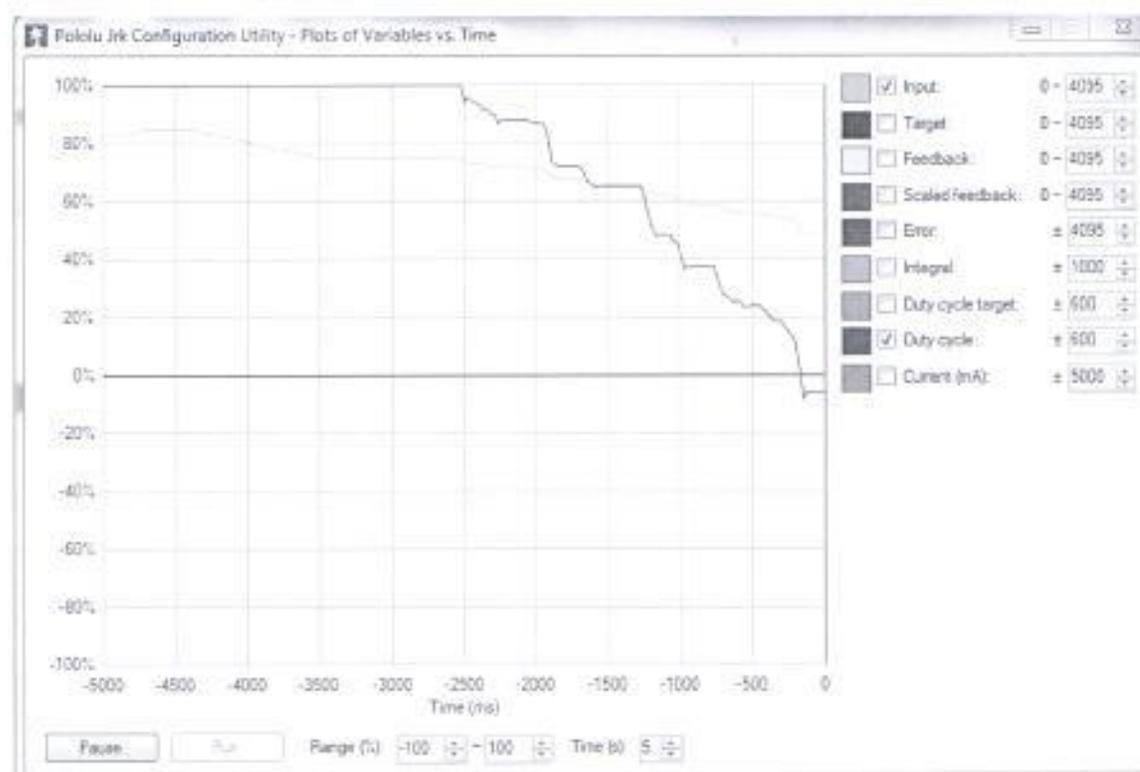


Figura 4.1.1b Ciclo de Trabajo (Sistema Desacelerado)

Se observa que el sistema esta llegando a su estabilidad, esto quiere decir, que la rueda no se encuentra en movimiento, por lo tanto la variable Entrada y el Ciclo de Trabajo disminuyen su valor al mínimo posible.

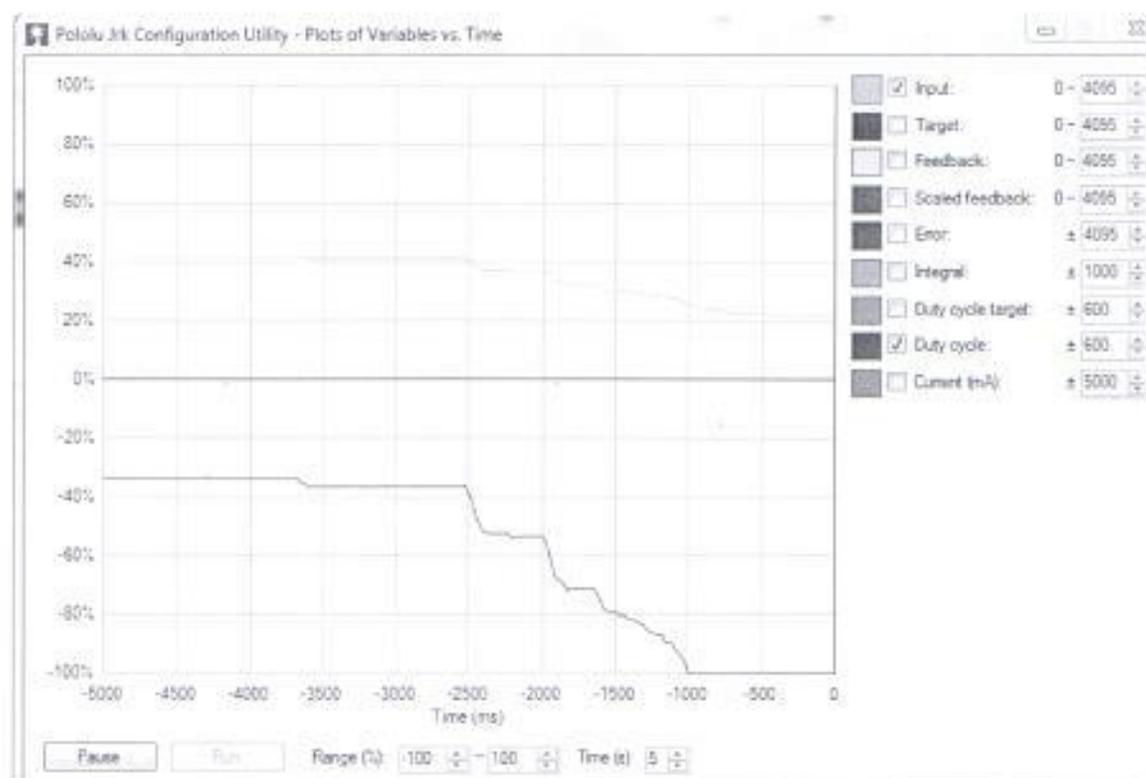


Figura 4.1.1c Ciclo de Trabajo (Sistema Acelerando en sentido invertido)

