



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
Y AVIÓNICA**

**TRABAJO DE GRADUACIÓN PARA LA OBTENCIÓN DE
TÍTULO DE:**

**TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN Y AVIÓNICA**

**TEMA: “IMPLEMENTACIÓN DE (4) MÓDULOS PARA EL
DESARROLLO DE PRÁCTICAS CON CELDAS DE CARGA PARA
EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA
UNIDAD DE GESTIÓN DE TECNOLOGÍAS DE LA UNIVERSIDAD
DE LAS FUERZAS ARMADAS-ESPE”**

AUTOR: MÉNDEZ RIVERA CHRISTIAN ANDRÉS

DIRECTOR: ING. MARÍA FERNANDA MOGRO

LATACUNGA

2015

CERTIFICADO

Certifico que el presente Trabajo de Graduación fue realizado en su totalidad por el Sr. **MÉNDEZ RIVERA CHRISTIAN ANDRÉS**, como requerimiento parcial para la obtención del título de **TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA**.

SRA. ING. MARÍA FERNANDA MOGRO
DIRECTORA DEL TRABAJO DE GRADUACIÓN

Latacunga, Julio del 2015

AUTORÍA DE RESPONSABILIDAD

Yo, Méndez Rivera Christian Andrés

DECLARO QUE:

El trabajo de grado denominado “IMPLEMENTACIÓN DE (4) MÓDULOS PARA EL DESARROLLO DE PRÁCTICAS CON CELDAS DE CARGA PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE.”, ha sido desarrollado en base a una investigación científica exhaustiva, respetando derechos intelectuales de terceros conforme las citas constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente, este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de trabajo de grado en mención.

Latacunga, Julio del 2015

Méndez Rivera Christian Andrés

C.I: 070539778-4

AUTORIZACIÓN

Yo, Méndez Rivera Christian Andrés

Autorizo a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución el trabajo “IMPLEMENTACIÓN DE (4) MÓDULOS PARA EL DESARROLLO DE PRÁCTICAS CON CELDAS DE CARGA PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE.”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, Julio del 2015

Méndez Rivera Christian Andrés

C.I: 070539778-4

DEDICATORIA

A Dios por darme salud y fuerza de voluntad, a unos padres tan maravillosos, “Jorge Méndez y Miriam Rivera” el pilar fundamental que me sostiene, ya que sin su sacrificio apoyo incondicional y consejos sabios y oportunos no hubiese podido culminar mi carrera.

A mi hija Cristina Méndez por quien cada día tiene sentido la testigo silenciosa de mi lucha diaria en busca de un mejor futuro para ella, mi esperanza, mi alegría, mi vida y la culminación de mi proyecto y lo que representa.

A mis maestros que con sus conocimientos, ayuda oportuna desinteresada contribuyeron a la finalización de este proyecto.

De igual manera a toda mi familia

Christian Méndez

AGRADECIMIENTO

Mi más sinceros y profundos agradecimiento a la Universidad de las fuerzas Armadas (ESPE), y en especial a la Unidad de Gestión de Tecnologías.

Al Ing. Pablo Pilatásig por brindarme el apoyo y consejos en el desarrollo de este proyecto.

A mis padres, hermano y familiares a quienes siempre tengo presente por darme el apoyo necesario en la realización de este trabajo de graduación para así poder obtener mi título.

A mis maestros y quienes siempre me han enseñado algo.

Gracias a todos

ÍNDICE DE CONTENIDO

CERTIFICADO	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDO	vi
INDICE DE TABLAS.....	x
INDICE DE FIGURAS	xi
RESUMEN	xiii
ABSTRACT	xiv
CAPÍTULO I	1
2.3.1 ANTECEDENTES.	1
2.3.2 PLANTEAMIENTO DEL PROBLEMA.....	1
2.3.3 JUSTIFICACIÓN E IMPORTANCIA.....	2
2.3.4 OBJETIVOS:	3
1.1.1 Objetivo General.....	3
1.1.2 Objetivos Específicos.	3
2.3.5 ALCANCE.	3
CAPÍTULO II	4
MARCO TEÓRICO.....	4
2.1 Celda de Carga.	4
2.1.1 Principio de funcionamiento.....	4
2.1.2 Impedancia de la celda de carga.	6

2.1.3	Aplicación de la celda de carga.	7
2.1.4	Selección de la celda de carga.	7
2.2	Amplificadores de Instrumentación.	11
2.2.1	AD620	12
2.2.1.1	Características.	12
2.4	Microcontrolador Atmega 8.	13
2.4.1	Características del Atmega 8.	13
2.4.2	Descripción de Pines.	15
2.4.3	Registros De Propósito General.	22
2.4.3.1	Registro X, registro Y y registro Z.	23
2.4.4	Unidad aritmética lógica.	23
2.4.5	Flash Programable.	23
2.4.6	Memoria de Datos SRAM.	24
2.4.7	Memoria de Datos EEPROM.	25
2.4.8	Registro de Estado.	25
2.4.8.1	Bit 7-I: Global Interrupt Enable.	25
2.4.8.2	Bit 6-T: Bit Copy Storage.	26
2.4.8.3	Bit 5-H: Half Carry Flag.	26
2.4.8.4	Bit 4-S: Sign Bit, $S = S (+) V$	26
2.4.8.5	Bit 3-V: Two's Complement Overflow Flag.	26
2.4.8.6	Bit 2-N: Negative Flag.	26
2.4.8.7	Bit 1-Z: Zero Flag.	26
2.4.8.8	Bit 0-C: Carry Flag.	26
2.4.9	Stack Pointer	27
2.4.10	Manejo del Reset y de las Instrucciones.	27

2.4.11 Fuentes de reset.....	29
2.5 Programación del Microcontrolador Atmega8.	29
2.5.1 Bascom AVR.	30
2.5.1.1 Fundamentos básicos de software Bascom.	30
2.6 Pantalla de Cristal Líquido.	38
2.6.1 Mandos de control de LCD.	42
2.6.1.1 Clear Display.....	42
2.6.1.2 Cursor Home.....	42
2.6.1.3 Entry Mode Set.	42
2.6.1.4 Display On/Off Control.	42
2.6.1.5 Cursor Display Shift.....	42
2.6.1.6 Function Set.	42
2.6.1.7 Set CGRAM Address.	42
2.6.1.8 Set DDRAM Address.....	43
2.6.1.9 Read busy flag Address	43
2.6.1.10 Write data to CGRAM o DDRAM.....	43
2.6.1.11 Read data to CGRAM o DDRAM	43
CAPÍTULO III	44
3.1 Preliminares.....	44
3.2 Implementación de la fuente.....	44
3.3 Pruebas funcionales de la celda de carga.....	45
3.4 Acondicionamiento de la Celda de Carga.	47
3.4.1 Amplificación.	47
3.4.2 Conversor A/D.....	49
3.4.3 Esquema del Circuito.	50

3.5 Pruebas	51
3.5.1 Programación del μ C.....	52
CAPÍTULO IV	54
CONCLUSIONES Y RECOMENDACIONES	54
4.1 CONCLUSIONES	54
4.2 RECOMENDACIONES.....	55
GLOSARIO DE TERMINOS.....	56
Referencia Bibliográfica.....	59
ANEXOS	60

INDICE DE TABLAS

Tabla 1. Característica de la celda de carga SEN128A3B	10
Tabla 2. Descripción de pines del Puerto B	17
Tabla 3. Descripción de pines del Puerto C.....	18
Tabla 4. Descripción de pines del Puerto D.....	19
Tabla 5. Fuentes de interrupcion del Atmega 8	28
Tabla 6. Tipos de archivos generados en la Compilacion	33
Tabla 7. Opciones de la Comunicación Rs-232.	37
Tabla 8. Distribución de las Líneas del LCD.....	40
Tabla 9. Mandos de Control de LCD.....	41
Tabla 10. Medición de respuesta de la Celda de Carga.	47
Tabla 11. Pruebas del funcionamiento de la Celda de Carga con amplificación.....	52

INDICE DE FIGURAS

Figura 1. Circuito de Puente de Wheatstone	6
Figura 2. Celda de Carga.	6
Figura 3. Aplicación de una fuerza ideal en una celda de carga.	7
Figura 4. Aplicación de una Fuerza Angular en una celda de carga.	8
Figura 5. Aplicación de una Fuerza Excéntrica en una celda de carga.	9
Figura 6. Diagrama de terminales correspondientes al amplificador de instrumentación.	13
Figura 7. Esquema de pines del Atmega 8.	15
Figura 8. Arquitectura del Atmega 8.	20
Figura 9. Estructura de los registros de propósito general.	22
Figura 10. Registros X, Y, y Z.	23
Figura 11. Memoria de datos SRAM.	24
Figura 12. Registro de Estado	25
Figura 13. Stack Pointer	27
Figura 14. Pantalla Principal.	31
Figura 15. Iconos del Bascom AVR.	32
Figura 16. Opciones del Bascom AVR.	32
Figura 17. Simulador de Bascom AVR.	34
Figura 18. Pantalla de Simulación.	35
Figura 19. Pantalla de Comunicación Rs-232.	36
Figura 20: Programador AVR Usb Usbasp.	38
Figura 21: Memoria Ram del Lcd.	39
Figura 22. Diagrama de bloques de una fuente de Voltaje DC.	45
Figura 23. Fuente de Alimentación.	45
Figura 24. Prueba del funcionamiento de la celda de carga.	46
Figura 25. Pesas patrón para la comprobación de la celda de carga.	46
Figura 26. Amplificador de instrumentación AD620.	48
Figura 27. Anulación de la tensión de offset de la AD620.	49

Figura 28. Esquema de pines del Atmega8.	49
Figura 29. Diagrama del Atmega 8, pantalla LCD y amplificador de instrumentación	50

RESUMEN

El proyecto que se ha desarrollado trata acerca de los esfuerzos físicos aplicados a un tipo de transductor muy utilizado: “las galgas extensométricas” y sus respectivas mediciones y aplicaciones. Para la captura de mediciones de las galgas extensométricas, se involucra el sensado de cambios muy pequeños en su resistencia. Para esto se debe primeramente realizar el acondicionamiento de esta señal las galgas están colocadas en una celda de carga y con el uso de un puente resistivo, además un amplificador de la señal del amplificador a se podrá visualizar el valor de la fuerza aplicada a la celda de carga en un despliegue visual. El AD620 debe ser acondicionado para eliminar el voltaje de offset o voltaje de desvío que esta alrededor de los 20 a 26 mV. Así pues, es un voltaje presente en la salida del amplificador aún cuando el voltaje de la entrada sea cero. En este trabajo práctico se verá su definición formal, el principio de funcionamiento y los parámetros más importantes que se deben conocer de estos sensor, además y como medida de prueba de este transductor, se realizará el prototipo de una balanza electrónica con visualización en una pantalla LCD, para la medición en gramos, se utiliza el convertidor analógico digital de un Atmega8. El proyecto se lo realizó como medida de prueba de las galgas extensométricas, para que así, estudiantes del Laboratorio de Electrónica de Instrumentación Virtual comprendan mejor su funcionamiento. También el prototipo llenó todas las expectativas, ya que con él se apreciará una de sus funciones muy utilizadas, que es medir un peso por medio de la celda de carga.

PALABRAS CLAVES: TRANSDUCTOR, GALGAS EXTENSOMÉTRICAS, MICROCONTROLADOR.

ABSTRACT

The project that has been developed is about the physical stresses applied to a transducer type widely used "strain gauges" and their respective measurements and applications. To capture measurements of strain gauges, the sensing involves very small changes in resistance. For this must first perform conditioning of this signal gauges are placed on a load cell and using a resistive bridge addition amplifier signal amplifier may display the value of the force applied to the cell Load a visual display. The AD620 should be conditioned to remove the offset voltage and offset voltage that is around 20-26 mV. So, it is a voltage present at the amplifier output even when the input voltage is zero in this practical work will look its formal definition, the working principle and the most important parameters to meet these sensor and test as a measure of this transducer, the prototype of an electronic balance with display will take place on an LCD screen, measuring in grams, the digital analog converter of a used Atmega8. The project was made as a measure of test strain gauges, so that students of the Electronics Laboratory Virtual Instrumentation better understand its operation. Also the prototype filled all expectations, because with him one of its functions widely used, which is a weight measured by the load cell will be appreciated.

KEYWORDS: TRANSDUCER, STRAIN GAGE, MICROCONTROLLER.

CAPÍTULO I

2.3.1 ANTECEDENTES.

Debido al avance científico, tecnológico y a las exigencias de la sociedad actual, la (LOES) de organismo reguladores de educación como, “EL CES” (CONSEJO DE EDUCACIÓN SUPERIOR), CEAASES (CONSEJO DE EVALUACIÓN ACREDITACIÓN Y ASEGURAMIENTO DE LA CALIDAD DE LA EDUCACIÓN SUPERIOR) y la SENECYT (SECRETARIA NACIONAL DE EDUCACIÓN SUPERIOR, CIENCIA, TECNOLOGÍA E INNOVACIÓN).

Es de suma importancia que los centros de educación posean laboratorios y talleres que estén a la par de la tecnología de la Unidad de Gestión de Tecnologías (UGT), es fundamental que los estudiantes desarrollen habilidades, destrezas y el consiguiente aprendizaje significativo.

Por lo expuesto es necesario que la Unidad de Gestión de Tecnologías cuente con los equipos actualizados, elementos, herramientas y técnicas que vayan a la par con la tecnología actual y que faciliten el desarrollo del proceso de aprendizaje, de manera particular el manejo y utilización de los módulos de medición de peso o masa (debido a la relación que existe entre ambas magnitudes) de un objeto o sustancia mediante una celda de carga que es un sensor de pesado electrónico.

2.3.2 PLANTEAMIENTO DEL PROBLEMA.

La Unidad de Gestión de Tecnología de la Universidad de las Fuerzas Armadas – ESPE no cuenta en su laboratorio de Instrumentación Virtual con módulos de pesaje mediante celdas de carga (sensores electrónicos de pesado).

Los módulos de pesado electrónico utilizan una celda de carga como el sensor de peso, son muy importantes ya que basa su funcionamiento en el efecto piezoresistivo de una galga extensiométrica, es decir que la deformación de la estructura con lleva una variación de la resistencia y entrega una señal eléctrica muy pequeña, es necesario un amplificador para aumentar el voltaje, siendo una contribución muy significativa para el área de instrumentación.

Al no poseer este tipo de módulos actualizados en la Unidad de Gestión de Tecnologías ha dado origen a:

- Que los docentes no cuenten con el material didáctico y guías de laboratorio para impartir conocimientos de instrumentos electrónicos que miden fuerza (carga).
- Dificultad para la manipulación de los equipos electrónicos.

El desarrollo del presente proyecto pretende incentivar a directivos y alumnos de la Unidad de Gestión de Tecnologías donde la Carrera de Electrónica tiene un porcentaje mayor involucrado en el desarrollo y progreso del país a través de la industria a formar profesionales con conocimientos actualizados y personas capaces de adaptarse a la tecnología avanzada, desempeñando cualquier función a la que esté a cargo.

