



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

FACULTAD DE INGENIERÍA

*“Desarrollo de panel de control para
máquina de inyección de plástico”*

TESIS

Que como parte de los requisitos para obtener el
grado de Ingeniero Eletromecánico

PRESENTA:

Isaías Cueva Pérez

ASESOR:

Dr. Roque Alfredo Osornio Ríos

San Juan del Río, Querétaro, Septiembre de 2014

A mis padres Everardo y Silvia, y a mi hermana Dulce María, este logro también es de ustedes, gracias por todo.

A mis amigos Andoni, Juan Carlos, Edgar, Juan, David, Florentino, Alejandro, Eduardo, Sergio, Susy, Ely, Paco y Amando, gracias por haber sido parte de esta maravillosa aventura.

A Sofi, mil gracias por todo tu apoyo y por estar siempre conmigo.

Índice general

Dedicatorias	I
Índice general	III
Índice de figuras	IV
I. Introducción	1
1.1. Antecedentes	3
1.2. Justificación	7
1.3. Objetivos	8
1.3.1. Objetivo General	8
1.3.2. Objetivos Particulares	8
1.4. Planteamiento General	9
II. Revisión de la literatura	11
2.1. Pantalla Táctil	11
2.2. Teclado	12
2.2.1. Controlador de Teclado	13
2.2.2. Escaneo de código	13
2.3. Protocolos de comunicación	14
2.3.1. RS232	14
2.3.2. PS/2	15
2.3.3. USB	16

ÍNDICE GENERAL

2.4. Interfaz Gráfica de Usuario	18
2.5. Bibliotecas GTKMM	19
2.6. Dispositivos reconfigurables	20
2.6.1. FPGA	20
2.6.2. PLC	21
2.7. La máquina de inyección de plástico	23
2.7.1. La unidad de inyección	23
2.8. El proceso de inyección de plástico	26
III Metodología	28
3.1. Diseño mecánico del panel	29
3.2. Diseño de interfaz gráfica	31
3.3. Programación de software	32
3.4. Fabricación de la estructura mecánica del panel	33
3.5. Integración de dispositivos periféricos y ensamble	34
3.6. Pruebas simuladas	38
IV Resultados	39
4.1. Diseño mecánico del panel	39
4.2. Diseño de la interfaz gráfica	40
4.3. Programación de software	44
4.4. Fabricación de estructura mecánica del panel	44
4.5. Integración de dispositivos periféricos y ensamble	47
4.6. Pruebas Simuladas	48
V. Conclusiones	55
Bibliografía	56
Apéndice	59

Índice de figuras

1.1. Esquema general de la propuesta del planteamiento del proyecto, la máquina inyectora es controlada mediante hardware que coordina entradas y salidas por medio de una tarjeta PLC que se comunica con el panel de control	9
1.2. Componentes físicos y su interconexión dentro del panel de control propuesto.	10
2.1. Panel industrial con pantalla táctil incluida.	11
2.2. Distribución común de un teclado de computadora.	13
2.3. Formas de onda de las señales de datos y de reloj de la interfaz USB.	16
2.4. Variantes de paquetes en la trama de comunicación de periférico USB a host.	17
2.5. Interfaz Gráfica de Usuario del Sistema Operativo Ubuntu.	19
2.6. Celda lógica en un FPGA.	21
2.7. Diagrama general de un PLC.	22
3.1. Componentes del panel de control propuesto.	28
3.2. Diagrama de la disposición de los elementos presentes en la carcasa del panel de control.	30
3.3. Secciones propuestas para la interfaz gráfica de usuario.	31
3.4. Diagrama general de las variables presentes en el proceso de inyección y su relación con la interfaz gráfica de usuario.	32
4.1. Vista isométrica del molde en CAD de la carcasa frontal y trasera del Panel de Control.	39
4.2. Sección principal de la interfaz gráfica de usuario.	40

ÍNDICE DE FIGURAS

4.3.	Segunda sección de la interfaz gráfica de usuario.	41
4.4.	Monitoreo de Presión en la interfaz gráfica de usuario propuesta.	42
4.5.	Sección de monitoreo de entradas de la interfaz gráfica de usuario propuesta.	43
4.6.	Parte frontal de la carcasa de fibra de vidrio del Panel de Control.	45
4.7.	Parte trasera de la carcasa de fibra de vidrio del Panel de Control.	45
4.8.	Placa de sujeción de la fuente de poder.	46
4.9.	Ensamblaje de elementos en la carcasa trasera del panel de control.	47
4.10.	Ensamblaje de elementos en la carcasa delantera del panel de control.	48
4.11.	Simulación de monitoreo de las variables principales a monitorear.	49
4.12.	Simulación de monitoreo de la temperatura en las resistencias.	50
4.13.	Simulación de monitoreo de las distintas presiones existentes en el proceso de inyección.	51
4.14.	Simulación de monitoreo de las entradas digitales en el proceso de inyección.	52
4.15.	Simulación de monitoreo de las banderas digitales en el proceso de inyección.	53

Capítulo I

Introducción

Una de las consecuencias más importantes del desarrollo tecnológico actual es la automatización, la cuál implica el uso de componentes electrónicos, computadoras, elementos de cómputo reconfigurable y sensores. Todo esto ha permitido mejorar de la calidad del producto final y ha facilitado la labor de los operadores y supervisores dentro de una línea de producción.

La mayor parte de las pequeñas y medianas empresas carecen de los recursos necesarios para la adquisición de maquinaria automatizada debido a los altos costos, esto deja a muchas empresas en desventaja frente a las empresas con mayor poder adquisitivo. Sin embargo, gran parte de la maquinaria con la que se cuenta actualmente puede ser automatizada, permitiendo así el ahorro de tiempo y de recursos.

En este proyecto se propone el desarrollo de un panel de control que controla y monitorea el proceso de una máquina inyección de plástico, el cual comprende la integración mecánica de los elementos de hardware (placa madre, fuente de alimentación, disco duro, pantalla táctil, teclado y accesorios) en una sola estructura mecánica, así como el desarrollo de una Interfaz Gráfica de Usuario que se muestra en la pantalla táctil y la programación de la interfaz de comunicación con el hardware controlador de la máquina.

El panel de control propuesto permite realizar pruebas de distintos eventos de dicha máquina, además de facilitar la interacción entre la máquina y el

usuario con el propósito de que éste sea capaz de manipular con facilidad las variables involucradas en un ciclo de inyección y monitorear el proceso.

Esta tesis se compone de cinco capítulos en los cuales se documenta el desarrollo de este proyecto. En el primer capítulo de este trabajo se abordan de manera breve algunos antecedentes relacionados con los proyectos de automatización desarrollados dentro de la Universidad Autónoma de Querétaro. Asimismo se describe de manera breve la justificación de este proyecto, se plantean los objetivos de ésta tesis y se muestra el planteamiento general, donde se describe de manera breve el desarrollo que se llevó a cabo, sus componentes y la interacción que tienen con los elementos de control existentes en el sujeto experimental (máquina inyectora de plástico).

En el capítulo dos se hace una revisión de la literatura de las herramientas teóricas necesarias para la realización de este proyecto. Se explica el funcionamiento de la máquina de inyección de plástico y los procesos involucrados en un ciclo de inyección. En el capítulo tres se muestran las actividades experimentales realizadas con el fin de integrar y desarrollar los módulos que componen la totalidad de este proyecto.

En el capítulo cuatro se muestran los resultados obtenidos mediante pruebas simuladas de la máquina de inyección. En el capítulo cinco se muestran las conclusiones de este trabajo y los posibles desarrollos a futuro sobre este proyecto. Por último se presentan distintos anexos con información adicional y las referencias consultadas.

1.1. Antecedentes

Si bien el ser humano ha realizado logros sorprendentes en cuanto a la producción de los bienes de consumo, no fue sino hasta el siglo XVIII con la llegada de revolución industrial, cuando comenzaron a ser introducidas las bases para la producción a gran escala. Gracias a la invención de la máquina de vapor por *James Watt (1776)*, fueron desarrolladas grandes máquinas que reemplazaron al trabajador como el elemento principal de la línea de producción.

Frederick Winslow Taylor propuso que se debía determinar el mejor método para elaborar productos en base a la observación, medición y análisis. *Henry Ford* aplicó este principio a la producción del modelo T en 1913 y observó que el tiempo requerido para ensamblar un automóvil podía reducirse significativamente.

A medida que se incrementó la capacidad de la maquinaria industrial, el ser humano comenzó a fabricar máquinas que podían realizar más de una tarea de manera secuencial. La automatización de procesos industriales fue apoyada paralelamente con la invención y evolución de las computadoras.

La invención de la primera computadora electrónica se atribuye al profesor de física John Atanasoff, con la finalidad de acelerar el cálculo de ecuaciones complejas. Durante los primeros años de desarrollo las computadoras se caracterizaron por un alto consumo de potencia, baja capacidad de procesamiento de datos y tener un volumen excesivo. Debido a los avances de la electrónica, las computadoras han reducido su consumo de energía y han aumentado su capacidad de almacenamiento y procesamiento de datos, permitiendo su uso para diversas aplicaciones en el hogar y en la industria.

El teclado fue inventado en 1860 por el inventor Christopher L. Sholes, permitiendo la introducción de caracteres en una máquina de escribir. Posteriormente fueron implementados en las computadoras con la distribución de teclas habitual en las máquinas de escribir.

1.1. Antecedentes

En los años 70, la compañía norteamericana Xerox introdujo el mouse como la forma de comunicación primaria entre usuario y máquina. En 1984 Apple lanzó al mercado un nuevo sistema con interfaz gráfica con un gran éxito comercial. En 1987 Apple introdujo la Macintosh II, la primera computadora con un monitor a color.

De esta manera, fue más sencillo para las industrias monitorear, supervisar y manipular los procesos mediante el uso de una computadora con sus distintos periféricos de entrada y salida.

En la Universidad Autónoma de Querétaro se han realizado diversos proyectos de automatización y control aplicados a distintas ramas de la industria. Un ejemplo de esto es el desarrollado por [Bravo \(2004\)](#) en el cual se diseñó e implementó un sistema de control aplicado a una máquina de inyección de plástico. Dicho proyecto cuenta con un software de control implementado en lenguaje C, cuenta con una interfaz gráfica mediante la cual se pueden monitorear y manipular las variables de control en una máquina inyectora de plástico.

[Enríquez \(2006\)](#) desarrolló un sistema de control de temperatura aplicado a una máquina de inyección de plástico. En este trabajo se identificaron las zonas de calentamiento del cañón de una máquina inyectora, se realizó una interfaz de comunicación diseñada bajo el bus estándar Eurocard generada a partir del puerto paralelo, implementada en *VHDL* y sintetizada en un *FPGA*. Se diseñó un controlador *PID* con un módulo *PWM*.

[Hernández \(2008\)](#) elaboró una tarjeta de interfaz estándar para control en una máquina inyectora de plástico utilizando componentes comercialmente disponibles, además de una tarjeta Spartan 3. Probó distintas rutinas consistentes en la lectura y escritura de datos. La comunicación entre la computadora y la interfaz se efectuó mediante el puerto paralelo. [García \(2008\)](#) implementó un controlador de presión y de velocidad para máquinas de inyección en *FPGA*.

1.1. Antecedentes

Chavarín (2011) implementó un módulo de control de posición de husillo para inyección de alto desempeño. Se utilizó un controlador difuso PID genérico (debido a la alta no linealidad del proceso) implementado en *VHDL* y programado en un *FPGA* (Spartan-3). El sistema fue probado con diferentes referencias de posición de temperatura de una zona de calentamiento de la máquina inyectora.

Muñoz (2011) implementó un *PLC* basado en *FPGA*, controladores difusos y una red de microprocesadores para controlar una máquina de inyección de plástico. El sistema fue desarrollado de manera modular con control de eventos continuos y discretos y un módulo de comunicación. Estos módulos tuvieron un excelente desempeño en pruebas individuales.

En lo relacionado al desarrollo de interfaces gráficas dentro de la universidad Reyes (2008) desarrolló una interfaz gráfica mediante el uso de la programación en Visual C++ que permite realizar el monitoreo y control de una granja acuícola, en el cual se hizo uso de una pantalla táctil. Este sistema es de arquitectura abierta, la cual permite que se incluyan o modifiquen algunos de los componentes existentes del sistema.

Dentro de esta institución se han realizado una gran cantidad de trabajos que utilizan el cómputo reconfigurable para diversas aplicaciones de control y monitoreo. Por ejemplo, Romero (2004) diseñó un sistema de procesamiento de señales para la detección de ruptura a herramientas en sistemas de manufactura por control numérico computarizado. Osornio (2004) diseñó un controlador de 3 ejes para máquinas herramienta basado en *FPGA*. Colín (2006) realizó la descripción de un controlador PID para máquinas herramienta mediante *VHDL*. Franco (2007) desarrolló un sistema de monitoreo para máquinas-herramienta basado en *FPGA*. Morales (2007) realizó la descripción de una unidad USB para control de posición y generación de perfiles en un intercambiador automático de herramientas.

1.1. Antecedentes

Jaen (2011) desarrolló perfiles polinomiales en 3D para control de posición en una máquina fresadora basado en curvas paramétricas *NURBS* y la instrumentación de los 3 ejes de dicha máquina. Morales (2010) realizó una plataforma de arquitectura abierta basada en unidades multi-agente aplicada a máquinas CNC, robots y máquinas de inyección de plástico.

González (2011) realizó la primera etapa de la reconversion de una máquina RX-4A, mediante el desarrollo de un control de movimiento de sus ejes implementado en un *FPGA*, este sistema fue implementado con una arquitectura abierta la cual permite incorporar nuevos módulos. Esta reconversion consistió en seleccionar y elaborar elementos necesarios para el sistema eléctrico de control. Se desarrolló una interfaz de usuario mediante el uso de la biblioteca *gtkmm* y del lenguaje C++, la cual permite monitorizar y controlar el movimiento de cada uno de los ejes de la máquina.

Hasta ahora se han llevado a cabo diversos proyectos de control e instrumentación de procesos en la universidad con excelentes resultados, sin embargo, estos sistemas desarrollados no son viables comercialmente debido a que no cuentan con una interfaz de usuario amigable y sencilla que permita la fácil manipulación, monitoreo y configuración de los parámetros de control.

1.2. Justificación

En base en los antecedentes se observa la necesidad de contar con un desarrollo de software que, en conjunto con el hardware de computadora (tarjeta madre y disco duro) y dispositivos periféricos externos, sea capaz de integrar los elementos de control implementados en la máquina inyectora de plástico en una sola estación de trabajo. El software debe ser amigable para un usuario promedio y contar con facilidad de manipulación de datos en pantalla.

Se observa la necesidad de contar con una estructura mecánica que integre los distintos componentes que se requieren para que el panel de control funcione adecuadamente (disco duro, tarjeta madre, fuente de poder y periféricos) con la finalidad de tener un solo sistema integral de hardware/software.

En base al estudio de los trabajos desarrollados dentro de esta institución, se puede observar que se cuenta con desarrollos de automatización implementados en hardware, pero la mayor parte de estos trabajos no cuentan con un buen respaldo en software que permita que la manipulación y el acceso a datos de manera sencilla y eficaz. Se requiere que el programa y la interfaz de usuario sean desarrollados en una plataforma que permita distribuir el software libremente, ya que de esta manera se puede reducir el costo final del producto además de permitir la reutilización de código en trabajos futuros.

Es necesario contar con una estructura física que permita integrar los componentes físicos del panel, este panel debe responder a las necesidades industriales (robustez física e implementación de tecnología actual disponible comercialmente).

1.3. Objetivos

1.3.1. Objetivo General

Del análisis realizado en la sección anterior, en cuanto a los trabajos elaborados, se plantea la necesidad de contar con un software amigable que permita realizar pruebas simuladas de la máquina de inyección para ejecutar el proceso completo además de ser capaz de modificar y establecer las variables de interés para un usuario, lo cual se ve reflejado en el objetivo general de este trabajo de tesis.

- ✓ Desarrollar un panel de control que muestre la Interfaz Gráfica del Usuario que permita establecer y monitorear las variables en el proceso de inyección de plástico.

1.3.2. Objetivos Particulares

- ✓ Desarrollar un software para comunicación con el hardware de control que permita monitorear las distintas variables existentes en el proceso de inyección de plástico.
- ✓ Integrar una pantalla táctil y un teclado industrial que sirvan como interfaz entre el usuario y la máquina de inyección.
- ✓ Proponer un prototipo para la estructura mecánica del Panel de Control que integre el equipo necesario para su funcionamiento.
- ✓ Desarrollar la Interfaz Gráfica de Usuario que permita establecer y monitorear las distintas variables a controlar en el proceso.
- ✓ Desarrollar la interfaz de comunicación con el controlador implementado en FPGA.

