



Universidad Autónoma de Querétaro  
 Facultad de Ingeniería  
 Doctorado en Ingeniería

**CONTROL DE ACCESO AUTOMATIZADO PARA INVERNADERO BASADO EN  
 FPGA**

**TESIS**

Que como parte de los requisitos para obtener el grado de  
 Maestro en Ciencias

**Presenta:**  
 Ing. Miguel Ángel Ríos Alcaraz

**Dirigido por:**  
 Dr. Jesús Roberto Millán Almaraz

**SINODALES**

Dr. Jesús Roberto Millán Almaraz  
 Presidente

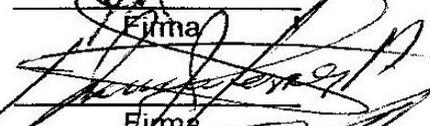
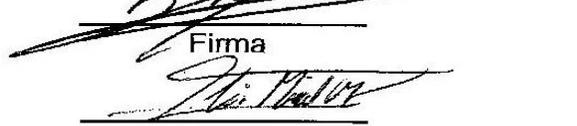
Dr. Irineo Torres Pacheco  
 Secretario

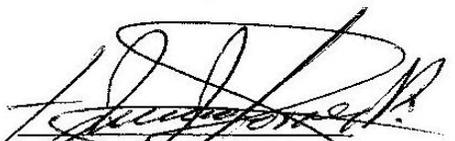
Dr. Ramón Gerardo Guevara González  
 Vocal

Dr. Enrique Rico García  
 Suplente

M.I. Luis Miguel Contreras Medina  
 Suplente

  
Dr. Aurelio Domínguez González  
 Director de la Facultad

  
 Firma  
  
 Firma  
  
 Firma  
  
 Firma  
  
 Firma

  
 Dr. Irineo Torres Pacheco  
 Director de Investigación y  
 Posgrado

Centro Universitario  
 Querétaro, Qro.  
 Mayo de 2012  
 México

## RESUMEN

Las plagas son un severo problema que ha enfrentado el hombre para producir cultivos de calidad a un bajo costo. Estos problemas generan grandes pérdidas económicas en la producción. Existen varios tipos de controles para el manejo integral de plagas, como lo son el control biológico (enemigos naturales) y el control químico (pesticidas y plaguicidas), siendo éste último dañino para el cultivo y el suelo, si no se tiene un buen manejo y control. Se propone una alternativa de solución en el cultivo bajo invernadero, controlando el acceso físico de manera automatizada, cumpliendo con protocolos de acceso y de salida. Mediante algunas medidas fitosanitarias se disminuye la presencia de parásitos y se mejora la exclusión e inocuidad. Se diseñó, implementó e instaló un sistema de acceso basado en plataforma FPGA por medio de periféricos de entradas y salidas, como lo son sensores y actuadores para la detección de medidas fitosanitarias y tomar acciones en concreto. La desinfección del calzado, el lavado de manos y la vestimenta apropiada son las condiciones que el usuario debe cumplir para acceder al invernadero. La identificación y autenticación del usuario para la entrada a la antesala, se realiza mediante radiofrecuencia (RFID). Se distribuyeron estratégicamente trampas amarillas en todo el invernadero para comparar la cuantificación de insectos atrapados entre ambos accesos. Se instaló una cortina de plástico a la mitad del invernadero de la Facultad de Ingeniería de forma simétrica para estar en igualdad de condiciones. De ésta manera se compara si el control de acceso automatizado es mejor que un control de acceso manual.

**Palabras clave:** Control de acceso físico, medidas fitosanitarias, FPGA, RFID, Plagas.

## SUMMARY

Pests are a serious problem faced by human beings in order to produce affordable and quality crops. These problems lead to great economic losses in production. There are several types of control for integrated pest management, such as biological control (natural enemies) and chemical control (pesticides and plaguicides), the latter being harmful to the crop and soil. The project proposes an alternative solution to protected agriculture, controlling the physical access in an automated manner, with an access and exit protocols. By some phytosanitaries measures, it reduces the presence of parasites and improves safety and exclusion. A FPGA-based access system was designed, deployed and installed through inputs and output peripherals, such as sensors and actuators in order to detect phytosanitaries measures based on concrete decisions. Desinfection of footwear, hand washing, and proper attire are conditions that users must comply in order to enter into the greenhouse. Identification and authentication is based on Radio Frequency (RFID). Yellow sticky cards were distributed throughout the greenhouse to compare the quantification of trapped insects in both accesses. A plastic curtain was installed at the half of the greenhouse at the Faculty of Engineering in order to be symmetrical equal. In this way, automated access control is compared against a manual access control, so that, it determines whose access control is the best.

**Key words:** Physical access control, phytosanitaries measures, FPGA, RFID, pests.

## **DEDICATORIAS**

### **A Dios:**

Por darme la salud para disfrutar, fuerza para no decaer y discernir entre lo bueno y lo malo. Así como las bendiciones necesarias para concluir una etapa más de mi vida.

### **A mis padres y hermanos:**

Por ser mi motivación, mi apoyo incondicional, por brindarme su amor que es de lo más preciado que tengo. Gracias por todos los cuidados que me han brindado y sobre todo, la paciencia que me han tenido.

### **A mi esposa Wendy e hijo Ángel:**

Gracias mi amor por estar conmigo en los buenos y malos momentos. Por compartir hermosas historias que hemos vivido, y que a pesar de las adversidades hemos salido adelante, mi amada esposa.

A ti mi bebe, te dedico ésta tesis. Primero, no sabrás ni de qué trata, pero espero, si así lo decides, cuando estés realizando tu licenciatura o posgrado te sirva de guía y motivación, ya que ustedes lo son para mí.

Los amo.

## AGRADECIMIENTOS

Al H. Consejo de Ciencia y Tecnología (CONACYT) por la oportunidad y el apoyo económico brindado durante éstos dos años de esfuerzo colectivo.

A la Universidad Autónoma de Querétaro por darme la oportunidad de realizar mis estudios de posgrado en la Maestría en Instrumentación y Control y al permitirme el uso de las instalaciones del invernadero de la Facultad de Ingeniería, donde se realizó ésta tesis.

Agradezco profundamente a mis padres, por ser el pilar que me sostiene. Por brindarme su apoyo incondicional. Por compartir momentos de alegría y tristeza. El distanciamiento es difícil, el calor de hogar no tiene comparación, gracias por soportar tanto tiempo sin estar a su lado. A ti Titita, gracias por ayudar a mis papás en los quehaceres de la casa, siempre me fue agradable llegar a Mazatlán y encontrar la casa arreglada. Ah y gracias por prestarme tu cama (aunque a veces me adueñaba, ni te pedía permiso =P).

A mi director de tesis, el Dr. Jesús Roberto Millán Almaraz, le agradezco su apoyo incondicional. Por brindarme sus conocimientos en electrónica, sobre todo, en diseño digital. Gracias por sus clases, por su paciencia, por su apoyo, por las pláticas que tuvimos... pero sobre todo, por ser buen amigo.

A mis sinodales, Dr. Irineo Torres Pacheco, Dr. Ramón Gerardo Guevara González, Dr. Enrique Rico García y M.I. Luis Miguel Contreras Medina, por las aportaciones a ésta tesis.

Citando la frase célebre de Cicerón, “Vivir sin amigos no es vivir”, les agradezco a mis amigos por brindarme su apoyo incondicional. A Carlos Duarte, Roberto Millán, Alejandro Romero y Jacky su esposa, Luis Contreras, Jimmy, Jaredt, Carlos González, Alejandro Vidales y su prometida Montserrat Campos, Arturo Fernández, Jaja y Ana, Rafa Muñoz, Kike y Karla su esposa. Gracias porque de ustedes aprendí mucho.

A ti Wendy, gracias por tenerme paciencia, por brindarme tu cariño, comprensión y confianza. Gracias por ayudarme en el invernadero, a pesar de tener tus ocupaciones, siempre estuviste dispuesta. El mapeo de cables en el cuartito quedó muy bien así como la distribución de las cartulinas. Gracias por ser mi esposa, y por ese bebé que ya viene en camino. 😊

A Gibrán y Rut, gracias por sus consejos, se optimizó el diseño de bloques en el FPGA, la propuesta de la lógica de las batas que me dieron fue muy buena por lo que se implementó. Mucha suerte con su posgrado, tienen calidad para la investigación.

# ÍNDICE

RESUMEN .....	i
SUMMARY .....	ii
DEDICATORIAS .....	iii
ÍNDICE .....	vi
ÍNDICE DE CUADROS.....	ix
ÍNDICE DE FIGURAS.....	x
1. INTRODUCCIÓN .....	1
2. OBJETIVOS .....	4
2.1. Objetivo General .....	4
2.2. Objetivos Particulares.....	4
3. HIPÓTESIS .....	5
4. REVISIÓN DE LITERATURA .....	5
4.1. Agricultura de Precisión .....	5
4.2. Automatización y Control .....	5
4.3. Control Biológico.....	6
4.4. Instrumentación y Control .....	7
4.5. Normas IP.....	8
4.6. Plataformas de Software para el Desarrollo de Interfaces Gráficas para Usuarios GUIs.....	10
4.7. Trampas para Insectos (Cartulinas Pegajosas de Colores) .....	11
4.8. Plagas.....	12
4.9. Plataformas Tecnológicas para Automatización.....	13
4.9.1. Arreglo de Compuertas Programables en Campo (FPGA).....	14
4.9.2. Microcontrolador MCU .....	15

4.9.3.	Procesador Digital de Señales (DSP) .....	15
4.10.	Redes de Sensores Inalámbricos .....	16
4.11.	Tecnología de Acceso para la Infraestructura .....	16
4.11.1.	Tipos de Acceso .....	17
4.11.1.1.	Control de Acceso Lógico.....	17
4.11.1.2.	Control de Acceso Físico.....	17
4.11.2.	Biometría.....	18
4.11.2.1.	Identificación por Huellas Dactilares .....	19
4.11.2.2.	Reconocimiento facial.....	20
4.11.3.	Sistemas de Identificación por Radio Frecuencia (RFID).....	20
4.11.4.	Identificación por Teclado Hexadecimal .....	21
5.	METODOLOGÍA .....	23
5.1.	Ambiente Experimental .....	23
5.2.	Instrumentación y Control .....	23
5.2.1.	Sensores.....	24
5.2.1.1.	Sensores Ópticos Infrarrojos.....	24
5.2.1.2.	Sensor Reflectivo QRD1114 .....	26
5.2.1.3.	Interruptores de Fin de Carrera .....	29
5.2.2.	Control de Acceso.....	30
5.3.	Protocolo de Acceso.....	31
5.3.1.	Bloque Registro de Usuario por RFID.....	32
5.3.2.	Bloque Protocolo de Acceso .....	34
5.4.	Protocolo de Salida .....	38
5.5.	Control de Acceso Automatizado para Invernadero basado en FPGA .....	39
5.6.	Modelo y Análisis Estadístico .....	41

5.7. Distribución de Trampas para Insectos .....	41
5.8. Captura de Insectos en Cartulinas Pegajosas .....	43
6. RESULTADOS Y DISCUSIONES .....	44
6.1. Etapa de Control.....	44
6.2. Etapa de Potencia .....	45
6.3. Fuente de Poder .....	45
6.4. Instalación de la Tarjeta de Desarrollo FPGA.....	46
6.5. Análisis de Varianza (ANOVA).....	47
6.5.1. Descripción de los Datos.....	49
6.5.2. Estadística de Prueba.....	51
6.6. Comparación de Insectos entre Accesos .....	51
7. CONCLUSIONES .....	55
8. BIBLIOGRAFÍA .....	57

## ÍNDICE DE CUADROS

Cuadro 4-1 Nivel de Protección contra el ingreso de objetos sólidos. ....	8
Cuadro 4-2 Nivel de Protección contra el ingreso de líquidos. ....	10
Cuadro 5-1 Formato de datos (ASCII) de salida del lector ID-12 Innovations. ....	31
Cuadro 5-2 Características Principales del RFID ID-12 Innovations. ....	31
Cuadro 6-1 Expresiones para el cálculo de la ANOVA de un factor ( $r$ indica el número de accesos y $N$ la cantidad de trampas de insectos). ....	49
Cuadro 6-2 Tabla de resultados de la cuantificación de tres especies en particular de insectos para el acceso manual, valor medio y total de tratamientos. ....	50
Cuadro 6-3 Tabla de resultados de la cuantificación de tres especies en particular de insectos para el acceso automatizado, valor medio y total de tratamientos. ....	51
Cuadro 6-4 Tabla ANOVA para los resultados de los cuadros 6-2 y 6-3. ....	51
Cuadro 6-5 Porcentajes de la captura de insectos por especie en el invernadero. ....	52

## ÍNDICE DE FIGURAS

Figura 4-1 Ejemplos de dispositivos biométricos para el reconocimiento de huellas digitales y rostro en 3D para el control de acceso físico.....	19
Figura 4-2 Arquitectura de un Teclado Hexadecimal en FPGA (Rios, 2010). .....	22
Figura 5-1 Vista isométrica del invernadero de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro. Sección A y Sección B. (Rios, 2009). .....	23
Figura 5-2 Etapa de Instrumentación y Control.....	24
Figura 5-3 Sensor Óptico Infrarrojo para la puerta exterior e interior. ....	25
Figura 5-4 Sensores ópticos Infrarrojos para la detección y desinfección del calzado. ....	26
Figura 5-5 Diagrama Sensor Óptico Infrarrojo. ....	26
Figura 5-6 Diagrama esquemático Sensor Reflectivo QRD1114. ....	27
Figura 5-7 Diagrama eléctrico para el acondicionamiento de señal para el sensor reflectivo QRD1114. ....	28
Figura 5-8 Instalación del sensor reflectivo en la tarjeta.....	28
Figura 5-9 Colocación de ganchos en los interruptores de fin de carrera para las batas de laboratorio. ....	29
Figura 5-10 Sistema de Identificación de Usuario por medio de Radiofrecuencia.....	30
Figura 5-11 Etapa de Protocolo de Acceso.....	32
Figura 5-12 Bloque Registro de Usuario mediante RFID.....	32
Figura 5-13 Bloque Protocolo de Acceso.....	35
Figura 5-14 Bloque Protocolo de Salida. ....	38
Figura 5-15 Bloque Control de Acceso Automatizado para Invernadero basado en FPGA. ....	40
Figura 5-16 Distribución de Trampas para Insectos en el Invernadero de la Facultad de Ingeniería. ..	42
Figura 5-17 Distribución de las trampas por surco en el invernadero. ....	43
Figura 5-18 Captura de insectos en las trampas amarillas. ....	44
Figura 6-1 Etapa de Control del sistema.....	45
Figura 6-2 Etapa de potencia del sistema.....	45
Figura 6-3 Fuentes de poder para la alimentación de periféricos. ....	46
Figura 6-4 Instalación del FPGA en la caja IP65.....	46

Figura 6-5 Comunicación del sistema de control de acceso automatizado con la PC.....	47
Figura 6-6 Sistema de Control de Acceso Automatizado para Invernadero basado en FPGA. ....	47
Figura 6-7 Trampa con mayor cantidad de mosquitas negras localizada en el acceso automatizado.	52
Figura 6-8 Comparación de medias de la captura de diferentes especies de insectos en ambos accesos físicos.....	53
Figura 6-9 Caja de valores extremos de la cuantificación de insectos. ....	54
Figura 6-10 Comparación de surcos entre el acceso automatizado y el acceso manual.....	55

## 1. INTRODUCCIÓN

La agricultura (del latín agri, 'campo o tierra de labranza' y cultura, 'cultivo, crianza') es el conjunto de técnicas y conocimientos para cultivar la tierra. Los avances de la ciencia y la tecnología en las primeras décadas del siglo XXI han impulsado todas las ramas del saber humano. La agricultura no es ni será ajena a estos acontecimientos, dado que a través del tiempo ha evolucionado desde el cultivo en cielo abierto hasta los modernos invernaderos de hoy en día (Tapia, 2006).

Se estima que la agricultura se ha desarrollado desde hace unos 8,000 a 10,000 años. Desde entonces, todos los pueblos de la Tierra han reconocido el valor que las plantas cultivadas tienen para la alimentación humana y de los animales domésticos. La agricultura es de vital importancia en la economía de las naciones; además de que las plantas y los procesos que las rigen gobiernan la vida sobre este planeta (Millán-Almaraz *et. al.*, 2009).

La agricultura protegida es la modificación del entorno natural para lograr el crecimiento óptimo de la planta. Las modificaciones pueden ser en ambiente aéreo o a nivel del suelo con el objetivo de incrementar el rendimiento del cultivo, extender la temporada de siembra y permitir el crecimiento de la planta durante periodos del año no comúnmente usados en cielo abierto. La agricultura protegida incluye sistemas integrales de control donde se engloban aspectos del medio natural que son modificados para un máximo crecimiento de la planta; a su vez son de bajo costo. El control se impone a variables como temperatura, luz, agua, humedad, dióxido de carbono (CO<sub>2</sub>) y nutrición.

Las instalaciones para la protección de cultivos pueden ser muy diversas entre sí por las características y complejidad de sus estructuras, así como por la mayor o menor capacidad de control ambiental. Una primera clasificación de los diversos tipos de protección, puede hacerse distinguiendo entre micro y macrotúneles, casas malla e invernaderos.

Los invernaderos son estructuras empleadas para proteger cultivos. Éstos permiten modificar y controlar de forma eficiente los principales factores ambientales que intervienen en el desarrollo y crecimiento de las especies vegetales. En el interior de un invernadero se producen microclimas artificiales ideales para aumentar los rendimientos agrícolas, sin importar las condiciones ambientales externas (Tapia, 2006).

El diseño de la estructura debe proporcionar protección contra el viento, lluvia, calor, frío, plagas y enfermedades. Las ventajas del empleo de invernaderos son: aumento en la calidad del cultivo, ahorro de agua y fertilizantes, mejor control de plagas y enfermedades, precocidad en los frutos, producción fuera de época, posibilidad de obtener más de un ciclo de cultivo, entre otras. Las desventajas del empleo de invernaderos es la alta inversión inicial, costos de operación elevados, además de que requiere personal especializado.

Uno de los mayores problemas en la agricultura es la pérdida en las producciones debido a las plagas. Las pérdidas por plagas a nivel mundial es del 10.2%, lo que representa 159 millardos de dólares (Agrios, 2005).

Actualmente los invernaderos sufren una serie de pérdidas por plagas, lo que provoca que la implementación y el desarrollo de técnicas alternativas para el control de plagas en invernaderos, sea una prioridad al instalar este tipo de infraestructura.

El objetivo de la exclusión es evitar el contacto de la plaga con el cultivo; éstas son las prácticas más caras y más difíciles de aplicar, pero son las más efectivas en el manejo de plagas y enfermedades.

Las plagas son el resultado de la actividad agrícola. El cultivo sirve de alimento para los insectos, los cuales se sienten atraídos a los cultivos, que seguramente son más atractivos por su excelente condición como resultado de los cuidados como la fertilización, el riego y el control climático que se les proporciona. Por lo tanto, es importante entender cómo se comportan las plagas, qué les gusta o

no les gusta, qué las atrae o repele, de manera que se pueda reducir su efecto sin tener que depender totalmente de controles biológicos y químicos.

La temprana detección de enfermedades es un reto que en la agricultura es difícil de vencer. El manejo integral de plagas combina métodos profilácticos, biológicos y físicos para enfrentar a bioagresores mientras se minimiza el uso de pesticidas (Boissard, 2008).

La mosca negra, mosquita blanca, áfidos y trips son bien conocidos como insectos dañinos causando severas pérdidas en el cultivo bajo invernadero. Como consecuencia, el manejo de plagas de insectos en invernadero es uno de los principales problemas en las prácticas agrícolas (Cho *et. al.*, 2007).

El artículo 2 de la Ley Federal de sanidad vegetal, SCJN DOF 18-11-2008, establece que la sanidad vegetal tiene como finalidad promover y vigilar las disposiciones fitosanitarias; diagnosticar y prevenir la diseminación e introducción de plagas de los vegetales, sus productos y subproductos; establecer medidas fitosanitarias; y regular la efectividad biológica, aplicación, uso y manejo de insumos, así como el desarrollo y prestación de actividades y servicios fitosanitarios.

Las medidas fitosanitarias tienen por objeto prevenir, confinar, excluir, combatir o erradicar las plagas que afectan a los vegetales, sus productos o subproductos, según lo establecido en el artículo 2 de la Ley Federal de Sanidad Vegetal, SCJN DOF 18-11-2008.

Existen dispositivos de control para la automatización de procesos utilizados en la industria e investigación. Algunos ejemplos son los microcontroladores, este dispositivo es un circuito integrado programable capaz de ejecutar instrucciones grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen con una tarea específica. Los procesadores digitales de señales (DSP), son dispositivos empleados la operación o transformación de una señal en un hardware digital según reglas bien definidas las cuales son introducidas al hardware a través de un software específico que puede o no manejar lenguajes tanto de alto como de

bajo nivel. Algunos ejemplos de aplicaciones de los DSPs son el manejo de imágenes digitales, reconocimiento de voz, reproductores digitales de audio, entre otras. Otros dispositivos son los arreglos de compuertas programables en campo (FPGA), Un FPGA es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada *'in situ'* mediante un lenguaje de descripción especializado. Las FPGAs tienen las ventajas de ser reprogramables (lo que añade una enorme flexibilidad al flujo de diseño), sus costes de desarrollo y adquisición son mucho menores para pequeñas cantidades de dispositivos y el tiempo de desarrollo es también menor.