2.3.3 JUSTIFICACIÓN E IMPORTANCIA

La Unidad de Gestión de Tecnologías por su carácter de establecimiento de Educación Superior está obligado a contar con laboratorios debidamente equipados para que sus alumnos puedan desarrollar prácticas y adquirir habilidades, destrezas que les faciliten su inserción al ámbito laboral una vez culminada su carrera.

El proyecto está enfocado en implementar cuatro (4) módulos para el desarrollo de prácticas con celdas de carga, las cuales transmiten una señal eléctrica proporcional a la deformación, la misma que se podrá visualizar a través de un display LCD 16x2.

De esta manera los docentes de la Unidad de Gestión de Tecnologías se verán beneficiados ya que con la ayuda de las respectivas guías de laboratorio podrán capacitar a los alumnos en este tema de una forma didáctica y fácil de entender.

2.3.4 OBJETIVOS:

1.1.1 Objetivo General.

Implementar cuatro (4) módulos para el desarrollo de prácticas con celdas de carga para el laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas-ESPE.

1.1.2 Objetivos Específicos.

- Determinar la factibilidad y características técnicas del equipo.
- Acondicionar la señal de la celda de carga mediante amplificador de instrumentación.
- Realizar la calibración para anular la tensión de offset que proporciona el amplificador de instrumentación.
- Realizar una guía de laboratorio, el cual facilite el uso de los módulos de comprobación de celda de carga de forma eficiente y precisa.

2.3.5 ALCANCE.

Este proyecto está dirigido a la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías para un conocimiento más amplio teórico-práctico de los alumnos y docentes de esta carrera. La finalidad de este proyecto es el desarrollo de una balanza electrónica capaz de medir pesos. El diseño está basado en un microcontrolador Atmega8, con encapsulado de 28 pines, que es el cerebro principal. Otros periféricos que forman parte del sistema son: una celda de carga compuesta de una galga extensométricas. Por último se utiliza una pantalla de cristal líquido o LCD para mostrar todos los datos que se quieren visualizar.

El sensor utilizado, comúnmente llamado celda de carga, tiene un rango de operación de 0 – 500 gr.

CAPÍTULO II

MARCO TEÓRICO

2.1 Celda de Carga.

Una celda de carga es un transductor que convierte la fuerza en una salida eléctrica medible. Aunque hay muchas variedades de celdas de carga, celdas de carga basados en deformación son el tipo más común. Salvo para determinados laboratorios donde aún se utilizan balanzas mecánicas de precisión, células de carga de bandas extensométricas dominan la industria pesada. Células de carga neumáticos se utilizan a veces cuando se desean la seguridad y la higiene intrínseca, y células de carga hidráulicas son considerados en lugares remotos, ya que no requieren una fuente de alimentación. Celdas de carga de bandas extensométricas ofrecen precisiones desde el interior de 0,03% a 0,25% de la escala completa y son adecuadas para casi todas las aplicaciones industriales. (Omega, Load cells , 2015)

2.1.1 Principio de funcionamiento.

Diseños de celda de carga se pueden distinguir según el tipo de señal de salida generada (neumática, hidráulica, eléctrica) o de acuerdo a la forma en que detectan el peso (flexión, cizalladura, compresión, tensión, etc.)

2.1.1.1 Celda de carga hidráulicas.

Son dispositivos vigor - balance, medir el peso como un cambio en la presión del fluido de relleno interno. En una celda de carga hidráulica de laminación de tipo diafragma, una carga o fuerza que actúa sobre un cabezal de carga se transfiere a un pistón que a su vez comprime un fluido de llenado confinado dentro de una cámara de diafragma elastomérico. Como fuerza aumenta, la presión del fluido hidráulico se eleva. Esta presión se puede indicar o transmitida para indicación remota o control a nivel local. La salida es lineal y relativamente poco afectada por la cantidad del fluido de llenado o por su temperatura. Si la celda de carga se ha instalado y calibrado adecuadamente, la precisión puede estar dentro de 0,25% de la escala completa o mejor, aceptable para la mayoría de aplicaciones de procesos de

pesaje. Debido a este sensor no tiene componentes eléctricos, es ideal para su uso en áreas peligrosas. Aplicaciones de celdas de carga hidráulicos típicos incluyen tanque Bin, y la tolva de pesaje. Para obtener la máxima precisión, el peso del depósito debe obtenerse mediante la localización de una celda de carga en cada punto de apoyo y sumando sus salidas.

2.1.1.2 Celda de carga neumática.

También operan sobre el principio de equilibrio de fuerzas. Estos dispositivos utilizan múltiples cámaras amortiguador de proporcionar una mayor precisión que puede un dispositivo hidráulico. En algunos diseños, la primera cámara de amortiguador se utiliza como una cámara de peso de tara. Celdas de carga neumáticos se utilizan a menudo para medir relativamente pequeñas pesas en industrias donde la limpieza y la seguridad son de suma preocupación. Las ventajas de este tipo de celda de carga incluyen su explosión ser inherentemente prueba e insensible a las variaciones de temperatura. Además, no contienen los líquidos que puedan contaminar el proceso si se rompe el diafragma. Las desventajas incluyen la velocidad relativamente lenta de la respuesta y la necesidad de, aire limpio y seco regulada o nitrógeno.

2.1.1.3 Celda de carga Strain-Guage.

Convierten la carga que actúa sobre ellos en señales eléctricas. Los medidores propios están unidos en una viga o elemento estructural que se deforma cuando se aplica peso. En la mayoría de los casos, cuatro calibradores de tensión se utilizan para obtener la máxima sensibilidad y compensación de temperatura. Dos de los medidores son por lo general en tensión, y dos en la compresión, y están cableados con ajustes de compensación, como se muestra en la (Figura 1). Cuando se aplica el peso, la cepa cambia la resistencia eléctrica de las galgas en proporción a la carga. Otras celda de carga están desvaneciendo en el olvido, como las celda de carga de bandas extensométricas continúan aumentando su precisión y reducir sus costos unitarios. (Omega, Load cells , 2015)

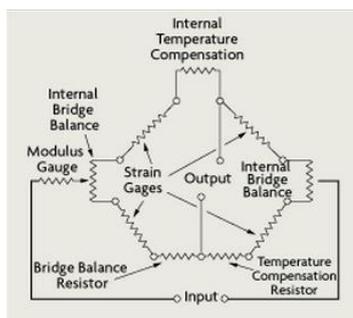


Figura 1. Circuito de Puente de Wheatstone.

Fuente: (Omega, load-cells, 2015)

2.1.2 Impedancia de la celda de carga.

Para identificar y distinguir excitaciones y señales, las empresas fabricantes utilizan un código de calor en el cableado de las celdas de carga, aunque dependiendo de la procedencia, en algunos modelos, el cable azul es cambiado por un negro, y el amarillo por un blanco (figura 2). (VITERY, 2011)

En nuestro caso, la celda usada es de procedencia China, por lo tanto los colores de los cables son: rojo (+), negro (GND), para la polarización; y los de señal son blanco (+) y azul (-). (SIMBAÑA, 2009)

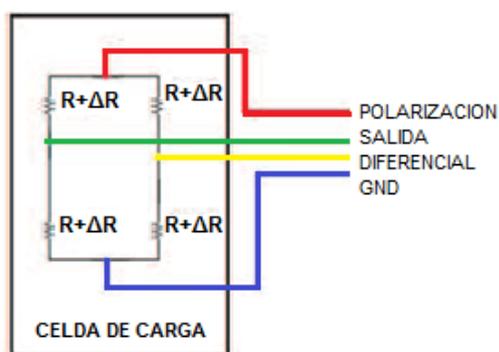


Figura 2. Celda de Carga.

Fuente: (SIMBAÑA, 2009)

Existen casos donde ninguno color corresponde a un código conocido, en este caso se puede utilizar el método de las impedancias, que consiste en medir cada con su compañero y tomar notas de las impedancias dadas, dependiendo de los fabricantes las celdas tendrán impedancias de 350Ω y

700 Ω u otras, luego se aplica los siguientes criterios para conocer el estado de la celda.

- El par de cables con mayor impedancia son la excitación.
- El par de cable restante son las señales.
- Ente excitación positiva y cada una de las señales debe haber la misma impedancia $\pm 2 \Omega$.
- Entre excitación negativa y cada una de las señales debe haber la misma impedancia $\pm 2 \Omega$ máximo.

Si cualquiera de estas condiciones no se cumplen la celda está dañada.

2.1.3 Aplicación de la celda de carga.

El entendimiento de la forma en que una carga o fuerza debe ser aplicada a la celda de carga es de vital importancia para el buen diseño de una balanza o sistema de peso. (VITERY, 2011)

2.1.4 Selección de la celda de carga.

El entendimiento de la forma exacta, en que una carga o fuerza debe ser aplicada a la celda de carga es de vital importancia para el buen diseño de una balanza o sistema de peso en tanques, tolvas y silos, permitiendo además una correcta selección del modelo de celda de carga adecuado para su aplicación. (SIMBAÑA, 2009)

2.1.4.1 Celda de carga Ideal.

Las especificaciones técnicas de las celdas de cargas han sido determinadas bajo condiciones de laboratorio, aplicando la carga o fuerza a la celda bajo condiciones lo más cercanas posibles a la perfección.

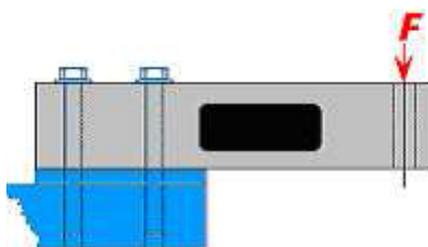


Figura 3. Aplicación de una fuerza ideal en una celda de carga.

Fuente: (VITERY, 2011)

En la (Figura 3) vemos una simple aplicación con una celda de carga de tipo viga de flexión. Uno de los lados está firmemente fijado a una base rígida, con el extremo opuesto libre para flexionar, de acuerdo al peso o carga aplicada. Bajo condiciones ideales, la superficie será perfectamente plana, horizontal y totalmente rígida.

La carga **F** se introduce en forma vertical con un mínimo de fuerzas extrañas aplicadas. Las celdas de carga están preparadas para ser insensibles en lo posible a todas las fuerzas distintas a la vertical.

Lamentablemente, en el mundo real, el montaje de las celdas y las condiciones de aplicaciones de fuerzas están generalmente alejados del ideal. La incorrecta aplicación de fuerzas es muchas veces la causa en problemas de precisión de una balanza. (VITERY, 2011)

2.1.4.2 Fuerzas Angulares.

Esta es una condición que la **F** es introducida al orificio de recepción de carga en un ángulo en la relación con el eje central. Esta fuerza se anula si el componente angular llega a los 90° en relación con el eje central. Este componente horizontal es una fuerza lateral a la que la celda de carga es totalmente insensible (figura 4).

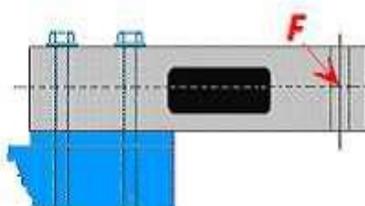


Figura 4. Aplicación de una Fuerza Angular en una celda de carga.

Fuente: (VITERY, 2011)

Si esta dirección de la fuerza aplicada es constante, la calibración compensará este error y la balanza será precisa. En cambio, si este ángulo varía al aplicar el peso, se producirá falta de linealidad en la balanza, fricciones en el sistema mecánico y errores por histéresis.

Las fuerzas angulares se pueden producir por accesorios de montaje no alineados debidamente, una base poco rígida, expansión o contracción térmica, deflexión al aplicar la carga, etc. (VITERY, 2011)

2.1.4.3 Cargas Excéntricas.

Esta es una condición en que la fuerza es aplicada a la celda en forma vertical, pero la línea del eje de acción está apartada del eje del orificio de carga de la celda de carga (figura 5).

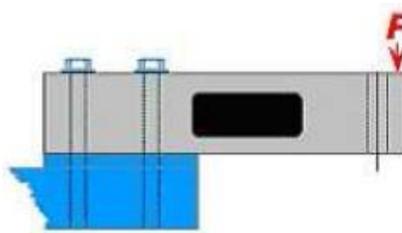


Figura 5. Aplicación de una Fuerza Excéntrica en una celda de carga.

Fuente: (VITERY, 2011)

Esta condición no afectará el normal funcionamiento de la balanza o sistema de pesaje si la posición es constante, pues al calibrar se compensará el error, en cambio, si este punto de aplicación varía durante el funcionamiento del sistema, se producirán errores de linealidad e histéresis.

Las cargas excéntricas se producen generalmente por accesorios de montajes defectuosos o mal diseñados y por expansión o contracción por acción térmica. (VITERY, 2011)

2.1.4.4 Tipos de celda de carga.

Las celdas que reciben la fuerza por compresión y por cizalla dura se utilizan en la medición de pesos grandes, mientras que las celdas que reciben la fuerza en forma de tensión se utilizan para la medición de pesos pequeños. Es necesario saber si el material a pesarse se colgará de una estructura o se apoyará sobre la misma previo a la selección de una celda.

2.1.4.5 Parámetros comerciales.

Los parámetros más utilizados para la elección de una celda de carga son el voltaje de excitación y la resolución. La capacidad nos indica el peso máximo al cual se puede someter la celda, por ejemplo 2000 Kg. La resolución indica la señal que la celda entrega a plena carga por voltaje de excitación, por ejemplo 2mV/V. La fórmula general usada para determinar la

señal (en milivoltios) entregada por una celda de carga a determinado esfuerzo o peso es la siguiente. (SIMBAÑA, 2009)

$$\text{Señal} = \frac{\text{Carga} \times \text{Res} \times \text{Vexc}}{\text{Capacidad}}$$

Dónde:

Carga= esfuerzo en Kg a la que la celda es sometida.

Res= resolución en mV/V

Vexc= voltaje con el que se alimenta la celda.

Capacidad= carga máxima soportada por la celda en Kg. Para el caso de este proyecto, la celda de carga utilizada es de 500gr=0.5Kg de capacidad con salida de 0.5 mV/V, se excita con 5V. Suponiendo que se le aplica 80g; aplicando la ecuación la señal resultante sería. (SIMBAÑA, 2009)

$$\text{Señal} = \frac{80g \times 0.5 \frac{mV}{V} \times 5V}{500g} = 0.4mV$$

La celda de carga utilizada en este proyecto, tiene las siguientes características proporcionadas por el fabricante.