1.4. Planteamiento General

El panel de control debe contar con una interfaz gráfica de usuario (GUI) y un programa de monitoreo y manipulación de variables que facilite la interacción con la máquina, y que el usuario ejecute de manera mas eficaz y rápida un ciclo de inyección o bien una parte de este. En la figura 1.1 se presenta un diagrama de bloques del sistema que se desarrolló en este trabajo y su relación con el resto del sistema existente en la máquina de inyección.

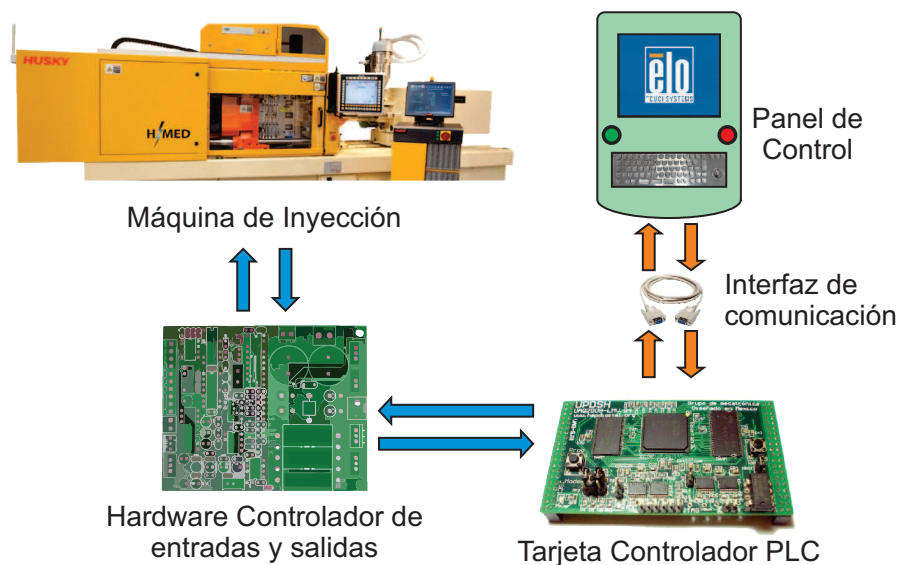


Figura 1.1: Esquema general de la propuesta del planteamiento del proyecto, la máquina inyectora es controlada mediante hardware que coordina entradas y salidas por medio de una tarjeta PLC que se comunica con el panel de control

Se propone la creación de una estructura física que permita contener y soportar la carga mecánica de los componentes necesarios para el funcionamiento del panel de manera óptima.

Se debe desarrollar una interfaz que permita la comunicación con la tarjeta PLC basada en FPGA mediante la lectura y escritura de registros en una memoria RAM integrada en un FPGA; además el programa cargado en el procesador debe ser capaz de interpretar adecuadamente los datos introducidos desde los periféricos de entrada como lo son el teclado industrial y la pantalla táctil;

1.4. Planteamiento General

así como las entradas y salidas del controlador PLC.

El programa debe tener la capacidad de responder adecuadamente a los eventos presentes en el ciclo de inyección, y también a los estímulos externos de un usuario, ya sea la introducción y establecimiento del valor de una variable o bien la opresión de un botón que inicie una secuencia dentro del proceso.

En la figura 1.2 se muestra un diagrama con los componentes físicos del panel de control.

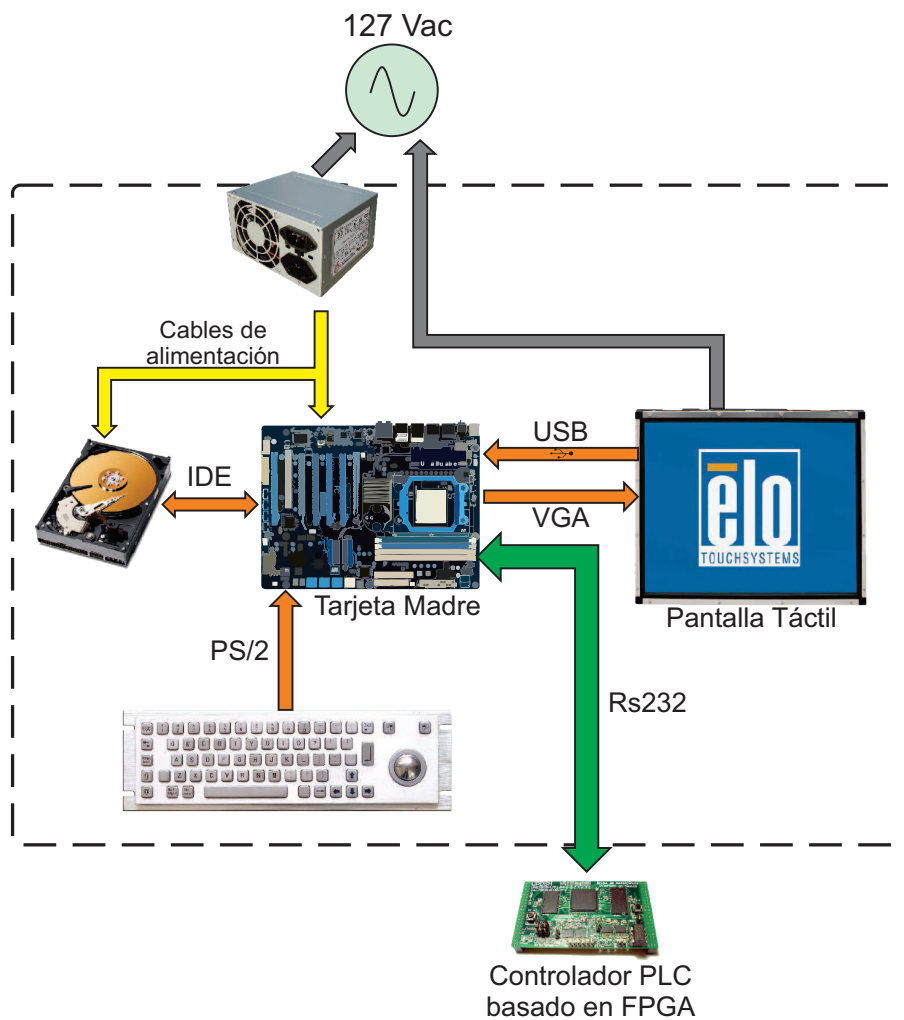


Figura 1.2: Componentes físicos y su interconexión dentro del panel de control propuesto.

Capítulo II

Revisión de la literatura

2.1. Pantalla Táctil

Es un dispositivo que mediante un toque directo sobre su superficie permite la entrada de datos y ordenes al dispositivo. De esta forma, una pantalla táctil puede actuar como periférico de entrada y de salida a la vez. Los beneficios de estos dispositivos son evidentes, en lugar de requerir dispositivos extras como un ratón, el usuario solo necesita su dedo para manipular los datos mostrados en pantalla. La popularidad de las pantallas táctiles ha incrementado debido a la gran cantidad de dispositivos donde pueden ser utilizadas y al desarrollo de interfaces gráficas más amigables, y pueden ser instaladas en los teléfonos celulares, computadoras de escritorio, cajeros automáticos, paneles industriales, etc.



Figura 2.1: Panel industrial con pantalla táctil incluida.

2.2. Teclado

Un teclado de computadora convencional es un dispositivo que funciona como un teclado de máquina de escribir y que utiliza una distribución de las teclas, las cuáles actúan como interruptores electrónicos mediante los cuales se envía información hacia la computadora.

Los teclados actuales constan de aproximadamente 101 teclas. A continuación se describen los tipos de teclas que se pueden encontrar comúnmente en el teclado.

- ✓ *Teclas de caracter* Básicamente, son las letras del abecedario, varían según la region del mundo en el que es utilizado. Encima de estas se encuentran los números del 0 al 9
- ✓ *Teclas de función* Están localizadas en la parte superior del teclado. Frecuentemente son usadas como atajos y en ocasiones son utilizadas en combinación con otras teclas como Ctrl, Alt o Shift.
- ✓ *Tecla Enter o Return* Esta tecla es utilizada para introducir comandos, mover el cursor a una línea nueva o escoger entre una serie de opciones.
- ✓ *Tecla Escape* Es usada para salir de un programa o de una pantalla previa. Generalmente esta tecla es utilizada como comando de salida.
- ✓ *Tecla ALT* Es usada en combinación con otras teclas, como teclas de caracter o teclas de función en diversos programas.
- ✓ *Tecla Control* También es usada en combinación con otras teclas. Por ejemplo, en editores de texto puede ser usada en combinación con teclas de caracter para cortar, copiar o pegar cadenas de texto.
- ✓ *Tecla Caps Lock* Cuando esta tecla esta activada, cada letra que es tipeada sera mayúscula.

2.2. Teclado

- ✓ *Pad Numérico* Es el conjunto de teclas que contiene los numeros del 0 al 9, además de tener como segunda función diversas teclas como avance y retroceso de página.



Figura 2.2: Distribución común de un teclado de computadora.

2.2.1. Controlador de Teclado

El teclado, que es conectado a la tarjeta madre a través de un cable, tiene un micro controlador interno. En la tarjeta madre existe un microcontrolador similar o un emulador de éste, y se comunican mediante un protocolo de comunicación serial bidireccional.

2.2.2. Escaneo de código

El microcontrolador dentro del teclado se encarga de detectar cuando una tecla es presionada y la identifica. Posteriormente el microcontrolador envía a la tarjeta madre un código exclusivo para cada tecla. Al presionar y soltar una tecla se generan distintos códigos, según sea la tecla oprimida y liberada. Cabe señalar que este código no tiene nada que ver con el código ASCII.

2.3. Protocolos de comunicación

2.3.1. RS232

Es una estándar de comunicación que permite el intercambio de forma serial de un conjunto de datos entre un dispositivo y una computadora. Los niveles de tensión están definidos entre -3 y -15 V para el 1 lógico y +3 y +15 V para el 0 lógico. Normalmente están establecidos en -12 V y +12 V. Este tipo de interfaz utiliza la lógica negativa.

Adicionalmente algunos parámetros deben ser configurados mediante software como lo son: la velocidad de recepción y transmisión en baudios, la verificación de paridad, el número de bits de parada por cada dato y la cantidad de bits por dato.

La manera mas común de enviar o recibir datos es mediante una trama de 11 bits, los cuales constan de un bit de inicio para indicar que la transmisión ha comenzado, 8 bits de datos, un bit de parada que indica el final de transmisión de una palabra digital y un bit de paridad con el que indica si la transmisión ha finalizado correctamente.

Señales del RS232

A continuación se describen las señales mas importantes del protocolo de comunicación RS232:

- ✓ *Request to Send (RTS)* Esta señal se envía de la computadora al dispositivo para indicar que se desean transmitir datos. Una vez que esta señal se pone en alto la computadora espera una respuesta por la línea CTS.
- ✓ *Clear to Send (CTS)* Esta señal se genera a partir del dispositivo conectado a la computadora y le indica a la computadora que la transmisión de datos puede comenzar.

2.3. Protocolos de comunicación

- ✓ *Data Terminal Ready (DTR)* Esta señal es generada a partir de la computadora e informa al dispositivo que esta lista para recibir datos.
- ✓ *Data Set Ready (DSR)* Esta línea es generada por el dispositivo en respuesta a una señal DTR de la computadora. La computadora checa el estado de esta línea después de poner en alto DTR para detectar si el dispositivo esta activo.
- ✓ *Transmit Data* Esta línea es usada para mandar la información hacia el dispositivo, los datos se transmiten de un bit a la vez.
- ✓ *Receive Data* Es la línea por donde el dato es recibido de un bit a la vez.

2.3.2. PS/2

El protocolo PS/2 es una comunicación serial bidireccional basada en 4 líneas (datos, reloj, tierra, alimentación de voltaje). El conector PS/2 esta formado por 6 pines que funcionan con voltajes TTL de las cuales 4 son utilizadas por las líneas mencionadas anteriormente. Los pines sobrantes no se utilizan.

El protocolo PS/2 transmite datos en cadenas de 11 bits sobre la línea de datos bidireccional. Cuando no se realiza transmisión alguna de datos la línea de reloj y la de datos permanecen en nivel alto (están inactivas).

Los bits que componen la trama de datos son los siguientes:

- ✓ 1 bit de inicio, siempre en 0.
- ✓ 8 bits de datos, el bit menos significativo primero.
- ✓ 1 bit opcional para indicar la paridad.
- ✓ 1 bit de parada, siempre en 1.

Si un dispositivo desea comunicarse con la computadora, primero el dispositivo verifica que ambas señales bidireccionales estén inactivas. Posteriormente

2.3. Protocolos de comunicación

el dispositivo fija los 10 bits restantes en la línea de datos, los cuales son validados en el flanco de subida del reloj. La transmisión finaliza cuando se fijan las señales de reloj y de datos a uno.

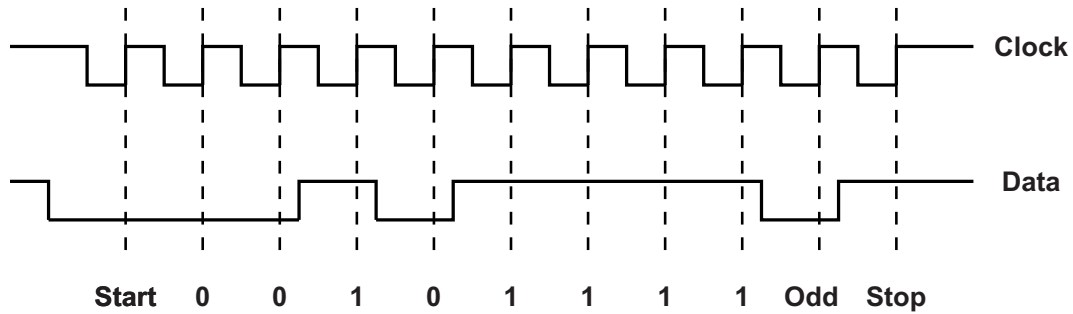


Figura 2.3: Formas de onda de las señales de datos y de reloj de la interfaz USB.

Si la computadora desea comunicarse con un dispositivo, primero la computadora fija la señal de reloj en 0 para evitar que el dispositivo mande algún dato. Posteriormente se fija la señal de datos en cero y la señal de reloj se levanta para indicar que se tienen datos para enviar al dispositivo. El dispositivo genera los flancos de reloj necesarios para capturar los 10 bits restantes. Después de fijar el bit de stop a uno se libera la señal de datos y el dispositivo deja de generar pulsos de reloj.

2.3.3. USB

El *USB* (Bus Universal en Serie) es un protocolo desarrollado a mediados de los años noventa por INTEL. Utiliza un conector de 4 pines, la transferencia de datos se realiza mediante 2 líneas de voltaje diferencial (D+ y D- respectivamente) el cual debe ser por lo menos de 1 V. También cuenta con una línea de tierra y una de alimentación ($V_{cc}=5$ V). No se requiere de una señal de reloj ya que es generada a partir del flujo de datos.

El conector USB permite el *Hot Plugging*, es decir, conectar o desconectar un periférico mientras la computadora esta encendida. Cada unidad USB recibe una dirección propia al ser conectada, con un máximo valor de 127, asignada por

2.3. Protocolos de comunicación

el host USB.

El control de la transferencia de datos parte siempre del host, ya que es capaz de crear interrupciones, de esta manera no es posible que ningún dispositivo USB inicie ninguna transferencia por su cuenta sin permiso del host.

De la identificación de los dispositivos se encargan un identificador del fabricante (Vendor ID, 2 Bytes), un identificador del dispositivo (Device ID, 2 Bytes), un número de versión (2 Bytes) y un código de clase de 3 Bytes. Durante la inicialización, el host que habitualmente se encuentra en el juego de chips de la tarjeta madre, comprueba el tipo de dato que se soporta y el de banda requerido para cada dispositivo.

Para solicitar una transferencia de datos se utilizan 3 paquetes diferentes: *Token*, *Data* y *Handshake*. Mediante el Packet ID es posible identificarlos y distinguirlos como se muestra en la figura 2.4.

Token:

PID								8 Bit	3 bit	5 bit
0	1	x	x	1	0	x	x	ADDR	ENDP	CRC5

Data:

PID								0-n bit	5 bit
0	0	x	x	1	1	x	x	ADDR	CRC16

Handshake:

PID							
1	0	x	x	0	1	x	x

Figura 2.4: Variantes de paquetes en la trama de comunicación de periférico USB a host.

El host divide el ancho de banda disponible en 3 diferentes canales de datos, que se denominan Pipes, los cuales son canales virtuales punto a punto entre el host y un dispositivo USB. Los datos se transfieren en una secuencia de Bytes sin estructura (*Stream Pipe*) con tamaño máximo de 1500 Bytes. Los mensajes que informan al host si hay datos disponibles se transfieren en el paquete

2.4. Interfaz Gráfica de Usuario

Token, poseen una estructura definida y se procesan como Message Pipes.

2.4. Interfaz Gráfica de Usuario

Una interfaz de usuario es el conjunto de técnicas, mecanismos y objetos que permiten interactuar con una computadora de forma gráfica. Los elementos con los que interactuar el usuario son los objetos, los cuales son entidades independientes mediante las cuales un usuario puede realizar distintas tareas y operaciones dentro de un programa o proceso.