Se propone utilizar la arquitectura FPGA para desarrollar los bloques de control adecuados para el cumplimiento de los protocolos de entrada y salida de la antesala del invernadero, utilizando los periféricos de entrada y salida, así como la capacidad de ejecutar procesos en paralelo que este dispositivo ofrece.

## **2. OBJETIVOS**

### **2.1. Objetivo General**

El objetivo general es desarrollar un sistema de control de acceso automatizado basado en tecnología FPGA y sistemas mecatrónicos para disminuir la presencia de insectos en invernaderos a un costo accesible.

### **2.2. Objetivos Particulares**

- Demostrar las ventajas de un sistema de control de acceso automatizado para disminuir la entrada de vectores en un invernadero y su implementación en cualquier invernadero.
- Desarrollar e instalar el sistema de control de acceso automatizado en un invernadero con un cultivo en crecimiento para comparar y cuantificar los insectos atrapados en trampas pegajosas amarillas.

### **3. HIPÓTESIS**

- El control de acceso automatizado basado en FPGA mediante identificación por Radio Frecuencia (RFID) disminuye la presencia de insectos en un invernadero.

### **4. REVISIÓN DE LITERATURA**

En este capítulo se describen los antecedentes para el desarrollo del proyecto. Se revisó la bibliografía reciente para el tipo de tecnología de acceso a la infraestructura. Se define las plataformas tecnológicas más utilizadas para la automatización. Los tipos de control utilizados para el manejo integral de plagas. Las características de las trampas amarillas para la captura de insectos. El software requerido para el desarrollo de interfaces gráficas de usuario y conceptos básicos sobre los temas de interés.

#### **4.1. Agricultura de Precisión**

La agricultura de precisión puede ser definida como el manejo de estrategias que utiliza sistemas informáticos para reunir información de múltiples fuentes de aplicación para influir en las decisiones relacionadas con el cultivo (Krishnan *et al.*, 2006; Clay y Shanahan, 2011). El concepto se basa en nuevas herramientas y fuentes de información proporcionada por modernas tecnologías, tales como dispositivos para el monitoreo de rendimientos, el suelo, las plantas, monitoreo de plagas y sensores remotos (Morais *et al.*, 2008).

La información es factible para los productores sólo si es oportuna, precisa, y de fácil acceso. El éxito en la agricultura de precisión está relacionado en que tan bien se puede valorar, manejar y aplicar el espacio-tiempo continuo en la producción del cultivo (Sparks, 1999).

#### **4.2. Automatización y Control**

La automatización consiste en incorporar a un proceso un conjunto de elementos y dispositivos tecnológicos que aseguren su control y buen comportamiento. Los objetivos principales de este proceso son: la reducción de costos de fabricación, calidad constante en los medios de producción y liberar al ser humano de las tareas tediosas, peligrosas e insalubres.

Un proceso automatizado asegura los parámetros apropiados para hacer mediciones, análisis de operaciones, mayor rentabilidad, cálculos en acciones de control. Además, el personal de la planta se mantiene informado sobre el proceso, sus capacidades y conocimientos son explotados; la detección de fallas se realiza con mayor facilidad (Samad, 2006).

#### **4.3. Control Biológico**

Las poblaciones de todos los organismos vivos son, en cierta medida, reducidas por la acción natural de sus depredadores, parásitos, antagonistas y enfermedades. Este proceso ha sido denominado como control natural, sin embargo, cuando se controlan las plagas, se le conoce como control biológico. Los agentes que ejercen el control biológico son frecuentemente llamados enemigos naturales.

El control biológico se ha definido en demasía, pero una definición comúnmente aceptada por los especialistas es la que se define a continuación: “El uso de organismos vivos para suprimir la población de una plaga en particular, haciéndola menos dañina o abundante de lo que podría ser” (Eilenberg *et al.*, 2001).

El uso del control biológico se incrementó debido a las necesidades prácticas para encontrar una solución a los problemas generados por plagas. Debido a que los pesticidas químicos no funcionaban o no eran apropiados para controlar cierto tipo de plagas. Otro gran impulso para la utilización del control biológico ha sido el hecho de que los plaguicidas químicos pueden causar efectos secundarios negativos, por consiguiente, ocasiona preocupación en la salud de los seres vivos y preservación en el medio ambiente. Los controles biológicos no dejan residuos químicos y suelen ser

bastante hospederos en comparación con los pesticidas químicos sintéticos (Hajek, 2004).

El control biológico es un método de control de plagas, enfermedades y malezas que consiste en utilizar organismos vivos para controlar las poblaciones de otro organismo. El control biológico en invernaderos es probable que aumente notablemente con el tiempo, a medida que los productores buscan una forma para reducir los efectos de resistencia a los plaguicidas, y cambiar restricciones en el manejo de plagas (Pilkington *et al.*, 2009).

Entre los métodos para el manejo integral de plagas (IPM), el control biológico es el más destacado ya que envuelve el uso de enemigos naturales como predadores, parasitoides y patógenos que son antagonistas de plagas de insectos y ácaros. El control biológico se ha expandido considerablemente, ya que es eficaz en el medio ambiente y ofrece resultados permanentes. El ambiente bajo invernadero, es susceptible al uso del control biológico, ya que son sistemas cerrados que proveen barreras contra la dispersión de plagas. Por otra parte, todo el sistema y la densidad de la plaga pueden ser controlados muy de cerca por lo que es menos vulnerable a invasiones de plagas comparado con los cultivos a cielo abierto (Perdikis *et al.*, 2008).

#### **4.4. Instrumentación y Control**

La instrumentación se ha desarrollado para la detección y medición de variables físicas que son esenciales para la industria, medio ambiente, investigación y desarrollo, transporte, equipo militar, y en nuestra vida diaria. Un conjunto de instrumentos forma un sistema de instrumentación, el cual es responsable de numerosas mediciones en un proceso complejo (Heren, 2006).

El control automático ha desempeñado una función vital en el avance de la ingeniería y la ciencia. Además de su extrema importancia en los sistemas de vehículos espaciales, de guiado de misiles, robóticos y similares; el control automático se ha vuelto una parte importante e integral de los procesos modernos

industriales y de manufactura. Por ejemplo, el control automático es esencial en el control numérico de las máquinas-herramienta de las industrias de manufactura, en el diseño de sistemas de pilotos automáticos en la industria aeroespacial, y en el diseño de automóviles y camiones en la industria automotriz. También es esencial en las operaciones industriales como el control de presión, temperatura, humedad, viscosidad y flujo en las industrias de proceso.

Debido a que los avances en la teoría y la práctica del control automático aportan los medios para obtener un desempeño óptimo de los sistemas dinámicos, mejorar la productividad, aligerar la carga de muchas operaciones manuales repetitivas y rutinarias, así como de otras actividades, casi todos los ingenieros y científicos deben tener un buen conocimiento de este campo (Ogata, 2001).

#### 4.5. Normas IP

El código IP (ANSI/IEC 60529-2004) consiste en las letras IP (International Protection Rating) seguidas por dos dígitos. El primer dígito numérico describe el nivel de protección contra el polvo, ya que el polvo no debe entrar bajo ninguna circunstancia (Cuadro 4-1).

Nivel	Tamaño del Objeto Entrante	Efectivo Contra
0	_____	Sin Protección
1	>50mm	El elemento que debe utilizarse para la prueba (esfera de 50 mm de diámetro) no debe entrar por completo.
2	>12.5 mm	El elemento que debe utilizarse para la prueba (esfera de 12,5 mm de diámetro) no debe entrar por completo.
3	>2.5 mm	El elemento que debe utilizarse para la prueba (esfera de 2,5 mm de diámetro) no debe entrar en lo más mínimo.
4	>1 mm	El elemento que debe utilizarse para la prueba (esfera de 1 mm de diámetro) no debe entrar en lo más mínimo.
5	Protección contra polvo	La entrada de polvo no puede evitarse, pero el mismo no debe entrar en una cantidad tal que interfiera con el correcto funcionamiento del equipamiento.
6	Protección fuerte contra polvo	El polvo no debe entrar bajo ninguna circunstancia.

**Cuadro 4-1** Nivel de Protección contra el ingreso de objetos sólidos.

El segundo dígito especifica el nivel de protección frente a líquidos (normalmente agua), para este caso en particular, el objeto debe resistir (sin filtración alguna) y obstaculizar el flujo de líquido hacia su interior (Cuadro 4-2).

<b>Nivel</b>	<b>Protección Contra</b>	<b>Método de Prueba</b>	<b>Resultados Esperados</b>
<b>0</b>	Sin protección	Ninguno	El agua entrará en el equipo.
<b>1</b>	Goteo de agua	Se coloca el equipo en su lugar de trabajo habitual.	No debe entrar el agua cuando se la deja caer, desde 200 mm de altura respecto del equipo, durante 10 minutos (a razón de 3-5 mm <sup>3</sup> por minuto)
<b>2</b>	Goteo de agua	Se coloca el equipo en su lugar de trabajo habitual.	No debe entrar el agua cuando de la deja caer, durante 10 minutos (a razón de 3-5 mm <sup>3</sup> por minuto). Dicha prueba se realizará cuatro veces a razón de una por cada giro de 15° tanto en sentido vertical como horizontal, partiendo cada vez de la posición normal de trabajo.
<b>3</b>	Agua nebulizada (spray)	Se coloca el equipo en su lugar de trabajo habitual.	No debe entrar el agua nebulizada en un ángulo de hasta 60° a derecha e izquierda de la vertical a un promedio de 10 litros por minuto y a una presión de 80-100kN/m <sup>2</sup> durante un tiempo que no sea menor a 5 minutos.
<b>4</b>	Chorros de agua	Se coloca el equipo en su lugar de trabajo habitual.	No debe entrar el agua arrojada desde cualquier ángulo a un promedio de 10 litros por minuto y a una presión de 80-100kN/m <sup>2</sup> durante un tiempo que no sea menor a 5 minutos.
<b>5</b>	Chorros de agua	Se coloca el equipo en su lugar de trabajo habitual.	No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión de 30kN/m <sup>2</sup> durante un tiempo que no sea menor a 3 minutos y a una distancia no menor de 3 metros.
<b>6</b>	Chorros muy potentes de agua	Se coloca el equipo en su lugar de trabajo habitual.	No debe entrar el agua arrojada a chorros (desde cualquier ángulo) por medio de una boquilla de 12,5 mm de diámetro, a un promedio de 100 litros por minuto y a una presión de 100kN/m <sup>2</sup> durante no menos de 3 minutos y a una distancia que no sea menor de 3 metros.
<b>7</b>	Inmersión completa de agua	El objeto debe soportar (sin filtración alguna) la inmersión completa a 1 metro durante 30 minutos.	No debe entrar agua.
<b>8</b>	Inmersión completa y continua en agua	El equipo eléctrico/electrónico debe soportar (sin filtración alguna) la inmersión completa y continua a la profundidad y durante el tiempo que especifique el fabricante del producto con el acuerdo del cliente, pero siempre que resulten condiciones más severas que las	No debe entrar agua.

		especificadas para el valor 7.	
--	--	--------------------------------	--

**Cuadro 4-2** Nivel de Protección contra el ingreso de líquidos.

En conclusión, se puede establecer que cuanto mayor sea el grado de protección IP, más protegido estará el equipo.

#### **4.6. Plataformas de Software para el Desarrollo de Interfaces Gráficas para Usuarios GUIs**

Algunas de las plataformas que se han utilizado para el desarrollo de interfaces gráficas de usuario son las Interfaces Programables Aplicables (API) de Windows, Guide de Matrix Laboratories (MATLAB), Java, Visual Basic entre otras. Entre ellas son similares, sin embargo, debido al nivel de abstracción y lenguaje, el procesamiento para ejecutar las instrucciones es afectado dependiendo de la aplicación.

Los especialistas en informática escriben instrucciones en diversos lenguajes de programación, algunos de los cuales comprende directamente la computadora, mientras que otros requieren pasos intermedios de traducción. En la actualidad se utilizan cientos de lenguajes de computación. Deitel (2008) los divide en tres tipos generales: Lenguajes Máquina, Lenguajes Ensambladores y Lenguajes de alto nivel.

Cualquier computadora puede entender de manera directa solo su propio lenguaje máquina, que es su lenguaje natural, y como tal, está definido por el diseño del hardware de dicha computadora. Por lo general, los lenguajes máquina consisten en cadenas de números que finalmente se reducen a unos y ceros, que instruyen a las computadoras para realizar sus operaciones más elementales, una a la vez.

Los programadores empezaron a utilizar abreviaturas del inglés para representar las operaciones elementales, lo que formó la base de los lenguajes ensambladores. Estos se desarrollaron para convertir los primeros lenguajes ensambladores a lenguaje máquina, a velocidad del microprocesador de la computadora.

Para agilizar el proceso de programación se desarrollaron los lenguajes de alto nivel, en donde podían escribirse instrucciones individuales para realizar tareas de mayor complejidad. Los programas traductores, denominados compiladores, convierten a lenguaje máquina los programas que están en lenguaje de alto nivel (Deitel, 2008).

Las presiones de salida al mercado y la proliferación de dispositivos incompatibles hacen que el diseño de software sea una tarea difícil para los desarrolladores de dispositivos de bajo consumo. Los desarrolladores de sistemas embebidos han adoptado a Java en los últimos años porque el lenguaje se abstrae del hardware subyacente, mejorando la portabilidad. Java es una buena opción para el desarrollo de sistemas embebidos por su capacidad para superar problemas en C y C++, como lo son protocolos de comunicación y portabilidad (Akira *et al.*, 2002).

#### **4.7. Trampas para Insectos (Cartulinas Pegajosas de Colores)**

Las cartulinas pegajosas son ampliamente usadas para el control y el manejo de varias plagas de insectos. Muchos estudios se han llevado a cabo sobre los factores que afectan a las trampas, por ejemplo, el tamaño de la cartulina, posicionamiento horizontal y vertical, ubicación de las trampas, entre otros. Existe escasa información acerca de las características del color, brillo, tono y longitud de onda de la cantidad de luz reflejada de las trampas (Durmusoglu *et al.*, 2009).

Las trampas para insectos se han estado utilizando para la captura de pulgones, mosquita blanca, minadores de hojas, entre otros, principalmente para llevar un control de éstos. Sin embargo, las trampas para insectos, (amarillas o azules) se han estado usando como un medio para suprimir poblaciones de insectos y han demostrado su eficacia contra la mosquita blanca en varios sistemas de cultivo (Gu, *et al.*, 2008).

En el diseño de una trampa para el monitoreo o trapeo masivo de plagas de insectos, a menudo es importante que estas variables visuales sean consideradas. Las trampas basadas en las características del color se han utilizado ampliamente en

el campo e invernaderos para una variedad de plagas. Las ventajas de las trampas que emplean señales visuales son que su presencia es instantáneamente establecida, que funcionan de forma independiente por el movimiento del aire, y generalmente, son eficaces desde la distancia y desde cualquier dirección, siempre que no haya obstáculos entre la trampa y el insecto (Blackmer *et al.*, 2008).

Solís *et al.* (2009) propusieron la aplicación de una técnica de visión por computadora para la identificación de la mosquita blanca (*Bemisia Tabaci Genn.*) donde el procesamiento digital de la imagen fue hecho segmentando los insectos capturados en las trampas. Así, fue posible cuantificar y clasificar por atributos geométricos como proyección de área, excentricidad y solidez del insecto segmentado. Por lo tanto, la cuantificación e identificación de insectos fue posible utilizando técnicas de visión por computadora que aprovechan la estructura de la mosquita blanca. Sin embargo, se sigue requiriendo diferenciar entre plagas que normalmente convergen con dos o más en el mismo cultivo.

Solís y colaboradores (2010) propusieron una solución donde integra una técnica de visión por computadora como lo es el algoritmo SIFT (Scale Invariant Feature Transform) al algoritmo LOSS (Solís *et al.*, 2009), de tal manera incrementando el potencial de detección de la plaga y permitiendo ser usado como una importante herramienta para la agricultura de precisión. Se identificó coleópteras (*Coleoptera:Chrysomelidae*), crisopas (*Lacewings spp.*), áfidos (*Aphis gossypii Genn.*), cigarrita verde (*Empoasca spp.*), trips (*Thrips tabaci L.*), y la mosquita blanca (*Bemisia tabaci Genn.*)

#### **4.8. Plagas**

Los seres humanos comparten el planeta con algunas 10 millones de especies de organismos. Cada especie se alimenta, crece y se reproduce de diferentes maneras y en diferentes lugares alrededor del mundo, sin embargo, prácticamente ninguna especie lo hace en aislamiento. Todas las especies están interconectadas

en cierta medida, con algunos organismos más dependientes de otros, especialmente los que se encuentran en lo más alto de la cadena alimenticia.

Los seres humanos tienen una posición dominante considerable en muchos ecosistemas y dependen de muchas otras especies para su alimentación y refugio. Sin embargo, los seres humanos también compiten con varios organismos y generalmente se piensa que estos competidores son las plagas.

Una plaga se puede definir como cualquier organismo que reduce la disponibilidad, la calidad o el valor de algunos recursos humanos (Flint & van den Bosch, 1981). La definición de plagas debe ser amplia, debido a la gran diversidad en que éstas afectan a los seres humanos. Los recursos de las plagas en cuestión pueden ser plantas o animales, que han sido destinados para la alimentación, fibra o recreación, por ejemplo, en mascotas (pulgas) o plantas para áreas naturales.

Los parásitos que se desean excluir son los insectos tipo voladores. Las dos plagas más problemáticas son la mosca blanca (*Bemisia Tabaci*) (Gould, 2008) y el trips (*Frankiniella occidentalis*), sobre todo por su actividad como vectores de los virus que provocan las mayores pérdidas. Ejemplos de estos virus son el virus del bronceado, el virus del bronceado del tomate (TSWV), que afecta de forma muy grave a cultivos como pimiento, tomate, lechuga, apio y alcachofa.

La mosca blanca se ha convertido en la plaga que más problemas causa en los cultivos hortícolas. En tomate, cucurbitáceas y judía por ser vector de varias enfermedades víricas; en pimiento por el daño directo y por 'la negrilla', causada por el micelio de los hongos *Cladosphaerospermum spp.* que crece sobre la maleza excretada por las larvas (Zanic *et al.*,2008).

Además, hay tipos de insectos no voladores que atacan a cultivos bajo invernadero como lo son el pulgón (*Aphis gossypil*), araña roja (*Tetranychus urticae*) y la araña blanca (*Polyphagotarsonemus latus*).

#### **4.9. Plataformas Tecnológicas para Automatización**

Se presentan las plataformas tecnológicas más utilizadas dentro de la automatización, como lo son los microcontroladores y microprocesadores, FPGA y DSP, se describen su funcionamiento así como algunas de las aplicaciones en la industria y vida cotidiana.

#### **4.9.1. Arreglo de Compuertas Programables en Campo (FPGA)**

Un FPGA es un arreglo bidimensional de bloques lógicos y flip-flops con interconexión electrónicamente programable entre ellos (Sulaiman *et. al.*, 2009).

Los FPGAs se basan en una cantidad muy grande de celdas lógicas, muy elementales, y una mucho mayor interconectividad que sus contrapartes. Permite un mucho menor consumo de potencia y un mayor nivel de integración. Este tipo de dispositivo puede contar con unidades que van desde las 50,000 hasta los 10'000,000 de compuertas lógicas equivalentes (Romero, 2007).

Recientemente, los FPGAs se han vuelto una solución alternativa para el diseño de sistemas digitales, donde previamente se encontraban dominados por sistemas de microprocesadores de propósito general. Los FPGAs basados en controladores, ofrecen ventajas como alta velocidad en el procesamiento, funcionalidad compleja y bajo consumo de energía. Estas características son atractivas en el punto de vista para el diseño de sistemas embebidos. Actualmente se ha reportado el uso de FPGA en sistemas con retroalimentación en control digital, como rodamientos magnéticos, inversores por modulación por ancho de pulso (PWM), motores de inducción, convertidores de ac/dc y controladores para variadores de velocidad. La implementación de controladores basados en FPGA no se ha enfocado en un uso óptimo de recursos en hardware. Estos diseños requieren un gran número de multiplicadores y sumadores, y no utilizan eficientemente las características de memoria del FPGA (Chan *et al.*, 2007).

Los FPGAs son ideales para realizar prototipos de diseños embebidos de manera rápida y para el desarrollo de nuevas arquitecturas para generar propiedad

intelectual. El propósito de un bloque lógico en una FPGA es proporcionar el cálculo básico y elementos de almacenamiento utilizados en los sistemas digitales.

La ventaja de un FPGA es la inmunidad a la obsolescencia, por lo que si la tarjeta se deja de fabricar, el código puede ser traspasado a otra tarjeta o familia haciendo cambios mínimos, por ejemplo en la asignación de pines y flanco de interruptores. Se pueden programar inalámbicamente, son reconfigurables; todo está integrado en un solo circuito (System On a Chip), su tamaño es reducido (en el orden de milímetros), son de bajo costo, entre otros.

Las principales “desventajas” de un FPGA son la cantidad de conocimiento que se requiere para manejarlo y el tiempo de desarrollo. Para desarrollar un núcleo, en específico, es necesario invertir tiempo, el cual se repone después. El código puede ser utilizado posteriormente para diferentes aplicaciones.