Tabla 1

Característica de la celda de carga [SEN128A3B](#)

Modelo	SEN128A3B
Capacidad gr	500 gr
Sensibilidad mv salida	0.5±0.1
No linealidad	%F.S 0.05
Histéresis	%F.S 0.05
Repetibilidad	%F.S 0.05

CONTINUA 

Fluencia (30min) variación de la señal de salida	%F.S 0.05
Efecto de la temperatura sobre la sensibilidad%	%F.S/10°C 0.05
Efecto de la temperatura sobre cero%	%F.S/10°C 0.05
Balance Cero%	%F.S ±0.5
Ω Resistencia de entrada (ohmios)	1120±10
Ω Resistencia de salida (ohmios)	1000 ± 10
Resistencia de aislamiento MΩ (ohmios)	≥2000
Excitación recomendada tensión v	5v
Código de color	+ excitación = rojo, + sig = blanco - excitación = negro, -sig = azul
Dimensión(mm)	45x9x6

Fuente: (Seeed Technology Limited, 2008)

2.2 Amplificadores de Instrumentación.

El Amplificador de instrumentación es uno de los circuitos más útiles, precisos y versátiles disponibles en la actualidad, pues en general tiene que amplificar señales muy débiles (mV) de un dispositivo sensor (temperatura, presión, señal eléctrica, etc.) en un ambiente ruidoso (modo común) del orden de voltios. Por ello, los amplificadores de instrumentación tienen que tener una serie de características muy especiales.

- a) Buena CMRR (100dB).
- b) Bajo nivel de ruido.
- c) Alta impedancia de entrada (para no cargar el sensor).
- d) Ganancia ajustable mediante una resistencia.
- e) Pequeñas tensiones y corrientes de desviación.
- f) Bajas derivas.

2.2.1 AD620

El AD620, el cual es un integrado de bajo costo y funciona como amplificador de instrumentación de alta precisión, es amplificador que sólo requiere una resistencia externa para establecer las ganancias de 1 a 1000, el fabricante determina la siguiente formula.

$$RG = \frac{49.4 K\Omega}{G + 1}$$

Además, este dispositivo se puede adquirir en los encapsulados SOIC o DIP de 8 terminales, como se aprecia en la (figura 6). Consume menor potencia y es considerado como un buen elemento para aplicaciones portátiles ya que puede ser alimentado con baterías, además con su alta precisión de 40 ppm, y un offset de 0.6 $\mu\text{V}/^\circ\text{C}$ máx., es ideal para el uso en sistemas de adquisición de datos con precisión, tales como basculas y las interfaces de transductores.

El bajo nivel de ruido, la baja corriente de entrada de polarización y baja potencia hace adecuado para aplicaciones médicas ideal para instrumentación médica, pues trabaja con bajo nivel de ruido y sirve para el acondicionamiento de señales con niveles de voltaje del orden milivoltios. Además aporta características importantes al diseño como alta impedancia de entrada y alto rechazo al modo común (CMRR). El AD620 es idóneo como amplificador ya que cuenta con una etapa de procesamiento, llamada superbeta, de modo que le es posible trabajar con corrientes de polarización muy bajas, en los niveles de nanoamperios.

2.2.1.1 Características.

- Amplificador de bajo ruido
- Opera con voltajes BIPOLARES de ± 2.3 a ± 18 voltios
- Disponible en empaquetado de 8 pines
- Consumo de 1.3 mA
- El desempeño en DC es excelente ya que solo tiene un máximo de 50 μV de offset.
- Desvió máximo de 0.6 $\mu\text{V}/^\circ\text{C}$

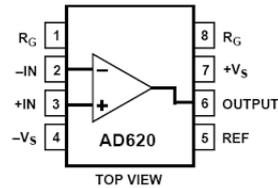


Figura 6. Diagrama de terminales correspondientes al amplificador de instrumentación.

Fuente: (DEVICES, 2015)

2.2.1.2 Aplicaciones.

- Medición de peso
- ECG e instrumentación médica
- Interfaces de transductores
- Sistemas de adquisición de datos
- Controles en procesos industriales
- Equipos portables y operados a baterías
- Entre otros

2.4 Microcontrolador Atmega 8.

2.4.1 Características del Atmega 8.

El Atmega 8 es un microcontrolador de la familia Atmel, contiene una arquitectura tipo RISC, sus instrucciones se ejecutan en solo un ciclo de máquina, su estructura elimina la necesidad de usar componentes externos ya que internamente tiene: osciladores, USART, SPI, resistencias pull-up, modulación por anchura de pulso (PWM), convertidor ADC, comparador analógico y cronómetros, éstas son solo algunas de las características que se encuentran en este dispositivo. Su programación es posible realizarla en Lenguaje C, assembly o Basic. Todo su sistema está depositado en una sola pastilla. “El Atmega 8 tiene las siguientes características.

- Utiliza arquitectura RISC.
- 130 instrucciones, las más simples se ejecutan en un solo ciclo de reloj.
- 8 Kbytes de Memoria Flash.
- ISP: Programable en sistema.
- Capacidad Read-While-Write, (Lee mientras escribe). Duración: 1.000. ciclos de escritura/borrado.

- 512 bytes de memoria EEPROM.
- Duración: 100.000 ciclos de escritura/lectura.
- 1 Kbyte de memoria SRAM.
- Seguro programable para la seguridad del software.
- 32 x 8 registros de trabajo de propósito general.
- 23 líneas de entrada/salida de propósito general.
- USART (Universal Synchronous Asynchronous Receiver Transmitter) programable.
- Reloj en tiempo real.
- Interfaz serial SPI.
- Vcc: 4.5 – 5.5 V.
- Frecuencia de funcionamiento 0 – 16 MHz.
- 3 timer/counter.
- Interrupciones internas y externas.
- Interfaz serial 2-wire.
- 6 canales ADC, donde 4 tienen 10 bits de exactitud, y 2 tienen 8 bits de exactitud.
- 1 Timer Watchdog programable con oscilador interno.
- 5 modos de ahorro de energía, seleccionado por software”.1
- “El modo **Idle** (ocioso) detiene la CPU, mientras permite que la memoria SRAM, los timer/counters, el puerto SPI, e interrupciones del sistema continúen funcionando.
- Modo **Power-down** (baja energía), guarda el contenido del registro, pero congela el oscilador, inhabilitando el resto de funciones del chip hasta la siguiente interrupción o hasta que se realice un reset del hardware.
- En el **Power save** (modo economizador), el contador de tiempo asincrónico continúa funcionando, permitiendo al usuario mantener una base de tiempo, mientras el resto de dispositivos continúa en reposo.
- En el modo de reducción del nivel de ruidos ADC, detiene la CPU y todos los módulos de entrada y salida excepto el contador asincrónico y el Conversor Análogo-Digital (ADC), para reducir al mínimo el ruido de conmutación durante la conversión ADC.

- En **standby** (modo de espera) el oscilador cristal/resonador sigue funcionando, mientras el resto de dispositivos sigue en reposo. Esto permite el reinicio muy rápido combinado con el consumo bajo de energía.

El microcontrolador AVR tiene y combina un amplio sistema de instrucciones, con 32 registros de propósito general. Las operaciones de la ALU entre registros son ejecutadas en el archivo de registro. Las operaciones son divididas en tres categorías: aritméticas, lógicas y funciones de bit. La arquitectura que resulta es más eficiente, y alcanza rendimientos de procesamiento de hasta diez veces más rápido que los microcontroladores con tecnología CISC.

La Memoria Flash ISP (Programable en sistema) permite que la memoria de programa del chip sea reprogramada en el propio sistema a través de la interfaz SPI o mediante un programador de memoria convencional. Combinando una tecnología RISC de 8 bits con un CPU con memoria Flash, el Atmega 8 resulta un poderoso microcontrolador que proporciona una alta flexibilidad en los diseños, a bajo costo aportando una solución muy efectiva para la mayoría de las aplicaciones de control.

El Atmega 8 está apoyado por un completo juego de programas y sistemas de desarrollo, incluyendo: compiladores, ensambladores, simuladores, emuladores en circuito, y kits de evaluación.

2.4.2 Descripción de Pines.

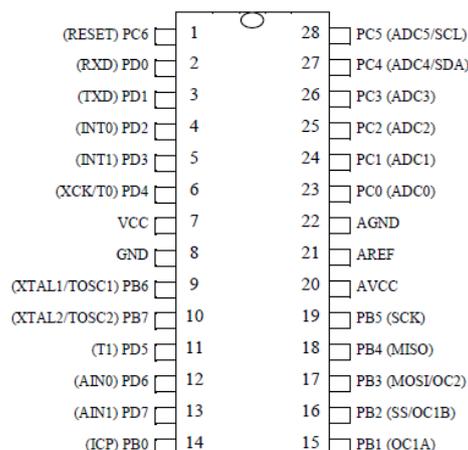


Figura 7. Esquema de pines del Atmega 8.

Fuente: (GORDON, 2006)

➤ **Vcc**

Fuente de voltaje digital

➤ **GND**

Tierra de la fuente de voltaje digital

➤ **RESET**

Entrada de RESET. Se genera un reset aplicando un nivel bajo de voltaje en este pin por un periodo de tiempo más largo que un ciclo de máquina, incluso si el reloj no está funcionando. Pulsos más cortos no generan un estado de reset.

➤ **XTAL 1**

Entrada al amplificador inversor del oscilador y entrada al circuito de operación interno del reloj.

➤ **XTAL2**

Salida del amplificador inversor del oscilador

➤ **AVCC**

Este es un pin de fuente de voltaje para el Puerto A y sus ADCs (Conversores Analógico-Digitales). Si sus ADCs no son utilizados, este pin debe ser conectado a Vcc. Si sus ADCs sí son utilizados, este pin debe ser conectado a Vcc pero mediante un filtro pasa-bajo.

➤ **AGND**

Tierra analógica si el tablero presenta diferentes tierras, se debe conectarlas a este pin. En otro caso se conecta a GND

➤ **AREF**

Este es el pin de referencia analógica para los ADC.

➤ **Puerto B / XTAL 1 / XTAL 2 / TOSC 1 / TOSC 2**

“El puerto B es un puerto de entrada/salida bi-direccional de 8 bits, con resistencias internas de pull-up. Como salidas, los pines del puerto B tienen limitada la corriente de salida, si las resistencias pull-up están activadas. Los pines del puerto B tienen tres estados cuando la condición de reset llega a activarse, incluso si el reloj no está funcionando”.

Dependiendo de la configuración de las opciones de reloj, el pin PB6 puede ser usado como entrada al amplificador inversor del oscilador y entrada al circuito de operación interno del reloj.

Dependiendo de la configuración de las opciones de reloj, el pin PB7 puede ser usado como salida del amplificador inversor del oscilador

Los pines del puerto B tienen también algunas funciones especiales, que se las resume en el siguiente cuadro. (GORDON, 2006)

Tabla 2

Descripción de pines del Puerto B

PIN DEL PUERTO B	FUNCION ALTERNA
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
PB5	SCK (SPI Bus Master clock Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2 (Timer/Counter2 Output Compare Match Output)
PB2	SS (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output)
PB0	ICP (Timer/Counter1 Input Capture Input)

Fuente: (GORDON, 2006)

➤ **Pórtico C / RESET**

“El puerto C es un puerto de entrada/salida bi-direccional de 8 bits, con resistencias internas de pull-up. Como salidas, los pines del puerto C tienen limitada la corriente de salida, si las resistencias pull-up están activadas.

Los pines del puerto C tienen tres estados cuando la condición de reset llega a activarse, incluso si el reloj no está funcionando”

PC6 es usado como entrada de Reset. Un nivel bajo de voltaje en este pin por un periodo de tiempo aproximado de 1.5 μ s generará un reset, incluso si

el reloj no está funcionando. Pulsos más cortos no generan un estado de reset.

Los pines del puerto C tienen también algunas funciones especiales, que se las resume en el siguiente cuadro.

Tabla 3

Descripción de pines del Puerto C.

PIN DEL PUERTO C	FUNCION ALTERNA
PC6	RESET (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

Fuente: (GORDON, 2006)

➤ Puerto D

“El puerto D es un puerto de entrada/salida bi-direccional de 8 bits, con resistencias internas de pull-up. Como salidas, los pines del puerto D tienen limitada la corriente de salida, si las resistencias pull-up están activadas. Los pines del puerto D tienen tres estados cuando la condición de reset llega a activarse, incluso si el reloj no está funcionando”.

Los pines del puerto D tienen también algunas funciones especiales, que se los resume en la siguiente tabla.

Tabla 4

Descripción de pines del Puerto D.

PIN DEL PUERTO D	FUNCION ALTERNA
PD7	AIN1 (Analog Comparator Negative Input)
PD6	AIN0 (Analog Comparator Positive Input)
PD5	T1 (Timer/Counter 1 External Counter Input)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

Fuente: (GORDON, 2006)

2.3.3 Arquitectura del Atmega 8

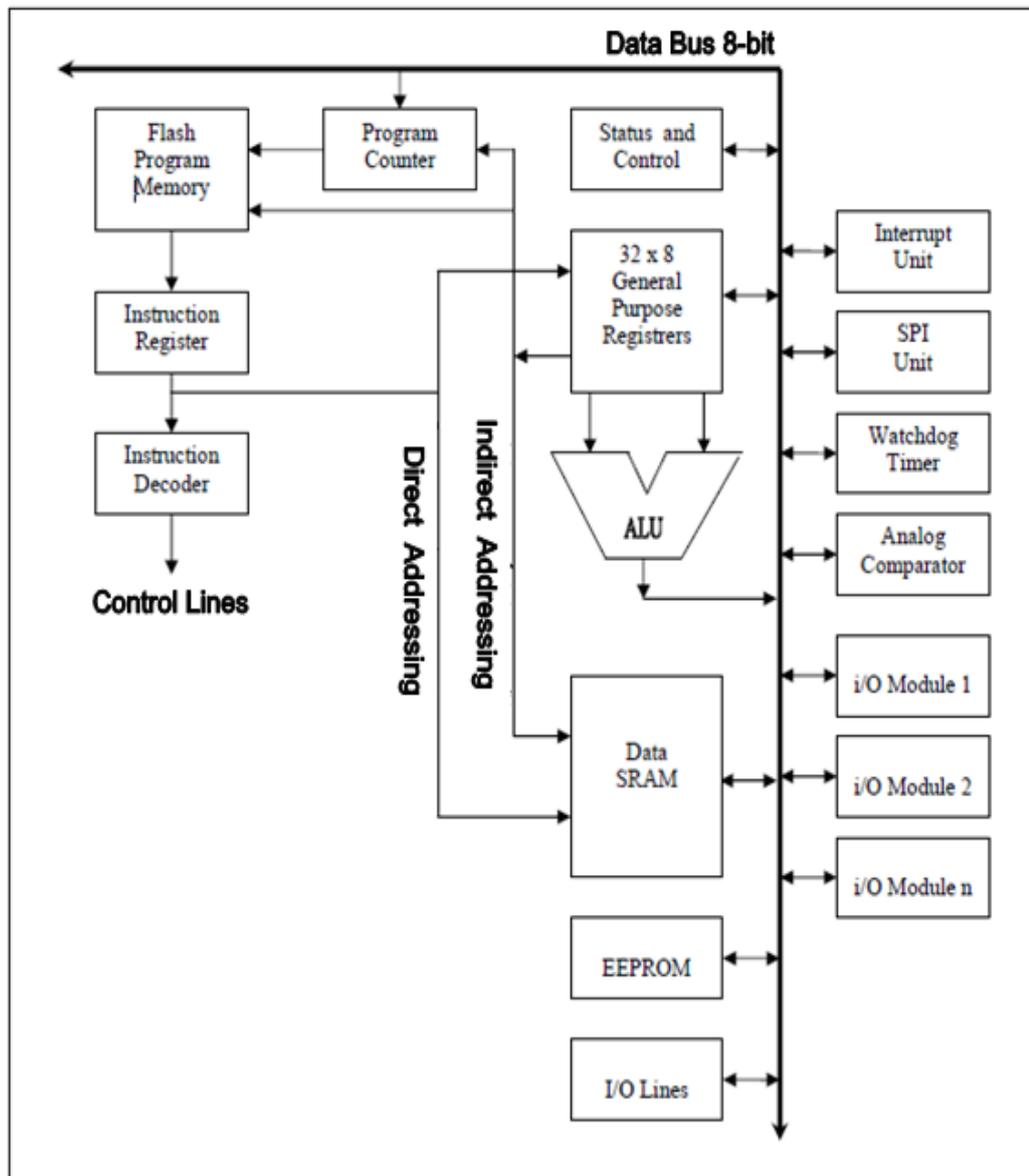


Figura 8. Arquitectura del Atmega 8.

