

**ASESORIA FORMATIVA Y TECNICA EN LOS PROGRAMAS DE
TECNOLOGIA EN ELECTRONICA INDUSTRIAL Y
AUTOMATIZACION DEL SENA REGIONAL NARIÑO.**



JAVIER HUMBERTO CHAMORRO ARTEAGA

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO
2007**

**ASESORIA FORMATIVA Y TECNICA EN LOS PROGRAMAS DE
TECNOLOGIA EN ELECTRONICA INDUSTRIAL Y
AUTOMATIZACION DEL SENA REGIONAL NARIÑO.**

JAVIER HUMBERTO CHAMORRO ARTEAGA

**TRABAJO DE GRADO PARA OPTAR POR EL
TITULO DE INGENIERO ELECTRONICO**

ING. DARIO FAJARDO FAJARDO
Asesor

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO
2007**

Nota de Aceptación

ING. ROLANDO BARAHONA C.
JURADO

ING. IVAN VINUEZA CORDOBA
JURADO

18 de Mayo de 2007

RESPONSABILIDAD

“Las ideas y conclusiones aportadas en el proyecto de grado son responsabilidad exclusiva de sus autores”.

Artículo Primero del Acuerdo N°. 324 de octubre 11 de 1966, emanada del Honorable Consejo Directivo de la Universidad de Nariño.

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

La Universidad de Nariño por permitir que las personas tengan un espacio donde formarse como profesionales, más aún con la vinculación del programa de pregrado de Ingeniería Electrónica.

Al SENA Regional Nariño por permitirme desarrollar actividades de pasantía en esta entidad y poder aportar un grano de arena a la consecución de prototipos que son base para el avance y reconocimiento de la tecnología de nuestra región.

El Ingeniero Darío Fajardo Fajardo, asesor de este proyecto por parte de la Universidad, quien amablemente invirtió tiempo y su conocimiento a la hora de plantear los retos emprendidos y culminados.

El Ingeniero Orlando Orbes, asesor y jefe inmediato del SENA, quien además de supervisar las actividades desarrolladas, me brindó la posibilidad de formar parte integral del equipo de trabajo de diseño y desarrollo de los prototipos alcanzados.

Directivos e Instructores del SENA, quienes me acogieron como un integrante más de trabajo dentro de la institución y me brindaron su apoyo en todo momento.

Estudiantes Tecnólogos de Electrónica Industrial y Automatización, con quienes tuve el agrado de trabajar en los retos emprendidos y poder sacar adelante productos de innovación y desarrollo tecnológico.

Mi familia y amigos cercanos, quienes en el transcurso de la carrera y en esta parte final del desarrollo de la pasantía me colaboraron y estuvieron siempre conmigo apoyándome y brindándome sus sabios consejos.

TABLA DE CONTENIDO

	Pág.
INTRODUCCION	12
2. MARCO TEORICO	16
2.1 LABVIEW	17
2.2 MICROCONTROLADORES	18
2.3 ADQUISICIÓN DE DATOS	19
2.4 CONTROL Y AUTOMATIZACIÓN	19
3. METODOLOGIA	21
4. DESARROLLO DEL PROYECTO	26
4.1 DESARROLLO DE RUTINAS COMUNES A PROYECTOS	32
4.1.1 Rutinas en mplab.	32
4.1.2 Rutinas en labview.	33
4.2 DESARROLLO DE RUTINAS EN MPLAB ESPECÍFICAS	39
4.2.1 Rutinas para sistema de adquisición de datos analógicos didáctico	39
4.2.2 Rutinas para sistema de control de motores por computador	40
4.2.3 Rutinas para sistema automatización de horno ahumador	41
4.2.4 Rutinas para sistema de ambientes controlados para crianza de cerdos	41
4.3 DESARROLLO DE RUTINAS EN LABVIEW ESPECÍFICAS	42
4.4 TUTORÍA A NIVEL DE HARDWARE	51
4.5 ACTIVIDADES DE MANTENIMIENTO DE EQUIPOS	51
5. RESULTADOS	53
CONCLUSIONES	54
RECOMENDACIONES	55
BIBLIOGRAFÍA	56
ANEXOS	57

LISTA DE FIGURAS

	Pág.
1. Sub VI arreglo a serial	34
2. Sub VI serial a arreglo	34
3. Diagrama de bloques sub VI interfaz serial	35
4. Panel frontal sub VI interfaz serial	36
5. Diagrama de bloques sub VI word a nibble	37
6. Diagrama de bloques ciclo word a nibble	37
7. Diagrama de bloques sub VI nibble a word	38
8. Diagrama de bloques adquisición canal analógico	39
9. Diagrama de bloques sub VI nibble a byte	43
10. Diagrama de bloques sub VI recepción	44
11. Panel frontal configuración tarjeta analógica	45
12. Panel frontal control tarjeta analógica	45
13. Panel frontal adquisición tarjeta analógica	46
14. Panel frontal configuración tarjeta de motores	46
15. Panel frontal control tarjeta de motores	47
16. Panel frontal adquisición tarjeta de motores	47
17. Panel frontal configuración sistema crianza de cerdos	48
18. Panel frontal control sistema crianza de cerdos	49
19. Panel frontal adquisición sistema crianza de cerdos	50
20. Funcionamiento de los microprocesadores	62
21. Microprocesador pic 16f87x	64
22. Arquitectura interna 16f87x	66

LISTA DE ANEXOS

	Pág.
ANEXO A. PROTOTIPO MODELO: SISTEMA DE ADQUISICIÓN DE DATOS ANALÓGICOS MICROPROCESADOS.	57
ANEXO B. FUNDAMENTOS DE LABVIEW	58
ANEXO C. FUNDAMENTOS DE MICROCONTROLADORES	62
ANEXO D. ADQUISICION DE DATOS	69
ANEXO E. CONTROL Y AUTOMATIZACION	72
ANEXO F. RUTINAS DESARROLLADAS EN MPLAB CONJUNTAMENTE CON LOS ESTUDIANTES	76

GLOSARIO

Actuador: Recibe la señal del controlador y modifica el agente de control.

Arreglo: Un array (más correctamente denominado vector), es un conjunto de variables o registros del mismo tipo que pueden estar almacenados en memoria principal o en memoria auxiliar. Los array de 1 dimensión se denominan vectores, los de 2 o más dimensiones se denominan matrices.

Asesoría: es el servicio en la que un estudiante distante recibe orientación por parte de un experto en la materia o contenido en relación a: estrategias de estudio, realización de trabajos, contenidos, problemas, o dificultades en las experiencias de aprendizaje.

Automatización: Ejecución automática de tareas industriales, administrativas o científicas haciendo más ágil y efectivo el trabajo y ayudando al ser humano que está sometido o expuesto a enfermedades profesionales.

Bit: Unidad de información o dígito binario.

Control de Procesos: Es un medio por el cual se incide o afecta el comportamiento de un proceso industrial para conseguir un objetivo específico. Conjunto de principios y normas que garantizan la adecuación entre datos, proceso e información, a fin de que los informes obtenidos reúnan las características demandadas por sus usuarios.

Controlador: instrumento que compara la variable controlada con un valor deseado y ejerce automáticamente una acción de corrección de acuerdo con la desviación.

Controlador Lógico Programable: Sistema lógico que combina hardware y software, de modo que sus instrucciones codificadas pueden modificarse por el uso de un teclado; proporciona automatización flexible en vez de dedicada.

Datos analógicos: Datos en forma de cantidades físicas de variaciones continuas de tensiones eléctricas.

Datos digitales: Señal que utiliza valores discretos en lugar de un espectro continuo de valores.

Display: Presentación visual de una señal.

Hardware: Se denomina hardware o soporte físico al conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

Instrumentación: Mecanismo para señalar la información sobre el estado de la dotación física de PC y del software para permitir a aplicaciones de la gerencia comprobar y cambiar el estado de una PC y ser notificado de cambios del estado.

Interfaz: Una interfaz es la parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

Mantenimiento: conjunto de técnicas y sistemas que permiten prever las averías, efectuar revisiones, y reparaciones eficaces, dando a la vez normas de buen funcionamiento a los operadores de las máquinas, a sus usuarios, contribuyendo a los beneficios de la empresa. Es un órgano de estudio que busca lo más conveniente para las máquinas, tratando de alargar su vida de forma rentable.

Microprocesador: circuito integrado que trae las instrucciones de programa, las decodifica y las ejecuta.

Modulación por ancho de pulso (PWM): técnica de variación continua de la potencia promedio a una carga eléctrica cambiando el ciclo de trabajo de pulsos aplicados a la carga.

Multitarea: El término multitarea se refiere a la capacidad del SO para correr más de un programa al mismo tiempo.

Proceso: Una serie encadenada de acciones y reacciones, que se combinan para producir un resultado final, considerado como el lugar donde se encuentra la variable física a controlar.

Programación: Se llama programación al acto de crear un programa de computadora, un conjunto concreto de instrucciones que una computadora puede ejecutar. El programa se escribe en un lenguaje de programación, aunque también se pueda escribir directamente en lenguaje de máquina, con cierta dificultad. Un programa se puede dividir en diversas partes, que pueden estar escritas en lenguajes distintos.

Prototipo: Un prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

Sensor: Un sensor es un dispositivo que detecta, o censa manifestaciones de cualidades o fenómenos físicos, como la energía, velocidad, aceleración, tamaño, cantidad, etc.

Señal Analógica: señal de salida del instrumento que es una función continua de la variable medida.

Señal Digital: señal de salida del instrumento que representa la magnitud de la variable medida en forma de una serie de cantidades discretas codificadas en un sistema de notación.

Software: También conocido como soporte lógico, compendio de todo tipo de programas, utilidades, aplicaciones, sistemas operativos, drivers que hacen posible que el usuario pueda trabajar con la máquina.

Tecnología: Tecnología en su sentido más elemental no es más que un proceso de ingeniería. Sin embargo, en un sentido más amplio, es entendido como un producto en sí mismo, el cual en adición con maquinaria y equipos, concesiones avanzadas, patentes, marca de fábrica, instrucciones, descripciones y experiencia de personal especializado, también incluye know-how.

Tutoría: Tiempo dedicado por el profesor a ejercer la tutela, orientación y consejo a los alumnos que estudian una asignatura.

RESUMEN

El proyecto que se presenta, hace referencia a las actividades de pasantía desarrolladas en el Centro Multisectorial Lope SENA, para optar por el título de Ingeniero Electrónico. La pasantía se desarrolla a nivel de asesoría de proyectos de innovación tecnológica para estudiantes de tecnología en Electrónica Industrial y Automatización del centro en mención.

Los proyectos adelantados están relacionados con el diseño y construcción de cinco diferentes prototipos, entre los cuales se encuentran sistemas de adquisición de datos de tipo didáctico, al igual que sistemas de automatización de procesos del sector de agroindustria del SENA Regional Nariño, tal es el caso del ambiente de crianza de cerdos y el horno de ahumado.

Se desarrollan actividades en cuanto a programación en lenguajes ensamblador para el control desde el microcontrolador y de entorno gráfico para el desarrollo de la interfaz gráfica, así como tutoría en el diseño, simulación y montaje de circuitos electrónicos para las tarjetas.

ABSTRACT

The project that is presented makes reference to the internship activities developed in the Multisectorial Lope Center SENA, to opt for the degree's title of Electronic Engineering. The internship is developed at level of consultancy of projects of technological innovation for technology students in Electronic Industrial and Automation of the center in mention.

The projects are related about the design and construction of five different prototypes, among which are systems of acquisition of data of didactic type, the same as systems of processes's automation of the sector of agroindustry of the SENA Regional Nariño, as they are it the atmosphere of upbringing of pigs and the oven of smoky.

A hard work is developed as for programming in languages assembler for the control from the microcontrolador and of environment graph for the development of the interfaz, the same as tutorship in the design, simulation and assembly of electronic circuits for the cards.

INTRODUCCION

Un profesional de Ingeniería aplica sus conocimientos por medio del trabajo que realiza con el fin de prestar un servicio o beneficio necesario y útil a la comunidad. Esto se evidencia más aún al tratarse de un área técnica en donde se pretende solucionar problemas de la sociedad, procurando mejorar la calidad de vida de las personas.

Específicamente, en lo relacionado a la Ingeniería Electrónica, el control industrial y automatización, se refiere a mecanismos que desarrollarán las tareas de una manera más rápida y exacta; eso es justamente lo que se pretende para que la producción en algún campo de aplicación sea fortalecida y tecnificada.

Otro punto importante es colaborar en la formación de personas en temas relacionados con este campo del conocimiento tecnológico, prestando asesoría y procurando que el trabajo se pueda evidenciar en el avance y buen desempeño de los estudiantes. Se pretende asesorar a estudiantes de Tecnología en Electrónica Industrial y Tecnología en Automatización del SENA Regional Nariño, a quienes no solamente se brinda acompañamiento activo en su formación, sino que también se ofrece mejores condiciones de trabajo y estudio dentro del mismo centro, con la puesta en ejecución de espacios en los cuales puedan desarrollar sus proyectos. Ellos a su vez, para poder certificarse, deben desarrollar proyectos finales, en los cuales apliquen los conocimientos que en teoría han adquirido, para cuyo fin necesitan asesoría de una persona conocedora del tema, que aporte sugerencias y soluciones ante problemas que se presenten en el desarrollo de los proyectos, de ésta forma se interviene con la asesoría necesitada.

La anterior descripción muestra un breve resumen de lo que se trabaja con el desarrollo de la pasantía en el SENA y su explicación más detallada se presenta a lo largo de este proyecto. Las actividades de pasantía a desarrollar se realizan en el SENA durante seis meses tiempo completo, con asesorías a nivel formativo dirigido a los estudiantes de las Tecnologías de Electrónica Industrial y Automatización del centro en mención, además de colaborar en actividades de mantenimiento de equipos electrónicos e informáticos específicos de los laboratorios de electrónica y automatización, así como también colaborar en la puesta en ejecución del aula de Gestión y Desarrollo de Prototipos del centro, con el desarrollo de proyectos por parte de estudiantes.

OBJETIVOS

- GENERAL:

Desarrollar actividades de asesoría en proyectos y mantenimiento de hardware en los programas de Tecnología en Electrónica Industrial y Automatización del sector de Industria del SENA Regional Nariño

- ESPECIFICOS:

Ejecutar actividades de tutoría para alumnos de las Tecnologías de Electrónica Industrial y Automatización del sector de industria, dirigidas al desarrollo de proyectos de innovación y desarrollo tecnológico.

Desarrollar labores de mantenimiento de los laboratorios de Electrónica, Mecatrónica y puesta en ejecución del aula de Gestión y Desarrollo de prototipos (Aula Abierta)

Desarrollar conjuntamente con los estudiantes de las tecnologías en Electrónica y Automatización, hardware y rutinas en LabVIEW para la adquisición de datos y control de diferentes proyectos o prototipos.

EL PROBLEMA

- DESCRIPCION DEL PROBLEMA: En el SENA Regional Nariño se desarrollan entre otros, los cursos de Tecnología en Electrónica Industrial y Tecnología en Automatización; existen 20 y 13 estudiantes, respectivamente para cada programa, al igual que docentes o instructores responsables de éstos y que a su cargo tienen la instrucción durante gran parte de los dos años, que corresponde al tiempo en que estos programas se prolongan.

El centro maneja actualmente la formación por proyectos, lo cual requiere que constantemente los alumnos desarrollen actividades de innovación de diferentes tipos, relacionadas con los conocimientos adquiridos en las asignaturas del plan de estudios de cada programa. Cada grupo de estudiantes maneja un proyecto distinto y requiere un acompañamiento activo por parte de los instructores para poder desarrollarlo; puesto que se tratan de proyectos de aplicación complejos para los cuales es necesaria una asesoría de una o más personas conocedoras de los temas que se estén tratando. Además, si se tiene en cuenta que muchos de estos proyectos van a ser implementados, los convierten en un compromiso serio y por el cual hay que responder a tiempo y presentar el prototipo trabajando eficientemente.

Aparte de ello, se están abriendo nuevos espacios físicos dentro del sector industrial del centro para que los estudiantes los utilicen en el desarrollo de sus

proyectos, tal es el caso del Aula de Gestión y Desarrollo de Prototipos que se está poniendo en ejecución con varios computadores, software necesario, conexión a Internet para desarrollo de consultas, bancos de trabajo para diseño y montaje de circuitos, entre otros elementos. Con esto se pretende que los estudiantes tengan un espacio donde adelantar sus proyectos finales y cuenten con gran parte de herramientas de diseño y simulación tanto de software como hardware. Sin embargo, para eso se necesita trabajar en la consecución de unos bancos de trabajo con tarjetas de adquisición de datos que permitan hacer la interfaz con el PC y realizar el control de los dispositivos desarrollados desde el computador.

Por todo lo explicado anteriormente, se hace no solo conveniente sino necesaria la participación de una persona externa o diferente de las anteriormente citadas, para que realice estas labores como son: el mantenimiento a los recursos informáticos específicos del área y equipos electrónicos, la configuración de la red para el sector industrial y agropecuario, la colaboración en la puesta en marcha del Aula de Gestión y Desarrollo de Prototipos. Dentro de ésta aula se podrá aportar un acompañamiento activo a los estudiantes para reforzar algunos temas y colaborar con ideas para mejorar el desarrollo de los proyectos planteados por los alumnos, así como aportar soluciones ante dificultades que se presenten en el desarrollo de los mismos.

Dentro del SENA, y específicamente hablando en las áreas de Electrónica y Automatización se ha trabajado con diferente software de simulación y control, como son las herramientas de Multisim, WinSPS, MPLab, Automation Studio, CAM, CAD, WorkBench, entre otros; pero, no se ha trabajado con la herramienta LabVIEW. Este software de programación de control y adquisición de datos, se ha adquirido hace poco tiempo y se desea darlo a conocer a la comunidad estudiantil de éste sector industrial, para que sea aplicada en el desarrollo de sus proyectos.

Se trata de una herramienta nueva y muy útil con la cual se desarrollarán rutinas y programas de adquisición y control para diferentes modelos de prototipos.

- FORMULACION DEL PROBLEMA

¿Es necesaria la asesoría técnica en la formación de estudiantes de las tecnologías en Electrónica Industrial y Automatización del SENA Regional Nariño?

JUSTIFICACION

La realización de la presente pasantía abre las puertas para que los estudiantes ofrezcan y realicen un trabajo de extensión a la comunidad cuyo desarrollo garantice beneficiar a uno o varios grupos de personas como se establece en el programa de Ingeniería Electrónica de la Universidad de Nariño.

Por otra parte, es conveniente tanto para la institución en la cual se lleva a cabo, como también lo es para el estudiante pasante de Ingeniería Electrónica de la Universidad de Nariño, puesto que se cuenta con la presencia de personal capacitado que contribuye con el desarrollo de proyectos de innovación de estudiantes de las Tecnologías de Electrónica Industrial y Automatización del SENA. Así mismo, con la asesoría prestada, se adquiere experiencia en el desarrollo de proyectos relacionados con la electrónica y específicamente con el área de control y automatización industrial, puesto que las propuestas están dirigidas al diseño e implementación de prototipos de control de procesos en su mayoría de carácter agropecuario, así como a la utilización de los productos en pro de la formación académica de nuevos estudiantes, por su carácter didáctico.

Son 33 estudiantes pertenecientes a los programas de Tecnología en Electrónica Industrial y Automatización, quienes se dividen en grupos de trabajo y desarrollan diferentes proyectos por cada grupo, con características y especificaciones distintas y con una complejidad relativa al carácter de los programas que se adelantan. Por ello se hace necesaria la participación de personal que aporte al grupo la oportunidad de resolver dudas que contribuya a sacar adelante y con éxito los proyectos. Además, si se tiene en cuenta que los proyectos que se realizan tienen desarrollo de software, así como también diseño e implementación de hardware; es conveniente el aporte a nivel de asesoría que se puede ofrecer con pautas tanto para el diseño, simulación e implementación del hardware como para la programación ya sea en microcontroladores o en un software de computador, de los cuales los estudiantes tienen escasos conocimientos, como son MPLab, Visual Basic, LabVIEW, entre otros.

Igualmente, es conveniente realizar el mantenimiento de equipos informáticos específicos y equipos electrónicos, procurando la apropiación de conocimiento y un ambiente de trabajo en buenas condiciones dentro del cual los estudiantes de las dos tecnologías puedan desarrollar con éxito los proyectos a su cargo, al contar con las herramientas necesarias para tal fin.

Con la construcción de bancos de trabajo, además de la configuración de red e Internet, se garantiza un espacio físico de desarrollo de actividades a los estudiantes y se pone en ejecución el aula de Gestión y Desarrollo de Prototipos.

Por todo lo descrito anteriormente, el aporte social que tiene el proyecto a nivel de la mejora del servicio de la educación en la institución se hace evidente. Igualmente se pueden observar los beneficios académicos que trae consigo, por cuanto se mejoran los métodos y condiciones laborales, los métodos de aprendizaje de los estudiantes quienes reciben apoyo directo en el desarrollo de sus proyectos.

2. MARCO TEORICO

El Centro Multisectorial Lope – SENA Regional Nariño, es una entidad de carácter oficial que tiene como fin la formación gratuita de personas, en diferentes campos de investigación. En el sector Industrial de esta regional, actualmente se llevan a cabo entre otros, los cursos de Tecnología en Electrónica Industrial y Tecnología en Automatización. Estos programas se abrieron al público en el año 2005 con un cupo de 25 estudiantes por cada uno de ellos.

La Tecnología en Electrónica Industrial es un programa que se ha abierto a la población en varias ocasiones, sin embargo, no se hace de manera continua. Este programa tiene una duración de 2 años de formación y 6 meses de práctica empresarial.

Anteriormente se contaba con el programa de Tecnología en Instrumentación y Control Industrial, que posteriormente fue reemplazado por el de Tecnología en Automatización que igualmente tiene una duración de 2 años de formación correspondiente al plan de estudios, además de 6 meses de práctica empresarial.

Los estudiantes que en anteriores oportunidades hicieron parte de estos programas realizaron unos proyectos finales, relacionados con el diseño y construcción de tarjetas de adquisición de datos análogas y/o de carácter específico para una aplicación; como tarjetas con demoduladores, reguladores de voltaje, limitadores de tensión, rectificadores, entre otras. Los anteriores proyectos fueron realizados por estudiantes de Tecnología en Electrónica Industrial, mientras que los estudiantes pertenecientes al campo de la Automatización desarrollaron proyectos de automatización de procesos industriales propios del Centro. Desde hace algunos años en el SENA se ha implementado la formación por proyectos, por lo cual el desarrollo de ideas innovadoras y con contenido tecnológico se ha masificado buscando la creación de unos productos útiles para la comunidad en general.