#### **4.9.2. Microcontrolador MCU**

Los microcontroladores son circuitos embebidos en una amplia gama de equipos y dispositivos que se encuentran en el hogar, industria, comunicaciones, y otras áreas que afectan nuestra vida cotidiana (Sugai *et al.*, 2009).

Un sistema con microcontrolador debe disponer de una memoria donde se almacena el programa que gobierna el funcionamiento del mismo que, una vez programado y configurado, sólo sirve para realizar la tarea asignada (Palacios, 2009).

#### **4.9.3. Procesador Digital de Señales (DSP)**

En 1972 aparece el primer microprocesador de propósito general y en 1980 el primer microprocesador especializado en el tratamiento de señal (DSP), diseñado para realizar eficientemente el cálculo reiterado de la combinación producto-acumulación (operación básica de la convolución). Al convertir la manipulación de las señales en una cuestión de cálculo numérico realizada en un computador, el DSP

pudo incorporar a su patrimonio todos los conocimientos matemáticos o de cualquier otra índole susceptibles de ser programados en un computador (Mariño *et al.*, 1995).

Un DSP que cuenta con un simple ciclo de multiplicación, es muy adecuado para aplicaciones donde se requiere de velocidad para ejecutar cálculos y para la optimización de algoritmos en procesamiento digital de señales (Ruzairi *et al.*, 2008).

El DSP ha facilitado, por ejemplo, que hoy en día pueda disponerse de servicios tales como la telefonía celular, las comunicaciones de datos vía red telefónica o la televisión digital de sistemas de conversión de texto a voz, o efectos musicales especiales que pueden convertir un salón en una catedral gótica o en el *Palau de la música*.

#### **4.10. Redes de Sensores Inalámbricos**

Los sensores enlazan el mundo físico con el digital mediante la captura de datos y revelando fenómenos del mundo real convirtiéndola en una forma que pueda ser procesado, almacenado, y actuar en consecuencia. Los sensores proporcionan un beneficio social enorme, ya que están integrados en numerosos dispositivos, máquinas y en el medio ambiente (Dargie, 2010). Pueden ayudar a evitar fallas catastróficas en la infraestructura, en la conservación de valiosos recursos naturales, mejora la seguridad, incrementa la productividad, y permite nuevas aplicaciones tales como domótica, robótica y agricultura de precisión (Ruíz 2009).

Una red de sensores inalámbricos consiste de un gran número de nodos interconectados a una corta distancia. Los sensores están densamente desplegados dentro del área de un fenómeno o lo más cercano posible (Akyildiz *et al.*, 2002). Una red de sensores inalámbrica (WSN) se puede aplicar a cualquier sistema, pero su versatilidad y flexibilidad requiere de una extensa investigación y desarrollo (Matese *et al.*, 2009).

#### **4.11. Tecnología de Acceso para la Infraestructura**

A continuación se describen los tipos de acceso y las tecnologías por las cuales el usuario es identificado para el acceso a una infraestructura y a la información. Se tienen la identificación por radio frecuencia (RFID), teclado matricial y biometría.

#### **4.11.1. Tipos de Acceso**

##### **4.11.1.1. Control de Acceso Lógico**

El control de acceso lógico se refiere a la protección del acceso a la información en PCs, redes, y telefonía móvil. Una variedad en tipos de etiquetas pueden ser utilizados para autenticar al usuario, por ejemplo, contraseñas, fichas, tarjetas o biometría. Estas etiquetas representan algo que el usuario ya conoce (una contraseña), que ya tiene (fichas y tarjetas), o una característica física (datos biométricos).

Un sistema de control de acceso lógico implementará un método para registrar y asociar las etiquetas con el usuario, para luego solicitar las etiquetas con el usuario para acceder a los recursos dependiendo de la aplicación (por ejemplo, sistemas de red, dispositivos o sistemas operativos).

El sistema de control de acceso lógico puede registrar todos los intentos de acceso para la auditoría y cuando alguien tiene acceso a un recurso específico. Algunas aplicaciones que utilizan el control de acceso lógico es el área financiera, ya que requiere estándares de alta seguridad para el acceso a datos financieros y confidenciales. Otra área es la salud, aunque los reglamentos no especifican el uso de la biometría de forma explícita, se requiere que el acceso a los datos de historial clínico debe ser restringido a través de una fuerte autenticación de usuarios (Li y Jain, 2009).

##### **4.11.1.2. Control de Acceso Físico**

El uso de la biometría en sistemas de control de acceso físico es uno de los sectores más comercializados dentro de esta área, fuera de las aplicaciones forenses. Un problema para la integración exitosa de la biometría dentro sistemas de control de acceso físico es la interfaz entre los datos biológicos y el control de acceso a la infraestructura. Por esta razón, el sistema biométrico debe ser diseñado para interactuar apropiadamente con una amplia gama de sistemas de control de acceso. Además, la demanda de uso de un sistema de control de acceso físico es significativa. Los tipos de control de acceso más significativos implementados en biometría son las huellas digitales, geometría de manos, cara e iris (Li y Jain, 2009).

#### **4.11.2. Biometría**

El término biometría se aplica de forma general a la ciencia que se dedica al estudio estadístico de las características cuantitativas de los seres vivos: peso, longitud, etc. Sin embargo, en épocas más recientes, este término se utiliza también para referirse a los métodos automáticos que analizan determinadas características humanas con el fin de identificar y autenticar a individuos (Nanavati *et al.*, 2002).

En función de las características usadas para la identificación se pueden establecer dos tipos, aspectos físicos y vinculados con la conducta. Actualmente se habla de biometría estática para referirse al estudio del conjunto de características físicas y de biometría dinámica para referirse a las conductuales.

Dentro de la biometría estática nos encontramos entre otras con las siguientes características: Huella dactilar, característica del ojo (retina e iris), rayas y geometría de la mano, poros de la piel, característica estática de la cara, composición química del olor corporal, venas de muñecas y manos.

Dentro de la biometría dinámica se encuentran las siguientes características: Escritura, voz y gesto y movimiento corporal.

Actualmente la autenticación de usuarios por biometría es un importante campo para la seguridad. Se basa en las características físicas y biológicas o en el

comportamiento de la persona. El propósito de un sistema biométrico genérico es el de identificar y/o verificar la identidad de una persona basado en por lo menos una modalidad biométrica (huellas digitales, iris, voz, etc.) (Kumel *et al.*, 2011).

#### **4.11.2.1. Identificación por Huellas Dactilares**

Como definición en el aspecto de la identificación, las huellas dactilares son el rasgo biométrico que cumple con las características de universalidad, ya que todas las personas deben poseerlas, unicidad; no debe haber más de dos personas con los rasgos seleccionados iguales, permanencia; deben ser invariantes con el tiempo y cuantificación, ya que admiten medidas cuantitativas.

El método de identificación mediante huellas dactilares es uno de los métodos más fiables que actualmente se conocen y en pocos años posiblemente se incorporará en nuestra vida diaria. En estos últimos años la biometría dactilar se ha acercado al público en general y existen varias aplicaciones en la utilización de detectores de huella dactilar para el acceso. En los ordenadores portátiles se incluyen detectores de huella dactilar para que un usuario previamente registrado pueda iniciar una sesión. Algunos dispositivos de identificación por huellas dactilares y facial se muestran en la Figura 4-1.



**Figura 4-1** Ejemplos de dispositivos biométricos para el reconocimiento de huellas digitales y rostro en 3D para el control de acceso físico.

Varios algoritmos se han propuesto para la extracción de minucias, como el algoritmo propuesto por (Maio *et al.*, 1997) en su trabajo: Minucias detectadas por huellas dactilares por medio de escala directa de grises (*Direct Gray-Scale Minutiae*

*Detection in Fingerprints*) utilizando técnicas matemáticas de procesamiento digital de señales como la Transformada Discreta de Fourier (DFT).

#### **4.11.2.2. Reconocimiento facial**

El reconocimiento facial se ha convertido en un enfoque cada vez más popular en una amplia variedad de aplicaciones. Es uno de los pocos enfoques fisiológicos que puede ser directamente aplicado de forma transparente. La razón es que el sensor (dispositivo biométrico para el reconocimiento facial) es capaz de capturar una muestra sin que el usuario necesite específicamente proveer de una muestra al sistema, ya en otros casos como el reconocimiento de huellas dactilares, patrones vasculares, y geometría de manos, el usuario debe tocar o poner su mano sobre el sensor.

Actualmente, el reconocimiento facial se ha centrado en entornos bien definidos, con controles o restricciones en iluminación, orientación facial y distancia entre el usuario y el sistema de reconocimiento. La habilidad del sensor, una cámara en este caso, provee una mejor muestra que una imagen o foto del usuario. Sin alguna instrucción o interfaz de usuario, el sensor es necesario para capturar la imagen de un rostro con suficiente claridad, de manera que un algoritmo de reconocimiento facial pueda ser ejecutado (Clarke, 2011).

(Cadoni *et al.*, 2011) propusieron un método para el reconocimiento facial usando un algoritmo basado en características faciales de extracción estable en 3D caracterizando la cara y la posterior construcción para su autenticación.

#### **4.11.3. Sistemas de Identificación por Radio Frecuencia (RFID)**

Un sistema RFID tiene 3 componentes básicos: un tag o transponder, un lector y un equipo de procesamiento de datos (Want, 2006). El tag contiene información de identificación única del objeto al cual es asignado; el lector emite y recibe ondas de radio para leer la información almacenada en el tag, y el equipo de procesamiento adquiere los datos colectados (Wu, 2006; Van Deursen, 2009).

Un sistema RFID se puede dividir en 2 clases: activo y pasivo. Los tags activos requieren de una fuente de alimentación. Pueden ser conectados a la toma de alimentación de una infraestructura o usar energía almacenada como lo es una batería la cual está almacenada en el dispositivo. Sin embargo, las baterías hacen que el costo, tamaño y vida útil de un tag activo sea impráctico.

Un tag pasivo consiste en 3 partes: una antena, un chip semiconductor adjuntado a la antena y el encapsulado. El tag lector es el encargado de energizar al tag. La antena captura la energía y la transfiere al tag pasivo o de Identificación de Usuario (ID), el chip semiconductor se encarga de coordinar este proceso. La encapsulación se encarga de mantener la integridad del tag además de proteger a la antena y el chip de condiciones ambientales adversas y reactivas.

Algunas de las aplicaciones de estos dispositivos son en el área de supermercados, gestión y logística de procesos, base de datos de personal en empresas, escuelas, bibliotecas, entre otras (Lee, 2008; Ferrer, 2009).

Una de las aplicaciones de RFID en la agricultura de precisión es en viñedos, (Luvisi *et al.*, 2010) marcaron plantas modificadas para obtener datos a lo largo de su vida. La información fue enviada hacia un servidor web para acceder fácilmente a una base de datos, garantizando al usuario información en línea de cada planta modificada.

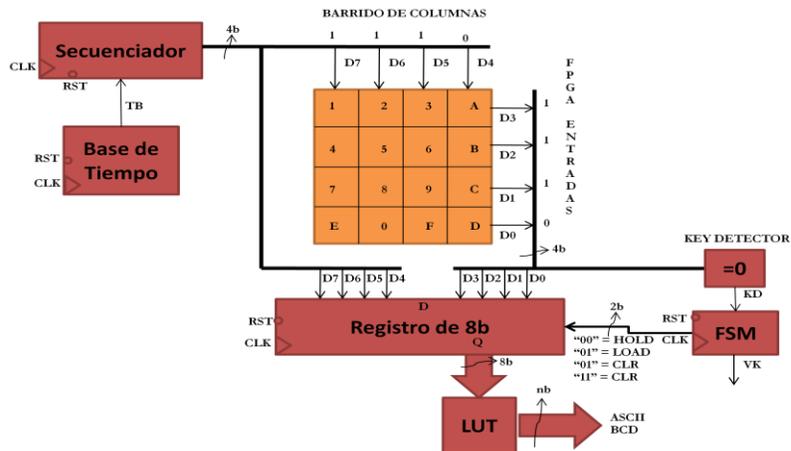
#### **4.11.4. Identificación por Teclado Hexadecimal**

Un teclado matricial está compuesto por una matriz de pulsadores en filas y columnas con el objetivo de reducir el número de líneas necesarias para su conexión (Palacios, 2009).

Las aplicaciones para este dispositivo son bastas, lo que se adecúa para el proyecto es la cerradura electrónica. Por medio de una serie de códigos guardados en un registro se identifica al individuo. El código puede estar dado por contraseña al

conectar una pantalla LCD de 2X16. Se muestra en la pantalla en forma de asteriscos la clave o el número de serie de la persona a autenticar.

En la Figura 4-2 se muestra la arquitectura de un teclado matricial desarrollado en FPGA. Se muestran los bloques por los cuales se guarda en un registro la tecla pulsada por el usuario.



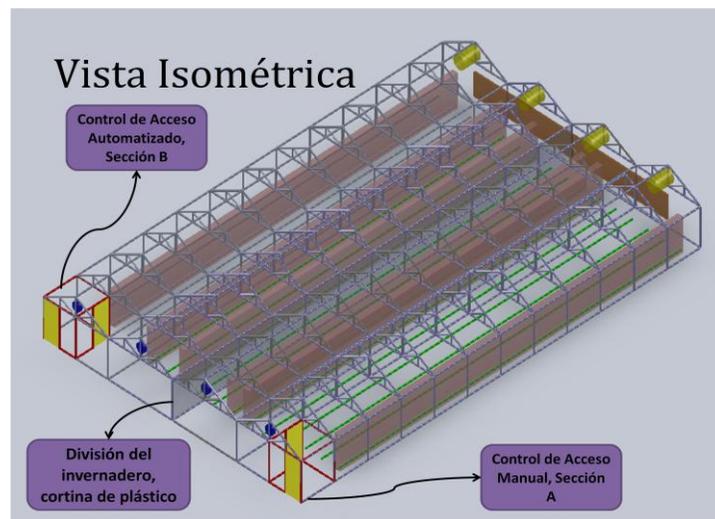
**Figura 4-2** Arquitectura de un Teclado Hexadecimal en FPGA (Rios, 2010).

Una base de tiempo de 30ms es la que realiza un barrido en las columnas activando un secuenciador de 4 bits. La lógica es negativa, es decir, se interpreta el valor 1 con los estados bajos de voltaje, un cero en las columnas del teclado y los 3 bits restantes a uno; al combinar con el vector de bits de los renglones (entradas al FPGA) se forma un vector de 8 bits que se guarda en un registro. Si se presiona una tecla en las columnas del teclado, un comparador se encarga de detectar la tecla y realizar el control por medio de una máquina de estados. La salida del registro de la tecla presionada entra a una tabla de vectores (LUT) la cual representa la codificación a un dispositivo para el despliegue de la tecla (pantalla LCD, displays de 7 segmentos, LEDs, entre otros).

## 5. METODOLOGÍA

### 5.1. Ambiente Experimental

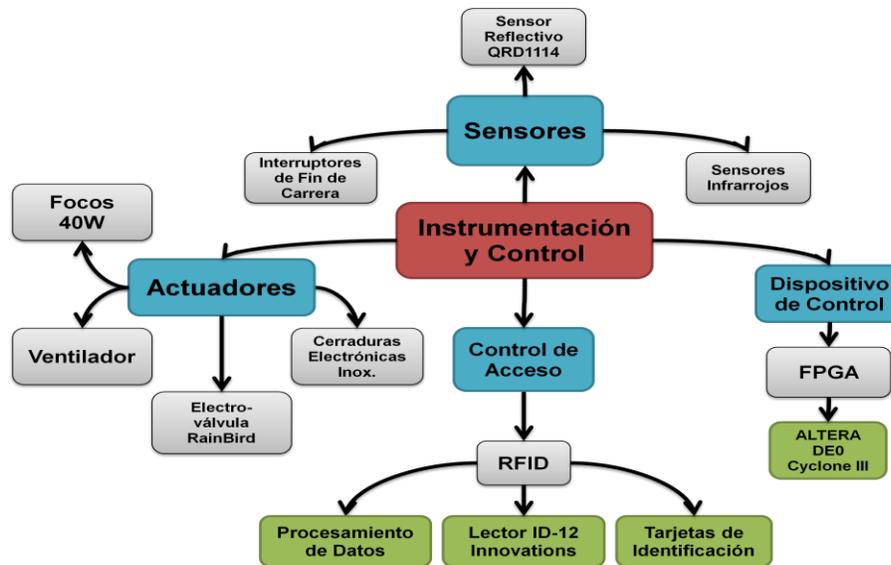
El proyecto se realizó en el invernadero de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro Campus Santiago de Querétaro. Sus dimensiones son de 37.2m X 24m. El invernadero se dividió en dos secciones instalando una cortina de plástico simétricamente. En la Figura 5-1, se identifica el acceso manual como Sección A y el control de acceso automatizado como Sección B. Esto, con el objetivo de comparar la cuantificación de insectos atrapados en las cartulinas en ambos accesos mediante análisis estadístico, ubicadas en lugares estratégicos.



**Figura 5-1** Vista isométrica del invernadero de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro. Sección A y Sección B. (Rios, 2009).

### 5.2. Instrumentación y Control

La Figura 5-2 representa la etapa de Instrumentación y Control, la cual se subdivide en cuatro áreas: sensores, actuadores, control de acceso y dispositivo de control.



**Figura 5-2** Etapa de Instrumentación y Control.

Las acciones de control de los actuadores están en función de los sensores antes mencionados. Se utiliza un ventilador, una electroválvula de la marca RainBird con alimentación a 24VAC a 50/60Hz y dos cerraduras electrónicas: una de acero inoxidable para la puerta exterior, y una de acero para la puerta interior; las dos cerraduras son de accionamiento a 12VDC.

### 5.2.1. Sensores

Un sensor es comúnmente definido como un dispositivo que recibe y responde a una señal o estímulo. Los sensores deben hablar el mismo lenguaje que los dispositivos a los cuales se comunican en un sistema en particular (Pallas *et al.*, 2001).

Los sensores que se implementaron en el cuarto para el control de acceso automatizado para el invernadero son sensores ópticos infrarrojos, un sensor reflectivo QRD1114 e interruptores de fin de carrera.

#### 5.2.1.1. Sensores Ópticos Infrarrojos

Los sensores ópticos infrarrojos están conformados por un fotodiodo y un fototransistor. Se usaron cuatro sensores ópticos los cuales se distribuyeron de la siguiente manera. Se instalaron dos sensores para las puertas exterior e interior respectivamente, en la parte superior por abajo del marco de cada puerta, como se muestra en la Figura 5-3.

Los sensores fueron atornillados por abrazaderas metálicas, con el propósito de fijarse y evitar movimientos indeseados. En cada puerta, en la parte superior se atornilló una barra de plástico color blanca para interrumpir el haz de luz de los sensores. De esta manera, se detecta el estado actual de cada puerta.



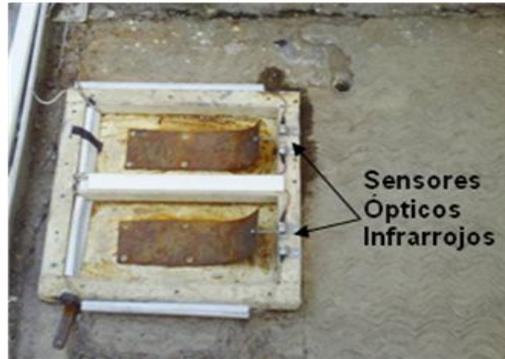
**Figura 5-3** Sensor Óptico Infrarrojo para la puerta exterior e interior.

Los dos sensores restantes se utilizaron para detectar si el usuario se desinfectó el calzado. Se armó una base con perfil tubular rectangular (PTR) de dimensiones 0.5m x 0.5m con una división a la mitad para seccionar ambos pies como se observa en la Figura 5-4. Se atornilló a la base una lámina de acero doblada 3/4 hacia adentro para emular el movimiento de los pies.

En el plano horizontal, se fijó un tornillo a la mitad de la lámina de acero. Cuando el usuario presione la lámina, ésta tenderá a doblarse hacia afuera y regresarse, por lo que el tornillo tocará la base de los pies donde estarán instalados ambos sensores, interrumpiendo el haz de luz y detectando ambos pies.

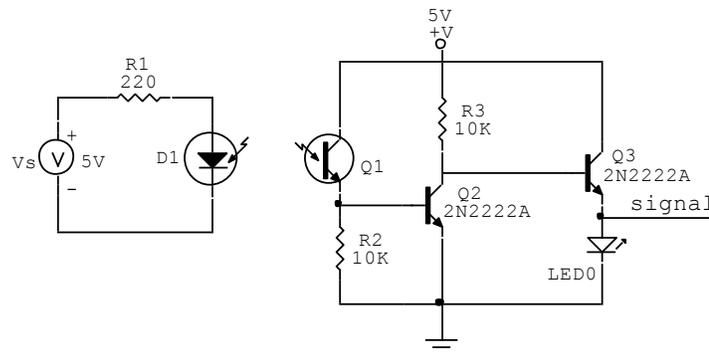
Los sensores de la base de los pies se instalaron de la misma forma que los sensores de las puertas. En la base de los pies, la sustancia por la cual se desinfectó

es por sales cuaternarias y cloro a un nivel considerable en función de la suela del calzado.