Fuente: (GORDON, 2006)

El rápido acceso a los registros, se basa en el concepto de que los 32 registros de 8 bits, son accedidos en un solo ciclo de reloj. Lo cual significa que, durante un ciclo de reloj es ejecutada una operación en la Unidad Aritmética Lógica (ALU).

Dos operandos son tomados del archivo de registros, las operaciones son ejecutadas y el resultado es colocado en el mismo archivo, todo en un ciclo de reloj.

“Seis (6) de los 32 registros de trabajo pueden usarse como 3 registros de 16 bits que sirven como punteros de direccionamiento indirecto para

direccionar el espacio de datos, lo cual permite un eficiente cálculo de direcciones. Estos registros con funciones especiales son el registro X, el Y y el Z, todos de 16 bits y ocupan las direcciones de memoria: 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F respectivamente”.

La ALU soporta operaciones aritméticas y lógicas entre registros o entre un registro y una constante. Las operaciones simples con registros son también ejecutadas por la ALU. Después de una operación aritmética el registro de estado se carga con la información del resultado de la operación.

Los registros de trabajo están colocados en las 32 primeras posiciones de la memoria de datos (\$00 - \$1F), permitiendo que su acceso sea como si fueran lugares de memoria convencionales.

El espacio de memoria contiene 64 direcciones para las funciones propias de la CPU como son los registros de Control, SPI, los Contadores/Timer, conversores A/D, y otras funciones. Estos registros están colocados a continuación de los de trabajo, de la dirección 0x20 a la 0x5F.

Para maximizar el funcionamiento, el AVR utiliza la arquitectura Harvard con memorias y buses separados para el programa y para los datos. Mientras que se está ejecutando una instrucción, la siguiente instrucción es buscada por la memoria del programa. Este concepto permite que las instrucciones sean ejecutadas en cada ciclo de reloj. La memoria del programa es una memoria flash re-programable en sistema (ISP).

Con las instrucciones de salto incondicional y con la de llamada, se puede acceder directamente a toda la memoria. La mayoría de las instrucciones AVR tienen una longitud de 16 bits. Cada dirección de la memoria de programa contiene una instrucción de 16 ó de 32 bits.

Durante las interrupciones y las llamadas a subrutinas, el contador de programa (PC) guarda las direcciones anteriores en el stack. El stack se encuentra alojado en los datos generales de la SRAM, y consecuentemente, el tamaño del mismo es únicamente limitado por el tamaño total de la SRAM y de su uso. Todos los programas del usuario deben inicializar el SP (Stack Pointer) en la rutina de reset (antes de ejecutar subrutinas o interrupciones). El stack pointer (SP) de 10 bits es seleccionado para lectura/escritura en el espacio de entrada/salida.

El 1 Kbyte de datos de SRAM pueden ser fácilmente accedidos mediante los cinco modos diferentes de direccionamiento que soporta la arquitectura AVR. Los espacios de memoria, en la arquitectura del AVR, son todos lineales y regulares.

Un módulo flexible de interrupción, tiene sus registros de control en el espacio de entrada/salida con un bit de habilitación de interrupción global en el registro de estado. Todas las diferentes interrupciones tienen un vector separado de interrupción en la tabla de vectores de interrupción al inicio de la memoria de programa. Las diferentes interrupciones tienen prioridad en concordancia con la posición del vector de interrupción. El vector de dirección más bajo es el que tiene la más alta prioridad.

2.4.3 Registros De Propósito General.

La (figura 9) muestra la estructura de los 32 registros de trabajo de propósito general presentes en el Atmega 8.

		7	0	Dirección	
REGISTROS DE PROPOSITO GENERAL	R0			0x00	
	R1			0x01	
	R2			0x02	
	...				
	R13			0x0D	
	R14			0x0E	
	R15			0x0F	
	R16			0x10	
	R17			0x11	
	...				
	R26			0x1A	Registro X byte bajo
	R27			0x1B	Registro X byte alto
	R28			0x1C	Registro Y byte bajo
	R29			0x1D	Registro Y byte alto
	R30			0x1E	Registro Z byte bajo
	R31			0x1F	Registro Z byte alto

Figura 9. Estructura de los registros de propósito general.

Fuente: (GORDON, 2006)

La mayoría de las instrucciones que funcionan en el archivo de registros tienen acceso directo a todos los registros, y la mayoría de ellos son instrucciones de un solo ciclo de máquina ciclo.

Cada registro es asignado a una dirección de memoria de datos, dirigiéndolos directamente dentro de las 32 primeras localidades del espacio de datos del usuario, aunque no sea físicamente implementada como localidad SRAM, esta organización de memoria permite gran flexibilidad en el acceso a los registros, puesto que los registros X, Y y Z pueden ser sesteados para indexar cualquier registro del archivo.

2.4.3.1 Registro X, registro Y y registro Z.

Los registros R26....R31 poseen algunas funciones adicionales. Estos registros son punteros de dirección para el direccionamiento indirecto del Espacio de Datos. Estos registros se encuentran definidos como se muestra en la (figura 10).

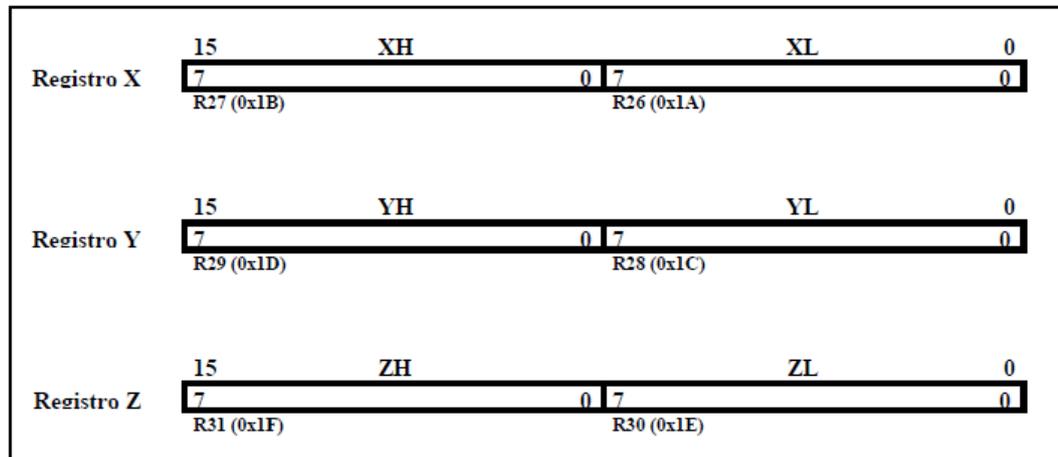


Figura 10. Registros X, Y, y Z

Fuente: (GORDON, 2006)

En los diferentes modos de direccionamiento, estos registros de dirección tienen funciones como realizar los desplazamientos, incrementos automáticos y decrementos automáticos.

2.4.4 Unidad aritmética lógica.

“La ALU de alto rendimiento trabaja en conexión directa con todos los 32 registros de trabajo de propósito general. Las operaciones aritméticas entre los registros de propósito general o entre un registro y un dato inmediato se ejecutan dentro de un solo ciclo de reloj. Las operaciones de la ALU se dividen en tres categorías principales - aritmética, lógico, y bit-funciones. Algunas implementaciones de la arquitectura también proporcionan un poderoso multiplicador que soporta multiplicaciones con signo, sin signo y formato fraccionario”.

2.4.5 Flash Programmable.

El Atmega 8 contiene 8 Kbyte de memoria. Como todas las instrucciones son de 16 o 32 bit, la memoria flash está organizada como 4 x 16. La memoria flash tiene una durabilidad de 1000 ciclos de escritura/borrado. El

contador de programa (PC) es de 12 bits, con lo que tenemos 4096 direcciones de la memoria de programa.

2.4.6 Memoria de Datos SRAM

Registers File		Data Address Space
R0		\$0000
R1		\$0001
R2		\$0002
....		
R29		\$001D
R30		\$001E
R31		\$001F
I/O Registers		
\$00		\$0020
\$01		\$0021
\$02		\$0022
....		
\$3D		\$005D
\$3E		\$005E
\$3F		\$005F

Internal SRAM	
	\$0060
	\$0061
	...
	\$045E
	\$045F

Figura 11. Memoria de datos SRAM.

Fuente: (GORDON, 2006)

La (figura 11) muestra como está organizada la memoria SRAM. “Las 1120 posiciones de memoria más bajas contienen los registros de propósito general, los registros de entrada/salida, y los datos internos SRAM. Las primeras 96 localizaciones contienen los registros de propósito general y los registros de entrada/salida, y las 1024 localizaciones siguientes contienen los datos internos SRAM”.

Los cinco diferentes modos de direccionamiento de memoria son: Directo, Indirecto con desplazamiento, Indirecto, Indirecto con pre-decremento, e Indirecto con post-incremento.

El direccionamiento directo alcanza todo el espacio de datos. El modo indirecto con desplazamiento alcanza 63 direcciones de localidad, de la base de direcciones dadas por el registro Y o Z. Cuando se usa el modo de direccionamiento indirecto con pre-decremento y post incremento automático, las direcciones de los registros X, Y y Z son decrementadas e incrementadas. Los 32 registros de trabajo, los 23 registros I/O y el 1 Kbyte

de datos internos de SRAM en el Atmega 8 son todos accesibles mediante estos modos de direccionamiento. (GORDON, 2006)

2.4.7 Memoria de Datos EEPROM.

El Atmega 8 contiene 512 bytes de memoria de datos EEPROM. La misma es organizada como espacios de datos separados. La EEPROM tiene una duración de aproximadamente 100.000 ciclos de lectura/escritura.

2.4.8 Registro de Estado.

El registro de estado contiene la información sobre el resultado de la última instrucción aritmética ejecutada. Esta información se puede utilizar para alterar el normal desarrollo del programa para realizar operaciones condicionales. El registro de estado es actualizado después de todas las operaciones de la ALU, según lo especificado en el set de instrucciones, dando por resultado un código 9 Microcontroller with 8 Kbytes In-System Programmable Flash, Advance Information, Atmel Corporation, 2001, 15. 27 más rápido y más compacto. El registro de estado no se almacena al incorporar una rutina de la interrupción y no se restaura automáticamente al volver de una interrupción. Esto se debe dirigir por software.

El registro de estado se denomina SREG y se encuentra en la dirección \$3F. La descripción de los bits es la siguiente. (Figura 12)

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

Figura 12. Registro de Estado

Fuente: (GORDON, 2006)

Todos los bits de este registro son de escritura o lectura. El valor inicial de este registro es 0x00. La descripción de los bits es la siguiente.

2.4.8.1 Bit 7-I: Global Interrupt Enable.

Este bit es el que permite habilitar las interrupciones cuando está a 1. El control individual de las interrupciones es configurado en los registros GIMSK y TIMSK.

Si este bit está a 0 todas las interrupciones están deshabilitadas independientemente del estado de los registros GIMSK y TIMSK. Este bit es puesto a 0 por hardware en el momento en el que se produce la interrupción y es puesto de nuevo a uno tras la instrucción RETI que sirve para producir el retorno de interrupción.

2.4.8.2 Bit 6-T: Bit Copy Storage.

Las instrucciones para copiar bits BLD (Bit Load) y BST (Bit Store) utilizan el bit T como fuente o como destino para un bit. Un bit de un registro del archivo de registros puede ser guardado en el bit T mediante la instrucción BST, y un bit en T puede ser copiado en un bit de un registro del archivo de registros mediante la instrucción BLD.

2.4.8.3 Bit 5-H: Half Carry Flag.

Este bit indica que se ha producido un half carry en algunas operaciones aritméticas.

2.4.8.4 Bit 4-S: Sign Bit, $S = S (+) V$.

El bit-S realiza una operación or exclusiva entre los bits N y el bit V del registro de estado.

2.4.8.5 Bit 3-V: Two's Complement Overflow Flag.

Esta bandera V, soporta aritmética en complemento de 2.

2.4.8.6 Bit 2-N: Negative Flag.

La bandera negativa N indica un resultado negativo tras realizar una operación aritmética o una lógica.

2.4.8.7 Bit 1-Z: Zero Flag.

Esta bandera indica un resultado de cero tras realizar una operación aritmética o una lógica.

2.4.8.8 Bit 0-C: Carry Flag.

Esta bandera indica un carry tras realizar una operación aritmética o una lógica.

2.4.9 Stack Pointer

“El stack pointer se utiliza principalmente para almacenar datos temporales, para almacenar variables locales y para guardar la dirección de retorno después de las interrupciones y las llamadas a las subrutinas”.

(Figura 13)

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Figura 13. Stack Pointer

Fuente: (GORDON, 2006)

Todos los bits son de lectura y escritura y en estado inicial los 16 bits están a cero.

Cuando las subrutinas y las interrupciones se producen, el Stack Pointer coloca los datos en una zona de la memoria SRAM. Este espacio para los datos en la memoria SRAM tiene que ser definido por el programa antes de que cualquier llamada a una subrutina o cualquier interrupción sea ejecutada. El Stack Pointer es decrementado en una unidad cuando un dato es colocado en el Stack mediante la instrucción PUSH y es decrementado en dos unidades cuando un dato es puesto en el Stack mediante una llamada a subrutina (CALL) o se produce una interrupción. El Stack Pointer es incrementado en una unidad cuando un dato es recuperado del Stack mediante la instrucción POP, y es incrementado en dos unidades cuando un dato es devuelto del Stack mediante un retorno de subrutina con la instrucción RET o por un retorno de una interrupción RET.

2.4.10 Manejo del Reset y de las Instrucciones.

El microcontrolador ATMEGA 8 posee 19 fuentes de interrupción, todas las interrupciones tienen un vector de reset propio en el espacio de memoria del programa. Todas las interrupciones tienen asignados unos bits que deben ser puestos a uno conjuntamente con el bit I del registro de estado para que las interrupciones puedan producirse.

Las direcciones más bajas del espacio de memoria de programa son automáticamente definidas para el vector de Reset y para los vectores de interrupción. La lista completa de vectores se muestra en la siguiente tabla. Esta lista también determina los niveles de prioridad de las diferentes interrupciones.