Los estudiantes de Tecnología en Electrónica Industrial, en el momento de inicio de su experiencia, presenta un desarrollo del 67% de la etapa lectiva, correspondiente a 1300 horas de formación, con un proceso de enseñanza-aprendizaje-evaluación orientado a adquirir conocimientos, destrezas y habilidades en las áreas de electrotecnia, electrónica análoga, electrónica digital, fundamentos de programación, fundamentos de control, representación gráfica, controles y máquinas eléctricas, mandos neumáticos e hidráulicos, informática básica, inglés técnico y desarrollo y construcción de prototipos electrónicos. Hasta este momento los estudiantes evidencian competencias para el análisis de objetos

y sistemas, síntesis y construcción de prototipos, trabajo en equipo, uso efectivo de la información y comunicación en las aulas, talleres y laboratorios del centro mediante una formación rica en metodologías activas y con las componentes teórico-prácticas requeridas para la especialidad.

La metodología de trabajo por proyectos aplicada al restante 33% de la formación, incluidos los bloques modulares de Electrónica Industrial, Diseño y Desarrollo de Proyectos Electrónicos, Soluciones con Sistemas Microprocesados y Microcontrolados, Gestión de Mantenimiento se establecen según directriz del ingeniero Orlando David Orbes, mediante la propuesta y seguimiento al desarrollo de productos establecido por los estudiantes, quienes conformando equipos de alto desempeño, atienden los retos planteados para la generación de soluciones con énfasis en la repotenciación del Aula de Gestión y Desarrollo de Prototipos como vector clave establecido para el desarrollo del Área de Electrónica y Automatización del Centro Internacional de Producción Limpia de San Juan de Pasto.

Por otra parte, los estudiantes de Tecnología en Automatización Industrial bajo la orientación del Ingeniero Iván Vinuesa Córdoba, tienen un conocimiento amplio en diferentes áreas que les permiten encarar los proyectos finales sin ningún tipo de objeción y por lo cual se garantiza el aporte que éstos estudiantes puedan brindar no solo en las etapas de investigación y diseño, sino también en solución de problemas gracias a la experiencia que poseen tras la realización de sus prácticas durante el desarrollo del programa.

La herramienta que se maneja para la interfaz gráfica, el control y la instrumentación de los proyectos es la versión 7.1 de LabVIEW de la cual el SENA tiene licencia, adquirida recientemente, que la constituye en un producto para explotar y aprovechar.

2.1 LABVIEW

Este programa es creado por National Instruments. “LabVIEW es una herramienta diseñada especialmente para monitorizar, controlar, automatizar y realizar cálculos complejos de señales analógicas y digitales capturadas a través de tarjetas de adquisición de datos, puertos serie y GPIBs (Buses de Intercambio de Propósito General).

Es un lenguaje de programación de propósito general, como es el Lenguaje C o Basic, pero con la característica que es totalmente gráfico, facilitando de esta manera el entendimiento y manejo de dicho lenguaje para el diseñador y programador de aplicaciones tipo SCADA.

Incluye librerías para la adquisición, análisis, presentación y almacenamiento de datos, GPIB y puertos serie. Además de otras prestaciones, como la conectividad con otros programas de cálculo, en especial MATLAB. Está basado en la programación modular, lo que permite crear tareas muy complicadas a partir de módulos o sub-módulos mucho más sencillos. Además, estos módulos pueden ser usados en otras tareas, con lo cual permite una programación más rápida y provechosa¹. (Ver Anexo 2)

2.2 MICROCONTROLADORES

Hoy en día es común encontrar microcontroladores en las cafeteras, hornos microondas, videograbadoras, alarmas, automóviles, etc. Las aplicaciones son infinitas, el único límite es la imaginación. La posibilidad de manejar señales de entrada y de salida, así como su capacidad para procesar datos y tomar decisiones, lo convierten en uno de los elementos electrónicos más versátiles que existen.

“Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes o secuencias que están grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica dentro del ordenamiento del mismo y a su vez permiten obtener configuraciones diferentes. Esto es, se pueden diferenciar según el tamaño y cantidad de sus elementos básicos y características especiales.

En muchas ocasiones, se requiere algo más que los pines de entrada y salida para controlar algún proceso. Pensando en ello los fabricantes de microcontroladores han adicionado algunos componentes especiales en algunos de sus modelos. Las posibilidades son amplísimas y el usuario puede escoger a la carta:

- Algunos microcontroladores tienen conversores análogo a digital (A/D), en caso de que se requiera medir señales no digitales, como por ejemplo temperatura, voltaje, luminosidad, etc.
- Si se requiere medir períodos de tiempo entre eventos, generar temporizaciones o salidas con frecuencia específica, algo muy común en electrónica, se puede disponer de uno o varios temporizadores programables (timers).
- Cuando se necesita establecer comunicación con otro microcontrolador o con un computador se puede disponer de una interfaz serial RS-232.
- Para desarrollar una aplicación donde los datos no se alteren a pesar de que se retire la alimentación, se puede usar un microcontrolador con memoria

¹ LAZARO, Antoni y DEL RIO, Joaquín. LabVIEW 7.1 Programación Gráfica para el Control de Instrumentación. Madrid : Paraninfo S.A, 2005. p. 6.

EEPROM, que es un tipo de memoria ROM que se puede programar o borrar eléctricamente sin necesidad de circuitos especiales.

- Para quienes utilizan salidas PWM (modulación por ancho de pulso) en el control de motores DC o cargas resistivas, existen microcontroladores que pueden ofrecer varias de ellas.
- Cuando se requiere atender eventos en tiempo real o se tienen procesos que no dan espera, se debe utilizar la técnica llamada de Interrupciones. Cuando una señal externa activa una línea de interrupción, el microcontrolador deja de lado la tarea que se encuentra ejecutando para atender una situación especial y luego puede regresar a continuar con la labor que estaba desempeñando”². (Ver Anexo 3)

2.3 ADQUISICION DE DATOS

La adquisición de datos se convierte en el principal recurso usado hoy en día para poder controlar y/o automatizar un proceso específico de una planta industrial. Las señales provenientes de elementos sensores se necesita llevarla hasta el computador o un microcontrolador para poder analizarlas y dependiendo de unos requerimientos activar o no unos actuadores que vinculados al proceso van a hacer que el estado anterior de éste cambie o se mantenga en unas condiciones deseadas. Esta adquisición es posible gracias a unas tarjetas existentes hoy en el mercado y expuestas por gran variedad de fabricantes que pretenden por medio del puerto serial, paralelo, usb, llevar las señales hacia el microcontrolador, PLC (Controlador Lógico Programable) o computador para su control. (Ver Anexo 4).

2.4 CONTROL Y AUTOMATIZACIÓN

“El control automático ha jugado un papel vital en el avance de la ciencia y de la ingeniería. Además de su extrema importancia en vehículos espaciales, sistemas de guía de proyectiles, sistemas de piloto automático de aeronaves, sistemas robóticos y otros, el control automático se ha vuelto parte integral e importante de los procesos industriales y de manufactura modernos. Por ejemplo, el control automático resulta esencial en el control numérico de las máquinas herramienta en las industrias manufactureras. También resulta esencial en operaciones industriales como el control de presión, temperatura, humedad y viscosidad, y flujo en las industrias de transformación.

Como los avances en la teoría y práctica del control automático brindan medios para lograr el funcionamiento óptimo de sistemas dinámicos, mejorar la productividad, liberarse de la monotonía de muchas operaciones manuales

² DUQUE, Edison. Curso Básico de Microcontroladores PIC. Pereira: Compañía Editorial Tecnológica CEKIT S.A, 2002. p. 12.

rutinarias y repetitivas, y otras ventajas, la mayoría de los ingenieros y científicos deben poseer un buen conocimiento de este campo”³. (Ver Anexo 5)

³ OGATA, Katsuhiko. Ingeniería de Control Moderna. 2 ed. Mexico : Prentice-Hall Hispanoamericana, S.A., 1993. p. 2

3. METODOLOGIA

Ejecución de actividades de tutoría a los alumnos de las especialidades de electrónica industrial y automatización en el desarrollo de proyectos de innovación y desarrollo tecnológico.

En esta parte del trabajo se pretende desarrollar en primera instancia, recopilar información concerniente al manejo de proyectos dentro de la institución, proyectos asignados que tengan los estudiantes con sus respectivos requerimientos.

Después de ello se puede diseñar, junto con los estudiantes y los instructores respectivos, un cronograma del proyecto con actividades y fechas claras tanto para la investigación, diseño, implementación y documentación de su idea investigativa. En esta parte se debe tener en cuenta que los estudiantes necesitan asesoría en algunos temas necesarios tanto para el desarrollo del software como para la implementación del dispositivo a desarrollar. Para solventar estos requerimientos se asignan horas fijas semanales en las cuales se lleva a cabo tutoría de algunos temas específicos necesarios para los alumnos, siempre procurando un avance conjunto con el desarrollo de su proyecto.

En la parte del desarrollo de los proyectos, los estudiantes cuentan con computadores, uno cada dos personas, los cuales están dotados del software tanto de electrónica como de automatización adecuados para que puedan programar PIC's si van a utilizar estos dispositivos, o para que puedan programar en software desde el computador si así van a implementar su proyecto.

Se presta asesoría en cuanto a los diferentes lenguajes de programación utilizados, brindando a los estudiantes apoyo para que puedan sacar adelante sus proyectos y tengan la opción de elegir el software de control que más le convenga a cada caso.

En la parte de hardware igualmente se acompaña y orienta a los estudiantes en el diseño en computador de los circuitos, luego en la fase de simulación en diferente software de electrónica que el centro tiene a disposición, en el diseño de los circuitos impresos, y finalmente en la implantación de las tarjetas necesarias con los elementos que el SENA facilita a los estudiantes.

Se desarrollan pruebas, para lo cual es necesario estar presente y prestar apoyo activo a los estudiantes en el desarrollo de los proyectos. Todo esto lleva a apreciar que el tiempo que se dedica a estas actividades ocupa casi toda la jornada, más aún si se tiene en cuenta que se presta asesoría a estudiantes

tanto de Tecnología en Electrónica Industrial como de Tecnología en Automatización y que se manejan diferentes horarios de clases para cada curso. Por ello se hace necesaria la asignación de varias horas fijas semanales con cada grupo para avanzar en los proyectos y solucionar las inquietudes presentadas.

Ya específicamente hablando de los proyectos a desarrollar por parte de los estudiantes, se lleva a cabo el diseño y construcción de 5 diferentes proyectos que incluyen modelos de tarjetas de adquisición de datos, así como también sistemas de automatización de procesos de carácter agropecuario del centro. Las condiciones y requerimientos de las tarjetas ya están adelantados y en base a estas características se inicia a trabajar junto con los estudiantes. Las características de los proyectos se presentan a continuación:

- 1- Tarjeta de expansión digital para puertos paralelo/serial.
 - 8 entradas digitales CMOS con indicación de estado a través de led.
 - 16 salidas digitales CMOS con indicación de estado a través de led.
 - 2 entradas para contadores digitales.
 - Convertidores de señal:
 - 4 entradas tratadas por optoacoplador.
 - 3 convertidores AC – tren de pulsos digitales.
 - 2 generadores de pulsos DC – 10 Khz ajustables por potenciómetro.
 - 2 comparadores de nivel ajustables a través de potenciómetro 0-5V
 - 4 salidas a través de relevo.
 - 4 salidas a través de triac y potencia de 1 KW.

- 2- Tarjeta de adquisición de datos a través de puertos paralelo/serial.
 - 4 entradas analógicas 0 a 5V
 - 4 entradas analógicas 4-20mA
 - 2 salidas analógicas 0-5V
 - 2 salidas PWM
 - 4 salidas digitales a través de relevo
 - 4 salidas analógicas a través de control de fase
 - 4 filtros para salida PWM 10 Khz –1A
 - 2 circuitos generadores de referencia de precisión DC
 - Conexión al PC a través de cable con puerto paralelo y RS232 para puerto serial.

- 3- Interfaz de control para PC
 - 2 motores paso a paso
 - 1 servomotor de AC
 - 1 motor DC
 - Conexión al PC a través de cable con puerto paralelo y RS232 para puerto serial

4- Sistema de Automatización de horno ahumador

Reset:
Salidas con control de Triac:
Entrada Sensor de Puerta.
Entrada Sensor de humedad relativa.
Entrada Sensor de Temperatura
Circuito emisor
Circuito receptor
Teclado Matricial
LCD (Pantalla de Cristal Líquida)

5- Sistema para ambientes controlados para crianza de cerdos

Salida para control de un motor paso a paso
Salida para control de un motor DC.
Operación de 5v, 12v y 110v.
Seis entradas analógicas con protección contra transitorios utilizando varistores y un circuito integrado TL084.
Ocho salidas de potencia manejadas a través de relés para el manejo de los actuadores del sistema.
Microcontrolador PIC16F877A o PIC18F452, a 20MHz.
Puerto de comunicaciones RS-232.
Circuito por separado para el manejo de un motor de DC para el desplazamiento del dispensador para evitar calentamiento del sistema.

Los estudiantes de Tecnología en Electrónica Industrial son 20 en total. Están divididos en 5 grupos de 4 estudiantes cada uno y se les ha asignado un proyecto a cada grupo. Estos grupos a su vez están divididos en dos subgrupos de 2 estudiantes que se encargan de hacer la parte de hardware y software, respectivamente.

Todas las tarjetas van a estar controladas por microcontrolador y para esto se debe diseñar el software, así como también para la comunicación con el PC. Por lo tanto, es de vital importancia que se brinde un acompañamiento activo a los estudiantes a la hora de programar, revisar rutinas y aconsejar en cuanto a ventajas que se puedan obtener acerca de los problemas que ellos desarrollen.

En la parte de hardware se acompaña en el diseño de los circuitos de las diferentes tarjetas, el acople de los diferentes elementos, el desarrollo de pruebas, la ubicación de los elementos, su conexión y posterior implementación. En esta parte se apoya con consejos, resolución de dudas ó problemas y verificación del cumplimiento del trabajo. También se apoya en la parte de la construcción de los circuitos impresos.

Por su parte los estudiantes de tecnología en Automatización se encargarán del programa a desarrollar en el PC para que la interfaz de las tarjetas con el usuario sea la mejor posible y también colaborar en el programa a realizar para la comunicación con los microcontroladores, al igual que las etapas de configuración, control y adquisición. Se apoyará en la programación y la construcción de una interfaz gráfica agradable y fácil de manejar por parte del usuario, quién es el que finalmente va a darle uso a las tarjetas. Junto con los estudiantes, se realizarán instrumentos virtuales en LabVIEW con rutinas generales que puedan ser usadas por los cinco grupos de trabajo, así como se diseñarán conjuntamente con los estudiantes las rutinas específicas de cada tarjeta para la creación de la interfaz.

En el desarrollo de este proyecto se desarrollará un sistema operativo en tiempo real RTOS a ser usado en el control del llamado de las tareas, buscando un buen desempeño cuando éstas sean implantadas en el programa general. Esta parte se realizará junto con algunos estudiantes de tecnología en Electrónica Industrial encargados de realizar rutinas que van a servir para constituir el sistema operativo.

La ayuda a brindar se centrará en la programación de microcontroladores, así como en el diseño de circuitos electrónicos básicos como amplificadores, comparadores, detectores de señales, entre otros. Posteriormente, se realizarán pruebas tanto de simulación de software como de hardware con pequeños montajes. La idea en general es ofrecerles un acompañamiento continuo en el desarrollo de los proyectos emprendidos desde los bloques modulares en ejecución.

Desarrollo de actividades de mantenimiento de los laboratorios de Electrónica y Mecatrónica y puesta en ejecución del aula de Gestión y Desarrollo de Prototipos (Aula Abierta)

Cabe destacar que se desarrollará el mantenimiento de los equipos informáticos del laboratorio de Mecatrónica del Centro Multisectorial Lope, dentro del cual se realizarán actividades de formateo de discos duros, instalación de sistemas operativos, software de la planta de mecatrónica que tiene el SENA, software del brazo robótico de Bosch con el que también cuentan, actualización de controladores de dispositivos de hardware, además del establecimiento y configuración de una red entre los equipos del aula (6 en total) asignando a uno de ellos la autonomía de ser el maestro, y conectarlos con el controlador o PLC de la planta mecatrónica y realizar pruebas para verificar el correcto funcionamiento.

Así mismo, se desarrollará mantenimiento a los equipos que tienen en el Aula de Gestión y Desarrollo de prototipos (Aula Abierta) a los cuales se les instalará software de diferentes tipos como LabVIEW 7.1, se les actualizarán los drivers, se harán pruebas de diferentes dispositivos y poco a poco se pondrá a pleno ésta aula para que los estudiantes tengan un espacio donde investigar y avanzar en el

desarrollo de sus proyectos emprendidos. Esta aula se piensa seguir equipando, con las tarjetas de adquisición de datos que se van a realizar y con unos bancos de trabajo, para lo cual también es necesaria la ayuda o aporte a la hora de hacer conexiones, poner a correr el software utilizado para las tarjetas, instalar el hardware e instruir a los alumnos con normas de seguridad en pro de la buena utilización de los equipos.

También, se hará mantenimiento al aula de laboratorio de electrónica en la cual existen equipos no solo informáticos, sino también electrónicos como lo son osciloscopios, entrenadores de microcontroladores, entrenadores de PLC's, entre otros.

Igualmente, se colaborará en la configuración y puesta en operación de la red para el sector agroindustrial, que servirá para garantizar una buena conexión a Internet en ésta área; así como también la conexión en red de los equipos de todas las aulas que lo componen y cuentan con equipo informático.

Desarrollo conjuntamente con los estudiantes de las tecnologías en Electrónica y Automatización, de hardware y rutinas en LabVIEW para la adquisición de datos y control de diferentes proyectos o prototipos.

Se desarrollarán rutinas de programación en LabVIEW junto con los estudiantes que sirvan para la adquisición de datos digitales, análogos, para la comunicación a través de puerto serial, paralelo, así como la creación de sistemas de control de diferentes prototipos que realicen los estudiantes, sobre procesos de carácter agropecuario o industrial. Se pretende utilizar esta poderosa herramienta de programación y control para poder manipular y controlar desde el computador los proyectos que se lleguen a implantar.

Así mismo, se realizarán rutinas para la divulgación de esta herramienta como una de las más usadas en la instrumentación y el control procurando que los estudiantes y demás personas se interesen por apropiarla para el control de sus proyectos de aplicación.

Una aclaración importante en ésta parte, es que el desarrollo de las actividades de la pasantía se desarrollarán durante seis meses, tiempo completo o el equivalente en tiempo.

4. DESARROLLO DEL PROYECTO

A continuación se presenta la forma como ha sido desarrollado este proyecto, los recursos empleados, y los resultados obtenidos. Cabe aclarar que como las actividades de pasantía han sido encaminadas por el lado de asesoría en proyectos de innovación y desarrollo tecnológico y debido además que los responsables directos de estos proyectos son grupos de estudiantes, todos los resultados no serán incluidos, se representarán por tanto, algunas tareas realizadas en las que se ha aportado mayor tiempo y esfuerzo.

El desarrollo de las actividades de pasantía se iniciaron el día lunes 04 de septiembre de 2006 y se prolongaron hasta el viernes 23 de marzo del año en curso con una intensidad horaria de 42 horas semanales, lo cual corresponden a un período de seis meses tiempo completo, como está estipulado en el acuerdo, en la sección que contempla esta posibilidad de pasantía para optar por el título de Ingeniero Electrónico de la Universidad de Nariño. Estas actividades fueron desarrolladas en el Centro Multisectorial Lope SENA Regional Nariño, con estudiantes de Tecnología en Electrónica Industrial y Tecnología en Automatización. En total fueron 33 estudiantes de las dos tecnologías quienes se dividieron en grupos de trabajo para el diseño y posterior desarrollo de los proyectos planteados. Algunos de estos proyectos a los cuales se hará referencia más adelante, fueron planteados por los instructores de las dos tecnologías y otros surgieron por inquietud de los propios estudiantes al hacer un recorrido por el centro en cuestión.

Se asistió a algunas clases para familiarizarse con los temas que los estudiantes recibían, para reforzar si en algún momento esto fuese necesario. Se pudo ver que los alumnos de Tecnología en Electrónica Industrial estaban aprendiendo a programar en MPLAB y que se les habían asignado el proyecto de una tarjeta de adquisición de datos, con diferentes requerimientos para cada grupo. Por otra parte, los estudiantes de Tecnología en Automatización estaban aprendiendo a programar en LabVIEW e igualmente se les asignó el trabajo de crear la interfaz gráfica con el usuario de los proyectos de las tarjetas de adquisición; una determinada tarjeta a cada grupo de trabajo.

Los proyectos emprendidos por los estudiantes y los cuales se mantuvieron hasta el final para su desarrollo junto con sus características más relevantes se describen enseguida:

Sistema de Adquisición de Datos Analógicos didáctico.

Al desarrollar un sistema de adquisición de datos a través de puertos del PC se desea conocer algunas de las funciones más importantes en el proceso y

almacenamiento de la información. Cuando se desarrolla un sistema de adquisición de datos se tienen componentes de hardware dedicados a ciertas funciones, como son el ingreso de datos analógicos, control de salidas analógicas y digitales, y todo lo que concierne al manejo de puertos y señales de entrada y salida. Este sistema de adquisición permitirá realizar funciones como monitoreo de procesos y aplicaciones de control, un ejemplo de esto pueden ser sistemas de monitoreo en laboratorios, para clases en universidades, procesos de control industrial, etc.

El reto está en diseñar e implementar el sistema completo consistente en una tarjeta de adquisición de datos y un PC que controla la operatividad del sistema. Dicho sistema abarca aspectos hardware tales como la tarjeta de adquisición de datos basada en un microcontrolador conectada al PC vía RS-232 a través del puerto serie, y aspectos software como es la programación de la propia tarjeta y el desarrollo del instrumento virtual. El instrumento virtual va a ser un módulo software que representa el panel frontal y constituye la interfaz gráfica interactiva con el usuario. La tarjeta posee:

- 4 entradas analógicas 0 a 5V
- 4 entradas analógicas 4-20ma
- 2 salidas analógicas 0-5V
- 2 salidas PWM
- 4 salidas digitales a través de relevo.
- 4 salidas a través de Triac.
- 4 filtros para salida PWM 10Khz
- 2 circuitos generadores de referencia de precisión DC (0 – 5 V).
- Conexión a PC a través de cable por puerto serial RS-232
- Conexión a PC a través de cable por puerto paralelo DB 25

Los estudiantes de tecnología en Electrónica Industrial que están encargados del desarrollo de este proyecto, son: Alexander Benavides, Oscar Mesías, Ricardo Palacios y Armando Rosales. Ellos recibieron apoyo por parte de dos estudiantes de Tecnología en Automatización respecto al desarrollo de una interfaz gráfica en LabVIEW para la tarjeta.

Sistema de desarrollo para la adquisición y control digital didáctico microprocesado.

Esta tarjeta representa un sistema de expansión digital para la adquisición de datos análogo - digital; diseñado para mejorar el desarrollo de nuevos prototipos electrónicos que aporten solución a la automatización de diferentes procesos y a la creación de nuevas tecnologías. Este proyecto consiste en crear un banco de pruebas útil para el montaje y simulación de prácticas y conceptos electrónicos. Con la creación de este sistema didáctico para el desarrollo de nuevas

tecnologías se pretende que los estudiantes de electrónica tengan un mayor grado de conocimiento de forma rápida, eficiente y real.