**Figura 5-4** Sensores ópticos Infrarrojos para la detección y desinfección del calzado.

Se alimenta al fotodiodo por medio de una fuente de 5V pasando una corriente de 25mA por medio de R1 a 220Ω. Esto permite generar el haz de luz imperceptible para el ojo humano, ya que su longitud de onda se encuentra en el infrarrojo. El fototransistor detecta el infrarrojo, por lo que su base es activada. Sin embargo, la señal es débil, no es suficiente para activar algún dispositivo de control, por lo que es amplificada por dos transistores NPN 2N2222A configurados en emisor común como se muestra en la Figura 5-5.



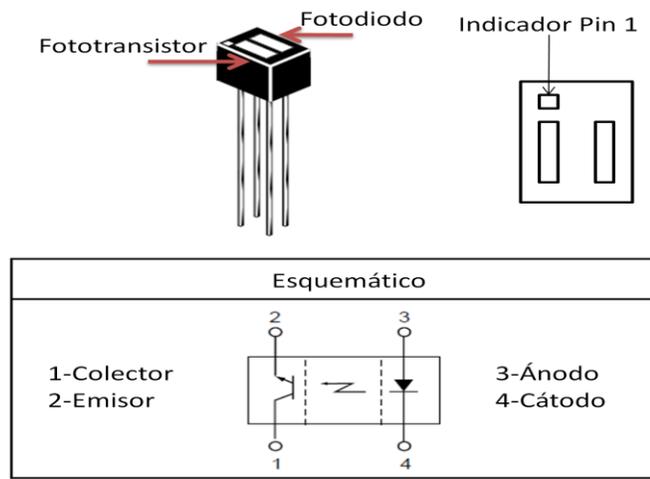
**Figura 5-5** Diagrama Sensor Óptico Infrarrojo.

#### 5.2.1.2. Sensor Reflectivo QRD1114

El lavado de manos se detecta mediante un sensor reflectivo fabricado por Fairchild Semiconductor con número de parte QRD1114. Las características de éste

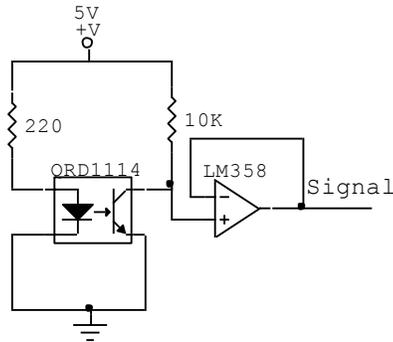
sensor es que tiene un diodo emisor infrarrojo junto con un fototransistor, ambos en un solo encapsulado, para detectar señales reflejadas. El rango para la detección de objetos del QRD1114 es de 0.5 a 1cm.

La Figura 5-6 representa el mapeo de pines para el sensor reflectivo donde la muesca indica el pin 1 que marca al colector del fototransistor, el pin 2 indica el emisor. Por otra parte, el pin 3 marca el ánodo y el pin 4 el cátodo del fotodiodo.



**Figura 5-6** Diagrama esquemático Sensor Reflectivo QRD1114.

La Figura 5-7 muestra el diagrama eléctrico para el acondicionamiento de señal de un sensor reflectivo. El fotodiodo se energiza por medio de una fuente de 5V pasando una corriente de 25mA por medio de una resistencia a 220Ω. Esto permite generar un haz de luz imperceptible para el ojo humano. El fototransistor detecta el infrarrojo, por lo que su base es activada. Sin embargo, la señal es débil. Ésta señal se aumenta mediante un amplificador operacional con número de parte LM358 configurado como seguidor de voltaje por lo que a su salida es aproximadamente de 5V enviando la señal de controla hacia el FPGA.



**Figura 5-7** Diagrama eléctrico para el acondicionamiento de señal para el sensor reflectivo QRD1114.

La Figura 5-8 muestra el sensor reflectivo instalado en la tarja. El sensor se fijó adentro de la tarja y simétricamente debajo del grifo; se barrenó con un taladro a un diámetro mayor al encapsulado del sensor con el objeto de tener una mayor libertad de movimientos para su acomodo. Una de las llaves se fijó en abertura para que el control del flujo del agua sea controlada mediante el sensor reflectivo activando y desactivando la electroválvula.



**Figura 5-8** Instalación del sensor reflectivo en la tarja.

Se enrolló parafilm al sensor; que es un material permeable con características especiales de transporte de vapor de agua y es resistente a la mayoría de reactivos comunes brindando una superficie hidrofóbica perfecta para controlar la entrada del fluido, su flexibilidad permite adaptarse a las superficies irregulares y complejas, pudiendo ser estirada 3 o 4 veces su longitud original antes

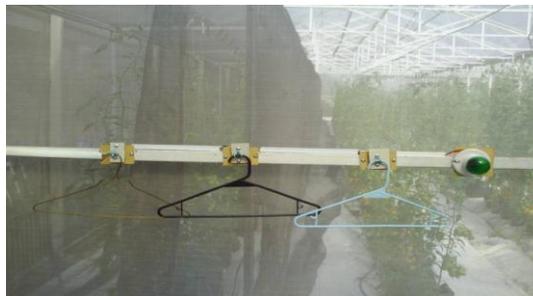
de romperse. Esto, para evitar que las salpicaduras de agua que puedan entrar al sensor cuando el usuario se lave las manos.

### **5.2.1.3. Interruptores de Fin de Carrera**

Dentro de los componentes electrónicos, se encuentra el final de carrera o sensor de contacto, son dispositivos eléctricos, neumáticos o mecánicos situados al final del recorrido de un elemento móvil, como por ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito. Internamente pueden contener interruptores normalmente abiertos (NA o NO en inglés), cerrados (NC) o conmutadores dependiendo de la operación que cumplan al ser accionados, de ahí la gran variedad de finales de carrera que existen en mercado.

Entre las ventajas encontramos la facilidad en la instalación, la robustez del sistema, es insensible a estados transitorios, trabaja a tensiones altas, debido a la inexistencia de imanes es inmune a la electricidad estática. Los inconvenientes de este dispositivo son la velocidad de detección y la posibilidad de rebotes en el contacto, además depende de la fuerza de actuación.

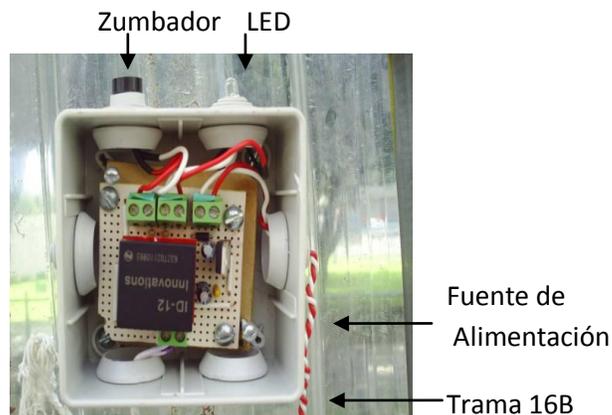
Los interruptores de fin de carrera se utilizaron para detectar la vestimenta del usuario. Se realizó un arreglo mecánico para la instalación en el invernadero, de tal forma que los ganchos pudieran ser colgados en sus respectivas argollas como se observa en la Figura 5-9.



**Figura 5-9** Colocación de ganchos en los interruptores de fin de carrera para las batas de laboratorio.

### 5.2.2. Control de Acceso

En la Figura 5-10 se muestra el sistema de RFID instalado adentro de la antesala del invernadero. Las dimensiones del lector RFID son de 26 mm x 25 mm x 7 mm, por lo tanto, se adecúa a las dimensiones de una caja eléctrica de 90 mm x 90 mm x 50 mm con norma IP44 mencionadas en el capítulo 4.



**Figura 5-10** Sistema de Identificación de Usuario por medio de Radiofrecuencia.

El lector de RFID se energiza por una fuente de 5v mediante un regulador de voltaje 7805. El voltaje de entrada para el regulador se recibe desde el riel de voltaje de 18v por medio de un transformador que a su salida manda 24v RMS a 5A. Cuando el lector detecta una tarjeta, éste manda un pulso por medio del pin 10 llamado BEEPER/LED, el cual, energiza a la base de un transistor, en este caso un 2N2222, para activar a un LED (Light Emisor Diode) y a un zumbador para la indicación visual y auditiva como señalización para la autenticación y detección del usuario.

La salida de los datos que envía el lector ID-12 Innovations está estructurada en formato ASCII de la siguiente manera. Primero se manda un byte de inicialización, en este caso es un 02h (la letra h significa que se encuentra en base 16 o hexadecimal). Seguido se envían diez bytes de datos, los cuales conforman el código de autenticidad y unicidad de la tarjeta, ya que estos diez bytes difieren entre cada tarjeta, por lo tanto siempre serán diferentes. Posteriormente, se realiza una

operación lógica OREX de los diez bytes de datos, seguidos de 2 bytes para el retorno de carro y finalmente un byte de fin de transmisión. El formato se muestra en el Cuadro 5-1.

<b>STX (02h)</b>	<b>DATA (10 ASCII)</b>	<b>CHECK SUM (2 ASCII)</b>	<b>CR</b>	<b>LF</b>	<b>ETX (03h)</b>
------------------	------------------------	----------------------------	-----------	-----------	------------------

**Cuadro 5-1** Formato de datos (ASCII) de salida del lector ID-12 Innovations.

En el cuadro 5-2 se resaltan las características más relevantes de éste dispositivo.

<b>Rango de lectura</b>	12 cm.
<b>Dimensiones</b>	26 mm x 25 mm x 7 mm.
<b>Frecuencia</b>	125 KHz.
<b>Formato de la tarjeta</b>	EM4001 o compatible.
<b>Codificación</b>	Manchester 64 bits, módulo 64.
<b>Voltaje</b>	5V @ 30mA nominal.
<b>Rango de voltaje</b>	4.6V hasta 5.4V.
<b>Interfaz serial</b>	9600bps, 8, N, 1, TTL.

**Cuadro 5-2** Características Principales del RFID ID-12 Innovations.

### 5.3. Protocolo de Acceso

El protocolo de acceso consiste en mejorar la exclusión cumpliendo con medidas fitosanitarias dentro de la antesala del invernadero. Los pasos para cumplir con el protocolo de acceso se muestran en la Figura 5-11. En primera instancia se inicia con la identificación del usuario, seguido de la exclusión, la cual se realiza mediante la disminución de la entrada de patógenos hacia la antesala. Posteriormente se realizan las detecciones de las medidas fitosanitarias; siendo éstas la desinfección del calzado, el lavado de manos y vestimenta apropiada. Finalmente se activa el acceso hacia el invernadero.

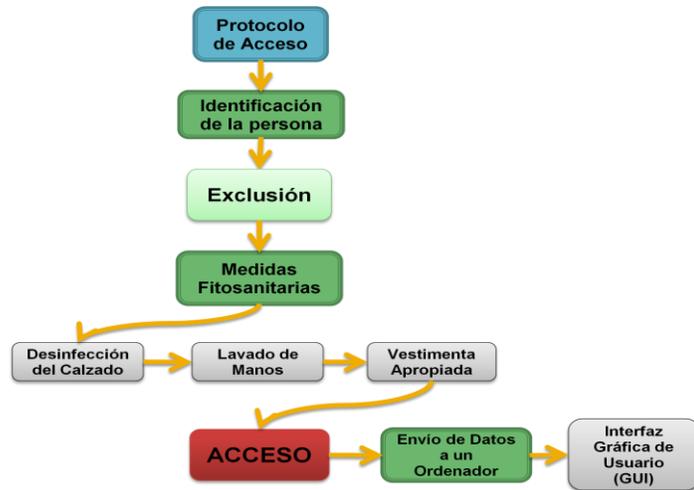


Figura 5-11 Etapa de Protocolo de Acceso.

### 5.3.1. Bloque Registro de Usuario por RFID

El proceso se lleva a cabo mediante la autenticación del usuario por medio de RFID. En la Figura 5-12 se diseñó la arquitectura del registro de usuario mediante RFID. Este bloque se subdivide en diversos módulos los cuales realizan diferentes acciones para guardar el código de la tarjeta de RFID.

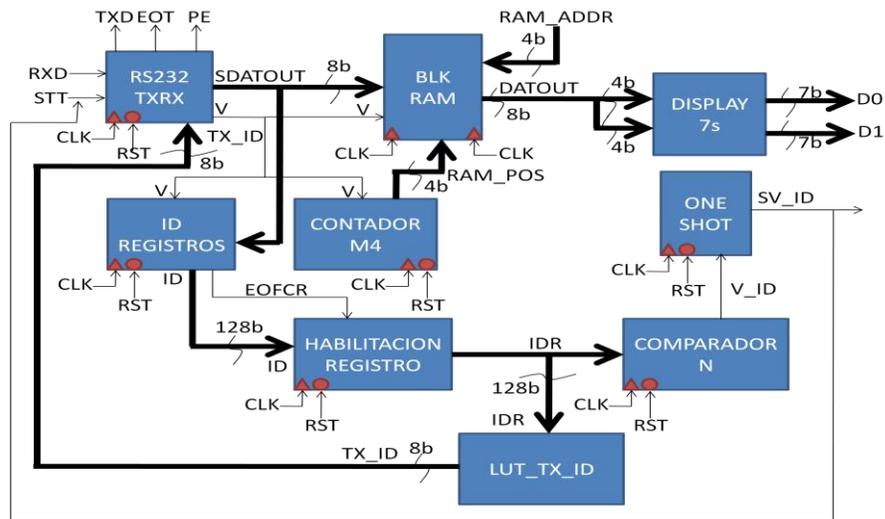


Figura 5-12 Bloque Registro de Usuario mediante RFID.

El bloque RS232\_TXRX captura los datos en tramas 8 bits enviados por el lector a una velocidad de transmisión de 9600bps por medio de la entrada RXD. Si la

transmisión es correcta, el bloque manda una bandera de válido por la señal de salida V y la trama de 8 bits generada por el lector SDATOUT. Esta señal a su vez, activa al módulo BLK\_RAM que consiste en una memoria RAM de doble puerto y se guarda el vector SDATOUT de forma secuencial.

Esta memoria contiene 16 posiciones debido al tamaño del vector generado por el código de la tarjeta RFID (128 bits). Para que la memoria almacene cada byte en su debida posición, se diseñó un bloque llamado CONTADOR\_M4, el cual consiste en una máquina secuencial de 16 estados. A este bloque también tiene como entrada la señal V generada por el bloque de recepción. Por lo tanto, cada vez que el transmisor envíe la bandera de válido, cambiará de posición en la memoria RAM en función del CONTADOR\_M4 guardando cada trama en secuencia.

Esto permite generar registros de forma secuencial con desplazamiento hacia la derecha, con el objetivo de concatenar las 16 tramas enviadas por el lector. Esta función lo realiza el bloque llamado ID\_REGISTROS. Este módulo contiene 16 registros interconectados en serie, esto con el objetivo ordenar las tramas de mayor a menor. Esto significa que la primera trama que llegó será la última en acomodarse. El control de los registros se realiza mediante una máquina de estados. Una vez que la máquina termina, envía una bandera de fin de transmisión llamada EOFER.

El módulo forma un vector de 128 bits llamado ID que fue generado por los 16 bytes del lector, lo que representa el código de autenticidad e identificación de la tarjeta que pertenece al usuario.

Posteriormente, se habilita un registro, se etiquetó con el nombre de HABILITACION\_REGISTRO, para guardar el vector ID con el objetivo de comparar con el estado actual del vector del módulo de COMPARADOR\_N. La salida del registro es identificada como IDR que es el mismo vector ID pero almacenado en una señal para no perder el código de autenticidad.

El módulo de COMPARADOR\_N es un bloque formado por una máquina de estados, el cual contiene el código de autenticidad de 6 tarjetas. Estos vectores se

guardan en señales, las cuales son susceptibles al cambio dentro de la máquina de estados, esto quiere decir si el vector que proporcionó el usuario es igual a cualquier señal de la tabla de códigos de la tarjeta (¿IDBD=IDR?). El COMPARADOR\_N envía una bandera de válido etiquetada como V\_ID, esta señal activa a un módulo llamado ONE\_SHOT, que solo manda un pulso de manera que la señal V\_ID no permanezca en un estado de nivel alto o un 1 lógico.

Para verificar si las tramas enviadas por el lector son correctas, se tienen 4 bits para generar las 16 combinaciones posibles. Éstas son asignadas a 4 botones de desplazamiento (Slide Switches) del FPGA, etiquetados como RAM\_ADDR. Si las tramas coinciden con el cuadro 5-1, el direccionamiento de las posiciones de la memoria RAM fueron las adecuadas.

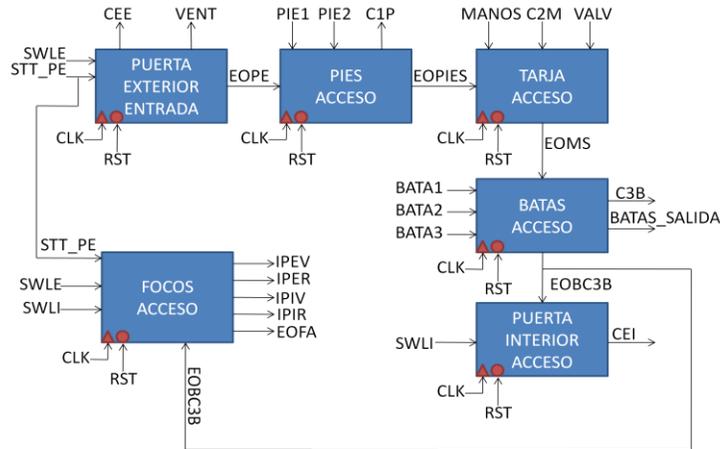
Se visualizan las tramas mediante la señal DATOUT de 8 bits particionada en un vector de 4bits con el objetivo de desplegarlos en BCD a 7 segmentos en el FPGA. El bloque que realiza tal función es llamado DISPLAY\_7s con salida a dos vectores de 7bits. Dependiendo de la tarjeta que proporcionó el usuario es la entrada que se selecciona y por consiguiente una salida de 8bits. Como se muestra en la figura 5-15, la entrada al módulo LUT\_TX\_ID es el vector IDR, el cual se encuentra almacenado.

Si la tarjeta no coincide con alguna de las señales de la tabla, el transmisor enviará un vector de 8 bits en 6 binario 0000 0110, sin embargo, si coincide con alguna trama, podría enviar desde un 0 en binario 0000 0000 hasta un 5 en binario 0000 0101 como máximo. Ya que son 6 tarjetas almacenadas en el multiplexor. La bandera de inicio de transmisión es la señal SV\_ID del módulo ONE\_SHOT.

### **5.3.2. Bloque Protocolo de Acceso**

El bloque que realiza el protocolo de acceso se muestra en la figura 5-13. El estado de la puerta exterior es determinado por un sensor óptico al cual se llamo SWLE; es una entrada externa al sistema. La entrada es aplicada al módulo PUERTA\_EXTERIOR\_ENTRADA, que consiste en una máquina de estados para

controlar las salidas de 2 actuadores, en éste caso particular, a la cerradura de la puerta exterior y al ventilador, éstas salidas se les asignaron el nombre de CEE y VENT respectivamente.



**Figura 5-13** Bloque Protocolo de Acceso.

La lógica del módulo de la puerta exterior es la siguiente. En la entrada llamada STT\_PE, se activa por medio de la señal de válido del bloque SV\_ID, mencionado anteriormente, SV\_ID inicia a la máquina de estados sólo si el usuario se identificó mediante el lector de RFID. Si se detectó la presencia del usuario (Si SV\_ID=1) de inmediato se activa al ventilador, con el objetivo de disminuir la entrada de parásitos hacia la antesala.

A su vez, SV\_ID inicializa al bloque llamado FOCOS\_ACCESO, que inicia la lógica de los indicadores de focos de 40W de los estados de las puertas y los estados que ya se han cumplido y están por cumplirse en el protocolo. FOCOS\_ACCESO tiene como entradas al sensor de la puerta exterior SWLE, al sensor de la puerta interior SWLI y EOBCR, que indica el fin de la máquina de estados del módulo BATAS\_ACCESO. Como salidas se tienen 4 focos de 40W. Se le llama a IPEV; indicador de la puerta exterior en color verde, IPIR; indicador puerta exterior en color rojo, IPIV; indicador puerta interior en color verde e IPIR, indicador puerta interior en color rojo.

Posteriormente se pregunta el estado de la puerta exterior mediante un sensor óptico. Si el sensor no detectó algún cambio, indica que la puerta aun no ha sido abierta y el ventilador sigue activado al igual que el indicador visual IPEV. Si el usuario abrió la puerta y entro hacia la antesala, pero aun no ha cerrado la puerta, los estados de VENT e IPEV siguen activados, estos actuadores se desactivan cuando el usuario haya cerrado la puerta, indicado por el sensor óptico de la puerta exterior, finalizando la máquina de estados y activando a la señal EOPE.

Seguido, el indicador IPER se activa mostrando que, aunque se proporcione la identificación de un segundo usuario, el sistema no dará acceso, ya que adentro de la antesala se encuentra una persona.

El siguiente paso es detectar los pies del usuario mediante la desinfección de su calzado. El módulo PIES\_ACCESO realiza la detección. En la base de los pies mencionada anteriormente, el usuario debe pararse sobre unos pedales, realizando movimientos suaves para la desinfección. El inicio del módulo es proporcionado por EOPE. Teniendo a dos sensores ópticos llamados PIE1 y PIE2, en primera instancia se detecta al pie derecho y posteriormente al pie izquierdo, el control lo realiza una máquina de estados, una vez detectado ambos pies se finaliza la máquina enviando una bandera de salida llamada EOPIES y activando al indicador C1P, que es un foco de 40W en color verde.