El resultado de ésta gráfica es bien parecida a la **Figura 4.1.1a** sino que el movimiento de la rueda es en sentido contrario, esto quiere decir, que gira en contra de las manecillas del reloj

4.1.2 La señal de Retroalimentación.

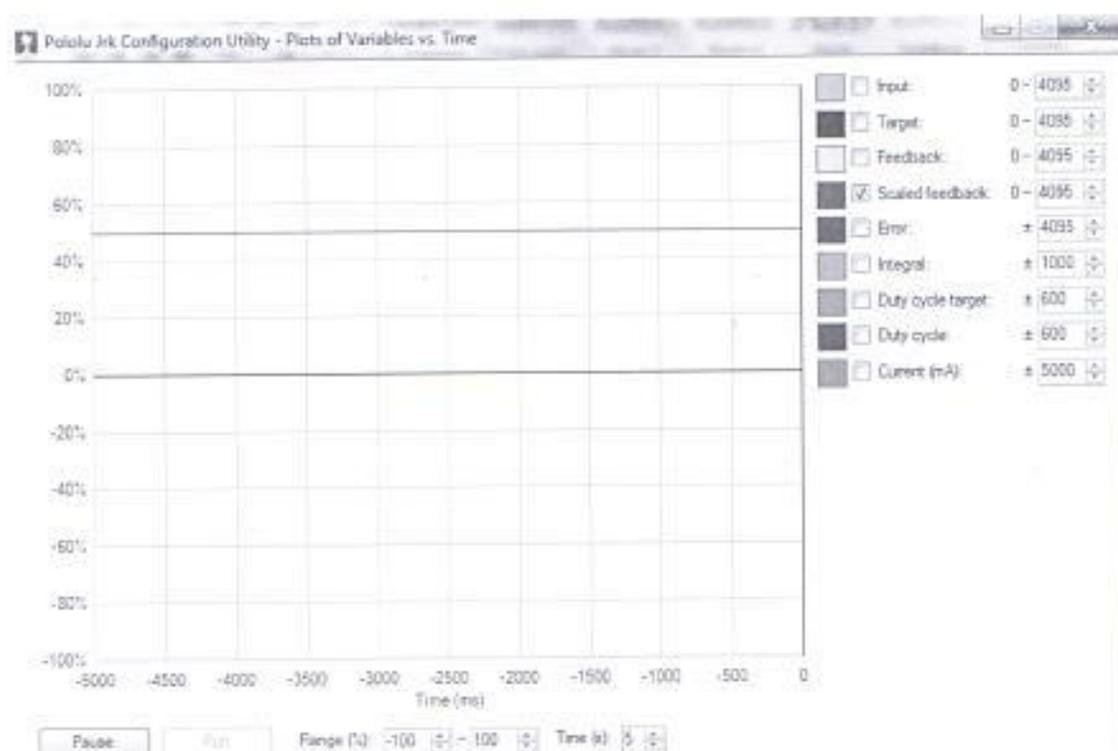


Figura 4.1.2a Señal de Retroalimentación

En ésta figura podemos observar la señal obtenida de nuestro sistema de realimentación (Encoder), la cual se encuentra conectada al pin Fb de la tarjeta controladora y mediante el software Jrk Configuration Utility nos permite estudiar su comportamiento en tiempo real.

4.1.3 La señal Integral.

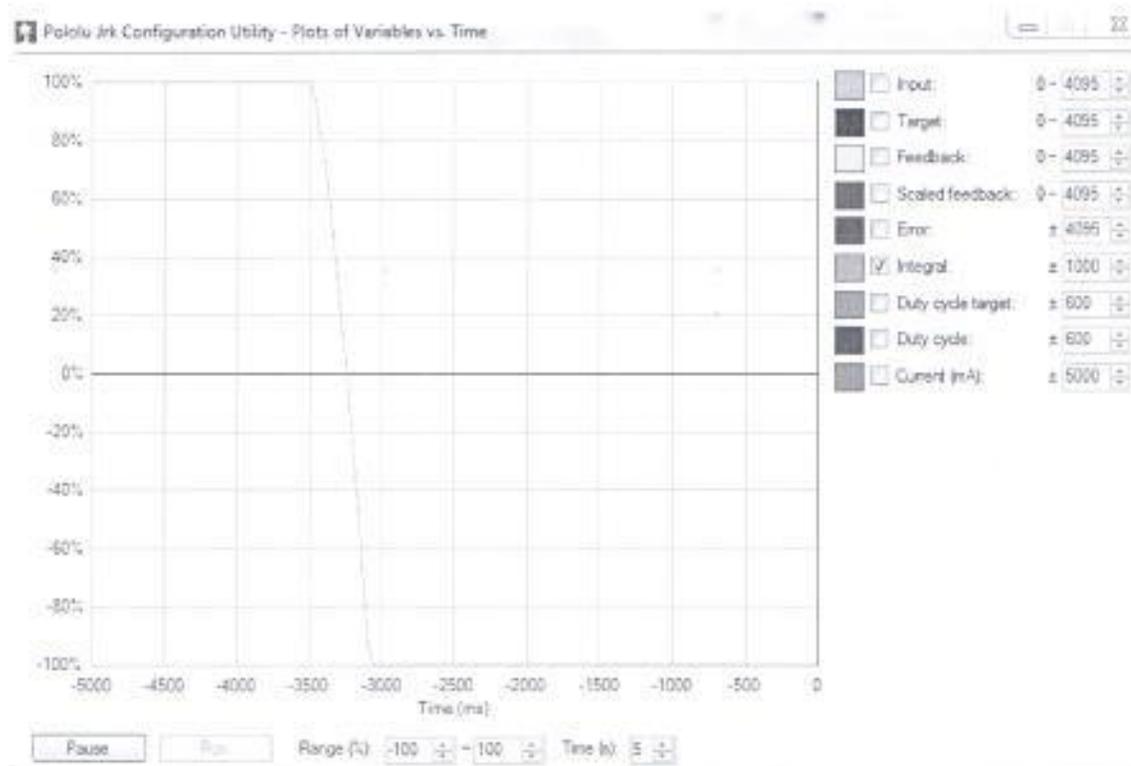


Figura 4.1.3a Señal Integral.