La tarjeta posee:

- 25 pines configurables como entradas o como salidas digitales
- 4 entradas para contadores digitales

Estas entradas y salidas son módulos acoplables a los pines configurables desde PC

- Entradas tratadas por opto-acoplador
- Convertidores AC a tren de pulsos digitales
- Generadores de pulsos DC PWM
- Comparadores de nivel
- Salidas a través de relevo
- Salidas a través de triac
- Conexión a PC a través de cable por puerto serial RS-232
- Conexión a PC a través de cable por puerto paralelo DB 25

Igualmente, este proyecto tiene cuatro estudiantes de Tecnología en Electrónica Industrial responsables: Carlos Rodríguez, Jhon Rosero, Richard Ramos y Christian Bermeo. Así mismo, hicieron parte de este proyecto estudiantes de Tecnología en Automatización por cuanto colaboraron en la realización de la interfaz gráfica en LabVIEW.

Sistema de control de Motores por PC:

Se pretende diseñar y realizar una tarjeta electrónica para conectar al PC y con ello crear un banco de pruebas útil para el montaje y simulación de proyectos de los alumnos, con el fin de controlar el funcionamiento de dos motores paso a paso PAP, y dos motores DC (Corriente Continua). A través de esta tarjeta se desea permitir el control de algunos dispositivos mediante el PC y llevar un a revisión constante de su estado. La tarjeta posee:

- 12 pines configurables como entradas o como salidas digitales
- Conversores de Señal
- Entradas digitales.

Estas entradas y salidas son módulos acoplables a los pines configurables desde PC

- Entradas tratadas por opto acoplador
- Convertidores AC a tren de pulsos digitales
- Generadores de pulsos DC
- Comparadores de nivel
- Conexión a PC a través de cable por puerto serial RS 232

Comunicación serial: Dentro de sus funciones está en primer lugar recibir y enviar las señales provenientes del PC y del sistema inteligente a través del cableado y la circuitería.

Cerebro electrónico: Es el encargado de almacenar todos los programas, y datos en su memoria, además de enviar las señales electrónicas requeridas para poner en marcha el funcionamiento de los distintos dispositivos, adecuar las señales externas que recibe del usuario y de las señales dadas por sensores conectados al medio que lo rodean, y así ejecutar las más apropiadas decisiones.

Protección: Mediante un dispositivo se puede aislar la etapa de potencia con las entradas o salidas del sistema inteligente, este sistema se utiliza para evitar que esta última sufra daños ya que el sistema internamente no tiene un control individual de protección.

Etapa de potencia: En esta parte de la tarjeta se reciben señales externas de todo tipo, las cuales son acondicionadas para suministrarlas al sistema inteligente y que posteriormente son adecuadas para la activación de dispositivos externos, o actuadores.

Funciones sistemas Software: A través de una aplicación desarrollada en el computador se puede configurar los pines de la tarjeta inteligente como entradas o salidas; visualizar el estado en que se encuentra cualquiera de las entradas; enviar señales que a su vez activan cualquier etapa de salida; entender las señales que son suministradas y tomar decisiones; ser un medio de interacción entre el usuario y el sistema de tal forma que las decisiones tomadas sean comprensibles para los dos.

Este proyecto en primera instancia fue emprendido por cuatro estudiantes, pero finalmente únicamente Iván Palacios quedó a cargo del desarrollo de esta tarjeta de adquisición de datos. Similar a lo ocurrido en los proyectos anteriormente descritos, estudiantes de Tecnología en Automatización Industrial apoyaron en la parte del diseño y desarrollo de la interfaz gráfica en LabVIEW.

Sistema de automatización de horno ahumador:

El horno de ahumado se encontrará ubicado en el área de cárnicos del centro LOPE SENA; ésta máquina permite al usuario realizar ahumado y secado de carne dependiendo de la exigencia del consumidor. El Horno de ahumado está provisto de diferentes dispositivos digitales y análogos, que en conjunto con un microprocesador se encargan de realizar las diferentes tareas que serán asignadas por el operador a través de un teclado ubicado en el panel principal del horno. Este dispositivo cuenta con una serie de elementos actuadores que son los encargados de realizar las acciones solicitadas; además, se implementa sensores

en puntos estratégicos; estos tienen como tarea indicar el estado del proceso en tiempo real.

El sistema cuenta con un sistema microcontrolado que es el encargado de direccionar las diferentes señales entregadas por todos los dispositivos que conforman la máquina; dicho sistema está interconectado con los módulos actuadores, estos módulos reciben y entregan señales al microcontrolador para que éste a su vez tome las decisiones pertinentes para dicha señal. Tomando como referencia este conjunto de señales el cerebro del sistema es el encargado de coordinar todo el proceso hasta su finalización. A continuación se describen las partes del horno ahumador:

- Cuerpo del ahumador
- Panel de control digital (LCD/TECLADO)
- Rejillas para la carne
- Generador de humo
- Bandeja para goteo
- Chimenea, regulador de aire
- Generador de aire

La tarjeta que se va a realizar para el control del horno ahumador contiene los siguientes módulos:

- Reset: restaura los parámetros a sus valores por defecto
- Salidas con control de Triac: encargado de controlar los actuadores
- Sensor de Puerta.
- Sensor de humedad relativa.
- Sensor de Temperatura
- Circuito emisor
- Circuito receptor
- Teclado Matricial
- LCD (Pantalla de Cristal Líquida)

Los responsables de sacar adelante este proyecto, son: Luís López, Ivonne Narváez, Alfredo Chávez y Giovanni Martínez, todos ellos estudiantes de Tecnología en Electrónica Industrial.

Sistema para ambientes controlados para la crianza de cerdos.

El sistema de control sería capaz de recibir y enviar señales desde y hacia el ambiente en el cual se encuentra trabajando, y además de las señales provenientes de los requerimientos del usuario, para el funcionamiento de este tipo de sistema; con esto, se quiere decir que el usuario podrá configurar las principales funciones del sistema como lo son: el control de temperatura, el control

del subsistema de dosificación de alimento y el subsistema de aseo de cocheras. Cada subsistema será regido por un tipo de señales, las cuales serán señales enviadas por sensor para el caso del control de temperatura y señales de temporización para los otros dos subsistemas. Debido a que el microcontrolador tiene integrados temporizadores, estos se utilizarán y la señal del sensor deberá ser acondicionada para su posterior análisis. Además de estas funciones el microcontrolador realizará un control sobre el funcionamiento de los motores que se encargan de la mecánica del dosificador de alimento.

A continuación, se presentan las características que tiene el sistema a desarrollar:

- Salida para control de un motor paso a paso y un motor DC.
- Operación de 5v, 12v y 110v.
- Protección contra inversiones de polaridad.
- Posibilidad de alimentarla con una sola fuente de energía o con fuentes separadas para motores y circuitos de control.
- Seis entradas analógicas con protección contra transitorios utilizando varistores y un circuito integrado TL084.
- Ocho salidas de potencia manejadas a través de relés para el manejo de los actuadores del sistema.
- Microcontrolador PIC16F877A o PIC18F452, a 20MHz.
- Puerto de comunicaciones RS-232.
- Interfaz grafica para la configuración y control del ambiente ejecutado en LABVIEW.
- Circuito por separado para el manejo de un motor de DC para el desplazamiento del dispensador para evitar calentamiento del sistema.
- Dimensiones: 165 x 150 mm.
- Peso: 145 gramos aproximadamente

Este proyecto fue adelantado por los siguientes estudiantes de Tecnología en Electrónica Industrial: Wilson Guerra, Ricardo Muñoz, Jesús Andrade y Esteban Anganoy. Estudiantes de Tecnología en Automatización aportaron en la constitución de la interfaz gráfica en LabVIEW realizada para este prototipo.

En primera instancia se recolectó información general de los requerimientos de los proyectos en tiempo y aplicaciones, así como los recursos con los cuales se cuentan para el desarrollo. Se estableció para cada proyecto la disponibilidad de tiempo y se planearon unas horas para una fundamentación importante para la realización de la interfaz gráfica en un programa de computadora como lo es LabVIEW. En esta parte es bueno aclarar que el SENA ha adquirido la licencia de la versión 7.1 de este programa con su paquete completo de librerías.

Así mismo, se cuenta con simuladores de electrónica para la parte de hardware como lo es la herramienta Multisim 9 de la cual también se posee una licencia.

En la parte de programación de microcontroladores se usa la herramienta MPLab (Ver Anexo A.3.5) la cual se la descarga libremente desde Internet de la página de Microchip. Igualmente, para esta parte se usa el programador de Microchip PICSTART Plus, existente en el Centro y con el cual se programan los microcontroladores con las rutinas realizadas por los estudiantes.

La idea en esta parte de la tutoría siempre ha sido que los estudiantes aprendan a programar por ellos mismos y que entiendan lo que tienen que hacer y cómo lo están haciendo, que aprendan para qué sirven y cuándo usar las funciones y registros pertenecientes al PIC16f877, el cual es el microcontrolador con el que están trabajando; por ello, la tutoría se extiende buscando que aprendan a diseñar rutinas en lenguaje ensamblador, para que posteriormente en el desarrollo de sus proyectos encuentren un camino más fácil y cómodo de controlar sus prototipos.

Para la parte de circuitos impresos, que igualmente se ha elaborado en el SENA, con la activa participación del instructor Argemiro Urbano, se utiliza la herramienta de computador PCB Wizard para diseñar la forma de la tarjeta, el tamaño de la misma, y la disposición que van a tener cada uno de los elementos que la componen. Esta herramienta permite realizar un enrutamiento de una manera más práctica y a escala real.

El encargado general de los proyectos en cuestión es el Ingeniero Orlando Orbes, instructor del área de Electrónica del SENA y con él se acordaron las actividades a seguir para estos proyectos. En seguida, se detallan y explican de manera amplia cada una de las actividades desarrolladas en esta pasantía.

4.1 DESARROLLO DE RUTINAS COMUNES A LOS PROYECTOS

Se acordó realizar unas rutinas comunes que sirvan a la mayoría de los grupos de trabajo para sus proyectos en la parte de programación o software. Estas rutinas tienen tanto parte de programación en microcontrolador o PIC, como programación en el programa de computador LabVIEW que se eligió para crear la interfaz gráfica con el usuario, buscando un ambiente más agradable y óptimo a la hora de responder ante un sistema de adquisición de datos y/o señales (Ver Anexo 2)

4.1.1 Rutinas en mplab. En las rutinas a realizar con microcontroladores en lenguaje ensamblador se han realizado algunas rutinas comunes a la mayoría de los proyectos de adquisición de datos y automatización de procesos. Se trabajó en esta parte con varios grupos de estudiantes con quienes se pudo desarrollar estas rutinas y que a la postre beneficien a todos los equipos de trabajo. Es así como se desarrollaron rutinas tales como un planeador cíclico basado en un sistema multitarea en tiempo real RTOS, se analizaron las variables que se buscaban con su ejecución y de acuerdo a los requerimientos respectivos para que el sistema pueda responder desde las tarjetas en tiempo real ante señales provenientes del

exterior, permitiendo además con el sistema multitarea, tener un procesador que a semejanza de un computador dirija tareas específicas, cada determinado tiempo (1 milisegundo para este caso), en pro del buen uso de la memoria del microcontrolador y de la realización de diferentes rutinas en el mismo instante.

En resumen, este sistema pretende acoger e integrar todas las rutinas y subrutinas que se están desarrollando para cada una de las tarjetas con el fin de direccionar y controlar su ejecución en un momento determinado donde sea estrictamente necesario, para evitar que se estén presentando gran cantidad de ciclos que provoquen un posible desbordamiento de pila al microprocesador, lo cual sería un grave inconveniente a la hora de una aplicación con los prototipos. Este programa se inició a desarrollar con un estudiante de Tecnología en Electrónica Industrial y posteriormente, junto con el Ingeniero Orlando Orbes se concluyó de realizar este programa y se terminó de documentarlo para que los estudiantes lo acojan y usen en sus proyectos, simplemente integrando sus rutinas específicas al espacio definido para incluirlas, determinado por las diferentes tareas: tarea0, tarea1,... tarea15. En el Anexo F se presenta el código realizado con su correspondiente explicación en su parte derecha.

Igualmente, se colaboró con la realización de unas rutinas de comunicación vía puerto serial a través del conector RS-232, la cual busca que una cadena de bits se empaqueten en bytes y puedan ser enviados y recibidos hacia y desde el computador. Primero con dos estudiantes se analizó el procesador a utilizar, sus características, se dio una corta explicación de los registros que este posee, especialmente los registros RCSTA, TXSTA que contiene el PIC 16f877. Luego de esta fase se empezó a diseñar unas rutinas cortas que sirvieran para la comunicación en sus niveles de transmisión y recepción. El código de esta rutina de comunicación serial se incluye en el Anexo F.

Además de ello, se brindó tutoría para la culminación de otras rutinas por parte de los estudiantes referentes a la transmisión de datos, exploración de entradas y salidas digitales/análogas, de las cuales no se incluye el código, respetando la autoría de sus creadores y reconociendo sí la participación en el desarrollo de las mismas. La tutoría prestada tiene que ver con la ilustración del PIC 16f877, sus registros, las instrucciones que éste maneja y cómo poder utilizarlas para las aplicaciones concretas. Se ilustró en la forma de realizar ciclos, llamado a subrutinas y llamado a subprogramas.

4.1.2 Rutinas en labview. Cuatro de los cinco grupos encargados de los proyectos decidieron realizar una interfaz en LabVIEW para el usuario y la tarjeta que desarrollaron, de tal manera que la persona que vaya a manipularla encuentre una agradable y fácil herramienta de manejo desde el computador. Por ello, se optó por utilizar este software de programación para adquirir las señales, analizarlas, visualizarlas, tomar decisiones y enviar hacia el microcontrolador señales que puedan ser aplicadas en las tarjetas por los diferentes tipos de

actuadores que éstas disponen. Igual que para el caso de las rutinas en MPLab, se desarrollaron rutinas en LabVIEW o SubVI's junto con los estudiantes de Tecnología en Automatización, que puedan ser utilizados en diferentes secuencias de un mismo programa y/o en diferentes programas.

Las subrutinas que se diseñaron comunes a todos los grupos de trabajo están encaminadas a facilitarles las pruebas de comunicación serial, buscando hacer unos SubVI's que hagan la comunicación enviando una respuesta al computador para ver si la tarjeta efectivamente está respondiendo y tanto el PIC como el programa de computador tienen sus programas corriendo eficientemente. En primera instancia se llevaron a cabo unas subrutinas para transformar los datos de arreglos booleanos a un tipo de dato que es posible enviarlo por el puerto serial del computador utilizando los comandos VISA Write, para lo cual fue necesario explicar a los estudiantes el tipo de variables que maneja LabVIEW, los instrumentos virtuales existentes y que se utilizan para la comunicación serial, la forma de configurar estos instrumentos virtuales y su conexión. De igual forma, se realizó la rutina que hace el proceso contrario y convierte el tipo de dato proveniente de la comunicación serial a un arreglo booleano que es la forma como se estableció en el protocolo inicial el envío de la información. En las figuras 1 y 2 se muestran los Sub VI's indicados.

Figura 1. Sub VI arreglo a serial.

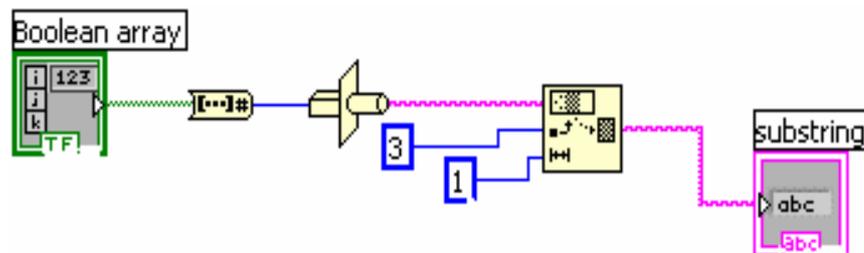
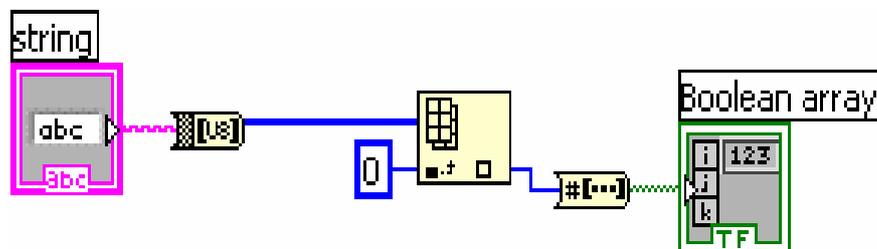


Figura 2. Sub VI serial a arreglo.



Para hacer la primera conversión se usa la herramienta type cast que convierte el tipo de dato como se ha descrito, además de un arreglo que selecciona un byte a partir del tercer paquete de información puesto que los dos primeros siempre van a llevar información propia de los elementos de comunicación VISA de LabVIEW. Para el segundo caso, se convierte la información que llega en forma de cadena a arreglo de byte, posteriormente, con el index array se convierte; se elige el primer byte que llega y se lo convierte a arreglo booleano que ya trae la información de acuerdo al protocolo especificado. Todos estos elementos utilizados se explicaron a los estudiantes y se buscó aplicabilidad para su posterior uso.

Posterior al desarrollo de estos SubVI's, junto con el Ingeniero Orlando Orbes, se creó una nueva subrutina para probar la comunicación serial primero entre dos computadores y luego entre el computador y las tarjetas de adquisición desarrolladas. Este programa envía un byte hacia la tarjeta y dependiendo si la información es correcta o no, la tarjeta o el otro PC para el primer caso; le devuelve el mismo dato que envió. A este programa se le puede configurar la velocidad de transmisión (57600 baudios para este caso), así como la paridad, los bits de parada y control de flujo. Además, ésta rutina guarda un historial de los diez últimos datos que le han llegado al computador teniendo la posibilidad de resetearlo cuando el usuario lo elija. En las figuras 3 y 4, se muestran el diagrama de bloques y el panel frontal de este Sub VI.

Figura 3. Diagrama de bloques sub VI interfaz serial

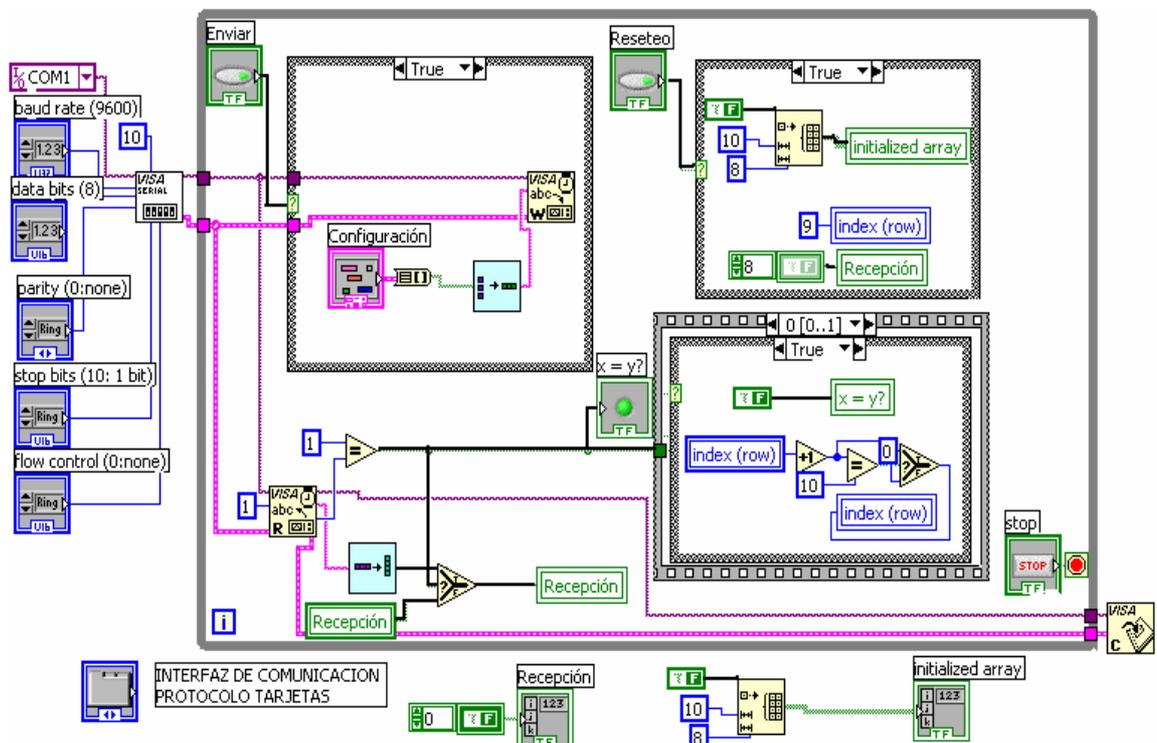
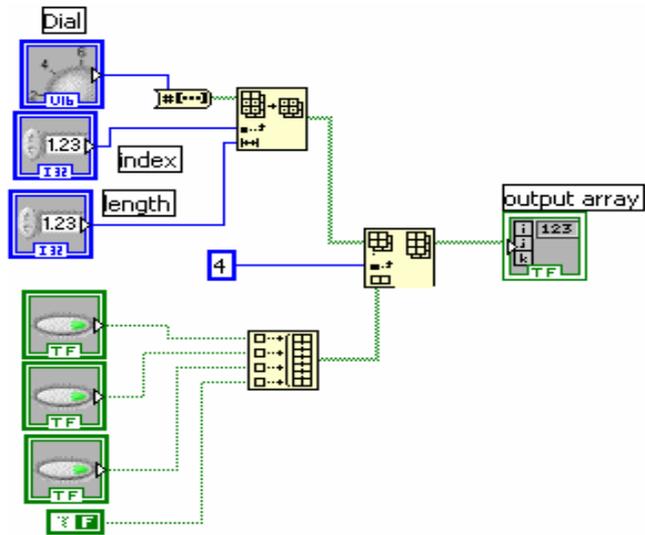


Figura 4. Panel frontal sub VI interfaz serial



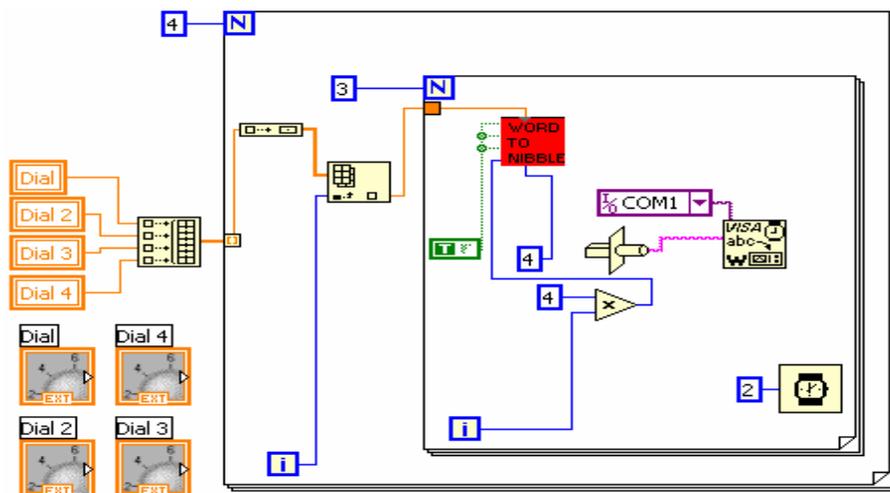
Se acordó con los estudiantes que la interfaz a desarrollar en LabVIEW entre el usuario y las tarjetas se pretende que tenga tres módulos básicos. El primero de ellos es la configuración dentro de la cual el usuario puede hacer una configuración automática o manual. Si la hace automática se habilitan todas las entradas y salidas existentes para cada tarjeta. Si se hace manual, se habilitan o no las entradas de sensores y salidas de actuadores que el usuario determine en un momento determinado. El segundo módulo tiene que ver con el control de los actuadores y salidas que posee cada tarjeta. Se tiene el control de salidas digitales o control on-off, el control de salidas analógicas como es el caso de las salidas de los motores que se los puede controlar por PWM o por ángulo para los paso a paso, salida de control de fase o voltaje. Justamente, para esta última parte que es la salida de señales análogas de la tarjeta, se debe enviar la información en diez bits y dado que el protocolo establece, que únicamente por cada paquete se envían cuatro bits de información, es necesario desempaquetar la palabra que contiene la información en tres paquetes y así poder enviarla. Es por ello que se ha constituido otro Sub VI para hacer el cambio de palabra a nibble (cuatro bits), el cual desempaqueta en tres bytes y envía la información iniciando con los cuatro bits menos significativos, siguiendo con los siguientes cuatro bits y por último envía los cuatro bits más significativos de la información. La figura 5, muestra el diagrama de bloques de esta subrutina.