El siguiente paso consiste en el lavado de manos, la acción lo realiza el módulo TARJA\_ACCESO. El módulo está compuesto por un sensor reflexivo, al cual se le llamo MANOS, y 3 salidas, C2M que es un indicador visual dado por un foco en color verde a 40W, una electroválvula llamada VALV y el fin de ciclo de la máquina de estados que controlo a éstos periféricos, llamado EOMS.

La señal EOPIES inicializa la máquina de estados para su control. Si aún no ha habido algún cambio en el sensor no se activa la electroválvula. En cambio, si ya se detectó las manos del usuario, VALV se activa durante 5 segundos, lo que se considera suficiente para lavarse bien las manos, además de ahorrar y limitar el uso

del agua. Cuando ha finalizado la base de tiempo de 5 segundos, se activa C2M y por consiguiente a la señal EOMS.

El módulo de BATAS\_ACCESO contiene 3 entradas externas, las cuales son BATA1, BATA2, y BATA3, que son los 3 interruptores de fin de carrera instalados en el acceso. Las 2 salidas son el indicador visual color verde C3B y BATAS\_SALIDA, que es un LED del FPGA, para indicar que ha finalizado el ciclo de la máquina de estados que controla al módulo.

Cuando se pregunta por la vestimenta, y es el primer usuario a entrar, la máquina de estados compara el estado actual del vector de las batas ( $BATAS\_VECTOR = BATA3 \& BATA2 \& BATA1$ ) contra el estado anterior BATAS. Si los vectores son iguales ( $BATAS\_VECTOR = BATAS$ ), no ha habido ningún cambio, por lo que representa un 7 en ambos vectores  $2^3$ . En cambio, si se quita una bata en la posición que desee el usuario, cualquier combinación que se genere en el vector actual BATAS\_VECTOR será menor a un 7, por lo que el sistema dará la señal que se quitó una bata y se continúa con el protocolo. A su vez, si un segundo usuario entra a la antesala, se guarda el registro anterior y se compara con el actual, cualquier combinación que se realice siempre será menor al actual, dando seguimiento al protocolo de acceso y finalizando el ciclo de la máquina de estados por medio de EOBC3B.

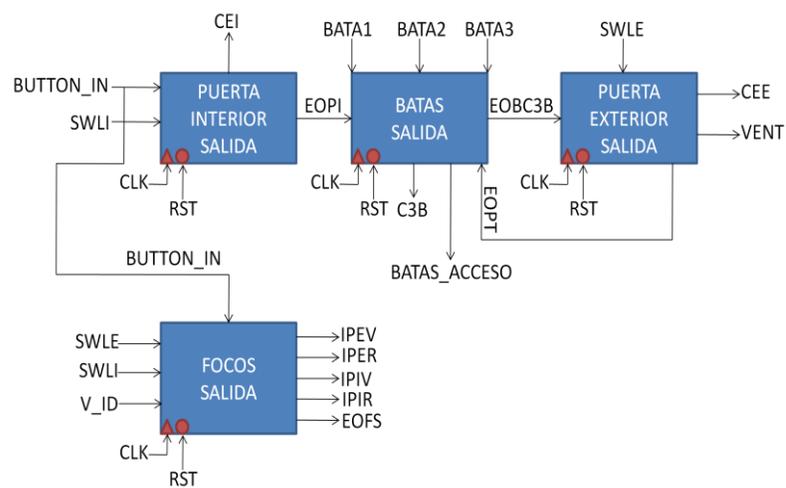
La señal interna EOBC3B conmuta al módulo FOCOS\_ACCESO para seguir con el estado siguiente de la máquina de estados para el algoritmo de la puerta interior. La señal EOBC3B activa al indicador visual de color verde IPIV e IPIR en color rojo. Si la puerta está cerrada, permanecen en ese estado, habrá un cambio siempre y cuando se cierre la puerta, si se cumple la condición ambos focos se desactivarán, esperando el ingreso o salida de un usuario.

Siguiendo con el protocolo de acceso, se han cumplido todas las condiciones anteriores, por lo tanto, el ingreso al invernadero es correcto. La abertura de la puerta interior se realiza por medio de una señal SWLI hacia la cerradura energizada por una bobina a 12VDC. El estado de la puerta está controlado por un sensor óptico

situado en la parte posterior del marco figura 5-15, se colocó una barra de plástico por dentro de la puerta para obstruir el haz de luz cuando se efectúe un cambio de estado. De esta manera, el sensor óptico manda la orden de desactivar la cerradura, por lo que se puede abrir y cerrar la puerta dando fin al protocolo de acceso.

#### 5.4. Protocolo de Salida

El protocolo de salida se muestra en la Figura 5-14 conteniendo 4 bloques los cuales son: PUERTA\_INTERIOR\_SALIDA, BATAS\_SALIDA, FOCOS\_SALIDA, y PUERTA\_EXTERIOR\_SALIDA. El objetivo de éste protocolo es obligar al usuario a dejar la bata en su respectivo gancho y posición, de lo contrario, la salida de la antesala no es permitida. Si el usuario cumple con éste requisito, el sistema pide al usuario identificarse por medio de la tarjeta de RFID, de ésta manera se registra el código del usuario en el FPGA permitiendo la salida de la antesala. El contenido y acción de los módulos del protocolo de salida se describen a continuación.



**Figura 5-14** Bloque Protocolo de Salida.

Se presiona un botón pulsador para interrumpir el paso de la corriente directa de la bobina de la cerradura de la puerta interior, la señal es llamada BUTTON\_IN. La señal activa el bloque llamado PUERTA\_INTERIOR\_SALIDA. Esto inicia una máquina de estados que realiza el algoritmo para el sensor óptico de la puerta interior.

A su vez, BUTTON\_IN activa al módulo FOCOS\_SALIDA, el cual contiene a los dos sensores ópticos de las puertas SWLE y SWLI, la señal de válido del lector RFID V\_ID. Como salidas están los 4 focos indicadores, IPEV, IPER, IPIV e IPIR.

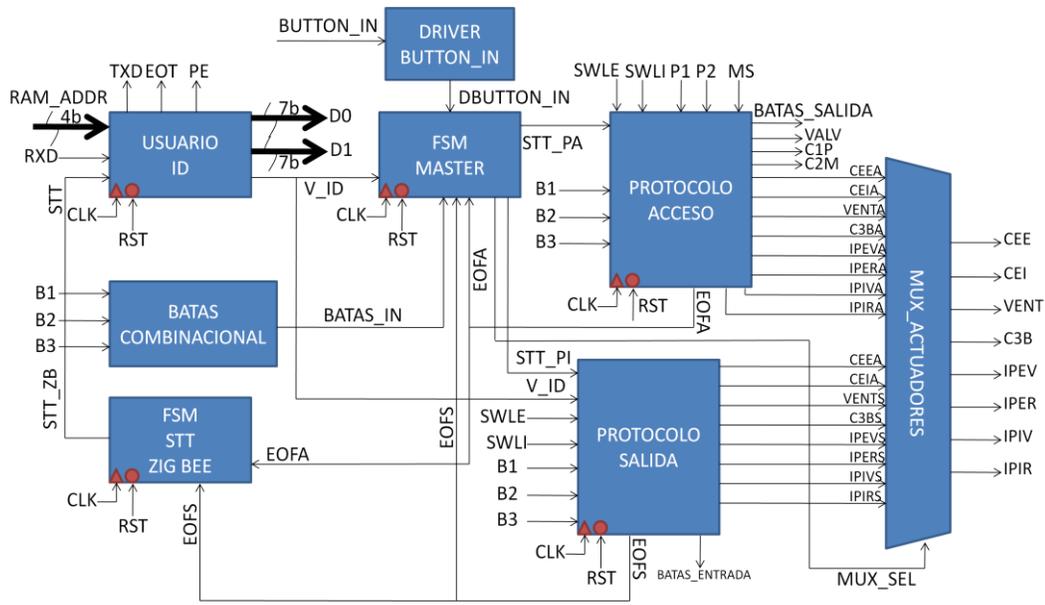
La lógica es similar al módulo FOCOS\_ACCESO, la diferencia es la señal que activa al algoritmo para la puerta exterior (EOBATAS). Cuando el usuario está adentro de la antesala indica que ya se ha cerrado la puerta interior por lo que la máquina de estados del módulo FOCOS\_ACCESO manda una bandera de fin del ciclo (EOPI) activando al bloque de BATAS\_SALIDA.

El algoritmo de este módulo es similar a BATAS\_ACCESO, la diferencia es en la comparación de los vectores. Si el tamaño del vector actual (BATAS\_VECTOR) es menor al vector anterior (BATAS), se manda una bandera de fin de ciclo (EOBATAS) para activar al módulo PUERTA\_EXTERIOR\_SALIDA.

El bloque PUERTA\_EXTERIOR\_SALIDA, retoma el algoritmo de PUERTA\_EXTERIOR\_ACCESO. El ventilador es activado por la señal EOBATAS. Para salir de la antesala, el usuario debe identificarse por RFID, por consiguiente, la puerta permanece cerrada. Si se identifica el usuario, el sistema dará la salida, activando la cerradura de la puerta exterior (CEE). Si la puerta exterior se mantiene abierta, el ventilador sigue activado al igual que los focos indicadores de estados. Estos actuadores se desactivan una vez que la puerta exterior se cierre disminuyendo la entrada de insectos hacia la antesala.

### **5.5. Control de Acceso Automatizado para Invernadero basado en FPGA**

La Figura 5-15 muestra el conjunto de bloques para generar el bloque de jerarquía llamado CONTROL\_ACCESO\_AUTOMATIZADO\_FPGA. El módulo USUARIO\_ID es interconectado con el módulo de la máquina de estados maestra (FSM\_MASTER), el cual controla a los bloques de PRTOCOLO\_ACCESO y PROTOCOLO\_SALIDA.



**Figura 5-15** Bloque Control de Acceso Automatizado para Invernadero basado en FPGA.

Dependiendo si el usuario entre o salga a la antesala es el bloque del protocolo que se pondrá a prueba. Ambos protocolos comparten salidas de actuadores. Por lo que la máquina de estados maestra controla cual protocolo se está ejecutando por medio de un multiplexor llamado MUX\_ACTUADORES. La preferencia es para el protocolo de acceso, ya que por naturaleza, primero se tiene que entrar a la antesala antes de salir.

El módulo BATAS\_COMBINACIONAL realiza una operación lógica, la cual determina si hay batas en la antesala. Si no hay batas colgadas, se indica que hay 3 usuarios trabajando en el invernadero, por lo tanto el acceso es negado. Se necesita por lo menos 1 bata colgada en la porta ganchos para acceder a la antesala.

El módulo FSM\_STT\_ZIGBEE contiene una máquina de estados la cual es controlada por las señales de fin del ciclo del protocolo de acceso y protocolo de salida, EOFA y EOFSS respectivamente. La salida de este módulo activa a la señal STT de módulo USUARIO\_ID para la transmisión de la trama del usuario por medio de ZigBee.

El módulo DRIVER\_BUTTON\_IN contiene una base de tiempo de 100ms para evitar el efecto antirebote generado por el botón pulsador. Esto garantiza que la señal se envía sin ruido hacia la máquina de estados maestra.

### **5.6. Modelo y Análisis Estadístico**

En un experimento diseñado, es decir, cuando se trata de la aplicación de  $r$  tratamientos a un conjunto de unidades experimentales, estas deben seleccionarse para que sean homogéneas, de forma que no se introduzcan factores de variación distintos del que se desea controlar. La asignación de los tratamientos a cada una de las unidades debe hacerse al azar. Es lo que se conoce como diseño “completamente al azar”. El modelo matemático subyacente a este tipo de diseño es el siguiente:

$$X_{ij} = \mu_i + \varepsilon_{ij}$$

$$X_{ij} = \mu + (\mu_i - \mu) + \varepsilon_{ij}$$

$$X_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

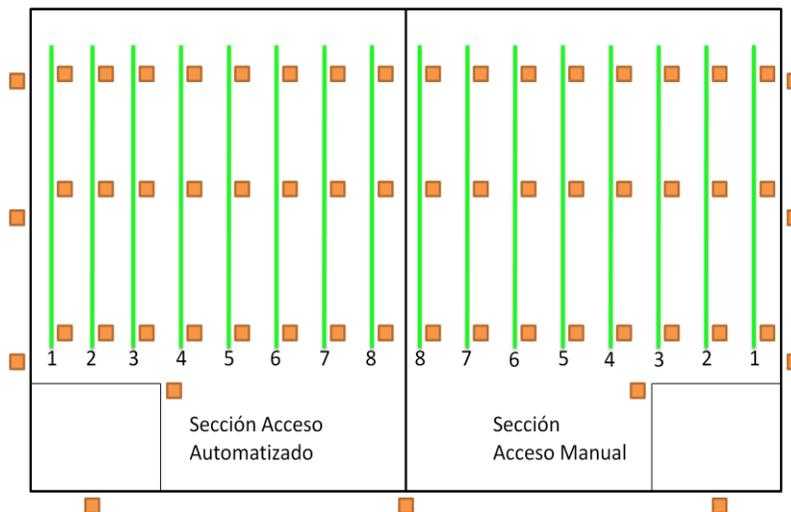
Donde  $\mu_i = \mu + \alpha_i$  es la cantidad que depende del tratamiento usado (descompuesta en una media global y un efecto del tratamiento) y  $\varepsilon_{ij}$  es la cantidad que depende solamente de la unidad experimental y que se identifica con el error experimental.

Los resultados obtenidos fueron expresados como la media  $\pm$  el error estándar (EE). La evaluación estadística de los datos se determinó mediante el análisis de varianza (ANOVA) a un nivel de confianza de 95%. Los análisis estadísticos se realizaron en el programa de Excel y Matrix Laboratories (MATLAB).

### **5.7. Distribución de Trampas para Insectos**

En la Figura 5-16 se muestra la distribución de las trampas correspondientes al inicio de la experimentación. Las trampas se distribuyeron en 3 cartulinas por cada

surco, teniendo 8 surcos por acceso totalizando 48 trampas adentro del invernadero. La enumeración de los surcos se realiza de la entrada al invernadero como el primer surco iniciando en 1 y aumentando hacia la derecha para el acceso automatizado y aumentando en izquierda para el acceso manual.



**Figura 5-16** Distribución de Trampas para Insectos en el Invernadero de la Facultad de Ingeniería.

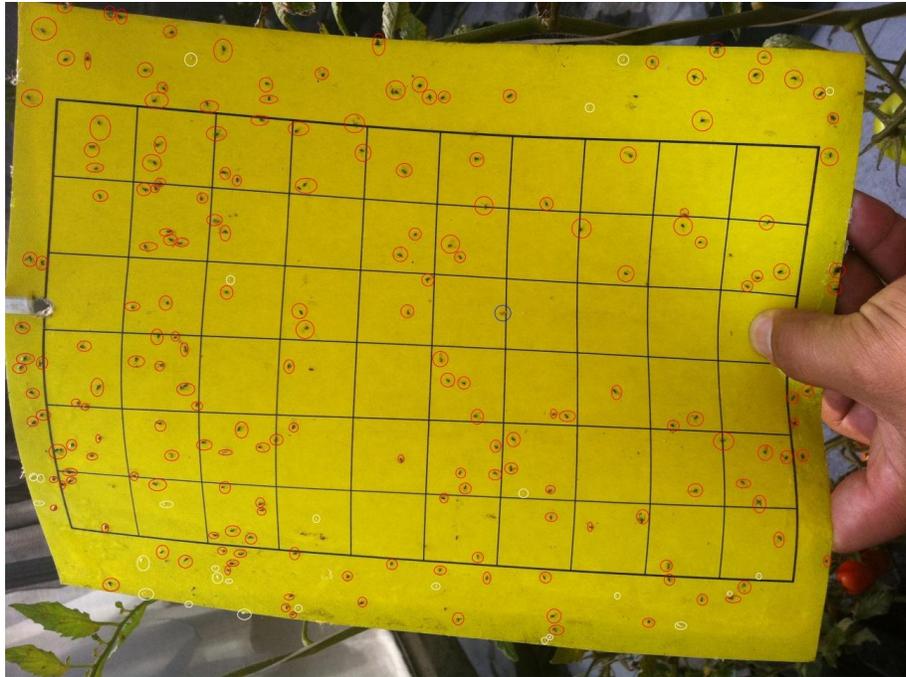
En la Figura 5-17 se observa un surco dentro de un habitáculo el cual se mide el efecto de la radiación en las plantas de jitomate. El invernadero cuenta con 8 habitáculos distribuidos al azar, en cada habitáculo se miden diferentes variables, las cuales son temperatura, humedad, CO<sub>2</sub> y radiación. Se tomaron muestras durante tres semanas de los insectos atrapados en las cartulinas.



**Figura 5-17** Distribución de las trampas por surco en el invernadero.

### **5.8. Captura de Insectos en Cartulinas Pegajosas**

La Figura 5-18 muestra la captura de insectos en las cartulinas pegajosas. Estas cartulinas tienen como característica principal, además del adhesivo, una cuadrícula para facilitar el conteo de los insectos. Como indicadores para identificar y diferenciar las especies de insectos se utilizaron distintos colores redondeando a la especie insectil de interés. En este caso, al final de la fase de experimentación, se observó que apareció con más frecuencia la mosquita negra (círculos color rojos), la mosquita blanca (círculos color blancos) y los trips (círculos color verdes). Éstos insectos son los marcados en la literatura como los más dañinos hacia el cultivo de jitomate. Los insectos restantes se redondearon en color azul identificándolos como otros.



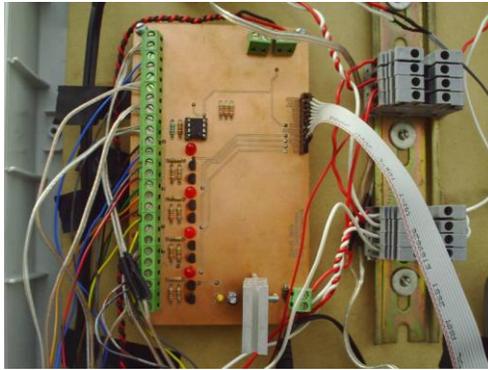
**Figura 5-18** Captura de insectos en las trampas amarillas.

## **6. RESULTADOS Y DISCUSIONES**

### **6.1. Etapa de Control**

La etapa de control se muestra en la Figura 6-1, contiene 4 sensores ópticos infrarrojos, 1 sensor reflexivo, 3 interruptores de fin de carrera y una salida para la transmisión por ZigBee. Las entradas de los sensores son conectados por medio de clemas en color verde, se muestran los diagramas en los anexos. Las señales de control de los sensores son enviadas hacia el FPGA por medio de una tira de 9 pines.

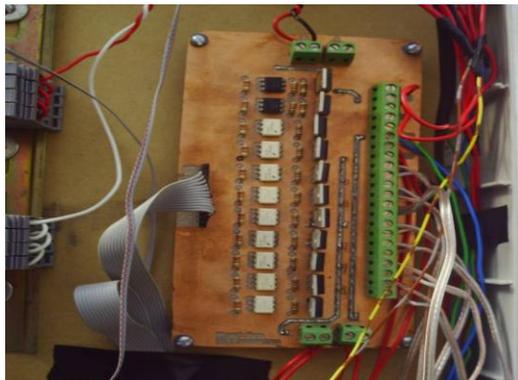
La etapa de control se alimenta por medio de una fuente de 5V, alimentado desde un riel de voltaje a 18V (ver capítulo 6.3). El regulador de voltaje 7805 se conectó a un disipador de aluminio para su distribución y disminución térmica.



**Figura 6-1** Etapa de Control del sistema.

## **6.2. Etapa de Potencia**

La etapa de potencia se muestra en la Figura 6-2, contiene una tira de 11 pines para el control de los actuadores por medio de FPGA. Se tiene el control de 2 cerraduras por medio de 2 optoacopladores 4N25, activando a un transistor Darlington con numeración TIP120. El hardware para controlar a la electroválvula, el ventilador y los 7 focos (110VAC) son por 9 optoacopladores mediante triacs MOC3011. La activación de los actuadores se realiza por Triacs con numeración MAC12MG a 12A.

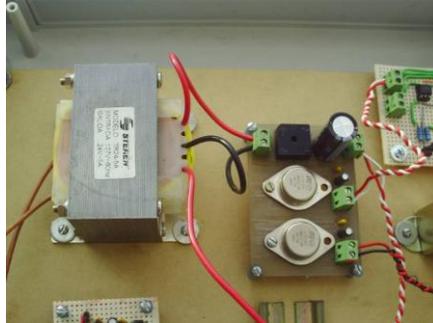


**Figura 6-2** Etapa de potencia del sistema.

## **6.3. Fuente de Poder**

La fuente de poder está compuesta por un transformador de 24V a 5A (Figura 6-3). La salida de voltaje de la tarjeta es de 18V conectada a un riel de voltaje. El riel

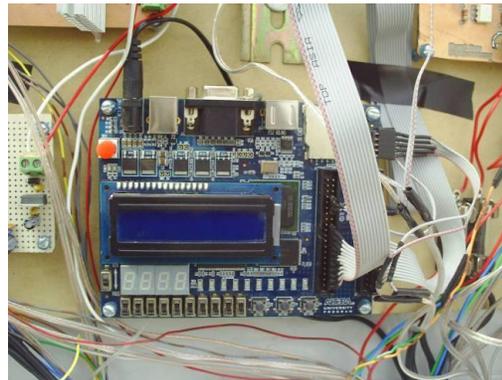
de voltaje distribuye la alimentación hacia la etapa de control, energizando a los sensores. A su vez, la tarjeta cuenta con 2 fuentes de 12V para activar a las cerraduras electrónicas.