La mayor prioridad tiene el vector de RESET y a continuación el vector INT0 y así sucesivamente.

Tabla 5

Fuentes de interrupción del Atmega 8.

Vector N ^o	Dirección	Fuente	Definición de la Interrupción
1	0x000	RESET	Cuatro fuentes de RESET(*)
2	0x001	INT0	Interrupción Externa Requerida 0
3	0x002	INT1	Interrupción Externa Requerida 1
4	0x003	TIMER2 COMP	Temporizador/Contador 2 Compare Match
5	0x004	TIMER2 OVF	Temporizador/Contador 2 Sobreflujo
6	0x005	TIMER1 CAPT	Temporizador/Contador 1 Evento de captura
7	0x006	TIMER1 COMPA	Temporizador/Contador 1 Compare Match A
8	0x007	TIMER1 COMPB	Temporizador/Contador 1 Compare Match B
9	0x008	TIMER1 OVF	Temporizador/Contador 1 Sobreflujo
10	0x009	TIMER0 OVF	Temporizador/Contador 0 Sobreflujo
11	0x00A	SPI, STC	Transferencia Serial Completa
12	0x00B	USART, RXC	Recepción Completa USART
13	0x00C	USART, UDRE	Registro de datos vacío USART
14	0x00D	USART, TXC	Transmisión Completa USART
15	0x00E	ADC	Conversión Completa ADC
16	0x00F	EE_RDY	Memoria EEPROM lista
17	0x010	ANA_COMP	Comparador Análogo
18	0x011	TW1	Interfaz serial 2-wire
19	0x012	SPM_RDY	Almacene Memoria de Programa Lista

Fuente: (GORDON, 2006)

2.4.11 Fuentes de reset.

Existen cuatro posibilidades para que se produzca un Reset:

- **Reset al encendido.-** El microcontrolador se resetea cuando el voltaje de la fuente de alimentación está por debajo del nivel requerido para el funcionamiento.
- **Reset externo.-** Cuando un nivel bajo es ingresado en el pin RESET durante más de dos ciclos de XTAL.
- **Reset por Watchdog.-** Si el watchdog está habilitado y su tiempo de temporización expira antes de producirse la instrucción WDR el microcontrolador se reseteará.
- **Reset por Brown-out.-** El microcontrolador se resetea cuando la fuente de voltaje Vcc está bajo el nivel de Brown-out Reset (VBOT) y el Brown-out detector está habilitado.

Una vez producido un RESET, todos los registros del microcontrolador son puestos con sus valores iniciales y el programa empieza su ejecución en la dirección \$000. En esta dirección debe colocarse una instrucción de salto RJMP para ejecutar la rutina de comienzo de programa. A continuación de la dirección \$000 y hasta la dirección \$00d, se encuentran los vectores para las diferentes interrupciones; pero si estos vectores de interrupción no son utilizados puede colocarse código de programa en ellos.

2.5 Programación del Microcontrolador Atmega8.

La utilización de los lenguajes más cercanos a la máquina (de bajo nivel) representan un considerable ahorro de código en la confección de los programas, lo que es muy importante dada la estricta limitación de la capacidad de la memoria de instrucciones. Los programas bien realizados en lenguaje

Ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución es muy rápida.

Los lenguajes de alto nivel más empleados con microcontroladores son el C y el BASIC de los que existen varias empresas que comercializan versiones de compiladores e intérpretes para diversas familias de microcontroladores. Hay versiones de intérpretes de BASIC que permiten la ejecución del programa línea a línea, y en ocasiones, residen en la memoria

del propio microcontrolador. Con ellos se puede escribir una parte del código, ejecutarlo y comprobar el resultado antes de proseguir.

Por esto se decidió utilizar un compilador Bascom AVR del cual veremos sus características a continuación.

2.5.1 Bascom AVR.

Para la compilación del microcontrolador Atmega 8 se utiliza el software Bascom AVR desarrollado por la empresa Holandesa MCS Electronic.

2.5.1.1 Fundamentos básicos de software Bascom.

El primer punto importante para destacar de este programa, el Bascom AVR, es que está desarrollado en lenguaje Basic, y es importante porque este lenguaje de programación es uno de los más fáciles que se ha diseñado; el nombre BASIC es una abreviatura para: **B**egginers **A**ll-purpose **S**ymbolic **I**nstruction **C**ode, indicando simplemente que es un “Lenguaje de programación para principiantes”. Esta cualidad “para principiantes” ha identificado al Basic por largo tiempo, así que muchos programadores con experiencia han evitado erróneamente usar de él.

Sin embargo, a pesar del tiempo y todos los nuevos lenguajes de programación, Basic tiene un lugar envidiable con su muy extendido uso, razón por la cual el lenguaje de programación Basic se convierte en una herramienta de programación moderna, de gran alcance, y extensamente aceptada. Es por esto que se menciona como una cualidad importante el uso del lenguaje de programación Basic en este programa.

Otra de las características del programa Bascom AVR es que ahora no es necesario contar con un ensamblador para escribir el código fuente y otro paquete de software para comprobar y simular un programa, todo esto se lo realiza con Basic; además con todo el avance de la tecnología, se han construido microprocesadores con memorias tipo flash, con capacidad de programación ISP (**I**n **S**ystem **P**rogramming), y otras herramientas, cualidades que la empresa holandesa MCS Electronics ha sabido aprovechar, y ha cubierto los siguientes componentes de software, con el paquete Bascom AVR.

- Redactor
- Basic recopilador

- Ensamblador
- Simulador
- Terminal Emulador
- LCD designer
- LIB manager
- Programador

Pero el programa Bascom no es solamente un compilador en lenguaje Basic, sino que nos ofrece un único Ambiente de Desarrollo Integrado (IDE). Inicializando el programa nos encontramos en un ambiente de desarrollo donde realizaremos todo nuestro trabajo, y el primer paso es comenzar a escribir el programa, para lo cual se cuenta con un muy buen editor, que conoce todas las opciones útiles que nos ofrece Windows. Por ejemplo presionando la tecla F1 el sistema abre la ayuda con toda la explicación del comando bajo el cursor. Junto a la explicación de la sintaxis para cada comando, se encuentra un breve ejemplo para el comando en particular. Podemos copiar este código (ejemplo) al redactor si lo queremos probar o utilizar. (Figura 14)

```

$regfile = "m8def.dat"
$crystal = 8000000
Config Lcdpin = Pin, Db4 = PORTB 0, Db5 = PORTB 1, Db6 = PORTB 2, Db7 = PORTB 3, E = PORTB 4, Rs = PORTB 5
Config ADC = Single, Prescaler = Auto, Reference = Avcc

Dim Valor As Word
Dim Voltaje As Single
Dim A As Single
Dim B As Single
Dim C As Word

Start ADC
Cursor Off

Do

Cls
Valor = Getadc(0)
Voltaje = Valor * 5
Voltaje = Voltaje / 1023
Voltaje = Voltaje - 0.10
A = Voltaje * 500
B = A / 2.5
C = B * 1
Locate 1, 1
Lcd "Voltaje=" : Lcd Fusing(voltaje, "#.##") : Lcd " V"
Locate 2, 1
Lcd "Peso=" : Lcd C : Lcd " gr"
Waitms 500

Loop

```

Figura 14. Pantalla Principal.

Dentro de esta primera pantalla se encuentra una presentación típica bajo el entorno de Windows, el área de trabajo y la barra de herramienta, dentro de la cual se encuentran iconos conocidos como por ejemplo: Abrir, Nuevo y Guardar.

Guardar como, Imprimir, Vista preliminar, etc. Pero es de gran interés conocer la ayuda que nos brindan los siguientes iconos.



Figura 15. Iconos del Bascom AVR.

Fuente: (GORDON, 2006)

➤ **Revisión de la Sintaxis.** 

El primer icono con una marca de visto realiza una revisión de los errores de sintaxis del código o programa fuente escrito, cuando se realiza esta acción, ningún código adicional es generado en este tiempo, salvo que se encuentre algún error.

➤ **Revisión del Programa.** 

El siguiente icono (Chip) da comienzo al compilador del programa, esta acción de compilar revisa errores de programación, como por ejemplo que alguna variable no esté definida, o esté mal definido, para realizar esta acción es necesario, que no existan errores de sintaxis en el programa fuente. Cuando se realiza esta acción se crea un archivo binario, archivo que es necesario para programar el microcontrolador, archivos adicionales son generados en este proceso aunque son opcionales (MENU – OPTIONS – COMPILER – OUTPUT) y son los siguientes.

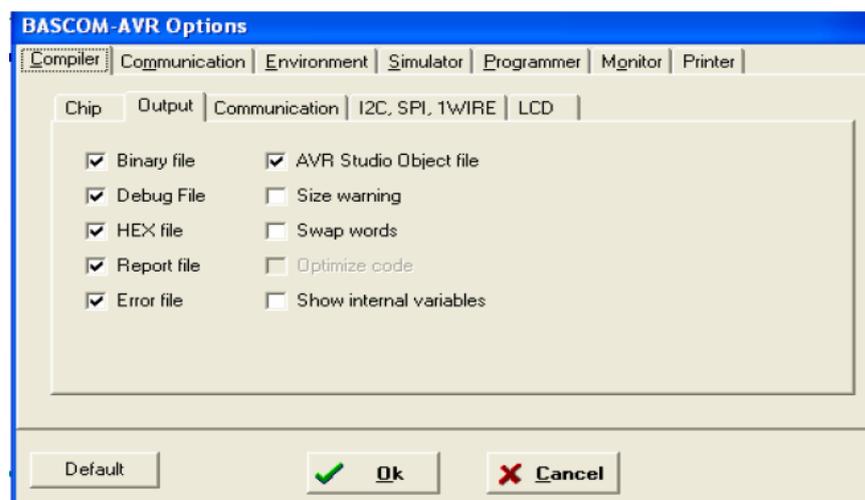


Figura 16. Opciones del Bascom AVR.

Fuente: (GORDON, 2006)

Tabla 6

Tipos de archivos generados en la Compilación.

ARCHIVO	DESCRIPCIÓN
xxx.BIN	(Bynary file) Archivo binario necesario para programar el microcontrolador.
xxx.DBG	(Debug file) Archivo actualizado que se necesita para el simulador.
xxx.OBJ	(Object file) Archivo objeto, se usa cuando se simula con el AVR Studio.
xxx.HEX	(Hexadecimal file) Archivo en formato hexadecimal que se necesita para algunos programadores.
xxx.ERR	(Error file) Archivo de error. Solamente se crea si se encuentran errores en el programa.
xxx.RPT	(Report file) Archivo de reporte (informe).
xxx.EEP	(EEPROM image file).

Fuente: (GORDON, 2006)

Si ocurre un error, se recibe un mensaje de error en una caja de diálogo y la compilación terminará. El resto de los errores serán exhibidos en el fondo sobre la barra de estado.

Cuando se da un clic en la línea con la información del error, se salta a la línea que contiene el error. El margen también exhibirá la muestra. En la compilación siguiente, la ventana del error desaparecerá.

➤ Informe de Resultados



El siguiente icono (La mano de la escritura), permite ver los resultados de la compilación (archivo de informe). En este informe indica a más de la información básica del sistema (versión del compilador, tipo de microcontrolador, fecha, etc.), una lista detallada de todos los recursos del microcontrolador que se encuentran disponibles, así como la lista de las variables que fueron utilizadas, su mecanografía y las direcciones en notaciones hexadecimal y decimal, y al final del informe también se presenta un inventario de las constantes usadas y sus valores.

➤ Simulador



En el desarrollo del proyecto, a menudo es necesario revisar alguna parte del programa fuente o de alguna rutina, si no contamos con una herramienta

de software adecuada, no se tiene otra opción que volver a escribir el programa en varias ocasiones hasta tener éxito. Bascom brinda la posibilidad de reducir drásticamente el número de repeticiones, gracias al simulador que está disponible en este programa.

El siguiente en la lista es el icono del chip (color rojo), dando un clic en este icono, se puede simular el programa, donde el programa fuente es cargado automáticamente del editor, pero para realizar este paso, es necesario compilar el programa primero, porque el simulador realiza su trabajo solamente si existe el archivo .DBG (debug) el cual está disponible solamente después de compilar el programa. Es importante tener esto presente, especialmente si a un archivo ya compilado se le realizó algún cambio, si no se le realiza nuevamente el proceso de compilar, el simulador se cargará nuevamente con el archivo antiguo.

En Bascom AVR se puede escoger el simulador interno o AVR Studio para la simulación. En el menú (OPTIONS – SIMULATOR) El AVR Studio puede ser cargado a Bascom-AVR.

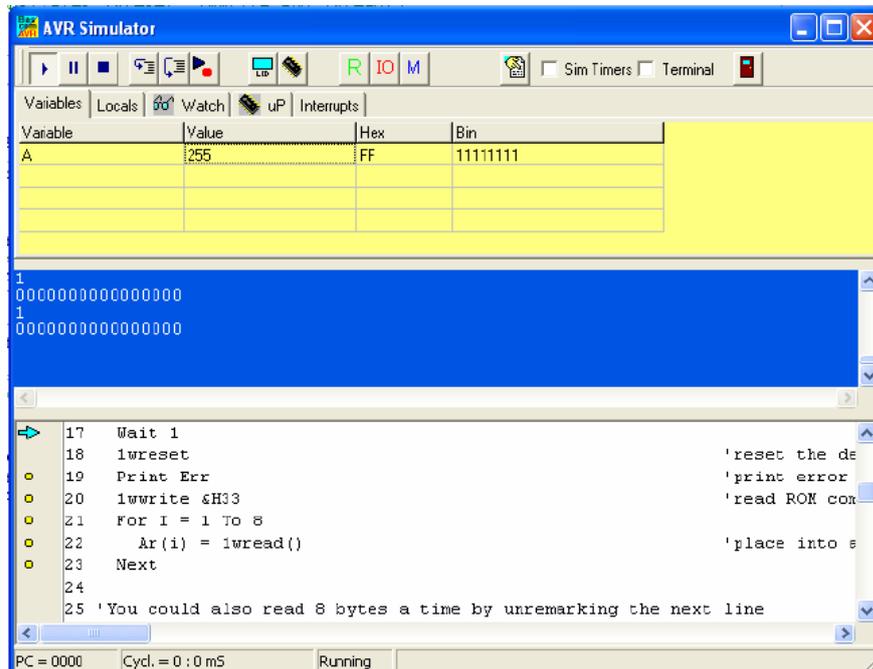


Figura 17. Simulador de Bascom AVR.

Fuente: (GORDON, 2006)

El simulador cuenta con las funciones de: **PLAY – PAUSE Y STOP** para iniciar, hacer una pausa o para parar la simulación, respectivamente, además se puede hacer la simulación paso a paso **STEP** a través del

programa, o por las subrutinas (Subrutinas, GOSUB, CALL) **STEP OVER-BUTTON**. Además se puede realizar la simulación del programa solamente hasta cierta línea, seleccionándola previamente con el botón **RUN TO CURRENT LINE**. El simulador hará una pausa al llegar a este punto.