Figura 5. Diagrama de bloques sub VI word a nibble



Para realizar este instrumento virtual se explicaron todas las funciones que poseen los arreglos, buscando que los estudiantes apropien este conocimiento y diseñen un Sub VI sencillo que cumpla la función que se está buscando. Este programa devuelve un nibble proveniente de una entrada numérica. Sin embargo, se desarrolló otro sub VI que desempaqueta tres, o más (cuatro para este caso) datos numéricos de diez bits de información cada uno. La forma como se lo hace es utilizando dos ciclos for que contienen: el primero, el número de nibbles que contiene cada dato (3) y el otro el número de entradas. Aquí se brindó tutoría en la programación estructurada, orientando a los estudiantes en las diferentes estructuras que posee el programa, su aplicación. La figura 6, muestra esta subrutina.

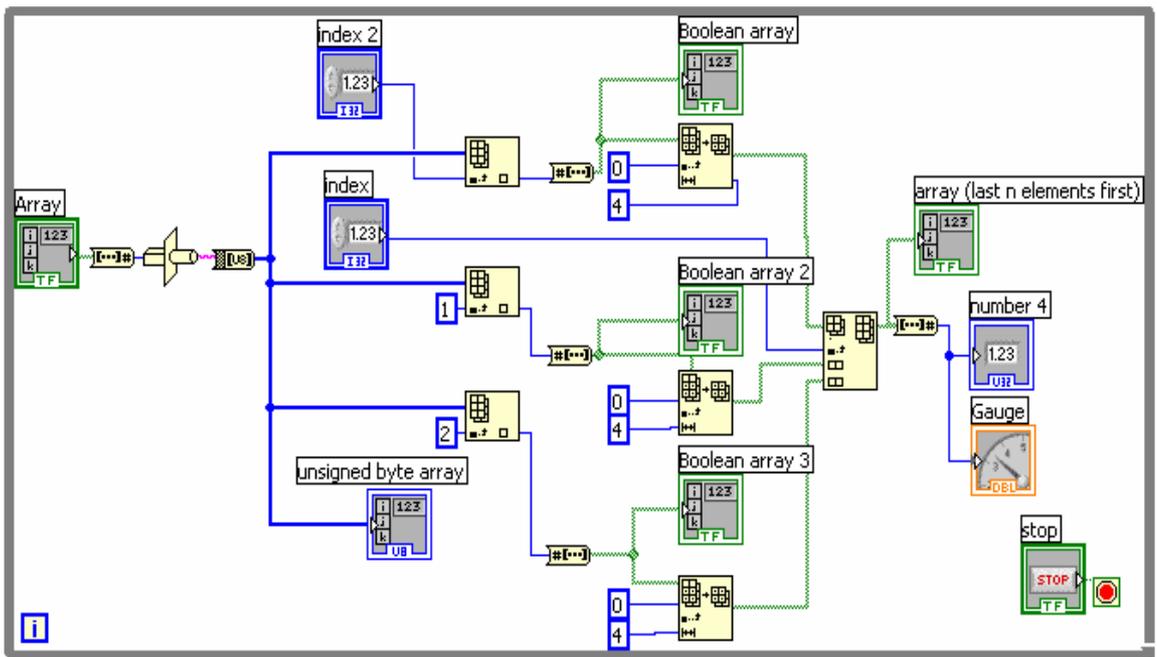
Figura 6. Diagrama de bloques ciclo word a nibble.



El tercer módulo del que se habla que contiene la interfaz en LabVIEW de las tarjetas, corresponde a la adquisición. Para adquirir se debe desarrollar el proceso contrario al control, es decir, los datos analógicos vienen desempaquetados en tres bytes y para poder obtener la señal se debe empaquetar la información.

A lo largo de esta sección se ha hablado frecuentemente de empaquetamiento y desempaquetamiento; esto es, se hace referencia a unir o desintegrar los bytes que provienen de la tarjeta y según un protocolo que se ha establecido con los estudiantes, para cada tarjeta, se puede constituir la información que sirve para la adquisición de datos o el control de los prototipos. Los estudiantes tienen claro ese protocolo y con base en éste, se desarrolla la programación en LabVIEW. Ese empaquetamiento ha llevado a construir otra subrutina útil para todos los grupos de trabajo. Lo que hace este Sub VI es tomar los cuatro bits menos significativos de tres paquetes y unirlos para formar el número correspondiente a la señal analógica que se está adquiriendo. En la Figura 7, se representa este Sub VI.

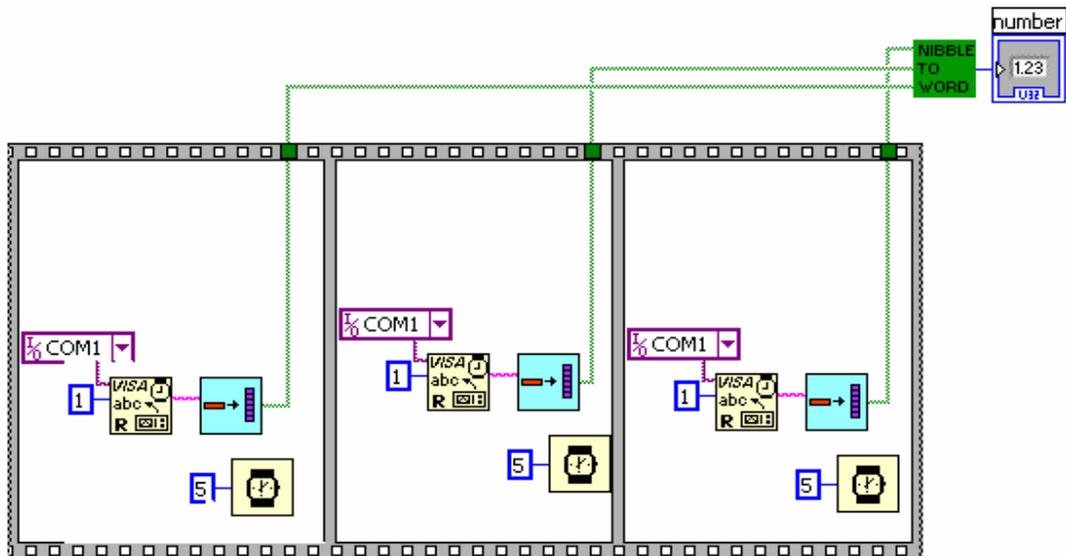
Figura 7. Diagrama de bloques nibbles a word



El paso siguiente, es crear un subprograma capaz de tomar la señal de cada canal y visualizarla en un indicador gráfico cualquiera, dependiendo de la aplicación que se esté llevando a cabo.

Para esta parte y debido a que esta rutina es muy común se decidió realizar el siguiente Sub VI, mostrado en la figura 8, que adquiere la señal de cada canal y la visualiza, utilizando para ello la subrutina de Nibbles a Word creada anteriormente.

Figura 8. Diagrama de bloques adquisición canal analógico.



4.2 DESARROLLO DE RUTINAS EN MPLAB ESPECÍFICAS

En cada uno de los proyectos se colaboró en la creación de rutinas específicas tanto de prueba como rutinas finales para la tarjeta. Rutinas de prueba que sirven para verificar el correcto funcionamiento de los módulos de la tarjeta en su parte de hardware y su comunicación con el computador. Se realizaron pruebas con el planeador descrito anteriormente, con cada una de las tarjetas para verificar su funcionamiento y poder adaptarlo haciendo modificaciones para cada caso. En seguida, se describen actividades de asesoría relacionadas con rutinas realizadas para cada uno de los proyectos.

4.2.1 Rutinas para sistema de adquisición de datos analógicos didáctico: se brindó asesoría correspondiente a rutinas de exploración de señales análogas, su almacenamiento en buffer de memoria del microcontrolador y rutinas de análisis y toma de decisiones, en cuanto a direccionamiento de tareas y organización de un proyecto para todas las rutinas de la tarjeta. En la parte de pruebas, se colaboró en una rutina de activación de relés controlados con una base de tiempo. En el Anexo F, se presenta el código para dicho programa.

Igualmente, se contribuyó en la realización de una rutina para probar las salidas PWM de la tarjeta, pudiéndose verificar el correcto funcionamiento de esta parte del hardware. En el anexo correspondiente a los códigos, se presenta el

relacionado con el envío de señales por los pines del puerto correspondiente del microcontrolador y establece una secuencia en los movimientos del motor controlado por temporización realizada a base de contadores.

Todas las rutinas desarrolladas con los estudiantes de este equipo de trabajo se realizaron posteriormente a una fundamentación. Se analizaron variables de entrada, variables de salida, resultados esperados y con esa información y los conocimientos en lenguaje ensamblador de los estudiantes, se inició con el desarrollo de éstas rutinas. Constantemente, se atendieron inquietudes de los estudiantes en el momento de la programación y posteriormente en la depuración de errores.

4.2.2 Rutinas para sistema de control de motores por computador: en esta tarjeta de control de motores, junto con el estudiante encargado de este proyecto, se realizó un trabajo más grande e importante, por cuanto los escasos conocimientos en programación en lenguaje ensamblador de los integrantes del grupo requerían un mayor apoyo e inducción. Se asesoró y colaboró en la realización de todo el programa en general de control de dos motores paso a paso y dos motores de corriente directa. A los motores paso a paso se les puede controlar el sentido de giro, el tipo de paso que van a llevar, si es medio paso o paso completo; además del tipo de control, si es por posición o por velocidad. Para el desarrollo de estas rutinas se inició primero con una fundamentación general acerca de la programación en lenguaje ensamblador. Se trabajaron con los estudiantes de este equipo, ejemplos sencillos y se dejó para programar rutinas cortas y sencillas. Con el paso del tiempo y a medida que el estudiante adquiría experiencia se completaron las rutinas del control de motores. Se prestó asesoría en cuanto a aspectos de generación de PWM por hardware desde el PIC 16f877, así como secuencias, llamada a subrutinas, subprogramas, ciclos a base de contadores, uso de los temporizadores con que cuenta el microprocesador, entre otros aspectos.

En el control por velocidad se puede elegir entre revoluciones por minuto y revoluciones por segundo, aclarando que este tipo de motores no están diseñados para trabajar a grandes velocidades y su principal aplicación la tienen en el control por posición. En el control por posición se puede elegir un control en lazo abierto, en el cual el control se dirige directamente al computador, un control en lazo cerrado en el cual la señal se retroalimenta y el control se lo hace desde el microcontrolador o como servomotor en el cual es acoplado un encoder al motor y se envía esta señal al computador para realizar un control automático de su posición. En primera instancia, se colaboró en la realización de una rutina que una vez le han llegado los datos, éstos se almacenan en una posición de la memoria del microcontrolador y se activan unas banderas para que en análisis y toma de decisiones se dirija a extraer los datos de la memoria y se dirija a realizar el control requerido. Para poder conocer cuál tipo de motor, si DC1, DC2, PAP1, PAP2 se ha establecido un protocolo de comunicación entre el

microcontrolador y el computador, y por medio de dos bits se identifica el motor a usar. En el Anexo F, se incluye el código de la rutina de carga de los datos y almacenamiento en la memoria, además de una rutina para control de motor DC y una de motor paso a paso. En realidad, son dos rutinas para cada tipo de motor, pero solo se incluye una de cada tipo puesto que son muy similares.

Igualmente, se colaboró en la realización de rutinas para probar el correcto funcionamiento de los módulos de salidas PWM por TRIAC para los motores DC y las salidas de los motores paso a paso. Para la prueba de motores DC se creó unas rutinas que activan las salidas de los Triac usando temporizadores para el cambio de estado, sentido de giro y detención. Para la prueba de los motores paso a paso se desarrolló una rutina que activa la secuencia de las salidas correspondientes a los pines del puerto C del microcontrolador, en condición de paso completo.

4.2.3 Rutinas para sistema de automatización de horno ahumador: para este proyecto se aportó asesoría en cuanto a la utilización del sistema multitarea creado, integrando las tareas de exploración de señales provenientes del teclado matricial y adquisición de señales de sensores de humedad, temperatura. Así mismo se colaboró en asesorar a los estudiantes para la realización de rutinas del teclado matricial, de exploración de las teclas y adquisición de estos datos para ser guardados en el microcontrolador, así como también rutinas del manejo y utilización de una pantalla LCD para visualización de caracteres ó mensajes. Se brindó tutoría en el desarrollo de rutinas para un control de humedad, tras obtener la señal del sensor existente en el prototipo. Así mismo, en rutinas para dos motores paso a paso a utilizar al abrir y cerrar unas ventilas que controlan la salida del humo del horno y la entrada de aire para disminuir la temperatura interna del mismo. Igualmente, se colaboró en la realización de una subrutina para el control de la temperatura del horno.

Para el desarrollo de estas rutinas se empezó por aclarar con los estudiantes cuales son las funciones del sistema de control a utilizar para el horno de ahumado, cuales son sus variables de entrada y cuales las de salida, posteriormente, se aconsejó la mejor forma de desarrollar un menú a base de un teclado matricial que guarde en el microprocesador la información y despliegue mensajes a la pantalla de cristal liquido, posteriormente, se asistió en el desarrollo como tal de las rutinas atendiendo las inquietudes presentadas por los estudiantes, depurando errores de los programas y orientando acerca de instrucciones propias del PIC utilizado.

Algunas de estas rutinas se detallan en el Anexo F.

6.2.4 Rutinas para sistemas de ambientes controlados para crianza de cerdos: en esta tarjeta se colaboró en la creación de unas rutinas de temporización, adquisición de señales digitales provenientes de sensores,

integración de las rutinas de adquisición y control al planeador, además, de unas rutinas de control de motores paso a paso y de atenuación de señales. Igual que para el caso anterior se prestó asesoría teniendo en cuenta las características y requerimientos del sistema de control a implementar. Se analizaron variables de entrada y salida y se aconsejó la mejor forma de control a usar para cada caso. En la parte del desarrollo de las rutinas se creó junto con los estudiantes un protocolo de comunicación a usar tanto para la programación en el microprocesador como en el programa de computador. Así mismo, se asistió permanentemente a los estudiantes en la resolución de dudas, inquietudes y problemas en general.

En general, se contribuyó en gran parte para terminar eficientemente las diferentes rutinas de cada uno de los proyectos y que sirven para el control de los prototipos desde el microcontrolador. Los resultados de estas rutinas se verificaron al realizar las pruebas necesarias y poner en funcionamiento los diferentes módulos de cada tarjeta de adquisición y/o control, que a la postre determinan la aplicabilidad del software y su desempeño a corto, mediano y largo plazo.

4.3 DESARROLLO DE RUTINAS EN LABVIEW ESPECÍFICAS

Con los estudiantes de Tecnología en Automatización se inició el trabajo en el diseño de la interfaz a través de LabVIEW, entre el usuario y las tarjetas de adquisición de datos.

Se explicaron algunos elementos de programación que contiene la herramienta, buscando que éstos sean usados por los estudiantes en el desarrollo de su trabajo de programación, como son la creación de instrumentos virtuales dentro de otros ya existentes, los elementos de comunicación serial, tanto para enviar como para recibir datos, la comunicación a través del puerto paralelo, los arreglos y clusters para manipulación de un grupo de elementos y su control posterior de una manera más sencilla, las estructuras de secuencia, las estructuras de ciclos, las estructuras de selección de casos o estados. Con todas estas herramientas brindadas a los estudiantes a medida que iban siendo necesarias para la programación, se buscó que ellos aprendan el manejo de la herramienta, el lenguaje de programación G, para que posteriormente sea aplicado en el desarrollo de éste proyecto de las tarjetas de adquisición, así como en sus proyectos finales y otros proyectos de aplicación.

Para la programación y construcción de la interfaz de las tarjetas, se asesoró en la constitución de cuatro etapas primordiales como son la configuración, el control, la adquisición y la comunicación. Se busca que haya una buena comunicación con los microcontroladores existentes en las tarjetas, además, llevando siempre un protocolo establecido para la comunicación donde se establecen todas las posibles situaciones y se prevén errores así como la forma de solucionarlos y

atender ante acontecimientos como éstos. Se colaboró en rutinas de selección, habilitación de variables que simulan entradas o salidas de las tarjetas, así como en la constitución del panel frontal de control de dichas tarjetas con su correspondiente programación, estableciendo secuencias para el envío de los datos hacia el microcontrolador.

El trabajo que había iniciado con los estudiantes de Tecnología en Automatización continuó con los estudiantes tecnólogos en Electrónica Industrial, con quienes se estableció la creación de una interfaz gráfica entre los proyectos en su parte de hardware con la tarjeta y software con el microcontrolador y los usuarios quienes son los que finalmente van a hacer uso de los prototipos.

Los tres proyectos de las tarjetas de adquisición de señales, tanto análogas, digitales como de control de motores, optaron por establecer un control desde el computador por medio de LabVIEW, luego de impartir tutoría sobre fundamentación y programación en este lenguaje gráfico. Igualmente, los estudiantes encargados del grupo de sistema de crianza de cerdos se interesaron por esta herramienta y con ellos se construyó una agradable interfaz desde la cual se puede controlar el prototipo. El último grupo encargado del sistema de ahumador del horno optó por establecer una interfaz con una pantalla LCD y un teclado matricial, utilizando para ello, la programación en el microcontrolador únicamente.

A continuación, en las figuras 9 y 10, se presenta el diagrama de bloques de unas rutinas realizadas con dos estudiantes del equipo de trabajo de la tarjeta de adquisición de datos digital, para la adquisición de señales y otras para la conversión de nibbles a bytes utilizadas en la interfaz creada para la tarjeta de adquisición de señales digitales.

Figura 9. Diagrama de bloques subVI nibble a byte

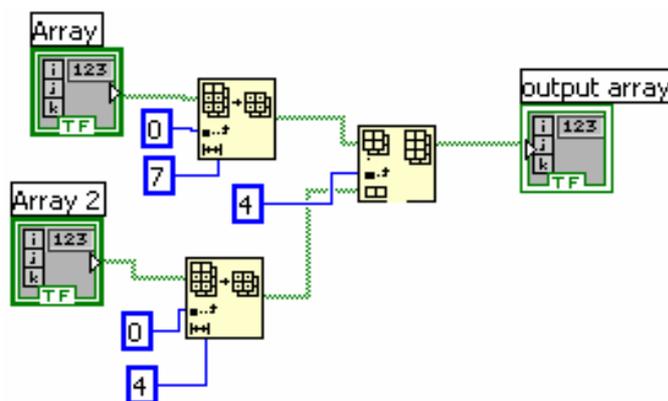
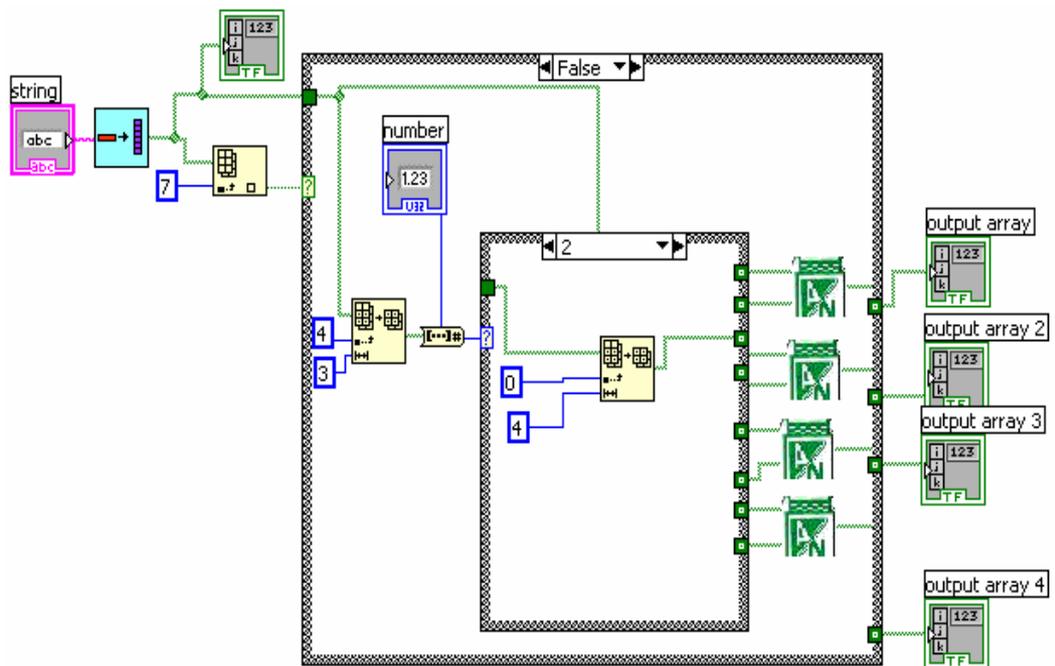


Figura 10. Diagrama de bloques VI recepción



A continuación, se presenta el panel frontal de la interfaz, realizada con algunos estudiantes para algunos proyectos, con una breve explicación. Como se ha descrito anteriormente, cada programa contiene unos módulos de configuración, control y adquisición. Para separarlos se utilizan unas tablas que en cada ventana establecen una de estas tres partes fundamentales de los proyectos.

En primera instancia se presentan las figuras 11, 12 y 13, correspondientes al panel frontal del proyecto Sistema de adquisición de datos analógicos didácticos, en sus tres módulos, desarrollado junto con dos estudiantes de este equipo de trabajo. Seguidamente se tiene el panel frontal de los tres módulos de la tarjeta correspondiente al sistema de control de motores por computador, igualmente se realizó junto con dos estudiantes de este grupo, representado en las figuras 14, 15 y 16. Por último, en las figuras 17, 18 y 19, se representa cada módulo del panel frontal del proyecto sistema para ambientes controlados para la crianza de cerdos, también desarrollado con dos estudiantes.