Una Interfaz Gráfica de Usuario (GUI) engloba al conjunto de objetos como imágenes, menus, ventanas y símbolos que permiten que la comunicación maquina-hombre sea mucho mas sencilla y se lleve a cabo de manera mas intuitiva. Para moverse a través de una Interfaz Gráfica es necesario contar con un dispositivo como un teclado, ratón, pantalla táctil, lápiz óptico o algún otro dispositivo periférico.

Las interfaces gráficas se basan en la manipulación de ventanas, las cuales son areas de trabajo que contiene un programa o datos. Los elementos de una interfaz gráfica están organizados en barras, contenedores, panales, menus contextuales, ventanas, etc.

Los menus muestran una lista de opciones que el usuario puede ejecutar, permitiendo facilitar el uso del programa al no tener que recordar los comandos básicos para su funcionamiento. Obviamente no es posible englobar todos los comandos de una aplicación en un solo menu, razón por la cual se crearon los menus secundarios y los cuadros de dialogo. Pueden ser agrupados visualmente de diversas maneras, según sea mas conveniente para el usuario final. Estos objetos generados están organizados de manera jerárquica dentro del programa y pueden tener acceso restringido o total a ciertas variables que maneja el programa.

2.5. Bibliotecas GTKMM

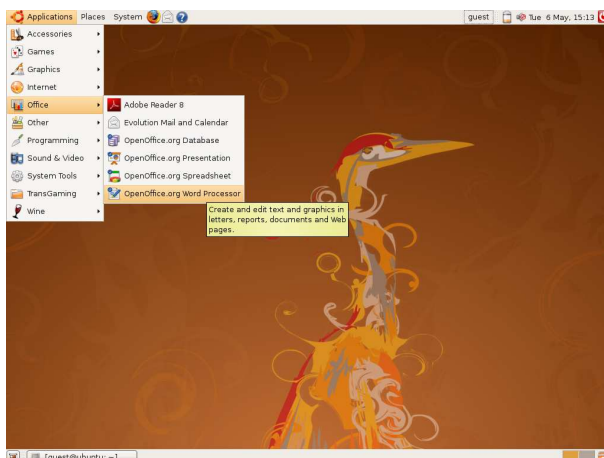


Figura 2.5: Interfaz Gráfica de Usuario del Sistema Operativo Ubuntu.

2.5. Bibliotecas GTKMM

Es una librería de estructuras y funciones escritas en C para facilitar el desarrollo de aplicaciones de interfaz de usuario. GTK provee las herramientas necesarias para crear ventanas, botones, etiquetas, texto, etc. Es de código libre y es distribuida bajo la licencia GNU LGPL.

La librería GTK esta orientada a objetos y maneja una jerarquía lógica para el manejo de los widgets. Algunos de los widgets incluidos en GTK son menus, barras de herramientas, árboles, barras de progreso, barras de estado, distintos tipos de botones, entrada de datos, calendarios entre otros.

GTK provee de un medio para interactuar con los objetos visibles que son creados, haciendo posible su activación mediante la acción de un usuario. Posee soporte para distintos sistemas operativos además de que puede ser utilizada junto con una gran cantidad de lenguajes de programación.

2.6. Dispositivos reconfigurables

2.6.1. FPGA

Un *FPGA* es un dispositivo lógico que contiene un arreglo de compuertas lógicas genéricas e interruptores programables. Una celda lógica puede ser configurada para que realice una función simple y un switch programable puede ser configurado para proveer las conexiones entre las celdas lógicas.

Las celdas básicas de un FPGA son mucho mas simples que las macroceldas de un PLD. Cada fabricante y cada familia de FPGA ofrece una version diferente de la celda básica.

Todo FPGA tiene 3 grupos de celdas básicas que son:

1. Celda Lógica
2. Celda de entrada/salida
3. Celda de distribución

La celda lógica es un pequeño bloque lógico de compuertas programables. Las celdas de entrada/salida están destinadas a proporcionar una conexión entre el FPGA y el exterior. Finalmente las celdas de distribución se encargan de manejar las señales de reloj hacia dentro del circuito.

La interconectividad en los FPGA esta restringida a celdas adyacentes exclusivamente. Con esto se permite un menor consumo de potencia y un mayor nivel de integración. Actualmente se puede contar con unidades que van desde las 50,000 hasta los 10,000,000 de compuertas lógicas equivalentes.

Poseen un gran número de ventajas en comparación con los *ASICs* (Circuitos Integrados para Aplicaciones Específicas), por ejemplo, puede tomar meses fabricar un ASIC con un costo de cientos de millones de dólares para obtener el

2.6. Dispositivos reconfigurables

primer dispositivo; en cambio los FPGAs son configurados en menos de un segundo (y pueden ser reconfigurados en caso de que se cometa un error) y su costo es de solo algunos cientos de dólares.

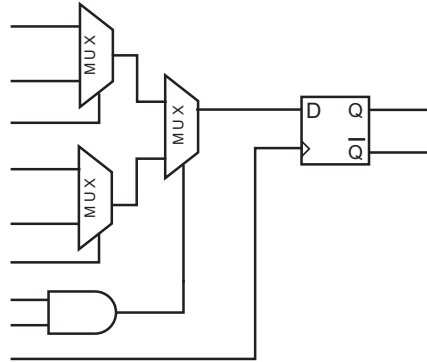


Figura 2.6: Celda lógica en un FPGA.

Los FPGAs representan una alternativa poderosa para implementación de sistemas digitales, debido a su rápida programación y bajo costo.

2.6.2. PLC

Es un dispositivo basado en microprocesadores que usa una memoria programable para almacenar instrucciones e implementar operaciones lógicas, aritméticas, secuenciales, de temporización y de conteo. Están diseñados para ser operados por personal con un conocimiento limitado de computadoras y lenguajes de programación.

El programa de control implementado en un PLC puede ser introducido usando una forma sencilla de lenguaje de programación. Un operador puede introducir en el PLC una secuencia de instrucciones o un programa en la memoria del PLC. El controlador entonces monitorea las entradas y salidas de acuerdo a la programación realizada y lleva a cabo las tareas de control.

Diversos dispositivos de entrada pueden ser conectados al PLC, por ejemplo, los sensores y los dispositivos de salida del sistema a controlar como pueden ser motores, válvulas, etc.

2.6. Dispositivos reconfigurables

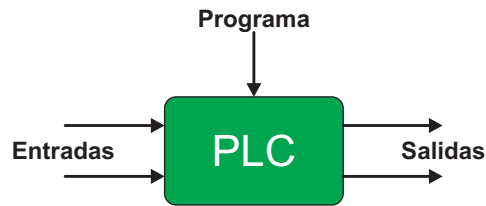


Figura 2.7: Diagrama general de un PLC.

Los PLCs tienen la gran ventaja de poder usar el mismo controlado básico para una gran variedad de sistemas. Si se desea modificar el sistema de control solo es necesario implementar un nuevo conjunto de instrucciones en el PLC. Los PLCs tienen las siguientes características:

- ✓ Son robustos y pueden soportar los efectos de las vibraciones mecánicas, de la temperatura, humedad y ruido.
- ✓ Contienen las interfaces para entradas y salidas dentro del controlador.
- ✓ Son fácilmente programables y el lenguaje característico de estos dispositivos es sencillo de entender.

Comúnmente los PLCs contienen componentes básicos funcionales como la unidad de procesamiento, la memoria, alimentación de voltaje, sección de interfaces de entrada y salida, una interfaz de comunicación el dispositivo programable. A continuación se describen estos elementos:

- ✓ *CPU*: es la unidad que contiene el microprocesador y se encarga de interpretar las señales de entrada y realiza las acciones de control, acordes al programa almacenado en memoria, comunicando las decisiones tomadas a las salidas.
- ✓ *La unidad de alimentación de voltaje*: es requerida para convertir el voltaje de CA en un nivel bajo de voltaje de CD (5 V) necesario para el procesador y los circuitos de interfaz de entradas y salidas.

2.7. La máquina de inyección de plástico

- ✓ *Dispositivo programable*: Es usado para introducir el programa en la memoria del procesador. El programa es implementado en el dispositivo y transferido a la unidad de memoria del PLC.
- ✓ *Unidad de memoria*: Aquí es almacenado el programa que llevara a cabo las acciones de control que deben ser ejecutadas por el microprocesador y los datos almacenados provenientes de las entradas para procesamiento y las salidas para canalizarlas hacia las líneas correspondientes.
- ✓ *Sección de entradas y salidas*: Es donde el procesador recibe la información de los dispositivos externos y envía información a los dispositivos externos. Los dispositivos de entrada y salida pueden ser clasificados mediante la señal que entregan al PLC, es decir, de señales discretas, digitales o analógicas.
- ✓ *Interfaz de comunicación*: Es usada para recibir y transmitir datos en redes de comunicación desde otros PLCs. Se encarga de tareas como la verificación de dispositivos, adquisición de datos, sincronización entre aplicaciones del usuario y administración de conexiones.

2.7. La máquina de inyección de plástico

La máquina de inyección de plástico esta compuesta por la unidad de inyección, que se encarga de inyectar el material, y la unidad de cierre que es usada para mantener el molde cerrado durante la fase de inyección.

Estas dos unidades se encargan de llevar a cabo 2 procesos primarios independientes que se complementan el uno con el otro.

2.7.1. La unidad de inyección

La unidad de inyección debe llevar a cabo varias tareas y contiene muchos componentes que contribuyen a que estas tareas se lleven a cabo.

2.7. La máquina de inyección de plástico

Cilindro de calefacción

Es la parte central de la unidad de inyección, usualmente esta hecho de acero barato y esta manufacturado en forma de un tubo largo y redondo. Su interior esta revestido de una capa delgada de acero duro de alta calidad.

El exterior del cilindro tiene bandas calefactoras activadas ecléticamente y están colocadas a lo largo de toda la extension del cilindro con una separación mínima. Existen 3 zonas de calentamiento en el cilindro, cada zona contiene 3 o mas bandas (dependiendo de la longitud del cilindro) y cada zona esta controlada individualmente por una unidad eléctrica localizada en el panel de control de la máquina.

Cada unidad de control de temperatura contiene un termopar montado en la pared del cilindro calefactor de la zona que se esta controlando. De esta manera, la unidad de control puede determinar si se requiere mas energía en las bandas calefactoras. Cuando la temperatura es alcanzada, la unidad de control desactiva el suministro de energía a las bandas calefactoras hasta que la temperatura descienda de nuevo.

Las límites de temperatura son establecidos en la unidad de control y son usados para determinar en que momento las bandas de calefacción son activadas o desactivadas.

Tolva

Aquí es donde los pellets de plástico son almacenados antes de ser introducidos al cilindro de calefacción. Esta unidad tiene lados cónicos para facilitar el vaciado de material hacia el cilindro. La base de la tolva debe contener un imán que tiene como propósito retener partículas sueltas de metal en el plástico.

2.7. La máquina de inyección de plástico

Husillo

Es un tornillo que está colocado dentro del cilindro de calefacción. Su función principal es llevar el material desde la zona de la tolva hacia el área de calefacción del cilindro. Este tornillo también genera calor debido a la fricción lo cual incrementa la temperatura del plástico.

Anillo de seguridad

La punta del tornillo está insertada en un anillo de retención y de seguridad diseñado para retener el material derretido y evitar que vuelva hacia el tornillo sin fin. La boquilla se encuentra en una cara del tornillo, usualmente de rosca izquierda para contrarrestar el movimiento giratorio del tornillo. Las máquinas de inyección comunes usan tornillos reciprocantes. Esto significa que el tornillo avanza y se retrae, actuando como un émbolo que inyecta el plástico derretido.

Valvula de retención y valvula de bola

El propósito de la valvula de retención es evitar que el plástico derretido vuelva hacia el tornillo mientras este se mueve hacia adelante para inyectar el material dentro del molde. El anillo de seguridad puede moverse hacia adelante mientras el tornillo extrae material para el siguiente ciclo. Esta acción del anillo de seguridad permite al material trasladarse hacia el frente de la punta del tornillo.

La boquilla

La boquilla de la máquina consiste en 2 piezas en forma de tubo que se atornilla en una cara del cilindro de inyección. Tiene un agujero cónico que atraviesa la punta de la boquilla. El radio de la punta de la boquilla emboya en el cojinete del molde de inyección. Notese que también hay una banda calefactora

2.8. El proceso de inyección de plástico

en la punta de la boquilla, la cual es controlada al igual que las otras bandas calefactoras en el cilindro de inyección.

2.8. El proceso de inyección de plástico

Es importante conocer el proceso de inyección de plástico, ya que este es el proceso que se desea controlar y monitorear mediante el desarrollo del panel.

El proceso de inyección requiere de la fundición de plástico en una extrusora y la utilización de un tornillo sin fin para posteriormente inyectar el plástico en un molde donde es enfriado.

Cuando se comienza la fabricación de una pieza por inyección de plástico se plantea maximizar la producción, esto se logra minimizando la duración del ciclo de inyección, que es el tiempo que se requiere para derretir el plástico, inyectarlo en un molde, enfriarlo y obtener una pieza terminada. Mediante la utilización de un molde mas grande que produzca mas de una pieza por ciclo también se puede maximizar la producción.

Otro factor que se desea controlar en el ciclo de inyección es la viscosidad del polímero. Aquí es donde entra un juego un gran numero de variables, pero un proceso con mayor probabilidad de éxito se lograra si se controlan y comprenden las siguientes variables:

1. La temperatura del plástico
2. La presión del plástico cuando se inyecta en el molde
3. La velocidad en que el plástico llena el molde
4. Las condiciones de enfriamiento

Las 4 variables son aplicables para casi todos los procesos de moldeo por inyección, sera diferente en cada caso dependiendo de la aplicación, de la materia prima y las preferencias del fabricante.

2.8. El proceso de inyección de plástico

Ventajas del moldeo por inyección:

- ✓ Se pueden producir formas complejas y piezas con pequeño espesor debido a las grandes presiones que se pueden aplicar.
- ✓ Los tiempos de operación son relativamente cortos y puede fabricarse mas de una pieza por ciclo con lo cual se maximiza la productividad.
- ✓ El proceso esta automatizado y muchas de las máquinas requieren de un solo operador.

Las desventajas del moldeo por inyección son:

- ✓ Muchas piezas de bajo precio no pueden ser producidas
- ✓ Los moldes tienen un elevado costo, así que no se suelen llevar a cabo pequeñas series de producción

Capítulo III

Metodología

En este capítulo se describe la metodología seguida para la realización de este proyecto. Este trabajo se compone de varias etapas, las cuales se muestran en el diagrama de bloques de la figura 3.1.

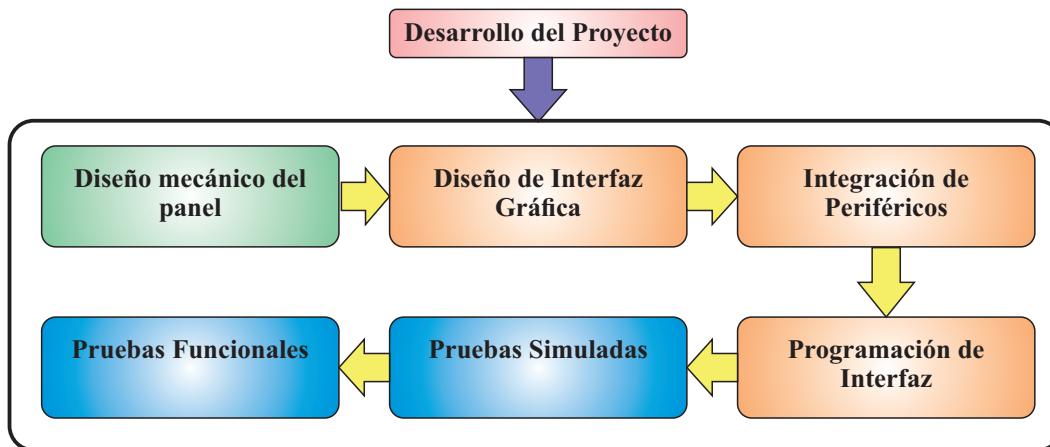


Figura 3.1: Componentes del panel de control propuesto.

La primera etapa consistió en el diseño mecánico de la estructura de soporte y de una carcasa para el panel de control, así como en la distribución de los componentes del panel. Se realizó el dimensionamiento de cada una de las piezas que contendrá el panel y se determinó la mejor distribución de los componentes internos para optimizar el espacio ocupado por el panel.

La segunda etapa del proyecto consistió en el diseño de la interfaz gráfica de usuario para realizar las pruebas de distintos eventos en la máquina de inyec-

3.1. Diseño mecánico del panel

ción. La siguiente etapa consistió en la integración de los dispositivos periféricos utilizados en este proyecto a la interfaz gráfica de usuario.

Posteriormente se realizó la programación de la interfaz gráfica que consistió en el desarrollo y la implementación del programa que realiza la manipulación de las variables de control, la interpretación de los eventos ocurridos en la interfaz gráfica y la comunicación con los controladores de la máquina.