**Figura 6-3** Fuentes de poder para la alimentación de periféricos.

#### **6.4. Instalación de la Tarjeta de Desarrollo FPGA**

La tarjeta de desarrollo FPGA se montó en medio y parte inferior de la caja para evitar ruidos del transformador y de los actuadores como se muestra en la Figura 6-4. Además, su posición permite un mejor mapeo de cables hacia las tarjetas de control y potencia reduciendo las longitudes de los cables.



**Figura 6-4** Instalación del FPGA en la caja IP65.

El código sintetizado se graba en la memoria flash del FPGA, esto evita el uso del cable JTAG y el ordenador, dejando instalado el sistema *in situ*. Esto reduce errores de comunicación entre la PC y el FPGA como se muestra en la Figura 6-5.



**Figura 6-5** Comunicación del sistema de control de acceso automatizado con la PC.

Se instaló el cableado de la antesala con cable para bocina de 2 hilos por medio de canaletas con el objetivo de robustecer y dar un toque de estética. Además, evita que los usuarios tropiecen causando daños y fallos al sistema.

La Figura 6-6 muestra la instalación del sistema de acceso automatizado, por lo tanto, se cumple con el primer objetivo particular que es desarrollar e instalar el sistema de control de acceso automatizado en un invernadero con un cultivo en desarrollo.



**Figura 6-6** Sistema de Control de Acceso Automatizado para Invernadero basado en FPGA.

## 6.5. Análisis de Varianza (ANOVA)

El análisis de la varianza se basa en la descomposición de la variabilidad total en dos partes, una parte debida a la variabilidad entre las distintas poblaciones o tratamientos (variabilidad entre grupos o variabilidad explicada por el diseño) y otra parte que puede considerarse como la variabilidad intrínseca de las observaciones (variabilidad dentro de los grupos o residual).

$$Q = Q_E + Q_R$$

La variabilidad entre grupos mide la discrepancia entre los grupos y la media global, de forma que si no hay diferencias entre ellos obtendremos variabilidades pequeñas. Si, por el contrario, existe diferencias cabe esperar que la variabilidad entre grupos sea grande.

$$Q_E = \sum_{i=1}^r n_i (\bar{x}_i - \bar{x})^2$$

La variabilidad dentro de los grupos mide la variabilidad intrínseca de las observaciones, es decir, si el experimento está bien diseñado y no se incluyen factores de variación distintos al estudiado, debe ser error puramente aleatorio producido como resultado de la variabilidad biológica del material experimental.

$$Q_R = \sum_{i=1}^r \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2$$

El vaciado de las ecuaciones anteriores se representa en el Cuadro 6-1 la cual se conoce como tabla de ANOVA y resume toda la información necesaria para realizar el correspondiente contraste y su descripción de la comparación de los datos de la fase de experimentación en la Cuadro 6-4.

Fuente	Suma de Cuadrados	Grados de Libertad	Varianza	F <sub>CAL</sub>
Entre Accesos	$Q_E = \sum_{i=1}^r n_i (\bar{x}_i - \bar{x})^2$	$r-1$	$S_E^2 = \frac{Q_E}{r-1}$	$F_{CAL} = \frac{S_E^2}{S_R^2}$

Dentro de los Accesos	$Q_R = \sum_{i=1}^r \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2$	$N-r$	$S_R^2 = \frac{Q_R}{n-r}$	
Total	$Q_R = \sum_{i=1}^r \sum_{j=1}^{n_i} (x_{ij} - \bar{x})^2$	$N-1$		

**Cuadro 6-1** Expresiones para el cálculo de la ANOVA de un factor ( $r$  indica el número de accesos y  $N$  la cantidad de trampas de insectos).

### 6.5.1. Descripción de los Datos

Los datos corresponden a 3 especies de insectos que se identificaron en mayor cantidad que otros atrapados en cartulinas pegajosas de color amarillo en ambos accesos al final de la fase de experimentación, junto con los totales del tratamiento y el valor medio, tal como se muestra en el cuadro 6-2 y 6-3 (Control de acceso Manual y Control de Acceso Automatizado respectivamente).

Control Manual						
N	Trampas	Mosquita Negra	Mosquita Blanca	Trips	Otros	Total
1	Tr1 S1	95	101	0	1	197
2	Tr2 S1	16	142	0	5	163
3	Tr3 S1	44	111	0	17	172
4	Tr1 S2	78	90	0	15	183
5	Tr2 S2	15	177	35	33	260
6	Tr3 S2	11	108	5	21	145
7	Tr1 S3	21	6	0	61	88
8	Tr2 S3	40	0	0	37	77
9	Tr3 S3	21	0	0	34	55
10	Tr1 S4	54	31	0	11	96
11	Tr2 S4	27	109	0	6	142
12	Tr3 S4	23	77	4	2	106
13	Tr1 S5	91	27	0	3	121
14	Tr2 S5	95	13	0	4	112
15	Tr3 S5	119	6	1	3	129
16	Tr1 S6	50	60	0	63	173
17	Tr2 S6	53	70	0	68	191
18	Tr3 S6	46	87	1	20	154
19	Tr1 S7	32	61	0	1	94
20	Tr2 S7	35	19	0	10	64

21	Tr3 S7	10	24	0	223	257
22	Tr1 S8	36	6	0	15	57
23	Tr2 S8	56	9	0	15	80
24	Tr3 S8	56	25	0	9	90
	<b>Suma</b>	1124	1359	46	677	<b>3168</b>
	<b>Valor Medio, <math>\bar{x}_k</math></b>	46.83	56.625	1.9166	28.21	<b>132</b>
	<b>N<sub>k</sub></b>					<b>24</b>
	<b>Media de todos los resultados</b>					<b>111.875</b>

**Cuadro 6-2** Tabla de resultados de la cuantificación de tres especies en particular de insectos para el acceso manual, valor medio y total de tratamientos.

En el Cuadro 6-2 y 6-3 se puede apreciar que la columna de trampas está dada en función de los surcos (Tr1 hasta Tr3). Esto quiere decir que se distribuyeron 3 trampas por surco, teniendo 8 surcos en total y 24 trampas por acceso, tal como se muestra en la Figura 5-16.

Control Automatizado						
N	Trampas	Mosquita Negra	Mosquita Blanca	Trips	Otros	Total
1	Tr1 S1	147	70	0	4	221
2	Tr2 S1	43	70	0	0	113
3	Tr3 S1	21	11	0	3	35
4	Tr1 S2	47	38	0	1	86
5	Tr2 S2	60	59	1	1	121
6	Tr3 S2	188	26	0	1	215
7	Tr1 S3	59	14	0	0	73
8	Tr2 S3	41	6	0	2	49
9	Tr3 S3	22	0	0	0	22
10	Tr1 S4	15	4	0	0	19
11	Tr2 S4	94	23	0	0	117
12	Tr3 S4	51	0	0	1	52
13	Tr1 S5	103	45	0	6	154
14	Tr2 S5	31	62	0	3	96
15	Tr3 S5	15	5	1	1	22
16	Tr1 S6	30	22	0	0	52
17	Tr2 S6	22	49	0	0	71
18	Tr3 S6	36	3	0	0	39
19	Tr1 S7	96	118	1	1	216
20	Tr2 S7	29	26	0	6	61
21	Tr3 S7	129	71	0	1	201

22	Tr1 S8	90	39	0	0	129
23	Tr2 S8	9	21	0	2	32
24	Tr3 S8	5	1	0	0	6
	<b>Suma</b>	1383	783	3	33	<b>2202</b>
	<b>Valor Medio, <math>\bar{x}_k</math></b>	57.625	32.625	0.125	1.375	<b>91.75</b>
	<b>N<sub>k</sub></b>					<b>24</b>
	<b>Media de todos los resultados</b>					<b>111.875</b>

**Cuadro 6-3** Tabla de resultados de la cuantificación de tres especies en particular de insectos para el acceso automatizado, valor medio y total de tratamientos.

### 6.5.2. Estadística de Prueba

La estadística de prueba se muestra en el Cuadro 6-4, el nivel de confianza  $F_{CAL} > p$ , en este caso se podría concluir que al menos uno de los accesos ha producido resultados de media aritmética de los cuales difiere de forma estadísticamente significativa.

El nivel de confianza indica que la probabilidad de ocurrencia si el acceso automatizado es mejor que el acceso manual es del 97.42%.

Tabla de ANOVA para los Resultados					
Fuente	Suma de Cuadrados	Grados de Libertad	Varianza	F <sub>CAL</sub>	P
Entre Accesos	21000.3	1	21000.3	5.31	0.0258
Dentro de los Accesos	181996.3	46	3956.4		
Total	202996.6	47	24956.7		

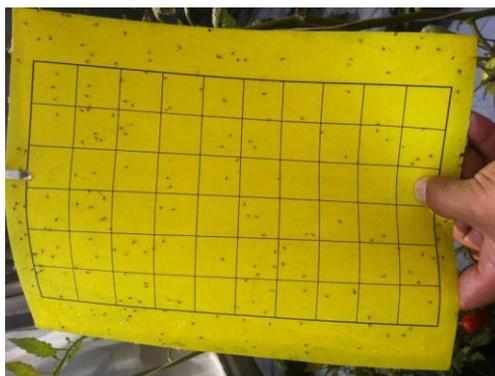
**Cuadro 6-4** Tabla ANOVA para los resultados de los cuadros 6-2 y 6-3.

### 6.6. Comparación de Insectos entre Accesos

La especie de insecto con mayor incidencia de captura durante la fase de experimentación fue la mosquita negra (*Fungus Gnat*). Las moscas “fungus gnats”, son de cuerpo frágil, tamaño diminuto (0.3 cm), con apariencia de mosquito y coloraciones negras o café oscuro. Están presentes en invernaderos con condiciones

húmedas o a exteriores entre residuos orgánicos con condiciones templadas y húmedas (Dreistadt, 2001).

De acuerdo al cuadro 6-2 y 6-3, en la trampa 3 del surco número 2 del acceso automatizado se cuantificaron 188 mosquitas negras, siendo la trampa con más insectos atrapados de ésta especie (ver Figura 6-7). Se capturaron 2507 mosquitas negras durante la fase experimentación, lo que indica el 46.36% del total de insectos capturados en el invernadero (ver Cuadro 6-5).



**Figura 6-7** Trampa con mayor cantidad de mosquitas negras localizada en el acceso automatizado.

El Cuadro 6-5 representa el porcentaje de la captura de insectos por especie en el invernadero. Se cuantificaron un total de 5408 insectos, la mosquita negra representa el 46.36%, siendo la mayor especie capturada. La contraparte le pertenece a los trips con el 0.91% correspondiendo el menor índice de captura.

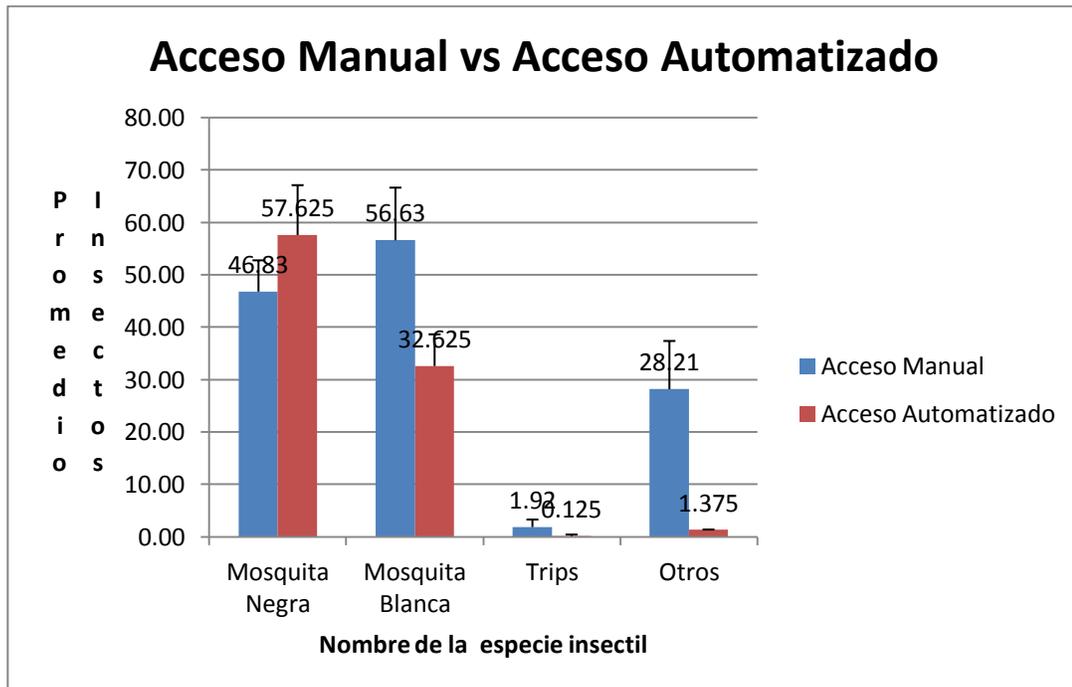
Especie	Mosquita Negra	Mosquita Blanca	Trips	Otros	
Total de Insectos	2507	2142	49	710	5408
% de Insectos Capturados	46.36%	39.61%	0.91%	13.13%	100.00%

**Cuadro 6-5** Porcentajes de la captura de insectos por especie en el invernadero.

En la Figura 6-8 se ilustra a la mosquita negra con una media de captura mayor en el acceso automatizado que las demás especies. De hecho, esta especie

con respecto a la mosquita blanca en el acceso manual tiene un promedio aproximadamente similar en su captura como dato relevante.

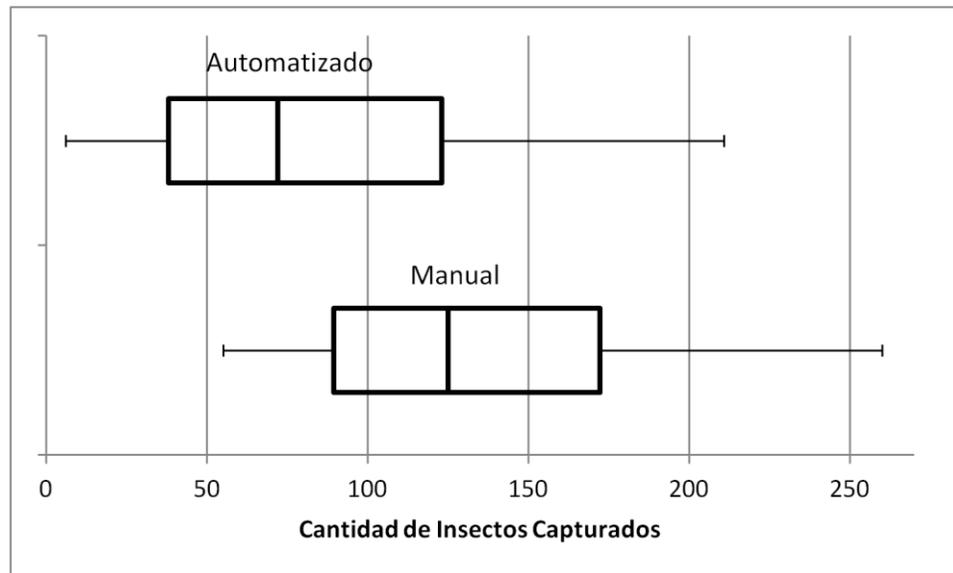
La razón por la cual se presenta una mayor cantidad de mosquita negra en el acceso automatizado se debe a la ubicación cercana frente de un par de árboles, a diferencia del acceso manual, que se encuentra despejado de cualquier planta, arbusto u otro ser vivo que pueda causar un foco de infección considerable hacia el invernadero.



**Figura 6-8** Comparación de medias de la captura de diferentes especies de insectos en ambos accesos físicos.

Al examinar la Figura 6-9 se observa que el 50% de las cartulinas pegajosas del acceso automatizado comprenden entre 38 y 123 insectos capturados. La mínima cantidad de muestras en las cartulinas de mismo acceso es de 6 insectos correspondiendo a la trampa 3 del surco 8. Por el contrario, se capturaron 221 insectos correspondiendo a la trampa 1 del surco 1 siendo la máxima cantidad en el acceso automatizado. El primer cuartil del acceso automatizado se lee como 38 insectos capturados, lo que significa que el 75% de las trampas se capturaron más

de 38 insectos, en comparación con el acceso manual donde se atraparon más 89 insectos. El tercer cuartil en ambos accesos indica que el 25% de las trampas en el acceso automatizado se obtuvieron más de 123 muestras contra 173 insectos del acceso manual.

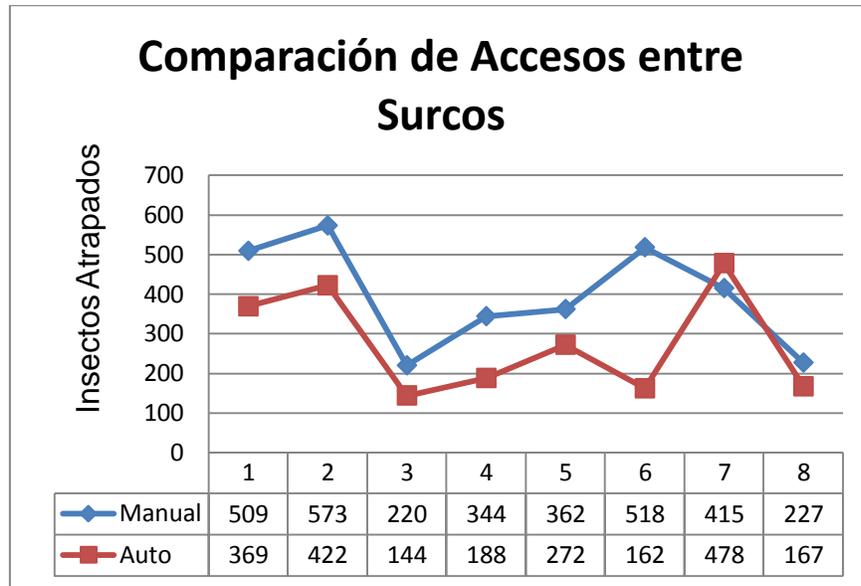


**Figura 6-9** Caja de valores extremos de la cuantificación de insectos.

La Figura 6-10 muestra la comparación de los surcos entre ambos accesos. Se observa en los surcos 2 y 6 del acceso manual una mayor cantidad de insectos atrapados con respecto a los demás surcos. Además, en el surco 2 del acceso manual se presenta la mayor captura de insectos en el invernadero. Esto se debe a la ubicación cercana hacia la entrada de la antesala del invernadero. Otro dato importante se observa en el surco 7 en el acceso automatizado, el cual presenta un mayor número de insectos atrapados lo que probablemente pueda atribuirse a la atracción de los insectos hacia el habitáculo.

La relación del surco 6 entre ambos accesos es potencialmente significativa en comparación con los demás surcos teniendo un aumento del más del 150%, esto probablemente se deba a la incorporación del habitáculo con tratamiento de CO<sub>2</sub> en el acceso manual, el cual por sus características de humedad y temperatura atraían

una mayor cantidad de insectos elevando el número de estos a diferencia del acceso automatizado en donde no se incorporó habitáculo.



**Figura 6-10** Comparación de surcos entre el acceso automatizado y el acceso manual.

## 7. CONCLUSIONES

En este trabajo es presentada una alternativa para el control de plagas en invernaderos mediante el control de acceso hacia la estructura. El sistema desarrollado es capaz de seguir obligatoriamente con un protocolo de medidas fitosanitarias para favorecer la inocuidad y disminuir la presencia de insectos y focos de infección.

La arquitectura FPGA utilizada a un bajo costo permite la conectividad de varios sensores y actuadores para la detección de las diferentes condiciones a las cuales el usuario se tiene que someter para cumplir con los protocolos que el sistema desarrollado especifica. Esta arquitectura permite el paralelismo, por lo tanto, se ejecutan varios procesos al mismo tiempo, lo cual se activan/desactivan diferentes máquinas de estados tomando las decisiones pertinentes para establecer el estado

donde se encuentran los protocolos de entrada y salida. Estas son las principales ventajas de utilizar una arquitectura paralela en relación con un microcontrolador.

El control de acceso automatizado obliga al usuario a cumplir con diferentes medidas fitosanitarias, como lo son la desinfección del calzado, lavado adecuado de manos y vestimenta apropiada con batas de laboratorio a diferencia de un acceso manual, el cual no restringe el acceso de personal, además de estar propenso al factor y descuido humano. El desarrollo del sistema permite ser instalado en cualquier invernadero que contenga toma de agua y electricidad. El sistema de control de acceso automatizado tiene robustez al ser instalado en cajas con norma IP el cual garantiza la protección fuerte contra el polvo, así como filtración de agua y humedad.

El nivel de confianza de un sistema de control automatizado para invernadero basado en FPGA es del 97.4% con relación a un acceso manual. Esto demuestra la eficacia entre ambos sistemas de control de acceso logrando disminuir la presencia de los principales insectos causantes de mayores daños a cultivos en jitomate, en los cuales se enfocó ésta investigación, a diferencia del acceso manual. Esta tendencia se puede observar en la Figura 6-8 del presente trabajo. Esto debido a la implementación de tecnología basada en FPGA y sistemas mecatrónicos llevando un control de usuarios mediante la identificación de usuarios por radio frecuencia (RFID).