Mediante ésta imagen podemos observar el comportamiento de la señal Integral de nuestro proyecto, de acuerdo al sentido de movimiento de la rueda la variable toma diferentes valores al 100% en cada escenario.

4.1.4 La señal Error.

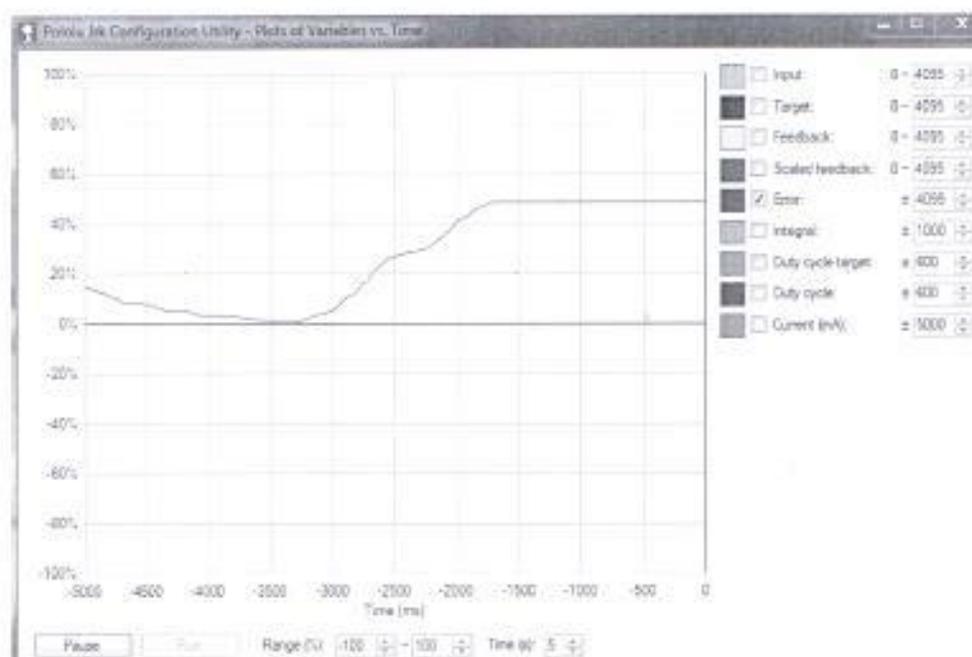


Figura 4.1.4a Error (Motor en contra de las manecillas de reloj)



Figura 4.1.4b Error (Motor en sentido de las manecillas de reloj)

4.3 Simulación del AVR con el ATMEL AVR Studio 4

El AVR Studio 4 es un entorno de desarrollo integrado para la depuración de AVR software. El estudio del AVR permite la simulación de chips y la emulación en circuito para la familia de microcontroladores AVR. La interfaz de usuario está especialmente diseñado para ser fácil de usar y dar información general a la información completa tanto para la simulación y emulación que proporciona una rápida curva de aprendizaje.

El AVR Studio 4 (Ver Fig. 4.3a) tiene una gama de los microcontroladores de la familia ATMEL de tal manera que para nuestro proyecto utilizaremos el ATtiny 2313 que es un micro eficaz y altamente confiable que cumple las características deseadas para su uso a la hora de compilar un programa.



Fig. 4.3 AVR Studio 4

4.3.1 Primeros Pasos

El AVR Studio utiliza un archivo de objeto COF para la simulación. Este archivo se crea con el compilador de C a través de la selección de COF como el tipo de archivo de salida. Al iniciar el AVR Studio, ya sea que la selección a través del menú Inicio o seleccionando el icono del programa (si está disponible). Cualquier método producirá el IDE que se muestra a continuación (**Ver Fig. 4.3.1a**) Una vez que el IDE está en ejecución, seleccione Abrir archivo ya sea a través del archivo de menú desplegable o haciendo clic en el botón Abrir archivo.