Finalmente se determinaron los casos de estudio, en los cuales se definieron las pruebas en simulación a realizar de la máquina de inyección mediante la asistencia de la interfaz gráfica y los periféricos de entrada/salida. Las pruebas consistieron en la simulación de los eventos continuos y discretos de un ciclo de inyección de plástico.

3.1. Diseño mecánico del panel

En base a los componentes de hardware necesarios para el funcionamiento del panel se propuso un prototipo de diseño realizado en CAD. La propuesta de diseño consta del dibujo de 2 moldes necesarios para la posterior fabricación de las carcasas delantera y trasera.

Cabe señalar que la propuesta fue realizada en base a las dimensiones previamente obtenidas de los elementos presentes en el panel de control, a la funcionalidad y a la disposición de los mismos dentro de una carcasa frontal y una trasera. En la figura 3.3 se muestra un diagrama de la disposición física de los elementos en la carcasa del panel de control.

3.1. Diseño mecánico del panel

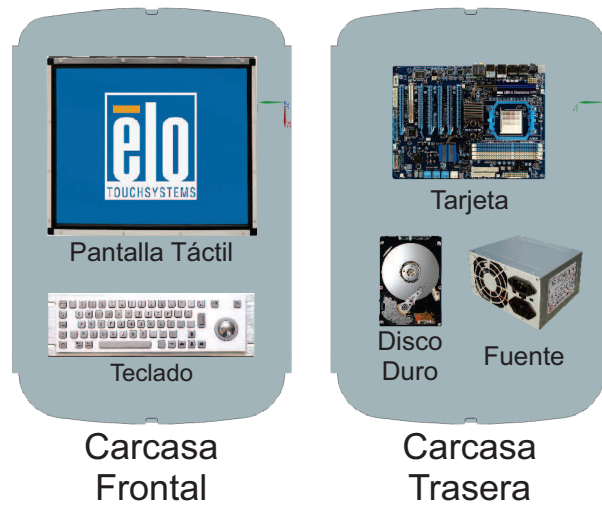


Figura 3.2: Diagrama de la disposición de los elementos presentes en la carcasa del panel de control.

En base a las necesidades descritas en el capítulo anterior fueron propuestos los siguientes componentes de hardware que serán parte del panel de control:

1. Pantalla táctil.
2. Teclado industrial.
3. Tarjeta Madre.
4. Disco Duro.
5. Fuente de alimentación.
6. Cables de datos.
7. Accesorios de montaje y tornillería.

3.2. Diseño de interfaz gráfica

Debido a la necesidad de monitoreo y de manipulación de las distintas variables presentes en los procesos de la máquina inyectora de plástico, fue necesario el diseño de una interfaz gráfica que muestre las variables más significativas presentes en el ciclo de inyección, además de los elementos gráficos necesarios que permitan una correcta y fácil interacción entre un usuario común y la máquina inyectora, por ejemplo una simulación gráfica del proceso de inyección, botones, etiquetas, gráficos entre otros.

La interfaz será dividida en distintas secciones que permitan al usuario una interacción mas sencilla con el proceso de supervisión de la máquina, ya que no es posible mostrar todas las variables importantes en el proceso en una sola ventana gráfica de manera clara y ordenada. El criterio para esta división será el agrupamiento de variables similares en distintas secciones de la interfaz gráfica. En la siguiente figura se muestra un diagrama de las secciones propuestas para la interfaz gráfica del panel de control.

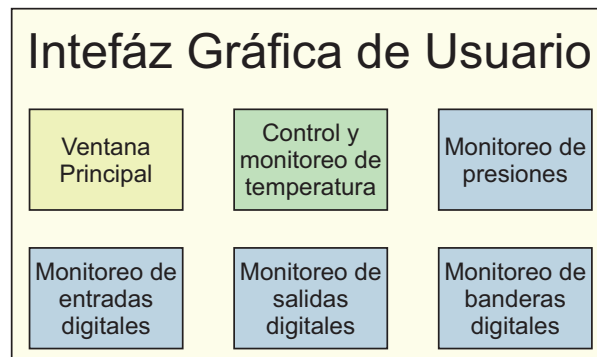


Figura 3.3: Secciones propuestas para la interfaz gráfica de usuario.

Mediante la correcta selección de los elementos visuales en el desarrollo de la interfaz, un usuario común puede comprender, manipular y observar las variables presentes en el proceso de inyección como la temperatura de las resistencias, la presión en las distintas partes del ciclo, las posiciones del molde y de la

3.3. Programación de software

bancada, etc.

Ya que la temperatura de las resistencias es una de las variables mas importantes a supervisar y controlar, la interfaz se diseñó con una sección dedicada únicamente al establecimiento y monitoreo de las variables del control de temperatura implementado actualmente en la máquina de inyección. En la siguiente figura se muestra un diagrama de las variables que son presentadas en la interfaz gráfica de usuario.

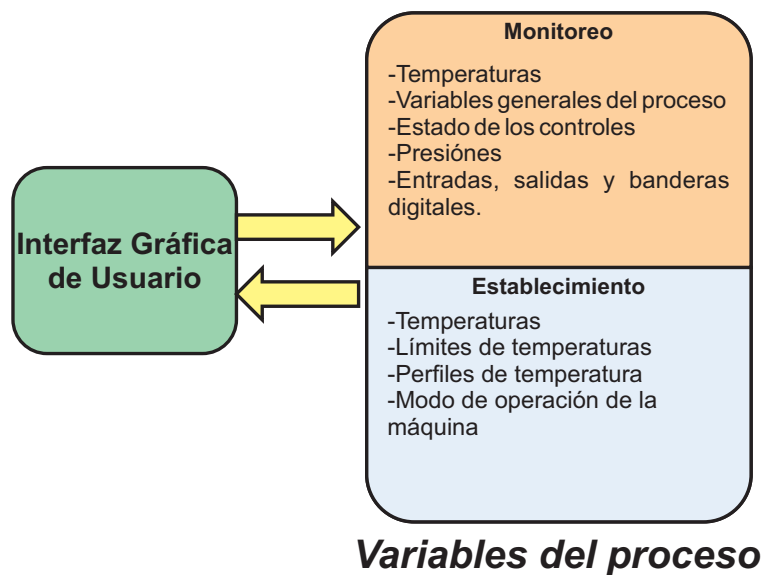


Figura 3.4: Diagrama general de las variables presentes en el proceso de inyección y su relación con la interfaz gráfica de usuario.

3.3. Programación de software

Una vez realizado el diseño de la estructura gráfica de la interfaz gráfica de usuario es necesario programar la estructura interna del software, debido a que el panel debe recibir y enviar datos constantemente hacia la máquina de inyección. Por lo tanto se hace necesario el interpretar los datos recibidos de una manera adecuada ya que de esta manera podrán ser visualizados o establecidos para un mejor control del proceso. El lenguaje utilizado para la implementación de esta

3.4. Fabricación de la estructura mecánica del panel

etapa es C++ debido a que es un lenguaje orientado a objetos y que permite la integración de bibliotecas gráficas adicionales al lenguaje estándar.

El código del programa es accionado en distintos módulos que permiten un mejor control y depuración del código necesario para la ejecución del software, éstos módulos están comprendidos por archivos de cabecera y archivos de implementación, dentro de los cuales serán propuestos las variables, métodos, instancias, objetos y clases necesarias para poder manipular correctamente los datos recibidos y enviados durante el funcionamiento de la máquina.

Estos módulos contienen los métodos y variables más adecuados para la interpretación y despliegue visual de los datos presentes en el proceso, además de interpretar correctamente los eventos de teclado y de la pantalla táctil. Ya que el software es una interfaz con elementos gráficos se hará necesario el uso de la biblioteca gtkmm, la cual fue seleccionada debido a la gran cantidad de recursos gráficos con los que cuenta, además de ser una biblioteca libre y que extiende los métodos y las clases existentes en el estándar de C++.

3.4. Fabricación de la estructura mecánica del panel

Una vez propuestos los elementos de hardware para el panel, la disposición de éstos y la forma general de la estructura mecánica se procede a la fabricación de las piezas mecánicas necesarias para el soporte de éstos elementos. A continuación se propone la fabricación de 2 moldes que servirán para la fabricación de las 2 piezas que componen la carcasa del panel.

El material que se propone para la fabricación de éstos moldes es espuma de poliuretano debido a que es fácil de moldear y de manipular desde el punto de vista de la manufactura. Se realizaron 2 bloques de espuma de poliuretano que posteriormente fueron formados mediante el uso de una máquina-herramienta de

3.5. Integración de dispositivos periféricos y ensamble

Control Numérico por Computadora. Una vez concluida esta etapa los moldes fueron cubiertos con fibra de vidrio y las piezas resultantes fueron desmontadas de los moldes.

Una vez concluida esta etapa de la fabricación de la carcasa mediante el uso de moldes de espuma se realizaron las modificaciones o detallamiento necesarios a las carcasas de fibra de vidrio que pueden ser realizadas con alguna máquina-herramienta de CNC, además de la fabricación de piezas adicionales que serán necesarias para el soporte de los elementos dentro de la estructura mecánica del panel de control. Posteriormente se realizó un acabado final sobre ambas piezas de la carcasa del panel.

3.5. Integración de dispositivos periféricos y ensamble

Esta etapa del proceso de fabricación comprende la integración de todos los componentes de hardware en la estructura mecánica propuesta anteriormente. Se seleccionaron los accesorios adicionales que sean necesarios para la correcta sujeción de los componentes o piezas presentes en el panel de control.

Se propone un ajuste final para la integración de ambas carcasas en una sola estructura mecánica, y que de esta forma la estructura pueda ser montada o desmontada de ser necesario para realizar alguna modificación sobre la estructura del panel o alguna operación de mantenimiento, o la integración de nuevos componentes en el panel.

3.5. Integración de dispositivos periféricos y ensamble

Periféricos a Integrar

En la presente sección se muestran los periféricos a integrar en el panel de control así como sus especificaciones técnicas.

Pantalla Táctil

La pantalla táctil marca EloTouch modelo 1739L que se integró en el panel de control tiene las características siguientes:

- ✓ 17 pulgadas de dimension diagonal
- ✓ Pantalla de tipo matriz activa TFT LCD
- ✓ Área de pantalla útil de 338 mm por 270 mm
- ✓ Resolución nativa de 1280 x 1024 pixeles.
- ✓ 16.7 millones de colores
- ✓ Conector mini D-sub de 15 pines VGA.
- ✓ Voltaje de alimentación de +12 V.
- ✓ Interfaz de retroalimentación USB o serial.
- ✓ 4 kg. de peso.

Teclado Industrial

Se integró un teclado industrial de la marca Logicbus, modelo LBKB35205, el cual cuenta con las siguientes características:

- ✓ Interfaces PS/2 y USB.
- ✓ Voltaje de alimentación de 5 V.
- ✓ Dimensiones de 330 mm por 125 mm.

3.5. Integración de dispositivos periféricos y ensamble

- ✓ A prueba de polvo y de agua.
- ✓ 86 teclas de acero inoxidable de alta calidad
- ✓ Trackball de 25 mm de diámetro
- ✓ Rack de montaje de aleación de aluminio.
- ✓ 2.1 kg. de peso.

Fuente de alimentación

La fuente de alimentación que fue introducida en el panel cuenta con las siguientes características:

- ✓ Alimentación de entrada de 127 V de CA. Salidas de voltaje de +12 V, -12 V, +5 V, y -3.3 V.
- ✓ Potencia de 250 W.
- ✓ 20 pines de alimentación para tarjeta madre.

Disco Duro

En la presente sección se muestran las características principales del disco duro utilizado en el panel de control.

- ✓ Marca IBM, modelo IC35L060AVV207-0.
- ✓ 3.5 pulgadas de largo.
- ✓ Interfaz serial ATA.
- ✓ 4500 rpm.
- ✓ Voltajes de alimentación de +12 V y +5 V.

3.5. Integración de dispositivos periféricos y ensamble

Tarjeta Madre

A continuación se presentan las principales características eléctricas y mecánicas de la tarjeta madre.

- ✓ Modelo VT-8361, marca VIA Technologies
- ✓ Tarjeta madre con factor de forma tipo mini-ATX.
- ✓ 20 pines para entrada de alimentación.
- ✓ Procesador AMD de 1 MHz de la familia x86.
- ✓ Memoria RAM de 1.5 GB.

3.6. Pruebas simuladas

Por último fueron propuestas una serie de pruebas de simulación, que permiten verificar el correcto funcionamiento de la interfaz de usuario y la interfaz de comunicación y del software programado anteriormente en C++ para su posterior implementación final y puesta a punto en el proceso de inyección de plástico.

Este tipo de pruebas consistieron en la generación de rutinas internas del software e integrando el hardware propuesto anteriormente que permitan verificar la interpretación de eventos y la manipulación de datos que realiza el software programado, además de la integración de estos componentes con la interfaz gráfica.

Una vez verificada la estabilidad y la funcionalidad del programa se procederá a realizar pruebas de simulación de la máquina de inyección de plástico con la tarjeta PLC y el hardware de comunicación, para que de esta manera se pueda verificar el comportamiento y la respuesta de la máquina de inyección de acuerdo a los eventos de hardware y la modificación de los distintos parámetros existentes en la interfaz gráfica de usuario.

Capítulo IV

Resultados

En el presente capítulo se muestran los resultados de las distintas etapas que se propusieron dentro del presente proyecto. Como se mencionó en el capítulo anterior estas pruebas se propusieron para validar la estabilidad mecánica del panel, la estabilidad del software y de la interfaz de comunicación con la máquina y para realizar la implementación final en la máquina inyectora.

4.1. Diseño mecánico del panel

En base a las necesidades señaladas en los objetivos del presente trabajo fue necesaria la realización del diseño de la estructura del panel de control mediante software CAD. En la figura 4.1 se muestra la propuesta de diseño final de los moldes para ambas partes de la carcasa del panel de control.

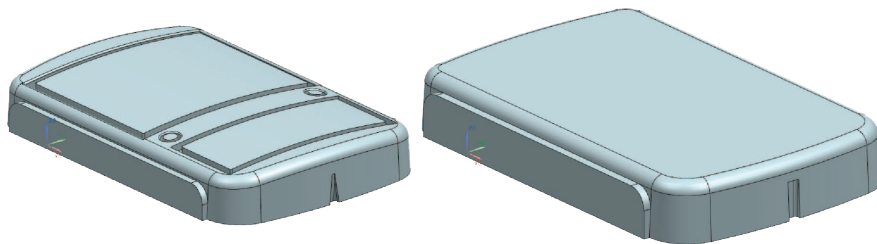


Figura 4.1: Vista isométrica del molde en CAD de la carcasa frontal y trasera del Panel de Control.

4.2. Diseño de la interfaz gráfica

Una vez establecidas las variables a monitorear y controlar dentro del proceso se diseñó la estructura de la interfaz gráfica en 6 secciones distintas separadas por pestañas, todo esto contenido en una sola ventana. Estas secciones fueron diseñadas en base a los conjuntos existentes de variables presentes en el proceso de inyección.

En la primera sección se despliegan las principales variables presentes en el proceso de inyección de plástico además de una representación gráfica de la máquina inyectora de plástico como se muestra en la figura 4.2.

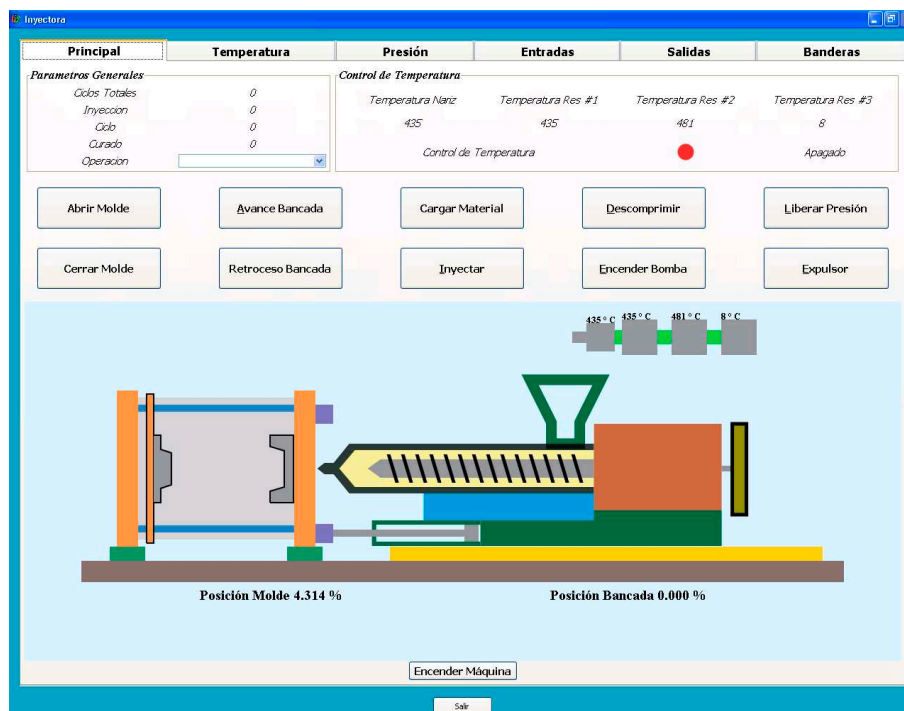


Figura 4.2: Sección principal de la interfaz gráfica de usuario.