Figura 11. Panel frontal de configuración de tarjeta analógica

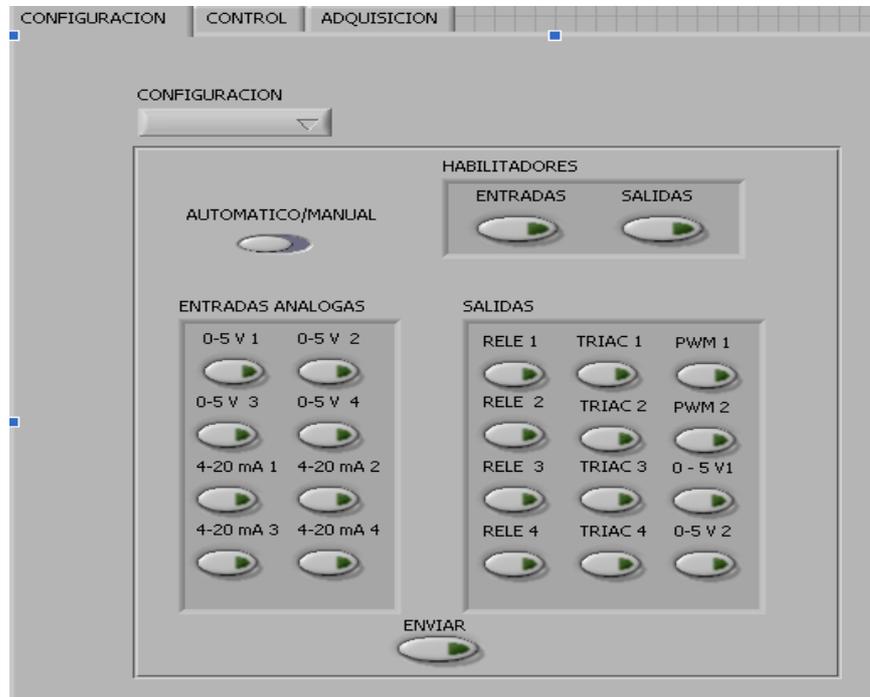


Figura 12. Panel frontal de control de tarjeta analógica

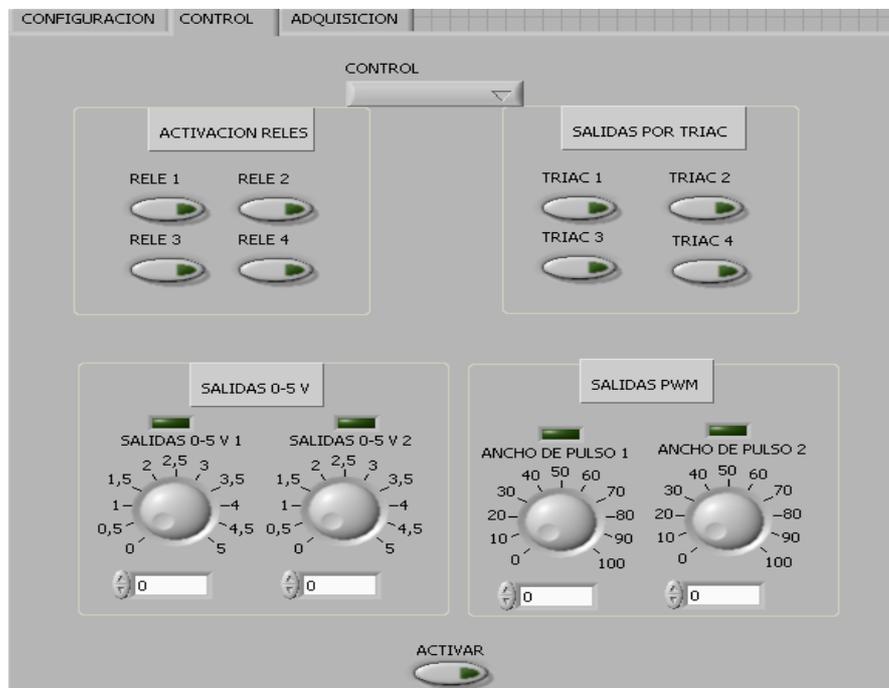


Figura 13. Panel frontal de adquisición de tarjeta analógica

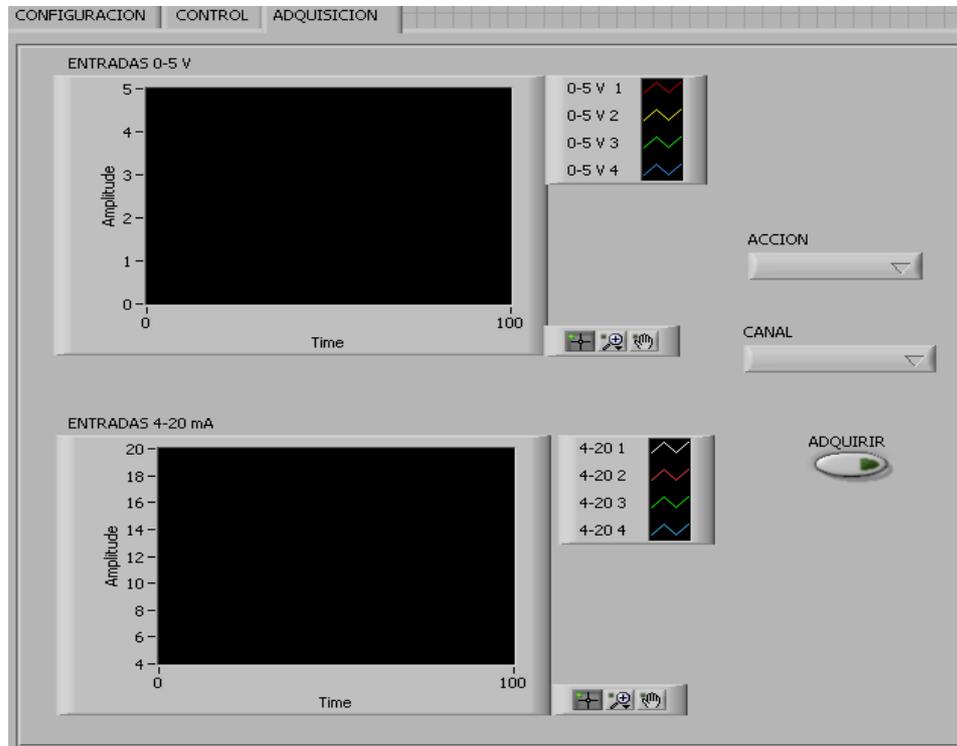


Figura 14. Panel frontal de configuración de tarjeta de motores

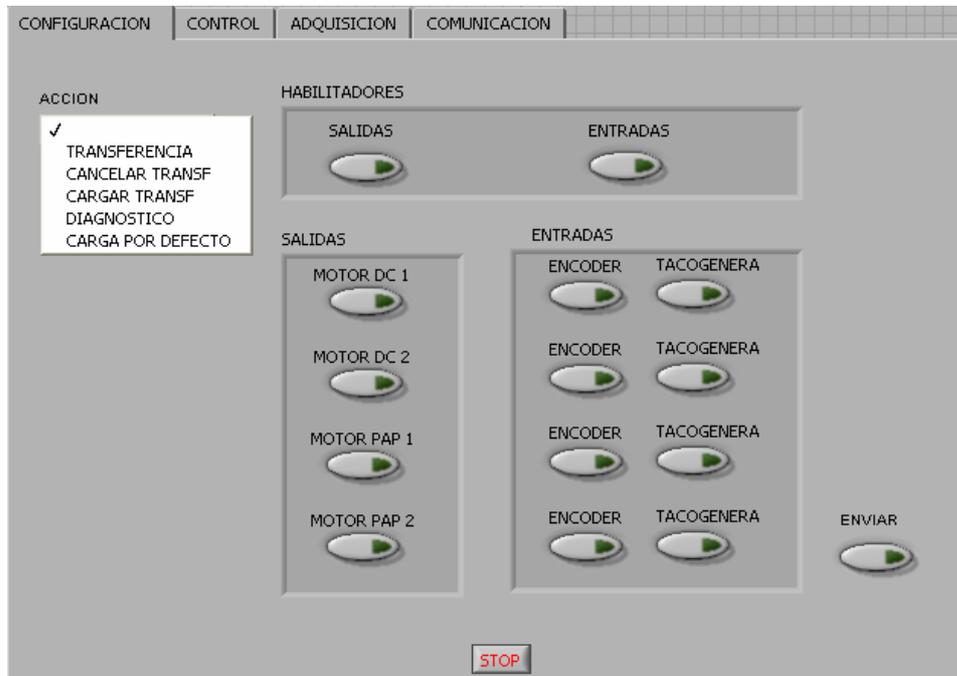


Figura 15. Panel frontal de control de tarjeta de motores



Figura 16. Panel frontal de adquisición de tarjeta de motores



Figura 17. Panel frontal de configuración de sistema crianza de cerdos.

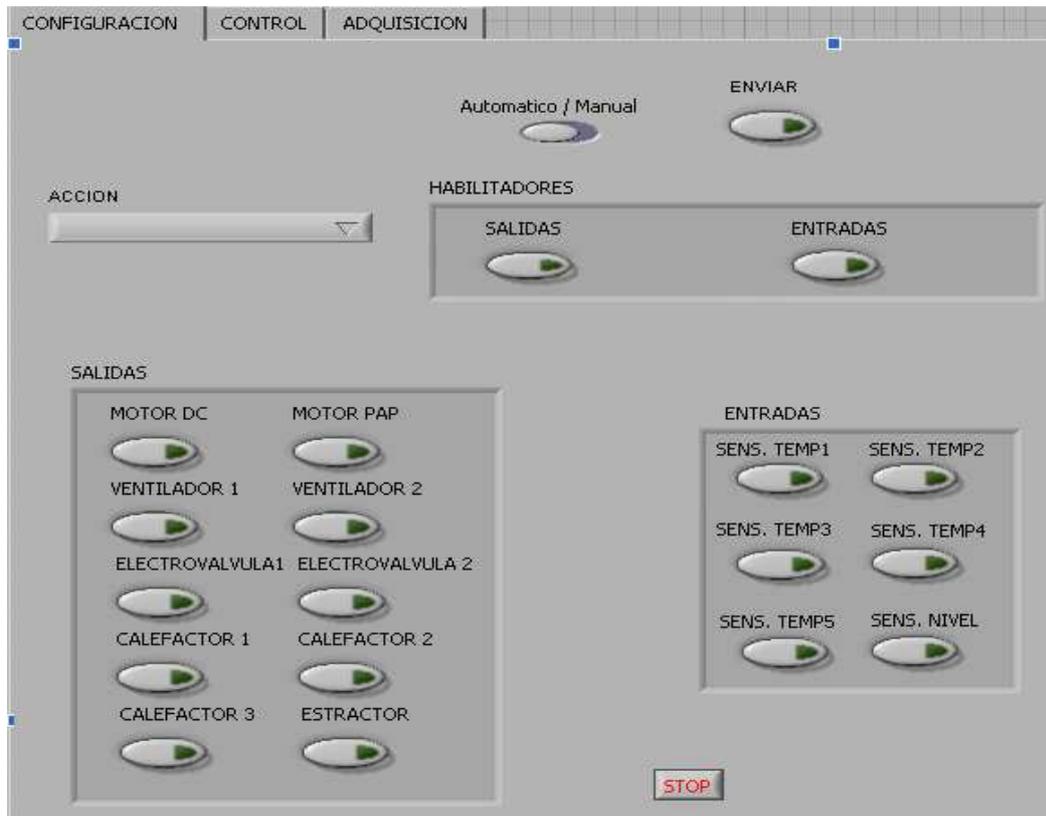


Figura 18. Panel frontal de control de sistema de criadero de cerdos.

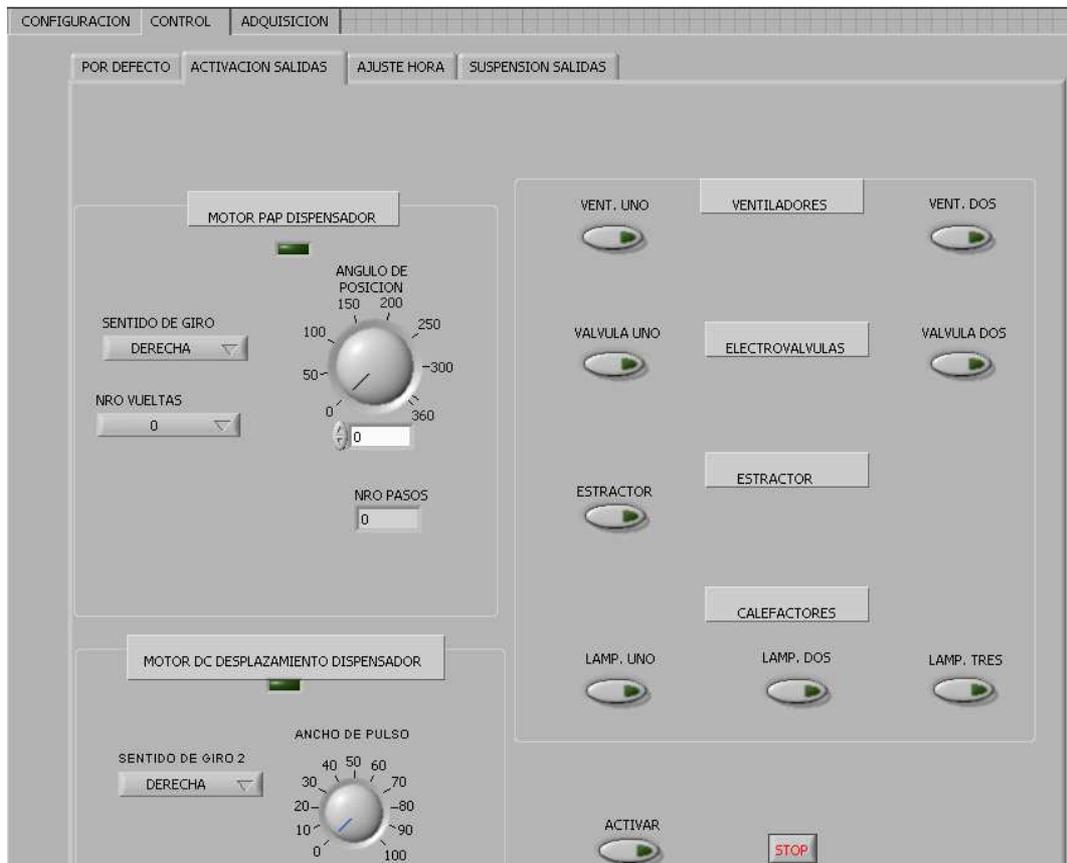
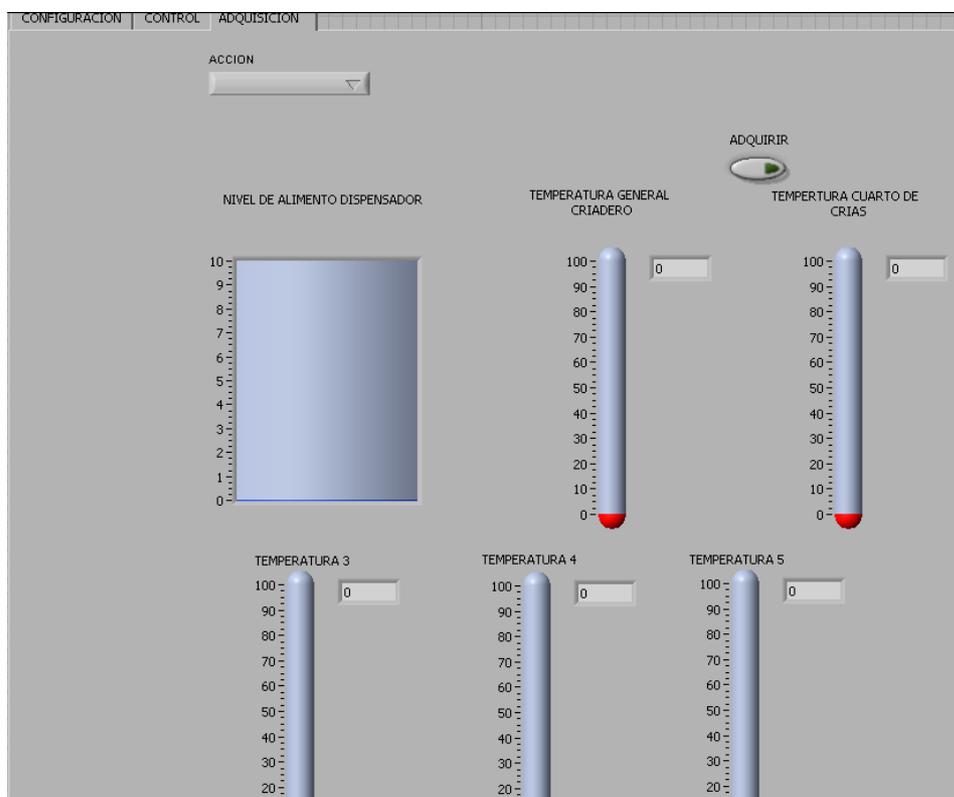


Figura 19. Panel frontal de adquisición de sistema de criadero de cerdos.



Como se puede ver en estas figuras la parte de configuración está relacionada con la habilitación o no de entradas y salidas que posee la respectiva tarjeta o sistema de control. Así mismo, se busca que el usuario pueda elegir entre configuración manual para que él mismo manipule los sensores y actuadores o configuración automática que hace que se establezca una configuración por defecto ya definida por cada creador del prototipo.

En el módulo de control de cada una de las tarjetas se busca tener absoluto control de los actuadores, pudiendo manipular salidas digitales para control on-off, como lo es en el caso de activación de electroválvulas, ventiladores, calefactores, puertas, y salidas análogas para el control de motores o salidas de variación de voltaje, corriente y/o fase.

En la parte de la adquisición se tiene la visualización de las entradas provenientes de los sensores que también van a servir en la parte de control para la activación de los actuadores. La visualización de procesos de nivel, temperatura o señales análogas cualesquiera se pueden observar en indicadores gráficos, termómetros, tolas, entre otros indicadores. Existen también indicadores booleanos para poder

observar las señales digitales en un momento determinado, de sensores como finales de carrera, sensores ópticos.

4.4 TUTORIA A NIVEL DE HARDWARE

En cuanto a la parte de hardware se colaboró a los grupos con consejos de mejoras en busca de un mayor rendimiento de los dispositivos a largo plazo, además de la tutoría prestada en cuanto a diferentes circuitos que se montaron en protoboard y/o circuito impreso para prueba de módulos específicos, así como también para las etapas de las tarjetas como tales.

Se aportó en el diseño de algunos circuitos específicos como la conversión de una señal de corriente de 4-20 mA a una señal de voltaje, en la cual se introdujeron mejoras al diseño que los estudiantes desarrollaron; la construcción de un puente H para el control de motores DC, la conexión de elementos como teclado matricial, pantalla de cristal líquido; el montaje de circuitos de comparación, conversión de señales, protección de puertos y operaciones lógicas en general. Esta asesoría, básicamente, consistió en consejos para el diseño y simulación de estos y otros circuitos pequeños con programas simuladores de electrónica existentes en el Centro como lo es la herramienta Multisim; así mismo, se colaboró en la realización de pruebas con los componentes en protoboards y/o impresos una vez que se tuvieron listas las tarjetas en su parte de hardware. La idea es respetar los diseños de los estudiantes asesorándolos para incluir mejoras que puedan hacer más eficiente la respuesta del circuito ó que sirvan para reducirlo en cuanto a componentes, siempre buscando buen funcionamiento a corto, mediano y largo plazo.

4.5 ACTIVIDADES DE MANTENIMIENTO DE EQUIPOS

En el centro multisectorial ILOPE SENA se llevaron a cabo igualmente actividades de mantenimiento de equipos informáticos específicos y electrónicos en general. Existen varios laboratorios dotados de computadores que necesitan un mantenimiento frecuente para procurar un buen rendimiento y de esta forma poder aprovechar al máximo los recursos. Se realizó formateo de los 6 equipos del laboratorio de Mecatrónica, se instaló el sistema operativo a cada uno de ellos. Se particionó el disco en cada PC, y se actualizaron sus controladores. Se realizó la instalación de programas de mecatrónica como son el WinSPS, WinDP, R0C5, correspondientes a los PLC's de Bosch existentes en el SENA. Posteriormente, se trabajó en la conexión en red de los 6 equipos del aula de Mecatrónica; se hicieron algunas pruebas con el controlador CL200 correspondiente al PLC de Bosch, cargando pequeños programas y verificando el buen funcionamiento de algunas estaciones de la planta de mecatrónica existente. De la misma manera, se instaló y puso en operación una herramienta de Microsoft como lo es Virtual PC para obtener mayor rendimiento de los equipos y conseguir tener varios sistemas

operativos dentro de un mismo equipo o host. Esto se adelantó con el fin de preservar programas con caducidad de tiempo en su aplicación.

En el aula de Gestión y Desarrollo de Prototipos (Aula Abierta) se trabajó en la instalación de software en los diez equipos existentes. El software que se instaló fue LabVIEW versión 7.1, así como también se instalaron actualizaciones de Java, del antivirus McAfee. Se realizó mantenimiento de estos equipos en varias ocasiones, depurando programas erróneos, liberando espacio de disco, refrescando espacio de disco duro y procurando una respuesta rápida de conexión a la red.

En el aula del laboratorio de Electrónica se instaló LabVIEW 7.1 en todos sus equipos (10 en total). En varias ocasiones se llevó a cabo jornadas de mantenimiento y al igual que para el caso del aula de gestión de prototipos, se contribuyó a mantener en buen estado y siempre funcionando estos elementos de gran utilidad para los estudiantes.

En el laboratorio de automatización se realizaron actividades de formateo e instalación de sistema operativo de cuatro equipos.

Otro punto importante que se llevó a cabo en el SENA Regional Nariño junto con Mario Ortega, pasante de la Universidad Nacional de Manizales, tiene que ver con la configuración de la red para el sector agroindustrial del centro. Si bien este trabajo aún no se termina, se adelantó gran parte con el reconocimiento, instalación, fundamentación y posterior aplicación del Windows Server 2003, sistema operativo con el cual se pretende tener un control no solo del acceso a Internet por parte de la comunidad SENA, sino también permitir el control de los discos duros como tales, estableciendo varios servidores en cada aula, dentro de los cuales se permita el acceso a determinados programas, a Internet, a privilegios especiales que en un momento dado se les pueden conceder a un usuario de la red.

El trabajo llevó gran parte del tiempo, desde la recopilación de información del sistema a utilizar como de la configuración de otras redes. Posteriormente, el aprendizaje de la herramienta Windows Server 2003, así como de los programas Free Proxy, Ethereal Real time, se constituyeron en punto de partida para establecer la configuración de la red. Ya se establecieron pruebas con un grupo pequeño de equipos y se dieron excelentes resultados.