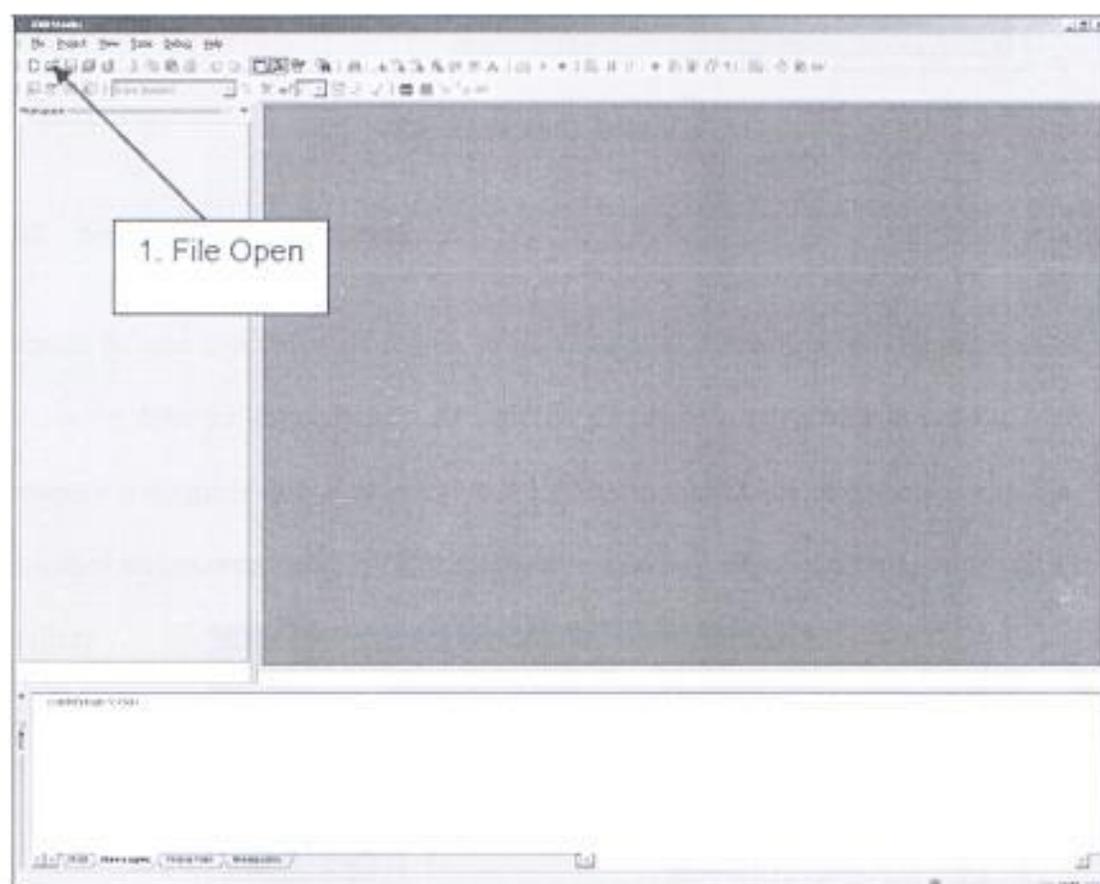


Fig. 4.3.1a AVR IDE

Seleccione el archivo COF (Ver Figura 4.3.1b) para la simulación a través de la ventana de Abrir archivo. Tenga en cuenta esta ventana de navegación utiliza el estándar de Windows. Haga doble clic sobre el archivo o haciendo clic en el archivo y seleccionando el botón Abrir puede abrir el archivo.

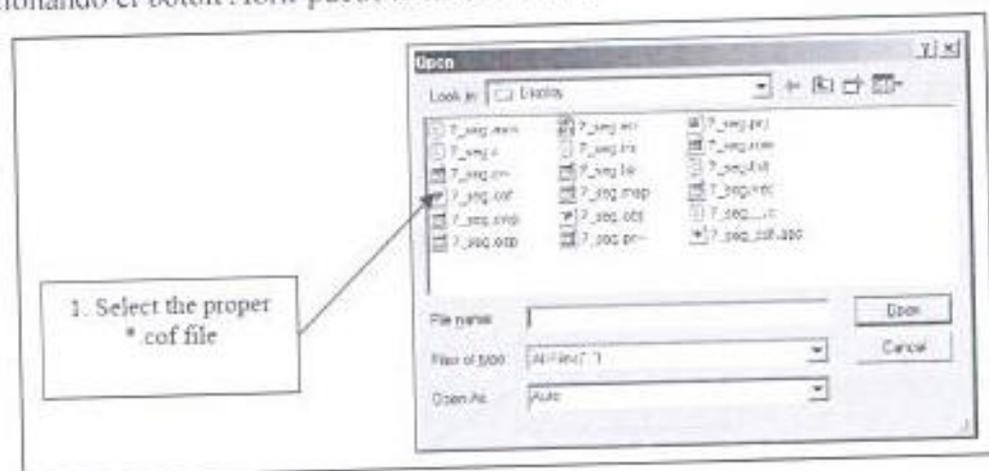


Fig. 4.3.1b Abrir archivo

4.3.2 Selección de dispositivos

Después de que el archivo de origen se ha abierto, el dispositivo y la plataforma de depuración debe ser especificado. Al hacer la simulación, seleccione la opción AVR Simulator y asegurar que el dispositivo AVR destino adecuado es seleccionado. Una vez que el microcontrolador AVR y la plataforma se han seleccionado, hacemos clip n

Finalizar

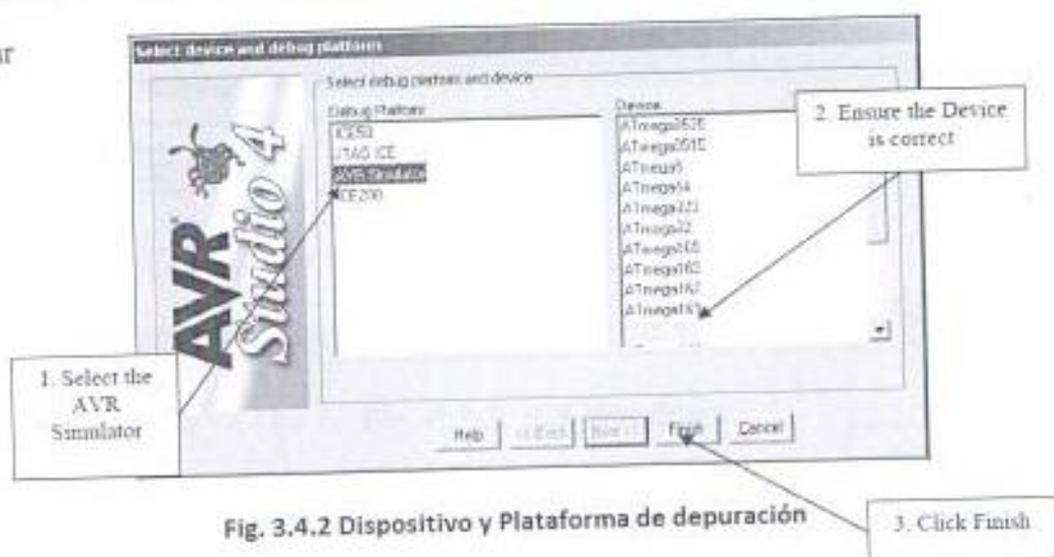


Fig. 3.4.2 Dispositivo y Plataforma de depuración

4.3.3 IDE Windows

El IDE tiene varias ventanas que proporcionan información importante para el usuario. Estas ventanas se pueden abrir de forma automática por el software o que necesitan ser activados por el usuario. Independientemente de cómo se activan las ventanas, que se pueden mover y cambiar de tamaño para ajustarse al gusto del usuario. Las ventanas principales de interés son las ventanas de área, código fuente, de salida, y el reloj(Ver Fig. 4.3.3 IDE Windows)