## 8. BIBLIOGRAFÍA

- Akira Ito Sérgio, Carro Luigi, 2001. Making Java Work for Microcontroller Applications. IEEE Design & Test of Computers. 18(5), 100-110.
- Akyildiz, I.F. and Su, W. and Sankarasubramaniam, Y. and Cayirci, E. 2002. A survey on sensor networks. Communications Magazine, IEEE. 40(8), 102-114.
- Boissard, P. and Martin, V. and Moisan, S. 2008. A cognitive vision approach to early pest detection in greenhouse crops. Computers and electronics in agriculture. 62(2), 81-93.
- Blackmer, J.L. and Byers, J.A. and Rodriguez-Saona, C. 2007. Evaluation of color traps for monitoring *Lygus spp.*: Design, placement, height, time of day, and non-target effects. Crop Protection. 27(2), 171-181.
- Cadoni, M. and Grosso, E. and Lagorio, A. and Tistarelli, M. 2011. From 3D faces to biometric identities. Biometrics and ID Management. p.p. 156-167.
- Chan, Y.F. and Moallem, M. and Wang, W. 2007. Design and Implementation of Modular FPGA-Based PID Controllers. Industrial Electronics, IEEE Transactions on. 54(4), 1898-1906.
- Cho, J. and Choi, J. and Qiao, M. and Ji, C. and Kim, H. and Uhm, K. and Chon, T. 2007. Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis. International Journal of Mathematics and Computer in Simulation. 346(246), 244.
- Clarke, N. 2011. Transparent User Authentication: Biometrics, Rfid and Behavioural Profiling. Ed. Springer Verlag. Cap. 5. p.p. 112-119.
- Clay, D. and Shanahan, J.F. 2011. GIS Applications in Agriculture: Nutrient Management for Energy Efficiency. Ed. CRC. Cap. 7. p.p. 123-143.
- Cyclone III Device Handbook, 2010. Altera, Volume 1.
- Dargie Walteneus, Poellabauer Christian. 2010. Fundamentals of Wireless Sensor Networks Theory and Practice. Ed. Wiley. Sección 1. Cap. 1. p.p. 4-7.
- Deitel. Paul J. y Harvey M. Deitel. 2008. *Cómo programar en Java*. México: Pearson Educación.
- Dreistadt, S.H. 2001. Integrated pest management for floriculture and nurseries. Agriculture & Natural Resources. Vol 3402,

- Durmusoglu, E. and Civelek, M.S.H.S. and Balci, A.H.H. 2009. Effect of various hues of yellow as sticky trap color on leafminers in cucumber growing greenhouses. 33(3), 162-170.
- Eilenberg, J. and Hajek, A. and Lomer, C. 2001. Suggestions for unifying the terminology in biological control. *BioControl*. 46(4), 387-400.
- Eren, H. 2006. *Wireless sensors and instruments: networks, design, and applications*. Ed. CRC Press. Cap. 1. p.p. 1-3.
- Ferrer, G. and Dew, N. and Apte, U. 2009. When is RFID right for your service?. *International Journal of Production Economics*. 124(2), 414-425.
- Flint, M. L. & van den Bosch, R. 1981. *Introduction to Integrated Pest Management*. New York: Plenum Press.
- Gould, J. and Hoelmer, K. and Goolsby, J. 2008. Classical Biological Control of *Bemisia Tabaci* in the United States: A Review of Interagency Research and Implementation. Ed. Springer Verlag. Vol 4. Cap. 1. p.p. 1-16.
- Gu, X.S. and Bu, W.J. and Xu, W.H. and Bai, Y.C. and Liu, B.M. and Liu, T.X. 2008. Population suppression of *Bemisia tabaci* (Hemiptera: Aleyrodidae) using yellow sticky traps and *Eretmocerus nr. rajasthanicus* (Hymenoptera: Aphelinidae) on tomato plants in greenhouses. *Insect Science*. 15(3), 263-270.
- Hajek, A.E. 2004. *Natural enemies: an introduction to biological control*. Ed. Cambridge Univ Pr. Cap. 2. p.p. 37-38.
- Krishnan, M. and Foster, C.A. and Strosser, R.P. and Glancey, J.L. and Sun, J.Q. 2006. Adaptive modeling and control of a manure spreader for precision agriculture. *Computers and electronics in agriculture*. 52(1-2), 1-10.
- Kummel, K. and Vielhauer, C. 2011. Biometric Hash algorithm for dynamic handwriting embedded on a java card. *Biometrics and ID Management*. p.p. 61-72.
- Ley Federal de Sanidad Vegetal, 2008. SCJN DOF 18-11-2008.
- Lee, L.S. and Fiedler, K.D. and Smith, J.S. Smith, 2008. Radio frequency identification (RFID) implementation in the service sector: A customer-facing diffusion model. *International Journal of Production Economics*. 112(2), 587-600.
- Li, S.Z. and Jain, A.K. 2009. *Encyclopedia of biometrics*. Ed. Springer Verlag. Vol 2. p.p. 2-8.

- Luvisi, A. and Triolo, E. and Rinaldelli, E. and Bandinelli, R. and Pagano, M. and Gini, B. 2010. Radiofrequency applications in grapevine: From vineyard to web. *Computers and Electronics in Agriculture*. 70(1), 256-259.
- Mariño Acebal J. B., Vallverdú I. B. F., Rodríguez Fonollosa J. A. Moreno B. A. 1995. TRATAMIENTO DIGITAL DE LA SEÑAL: Una introducción experimental. GRUPO DE PROCESADO DE SEÑAL Departamento de Teoría de la Señal y Comunicaciones.
- Morais, R. and Fernandes, M.A. and Matos, S.G. and Serodio, C. and Ferreira, P. and Reis, M. 2008. A ZigBee multi-powered wireless acquisition device for remote sensing applications in precision viticulture. *Computers and Electronics in Agriculture*. 62(2), 94-106.
- Matese, A. and Di Gennaro, SF and Zaldei, A. and Genesio, L. and Vaccari, FP. 2009. A wireless sensor network for precision viticulture: The NAV system. *Computers and Electronics in Agriculture*. 69(1), 51-58.
- Nanavati, S. and Thieme, M. and Raj, N. and Nanavati, R. 2002. Biometrics: identity verification in a networked world. Ed. Wiley. Cap. 1. p.p. 3-7.
- Ogata, K. 2001. Modern control engineering. Ed. Prentice Hall PTR. Cap 1. p.p. 1.
- RAE. 2007. Diccionario de la Real Academia Española. 2007. [www.rae.es/rae.html](http://www.rae.es/rae.html).
- Pallas-Areny, R. and Webster, J.G. and Knovel. 2001. Sensors and signal conditioning Second Edition. Ed. Wiley Cap1. p.p. 1.
- Perdikis, D. and Kapaxidi, E. and Papadoulis, G. 2008. Biological Control of Insect and Mite Pests in Greenhouse Solenaceous Crops. *The European Journal of Plant Science and Biotechnology*. 2, 125-144.
- Pilkington Leigh J., Messelink Gerben, Van Lenteren Joop C., Le Mottee Kristian. 2009. "Protected Biological Control" – Biological pest management in the greenhouse industry. *Biological Control*. 52(3), 216-220.
- Romero-Troncoso, R.J. 2007. *Electrónica digital y lógica programable*. Guanajuato : Universidad de Guanajuato.
- Ruiz-Garcia Luis, Lunadei Loredana, Barreiro Pilar and Robla Jose Ignacio. 2009. A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends. *Sensors*. 9(6), 4728-4750.
- Solis-Sánchez, LO and García-Escalante, JJ and Castañeda-Miranda, R. and Torres-Pacheco, I. and Guevara-González, R. 2009. Machine vision algorithm for

whiteflies (*Bemisia tabaci* Genn.) scouting under greenhouse environment. *Journal of Applied Entomology*. 133(7), 546-552.

Solis-Sánchez, L.O. and Castañeda-Miranda, R. and García-Escalante, J.J. and Torres-Pacheco, I. and Guevara-González, R.G. and Castañeda-Miranda, C.L. and Alaniz-Lumbreras, P.D. 2010. Scale Invariant Feature Approach for Insect Monitoring. *Computers and Electronics in Agriculture*. Art. In Press.

Sparks, D.L. 1999. *Advances in Agronomy* Vol 67. Ed. Academic Press. Cap. 1. p.p. 2-5.

Sulaiman, N. and Obaid, Z.A. and Marhaban, MH and Hamidon, MN. 2009. Design and Implementation of FPGA-Based Systems - A Review. *Australian Journal of Basic and Applied Sciences*. 3(4) 3575-3596.

Ruzairi, A.R. and Chiam, K.T. and Hafiz, M. and Rahiman, F. 2008. An Optical Tomography System Using a Digital Signal Processor. *Sensors*. 8(4), 2082-2103.

Samad, T. and McLaughlin, P. and Lu, J. 2006. System architecture for process automation: Review and trends. *Journal of Process Control*. 17 (3), 191-201.

Sugai M., Nishimura K., Takamatsu K., Fujinaga T. 1999. Low Power Consumption Microcontrollers and Their Applications. *Hitachi Review*. 48(6), 313-317.

Tapia, A.B. and Mir, L.V. and Universidad Autónoma de Chapingo México. 2006. Manejo y operación de invernaderos agrícolas.

Van Deursen, T. and Radomirovic, S. 2009. Security of RFID Protocols – A Case Study. *Electronic Notes in Theoretical Computer Science*. 244, 41-52.

Want, R. 2006. An Introduction to RFID Technology. *IEEE, Pervasive Computing*. 5(1), 25-33.

Wu, N.C. and Nystrom, MA and Lin, T.R. and Yu, H.C. 2006. Challenges to global RFID adoption. *Technovation*. 26(12), 1317-1323.

Zanic, K. and Goreta, S. and Perica, S. and Sutic, J. 2008. Effects of alternative pesticides on greenhouse whitefly in protected cultivation. *Journal Pest Science*. 81(3), 161-166.

## **ANEXOS**

library IEEE;

use IEEE.std\_logic\_1164.all;

entity CONTROL\_ACCESO\_AUTO\_FPGA is

port(

RST: in std\_logic;

CLK: in std\_logic;

RAM\_ADDR: in std\_logic\_vector (3 downto 0);

RXD: in std\_logic;

BUTTON\_IN: in std\_logic;

SWLE: in std\_logic;

SWLI: in std\_logic;

PIE1: in std\_logic;

PIE2: in std\_logic;

MANOS: in std\_logic;

BATA1: in std\_logic;

BATA2: in std\_logic;

BATA3: in std\_logic;

TXD: out std\_logic;

EOT: out std\_logic;

PE: out std\_logic;

D0: out std\_logic\_vector(0 to 6);

D1: out std\_logic\_vector(0 to 6);

BATAS\_ACCESO: out std\_logic;

BATAS\_SALIDA: out std\_logic;

CEE: out std\_logic;

CEI: out std\_logic;

VALV: out std\_logic;

VENT: out std\_logic;

C1P: out std\_logic;

```

C2M:                out std_logic;
C3B:                out std_logic;
IPEV:              out std_logic;
IPER:              out std_logic;
IPIV:              out std_logic;
IPIR:              out std_logic;
EOZB:              out std_logic
);
end CONTROL_ACCESO_AUTO_FPGA;

architecture JERARQUIA of CONTROL_ACCESO_AUTO_FPGA is
component USUARIO_ID
port(
RST:   in  std_logic;   --Reset maestro
CLK:   in  std_logic;   --Reloj maestro
RXD:   in  std_logic;   --Entrada Serial
STT:   in  std_logic;   --Inicio de Transmisión Serial
RAM_ADDR:  in  std_logic_vector(3 downto 0); --Direccion de lectura de la RAM
TXD:   out std_logic;   --Dato a Transmitir ZigBee
EOT:   out std_logic;   --Fin de Transmisión Serial
PE:    out std_logic;   --Error de Paridad
SV_ID:  out std_logic;  --ID Válida
D0,D1: out std_logic_vector(0 to 6) --Despliegue de Datos de Tramas por RFID
);
end component;

component FSM_MASTER
port(
RST:   in  std_logic;

```

```

CLK:      in std_logic;
V_ID:     in std_logic;
BATAS_IN: in std_logic;
BUTTON_IN: in std_logic;
EOFA:     in std_logic;
EOFS:     in std_logic;
SV_ID:    out std_logic;
SBUTTON_IN: out std_logic;
MUX_SEL:  out std_logic
);
end component;

component BATAS_COMB
port(
BATA1:  in std_logic;
BATA2:  in std_logic;
BATA3:  in std_logic;
BATAS_IN:  out std_logic
);
end component;

component PROTOCOLO_ACCESO_TESIS
port(
RST:           in std_logic;  --Reset maestro
CLK:           in std_logic;  --Reloj maestro
SV_ID:         in std_logic;
SWLE:         in std_logic;
SWLI:         in std_logic;
PIE_1:        in std_logic;

```

```

PIE_2:          in std_logic;
MANOS:          in std_logic;
BATA1:          in std_logic;
BATA2:          in std_logic;
BATA3:          in std_logic;
BATAS_SALIDA:  out std_logic;
CEE:            out std_logic;
CEI:            out std_logic;
VALV:          out std_logic;
VENT:          out std_logic;
C1P:           out std_logic;
C2M:           out std_logic;
C3B:           out std_logic;
IPEV:          out std_logic;
IPER:          out std_logic;
IPIV:          out std_logic;
IPIR:          out std_logic;
EOFA:          out std_logic
);
end component;

component PROTOCOLO_SALIDA_TESIS
port(
RST:           in std_logic;  --Reset maestro
CLK:           in std_logic;  --Reloj maestro
BUTTON_IN:     in std_logic;
SWLE:          in std_logic;
SWLI:          in std_logic;

```

```

SV_ID:      in std_logic;

BATA1:      in std_logic;

BATA2:      in std_logic;

BATA3:      in std_logic;

CEI:        out std_logic;

CEE:        out std_logic;

VENT:       out std_logic;

C3B:        out std_logic;

IPEV:       out std_logic;

IPER:       out std_logic;

IPIV:       out std_logic;

IPIR:       out std_logic;

EOFS:       out std_logic;

BATAS_ACCESO: out std_logic

);

end component;

component FSM_STT_ZIGBEE

port(

RST:      in std_logic;

CLK:      in std_logic;

EOFS,EOFA: in std_logic;

STT_ZB,EOZB: out std_logic

);

end component;

component DRIVER_BUTTON_IN

port(

RST:      in std_logic;

```

```

CLK:      in std_logic;

BUTTON_IN:      in std_logic;

Z_100ms:      in std_logic;

S_BUTTON_IN:      out std_logic;

OPC_CONT_100ms: out std_logic_vector(1 downto 0)

);

end component;

component MUX_ACTUADORES

port(

X1:      in std_logic;

X2:      in std_logic;

S0:  in std_logic;

Y:      out std_logic

);

end component;

component BASE_TIEMPO_100ms

port(

RST:      in std_logic;          --Reset maestro

CLK:      in std_logic;          --Reloj maestro

OPC_CONT_100ms: in std_logic_vector(1 downto 0);  --Control del contador

Z_100ms:      out std_logic          --Bandera de Fin de
Cuenta

);

end component;

signal
STT_ZB,V_ID,SV_ID,BATAS_IN,SBUTTON_IN,EOfA,EOfS,D_BUTTON_IN,MUX_SEL,Z_10
0ms: std_logic;

```

```

signal
CEEA,CEES,CEIA,CEIS,VENTA,VENTS,C3BA,C3BS,IPEVA,IPEVS,IPERA,IPERS,IPIVA,IPIV
S,IPIRA,IPIRS: std_logic;

signal OPC_CONT_100ms: std_logic_vector(1 downto 0);

begin

U0: USUARIO_ID port map (RST,CLK,RXD,STT_ZB,RAM_ADDR,TXD,EOT,PE,V_ID,D0,D1);

U1:          FSM_MASTER          port          map
(RST,CLK,V_ID,BATAS_IN,D_BUTTON_IN,EOFA,EOFS,SV_ID,SBUTTON_IN,MUX_SEL);

U2: BATAS_COMB port map (BATA1,BATA2,BATA3,BATAS_IN);

U3:          PROCOLO_ACCESO_TESIS          port          map
(RST,CLK,SV_ID,SWLE,SWLI,PIE1,PIE2,MANOS,BATA1,BATA2,BATA3,BATAS_SALIDA,CE
EA,CEIA,VALV,VENTA,C1P,C2M,C3BA,IPEVA,IPERA,IPIVA,IPIRA,EOFA);

U4:          PROCOLO_SALIDA_TESIS          port          map
(RST,CLK,BUTTON_IN,SWLE,SWLI,V_ID,BATA1,BATA2,BATA3,CEIS,CEES,VENTS,C3BS,I
PEVS,IPERS,IPIVS,IPIRS,EOFS,BATAS_ACCESO);

U5: FSM_STT_ZIGBEE port map (RST,CLK,EOFS,EOFA,STT_ZB,EOZB);

U6:          DRIVER_BUTTON_IN          port          map
(RST,CLK,SBUTTON_IN,Z_100ms,D_BUTTON_IN,OPC_CONT_100ms);

U7: MUX_ACTUADORES port map (CEEA,CEES,MUX_SEL,CEE);

U8: MUX_ACTUADORES port map (CEIA,CEIS,MUX_SEL,CEI);

U9: MUX_ACTUADORES port map (VENTA,VENTS,MUX_SEL,VENT);

U10: MUX_ACTUADORES port map (C3BA,C3BS,MUX_SEL,C3B);

U11: MUX_ACTUADORES port map (IPEVA,IPEVS,MUX_SEL,IPEV);

U12: MUX_ACTUADORES port map (IPERA,IPERS,MUX_SEL,IPER);

U13: MUX_ACTUADORES port map (IPIVA,IPIVS,MUX_SEL,IPIV);

U14: MUX_ACTUADORES port map (IPIRA,IPIRS,MUX_SEL,IPIR);

U15: BASE_TIEMPO_100ms port map (RST,CLK,OPC_CONT_100ms,Z_100ms);

end architecture;

library IEEE;

use IEEE.std_logic_1164.all;

entity USUARIO_ID is

```

```

port(
RST:  in std_logic;  --Reset maestro
CLK:  in std_logic;  --Relej maestro
RXD:   in std_logic;      --Entrada Serial
STT:   in std_logic;      --Inicio de Transmisión Serial
RAM_ADDR:  in std_logic_vector(3 downto 0); --Direccion de lectura de la RAM
TXD:   out std_logic;      --Dato a Transmitir ZigBee
EOT:   out std_logic;      --Fin de Transmisión Serial
PE:   out std_logic;      --Error de Paridad
SV_ID:  out std_logic;  --ID Válida
D0,D1:  out std_logic_vector(0 to 6) --Despliegue de Datos de Tramas por RFID
);
end USUARIO_ID;

architecture JERARQUIA of USUARIO_ID is
component RS232_TXRX
port(
RST:  in std_logic;  --Reset maestro
CLK:  in std_logic;  --Relej maestro
STT:  in std_logic; --Start transmission
TXD:  out std_logic; --Salida de transmision
DATIN: in std_logic_vector(7 downto 0); --Dato de entrada a transmitir...
EOT:  out std_logic; --Fin de transmision
RXD:  in std_logic; --Entrada serial
DATOUT: out std_logic_vector(7 downto 0);--Dato recibido
PE:  out std_logic;  --Error de paridad
V:   out std_logic  --Indicador de dato valido
);

```

```

end component;

component BLK_RAM

generic(

n: integer := 16;-- No de datos

m: integer := 8; -- ancho de la trama

p: integer := 4          -- ancho del bus de direcciones

);

port(

    RAM_CKA:  in  std_logic;  --Reloj de escritura

    RAM_CKB:  in  std_logic;  --Reloj de lectura

    RAM_DWR:  in  std_logic_vector (m-1 downto 0);  --Dato entrada

    RAM_ADA:  in    std_logic_vector (p-1 downto 0);  --Direccion de escritura

    RAM_ADB:  in  std_logic_vector (p-1 downto 0);  --Direccion de lectura

    RAM_WRE:  in  std_logic;  --Habilitacion de escritura

    RAM_RDA:  out std_logic_vector (m-1 downto 0)  --Dato salida

);

end component;

component CONTADOR_M4

port(

RST : in  std_logic; --Reset maestro

CLK : in  std_logic; --Reset maestro

V   : in  std_logic; --Bandera de Válido de la Recepción de Datos del RFID

RAM_POS  : out std_logic_vector(3 downto 0) --Posición de la RAM

);

end component;

component ID_REGISTROS

port(

```

```

RST:    in std_logic; --Reset maestro
CLK:    in std_logic; --Reloj maestro
V:      in std_logic; --Trama Valida
SDATAOUT: in std_logic_vector(7 downto 0); --Dato de Entrada a los Registros
EOFCR:  out std_logic; --Bandera de Fin de la Máquina de Estados
ID:     out std_logic_vector(127 downto 0) --Vector ID del RFID
);
end component;

component COMPARADOR_N
generic( n: integer := 128 --Tamaño del vector
);
port(
RST: in std_logic;
CLK: in std_logic;
IDR: in std_logic_vector(n-1 downto 0);
V_ID: out std_logic
);
end component;

component LUT_TX_ID
port(
ID:          in std_logic_vector(127 downto 0); --Selección de ID
TX_ID: out std_logic_vector(7 downto 0)      --Tx por ZigBee
);
end component;

component ALTERADE0_SSDISPLAY_DECODER
port(
BCD0,BCD1: in std_logic_vector(3 downto 0); --Entradas para los Registros de las Tramas
IDs