En el simulador se puede realizar las siguientes acciones:

- Se puede ver el valor de una variable, ubicando el cursor del mouse sobre el nombre de la variable seleccionada
- Es posible agregar una nueva variable, tecleando la variable dentro de la ventana **WATCH**.
- Pulsando los botones INT0, INT1, TO, T1 y SER se puede simular una interrupción. Pueden desplegarse botones adicionales cuando el chip utilizado tiene interrupciones adicionales.
- Con el botón **ENABLE/DISABLE REAL HARDWARE SIMULATION** se habilita o deshabilita la simulación real de hardware, pero para esto se necesita contar con el hardware adicional.

El simulador de hardware simulará el puerto 1 y 3, de esta manera se puede probar el programa en el circuito sin programar el dispositivo.

Pulsando el botón de simulación de hardware **LED** se visualiza la siguiente ventana, una característica destacada del simulador es el emulador de displays LCD, teclado y puertos como se observa en la (figura 18) Permite incluso emular gráficos en el LCD realizados a medida.

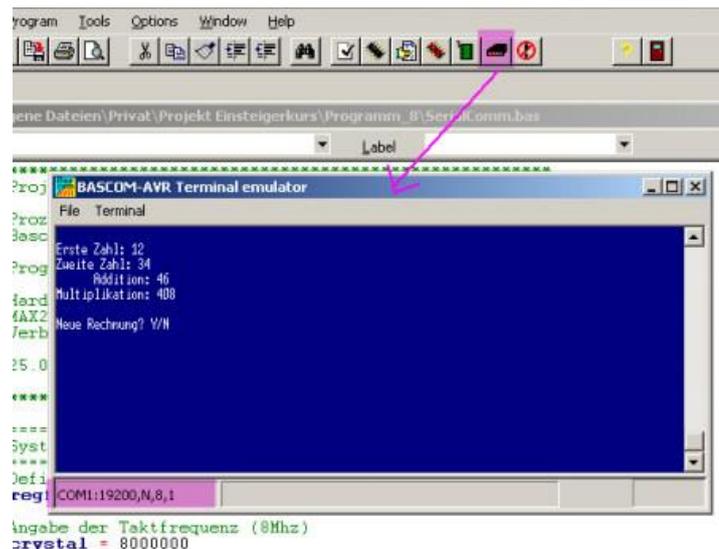


Figura 18. Pantalla de Simulación.

Fuente: (GORDON, 2006)

➤ **Programador** 

El icono con la (figura 18) de un zócalo tipo ZIF, nos brinda la posibilidad de usar un programador, dependiendo de la configuración del programa Bascom; menú (OPTIONS – PROGRAMER) podemos seleccionar 9 diferentes tipos de programadores, más uno externo, claro está que a este último se lo debe diseñar y construir. En este caso se utilizó el programador AVR Usb Usbasp.

➤ **Emulador del Terminal Rs-232.** 

El siguiente icono en la barra de herramientas, representa el emulador del terminal, se utiliza para la comunicación vía interfaz RS-232 con el microcontrolador. El Emulador se encuentra en la barra de herramientas (TOOLS) TERMINAL EMULATOR) o presionando Ctrl + T. Los parámetros de la comunicación pueden ser seleccionados o cambiados en el menú (OPTIONS – COMMUNICATION)

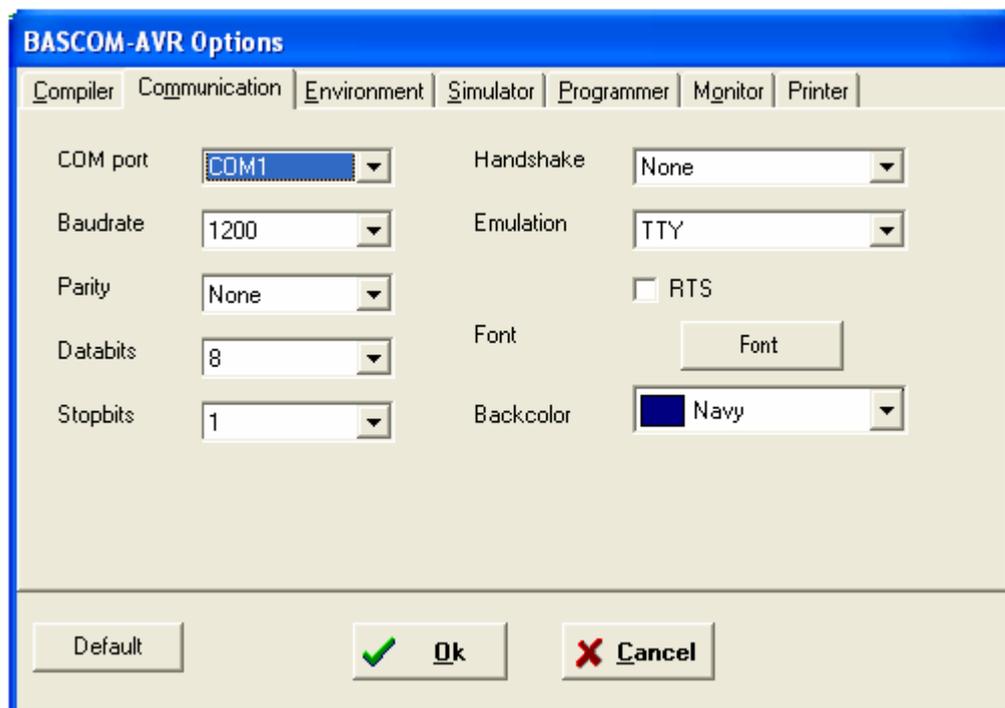


Figura 19. Pantalla de Comunicación Rs-232.

Fuente: (GORDON, 2006)

Tabla 7

Opciones de la Comunicación Rs-232.

ÍTEM	DESCRIPCION
COM port	Puerto de comunicación que utiliza la PC para la comunicación con el terminal emulador.
Baud rate	Es la velocidad en baudios que se va a usar.
Parity	Paridad, por defecto es None
Data bits	Número de bits de datos. Por defecto 8
Stop bits	Número de bits de parada. Por defecto 1
Handshake	Handshake usado, por defecto es None
Emulation	Emulación usada, por defecto BBS ANSI
Font	Tipo de letra y color usado por el emulador
Back ground	Color de fondo del emulador

Fuente: (GORDON, 2006)

Se debe tener en cuenta que la velocidad en baudios del emulador y la velocidad en baudios del compilador debe ser la misma para que trabaje correctamente la comunicación entre la PC y el microcontrolador.

2.5.1.2 Características de Bascom AVR

- Programación estructurada con sentencias IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND and SELECT- CASE.
- Generación de código máquina nativo en lugar de código interpretado.
- Las variables y etiquetas pueden tener una longitud de 32 caracteres.
- Variables de bits, byte, entero (16 bits), entero (32 bits), flotante y cadenas.
- Las instrucciones y comandos de este Basic son bastante similares a las del Visual Basic y Quick Basic de Microsoft.
- Comandos específicos para el manejo de displays LCD, integrados I2C e integrados 1WIRE.
- Simulador para depuración incluido.
- Soporte para programador flas y SPI, PG2051, PG302, SE512, SE514, TAFE.
- Editor con subrayado de sentencias.
- Ayuda ON LINE en el editor.

2.3.1.3 Programación Física Atmega8.

Para la programación del microcontrolador se ha utilizado el programador AVR Usb Usbasp véase (Fig. 20) que utiliza Bascom AVR para el envío del programa al microcontrolador, se utiliza este programador por su fácil manejo debido a que no es necesario desmontar el Microcontrolador de la placa principal para su respectiva programación como los diferentes programadores, esta es otra ventaja del microcontrolador Atmega 8.

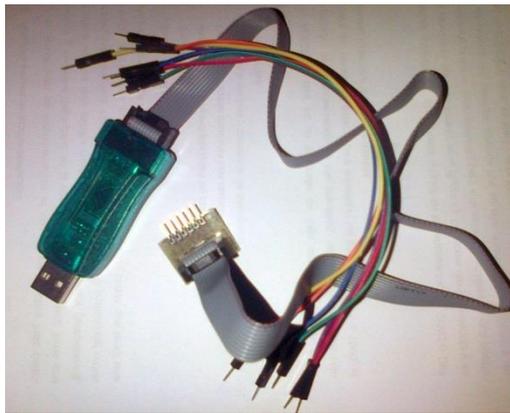


Figura 20: Programador AVR Usb Usbasp.

2.6 Pantalla de Cristal Líquido.

La pantalla de cristal líquido o LCD es un dispositivo de visualización gráfico para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos). En este caso se utiliza uno de 2 filas de 16 caracteres cada una y cada carácter dispone de una matriz de 5x7 puntos (píxeles), aunque los hay de otro número de filas y caracteres

La pantalla de cristal líquido LCD, es un periférico de salida visualizador muy potente, reflexible, eficaz y económico, siendo los microcontroladores los dispositivos más adecuados para optimizar sus prestaciones y control. En realidad, el módulo LCD tiene incrustado un microcontrolador específico para regular su funcionamiento y una memoria RAM.

En la memoria RAM se guardan los caracteres que se van escribiendo, y el microcontrolador se encarga de mostrar en el display la información contenida en esta RAM. La organización lógica de la memoria RAM es como sigue.

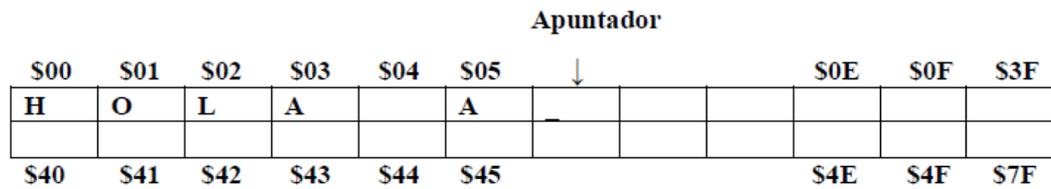


Figura 21: Memoria Ram del Lcd.

Fuente: (MERCADO, 2014)

La flecha simboliza un apuntador de memorias de datos que coinciden con la posición del cursor. El display sólo muestra una pequeña área de la memoria RAM; por esta razón es necesario que al llegar al final de una línea se repositone el cursor al principio de la otra, de no ser así, los caracteres que se sigan enviando no se visualizarán hasta que se llegue al final del renglón-RAM y el cursor brinque automáticamente a la otra línea (arreglo de pila circular).

La instrucción SET DDRAM ADDRESS modifica el valor del APUNTADOR mediante el operando ADD (bits 0 a 6 de la palabra-instrucción), por lo tanto, se puede utilizar esta instrucción para posicionar el cursor en cualquier lugar de la pantalla.

Esta instrucción tiene el BIT 7 en 1, por lo que suele referenciar las direcciones de la pantalla incluyendo a este BIT, es decir:

Línea 1: \$80 \$8F

Línea 2: \$C0 \$CF

Para poder utilizar el módulo LCD correctamente es necesario enviar los siguientes parámetros de inicialización:

1. \$38: Interface de 8 bits, 2 líneas en display y caracteres de 5x7.
2. \$0F: Para encender el display, cursor encendido, mostrarlo parpadeando.
3. \$01: Borrar display.
4. \$06: Modo de incrementar memoria sin corrimiento.
5. \$80: Primer carácter en primera línea, posición cero.

Las pantallas LCD más comunes presentan un número variable de caracteres formados por una matriz de 5x7 pixeles en una o varias líneas. Nos referimos al módulo LCD con microcontrolador incrustado, que tiene la posibilidad de visualizar dos líneas de 16 caracteres cada una. Dispone de 14 terminales, que se especifican en la tabla 8.

Tabla 8

Distribución de las Líneas del LCD.

PIN	SIMBOLO	DESCRIPCION
1	Vss	Tierra de alimentación GND
2	Vdd	Alimentación de +5V
3	Vo	Contraste del cristal líquido. (0 a +5V)
4	RS	Selección del registro de control/registro de datos: RS=0 Selección registro de control RS=1 Selección registro de datos
5	R/W	Señal de lectura/escritura: R/W=0 Escritura (Read) R/W=1 Lectura (Write)
6	E	Habilitación del módulo: E=0 Modulo desconectado E=1 Modulo conectado
7-14	D0-D7	Bus de datos bidireccional

Fuente: (MERCADO, 2014)

Ocho de sus terminales (D0-D7) reciben los caracteres ASCII a representar, así como ciertos códigos de control que regulan los efectos de visualización. También por ellos, el módulo LCD envía información sobre su estado interno.

Por las terminales Vdd y Vss se aplica la alimentación de 5 volts de corriente directa y tierra respectivamente.

La Terminal Vo regula el contraste de la pantalla de cristal líquido al aplicarle, mediante un potenciómetro, una tensión variable comprendida entre 0 y 5 V.

Las terminales restantes controlan las funciones principales del módulo.

- E: Señal de activación. Si E=0, el módulo está desactivado, y no funcionan las señales restantes.
- R/W#: Lectura / escritura. Si R/W=0, se escribe el módulo y si R/W=1, es leído.
- RS: Selección de registro de control RS = 0 o de datos RS = 1. Por D0-D7 se transfiere información de control-comandos o datos de carácter a escribir.

En la tabla 9 se muestran los mandos que controlan al módulo LCD, y más adelante se describen detalladamente

Tabla 9

Mandos de Control de LCD.

Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clock Cycles
NOP	0	0	0	0	0	0	0	0	0	0	No operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clear display & set address counter to zero	3
Cursor Home		0	0	0	0	0	0	0	1	x	Set address counter to zero, return shilled display to original position DDRAM contents remains incharged	3
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Set cursor move direction (I/D) and specity automatic display shift (8)	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turn display (D) cursor on/off (C) and cursor banking (B).	3
Cursor / Display Shift	0	0	0	0	0	1	S/C	R/L	X	x	Shift display or more cursor (S/C) and cursor direction (R/L)	3
Fuction Set	0	0	0	0	1	DL	N	F	X	x	Set interface data widit(DL) number of display lines(N) and character font (F).	3
Set CGRAM Address	0	0	0	1	DGRAM Address					Set DDRAM address DDRAM data a sent afterwards		3
Set DDRAM Address	0	0	1	DDRAM Counter					Set DDRAM address CGRAM data a sent afterwards		3	
Busy Flag & Address	0	1	BF	Address Counter					Read busy fiag (BF) and address counter		0	
Write Data	1	0	Data					Write data into DDRAM or CGRAM		3		
Read Data	1	1	Data					Read data from DDRAM		3		
X:Don't care	I'D	1	Increment					R/L		1	Shift to the right	
		0	Decrement							0	Shift to the left	
	S	1	Automatic display shift					CL		1	8 bit interface	
		0								0	4 bit interface	
	D	1	Display ON					N		1	2 lines	
		0	Display OFF							0	1 line	
C	1	Cursor ON					F		1	5x10 dats		
	0	Cursor OFF							0	5x7 dats		
D	1	Cursor banking					DDRAM: Display Data RAM DGRAM: Character Generator RAM					
	0											
S/C	1	Display shift										
	0	Cursor move										

Fuente: (MERCADO, 2014)

2.6.1 Mandos de control de LCD.

2.6.1.1 Clear Display.

Borra la pantalla del módulo LCD y coloca al cursor en la primera posición, que es la dirección 0. Por defecto, pone el bit I /D =1 para autoincremento de la posición del cursor.