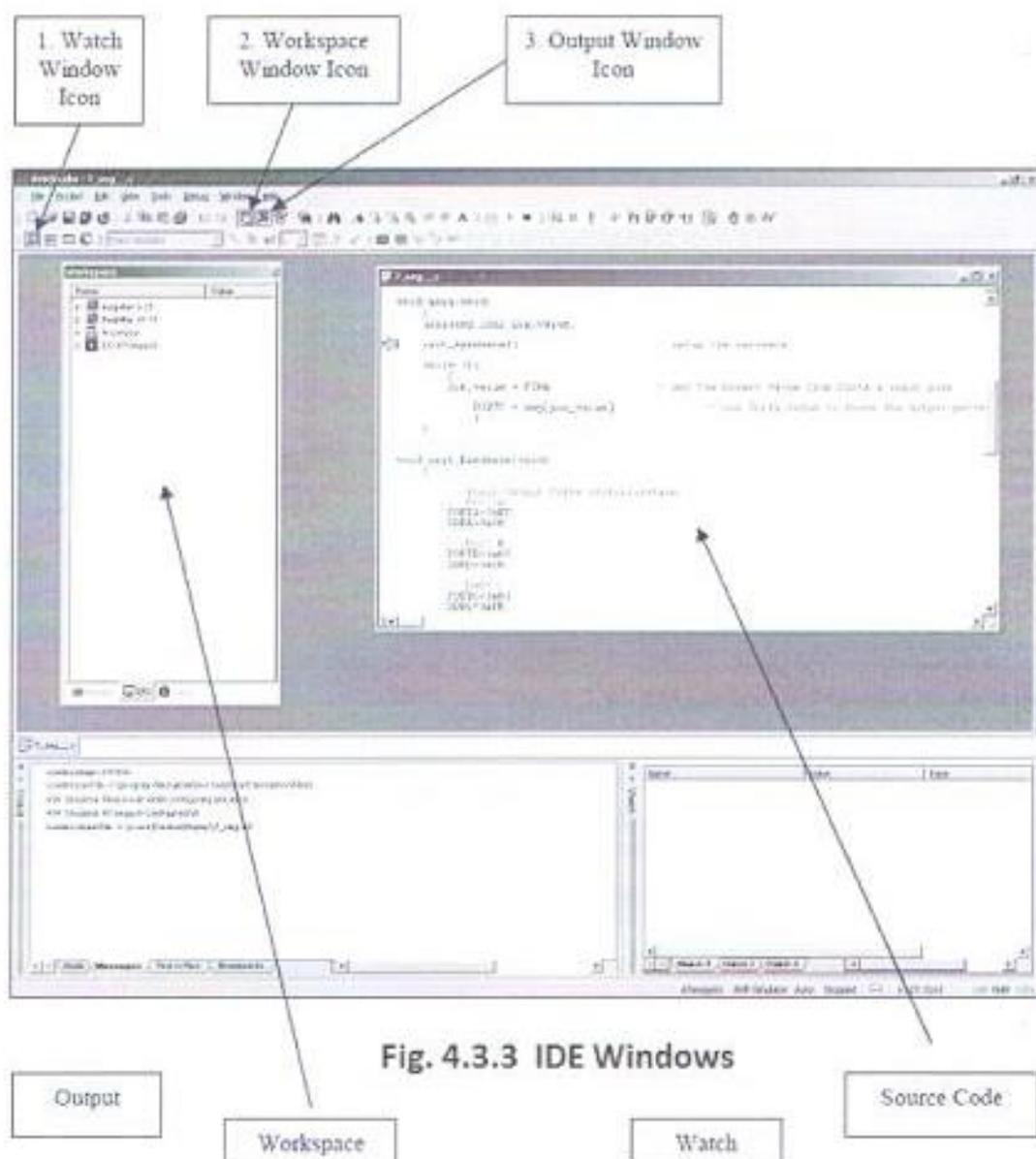
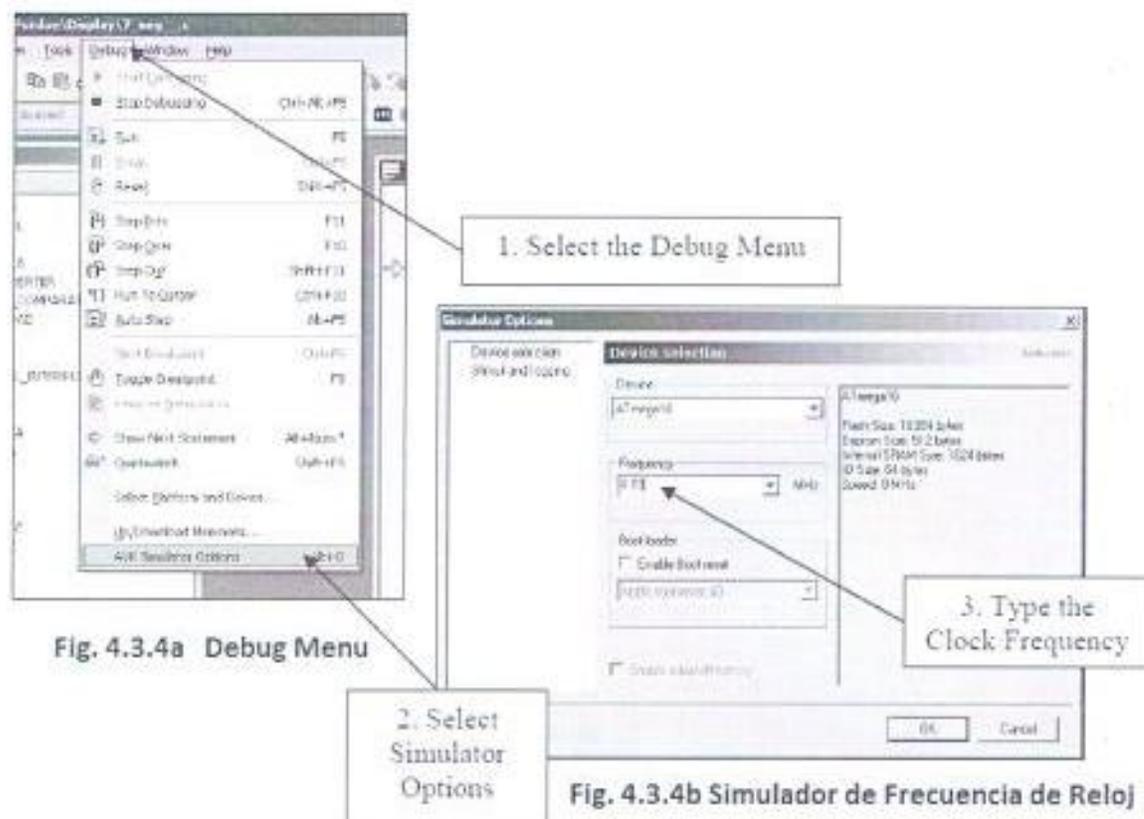