4.2. Diseño de la interfaz gráfica

La segunda sección de la interfaz gráfica contiene los elementos necesarios para el monitoreo y establecimiento de las variables del control de temperatura de la máquina mediante un teclado físico. Esta sección permite además establecer las temperaturas de referencia para el controlador de la máquina inyectora implementado en hardware, y encender o apagar el controlador de temperatura. También cuenta con un teclado numérico en pantalla de ser necesario en caso de existir algún problema en el teclado industrial del panel, como se muestra en la figura 4.3.

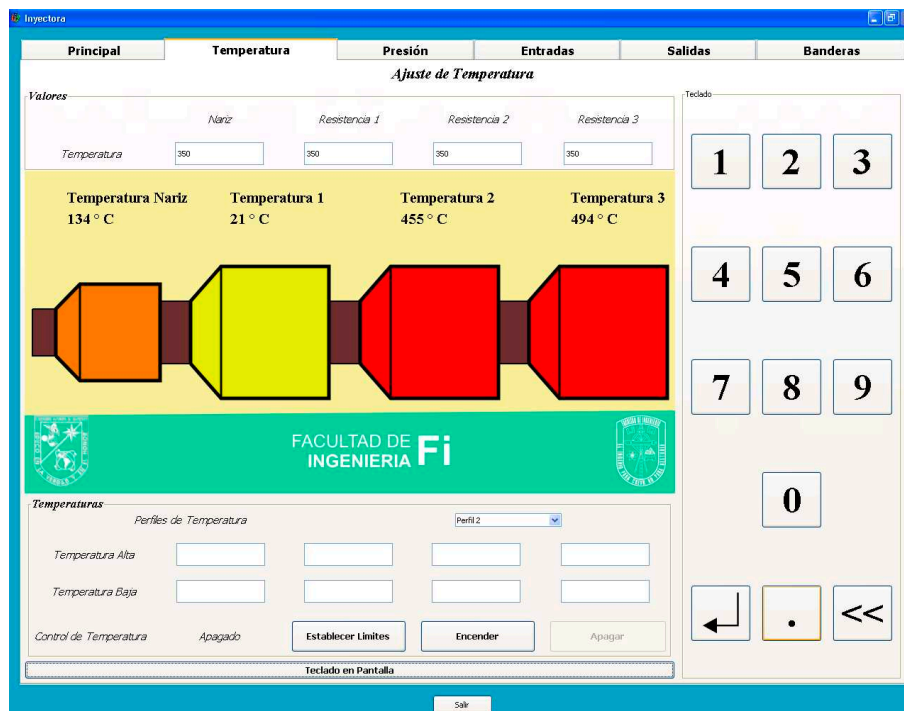


Figura 4.3: Segunda sección de la interfaz gráfica de usuario.

4.2. Diseño de la interfaz gráfica

La tercera sección consta de unos indicadores gráficos que muestran los valores de presión existentes en el proceso de inyección de plástico como se muestra en la figura 4.4.

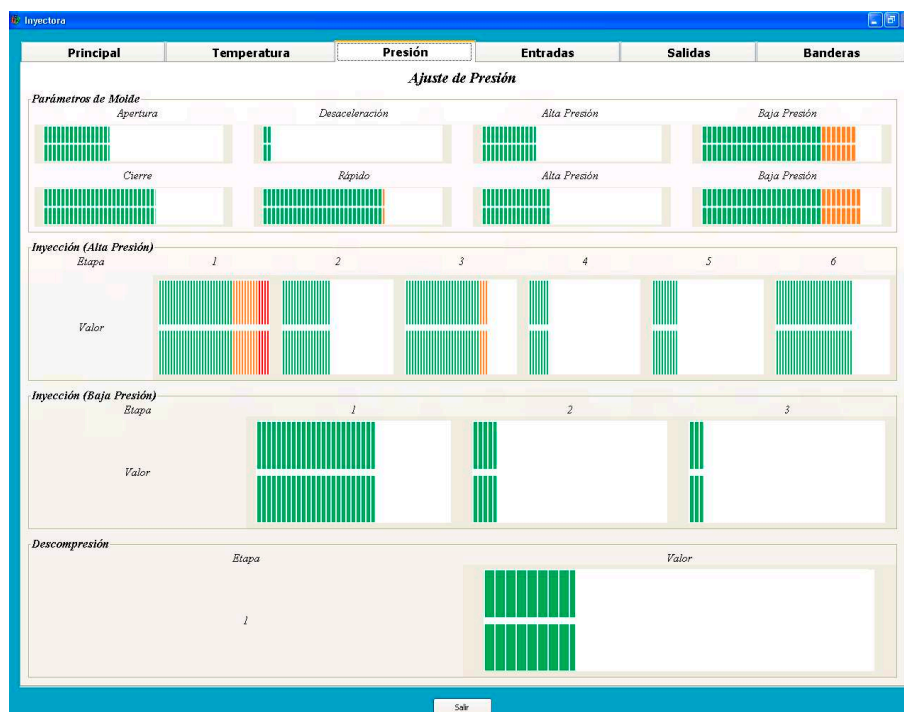
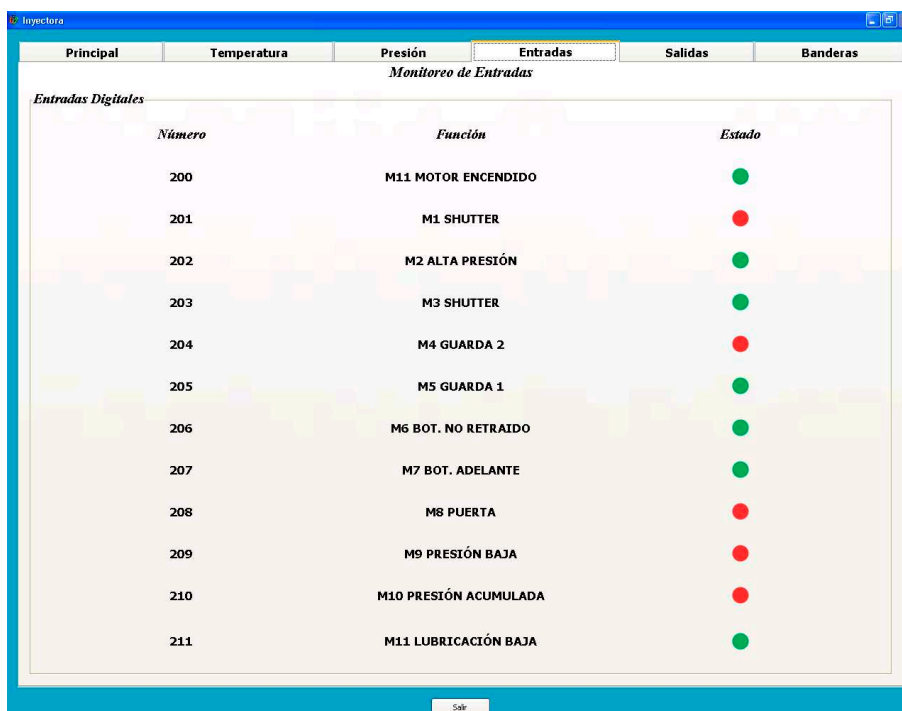


Figura 4.4: Monitoreo de Presión en la interfaz gráfica de usuario propuesta.

4.2. Diseño de la interfaz gráfica

Las secciones siguientes muestran los valores de las entradas, salidas y banderas digitales que cambian entre 2 estados (activo e inactivo), en estas secciones de la interfaz se muestra de manera gráfica el cambio de estado de cada entrada, salida o bandera digital como se muestra en la figura 4.5.



Número	Función	Estado
200	M11 MOTOR ENCENDIDO	●
201	M1 SHUTTER	●
202	M2 ALTA PRESIÓN	●
203	M3 SHUTTER	●
204	M4 GUARDA 2	●
205	M5 GUARDA 1	●
206	M6 BOT. NO RETRAIDO	●
207	M7 BOT. ADELANTE	●
208	M8 PUERTA	●
209	M9 PRESIÓN BAJA	●
210	M10 PRESIÓN ACUMULADA	●
211	M11 LUBRICACIÓN BAJA	●

Figura 4.5: Sección de monitoreo de entradas de la interfaz gráfica de usuario propuesta.

4.3. Programación de software

En base al planteamiento presentado en la metodología del presente trabajo se realizó la programación de los distintos módulos que comprende el software del panel de control, los cuales son la pantalla principal de la interfaz gráfica y los módulos de monitoreo y control de temperaturas, monitoreo de presiones, monitoreo de entradas digitales, monitoreo de salidas digitales y monitoreo de banderas digitales, los cuales reciben los datos de la tarjeta PLC de manera constante y muestran en pantalla los datos mas significativos del proceso de inyección.

Fue necesario escribir múltiples archivos de cabecera y archivos de implementación del lenguaje C++ que se muestran en el Apéndice B del presente trabajo. En dicho apéndice se muestran los archivos indicador.h, resistencias.h, rs232.h, e indicador.cpp en los cuales se muestran las distintas definiciones de variables internas, de miembros y de funciones de cada clase del software desarrollado.

4.4. Fabricación de estructura mecánica del panel

Una vez completada la etapa de diseño de una estructura mecánica necesaria para el soporte de los componentes en hardware del panel se procedió a la fabricación de moldes para la carcasa del panel fabricados con espuma de poliuretano.

Se completó la fabricación de 2 bloques rectangulares de espuma de poliuretano. Posteriormente se procedió a manufacturar los 2 moldes propuestos en CAD mediante el uso de la máquina de CNC que se encuentra en el laboratorio de la Facultad de Ingeniería Campus San Juan del Río de la UAQ.

Una vez completada la manufactura de los moldes se procedió a recubrir-

4.4. Fabricación de estructura mecánica del panel

los de resina de fibra de vidrio y posteriormente desmontar las piezas resultantes de ambos moldes. En las figuras 4.6 y 4.7 se muestra el resultado final de la manufactura de ambas carcasas.



Figura 4.6: Parte frontal de la carcasa de fibra de vidrio del Panel de Control.



Figura 4.7: Parte trasera de la carcasa de fibra de vidrio del Panel de Control.

4.4. Fabricación de estructura mecánica del panel

Fue necesario realizar algunos detalles finales sobre la carcasa como la manufactura de barrenos, fresado y detallado de algunos bordes, así como la manufactura de algunas piezas adicionales para el montaje de componentes como la fuente de poder y el disco duro (perfiles de aluminio y trabajo en lámina de acero). Una vez concluida la disposición de los elementos dentro del panel fue necesario dar un acabado final de pintura al panel. En la figura 4.8 se muestra la placa de lámina de acero montada sobre la carcasa trasera del panel.



Figura 4.8: Placa de sujeción de la fuente de poder.

4.5. Integración de dispositivos periféricos y ensamble

En esta etapa se integraron físicamente todos los dispositivos de hardware del panel a la carcasa fabricada en fibra de vidrio en una sola estructura mecánica. La pantalla táctil, los botones de inicio y de paro de emergencia, y el teclado industrial fueron montados en la carcasa delantera del panel de control.

La fuente de poder, el disco duro y la tarjeta madre fueron montados en la parte trasera de la carcasa del panel como se muestra en la figura 4.9



Figura 4.9: Ensamblaje de elementos en la carcasa trasera del panel de control.

4.6. Pruebas Simuladas

Cabe señalar que fueron requeridos distintos accesorios de tornillería para la sujeción de éstos elementos en la estructura propuesta. Por otra parte el teclado industrial y la pantalla táctil fueron integrados a la carcasa delantera del panel de control como se muestra en la figura 4.10.



Figura 4.10: Ensamblaje de elementos en la carcasa delantera del panel de control.

4.6. Pruebas Simuladas

Una vez ensamblados los elementos de hardware dentro de la estructura del panel de control fue necesario realizar pruebas en simulación para validar el correcto funcionamiento del software desarrollado en el sistema operativo Windows XP edición Professional con Service Pack 3, en conjunto con la tarjeta PLCUAQ816 que cuenta con 8 entradas analógicas y 8 salidas analógicas de +10 y -10V, 16 entradas y 16 salidas digitales TTL, convertidores ADC Y DAC, puertos USB y RS232, un FPGA de la familia Spartan 3E y un oscilador de 48 Mhz.

Estas pruebas fueron realizadas introduciendo voltajes variables a la entrada de los convertidores analógico/digitales de dicha tarjeta para las variables de temperatura y de presión, de tal manera que la variación de voltaje produce

4.6. Pruebas Simuladas

también un cambio en los indicadores de dichas variables. En las figuras 4.11, 4.12 y 4.13 se muestran la ventana principal y las ventanas de monitoreo de presión y de temperatura respectivamente ante un estímulo de voltaje externo en la tarjeta PLCUAQ816.

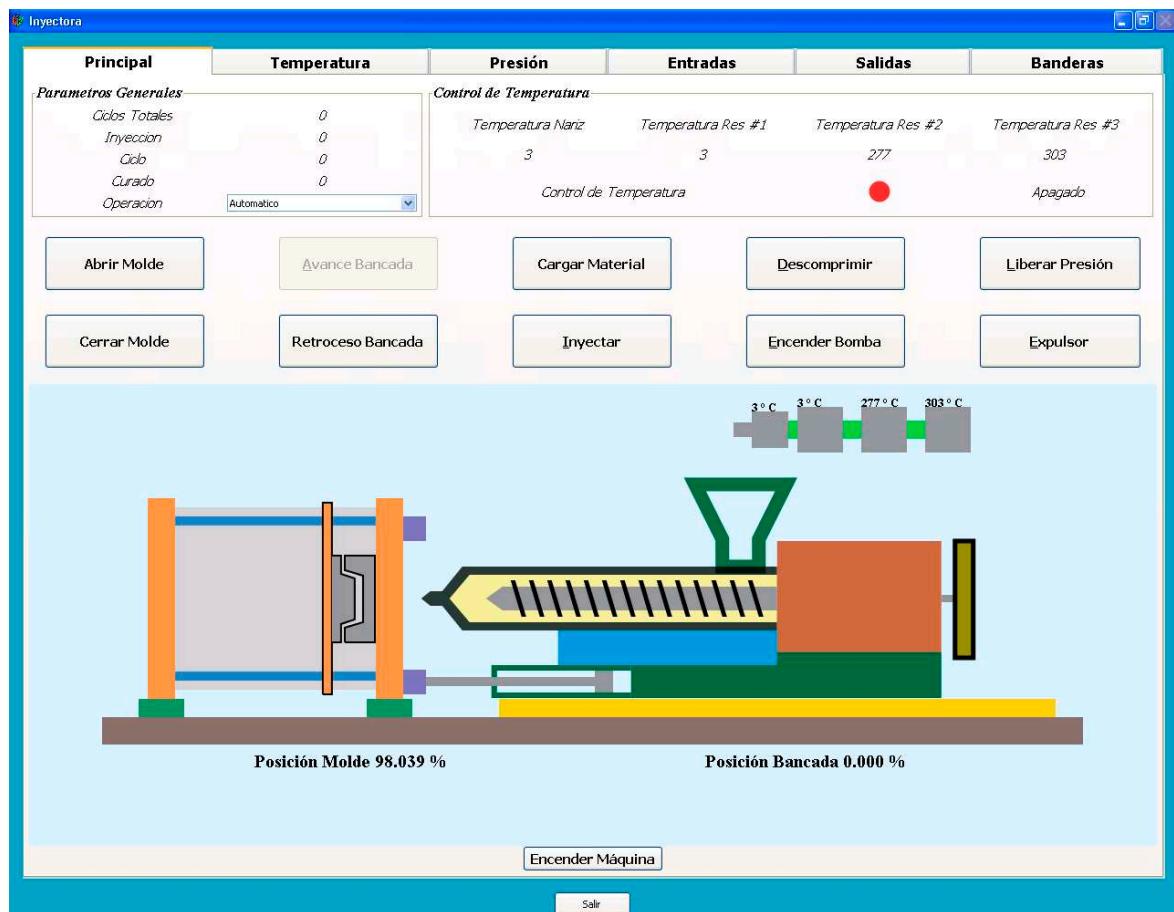


Figura 4.11: Simulación de monitoreo de las variables principales a monitorear.

Para la simulación de cambio de las variables como las entradas y banderas digitales se introdujeron valores discretos de 0 y 1 con la ayuda de un circuito externo mediante las entradas digitales TTL de la tarjeta PLCUAQ816, en las figuras 4.14, 4.15 y 4.16 se muestran las ventanas de entradas, salidas y banderas digitales ante un estímulo introducido en las entradas digitales de la tarjeta PLC.