2.6.1.2 Cursor Home.

Pone el cursor en la dirección 0. No varía el contenido de la memoria DDRAM que guarda los datos y que queda diseccionada desde la posición 0.

2.6.1.3 Entry Mode Set.

Establece la dirección del movimiento del cursor (I /D); si pone a 1 el bit S, desplaza la visualización cada vez que se escribe un dato. Si S = 0, la presentación es la normal.

2.6.1.4 Display On/Off Control.

Activa o desactiva al display (D) y al cursor (C) y determinar si parpadea o no.

2.6.1.5 Cursor Display Shift.

Mueve el cursor y desplaza la visualización sin cambiar el contenido de la memoria DDRAM.

2.6.1.6 Function Set.

Establece el número de líneas de la interface con el bus de datos, siendo habitual que sea de 8 bits, con lo que DL = 1. Especifica el número de líneas de caracteres, para que sean 2 se deben poner N = 1, y el formato del carácter, que es de 5x7 píxeles, si F = 0.

2.6.1.7 Set CGRAM Address.

El módulo LCD tiene definido los caracteres ASCII, pero además permite que el usuario pueda definir un máximo de 9 caracteres nuevos. Estos se guardan en la memoria CGRAM de 64 bits. Cada carácter está especificado con un número de bits comprendido entre 6 y 16 bits, según su complejidad. Dichos bits se almacenan en las sucesivas posiciones de la CGRAM. Con

este mando se indica la dirección de CGRAM a partir de la cual se irán almacenando los bits que definen el nuevo carácter. Tras ejecutar este mando, todos los datos que se lean o escriban posteriormente lo hacen desde esa posición de la CGRAM.

2.6.1.8 Set DDRAM Address.

Establece la dirección de la DDRAM a partir de la cual todos los datos que se lean o escriban posteriormente lo harán desde esa posición. Los 16 caracteres del primer renglón ocupan la dirección 80h-8Fh, y los del segundo desde la C0h a la CFh.

2.6.1.9 Read busy flag Address

Se trata de un mando para la lectura de la bandera BUSY que indica si todavía se está ejecutando un mando previo en el módulo LCD.

Además, proporciona la dirección de la CGRAM o DDRAM que se haya utilizado la última vez.

2.6.1.10 Write data to CGRAM o DDRAM

Se escribe en la DDRAM los datos (Caracteres ASCII) que se quieren visualizar. También se escriben en la CGRAM los bits de los nuevos caracteres creados por el usuario. Se usa una memoria u otra según haya sido la instrucción de direccionamiento previa, que hará que se refiera a la CGRAM o a la DDRAM.

2.6.1.11 Read data to CGRAM o DDRAM

Igual que en el mando anterior, pero en esta ocasión en modo de lectura.

CAPÍTULO III

DESARROLLO DEL TEMA

3.1 Preliminares.

En presente capitulo se detalla paso a paso como se realiza la construcción de un prototipo diseñado para la medición de peso, mediante un sensor de peso (celda de carga).

Ya que el sensor entrega una señal eléctrica muy pequeña, del orden de los milivolts, es necesario utilizar un amplificador para aumentar el voltaje. Para esta etapa se utiliza un amplificador operacional AD620 que aumenta la señal a niveles fácilmente manipulables para el microcontrolador Atmega 8.

Una vez que se ha hecho la lectura del peso, lo ideal es que pueda mostrar de una manera clara y rápida en una pantalla LCD 16x2.

3.2 Implementación de la fuente.

En este proyecto se requiere una tensión fija continua y estable de un valor determinado para este caso +5v, +12v, -12v. Un regulador de voltaje proporciona un voltaje de salida de DC constante que es prácticamente independiente del voltaje de entrada, la corriente de carga de salida.

Para suministrar al circuito con los voltajes de alimentación necesarios, se diseñaron dos fuentes:

- a) La primera, requiere una tensión de 5 Vcc; con esta fuente se alimenta el Atmega8, la pantalla de cristal líquido, y la celda de carga de 0 a 500 gr.
- b) La segunda, está encargada de aplicar un voltaje de excitación de ± 12 Vcc que requiere al amplificador de instrumentación AD620.

La (figura 23) muestra el diagrama esquemático del circuito de fuente para 5 V y ± 12 v, En esta, se emplea un transformador de 125 Vac a 12 Vac 1A. Para la rectificación de este voltaje se emplea un puente rectificador de 1 A, proporciona una señal pulsante, compuesta de una señal continua y rizada.

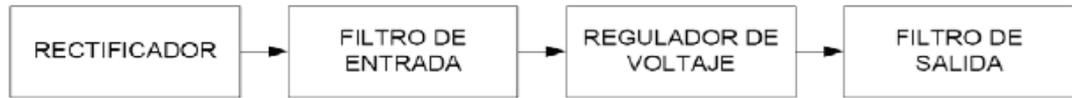


Figura 22. Diagrama de bloques de una fuente de Voltaje DC.

La función de los capacitores $C1$, $C2$, $C3$, $C4$, $C5$ y $C6$ que se hallan en la entrada del regulador es filtrar la tensión de posibles transitorios y picos indeseables mientras que los capacitores de la salida del regulador $C7$, $C8$, $C9$, $C10$, $C11$ y $C12$ se usan para filtrar disminuir la tensión de rizado a la vez que evitan oscilaciones, proporciona una señal continua y lineal.

El Regulador, el cual termina por estabilizar la señal y nos ayuda a obtener una señal continua a la salida de la fuente.

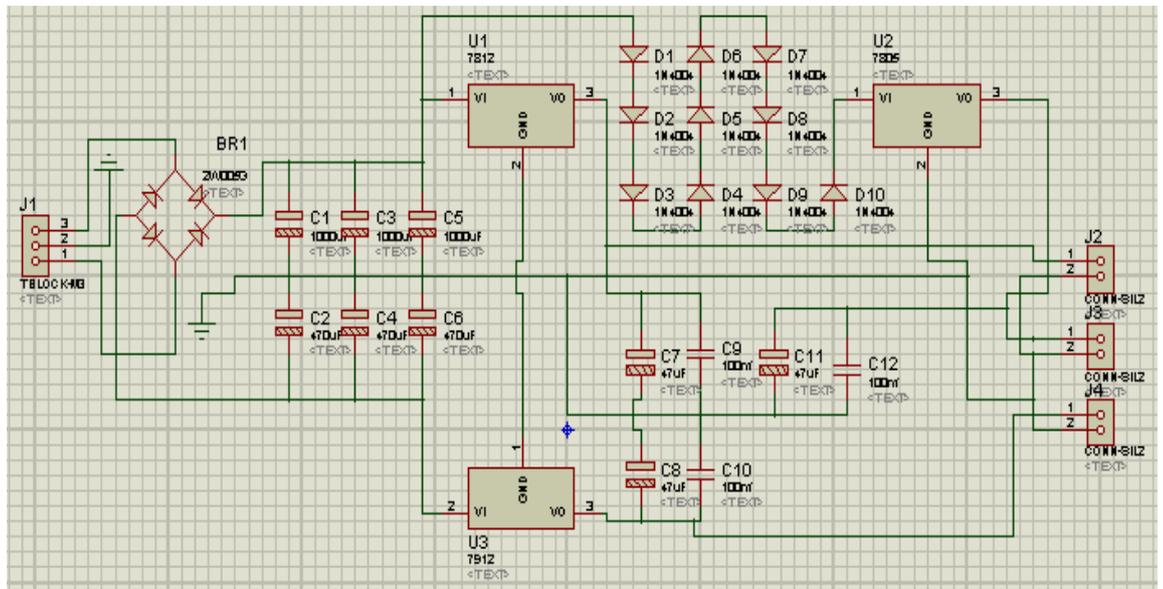


Figura 23. Fuente de Alimentación.

3.3 Pruebas funcionales de la celda de carga.

Estas pruebas permiten verificar el correcto funcionamiento de la celda de carga tanto sin peso como utilizando pesos conocidos y de esta manera podemos medir las distintas respuestas de voltaje ante estas situaciones. (Figura 24).



Figura 24. Prueba del funcionamiento de la celda de carga.

El valor teórico de la señal máxima que entregue la celda de carga es de:

$$\text{Señal Máx} = \text{Sensitividad} \times \text{Voltaje de Excitación}$$

$$\text{Señal Máx} = 0.5 \left[\frac{mV}{V} \right] \times 5[V] = 2.5mV$$

El valor real de la señal máxima que entrega la celda de carga es de:

$$\text{Señal Máx} = 0.46 \left[\frac{mV}{V} \right] \times 4.98[V] = 2.39mV$$

Esta prueba consiste en colocar pesos (figura 25) conocidos sobre el eje de carga, para luego medir la respuesta del voltaje a la salida en las terminales de la celda de carga. Con estos datos se puede determinar el funcionamiento de la celda de carga en la tabla 10.



Figura 25. Pesas patrón para la comprobación de la celda de carga.

Tabla 10

Medición de respuesta de la Celda de Carga.

Peso (gr)	Voltaje (mV)
0	0
5	0.026
10	0.049
20	0.100
50	0.249
100	0.499
200	0.998
500	2.499

Para los pesos conocidos de 5, 10, 20 y 50 gr, la respuesta de la celda de carga va a ser muy pequeña, que no se la puede representar incluso para un multímetro de alta sensibilidad.

3.4 Acondicionamiento de la Celda de Carga.

3.4.1 Amplificación.

La señal de salida de la celda de carga está en el orden de unos pocos milivoltios por lo que no es suficiente para ser procesada, esto quiere decir que necesita ser amplificada.

Para este proyecto, ha sido preciso utilizar un amplificador de instrumentación debido a que los amplificadores operacionales normales introducen altos ruidos al amplificar señales pequeñas.

El amplificador utilizado es el AD620 (Figura 6), que puede ser alimentado con ± 2.3 a $\pm 18V$ además tiene una entrada diferencial y la salida con respecto a un PIN de referencia. La ganancia se puede elegir al conectar una resistencia que está dada por la fórmula siguiente.

$$RG = \frac{49.4 K\Omega}{G+1}$$

De donde RG es la resistencia que establece la ganancia y G es la ganancia que en el AD620 va de 1 a 1000. Mientras que la alimentación para este caso fue de $\pm 12V$.

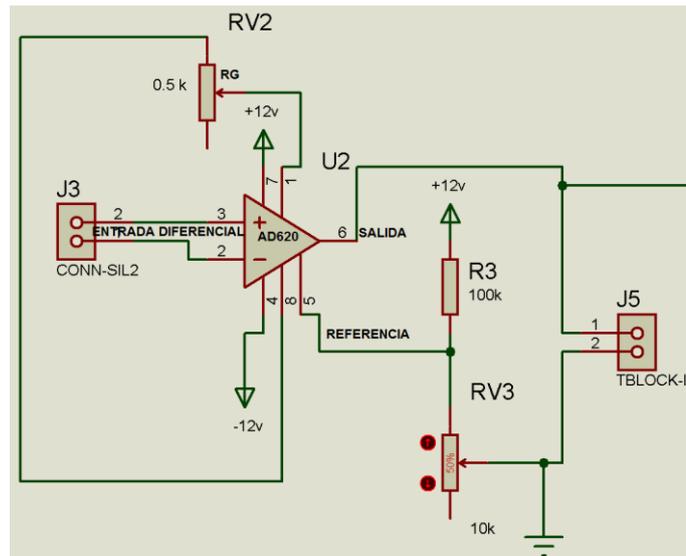


Figura 26. Amplificador de instrumentación AD620.

Debido a que el voltaje diferencial entregado por la celda de carga es demasiado pequeño, es necesario una ganancia de 1000, lo máximo que entrega el amplificador de instrumentación; aplicando la ecuación se obtendrá el valor de la resistencia.

$$RG = \frac{49.4 \text{ K}\Omega}{G - 1}$$

$$RG = \frac{49.4 \text{ K}\Omega}{1000 - 1} = 49.45\Omega$$

3.4.1.1 Anulación de la tensión de offset del AD620.

La tensión offset del AD620 es una tensión generada por el ruido que el amplificador amplifica y puede ser considerada como una fuente de voltaje insertada la cual puede ser corregida o anulada conectando a la salida del pin 5 del AD620 se conecta una fuente variable como indica la (figura 6).

De esta manera el potenciómetro se va ajustando hasta que la tensión de offset se reduce a un valor insignificante.

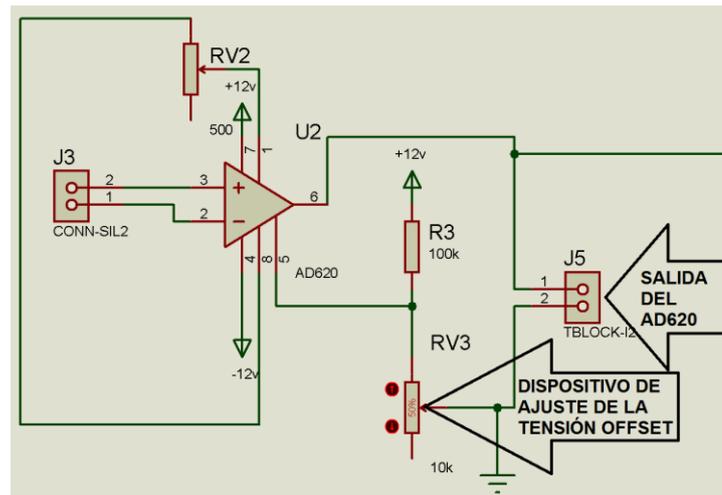


Figura 27. Anulación de la tensión de offset de la AD620.

3.4.2 Conversor A/D.

Una vez amplificada la señal entregada por el amplificador de instrumentación, un conversor A/D, incorporado en el Atmega 8 de ATMEL, recoge esta señal y realiza la conversión necesaria para la visualización de la cantidad medida en la pantalla LCD.

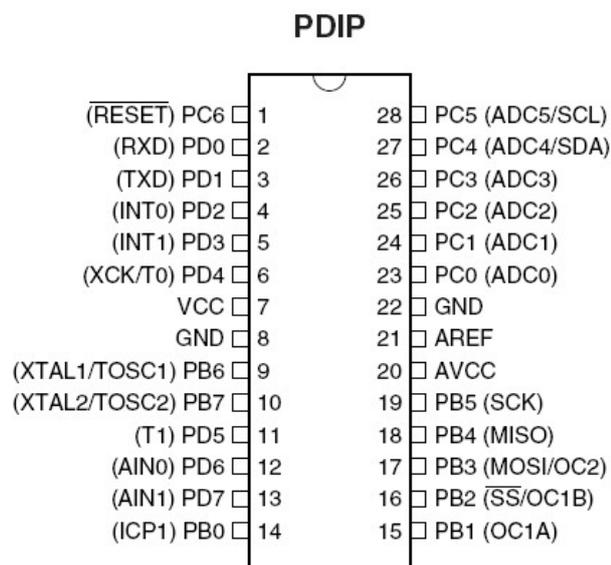


Figura 28. Esquema de pines del Atmega8.