Fig. 4.3.3 IDE Windows

4.3.4 Opciones de Simulación

Antes de iniciar realmente una simulación, la frecuencia de la AVR objetivo debe estar ajustado. El Consejo de Desarrollo de MegaAVR PRLLC funciona a una frecuencia de reloj de 6.0MHz. Esta frecuencia sucede NO ser una de las selecciones del menú desplegable. Por lo tanto, el usuario debe escribir manualmente este valor en el programa.



Después de haber realizado todo estos pasos para la selección respectiva del Microcontrolador y almacenamiento del archivo procedemos a la compilación de tal manera que observaremos su proceso que fue eficiente sin ningún error de compilación y que su uso se encuentre listo para ser cargado al Microcontrolador que en este caso estamos trabajando para nuestro proyecto.

4.4 Análisis del código Fuente en Lenguaje C en AVR Studio 4

```

#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 4000000UL
#include <util/delay.h>

/* Prototypes */
void InitUART (unsigned char baudrate);
unsigned char ReceiveByte (void);
unsigned char dato0;
unsigned char dato1;
unsigned short target;
unsigned char i;
void TransmitByte (unsigned char data);

/* -----
Main - program that receives a character then transmits back the next character.
An example would be if you send in an A, the chip will return a B
----- */
int main (void)
{

    DDRB=0x00; // puerto A como entrada
    target = 0xC9D;
    InitUART (51); /* Set the baudrate to 9600 bps using a 8MHz crystal */

    while (1)
    {

        _delay_ms(550);
        i = PINB;

```

```

if (i == 0x01)
{
    target++;
    dato0 = 0xC0+(target&0x1F);
    dato1 = (target >> 5) & 0x7F;

    TransmitByte (0xAA);
    TransmitByte (0x0C);
    TransmitByte (dato0);
    TransmitByte (dato1);
}
else if (i == 0x02)
{
    target--;
    dato0 = 0xC0+(target&0x1F);
    dato1 = (target >> 5) & 0x7F;

    TransmitByte (0xAA);
    TransmitByte (0x0C);
    TransmitByte (dato0);
    TransmitByte (dato1);
}
}
}

```

/ Initialize UART */*

void InitUART (unsigned char baudrate)

```

{
    /* Set the baud rate */
    UBRRL = baudrate;

    /* Enable UART receiver and transmitter */
    UCSRB = (1 << RXEN) | (1 << TXEN);

```

```
/* set to 8 data bits, 1 stop bit */
UCSRC = (0<<USBS) | (0<<UCSZ2) | (1<<UCSZ1) | (1<<UCSZ0);

}

/* Read and write functions */
unsigned char ReceiveByte (void)
{
    /* Wait for incoming data */
    while (!(UCSRA & (1<<RXC)));

    /* Return the data */
    return UDR;
}

void TransmitByte (unsigned char data)
{
    /* Wait for empty transmit buffer */
    while (!(UCSRA & (1<<UDRE)));

    /* Start transmission */
    UDR = data;
}
}
```