4.6. Pruebas Simuladas

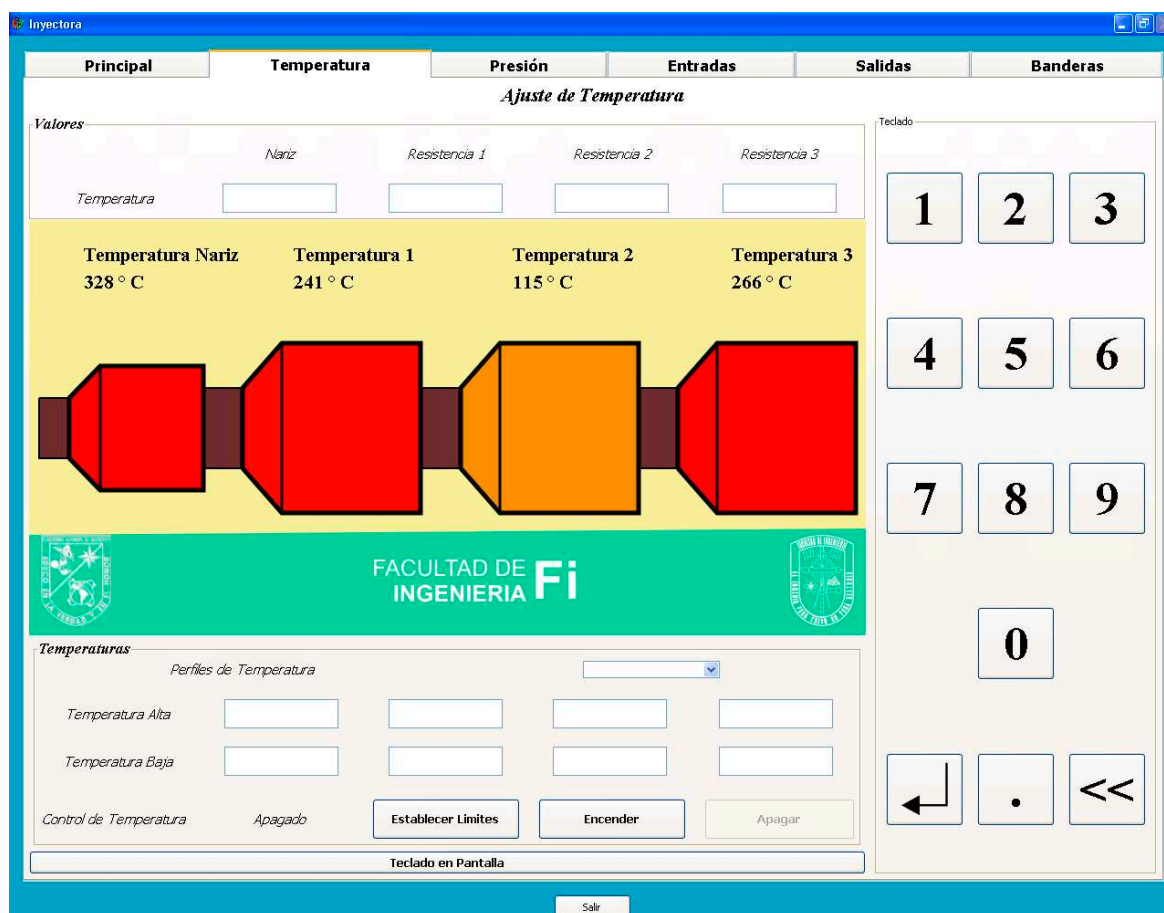


Figura 4.12: Simulación de monitoreo de la temperatura en las resistencias.

4.6. Pruebas Simuladas

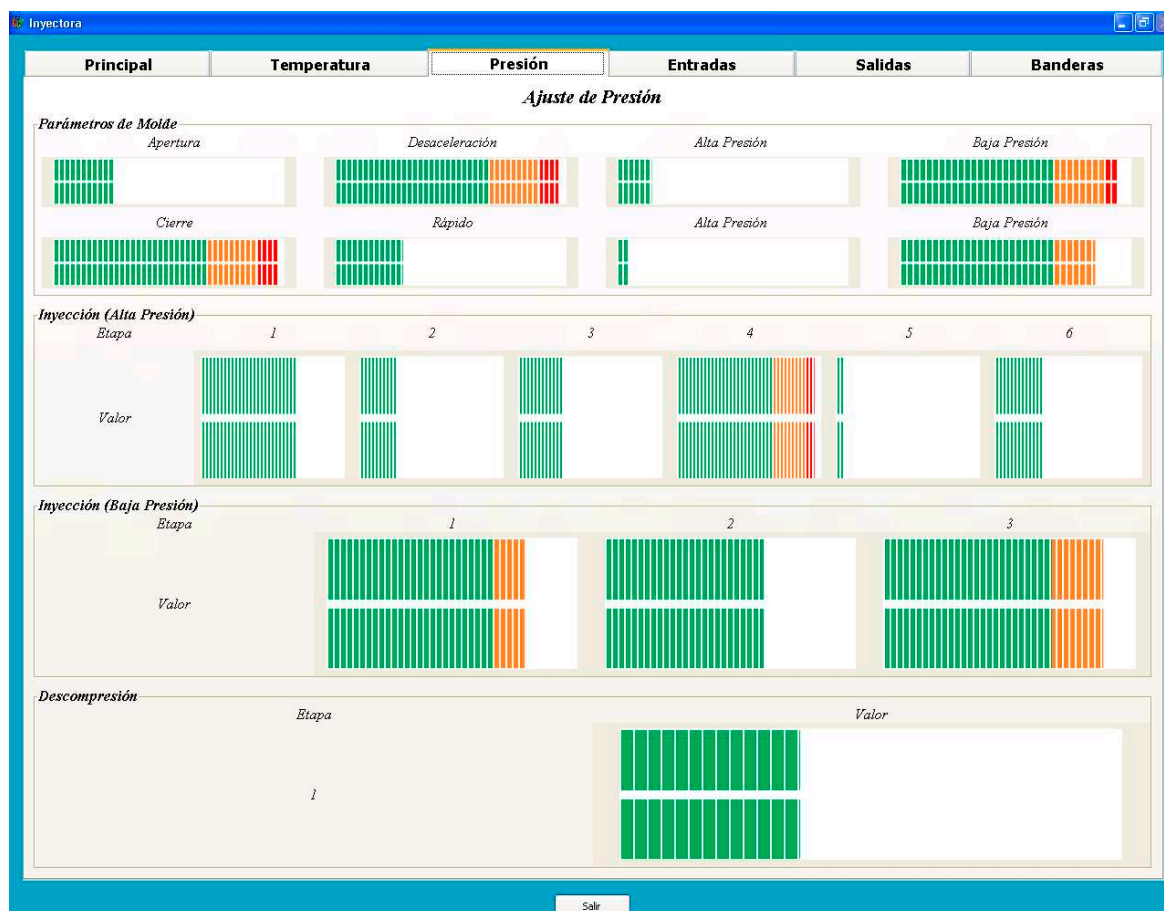


Figura 4.13: Simulación de monitoreo de las distintas presiones existentes en el proceso de inyección.

4.6. Pruebas Simuladas

Monitoreo de Entradas

Entradas Digitales

Número	Función	Estado
200	M11 MOTOR ENCENDIDO	●
201	M1 SHUTTER	●
202	M2 ALTA PRESIÓN	●
203	M3 SHUTTER	●
204	M4 GUARDA 2	●
205	M5 GUARDA 1	●
206	M6 BOT. NO RETRAIDO	●
207	M7 BOT. ADELANTE	●
208	M8 PUERTA	●
209	M9 PRESIÓN BAJA	●
210	M10 PRESIÓN ACUMULADA	●
211	M11 LUBRICACIÓN BAJA	●

Salir

Figura 4.14: Simulación de monitoreo de las entradas digitales en el proceso de inyección.

4.6. Pruebas Simuladas

Monitoreo de Relevadores Internos

Banderas

Número	Descripción	Estado	Número	Descripción	Estado
000	Bandera de Inyección	●	008	Inyección Etapa Baja	●
001	Inyección Etapa 1	●	009	Inyección Etapa Baja	●
002	Inyección Etapa 2	●	010	Carga	●
003	Inyección Etapa 3	●	011	Carga Etapa 1	●
004	Inyección Etapa 4	●	012	Carga Etapa 2	●
005	Inyección Etapa 5	●	013	Carga Etapa 3	●
006	Inyección Etapa 6	●	014	Descompresión Final	●
007	Inyección Etapa Baja	●			

Salir

Figura 4.15: Simulación de monitoreo de las banderas digitales en el proceso de inyección.

4.6. Pruebas Simuladas

Cabe señalar que estos valores una vez capturados por la tarjeta fueron enviados a la tarjeta madre a través del puerto USB de la tarjeta PLC, de esta manera la computadora recibe, interpreta y despliega los datos agrupados en 8 y 16 bits en la interfaz gráfica de usuario del software propuesto.

Capítulo V

Conclusiones

En el presente trabajo se llevó a cabo el desarrollo de un panel de control para una máquina de inyección de plástico, para la cual ya se contaba con diversos módulos de control, monitoreo y comunicación de datos en hardware, siendo necesaria la integración hardware/software mediante el desarrollo de software propuesto.

Gracias a este desarrollo de software, en conjunto con la propuesta de una estructura mecánica para el panel que permite integrar distintos elementos de hardware de computadora es posible monitorear y controlar las distintas variables existentes en un ciclo de la máquina de inyección de plástico.

El software desarrollado permite ser modificado y/o reestructurado en el futuro para ser integrado en otros desarrollos realizados dentro de la Facultad de Ingeniería del Campus San Juan del Río que utilicen plataformas de hardware similares.

Respecto a la generación de software en conjunto con elementos de hardware, en el futuro se puede implementar un sistema independiente de hardware de computadora que permita reducir las dimensiones de la estructura mecánica, y optimizar el consumo de recursos debido a que las aplicaciones de monitoreo y ajuste de variables no requieren una gran capacidad de cómputo.

Bibliografía

- Axelsson. J. USB Complete: everything you need to develop custom USB peripherals. Lakeview Research. 2005.
- Bravo C. A. Diseño de un control modular para maquinaria aplicado a una máquina de inyección de plástico. Tesis. Universidad Autónoma de Querétaro. 2004.
- Chavarín M. P. Control proporcional de servovalvula con FPGA aplicado a máquina de inyeccion de plástico. Tesis de Maestría. Universidad Autónoma de Querétaro. 2011.
- Chu. Pong P. FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version. John Wiley and Sons. 2011.
- Colín R. J. A. Descripción de los bloques funcionales de un controlador digital PID. Tesis de Maestría Universidad Autónoma de Querétaro. 2006.
- Dembowski K. Gran libro del hardware: información sobre la totalidad del hardware, de rápido acceso. Marcombo. 2003.
- Douglas M. Bryce. Plastic Injection Molding: Manufacturing Process Fundamentals. SME. 1996.
- Enríquez G. E. Sistema de control de temperatura aplicado a máquinas de inyeccion de plástico. Tesis. Universidad Autónoma de Querétaro. 2006.

BIBLIOGRAFÍA

- Franco G. L. A. Desarrollo de un sistema de monitoreo para máquinas-herramienta basado en FPGAs. Tesis de Doctorado. Universidad Autónoma de Querétaro. 2007.
- García H. A. El Control Automático en la Industria. Universidad de Castilla-La Mancha. 2005.
- García M. A. Algoritmos de control digital aplicados al sistema hidráulico de una máquina inyectora de plástico. Tesis de Maestría. Universidad Autónoma de Querétaro. 2008.
- González C. J. Reconversión y control de los ejes de una máquina RX-4A basado en FPGA. Tesis de Licenciatura. Universidad Autónoma de Querétaro. 2011.
- Hernández Z. J. Tarjeta de interfaz para aplicaciones de control en máquina inyectora de plástico usando lógica programable. Tesis de Licenciatura. 2008.
- Jaen. C. A. Desarrollo de perfiles polinomiales 3D basado en FPGA para control de posición en una máquina fresadora CNC. Tesis de Maestría. 2011.
- Kuon, I, R. Tessier y J. Rose. FPGA Architecture. Now Publishers Inc. 2008.
- Montes R. J., L. Castro y J. C. del Real. Procesos industriales para materiales no metálicos. Segunda Edición. Editorial Vision Net. 2006.
- Morales V. L. Unidad USB de control de posición y generación de perfiles para un intercambiador automático de herramientas. Tesis de Maestría. Universidad Autónoma de Querétaro. 2007.
- Morales V. L. Diseño de plataforma hardware-software para el desarrollo de aplicaciones industriales basadas en FPGA. Tesis de Doctorado. Universidad Autónoma de Querétaro. 2010.

BIBLIOGRAFÍA

- Muñoz B. B. Diseño de un procesador PLC basado en FPGA para aplicacion en maquinaria CNC. Tesis de Licenciatura. Universidad Autónoma de Querétaro. 2009.
- Muñoz B. B. Controlador Modular Reconfigurable para Maquina de Inyección de Plastico basado en FPGA. Tesis de Maestria. Universidad Autónoma de Querétaro. 2011.
- Osornio R. R. Diseño y construccion de una tarjeta controladora de 3 ejes. Tesis de Maestría. Universidad Autónoma de Querétaro. 2004.
- Reyes G. C. Desarrollo e implementacion de un interfaz usuario-maquina TOUCH-SCREEN para un sistema de monitoreo y control de granjas acuicolas. Tesis. Universidad Autónoma de Querétaro. 2008.
- Romero T. R. J. Procesamiento de señales para la deteccion de ruptura a herramientas en sistemas de manufactura por control numerico computarizado. Tesis de Doctorado. Universidad Autónoma de Querétaro. 2004.
- Stone D., Caroline J., Mark W. y Shailey M. User Interface Design and Evaluation. Editorial Morgan Kaufmann. 2005.

Apéndice

En las siguientes secciones se describirán de forma general algunos de los requisitos requeridos en primer lugar para poder utilizar las bibliotecas de gtkmm en Windows, compilar y generar los archivos ejecutables del software propuesto.

Instalación de Code::Blocks

Para la compilación del código en C++ del presente proyecto se usó el entorno de desarrollo libre Code::Blocks, ya que tiene la capacidad de ser muy extensible y configurable.

Code::Blocks puede ser extendido mediante el uso de plug-ins, además de que puede ser utilizado en los sistemas operativos Windows y Linux. Como primer punto fue necesario descargar el instalador de Code::Blocks sin compilador para el sistema operativo Microsoft Windows de la página oficial de Code::Blocks. Una vez realizada la instalación se procedió a la instalación de un compilador para C++.

Instalación de MinGW y utilidades para Win32

Una vez instalado el entorno de desarrollo se procedió a instalar el conjunto de utilidades y extras de MinGW(Minimalist GNU for Windows) para la arquitectura Win32.

MinGW provee un entorno nativo para desarrollo de aplicaciones en el sistema operativo Microsoft Windows, además de ser de código abierto y que no depende de enlaces a librerías dinámicas de terceros. Una vez instalado MinGW en la carpeta raíz del sistema, Code::Blocks está listo para compilar código que contenga funciones de la librería gtkmm, ya que MinGW incluye una gran cantidad de librerías además del compilador.