5. RESULTADOS

Los resultados que se esperaban obtener se cumplieron con satisfacción plena no solo por parte del estudiante pasante, sino también por parte de los estudiantes de tecnología a quienes se brindó asesoría y por parte del Ingeniero Orlando Orbes, jefe inmediato del Sena, quien en muchas ocasiones así lo ha manifestado.

El desarrollo efectivo de los proyectos que ambiciosamente se habían planteado conllevan a demostrar el éxito de la pasantía, pues los prototipos han sido implantados y se encuentran instalados en los laboratorios de electrónica del SENA Regional Nariño, con la posibilidad de expandir la producción de los prototipos para montar el aula de gestión y desarrollo de prototipos y dotarla de herramientas útiles, versátiles y de tipo didáctico, a las cuales se les puede sacar mucho provecho. En el anexo A se incluye la figura de uno de estos prototipos implantados en el laboratorio y entregado ante los directivos de la entidad por parte de los estudiantes.

No solo el elemento final es importante, la certeza que los estudiantes han aprendido de las asesorías y saben programar en los diferentes tipos de lenguajes empleados en el desarrollo de los proyectos, da la satisfacción de haber cumplido unos objetivos que se habían propuesto al buscar una formación buena de los estudiantes de tecnología en Electrónica Industrial y Automatización.

CONCLUSIONES

Proyectos como los sistemas de adquisición de datos y control, así como de automatización de procesos agroindustriales fortalecen el crecimiento de la tecnología en la región del sur occidente de éste país. Sobre todo en Nariño, los avances tecnológicos son escasos y prototipos como los que se adelantan en el sector de industria del Centro Multisectorial Lope, solventan una gran necesidad que se tiene de tecnología, de productos de gran aplicabilidad y utilidad para tecnificar el entorno que habitamos.

Como se ha descrito en este proyecto, dos de los prototipos adelantados están relacionados con la automatización de procesos industriales propios del SENA Regional Nariño, pues es el sector agroindustrial el medio a explotar en cuanto a desarrollo de proyectos de tipo tecnológico que optimicen los procedimientos y procuren una producción de mayor calidad y cantidad. Proyectos como el sistema de ambientes controlados para crianza de cerdos, así como el sistema de automatización de horno de ahumado, solventan necesidades y producen el desarrollo de este sector que tanto se espera.

Cabe resaltar que aunque el trabajo de asesoría que se brindó a los diferentes grupos de estudiantes, fue arduo y de gran responsabilidad, queda la satisfacción de haber podido compartir con grupos de personas interesadas en el área de la electrónica y la automatización, aprender de ellas y tener la certeza que ellas aprendieron de la tutoría prestada, con la consecución final de los prototipos. El nivel de la pasantía por tanto es alto, y la disponibilidad de tiempo completo que se manejó muchas veces se hacía insuficiente para poder tener un avance significativo en el desarrollo de los proyectos y demás actividades del trabajo de grado.

Otra conclusión interesante se destaca al poder dotar al aula de gestión y desarrollo de prototipos del SENA, de herramientas de carácter didáctico, como son las tarjetas de adquisición de datos, que no solamente van a ser útiles para el control y automatización de procesos industriales, sino que también servirán para promover la formación académica, investigativa de nuevos estudiantes que ingresen al Centro y hagan parte de la familia de la Electrónica y la Automatización.

RECOMENDACIONES

Promover la masificación de la producción y ensamble de los cinco prototipos diseñados, así como incentivar en los nuevos estudiantes, el diseño y desarrollo de nuevas herramientas de este tipo, que sirvan como bancos de prueba para laboratorio o como sistema de automatización de un proceso agroindustrial del centro.

BIBLIOGRAFIA

DUQUE, Edison. Curso Básico de Microcontroladores PIC.
Pereira: Compañía Editorial Tecnológica CEKIT S.A, 2002. 185 p.

LAZARO, Antoni y DEL RIO, Joaquín. LabVIEW 7.1 Programación Gráfica para el Control de Instrumentación.
Madrid : Paraninfo S.A, 2005. 326 p.

OGATA, Katsuhiko. Ingeniería de Control Moderna. 2 ed.
Mexico : Prentice-Hall Hispanoamericana, S.A, 1993. 1021 p.

ANEXO A
PROTOTIPO MODELO: SISTEMA DE ADQUISICIÓN DE DATOS
ANALOGICOS MICROPROCESADOS



ANEXO B FUNDAMENTOS DE LABVIEW

Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y Linux y va por la versión 8.2.

Es una herramienta gráfica de test, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje G. El LabView es un lenguaje de programación de alto nivel, de tipo gráfico, y enfocado al uso en instrumentación. Pero como lenguaje de programación, debido a que cuenta con todas las estructuras, puede ser usado para elaborar cualquier algoritmo que se desee, en cualquier aplicación, como en análisis, telemática, juegos, manejo de textos, etc.

Es un sistema abierto, en cuanto a que cualquier fabricante de tarjetas de adquisición de datos o instrumentos en general puede proporcionar el driver de su producto en forma de VI dentro del entorno de LabVIEW. También es posible programar módulos para LabVIEW en lenguajes como C y C++, estos módulos son conocidos como Sub-VIs y no se difieren a los VI creados con LabVIEW salvo por el interfaz del lenguaje en el que han sido programados. Además estos Sub-VIs son muy útiles por ejemplo en el campo de cálculos numéricos complejos que no se encuentran incluidos en las librerías de LabVIEW.

Los programas hechos con LabVIEW se llaman VI (*Virtual Instrument*), lo que da una idea de uno de sus principales usos: el control de instrumentos. El lema de LabVIEW es: "*La potencia está en el Software*". Esto no significa que la empresa haga únicamente software, sino que busca combinar este software con todo tipo de hardware, tanto propio -tarjetas de adquisición de datos, PAC, Visión, y otro Hardware- como de terceras empresas.

LabVIEW permite diseñar interfaces de usuario mediante una consola interactivo basado en software. Se puede diseñar especificando su sistema funcional, su diagrama de bloques o una notación de diseño de ingeniería. LabVIEW es a la vez compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab. Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes).

También ofrece la ventaja de "debugging" en cualquier punto de la aplicación. Permite la posibilidad de poner "break points", ejecución paso a paso, ejecución hasta un punto determinado y se puede observar como los datos van tomando

valores a medida que se va ejecutando la aplicación. Además también lleva incorporado generadores de señales para poder hacer un simulador.

El software LabVIEW de National Instruments brinda una flexibilidad y facilidad de uso únicas en aplicaciones demandantes de medidas, automatización y control industrial. Se puede desplegar programas de LabVIEW a controladores de automatización programables (PACs) para crear sistemas distribuidos de control y medidas confiables así como para conectarse a sus controladores de lógica programable (PLCs) existentes y sistemas de empresa.

Es usado principalmente por ingenieros y científicos para tareas como:

- Control de instrumentos
- Adquisición de datos
- Automatización industrial o PAC (Controlador de Automatización Programable)
- Diseño de control: prototipaje rápido y hardware-en-el-bucle (HIL)

Su principal característica es la facilidad de uso, personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él. Con LabVIEW pueden crearse programas de miles de VIs (páginas de código) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, etc. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación.

Presenta facilidades para el manejo de:

- Interfaces de comunicaciones:
 - Puerto serie
 - Puerto paralelo
 - GPIB
 - PXE
 - VXI
 - TCP/IP, UDP, DataSocket
 - Irda
 - Bluetooth
 - USB
 - OPC...
- Capacidad de interactuar con otras aplicaciones:
 - dll

- ActiveX
 - Matlab
 - Simulink...
-
- Herramientas para el procesado digital de señales.
 - Visualización y manejo de gráficas con datos dinámicos.
 - Adquisición y tratamiento de imágenes.
 - Control de movimiento.
 - Tiempo Real estrictamente hablando.
 - Programación de FPGAs.
 - Sincronización.

Existen tres conceptos básicos en LabVIEW:

B.1 EL PANEL FRONTAL

Es el interfaz que el usuario está viendo y puede ser totalmente parecido al instrumento del cual se están recogiendo los datos, de esta forma el usuario sabe de manera precisa cual es el estado actual de dicho instrumento y los valores de las señales que se están midiendo. Contiene controles e indicadores y existe una gran variedad de ellos, pero además, se pueden diseñar controles e indicadores personalizados, lo cual permite ampliar la gama de estas herramientas.

Un control puede tomar muchas formas, y muchas de estas formas el dibujo real usado en instrumentos reales. Otros son estrictamente conceptos digitales o analógicos. Pero todos los controles tienen una forma visual que indican al usuario cual es el estado de dicho control en el instrumento real.

Es muy importante en un sistema SCADA que el usuario no tenga que interpretar nada, sino que todo sea claro y conciso, las interpretaciones pueden dar lugar a falsas actuaciones y, por consiguiente, podrían existir lamentables errores. Además, dos usuarios podrían interpretar de manera diferente cualquier evento.

B.2 DIAGRAMA DE BLOQUES

Muestra el programa en código gráfico G. Se usan en este diagrama estructuras de programación y flujo de datos entre las diferentes entradas y salidas, a través de líneas. En este las subrutinas son mostradas como iconos de cajas negras, con unas entradas y unas salidas determinadas, donde en el interior se cumple una función específica. El flujo se aprecia, como se dibujaría en un bosquejo de sistemas, cuando se habla de teoría de sistemas, donde cada subsistema se representa como un cuadro con entradas y salidas. Todos los indicadores y controles ubicados en el panel frontal están respaldados por un terminal de conexión en el diagrama de bloques tal como si se tuviera un tablero de control de

una máquina o un avión, donde por el frente se ven los indicadores y por el lado posterior se aprecian todos los cables y terminales de conexión.

B.3 EL ICONO DE CONEXIÓN

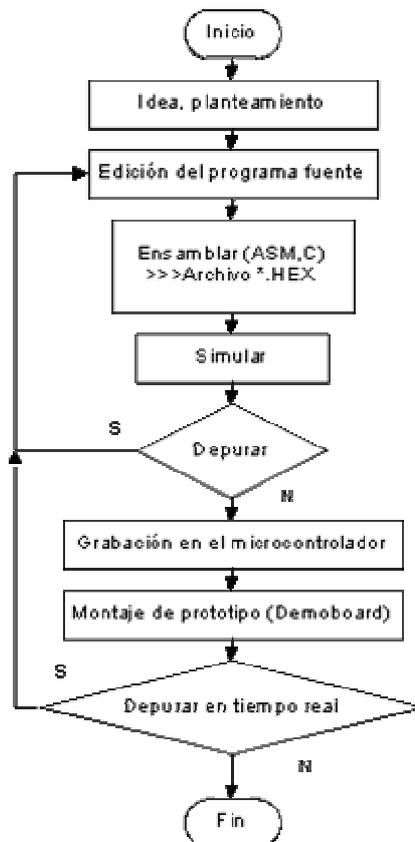
Se usa para utilizar el programa creado como subrutina en otro programa, donde el icono será la caja negra, y las entradas son las conexiones a los controles del programa subrutina, y las salidas son las conexiones a los indicadores del mismo subprograma. La idea es crear un sistema de programación modular, donde cada rutina creada llame otras rutinas, y estas a su vez otras de menor nivel, en una cadena jerárquica con cualquier límite deseado. Así cuando se use un módulo, no se requiere saber como funciona interiormente, simplemente solo basta conocer sus entradas y salidas para ser así usado.

ANEXO C FUNDAMENTOS DE MICROCONTROLADORES

Hasta antes de la aparición de los microprocesadores (1971), la mayor parte de las aplicaciones digitales en electrónica se basaban en la llamada lógica cableada, es decir, si existía un problema éste era analizado y se sintetizaba una función en base a la lógica de boole que era la solución al problema planteado.

Con la aparición de los microprocesadores, se varió el esquema de diseño de tal forma que un problema era descompuesto en una serie de tareas más simples; el microprocesador ejecutaba una serie de pasos o instrucciones para llevar a efecto cada una de las tareas, en ocasiones no era necesario volver a armar un circuito para solucionar otro problema sino que se cambiaba las instrucciones (programa) para obtener otra aplicación. En la figura 20 se muestra un diagrama de flujo del funcionamiento del microprocesador.

Figura 20. Funcionamiento de los microprocesadores



El microprocesador es como el cerebro que ejecuta operaciones de índole aritméticas y lógicas, por tanto no manejaba líneas externas (periféricos) más aún tampoco tenía un lugar donde almacenar el programa y los datos que necesitaba el programa para encontrar la respuesta al problema. El microprocesador buscaba una instrucción y la ejecutaba. Al conjunto de circuitos (hardware) que daban el soporte necesario al microprocesador se le llamó sistema mínimo.

Con el pasar de los años el sistema mínimo se convirtió en un estándar, por otro lado la escala de integración mejoró y permitió (1976) sintetizar en un solo chip un sistema mínimo, al cual se le llamó SISTEMA A que no era otra cosa que el primer microcontrolador. En consecuencia definimos así a un microcontrolador; como un procesador con su sistema mínimo en un chip (incluye memoria para programa y datos, periféricos de entrada / salida, conversores de AD y DA, módulos especializados en la transmisión y recepción de datos). Desde luego que hay diferencias sustanciales como la arquitectura cerrada de un microcontrolador, en cambio en un microprocesador es abierta, podemos sumar nuevos dispositivos en hardware en función a las necesidades que la aplicación demande.

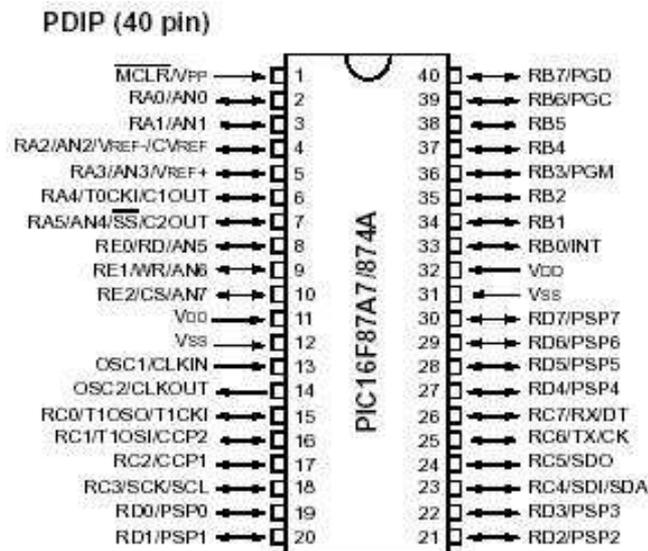
Otra diferencia entre los microcontroladores y los microprocesadores es que los primeros cuentan con un set de instrucciones reducido en cambio la mayoría de los microprocesadores tienen mayor cantidad de instrucciones. Por otro lado la mayoría de los microcontroladores posee una arquitectura Harvard frente a una arquitectura Von Neuman de los microprocesadores.

Los microcontroladores se especializan en aplicaciones industriales para resolver problemas planteados específicos por ejemplo: se los encuentra en los teclados o mouse de las computadoras, son el cerebro de electrodomésticos, también se los halla en la industria automotriz, en el procesamiento de imagen y video. Cabe señalar que el aumento progresivo de la escala de integración y las técnicas de fabricación hacen que cada vez aparezcan microcontroladores más poderosos y rápidos.

C.1 Microcontroladores microchip 16F87X

Casi todos los fabricantes de microprocesadores lo son también de microcontroladores; en el mercado existen una serie de marcas bastante conocidas y reconocidas como es el caso de Microchip, Motorola, Hitachi, etc. Se ha seleccionado a Microchip y en particular la serie 16F87X, debido a la abundante información y herramientas de diseño existente en el mercado (tanto local como internacional). También salta a la vista el hecho que es sencillo en el manejo y contiene un buen promedio elevado en los parámetros (velocidad, consumo, tamaño, alimentación). En la figura 21 se puede observar el microcontrolador y la descripción de sus pines.

Figura 21. Microprocesador pic 16f87x



Las principales características con que cuenta el 16F87X son:

- Procesador de arquitectura RISC avanzada
- Juego de 35 instrucciones con 14 bits de longitud. Todas ellas se ejecutan en un ciclo de instrucción menos las de salto que tardan 2.
- Frecuencia de 20 Mhz
- Hasta 8K palabras de 14 bits para la memoria de código, tipo flash.
- Hasta 368 bytes de memoria de datos RAM
- Hasta 256 bytes de memoria de datos EEPROM
- Pila con 8 niveles
- Modos de direccionamiento directo, indirecto y relativo
- Perro guardián (WDT)
- Código de protección programable
- Programación serie en circuito con 2 patitas
- Voltaje de alimentación comprendido entre 2 y 5.5 voltios
- Bajo consumo (menos de 2 mA a 5 V y 5 Mhz)

C.2 Arquitectura 16F87X

Los PIC16F87X de Microchip pertenecen al tipo de procesador RICS que es un procesador de instrucciones reducidas. Se caracteriza por que el número de instrucciones es pequeño y además casi todas se realizan en la misma cantidad de tiempo, por otro lado posee unidades que trabajan en paralelo conectadas por pipes o tuberías. Este tipo de procesador emplea una arquitectura Harvard lo que significa que trabaja las zonas de memoria de programa y datos en forma

separada. En la arquitectura Harvard existen dos bloques de memoria separados. Un bloque para instrucciones y otro para datos. Hay dos buses independientes de direcciones y el bus de instrucciones solo tiene una dirección, a diferencia del bus de datos que es de naturaleza bidireccional. La arquitectura Harvard mejora el ancho de banda porque el bus de datos es de 14 bits frente a los de 8 de un bus tradicional Von Neumann, por tanto, en una sola lectura puede llevar mayor cantidad de datos.

C.3 Arquitectura interna 16F87x

Se ha señalado que el microcontrolador posee varios elementos en forma interna: el procesador, memoria de programa, memoria de datos, periféricos, contadores.

En la figura 22 se puede identificar la memoria del Programa en la parte superior izquierda con 8K posiciones por 14 bits; también está presente la memoria de datos (RAM) de 368 posiciones por 8 bits, la memoria EEPROM 256 posiciones x 8 bits. El procesador propiamente dicho está formado por la ALU (unidad aritmética lógica), el registro de trabajo W. Se tienen los periféricos I/O Port A, B, C, D, E; el TMR0 (temporizador contador de eventos), TMR1 y TMR2 entre otros módulos.

También se cuenta con un registro de instrucción que se carga cada vez que la ALU solicita una nueva instrucción a procesar. En la parte intermedia se encuentran algunos bloques como son el registro Status, que es el registro de estado encargado de anotar el estado actual del sistema, cada vez que se ejecuta una instrucción se llevan a cabo cambios dentro del microcontrolador como desborde, acarreo, etc. Cada uno de esos eventos está asociado a un bit de este registro. Existe un registro de vital importancia que se llama el Program Counter o contador de programa, el cual indica la dirección de la instrucción a ejecutar. El registro en cuestión no es necesariamente secuencial, esto es, no se incrementa necesariamente de uno en uno ya que puede darse el caso en el que salte dependiendo si hay una instrucción de bifurcación de por medio o puede haber alguna instrucción de llamada a función y/o procedimiento. También se observa el bloque de la pila; la función de la pila es ser un buffer temporal en el que se guarda el contador de programa cada vez que se suscita una llamada a un procedimiento y/o función (incluyendo interrupciones). Por tanto el nivel de anidamiento es de hasta 8 llamadas. También está presente el registro FSR, que es el registro que cumple una función similar a la del contador de programa direccionando en este caso la RAM. El FSR es un puntero a una dirección de la RAM. La aparición de multiplexores se debe a que los datos pueden tener diferentes fuentes.

C.4 Organización de la memoria

La memoria se divide en memoria de datos y programa. La de datos a su vez se divide en:

- SFR (Special Function Register) Registros de propósito especial, son registros que ayudan a configurar el hardware interno, así como sirven para escribir o leer valores de los diferentes componente que constituyen el microcontrolador. Por ejemplo el registro "trisa" que nos permite configurar el modo de trabajo de las líneas del puerto A
- GFR (General Function register) Registros de propósito general, son posiciones de memoria que podemos usar para almacenar valores que emplean nuestros programa.

Debe señalarse también que la memoria de datos se divide en cuatro bancos (esto para el caso específico del 16F87X). Las posiciones bajas siempre están reservadas para los SFR en tanto que las altas para los GFR. También se tiene una memoria EEPROM, con 256 posiciones, para acceder a la memoria; no se puede leer o escribir directamente, es decir, colocar la dirección y obtener o dejar el valor. Para trabajarla se debe apoyar en los registros adicionales, de tal forma que se use indirectamente.

El contador de programa tiene 13 bits con los cuales se pueden direccionar 8K posiciones. Cuando el microcontrolador es levantado, el contador de programa siempre apunta a una dirección conocida como el VECTOR DE RESET, cuya dirección es la posición de memoria 0000h. También existe otro vector llamado VECTOR DE INTERRUPCIONES que ocupa la posición 0004h. Cuando se lleva a cabo una petición de interrupción el contador de programa va automáticamente a esa posición en busca de la instrucción que atiende la petición de interrupción. Como se ha mencionado la pila trabaja con el contador de programa; cada vez que hay una instrucción de llamada a procedimiento o función (call) el contador de programa se almacena allí y va en busca de la rutina; cuando acaba la rutina (con la ejecución de una instrucción return, retfie o retlw) se restituye el valor del contador de programa. La capacidad de la pila es de 8 posiciones; en caso de tener un desborde (ej 9 llamadas anidadas) la pila se dice que se desborda y vuelve a 0. Por tanto, se ha de pensar que la pila también cuenta con un contador que indica cual es la siguiente dirección vacía.

C.5 Herramienta MPLAB

Para programar es necesario contar con herramientas en hardware y software. En el mercado existen muchas herramientas que van de ensambladores a simuladores, emuladores o debugger físicos.

El MPLAB es un entorno de desarrollo, es decir, es un recipiente que incluye varias herramientas; contiene un editor de textos que no permite ingresar el programa expresado en códigos nemónico (o simplemente llamado ensamblador), normalmente éste se guarda en un archivo con extensión ASM. Una vez que se ha ingresado el programa dentro de un archivo, se crea un proyecto dentro del MPLAB. El proyecto puede contener a su vez varios archivos ASM que se relacionen a través de llamadas a rutinas o compartan y/o variables; adicionalmente el proyecto tiene un grupo de variables que se deben configurar, como es el tipo de microcontrolador que se va a usar ya que el MPLAB soporta todas las familias de microcontroladores Microchip (MPLAB es producido por Microchip). A continuación, se procede a llamar al programa ensamblador que lleva el MPLAB, capaz de transformar los código nemónicos (instrucciones) a los correspondientes valores binarios que a su vez se grabarán en el microcontrolador. El ensamblador (MPASMWIN) también genera otros archivos de salida que ayudan en el diseño de aplicaciones. El MPLAB es un software de libre distribución.