4.5 Simulación - Prueba de Comunicación Serial UART con el ATtiny 2313

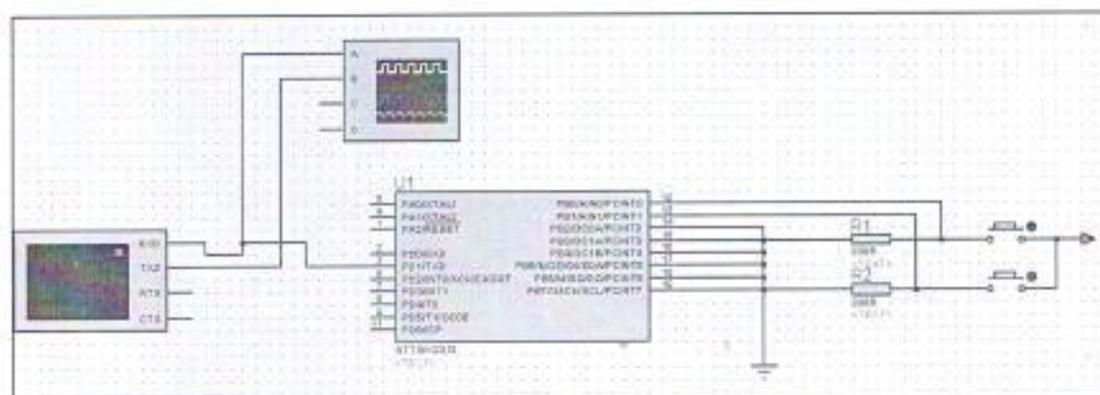


Fig. 4.5a Simulación - Transmisión UART con el ATtiny 2313

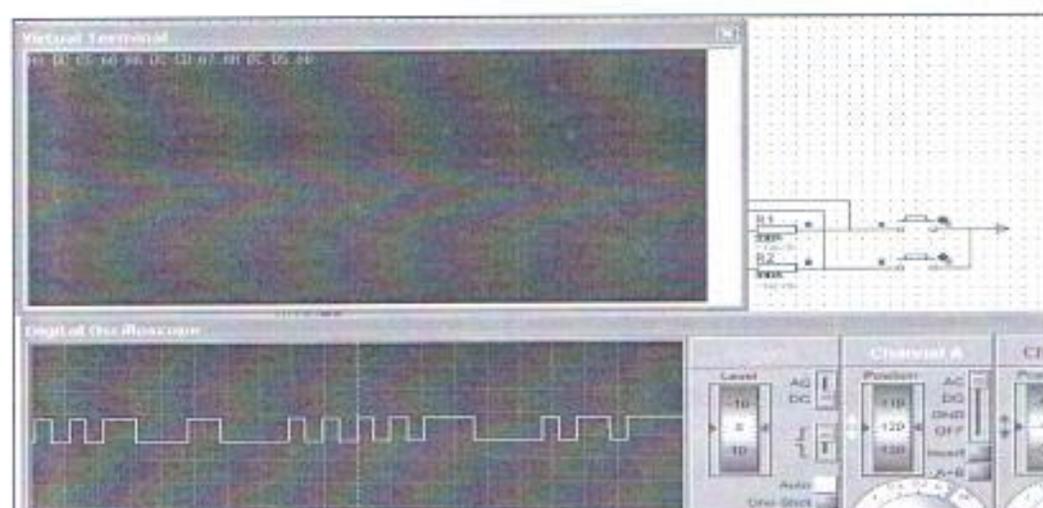


Fig. 4.5b Transmisión Compac Protocol Aumento de Velocidad Hexadecimal y Binario

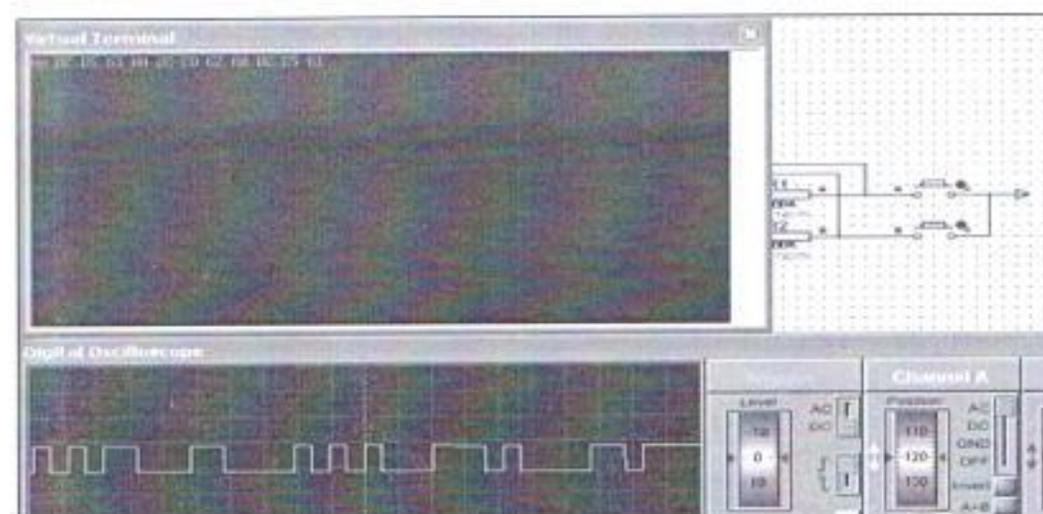


Fig. 4.5c Transmisión de Compac Protocol Disminución de Velocidad Hexadecimal y Binario

4.6 Implementación del Proyecto

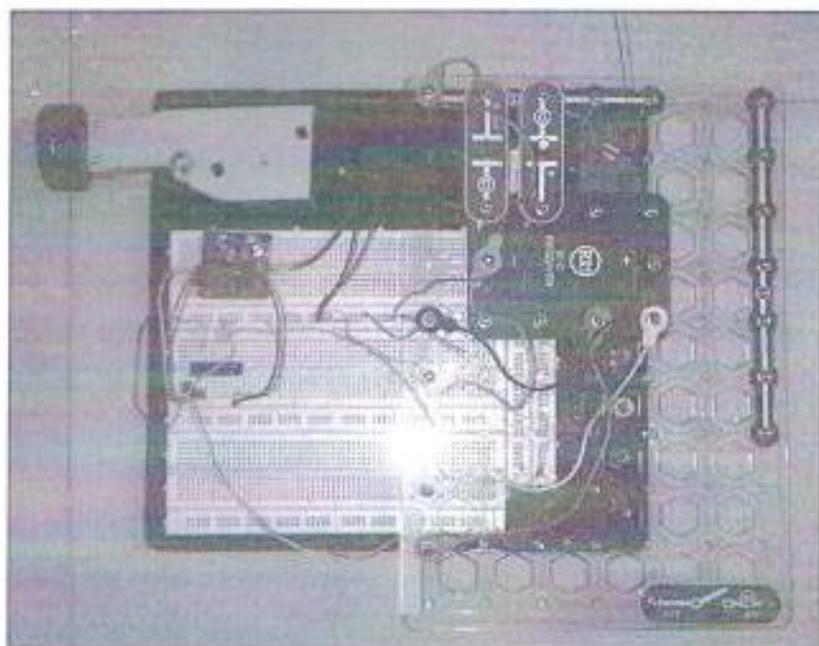


Figura 4.6a Proyecto Implementado.

En esta foto podemos darnos cuenta que el Controlador de Velocidad de un Motor DC mediante RC (radio Comunicación) se encuentra totalmente implementado y funcionando al cien por ciento. La fuente generadora de poder es de 15 V y es conectada a los pines de de alimentación del motor en la tarjeta Jrk 21v3 USB Controladora con Realimentación (Feedback). Se puede a mirar a simple vista que el sistema de realimentación (Encoder) esta conectado a la tarjeta controladora, lo cual permite cerrar el lazo de control de nuestro proyecto.