Código del software desarrollado en C++ y especificaciones de montaje de pantalla táctil

En esta sección se presentan algunos de los archivos de cabecera y de implementación generados mediante el entorno de desarrollo de Code::Blocks.

```
#ifndef __INDICADOR_H__
#define __INDICADOR_H__

#include <cairomm/context.h>
#include <gtkmm/drawingarea.h>

using namespace std;

class Indicador : public Gtk::DrawingArea
{
public:
    float presion;

    Indicador();
    ~Indicador();
    void valor_presion(char *value1);

private:
    void ClearBackground(Cairo::RefPtr<Cairo::Context> cr);
    void DibujarIndicador(Cairo::RefPtr<Cairo::Context> cr);
    bool on_expose_event(GdkEventExpose* event);
    bool redraw();
};

#endif // __INDICADOR_H__
```

```

#ifndef __RESISTENCIAS_H__
#define __RESISTENCIAS_H__

#include <cairomm/context.h>
#include <gtkmm/drawingarea.h>

class Resistencias : public Gtk::DrawingArea
{
public:
    Resistencias();
    ~Resistencias();
    void redraw();
    void set_temp(int *vt1, int *vt2, int *vt3, int *vt4);

protected:
    void ClearBackground(Cairo::RefPtr<Cairo::Context> cr);
    void DibujarResistencias(Cairo::RefPtr<Cairo::Context> cr);
    void Texto(Cairo::RefPtr<Cairo::Context> cr);
    void DibujarImagen();
    bool on_expose_event(GdkEventExpose* event);

private:
    float tx, ty, tz, tx1, ty1, tz1, tx2, ty2, tz2, tx3, ty3, tz3;
    int tmp1, tmp2, tmp3, tmp4, trlx, trly;
};

#endif // __RESISTENCIAS_H__

```

```
#ifndef __RS232_H__
#define __RS232_H__

#include <windows.h>

class RS232
{
private:
    HANDLE fd;
    DCB OldConf;
    int n, op, i;

public:
    RS232();
    ~RS232();
    char cad2[16];

    void abrir_puerto();
    void cerrar_puerto();
    void leer_datos(const int);
    void escribir_datos(char*, const int);
};

#endif // __RS232_H__
```

```

#include "indicador.h"

Indicador::Indicador()
{
    Glib::signal_timeout().connect( sigc::mem_fun(*this, &Indicador::redraw), 50 )
        ;
    #ifndef GLIBMM_DEFAULT_SIGNAL_HANDLERS_ENABLED
        //Connect the signal handler if it isn't already a virtual method override:
        signal_expose_event().connect(sigc::mem_fun(*this, &Indicador::
            on_expose_event), false);
    #endif //GLIBMM_DEFAULT_SIGNAL_HANDLERS_ENABLED
}

Indicador::~~Indicador()
{
}

void Indicador::ClearBackground(Cairo::RefPtr<Cairo::Context> cr)
{
    cr->save();
    cr->set_source_rgba(1, 1, 1, 0);
    cr->paint();
    cr->stroke();
    cr->restore();
}

void Indicador::DibujarIndicador(Cairo::RefPtr<Cairo::Context> cr)
{
    Gtk::Allocation allocation = get_allocation();
    const int width = allocation.get_width();
    const int height = allocation.get_height();
    int a=width/20, b=height/20;

    cr->save();
    cr->set_source_rgb(0, .6588, 0.349);
    cr->rectangle((width/20), (height/20), ((13*width)/20), ((18*height)/20));
    cr->fill();
    cr->stroke();
    cr->restore();

    cr->save();
    cr->set_source_rgba(.9803, .5019, .1843, 1);
    cr->rectangle(((13*width)/20), (height/20), ((4*width)/20), ((18*height)/20));
}

```

```

cr->fill();
cr->stroke();
cr->restore();

cr->save();
cr->set_source_rgba(1, 0, 0, 1);
cr->rectangle(((17*width)/20), (height/20), ((2*width)/20), ((18*height)/20));
cr->fill();
cr->stroke();
cr->restore();

cr->save();
cr->set_source_rgba(1, 1, 1, 1);
cr->move_to(width/20, height/2);
cr->set_line_width(height/15);
cr->line_to(width/20, height/2);
cr->line_to(((19*width)/20), height/2);
cr->stroke();
cr->restore();
cr->save();

for(int i=(width/20); i<=((19*width)/20); i=i+width/40)
{
    cr->set_source_rgba(1, 1, 1, 1);
    cr->move_to((width/20), (height/20));
    //cr->set_line_width(height/15);
    cr->line_to(i,(height/20));
    cr->line_to(i,((19*height)/20));
    cr->stroke();
    cr->restore();
    cr->save();
}

cr->set_source_rgba(1, 1, 1, 1);
cr->save();
cr->line_to(((19*width)/20),(height/20));
cr->line_to(((19*width)/20),((19*height)/20));
cr->line_to(((19*width)/20)-((18*width/20)*presion),((19*height)/20));
cr->line_to(((19*width)/20)-((18*width/20)*presion),(height/20));
cr->close_path();
cr->fill();
cr->stroke();
cr->restore();

```

```

}

bool Indicador::on_expose_event(GdkEventExpose* event)
{
    //printf("\nLa ventana tiene tamaño %t%d -z %t%d", width1, height1);
    Glib::RefPtr<Gdk::Window> win = get_window();
    if(!win) return false;
    Cairo::RefPtr<Cairo::Context> cr = win->create_cairo_context();

    ClearBackground(cr);
    DibujarIndicador(cr);

    return true;
}

bool Indicador::redraw()
{
    // force our program to redraw the entire clock.
    Glib::RefPtr<Gdk::Window> win = get_window();
    if (win)
    {
        Gdk::Rectangle r(0, 0, get_allocation().get_width(),
            get_allocation().get_height());
        win->invalidate_rect(r, false);
    }

    return true;
}

void Indicador::valor_presion(char *value1)
{
    presion = *value1;
    presion = 1-((float)((unsigned char)(presion))/255);
}

```

```

#include "resistencias.h"
#include <iostream>

using namespace std;

Resistencias::Resistencias(): tx(1), ty(0.4), tz(0), tx1(1), ty1(0.4),
tz1(0), tx2(1), ty2(0.4), tz2(0), tx3(1), ty3(0.4), tz3(0), tmp1(0),
tmp2(0), tmp3(0), tmp4(0)
{
// Glib::signal_timeout().connect(sigc::mem_fun(*this, &Resistencias::redraw),
// 50);
//
// #ifndef GLIBMM_DEFAULT_SIGNAL_HANDLERS_ENABLED
// //Connect the signal handler if it isn't already a virtual method
// override:
// signal_expose_event().connect(sigc::mem_fun(*this, &Resistencias::
// on_expose_event), true);
// #endif //GLIBMM_DEFAULT_SIGNAL_HANDLERS_ENABLED
}

Resistencias::~Resistencias(){
}

void Resistencias::ClearBackground(Cairo::RefPtr<Cairo::Context> cr)
{
    Glib::RefPtr<Gdk::Window> win = get_window();
    Gtk::Allocation allocation = get_allocation();
    const int width1 = allocation.get_width();
    const int height1 = allocation.get_height();

    cr->save();
    cr->set_source_rgb(.949, .9333, 0.6235);
    cr->paint();
    cr->stroke();
    cr->restore();
    cr->save();

    cr->set_source_rgb(.25, .8, .6);
    //cr->set_line_width(20.0); // make the line wider
    cr->line_to(0, 345);
    cr->line_to(width1, 338);
    cr->line_to(width1, height1);
    cr->line_to(0, height1);
}

```

```

        cr->close_path();
        cr->fill();
        cr->stroke();
        cr->restore();
        cr->save();
    }

bool Resistencias::on_expose_event(GdkEventExpose* event)
{
    Glib::RefPtr<Gdk::Window> win = get_window();
    Gtk::Allocation allocation = get_allocation();
    const int width1 = allocation.get_width();
    const int height1 = allocation.get_height();

    if(!win) return false;
    Cairo::RefPtr<Cairo::Context> cr = win->create_cairo_context();

    ClearBackground(cr);
    DibujarResistencias(cr);
    DibujarImagen();
    Texto(cr);

    return true;
}

void Resistencias::DibujarImagen()
{
    Glib::RefPtr<Gdk::Window> win = get_window();
    Gtk::Allocation allocation = get_allocation();
    const int width2 = allocation.get_width();
    const int height2 = allocation.get_height();

    Glib::RefPtr<Gdk::Pixbuf> image = Gdk::Pixbuf::create_from_file("pie.png");
    image->render_to_drawable(get_window(), get_style()->get_black_gc(), 0, 0, (
        width2-(image->get_width())/2, 345,
        image->get_width(), image->get_height(), // draw the whole
        image (from 0,0 to the full width,height) at 100,80
        in the window
        Gdk::RGB_DITHER_NONE, 0, 0);
}

void Resistencias::DibujarResistencias(Cairo::RefPtr<Cairo::Context> cr)
{

```

```

Gtk::Allocation allocation = get_allocation();
const int width1 = allocation.get_width();
const int height1 = allocation.get_height();

trlx=(width1-895)/2;
trly=(height1-187)/2;

cr->save();
//resistencia de la nariz(1er rectangulo)
cr->set_source_rgb(0.4, 0.2, 0.2);
cr->line_to(0+trlx,61+trly);
cr->line_to(32+trlx,61+trly);
cr->line_to(32+trlx,127+trly);
cr->line_to(0+trlx,127+trly);
cr->line_to(0+trlx,61+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();

//resistencia de la nariz 2
cr->set_source_rgb(tx, ty, tz);
cr->line_to(33+trlx,127+trly);
cr->line_to(67+trlx,162+trly);
cr->line_to(180+trlx,162+trly);
cr->line_to(180+trlx,26+trly);
cr->line_to(67+trlx,26+trly);
cr->line_to(33+trlx,60+trly);
cr->line_to(33+trlx,127+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();

//parte 1 del cañon
cr->set_source_rgb(0.4, 0.2, 0.2);
cr->line_to(180+trlx,138+trly);
cr->line_to(222+trlx,138+trly);
cr->line_to(222+trlx,50+trly);
cr->line_to(180+trlx,50+trly);
cr->line_to(180+trlx,138+trly);
cr->fill();
cr->stroke();

```

```
cr->restore();
cr->save();

//resistencia 1
cr->set_source_rgb(tx1, ty1, tz1);
cr->line_to(223+trlx,138+trly);
cr->line_to(266+trlx,187+trly);
cr->line_to(418+trlx,187+trly);
cr->line_to(418+trlx,1+trly);
cr->line_to(266+trlx,1+trly);
cr->line_to(223+trlx,50+trly);
cr->line_to(223+trlx,138+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();

//parte 2 del cañon
cr->set_source_rgb(0.4, 0.2, 0.2);
  cr->line_to(418+trlx,138+trly);
cr->line_to(461+trlx,138+trly);
cr->line_to(461+trlx,50+trly);
cr->line_to(418+trlx,50+trly);
cr->line_to(418+trlx,138+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();

//resistencia 2
  cr->set_source_rgb(tx2, ty2, tz2);
  cr->line_to(462+trlx,138+trly);
cr->line_to(505+trlx,187+trly);
cr->line_to(657+trlx,187+trly);
cr->line_to(657+trlx,1+trly);
cr->line_to(505+trlx,1+trly);
cr->line_to(462+trlx,50+trly);
cr->line_to(462+trlx,138+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();
```

```

//parte 3 del cañon
cr->set_source_rgb(0.4, 0.2, 0.2);
  cr->line_to(656+trlx,138+trly);
cr->line_to(699+trlx,138+trly);
cr->line_to(699+trlx,50+trly);
cr->line_to(656+trlx,50+trly);
cr->line_to(656+trlx,138+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();

//resistencia 3
  cr->set_source_rgb(tx3, ty3, tz3);
  cr->line_to(700+trlx,138+trly);
cr->line_to(743+trlx,187+trly);
cr->line_to(895+trlx,187+trly);
cr->line_to(895+trlx,1+trly);
cr->line_to(743+trlx,1+trly);
cr->line_to(700+trlx,50+trly);
cr->line_to(700+trlx,138+trly);
cr->fill();
cr->stroke();
cr->restore();
cr->save();

//contorno resistencia nariz 1
cr->set_source_rgb(0, 0, 0);
cr->set_line_width(2);
  cr->line_to(0+trlx,61+trly);
cr->line_to(32+trlx,61+trly);
cr->line_to(32+trlx,127+trly);
cr->line_to(0+trlx,127+trly);
cr->line_to(0+trlx,61+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno resistencia nariz 2
cr->set_source_rgb(0, 0, 0);
cr->set_line_width(5);
  cr->line_to(33+trlx,127+trly);
cr->line_to(67+trlx,162+trly);

```

```

cr->line_to(180+trlx,162+trly);
cr->line_to(180+trlx,26+trly);
cr->line_to(67+trlx,26+trly);
cr->line_to(33+trlx,61+trly);
cr->line_to(33+trlx,127+trly);
cr->stroke();
cr->restore();
cr->save();

cr->set_line_width(3);
cr->move_to(67+trlx,26+trly);
cr->line_to(67+trlx,26+trly);
cr->line_to(67+trlx,162+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno parte 1 del cañon
cr->set_source_rgb(0, 0, 0);
cr->set_line_width(2);
    cr->line_to(180+trlx,138+trly);
cr->line_to(222+trlx,138+trly);
cr->line_to(222+trlx,50+trly);
cr->line_to(180+trlx,50+trly);
cr->line_to(180+trlx,138+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno resistencia 1
cr->set_source_rgb(0, 0, 0);
cr->set_line_width(5);
    cr->line_to(223+trlx,138+trly);
cr->line_to(266+trlx,187+trly);
cr->line_to(418+trlx,187+trly);
cr->line_to(418+trlx,1+trly);
cr->line_to(266+trlx,1+trly);
cr->line_to(223+trlx,50+trly);
cr->line_to(223+trlx,138+trly);
cr->stroke();
cr->restore();
cr->save();

```

```
cr->set_line_width(3);
cr->move_to(266+trlx,1+trly);
cr->line_to(266+trlx,1+trly);
cr->line_to(266+trlx,187+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno parte 2 del cañon
cr->set_source_rgb(0, 0, 0);
cr->set_line_width(2);
  cr->line_to(418+trlx,138+trly);
cr->line_to(461+trlx,138+trly);
cr->line_to(461+trlx,50+trly);
cr->line_to(418+trlx,50+trly);
cr->line_to(418+trlx,138+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno resistencia 3
  cr->set_source_rgb(0, 0, 0);
  cr->set_line_width(5);
  cr->line_to(462+trlx,138+trly);
cr->line_to(505+trlx,187+trly);
cr->line_to(657+trlx,187+trly);
cr->line_to(657+trlx,1+trly);
cr->line_to(505+trlx,1+trly);
cr->line_to(462+trlx,50+trly);
cr->line_to(462+trlx,138+trly);
cr->stroke();
cr->restore();
cr->save();

cr->set_line_width(3);
cr->move_to(505+trlx,1+trly);
cr->line_to(505+trlx,1+trly);
cr->line_to(505+trlx,187+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno parte 3 del cañon
```

```

cr->set_source_rgb(0, 0, 0);
cr->set_line_width(2);
cr->line_to(656+trlx,138+trly);
cr->line_to(699+trlx,138+trly);
cr->line_to(699+trlx,50+trly);
cr->line_to(656+trlx,50+trly);
cr->line_to(656+trlx,138+trly);
cr->stroke();
cr->restore();
cr->save();

//contorno resistencia 3
cr->set_source_rgb(0, 0, 0);
cr->set_line_width(5);
cr->line_to(700+trlx,138+trly);
cr->line_to(743+trlx,187+trly);
cr->line_to(895+trlx,187+trly);
cr->line_to(895+trlx,1+trly);
cr->line_to(743+trlx,1+trly);
cr->line_to(700+trlx,50+trly);
cr->line_to(700+trlx,138+trly);
cr->stroke();
cr->restore();
cr->save();

cr->set_line_width(3);
cr->move_to(743+trlx,1+trly);
cr->line_to(743+trlx,1+trly);
cr->line_to(743+trlx,187+trly);
cr->stroke();
cr->restore();
cr->save();
}

void Resistencias::Texto(Cairo::RefPtr<Cairo::Context> cr)
{
    Gtk::Allocation allocation = get_allocation();
    const int width1 = allocation.get_width();
    const int height1 = allocation.get_height();
    //cairo_text_extents_t te;

    char t1[10];
    char t2[10];

```

```

char t3[10];
char t4[10];

cr->move_to(49+trlx, height1/10);
cr->set_source_rgb(0, 0, 0);
cr->select_font_face("Times New Roman", Cairo::FONT_SLANT_NORMAL , Cairo::
    FONT_WEIGHT_BOLD);
cr->set_font_size(21);
cr->show_text("Temperatura Nariz");

cr->move_to(49+trlx, height1/10 + 30);
    sprintf(t1, "%d", tmp1);
    strcat(t1, " \xC2\xB0 C");
cr->show_text(t1);
cr->stroke();
cr->restore();

cr->save();
cr->move_to(279+trlx, height1/10);
cr->set_source_rgb(0, 0, 0);
cr->select_font_face("Times New Roman", Cairo::FONT_SLANT_NORMAL , Cairo::
    FONT_WEIGHT_BOLD);
cr->set_font_size(21);
cr->show_text("Temperatura 1");

cr->move_to(279+trlx, height1/10+30);
    sprintf(t2, "%d", tmp2);
    strcat(t2, " \xC2\xB0 C");
cr->show_text(t2);
cr->stroke();
cr->restore();

cr->save();
cr->move_to(519+trlx, height1/10);
cr->set_source_rgb(0, 0, 0);
cr->select_font_face("Times New Roman", Cairo::FONT_SLANT_NORMAL , Cairo::
    FONT_WEIGHT_BOLD);
cr->set_font_size(21);
cr->show_text("Temperatura 2");

cr->move_to(519+trlx, height1/10+30);
    sprintf(t3, "%d", tmp3);
    strcat(t3, " \xC2\xB0 C");