Fuente: (AVRC++, 2015)

3.4.3 Esquema del Circuito.

Los siguientes gráficos muestran el diagrama de los esquemáticos con las conexiones y valores de cada elemento.

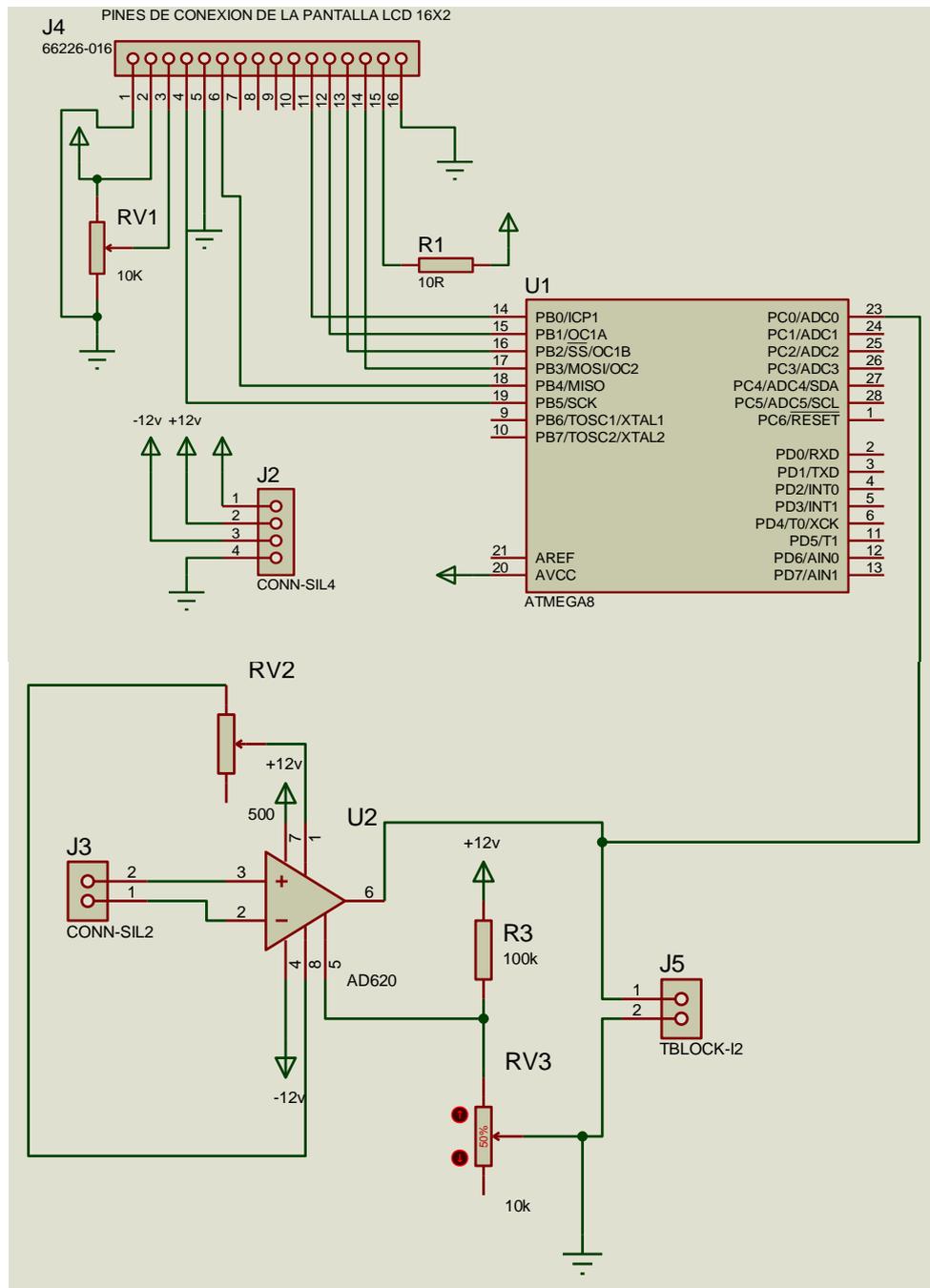


Figura 29. Diagrama del Atmega 8, pantalla LCD y amplificador de instrumentación

3.5 Pruebas.

Se realizaron pruebas a este módulo para verificar la medición del sensor y para verificar el peso máximo que se puede aplicar. Para el desarrollo de esta prueba se midió el voltaje de salida del sensor con diferentes pesos para compararlos con los valores de voltaje calculados, con el dato de 0.5 mV/V.

Peso Calculado.

$$0.5 \left[\frac{mV}{V} \right] x 5[V] = 2.5mV x 1000$$

$$\mathbf{2.5 V = 500gr}$$

Para calcular los valores, se realiza una simple regla de 3.

1) $V \rightarrow 500gr$

$x \rightarrow 200gr$

1.0V = 200gr

2) $2.5V \rightarrow 500gr$

$x \rightarrow 100gr$

0.5V = 100gr

3) $2.5 V \rightarrow 500gr$

$x \rightarrow 50gr$

0.250 V = 50gr

4) $2.5 V \rightarrow 500gr$

$x \rightarrow 20gr$

0.100 V = 20gr

5) $2.5 V \rightarrow 500gr$

$x \rightarrow 10gr$

0.050 V = 10gr

6) $2.5 V \rightarrow 500gr$

$x \rightarrow 5gr$

0.025 V = 5gr

Tabla 11

Pruebas del funcionamiento de la Celda de Carga con amplificación.

Peso (gr)	Voltaje de la celda de carga (mV)	Voltaje salida AD620	Valor en el μ C.	Valor en el LCD
0	0	0	0	0
5	0,026	0,025	10,23	0,02
10	0,049	0,050	20,46	0,05
20	0,010	0,100	40,92	0,10
50	0,249	0,250	102,3	0,25
100	0,499	0,500	204,6	0,5
200	0,998	1,000	409,2	1,0
500	2,499	2,5	1023	2,5

3.5.1 Programación del μ C.

La programación del μ C se realizó en lenguaje C usando el compilador BASCOM-AVR 2.0.3.7.

Las principales funciones del μ C son, adquirir por medio del conversor A/D interno y también, mostrar mensajes de pesos y voltaje en el LCD.

Conversión del valor del voltaje de entrada en gramos, por medio de una ecuación matemática, validación del peso y mostrar en la pantalla LCD el voltaje y el peso.

Diseño del programa a ser cargado en μ C.

PROGRAMA

```

$regfile = "m8def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = PORTB.0 , Db5 = PORTB.1 , Db6 =
PORTB.2
, Db7 = PORTB.3 , E = PORTB.4 , Rs = PORTB.5
Config ADC = Single , Prescaler = Auto , Reference = Avcc
Dim Valor As Word
Dim Voltaje As Single
Dim A As Single
Dim B As Single
Dim C As Word

```

Start ADC
Cursor Off

Do

Cls

Valor = Getadc(0)

Voltaje = Valor * 5

Voltaje = Voltaje / 1023

Voltaje = Voltaje - 0.1

A = Voltaje * 500

B = A / 2.5

C = B * 1

Locate 1 , 1

Lcd "Voltaje= " : Lcd Fusing(voltaje , "#.##") : Lcd " V"

Locate 2 , 1

Lcd "Peso= " : Lcd C : Lcd " gr"

Waitms 500

Loop

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- La celda de carga utilizada puede soportar un peso máximo de 500 gr, con una sensibilidad de 0.5 mV/V. Esto quiere decir que si se alimenta con 5V, tendrá una salida máxima de 2.5 mV con el peso máximo.
- La salida máxima de la celda de carga es de 2.5 mV, por lo cual se debe amplificar con un amplificador de instrumentación y establecer una ganancia de 1000. Esto quiere decir que va a ser amplificado a 2.5 V la señal de la celda de carga.
- La señal amplificada de 2.5V, ingresa al microcontrolador para ser convertida y representada en gramos, por medio de una ecuación matemática.
- La tensión de offset debe ser completamente eliminada porque puede representar un error en la medición.
- La guía de laboratorio se desarrolló para facilitar la manipulación de los módulos.

4.2 RECOMENDACIONES.

- Para poder utilizar el módulo de celda de carga es necesario primero conocer las características técnicas de la celda de carga.
- No se debe sobre pasar del peso máximo que establece el fabricante ya que la celda de carga se puede deformar permanente el sensor.
- En el encendido del circuito no siempre va aparecer el valor inicial de cero, puede darse el caso que por el traslado de la celda de carga o algún movimiento que haya ocurrido anteriormente, para ello se deberá calibrar la balanza antes de empezar a pesar los diferentes pesos en gramos.

GLOSARIO DE TERMINOS.

B

BAUDIOS. Es una unidad de medida utilizada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital.

C

CELDA DE CARGA. Una **celda de carga** es un transductor que es utilizado para convertir una fuerza en una señal eléctrica. Esta conversión es indirecta y se realiza en dos etapas. Mediante un dispositivo mecánico, la fuerza que se desea medir deforma una galga extensiométrica.

TECNOLOGÍA CISC. (Del inglés **Complex Instruction Set Computer**, en español **Computador con Conjunto de Instrucciones Complejas**), Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos.

CMRR. La razón de rechazo al modo común (o CMRR, de las siglas en inglés **Common Mode Rejection Ratio**) es uno de los parámetros de un amplificador operacional u opamp, (en inglés **Operational Amplifier**).

E

EFEECTO PIEZORRESISTIVO. Es la propiedad de algunos materiales conductores y semiconductores, cuya resistencia eléctrica cambia cuando se los somete a un esfuerzo o estrés mecánico (tracción o compresión) que los deforma.

G

GALGA EXTENSIONOMÉTRICA. O también llamado **extensómetro** es un sensor, para medir la deformación, presión, carga, torque, posición, entre otras cosas, que está basado en el efecto piezorresistivo.

H

HISTÉRESIS. Es la tendencia de un material a conservar una de sus propiedades, en ausencia del estímulo que la ha generado. Podemos encontrar diferentes manifestaciones de este fenómeno. Por extensión se aplica a fenómenos que no dependen sólo de las circunstancias actuales, sino también de cómo se ha llegado a esas circunstancias.

L

LINEALIDAD. Capacidad de un instrumento de medición para proporcionar una indicación que tenga una relación lineal con una magnitud determinada distinta de una magnitud de influencia.

M

MEMORIA SRAM. Son las siglas de la voz inglesa Static Random Access Memory, que significa memoria estática de acceso aleatorio (o RAM estática), para denominar a un tipo de tecnología de memoria RAM basada en semiconductores, capaz de mantener los datos, mientras siga alimentada.

P

PUENTE DE WHEATSTONE. Se utiliza para medir resistencias desconocidas mediante el equilibrio de los brazos del puente. Estos están constituidos por cuatro resistencias que forman un circuito cerrado, siendo una de ellas la resistencia de bajas medidas.

R

REPETIBILIDAD. Aptitud de un instrumento de medición para proporcionar indicaciones próximas entre sí para aplicaciones repetidas del misma mensurando bajo las mismas condiciones de medición.

RESISTENCIA DE AISLAMIENTO. Voltaje máximo al que puede exponerse un material sin provocarle perforación alguna; expresado en voltios o kilovoltios por unidad de grosor. También llamada resistencia dieléctrica.

TECNOLOGIA RISC. (Del inglés **R**educed **I**nstruction **S**et **C**omputer, en español Computador con Conjunto de Instrucciones Reducidas) es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores con las siguientes características fundamentales:

1. Instrucciones de tamaños fijos y presentados en un reducido número de formatos.
2. Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

RESISTENCIAS PULL-UP. En electrónica se denomina *pull-up* a la acción de elevar la tensión de salida de un circuito lógico, a la tensión que, por lo general mediante un divisor de tensión, se pone a la entrada de un amplificador con el fin de desplazar su punto de trabajo.

S

SENSIBILIDAD. Cociente del incremento de la respuesta de un instrumento de medida por el incremento correspondiente de la señal de entrada.

Referencia Bibliográfica

- A. CARRETERO, F. F.-I.-I. (s.f.). *ELECTRONICA*. EDITEX.
- AVRC++. (15 de MAYO de 2015). *ATmega8/index.html*. Obtenido de <http://avr-cpp-lib.sourceforge.net/>.
- DEVICES, A. (2 de JUNIO de 2015). *ad620.html#product-overview*. Obtenido de [http://www.analog.com/:
http://www.analog.com/en/products/amplifiers/instrumentation-amplifiers/ad620.html#product-overview](http://www.analog.com/:http://www.analog.com/en/products/amplifiers/instrumentation-amplifiers/ad620.html#product-overview)
- GORDON, N. (MAYO de 2006). *bitstream/15000/1955/1/CD-0193.pdf*. Recuperado el MARTES, 23 de DICIEMBRE de 2014, de [http://bibdigital.epn.edu.ec/:
http://bibdigital.epn.edu.ec/bitstream/15000/1955/1/CD-0193.pdf](http://bibdigital.epn.edu.ec/:http://bibdigital.epn.edu.ec/bitstream/15000/1955/1/CD-0193.pdf)
- MERCADO, E. (16 de NOVIEMBRE de 2014). *c/document_library/get_file?uuid=6878171a-72ce-4a83-89f4-86d001a325d7&groupId=54327*. Obtenido de <http://ice.uaz.edu.mx/>.
- NAVARRETE, V. (s.f.). Recuperado el 7 de diciembre de 2014, de <http://bibdigital.epn.edu.ec/bitstream/15000/4041/1/CD-3794.pdf>
- Omega. (2 de junio de 2015). *Load cells*. Obtenido de Introducción a células de carga: <http://www.omega.com/prodinfo/loadcells.html>
- Omega. (2 de junio de 2015). *load-cells*. Obtenido de introduction to load cells: <http://www.omega.co.uk/prodinfo/load-cells.html>
- Seeed Technology Limited. (2008). *depot/Weight-Sensor-Load-Cell-0500g-p-525.html*. Obtenido de <http://www.seeedstudio.com/>.
- SIMBAÑA, T. (7 de JUNIO de 2009). *bitstream/15000/1728/1/CD-2336.pdf*. Recuperado el 7 de MAYO de 2015, de [http://bibdigital.epn.edu.ec/:
http://bibdigital.epn.edu.ec/bitstream/15000/1728/1/CD-2336.pdf](http://bibdigital.epn.edu.ec/:http://bibdigital.epn.edu.ec/bitstream/15000/1728/1/CD-2336.pdf)
- VITERY, N. (JUNIO de 2011). *bitstream/15000/4041/1/CD-3794.pdf*. Recuperado el 7 de DICIEMBRE de 2014, de [http://bibdigital.epn.edu.ec/:
http://bibdigital.epn.edu.ec/bitstream/15000/4041/1/CD-3794.pdf](http://bibdigital.epn.edu.ec/:http://bibdigital.epn.edu.ec/bitstream/15000/4041/1/CD-3794.pdf)

ANEXOS