CONCLUSIONES

- 1) El Jrk es un dispositivo Diseñado específicamente para el manejo de Motores DC que puede controlar de 4 formas diferentes la velocidad de un motor así también como su posición y sentido de giro, dando una amplia variedad de aplicaciones para este tipo de proyectos; también cuenta con un sistema de control PID con retroalimentación el cual nos ayuda a equilibrar nuestro sistema en caso de cambios externos.
- 2) Para controlar la velocidad del motor se utilizo la comunicación UART entre el ATtiny 2313 y el Jrk en modo UART debido a que el receptor nos daba 5v o 0v si presionábamos uno de los 4 botones que usamos y solo se tiene una entrada en el Jrk; para este modo de comunicación se necesita enviar un protocolo de tramas llamado Compac protocol el cual consta de 4 tramas.
- 3) Para tener una idea mas exacta de cómo establecer la comunicación mediante el Compac protocol se utilizó el Pololu Serial Transmitter V1.3 con el cual se envió las 4 tramas de datos, este programa tiene un display en Hexadecimal para ver los datos enviados y recibidos, así se pudo observar cómo se envía las tramas al Jrk y tener una comunicación exitosa.
- 4) Los controladores PID se usan ampliamente en control industrial. Desde una perspectiva moderna, un controlador PID es simplemente un controlador de segundo orden con integración. Históricamente, sin embargo, los controladores PID se ajustaban en términos de sus componentes P, I y D. La estructura PID ha mostrado empíricamente ofrecer suficiente flexibilidad para dar excelentes resultados en muchas aplicaciones.
- 5) El término básico en el controlador PID es el proporcional P, que origina una actuación de control correctiva proporcional el error. El término integral I brinda una corrección proporcional a la integral del error. Esta acción tiene la ventaja de asegurar que en última instancia se aplicará suficiente acción de control para reducir el error de regulación a cero. Sin embargo, la acción integral también tiene un efecto desestabilizador debido al corrimiento de fase agregado. El término derivativo D da propiedades predictivas a la actuación, generando una acción de control proporcional a la velocidad de cambio del error. Tiende dar más estabilidad al sistema pero suele

generar grandes valores en la señal de control. Varios métodos empíricos pueden usarse para determinar los parámetros de un PID para una dada aplicación. Sin embargo, el ajuste obtenido debe tomarse como un primer paso en el proceso de diseño.

RECOMENDACIONES

- 1) El Receiver tiene que trabajar con un voltaje entre 6 y 8 voltios ya que con un voltaje menor la recepción tiene poco muy alcance con el Joystick. Se debe tener en cuenta que el ATtiny 2313 solo puede trabajar hasta 5.5v, debido a esto se tuvo que alimentar el microcontrolador y el encoder con las salidas de voltajes regulados que tiene el Jrk.
- 2) Al trabajar en este proyecto con motores de Corriente Continua tenemos que tener mucho cuidado al configurar los diferentes parámetros del motor, como son la corriente máxima, el ciclo de trabajo máximo y la frecuencia de la señal PWM, para así lograr el funcionamiento deseado y no averiar alguna parte interna del motor DC.
- 3) Recomienda utilizar la detección automática del baud rate para que no exista ningún error de este tipo, sin embargo sino se usa este modo se tiene que tener en cuenta configurar el mismo baud rate tanto para el transmisor como para el receptor para que pueda haber comunicación entre los dos dispositivos.
- 4) Una de las recomendaciones en la transmisión de datos es que no debe haber ningún tipo de delay ni espacios entre cada trama debido a que el protocolo del Jrk no reconocerá ninguna trama y no habrá comunicación.
- 5) Tener siempre precaución al alimentar la tarjeta controladora con un voltaje mayor a 6 V, para que funcione correctamente el regulador interno y pueda éste generar una fuente DC controlada de 5 V que servirá como alimentación para el encoder

BIBLIOGRAFIA

- [1] Spang, Pololu Jrk 21v3 USB Motor Controller with Feedback, Pagina HTML
http://www.skpang.co.uk/catalog/product_info.php?products_id=577
Fecha: 18 de Abril del 2011
- [2] Sapiensman, Fundamentos del Control Industrial, Pagina HTML
http://www.sapiensman.com/control_automatico/index.htm
Fecha: 20 de Abril del 2011
- [3] Direct Industry, Controladores de velocidad de motor DC, Pagina HTML
<http://www.directindustry.es/prod/faulhaber/controladores-de-velocidad-para-motores-dc-7023-536554.html>
Fecha: 24 de Abril del 2011
- [4] Pololu, Pololu Jrk USB Motor Controller User's Guide, Pagina HTML
<http://www.directindustry.es/prod/dunkermotoren-gmbh/controladores-de-velocidad-para-motores-dc-14411-474364.html>
Fecha: 26 de Abril del 2011
- [5] Direct Industry, Controladores de velocidad de motor DC, Pagina HTML
<http://www.directindustry.es/prod/dunkermotoren-gmbh/controladores-de-velocidad-para-motores-dc-14411-474364.html>
Fecha: 28 de Abril del 2011
- [6] Wikipedia, Modulación por ancho de pulsos, Pagina HTML
http://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_ancho_de_pulsos
Fecha: 2 de Mayo del 2011
- [7] Wikipedia, TTL Serial Universal (UART), Pagina HTML
<http://www.pololu.com/catalog/product/1393/faqs>
Fecha: 4 de Mayo del 2011
- [8] Wikipedia, Motores de corriente Continúa, Pagina HTML
http://es.wikipedia.org/wiki/Motor_de_corriente_continua

Fecha: 6 de Mayo del 2011

- [9] Wikipedia, Proporcional Integral Derivativo, Pagina HTML
http://es.wikipedia.org/wiki/Proporcional_integral_derivativo

Fecha: 7 de Mayo del 2011

- [10] Governot, R/C receiver Pro, Pagina HTML
<http://www.helicopteros-rc.es/governor.html>

Fecha: 8 de Mayo del 2011