```

```

cr->show_text(t3);
cr->stroke();
cr->restore();

cr->save();
cr->move_to(759+trlx, height1/10);
cr->set_source_rgb(0, 0, 0);
cr->select_font_face("Times New Roman", Cairo::FONT_SLANT_NORMAL , Cairo::
    FONT_WEIGHT_BOLD);
cr->set_font_size(21);
cr->show_text("Temperatura 3");

cr->move_to(759+trlx, height1/10 + 30);
    sprintf(t4, "%i", tmp4);
    strcat(t4, " \xC2\xB0 C");
cr->show_text(t4);
cr->stroke();
cr->restore();
}

void Resistencias::set_temp(int *vt1, int *vt2, int *vt3, int *vt4)
{
    tmp1=*vt1;
    tmp2=*vt2;
    tmp3=*vt3;
    tmp4=*vt4;

    tx = .9019;
    ty = (1-((float)(*vt1)/255));
    tz = 0;

    tx1 = .9019;
    ty1 = (1-((float)(*vt2)/255));
    tz1 = 0;

    tx2 = .9019;
    ty2 = (1-((float)(*vt3)/255));
    tz2 = 0;

    tx3 = .9019;
    ty3 = (1-((float)(*vt4)/255));
    tz3 = 0;

    //cout<<*vt1<<" "<<*vt1<<" "<<*vt2<<" "<<*vt3<<" "<<*vt4<<" "<<endl;

```

```
}  
  
void Resistencias::redraw()  
{  
    // force our program to redraw the entire clock.  
    Glib::RefPtr<Gdk::Window> win = get_window();  
    if (win)  
    {  
        Gdk::Rectangle r(0, 0, get_allocation().get_width(),  
            get_allocation().get_height());  
        win->invalidate_rect(r, false);  
    }  
}
```

```

#include <iostream>
#include "rs232.h"

#define __WINDOWS_COM__
#include "serial.h"

using namespace std;

RS232::RS232()
{
}

RS232::~~RS232()
{
}

void RS232::abrir_puerto()
{
    fd = Open_Port("COM1");           // Abre el puerto serie COM#
    OldConf = Get_Configure_Port(fd); // Devuelve una estructura con la
        configuracion actual del puerto
    Configure_Port(fd , B9600, "8N1"); // Se configura el puerto
    IO_Blocking(fd,FALSE);           // Selecciona la opcion no
        bloqueante
}

void RS232::cerrar_puerto()
{
    Set_Configure_Port(fd,OldConf);   // Restituyo la antigua
        configuracion del puerto
    Close_Port(fd);
}

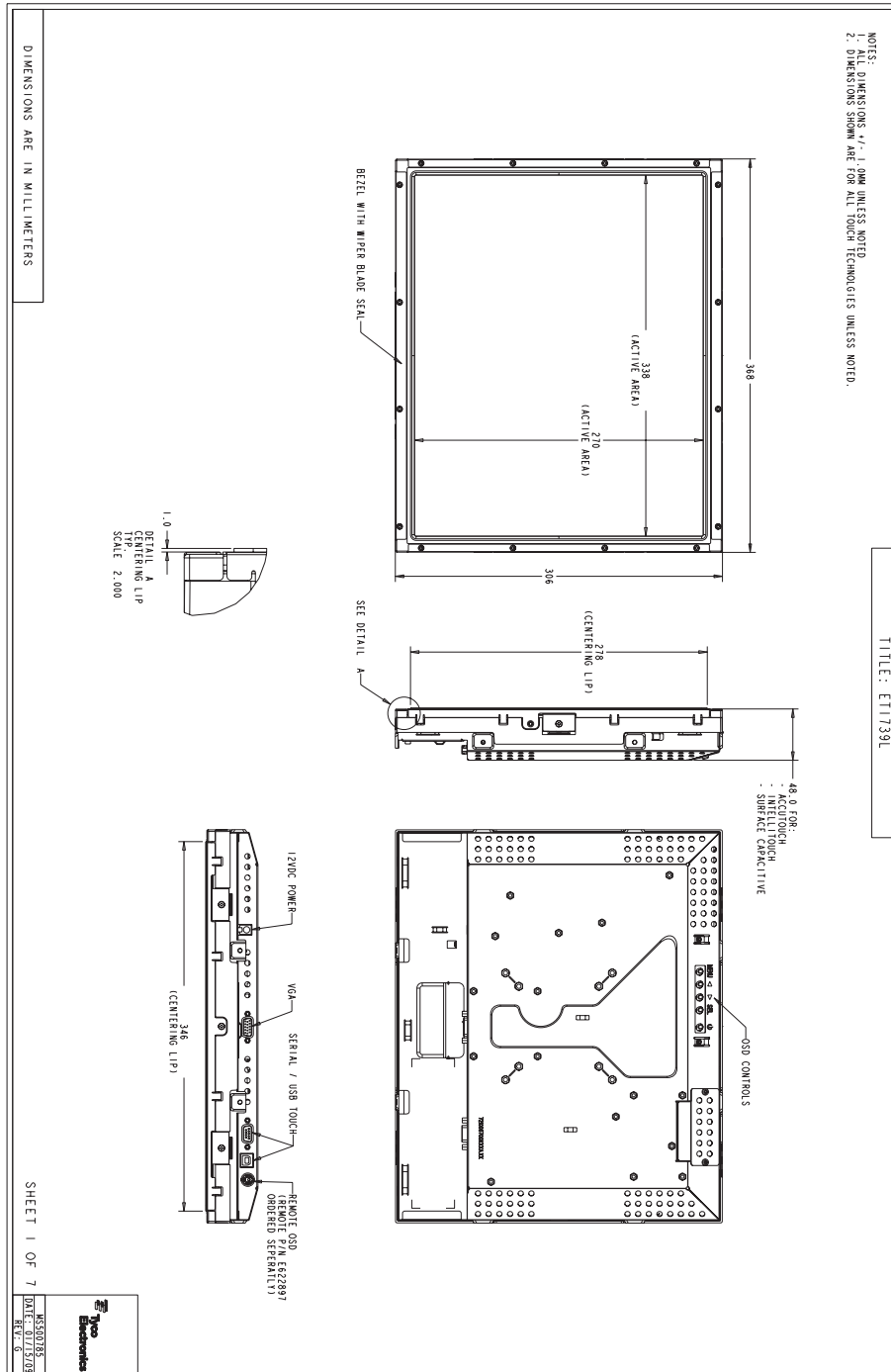
void RS232::leer_datos(const int nrd)
{
    while (Kbhit_Port(fd)<nrd);       // Espero a leer hasta que
        tenga 16 bytes en el buffer de entrada
    {n = (int) Read_Port(fd , cad2 , nrd);} // Leo el puerto serie
    //printf("\n %s", cad2);
}

void RS232::escribir_datos(char *s, const int nwd)

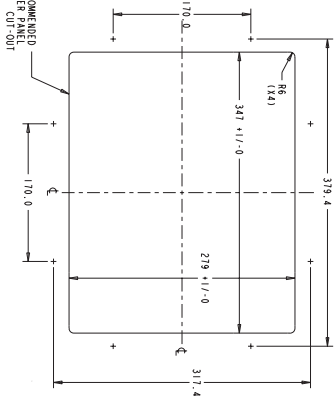
```

```
{  
    n = (int) Write_Port(fd, s, nwd);           // Escribo en el puerto  
        serie  
    //printf("\n%s", s);  
}
```

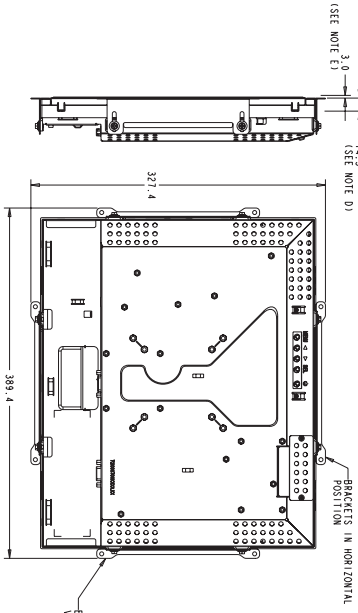
Especificaciones de la pantalla táctil EloTouch 1739L



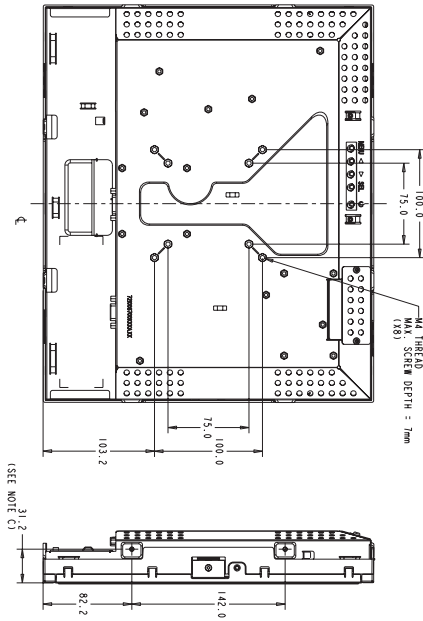
**ENDED CUSTOMER
UT-OUT (SEE NOTE A)**



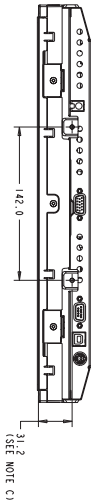
**REAR MOUNT WITH BRACKETS,
HORIZONTAL OR VERTICAL (SEE NOTE B)**



VESA MOUNT



**SIDE MOUNT,
HORIZONTAL**



DIMENSIONS ARE IN MILLIMETERS

SHEET 5 OF 7

REV. 6

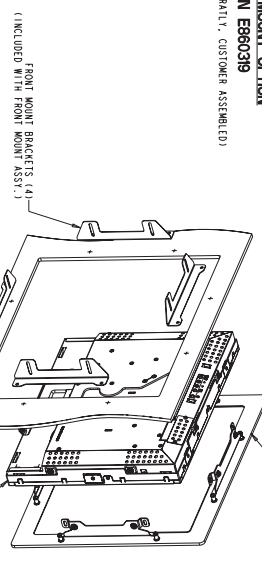
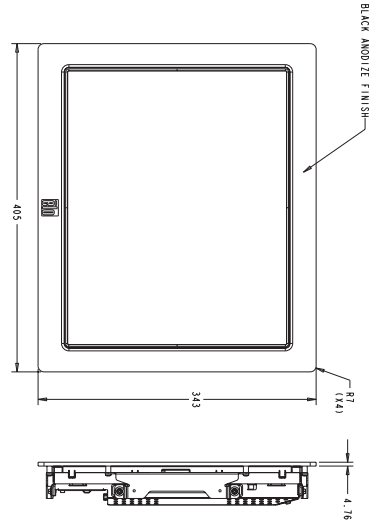
DATE: 03/27/08

DESIGNER: J. W. W.

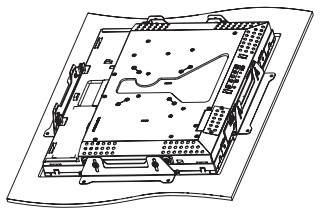
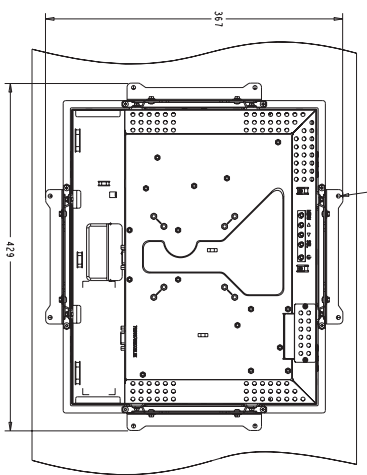
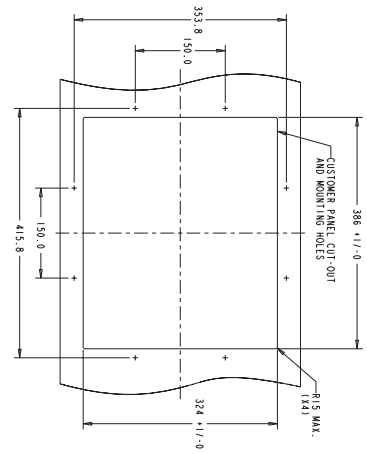


TITLE: ET1739L

FRONT MOUNT OPTION
P/N: E96039
 (ORDERED SEPARATELY, CUSTOMER ASSEMBLED)



- Ø5.0 NOTE:
 (1X3) 1. IF CUSTOMER PANEL MATERIAL IS WOOD:
 Ø4.5mm HOLES THRU FOR M4 SCREWS, Ø4mm OR NO THRU HOLES; SCREWS FOR WOOD Ø4mm.
 2. IF CUSTOMER PANEL MATERIAL IS SHEET METAL:
 CONCEALED HEAD THREADED STUDS M4 WITH NUTS.



DIMENSIONS ARE IN MILLIMETERS

SHEET 6 OF 7



1739L 17" LCD Rear-Mount Touchmonitor Specifications

Case/Bezel Color	Steel/black
Display	Size: 17.0" diagonal Type: Active matrix TFT LCD Aspect ratio: 5 x 4
Useful Screen Area	Horizontal: 13.3" (338 mm); Vertical: 10.6" (270 mm)
Monitor Dimensions	Width: 14.5" (368 mm) Height: 12.0" (306 mm) Depth: 1.9" (49 mm)
Optimal (native) Resolution	1280 x 1024
Colors	16.7 million
Brightness	LCD panel: 300 cd/m ² (typical); with AccuTouch: 246 cd/m ² (typical); with CarrollTouch: 276 cd/m ² (typical); with IntelliTouch: 276 cd/m ² (typical); with SecureTouch: 273 cd/m ² (typical); with Surface Capacitive: 255 cd/m ² (typical)
Response Time	7.2 msec (typical)
Viewing Angle (typical, from center)	Horizontal (left/right): ±80° or 160° total Vertical (up/down): ±80° or 160° total
Contrast Ratio	800:1 typical
Input Video Format	RGB analog only
Input Sync Format	Separate horizontal and vertical sync, composite sync and sync-on-green
Input Video Signal Connector	Mini D-Sub 15-Pin VGA type
Input Frequency	Horizontal: 31.5-80.0 kHz; Vertical: 56.3-75 Hz
Power supply	External DC—optional power brick (sold separately) Input voltage—DC: +12VDC ±5% at 2.5 A max. Input power connector specification (on monitor)—Type: DC Barrel Jack; Barrel inner diameter: 6.4 mm (±0.3 mm); Pin outer diameter: 2.0 mm (+0.0 -0.1 mm); Barrel depth: 8.8 mm (±0.3 mm) Power connector (on power brick)—Type: DC Barrel Plug; Barrel outer diameter: 5.5 mm (±0.1 mm); Pin inner diameter: 2.1 mm (±0.1 mm); Barrel length: 9.5 mm (±0.5 mm)
Power consumption (typical)	30W
Power Dissipation	Monitor only: 30 W typical, 36 W max. Monitor and power brick: 33 W typical, 40 W max.
Temperature	Operating: 0°C to 40°C; Storage: -20°C to 60°C
Humidity (noncondensing)	Operating: 20%-80%; Storage: 10%-90
Weight (approx.)	Actual: 8.8 lb (4.0 kg); Shipping: 11.5 lb (5.2 kg)
Warranty	Monitor: 3 years Backlight lamp life: typical 50,000 hours to half brightness MTBF: 50,000 hours demonstrated
Agency Approvals	UL, cUL, SEMKO, CB, CE/FCC/VCCI/C-Tick (Class B) On-screen display (OSD): Digital OSD or optional remote OSD with 1.8 m cable
User's Controls	Optional remote OSD: Auto/Sel, up, down, menu OSD buttons: menu, left, right, select, power OSD: contrast, brightness, H/V position, RGB (color temp), clock, phase, recall, language (English, German, Spanish, Japanese, French) OSD disable/enable: power, OSD menu
Mounting Options	75 mm or 100 mm VESA mount Rear mount includes standard mounting brackets Front mount with optional bezel (see Accessories for details) Rack mount with optional bracket (see Accessories for details)
Other Features	Fully RoHS compliant; patent pending bezel seal; patented touch technology
Other Supported Resolutions	See www.elotouch.com for additional supported resolutions

To find out more about our extensive range of Elo touch solutions, go to www.elotouch.com, or call the office nearest you.

North America	800-ELO-TOUCH	Europe	Asia-Pacific	Latin America
Elo TouchSystems	Tel 1-650-361-4800	Tel +32 (0)16 35 21 00	Tel +81 (45) 478-2161	Tel 786-923-0251
301 Constitution Drive	Fax 1-650-361-4722	Fax +32 (0)16 35 21 01	Fax +81 (45) 478-2180	Fax 305-931-0124
Menlo Park, CA 94025-1110	customerservice@elotouch.com	elosales@elotouch.com	www.tps.co.jp	www.elotouch.com.ar



Tyco Electronics Corporation and its Affiliates in the TE Touch Solutions business unit of the TE Connectivity Ltd. family of companies (collectively "TE") reserves the right to change or update, without notice, any information contained herein; to change, without notice, the design, construction, materials, processing or specifications of any products; and to discontinue or limit production or distribution of any products.
CarrollTouch, Elo (logo), Elo TouchSystems, IntelliTouch, TE Connectivity, TE connectivity (logo) and TE (logo) are trademarks of the TE Connectivity Ltd. family of companies.

Copyright 2011 Tyco Electronics Corporation, a TE Connectivity Ltd. Company. All rights reserved. 04/11