ANEXO D ADQUISICION DE DATOS

En la actualidad el vertiginoso desarrollo de la electrónica y la microelectrónica han motivado que todas las esferas de la vida humana se estén automatizando, por ejemplo: la industria, el hogar, los comercios, la agricultura, la ganadería, el transporte, las comunicaciones, etc. En todo ese proceso de automatización el microprocesador y el microcontrolador juegan un papel de suma importancia. Ellos han permitido el desarrollo de sistemas inteligentes que resuelven los más diversos problemas, son los llamados Sistemas de Adquisición de Datos.

El objetivo básico de los Sistemas de Adquisición de Datos es la integración de los diferentes recursos que lo componen: Transductores de diferentes tipos y naturaleza, multiplexores, amplificadores, conversores A/D y D/A, además el uso del microcontrolador 8051 como CPU del SAD diseñado, utilizando de este microcontrolador todas sus prestaciones: interrupciones, temporizadores, comunicación serie, así como hacer uso de memorias y puertos externos y creando con todo ello un sistema que se encargue de una aplicación específica como es chequear una variables (PH, humedad relativa, temperatura, iluminación, concentración, etc) para una posterior utilización de la misma ya sea con fines docentes, científicos, de almacenamiento o control y utilización de la misma.

Un Sistema de Adquisición de Datos no es más que un equipo electrónico cuya función es el control o simplemente el registro de una o varias variables de un proceso cualquiera; de forma general puede estar compuesto por los siguientes elementos.

- Sensores.
- Amplificadores operacionales.
- Amplificadores de instrumentación.
- Aisladores.
- Multiplexores analógicos.
- Multiplexores digitales.
- Circuitos Sample and Hold.
- Conversores A-D.
- Conversores D-A.
- Microprocesadores.
- Contadores.
- Filtros.
- Comparadores.
- Fuentes de potencia.

El S.A.D debe tener una estructura y organización muy equilibrada que le permita su buen funcionamiento, de ello depende de que el mismo rinda al máximo y sin ningún defecto.

Los sensores tienen un rol vital en todo SAD, ellos tienen la función de convertir la variable física que se desea registrar en una magnitud eléctrica (voltaje, corriente, resistencia, capacidad, Inductancia, etc.). Entre las magnitudes físicas más importantes a registrar se tienen: temperatura, humedad, presión, concentración, iluminación, flujo, posición, nivel, peso, etc. Diversas pueden ser las variables ambientales, industriales, biológicas, químicas, etc. que en un momento determinado se necesiten controlar; esto provoca que sean también numerosos los tipos de sensores así como su principio de funcionamiento, lo cual determina generalmente el costo de sensor que será necesario utilizar.

La etapa acondicionadora esta formada básicamente por amplificadores operacionales, comparadores de nivel y amplificadores de instrumentación.

Los amplificadores operacionales en sus configuraciones básicas (inversora, no inversora, amplificadora, conversor de corriente a voltaje, etc.), son usados para garantizar que al conversor A/D le sea suministrado el rango máximo de voltaje y así el mismo pueda dar el mayor número de combinaciones posibles.

El amplificador de instrumentación puede alternadamente sustituir al amplificador operacional, siempre que la aplicación lo exija, pues los mismos tienen prestaciones superiores a los amplificadores operacionales normales, lo cual hace que sean más costosos. Entre las características de los amplificadores de instrumentación están, una impedancia de entrada infinita y una ganancia ajustable en ocasiones mediante una red resistiva de precisión externa o mediante resistores internos de precisión por interruptores o por software.

Los aisladores por su parte son dispositivos de mucha importancia principalmente en sistemas médicos donde se requiere aislar completamente al paciente del equipo de medición con el fin de evitar que en caso de desperfectos del equipo los pacientes estén expuestos altos niveles de voltaje o corriente, también en equipos o instrumentos que manejen altas tensiones es necesario garantizar el aislamiento entre los instrumentos de medición y las fuentes de alta tensión. Entre los dispositivos más comunes son los opto-acopladores.

Los multiplexores ya sean analógicos o digitales son dispositivos que permiten multiplexar varias entradas en una única salida. Ellos permiten que para registrar varias señales diferentes se use un único conversor A/D y con ello disminuir de forma considerada el costo de un SAD. Generalmente los multiplexores se pueden dividir por el tipo de salida en simples y diferenciales o por el número de entradas en de 2, 4, 8 ó 16 entradas. El hecho de existir una gran variedad de multiplexores obliga a hacer una correcta selección según las exigencias del sistema, sobre la

base de disminuir los costos del mismo. Los multiplexores diferenciales de mayor costo que los de salida simple, son usados normalmente cuando son utilizadas para multiplexar señales de naturaleza diferentes por ejemplo: temperatura, presión, concentración, etc. Los amplificadores de salida simple se recomiendan cuando se multiplexan señales de naturaleza semejante: por ejemplo cuando registramos la temperatura en diferentes puntos. En esencia la diferencia entre los multiplexores de salida simple y diferencial está en que para los últimos, la señal de referencia (tierra) es también multiplexada lo cual no ocurre para los multiplexores de salida simple. En la medida que aumenta el número de entradas de un Multiplexor también aumenta su costo y el número de terminales de control que el mismo necesita, por lo cual, es también muy necesario utilizar en una aplicación un Multiplexor con el número de entradas que se requiera.

Los microprocesadores son los que se encargan del almacenamiento y procesamiento de los datos, son dispositivos que se encargan de todas las funciones de procesamiento de la señal. Estos son de gran importancia porque son como el corazón del sistema de adquisición de datos.

ANEXO E CONTROL Y AUTOMATIZACIÓN

Hoy en día se ha vuelto de vital importancia que las fábricas, industrias, y en sí cualquier persona o entidad que posea una planta donde se procese algún tipo de producto, pueda controlarla. Al hablar de una planta nos estamos refiriendo a un equipo, quizá simplemente a un juego de piezas de una máquina, funcionando de manera conjunta, que tienen como fin realizar un procedimiento determinado. O simplemente nos podemos referir a una planta como cualquier objeto físico que deba controlarse.

Este tipo de plantas se encuentran en cualquier lugar por donde se pase y los sistemas de control utilizados para justamente controlar, automatizar este tipo de plantas por lo general son de carácter mecánico, hidráulico, neumático y electrónico, y se han vuelto parte integral e importante de los procesos industriales y de manufactura modernos. Esto se hace con el fin de optimizar procesos y hacer que se cumpla con el objetivo final, alcanzando eficiencia y calidad.

Se ha venido hablando de procesos alrededor de este tema y se ha definido ésta palabra según el diccionario Merriam-Webster como “una operación o desarrollo natural, caracterizado por una serie de cambios graduales, progresivamente continuos, que se suceden uno a otro de un modo relativamente fijo, y que tienden a un determinado resultado o final.”

Es así, como se los diferencia dependiendo del contexto donde se los trate. Se habla de procesos industriales, químicos, biológicos, económicos, entre otros.

Un proceso industrial se refiere básicamente a un proceso típico de una industria donde es absolutamente necesario controlar y mantener constantes algunas magnitudes, tales como nivel, temperatura, presión, caudal, pH, conductividad, velocidad, posicionamiento, tamaño, peso, humedad, punto de rocío, etcétera; las cuales son manejadas para darle paso a otra parte que sucede a la operación anteriormente hecha. La variable controlada es la cantidad o condición que se mide y controla. La variable manipulada es la cantidad o condición modificada por el controlador, a fin de afectar la variable controlada. Normalmente la variable controlada es la salida del sistema.

Cuando se habla de controlar un proceso se busca optimizar los resultados; de alguna forma, si un proceso es manejado por humanos, como sucedía en los inicios de la era industrial, donde el operario realizaba o realiza un control manual, se corre el riesgo de que se cometan errores, que falle por algún motivo los sensores y/o actuadores y las personas no se percatan de la situación, ya que solamente se contaba con instrumentos de medida y actuadores simples. Este

control era suficiente por la simplicidad de los procesos. Sin embargo, la gradual complejidad con que éstos se han ido desarrollando ha exigido el control automático, con el cual se desea evitar esos errores y procurar que las personas estén simplemente monitoreando, vigilando que todo se cumpla satisfactoriamente. Control significa medir el valor de la variable controlada del sistema, y aplicar al sistema la variable manipulada para corregir o limitar la desviación del valor medido, respecto al valor deseado.

Además, un proceso como ya se dijo es una serie de operaciones graduales y el paso de una operación a otra es determinada por el estado de unos sensores que según hayan sido programadas sus salidas dan paso a una operación siguiente. Con el control manual esto es muy dependiente de la calidad y capacidad de los operarios y no garantizan que lo que se hace, esté bien hecho; en cambio, con el control automático se está garantizando fiabilidad al proceso. Esto está demostrado que ha mejorado la productividad que tanto se busca.

En el proceso de diseño se podría decir que se sigue un procedimiento muy general y es que dada una planta industrial, primero se deben elegir sensores y actuadores apropiados. Después de esto, como ya se ha manifestado hay que hacer un modelado matemático adecuado de la planta, así como también de los sensores y los actuadores. Utilizando esos modelos matemáticos construidos, se pasa a diseñar un sistema de control de tal manera que se satisfagan las especificaciones de diseño dadas cuando se plantea el objetivo a seguir, o el proceso a controlar. El controlador que aquí se diseña es una parte de la solución del problema. Se habla de que es solo una parte por cuanto solo es la solución a la versión matemática.

Una vez se haya completado el diseño matemático se entra a simular dicho modelo en un programa de computadora con el fin de vislumbrar una aproximación de cómo respondería el sistema final a diversas señales que le entren. Aquí se verifica el comportamiento de sensores y actuadores.

Este proceso involucra diseño y análisis y se ha terminado solo hasta cuando se esté satisfecho de lo que se observa y ya se puede entrar a construir un prototipo del sistema físico, que represente al modelo matemático, pero, ya con una exactitud razonable, si bien, se asemeja o se entra a evidenciar la realidad. Se deben hacer las pruebas necesarias con este prototipo y el diseño se completa cuando la respuesta del sistema sea completamente satisfactoria.

En los sistemas de control también se pueden usar modelos normalizados, tal es el caso de controladores PID, básicamente, donde no existe un mecanismo de diseño exacto a seguir, sino solo con la técnica del tanteo, de obtener unos parámetros exactos o aproximados y así se pueden conseguir respuestas esperadas. Todo esto depende de la aplicación para la cual se esté realizando los sistemas de control.

El hecho de hacer una investigación detallada del comportamiento de una planta, de un proceso, cuyo modelamiento matemático ya conocido, se lo identifica como análisis.

Un sistema de control electrónico es usado para controlar desde una central local o remota al proceso, el estado de unos actuadores, basado en el estado que presenten en un determinado instante unos sensores.

Sensores son elementos que brindan información del estado de unas variables en un tiempo determinado. Variables como nivel, presión, temperatura, entre otras; y los sensores miden o censan si los estados actuales de esa variable coinciden con el estado que se les ha programado previamente a los sensores.

Los actuadores por su parte, son elementos que basándose en unas órdenes provenientes del sistema de control actúan sobre las variables y las hacen seguir un rumbo predeterminado.

Antes de poder realizar un sistema de control es preciso hacer un modelado de la planta a tratar, de los actuadores utilizados, tal es el caso de válvulas, motores, etc.

El modelado matemático se usa para no operar con los dispositivos físicos como tales, ya que los sistemas de control abarcan componentes electromecánicos, hidráulicos, neumáticos, electrónicos y asociarlos todos físicamente por medio de expresiones puede resultar un problema casi imposible de solucionar.

Para hacer el modelado se pueden seguir varios métodos de análisis que existen. Si se basa en la teoría de control clásica, está el método del lugar geométrico de las raíces, el mecanismo de análisis por espacio de estado, donde se toman en cuenta sistemas lineales, invariantes en el tiempo. Generalmente se analizan sistemas con una sola entrada y una salida. Se pueden usar el método de las funciones de transferencia que basándose en ecuaciones diferenciales relaciona una variable que se considera como de salida, con una variable de entrada. Para construir las ecuaciones diferenciales es muy común usar la Transformada de Laplace y aplicarla a las ecuaciones de cada variable o sección que hayan sido sacadas.

Otro mecanismo muy comúnmente utilizado es el análisis en el dominio de la frecuencia.

Sin embargo, ya que no todos los sistemas son lineales, invariantes en el tiempo surgió la teoría de control moderna cuyas técnicas de modelamiento permiten relacionar o analizar sistemas con múltiples entradas y múltiples salidas, además de los aspectos de no linealidad y dependencia del tiempo ya expuestos. Estos

sistemas también involucran gran cantidad de operaciones matemáticas que el usuario debe manipular, lo cual incrementa la complejidad.

Los sistemas de control con alto grado de calidad y optimización cumplen con unos requisitos generales que se deben tener presente a la hora de diseñar. Todo sistema de control debe ser estable. La respuesta del sistema que tanto se ha hablado debe tener una estabilidad relativa razonable, es decir, un amortiguamiento aceptable. Así mismo, la velocidad de respuesta debe ser rápida, ya que por lo general los procesos industriales así lo exigen. Ya que el sistema de control depende del tiempo, hay dependencia tanto de las señales de entrada (sensores) como de las señales de salida (actuadores) los cuales precisamente ejecutan, realizan el procedimiento que el usuario espera.

El sistema de control debe ser capaz de reducir los errores a un valor muy cercano a cero, ya que por lo general hay perturbaciones en estos sistemas físicos reales, los cuales determinan la existencia de señales externas indeseadas; así, como también, los equipos que son utilizados llevan consigo un rango o margen de error que aunque es bajo, se busca que éste no se incremente al unificar varios de estos dispositivos físicos.

ANEXO F RUTINAS DESARROLLADAS EN MPLAB CONJUNTAMENTE CON LOS ESTUDIANTES

“Rutina de Planeador Cíclico”

```

list    p=16f877          “lista directa para definir procesador”
#include <p16f877.inc>    “definición de variables específicas del procesador.”
#include <protreg.inc>

__config _cp_off & _wdt_off & _boden_off & _pwrt_e_off & _xt_osc &
_wrt_enable_off & _lvp_off & _debug_off & _cpd_off

intadq equ  d'20' “intervalo interrupcion: intadq = 255 - (tadq*fosc)/4”
intpln equ  d'20' “intervalo ejecucion de planeador intpln*intadq”
numfil equ  d'8'  “parametros de adq”
num_canal equ d'5' “define el numero de canales a utilizar”
extern tarea0,tarea1,tarea2

code    org  0x000          “ vector de reset”
        clrf  pclath        “ bits de pagina son clareados”
        goto  main          “ ir a comenzar el programa principal”
        org  0x004          “ vector de interrupción”
        goto  atn_int
        org  0x005
atn_int movwf  w_temp0      “ preservar el registro w”
        movf  status,w
        movwf status_temp0 “ preservar el registro status”
        nop
        movlw intadq        “ carga de tmr0”
        movwftmr0
        bcf   intcon,2
est_conf movlw num_canal
        sublw 0
        btfsc status,z
        goto  act_pln
        btfss estado,7     “toma la decisión según el estado”
        goto  adc1         “estado 1”
        goto  adc2         “estado 2”

“conversión analógica y transferencia rápida a través de puerto paralelo esclavo
ciclo de solicitud, conversión y envío de segundo byte de adquisición”
adc2    bsf   adcon0,go    “inicia conversión”

```

	bcf	status,rp0	“banco 0”
entre tareas”	bcf	estado,7	“clarea el bit 7 del registro estado para alternar
	swapf	conta,w	“toma el contador del canal y lo adecúa “para enviar por paralelo”
	movwf	temp	
	bcf	status,c	“limpia el carry”
	rlf	temp,1	“rota a la izquierda temporal”
	swapf	conv2,w	“se adecúa para enviar 5 bit + numero del canal”
	movwf	temporal	
	bcf	status,c	
	rrf	temporal,w	
	movwf	byte_one	
	swapf	conv1,w	
	movwf	temporal	
	bcf	status,c	
	rrf	temporal,w	
	andlw	b'00000111'	
	xorwf	byte_one,w	
	xorwf	temp,w	
	movwf	byte_one	“guarda la primera parte del envío en registro”
	movlw	b'00011111'	
	andwf	conv1,w	
	xorwf	temp,w	
	movwf	byte_two	“guarda la segunda parte del envío en registro”
	btfss	pir1,pspif	“prueba si se ha producido ininterrupción por pp”
	goto	act_pln	“sino se va al planeador”
	movwf	portd	“si hay ininterrupción va por puerto paralelo”
	bcf	pir1,pspif	“se limpia para nueva ininterrupción”
	movwf	portd	“envía dato a puerto paralelo”
	goto	act_pln	“va a análisis y toma de decisiones”
“ciclo de recolección, almacenamiento y envío de primer byte de adquisición”			
adc1	movf	adresh,w	“mueve la conversión a w”
	movwf	conv1	“conversión se guarda en registro”
	bsf	status,rp0	“cambia al banco 1”
	movf	adresl,w	“mueve la conversión a w”
	bcf	status,rp0	“cambia al banco 0”
	movwf	conv2	“conversión se guarda en registro”
	movf	in_an,w	“se toma el numero de entradas y guarda en w”
	incf	conta,1	“incrementa contador de canal”
	subwf	conta,w	“hace la resta entre w y contador de canal”
	btfss	status,c	“prueba si la resta dio cero”
	goto	ok_ch	“sino va a adecuar el contador de canal”
	clrf	conta	“de lo contrario reinicia el contador de canal”

```

ok_ch      movf  conta,w      "mueve el contenido del contador a w"
           movwfmensaje "mueve el contenido de w a mensaje"
           swapf mensaje,1 "hace las rotaciones correspondientes"
           rrf  mensaje,1 "escoge canal"
           movlwb'10000001' "con esta palabra configuramos"
           andwf adcon0,1
           xorwf mensaje,w "frec oscilación, adon encendido"
           iorwf adcon0,f "los bit 5,4,3 conf, canal"
           movf  byte_one,w
           bsf  estado,7 "se coló"
           bsf  adcon0,adon
           nop
           btfss pir1,pspif "prueba si se ha producido interrupción por pp"
           goto  act_pln "sino se va al planeador"
           movwfpord "si hay interrupción manda por puerto paralelo"
           bcf  pir1,pspif "se limpia para nueva interrupción"
           goto  act_pln "va a análisis y toma de decisiones retorna"
"activación del planeador cíclico"
act_pln    incf  inpl,f
           movlwintpln
           xorwf inpl,w
           btfsc status,z
           call  planeador
           movf  status_temp0,w
           movwfmensaje "mueve status register a registro w"
           movf  w_temp0,w "salva contenido de registro w"
           retfie

"planeador cíclico: hasta 16 tareas, con un numero de palabras de numfil"
planeador  movwf  w_temp1 "preservar el registro w"
           movf  status,w
           movwfmensaje "preserva el registro status"
           clrf  inpl "clarea los contadores 2-3"
           movlw d'1'
           movwffpos
principal  call  tabla_1
           movwffila
mov_fila_a rrf  fila,f
           btfsc status,c
           call  tarea
           incf  fpos,f
           movlw d'9'
           xorwf fpos,w
           btfss status,z
           goto  mov_fila_a

```

```

call    tabla_2
movwffila
movlw d'0'
xorwf  fila,w
btfss  status,2
goto   mov_fila_b
movlw d'16'
movwffpos
mov_fila_b  rrf    fila,f
btfsc  status,c    "setea el bit de carry. si este es un 1 "
call   tarea      "se realiza la llamada a tarea"
incf   fpos,f      "va al siguiente bit sobre la palabra de activación"
movlw d'17'       "determina si es ultimo bit de activación de tarea"
xorwf  fpos,w
btfss  status,z
goto   mov_fila_b
incf   apfl,f      "apunta a la siguiente palabra de activación – fila"
movlw d'1'
movwffpos        "reinicia apuntador de bit sobre palabra"
movlw numfil     "identifica ultima palabra de activación"
xorwf  apfl,w
btfss  status,z
goto   restaurar ;
clrf   apfl      "va a primera palabra si se ha terminado la lista"
restaurar  movf   status_temp1,w "retrae una copia del registro status"
movwffstatus "mueve lo que hay en w al registro de status"
movf   w_temp1,w "guarda el contenido del registro w"
return   "retorna de la interrupción"
"palabras de activacion"
tabla_1   movf   apfl,w
addwf  pcl,f
dt     07h,03h,07h,03h,07h,03h,07h,03h

```

“valores representados en hexadecimal los cuales muestran los 8 primeros de los 16 bits que modificaremos según los requerimientos”

```

tabla_2   movf   apfl,w
addwf  pcl,f
dt     00h,00h,00h,00h,00h,00h,00h,00h

```

“valores representados en hexadecimal los cuales muestran los 8 últimos de los 16 bits que se modifican según requerimientos. Si no existen más de 8 tareas estos bits serán ceros”

“fuente de tareas”

```

tarea     movf   fpos,w
addwf  pcl,f

```

```

nop
goto tarea0      "va a la tarea asignada por los bits de las tablas"
goto tarea1
goto tarea2
goto tarea3
goto tarea4
goto tarea5
goto tarea6
goto tarea7
goto tarea8
goto tarea9
goto tarea10
goto tarea11
goto tarea12
goto tarea13
goto tarea14
goto tarea15

```

"Desarrollo de tareas"

```

main      clrf   inpl
          clrf   apfl
          movlw d'1'
          movwffpos
          bcf   status,rp1
          bsf   status,rp0

```

"inicialización de puertos"

```

clrf   trisb   "puerto b como salidas"

```

"inicialización interrupción periódica"

```

movlw b'00001000'
"rbpu intedg tocs tose psa ps2 ps1 ps0"
" | | | | | 000: preescala 1:1"
" | | | | | preescalar asignado a wdg"
" | | | | | flanco subida temporizador"
" | | | | | reloj interno para temporizador"
" | | | | | flanco señal int como caída"
" pull-up habilitado"
movwfoption_reg "inicialización pp esclavo y entradas análogas"
movlw b'10000010' "adfm - - - pcf3 pcf2 pcf1 pcf0"
"configuración de número de entradas análogas"
"configura justificación 1: derecha 0:izquierda "
movwfadcon1      "5 entradas análogas - puertos a.0-1-4 y e digital"
                  "datos en adresl y 2 bits en adresh."
movlw b'00010111' "configuración de puerto paralelo esclavo"
"ibf obf ibov pspmode - direccion datos"
" | | | | | 111 : entradas 'cs 'wr 'rd"

```

```

“ | | | 1: habilitación puerto esclavo”
“ | | bandera detección de sobreflujo”
“ | bandera buffer de salida lleno”
“ bandera buffer de entrada lleno”
movwftrise      “carga el registro correspondiente”
bcf  pie1,pspie “desactivación interrupción puerto paralelo”
bcf  status,rp0
bsf  intcon,gie “habilitador de interrupciones global”
bsf  intcon,t0ie
bsf  adcon0,adon
movlwintadq     “carga valor de intervalo de adquisición”
movwftmr0
movlw num_canal
movwfin_an
ciclo           “rutina de analisis y toma de decisiones”
goto  ciclo
end

```

“Rutina de Comunicación Serial”

```

list          p=16f877      “lista directa para definir procesador”
#include      <p16f877.inc>  “definición de variables específicas del procesador”

dato_rx      equ 20h
paridad      equ 21h

__config __cp_off & __wdt_off & __boden_off & __pwrte_off & __hs_osc &
__wrt_enable_off & __lvp_off & __debug_off & __cpd_off

code
org 0x000     “vector de reset”
goto main    “ir a comenzar el programa principal”
org 0x005
main         bsf  status,rp0
movlw b'00000000'
movwf trisc  “configura los pines para la transmisión serie”
movlw 0x00
movwf intcon “configura interrupciones globales y periféricas”
movlw 0x00
movwf pie1   “configura pie1”
movlw b'00000110' “inicialmente tenia 0x00”
movwf txsta  “configura txsta”
movlw d'25'  “inicialmente tenia d 51 -1200 baudios”
movwf spbrg
clrf  trisb

```

```

                bcf    status,rp0
                movlw 0x90
                movwf rcsta
                movlw 66h
                movwf portb
                clrf   portc
recep          btfss  pir1,rcif    "testea si el buffer de rx esta lleno"
                goto  recep
                movf  rcreg,w
                movwf dato_rx
                movf  rcsta,w
                andlw b'00000110' "enmascara los bits feer y oeer"
                btfss status,z
                bcf   rcsta,cren
                bsf   rcsta,cren
                movf  dato_rx,w
                movwf portb
vacio1        movwftxreg
                bsf   status,rp0
                bsf   txsta,txen
trel          btfss  txsta,trmt    "testea si el shift register de tx esta vacio"
                goto  trel
                bcf   txsta,txen
                bcf   status,rp0
                goto  recep
end

```

"Rutina de prueba de activación de Relés"

```

list    p=16f877                "lista directa para definir procesador"
include <p16f877.inc>           "definición variables específicas del procesador"
ref     equ    20h
ton     equ    21h
toff   equ    22h                "Asignación de espacio de memoria a registros
ton_auxequ 23h                "propios del programa "
toff_auxequ 24h
c1      equ    25h
c3      equ    26h
c5      equ    27h

inicio  bsf    status,rp0
        bcf    trisb,7          "configura pin de Puerto b como salida"
        bcf    status,rp0
        bsf    portb,7         "setea salida de relé"

```

```

        movlw .100                "carga valor para retardo"
        movwf ref
s3      movlw .50                  "incrementa valor del retardo"
        movwfton
        movwfton_aux
        subwf ref,0
        movwftoff
        movwftoff_aux
s1      nop
        decfsz ton,1
        goto s1
        movlw .0
        movwf portb
s2      nop
        decfsz toff,1
        goto s2
        movlb '10000000'
        addwf portb
        goto s5
s5      movlw .100
        movwfc1
ciclo1 decfszc1
        goto c3c
        goto s3
c3c    movlw .255
        movwfc3
ciclo3 decfszc3
        goto c5c
        goto ciclo1
c5c    movlw .255
        movwfc5
ciclo5 decfszc5
        goto ciclo5
        goto ciclo3
end

```

"Rutina de prueba para salidas PWM"

```

list    p=16f877
include <p16f877.inc>
ref     equ 20h
ton     equ 21h
toff    equ 22h
ton_aux equ 23h

```

toff_aux	equ	24h	
c1	equ	25h	
c3	equ	26h	
c5	equ	27h	
c4	equ	28h	
c6	equ	29h	
c2	equ	30h	
in	equ	31h	
	extern	deci, buffer3	
	org	00h	
	goto	main	“ va a rutina principal”
	org	05h	
inicio	bsf	status,rp0	“se cambia al banco 1”
	bcf	trisc,3	“activa bit 3 del Puerto c como salida”
	bcf	status,rp0	“se cambia al banco 0”
c0	btfsc	deci,4	“testea estado anterior de la bandera”
	goto	main	“va a principal”
	bsf	portc,3	“pone en alto la salida de la señal PWM”
	movfw	in	“ mueve el valor del registro in a w”
	movwfc	c2	“lo lleva a c2”
cl0	decfszc	c2	“decrementa el valor y salta si es cero”
	goto	ciclo2	“va a ciclo2”
	goto	s3	“va a la etiqueta s3”
ciclo2	movlw	.100	“mueve al acumulador el valor 100”
	movwfc	c4	“mueve el valor a c”
cl2	decfszc	c4	“decrementa y salta si es cero”
	goto	ciclo4	“va a ciclo4”
	goto	cl0	
ciclo4	movlw	.100	“mueve 100 a c6”
	movwfc	c6	
cl4	decfszc	c6	“decrementa hasta que sea cero”
	goto	cl4	
	goto	cl2	“va a cl2”
s3	bcf	portc,3	“pone en bajo la salida de la señal PWM”
	movlw	.100	
	subwf	in	“resta el valor de la entrada de 100”
	movwfc	c1	“el resultado lo lleva a c1”
ciclo1	decfszc	c1	“decrementa ese valor hasta que sea cero”
	goto	c3c	
	goto	c0	“va a c0”
c3c	movlw	.100	
	movwfc	c3	“mueve 100 a c3”
ciclo3	decfszc	c3	
	goto	c5c	“decrementa hasta que sea cero”
	goto	ciclo1	“regresa a ciclo1”

c5c	movlw .100	
	movwfc5	“mueve 100 a c5”
ciclo5	decfszc5	
	goto ciclo5	“decrementa hasta que sea cero”
	goto ciclo3	“va a ciclo3”
main	movlw buffer3	“lleva el valor de buffer3 a w”
	movwffsr	“de ahí lo lleva al apuntador de memoria”
	movfwindf	
	movwfin	
	goto inicio	“va a iniciar el programa”
	end	

“Rutina de carga de datos y almacenamiento en memoria”

```
list p=16f877
include <p16f877.inc>
    udata
    org 05h
```

sentido_giro	res	1	“Registro de sentido de giro de los motores dc”
comp_com	res	1	“Registro para comprobar comando o a dato”
inf_vel	res	1	“Registro de velocidad “
dato0	res	1	
dato1	res	1	
dato2	res	1	
dato3	res	1	
dato4	res	1	“Registros temporales de velocidad del motor”
w_temp	res	1	
status_temp	res	1	
uni_vel	res	1	
sel_mot	res	1	
nro_ret	res	1	
unidades	res	1	
tipo_paso	res	1	
nro_con	res	1	
veloc	res	1	
pasos_l1	res	1	
pasos_l2	res	1	
pasos_l3	res	1	
pasos_h	res	1	
cont_pos	res	1	
ban_con	res	1	
ban_vel	res	1	
inf_per	equ	.24	

global	w_temp, status_temp, start, tipo_paso, unidades, nro_ret	
global	uni_vel, dato0, dato2, dato3, dato4, sel_mot, sentido_giro, dato1, inf_vel, comp_com	
global	veloc, pasos_l1, pasos_l2, pasos_l3, pasos_h, cont_pos, ban_con, ban_vel, nro_con	
extern	motordc_1, motordc_2, pap2	
start	goto main movlw 50h movw fsr movw indf movw dato0 movw sel_mot incf fsr movw indf movw dato1 incf fsr movw indf movw dato2 incf fsr movw indf movw dato3 incf fsr movw indf movw dato4	“Se define velocidad y tipo de motor”
sel_motor	bcf pir1,1 movw sel_mot andlw d'3' movw sel_mot btfsc status,z goto motordc_1 decf sel_mot,1 btfsc status,z goto motordc_2 goto pap2	“cleara bit 1 de pir1, tras tmr2 = pr2” “Motor es el dc 1” “si no es motor dc evalúa motores PAP”
main	movw w_temp movw status movw status_temp clrf ccpr1 movlw b'00000101' movw ft2con movlw b'00001100'	“preserva valor inicial de ccpr1 en "0”” “habilita tmr2, establece predivisor en 1:4”

	addwf ccp1con,f	“configura modo pwm para motor dc 1”
	clrf ccpr2l	“preserva valor inicial de ccpr2l en "0"”
	movlw b'00001100'	
	addwf ccp2con,f	“configura modo pwm para el motor dc 2”
	bsf status,rp0	
	bcf status,rp1	“accede al banco 1”
	movlw inf_per	“carga pr2 con el periodo de la señal”
	movwf pr2	
	bcf trisc,0	
	bcf trisc,2	“habilita pines "0" y "2" del puerto c”
	bcf trisc,1	
	bcf trisc,3	“habilita pines "1" y "3" del puerto c”
	movlw b'00000000'	
	movwf pie1	“habilita interrupción pr2 = tmr2”
	bcf intcon,6	
	bcf intcon,7	
	clrf portb	
	movlw b'00000111'	
	movwf porta	
	bcf status,5	
	clrf uni_vel	
carga_datos	movlw 50h	
	movwf fsr	
	movlw b'01000111'	
	movwf indf	
	incf fsr	
	movlw b'00110010'	
	movwf indf	
	incf fsr	
	movlw b'00100111'	
	movwf indf	
	incf fsr	
	movlw b'00010001'	
	movwf indf	
ciclo	movlw d'0'	
	nop	
	xorwf tmr2,w	
	btfss status,z	
	goto ciclo	
	goto start	
end		

“Rutina de control de motor DC”

```
list p=16f877
include <p16f877.inc>

global motordc_1
extern dato0, sentido_giro, dato1, inf_vel, comp_com, dato2, dato3, w_temp,
        status_temp, start
code
motordc_1  movfwdato0          “carga w con dc 1 y sentido de giro escogido”
           andlw b'00001000'
           movwfsentido_giro
           rrf  sentido_giro,f
           rrf  sentido_giro,f          “lleva bit de sentido de giro al registro
           rrf  sentido_giro,f          “para sacarlo por bit "0" del puerto c”
sent_giro  movlw d'1'              “comprueba el sentido de giro”
           xorwf sentido_giro,w
           btfss status,z           “1” giro izquierda ó "0" giro derecha”
giro_der  bcf  portc,0
giro_izq  bsf  portc,0
ancho_pulso movfwdato1            “carga en w el byte con el ancho de pulso”
           movwfinf_vel            “ se lleva al registro asignado para tal fin”
           movwfcomp_com
           rlf  comp_com,f          “es dato o comando?”
           btfsc status,c          “comando= va a fin y espera interrupción”
           goto fin                “dato= se continúa con el programa”
           rrf  comp_com,f
           swapf comp_com,f
           movf comp_com,w          “cambia nibbles y verifica 1er paq.”
           andlw b'00000111'
           xorlw d'2'
           btfss status,z
           goto fin
           movlw b'00110000'        “cambia nibbles de inf_vel y elije bits 4 y 5”
           swapf inf_vel,f          “corresponden 2 bits menos significativos
           andwf inf_vel,w          “de ancho de pulso”
           addwf ccp1con,f
           movlw b'00001100'
           swapf inf_vel,f
           andwf inf_vel,w
           movwfccpr1l
           rrf  ccpr1l,f
           rrf  ccpr1l,f          “Lleva 2 bits del de la velocidad al ccpr1l”
           movfwdato2
           movwfinf_vel
```

```

movwf comp_com
rlf comp_com,f
btfsc status,c
goto fin
rrf comp_com,f
swapf comp_com,f
movf comp_com,w
andlw b'00000111'
xorlw d'1'
btfss status,z
goto fin
rlf inf_vel,f
rlf inf_vel,f
movlw b'00111100'
andwf inf_vel,w
addwf ccpr1l,f
movf wdata3
movwf inf_vel
movwf comp_com
rlf comp_com,f
btfsc status,c
goto fin
rrf comp_com,f
swapf comp_com,f
movf comp_com,w
andlw b'00000111'
xorlw d'0'
btfss status,z
goto fin
rlf inf_vel,f
rlf inf_vel,f
movlw b'00001100'
andwf inf_vel,f
swapf inf_vel,w
addwf ccpr1l,f
movfw status_temp
movwf status
movfw w_temp
movlw 50h
movw ffsr
incf indf
goto inicio

```

“elige 4 bits de inf para bits 2-5 del ccpr1l”

“ elige sig. byte y lo cargo en los registros”

“2 lsb de inf los llevo a bits 6 y 7 de ccpr1l”

“regresa valores guardados de w y status”

end

“Rutina de Control de motor Paso a paso”

```
list          p=16f877
include       <p16f877.inc>
```

```
global cont_pos1, cont_pos2
```

```
extern dato1, comp_com, tipo_paso, cont_pos, dato2, pasos_l1, dato3, pasos_l2,
        dato4, pasos_h, pasos_l3, ban_con, sentido_giro, start, dato0
```

```
code
```

```
cont_pos1    movfwdato1          « carga en w el byte del 1er paquete de inf ».
              movwfcomp_com
              movwftipo_paso
              movwfcont_pos
              rlf    comp_com,f      « comprueba si es dato »
              btfsc status,c
              goto  start
              rrf    comp_com,f
              swapf comp_com,w
              andlw d'7'
              xorlw d'3'
              btfss status,z        « comprueba nro de paquete recibido »
              goto  start
              rrf    tipo_paso,f
              rrf    tipo_paso,f    « almacena el tipo de paso »
              movlw d'3'           « "0" medio paso "1" paso completo »
              andwf cont_pos,1
              btfsc status,z        « almacena tipo de funcionamiento »
              goto  laz_ab
              decf  cont_pos,1
              btfsc status,z
              goto  laz_cerr

servomotor   call  inf_posi
              call  ciclo3          « ciclo activa salidas del motor »
              goto  start

laz_ab       call  inf_posi
              goto  start

laz_cerr     call  ciclo4
              goto  start

inf_posi     movfwdato2
              movwfcomp_com
              movwfpasos_l1
```

<pre> rif comp_com,f btfsc status,c goto start rif comp_com,f swapf comp_com,w andlw d'7' xorlw d'2' btfsc status,z goto start movlw d'15' andwf pasos_l1,f movfwdato3 movwfc comp_com movwfpasos_l2 rif comp_com,f btfsc status,c goto start rif comp_com,f swapf comp_com,w andlw d'7' xorlw d'1' btfss status,z goto start movlw d'15' andwf pasos_l2,f swapf pasos_l2,w addwf pasos_l1,f movfwdato4 movwfc comp_com movwfpasos_h rif comp_com,f btfsc status,c goto start rif comp_com,f swapf comp_com,w andlw d'7' btfss status,z goto start movlw d'15' andwf pasos_h,f incf pasos_h,f movwfpasos_l1 movwfpasos_l3 return movlw d'1' </pre>	<pre> « comprueba si es dato » « mira nro de paquete de información » « elige 4 lsb de pasos_l1 » « que llevan 1ra parte de información » « verifica si es dato y no un comando » « mira nro de paquete de información » « elige 4 lsb de pasos_l2 que llevan « la 2da parte de la inf. del nro de pasos » « verifica que se trata de dato. » </pre>
---	--

ciclo3

	movwfban_con	« pone en uno la bandera del contador »
	movfwpasos_l1	
	movwfpasos_l3	
	btfss tipo_paso,0	« es paso completo o medio paso »
	goto med_paso3	
paso_comp3	btfsc sentido_giro,0	« analiza el sentido de giro »
	goto a_izqu3	
	call gira_der3	
	return	
a_izqu3	call gira_izq3	
	return	
gira_izq3	movlw b'10000000'	« primer paso. »
	btfss ban_con,0	« testea bit 0 de la bandera del contador »
	return	
	call activasalida2	
	movlw b'11000000'	« segundo paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'01000000'	« tercer paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'01100000'	« cuarto paso »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'00100000'	« quinto paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'00110000'	« sexto paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'00010000'	« séptimo paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'10010000'	« octavo paso »
	btfss ban_con,0	
	return	
	call activasalida2	
	btfss ban_con,0	
	return	

	goto gira_izq3	« si se ha cumplido el ciclo y la bandera
del contador	aún no ha cambiado de estado reinicia el ciclo de gira izq3 »	
gira_der3	movlw b'10010000'	« primer paso »
	btfss ban_con,0	
	return	
	call activasalida2	« envía a salida del motor pap »
	movlw b'00010000'	« segundo paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'00110000'	« tercer paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'00100000'	« cuarto paso »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'01100000'	« quinto paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'01000000'	« sexto paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'11000000'	« séptimo paso. »
	btfss ban_con,0	
	return	
	call activasalida2	
	movlw b'10000000'	« octavo paso »
	btfss ban_con,0	
	return	
	call activasalida2	
	btfss ban_con,0	
	return	
	goto gira_der3	
med_paso3	btfsc sentido_giro,0	« analiza sentido de giro »
	goto a_izq3	
	call gira_derech3	
	return	
a_izq3	call gira_izquier3	
	return	
gira_izquier3	movlw b'11000000'	« primer paso. »

```

    btfss ban_con,0
    return
    call activasalida2
    movlw b'01100000'      « segundo paso »
    btfss ban_con,0
    return
    call activasalida2
    movlw b'00110000'      « tercer paso. »
    btfss ban_con,0
    return
    call activasalida2
    movlw b'10010000'      « cuarto paso »
    btfss ban_con,0
    return
    call activasalida2
    btfss ban_con,0
    return
    goto gira_izquier3
gira_derech3 movlw b'00010000'      « primer paso. »
    btfss ban_con,0
    return
    call activasalida2
    movlw b'00100000'      « segundo paso »
    btfss ban_con,0
    return
    call activasalida2
    movlw b'01000000'      « tercer paso. »
    btfss ban_con,0
    return
    call activasalida2
    movlw b'10000000'      « cuarto paso »
    btfss ban_con,0
    return
    call activasalida2
    btfss ban_con,0
    return
    goto gira_derech3

activasalida2 movwf portb      « saca el valor del paso correspondiente
    por los bits del puerto b »
    decf pasos_l3,1
    btfsc status,z
    decf pasos_h,1
    btfsc status,z
    goto term

```

```

        movfwpasos_l1
        movfwpasos_l3
        return

term      decf  ban_con,1      « pone en cero la bandera del contador »
          return
include <retardos.inc>
end

```

“Rutina del control de humedad”

“este programa realiza una inspección en el horno ahumador para
“asegurar que la humedad en el ahumado no sea mayor al 60%”

```

list      p=16f877
include<p16f877.inc>

dato_sh   equ 20h
actuadores equ 22h
ran_hum   equ 23h
ref       equ 25h
in_hu     equ 26h
conta2    equ 27h

“ parámetros temperatura grados”
ctrl_hum  equ d'60'      “para 60%”

        org 0h
        goto main
        org 05h
inicio

control_h  movlw ctrl_hum
           subwf dato_sh,0 “resta el valor de rango_min 20º a la tº del Im35”
           btfsc status,c “revisar si el resultado es + - ò cero”
           “sí+ revisar rango máximo “
           goto enc_estrac “no- aumentar tº del horno”
           goto rango_deter
main      bcf  status,5
           movlw .152
           movwfin_hu
           goto conv_hu_re
enc_estrac goto conv_hu_re
rango_deter goto conv_hu_re

```

```

conv_hu_re  clrf   conta2
              movlw .10
              movwfwref
ciclo2      movwfwref
              subwf in_hu,1
              btfss status,c
              goto  erro2
              incf   conta2
              goto  ciclo2
erro2       rlf    conta2
              rlf    conta2
              addwf in_hu,1
              rrf    in_hu,1
              movlw .15
              andwf in_hu,1
              movfw in_hu
              addwf conta2,1
              movfw conta2
              movwfw dato_sh
              goto  inicio
end

```

“Rutinas del control de Temperatura”

“este programa realiza una inspección en el horno ahumador para “
“asegurar que la temperatura en el ahumado frío no sea < 20°C ò > 25°C”

```

list    p=16f877
include<p16f877.inc>

```

```

dato_st    equ 20h
actuadores equ 22h
ran_min    equ 23h
ran_max    equ 24h
ref        equ 25h
mndo       equ 26h
conta      equ 27h

```

“ parametros temperatura grados”

```

ctrl_min    equ d'20'
ctrl_max    equ d'25'

```

```

org 0h
goto main

```

```

    org    05h
inicio
    movlw d'35'           "valor para pruebas de código simulando"
    movwfdato_st         "una captura de temperatura por el lm35"
control_t    movlwctrl_min  "se carga el valor de rango mínimo para el
control de la temperatura en un ahumado frío"
    subwf dato_st,0      "resta el valor de rango_min 20° a la t° del lm35"
    btfss status,c      "revisar si el resultado es + - ò cero"
                    "sí+ revisar rango máximo"
    goto  enc_calef      "no- aumentar t° del horno"
    movwfdato_st         "revisar si sobrepasa rango_max 25°c"
    sublw ctrl_max      "se resta el valor rango max (permisible en la
camara) a la lectura del lm35"
    btfsc status,c      "revisar si el resultado es + - ò cero "
    goto  rango_deter
    goto  enc_estrac
main    bcf    status,5
    movlw .47
    movwfmndo
    goto  conv_grados    "convierte el valor obtenido "

enc_calef    goto  conv_grados
            return
enc_estrac  goto  conv_grados
            return
rango_deter goto  conv_grados
            return

```

“en esta parte de la rutina se realizara la conversión del valor captado por el sensor de temperatura. El resultado de la conversión será en grados centígrados

```

conv_grados clrf    conta
            movlw .10
            movwfreq
ciclo      movfwref
            subwf mndo,1
            btfss status,c
            goto  error1
            incf  conta
            goto  ciclo
error1     rlf    conta
            rlf    conta
            addwf mndo,1
            rrf    mndo,1
            movlw .15

```

```

        andwf mndo,1
        movfw mndo
        addwf conta,1
        movfw conta
        movwf dato_st
        goto inicio
end

```

“Rutina prueba de motores Paso a paso”

```

list    p=16f877
        include<p16f877.inc>

```

```

c_pasos    equ    40h
            org    0h
            org    05h
enc_estruc movlw .9           “introduce el valor de nro de pasos deseado”
            movwfc_pasos
            goto   gira_der    “va a girar a la derecha”
off_estruc movlw .9           “introduce el valor de nro de pasos deseado”
            movwfc_pasos
            goto   gira_izq    “va a girar a la izquierda”
gira_der   movlw b'00000001'   “activa el primer paso de la secuencia”
            call   activasalida “va a sacar el dato por el puerto indicado”
            decfsz c_pasos     “decrementa el nro de pasos. Si es cero terminó”
            return
            movlw b'00000010'   “ si no es cero activa el siguiente paso”
            call   activasalida
            decfsz c_pasos
            return
            movlw b'00000100'   “ si no es cero activa el siguiente paso”
            call   activasalida
            decfsz c_pasos
            return
            movlw b'00001000'   “ si no es cero activa el siguiente paso”
            call   activasalida
            decfsz c_pasos
            return
            goto   gira_der     “si aún no se ha terminado el nro de pasos va al
inicio de gira derecha”
gira_izq   movlw b'00001100'   “activa el primer paso de gira a izquierda”
            call   activasalida
            decfsz c_pasos     “decrementa nro de pasos y continúa el
            return           “programa siempre y cuando no sea cero”
            movlw b'00000110'

```

```
call  activasalida
decfszc_pasos
return
movlwb'00000011'
call  activasalida
decfszc_pasos
return
movlwb'00001001'
call  activasalida
decfszc_pasos
return
goto  gira_izq      "si no ha terminado nro de pasos va a gira izquierda"
```

```
activasalida  movwfportc      "saca la secuencia por el puerto c"
return
end
```