```

```

D0,D1: out std_logic_vector(0 to 6)          --Salida BCD a 7 Segmentos
);
end component;

component REG_H_CLR
port(
RST: in std_logic; --Reset Maestro
CLK: in std_logic;  --Reloj Maestro
H:   in std_logic;  --Habilitación del Registro
ID:          in std_logic_vector(127 downto 0); --Entrada al Registro
IDR: out std_logic_vector(127 downto 0) --Salida del Registro
);
end component;

component ONE_SHOT
port(
RST: in std_logic;          --Master Reset
CLK: in std_logic;          --Master Clock
BUTTON: in std_logic; --Input button
SHOT: out std_logic          --One Shot Out
);
end component;

signal V,EOF,CR,V_ID: std_logic;

signal SDATAOUT,DATAOUT,TX_ID: std_logic_vector(7 downto 0);

signal RAM_POS: std_logic_vector(3 downto 0);

signal ID,IDR: std_logic_vector(127 downto 0);

begin

U0: RS232_TXRX port map (RST,CLK,STT,TXD,TX_ID,EOT,RXD,SDATAOUT,PE,V);

U1: BLK_RAM port map(CLK,CLK,SDATAOUT,RAM_POS,RAM_ADDR,V,DATAOUT);

```

```

U2: CONTADOR_M4 port map (RST,CLK,V,RAM_POS);

U3: ID_REGISTROS port map(RST,CLK,V,SDATAOUT,EOFCR,ID);

U4: COMPARADOR_N port map (RST,CLK,IDR,V_ID);

U6: ALTERADE0_SSDISPLAY_DECODER port map (DATAOUT(3 downto 0),DATAOUT(7
downto 4),D0,D1);

U7: LUT_TX_ID port map (IDR,TX_ID);

U8: REG_H_CLR port map (RST,CLK,EOFCR,ID,IDR);

U9: ONE_SHOT port map (RST,CLK,V_ID,SV_ID);

end architecture;

library IEEE;

use IEEE.std_logic_1164.all;

entity FSM_MASTER is

port(

RST:    in std_logic;

CLK:    in std_logic;

V_ID:   in std_logic;

BATAS_IN:  in std_logic;

BUTTON_IN: in std_logic;

EOFA:     in std_logic;

EOFS:     in std_logic;

SV_ID:    out std_logic;

SBUTTON_IN: out std_logic;

MUX_SEL:  out std_logic

);

end FSM_MASTER;

architecture compuesta of FSM_MASTER is

type FSM_STATES is (S0,S1,S2,S3,S4,S5);

signal Qp,Qn: FSM_STATES;

```

```
begin
```

```
Combinacional: process (Qp,V_ID,BATAS_IN,BUTTON_IN,EOFA,EOFS)
```

```
begin
```

```
    case Qp is
```

```
        when S0 =>
```

```
            if (BATAS_IN='1') then
```

```
                Qn <= S1;
```

```
            elsif (BUTTON_IN='1') then
```

```
                Qn <= S2;
```

```
            else
```

```
                Qn <= Qp;
```

```
            end if;
```

```
            SV_ID <= '0';
```

```
            SBUTTON_IN <= '0';
```

```
            MUX_SEL <= '0';
```

```
        when S1 =>
```

```
            if (V_ID='1') then
```

```
                Qn <= S3;
```

```
            elsif (BUTTON_IN='1') then
```

```
                Qn <= S2;
```

```
            else
```

```
                Qn <= Qp;
```

```
            end if;
```

```
            SV_ID <= '0';
```

```
            SBUTTON_IN <= '0';
```

```
            MUX_SEL <= '0';
```

```
        when S2 =>
```

```

Qn <= S4;
SV_ID <= '0';
SBUTTON_IN <= '1';
MUX_SEL <= '1';
when S3 =>
Qn <= S5;
SV_ID <= '1';
SBUTTON_IN <= '0';
MUX_SEL <= '0';
when S4 =>
if (EOFS='0') then
    Qn <= Qp;
else
    Qn <= S0;
end if;
SV_ID <= '0';
SBUTTON_IN <= '0';
MUX_SEL <= '1';
when S5 =>
if (EOFA='0') then
    Qn <= Qp;
else
    Qn <= S0;
end if;
SV_ID <= '0';
SBUTTON_IN <= '0';
MUX_SEL <= '0';

```

```

        when others =>

            Qn <= S0;

            SV_ID <= '0';

            SBUTTON_IN <= '0';

            MUX_SEL <= '0';

        end case;
end process Combinacional;

secuencial: process(RST,CLK)
    begin
        if (RST='0') then
            Qp <= S0;
        elsif (CLK'event and CLK='1') then
            Qp <= Qn;
        end if;
    end process secuencial;
end architecture compuesta;

library IEEE;
use IEEE.std_logic_1164.all;

entity BATAS_COMB is
    port(
        BATA1:  in std_logic;
        BATA2:  in std_logic;
        BATA3:  in std_logic;
        BATAS_IN:  out std_logic
    );

```

```

end BATAS_COMB;

architecture simple of BATAS_COMB is

begin

  Combinacional: process (BATA1,BATA2,BATA3)

  begin

    BATAS_IN <= BATA1 or BATA2 or BATA3;

  end process Combinacional;

end simple;

library IEEE;

use IEEE.std_logic_1164.all;

entity PROTOCOLO_ACCESO_TESIS is

port(

RST:          in std_logic;  --Reset maestro

CLK:          in std_logic;  --Reloj maestro

SV_ID:        in std_logic;

SWLE:         in std_logic;

SWLI:         in std_logic;

PIE_1:        in std_logic;

PIE_2:        in std_logic;

MANOS:        in std_logic;

BATA1:        in std_logic;

BATA2:        in std_logic;

BATA3:        in std_logic;

BATAS_SALIDA: out std_logic;

CEE:          out std_logic;

CEI:          out std_logic;

VALV:        out std_logic;

```

```

VENT:      out std_logic;
C1P:      out std_logic;
C2M:      out std_logic;
C3B:      out std_logic;
IPEV:     out std_logic;
IPER:     out std_logic;
IPIV:     out std_logic;
IPIR:     out std_logic;
EOFA:     out std_logic
);
end PROTOCOLO_ACCESO_TESIS;
architecture compuesta of PROTOCOLO_ACCESO_TESIS is
component PUERTA_EXTERIOR_ACCESO
port(
RST:      in  std_logic;
CLK:      in  std_logic;
SWLE:     in  std_logic;
STT_PE:   in  std_logic;
CEE:      out std_logic;
VENT:     out std_logic;
EOPE:     out std_logic
);
end component;
component PIES_ACCESO
port(
RST:      in  std_logic;
CLK:      in  std_logic;

```

```

PIE1:    in std_logic;
PIE2:    in std_logic;
STT_PIES: in std_logic;
EOPT:    in std_logic;
C1P:     out std_logic;
EOPIES:  out std_logic
);
end component;

component TARJA_ACCESO
port(
RST:     in std_logic;
CLK:     in std_logic;
MANOS:   in std_logic;
STT_MS:  in std_logic;
EOPT:    in std_logic;
VALV:    out std_logic;
C2M:     out std_logic;
EOMS:    out std_logic
);
end component;

component BATAS_ACCESO
port(
RST:     in std_logic;
CLK:     in std_logic;
H:       in std_logic;
STT_BATAS: in std_logic;
BATA1:   in std_logic;

```

```
BATA2:    in std_logic;
BATA3:    in std_logic;
EOPT:     in std_logic;
C3B:      out std_logic;
EOBC3B:   out std_logic;
BATAS_SALIDA: out std_logic
```

```
);
```

```
end component;
```

```
component FOCOS_ACCESO
```

```
port(
```

```
RST:      in std_logic;
```

```
CLK:      in std_logic;
```

```
V_ID:    in std_logic;
```

```
SWLE:    in std_logic;
```

```
SWLI:    in std_logic;
```

```
EOBC3B:  in std_logic;
```

```
IPEV:    out std_logic;
```

```
IPER:    out std_logic;
```

```
IPIV:    out std_logic;
```

```
IPIR:    out std_logic;
```

```
EOFF:    out std_logic
```

```
);
```

```
end component;
```

```
component PUERTA_INTERIOR_ACCESO
```

```
port(
```

```
RST:      in std_logic;
```

```
CLK:      in std_logic;
```

```

SWLI:      in std_logic;

STT_PI:    in std_logic;

CEI:      out std_logic;

EOPI:     out std_logic

);

end component;

signal EOPE,EOPIES,EOMS,EOPT,EOBC3B: std_logic;

begin

U0: PUERTA_EXTERIOR_ACCESO port map (RST,CLK,SWLE,SV_ID,CEE,VENT,EOPE);

U1:      FOCOS_ACCESO      port      map
(RST,CLK,SV_ID,SWLE,SWLI,EOBC3B,IPEV,IPER,IPIV,IPIR,EOFA);

U2: PIES_ACCESO port map (RST,CLK,PIE_1,PIE_2,EOPE,EOPT,C1P,EOPIES);

U3: TARJA_ACCESO port map (RST,CLK,MANOS,EOPIES,EOPT,VALV,C2M,EOMS);

U4:      BATAS_ACCESO      port      map
(RST,CLK,SV_ID,EOMS,BATA1,BATA2,BATA3,EOPT,C3B,EOBC3B,BATAS_SALIDA);

U5: PUERTA_INTERIOR_ACCESO port map (RST,CLK,SWLI,EOBC3B,CEI,EOPT);

end architecture;

library IEEE;

use IEEE.std_logic_1164.all;

entity PROTOCOLO_SALIDA_TESIS is

port(

RST:      in std_logic;  --Reset maestro

CLK:      in std_logic;  --Reloj maestro

BUTTON_IN:      in std_logic;

SWLE:      in std_logic;

SWLI:      in std_logic;

SV_ID:      in std_logic;

BATA1:      in std_logic;

```

```

BATA2:          in std_logic;
BATA3:          in std_logic;
CEI:           out std_logic;
CEE:           out std_logic;
VENT:          out std_logic;
C3B:           out std_logic;
IPEV:          out std_logic;
IPER:          out std_logic;
IPIV:          out std_logic;
IPIR:          out std_logic;
EOFS:          out std_logic;
BATAS_ACCESO:  out std_logic
);
end PROTOCOLO_SALIDA_TESIS;

architecture compuesta of PROTOCOLO_SALIDA_TESIS is
component PUERTA_INTERIOR_SALIDA
port(
RST:          in std_logic;
CLK:          in std_logic;
SWLI:         in std_logic;
BUTTON_IN:   in std_logic;
CEI:          out std_logic;
EOPI:         out std_logic
);
end component;

component BATAS_SALIDA
port(

```

```

RST:      in std_logic;
CLK:      in std_logic;
H:                in std_logic;
STT_BATAS:      in std_logic;
BATA1:    in std_logic;
BATA2:    in std_logic;
BATA3:    in std_logic;
EOPT:     in std_logic;
C3B:      out std_logic;
EOBC3B:   out std_logic;
BATAS_ACCESO: out std_logic
);
end component;

component FOCOS_SALIDA
port(
RST:      in std_logic;
CLK:      in std_logic;
BUTTON_IN: in std_logic;
V_ID:    in std_logic;
SWLE:    in std_logic;
SWLI:    in std_logic;
EOBC3B:  in std_logic;
IPEV:    out std_logic;
IPER:    out std_logic;
IPIV:    out std_logic;
IPIR:    out std_logic;
EOFS:    out std_logic

```

```

);
end component;

component PUERTA_EXTERIOR_SALIDA

port(

RST:      in std_logic;

CLK:      in std_logic;

V_ID:          in std_logic;

EOBC3B:   in std_logic;

SWLE:     in std_logic;

SWLI:     in std_logic;

CEE:      out std_logic;

VENT:     out std_logic;

EOPE:          out std_logic

);

end component;

signal EOPI,EOBC3B,EOPT: std_logic;

begin

U0: PUERTA_INTERIOR_SALIDA port map(RST,CLK,SWLI,BUTTON_IN,CEI,EOPI);

U1:          BATAS_SALIDA          port          map
(RST,CLK,BUTTON_IN,EOPI,BATA1,BATA2,BATA3,EOPT,C3B,EOBC3B,BATAS_ACCESO);

U2:          FOCOS_SALIDA          port          map
(RST,CLK,BUTTON_IN,SV_ID,SWLE,SWLI,EOBC3B,IPEV,IPER,IPIV,IPIR,EOPS);

U4:          PUERTA_EXTERIOR_SALIDA          port          map
(RST,CLK,SV_ID,EOBC3B,SWLE,SWLI,CEE,VENT,EOPT);

end architecture;

library IEEE;

use IEEE.std_logic_1164.all;

```

```

entity FSM_STT_ZIGBEE is
port(
RST:      in std_logic;
CLK:      in std_logic;
EOFS,EOFA:  in std_logic;
STT_ZB,EOZB:  out std_logic
);
end FSM_STT_ZIGBEE;

architecture compuesta of FSM_STT_ZIGBEE is
type FSM_STATES is (S0,S1,S2);
signal Qp,Qn: FSM_STATES;
begin
Combinacional: process (Qp,EOFS,EOFA)
begin
    case Qp is
        when S0 =>
            if (EOFS='1') then
                Qn <= S1;
            elsif (EOFA='1') then
                Qn <= S2;
            else
                Qn <= Qp;
            end if;
            STT_ZB <= '0';
            EOZB <= '0';
        when S1 =>
            Qn <= S0;
    end case;
end process;
end architecture;

```

```

        STT_ZB <= '1';
        EOZB <= '1';
    when S2 =>
        Qn <= S0;
        STT_ZB <= '1';
        EOZB <= '1';
    when others =>
        Qn <= S0;
        STT_ZB <= '0';
        EOZB <= '0';
    end case;
end process Combinacional;
secuencial: process(RST,CLK)
    begin
        if (RST='0') then
            Qp <= S0;
        elsif (CLK'event and CLK='1') then
            Qp <= Qn;
        end if;
    end process secuencial;
end architecture compuesta;
library IEEE;
use IEEE.std_logic_1164.all;
entity DRIVER_BUTTON_IN is
    port(
        RST:    in std_logic;
        CLK:    in std_logic;

```

```

BUTTON_IN:      in std_logic;
Z_100ms:       in std_logic;
S_BUTTON_IN:   out std_logic;
OPC_CONT_100ms: out std_logic_vector(1 downto 0)
);
end DRIVER_BUTTON_IN;

architecture simple of DRIVER_BUTTON_IN is
type FSM_STATES is (S0,S1,S2,S3,S4,S5,S6);
signal Qp,Qn: FSM_STATES;

begin
Combinacional: process (Qp,BUTTON_IN,Z_100ms)
begin
    case Qp is

        when S0 =>
            if (BUTTON_IN='1') then
                Qn <= Qp;
            else
                Qn <= S1;
            end if;
            S_BUTTON_IN <= '0';
            OPC_CONT_100ms <= "11";
            when S1 =>
                if (Z_100ms = '0') then
                    Qn <= Qp;
                else
                    Qn <= S2;

```

```

        end if;

        S_BUTTON_IN <= '0';

        OPC_CONT_100ms <= "01";

        when S2 =>

        Qn <= S0;

        S_BUTTON_IN <= '1';

        OPC_CONT_100ms <= "11";

        when others =>

        QN <= S0;

        S_BUTTON_IN <= '0';

        OPC_CONT_100ms <= "11";

        end case;

    end process Combinacional;

secuencial: process(RST,CLK)

    begin

        if (RST='0') then

            Qp <= S0;

        elsif (CLK'event and CLK='1') then

            Qp <= Qn;

        end if;

    end process secuencial;

end simple;

library IEEE;

use IEEE.std_logic_1164.all;

entity MUX_ACTUADORES is

    port(

        X1:          in std_logic;

```

```

X2:          in std_logic;

S0:  in std_logic;

Y:          out std_logic

);

end MUX_ACTUADORES;

architecture simple of MUX_ACTUADORES is

begin

process (S0,X1,X2)

begin

    case S0 is

        when '0' => Y <= X1;

        when others => Y <= X2;

    end case;

end process;

end architecture simple;

library IEEE;

use IEEE.std_logic_1164.all;

use IEEE.std_logic_arith.all;

use IEEE.std_logic_unsigned.all;

entity BASE_TIEMPO_100ms is

port(

RST:        in std_logic;           --Reset maestro

CLK:        in std_logic;           --Reloj maestro

OPC_CONT_100ms: in std_logic_vector(1 downto 0); --Control del contador

Z_100ms:    out std_logic           --Bandera de Fin de Cuenta

);

end BASE_TIEMPO_100ms;

```

```

architecture compuesta of BASE_TIEMPO_100ms is
signal QP,QN: std_logic_vector(24 downto 0);  --estados de contador

begin

comparador:process(QP)

begin

    if (QP="10111110101111100000111111") then

        Z_100ms <= '1';

    else

        Z_100ms <= '0';

    end if;

end process comparador;

combinacional:process(QP,OPC_CONT_100ms)

begin

    case OPC_CONT_100ms is

        when "00" =>          --Hold

            QN<=QP;

        when "01" =>          --Count

            QN<=QP+1;

        when "10" =>          --Clear

            QN<=(others=>'0');

        when others =>        --Clear

            QN<=(others=>'0');

    end case;

    --Q <= Qp;

end process combinacional;

secuencial:process(RST,CLK)

begin

```

```

if(RST='0') then
    QP<=(others=>'0');
elsif(CLK'event and CLK='1') then
    QP<=QN;
end if;
end process secuencial;
end architecture compuesta;

```

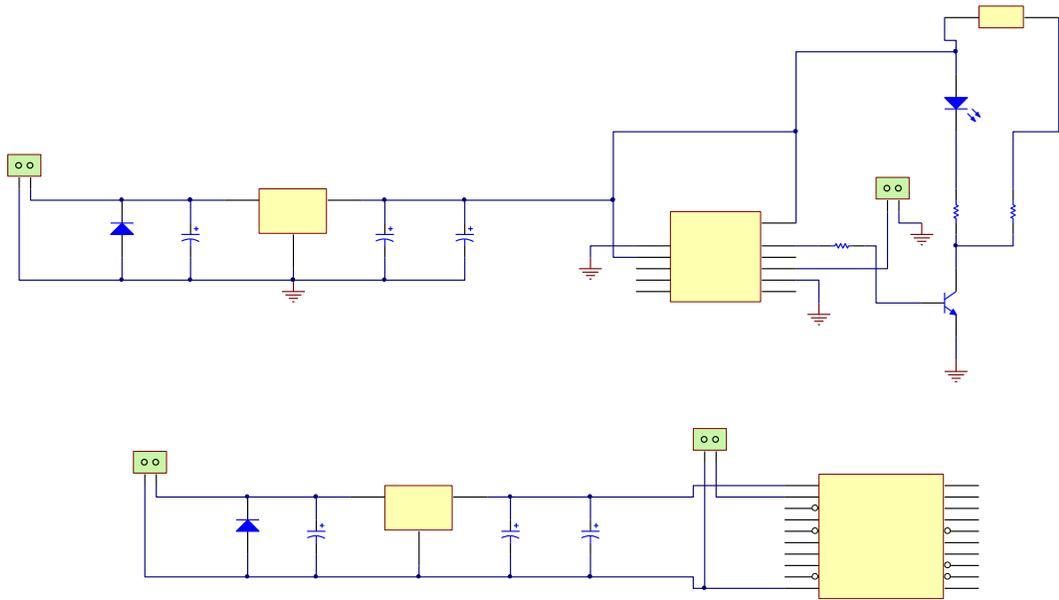


Diagrama eléctrico RFID y módulo de TXRX ZigBee.

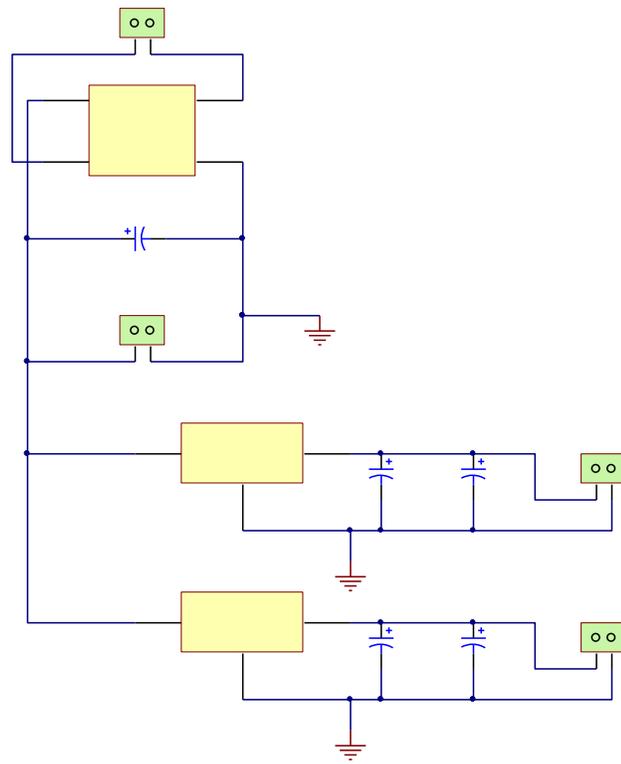


Diagrama Fuente de Voltaje 18V y 12V de salida.

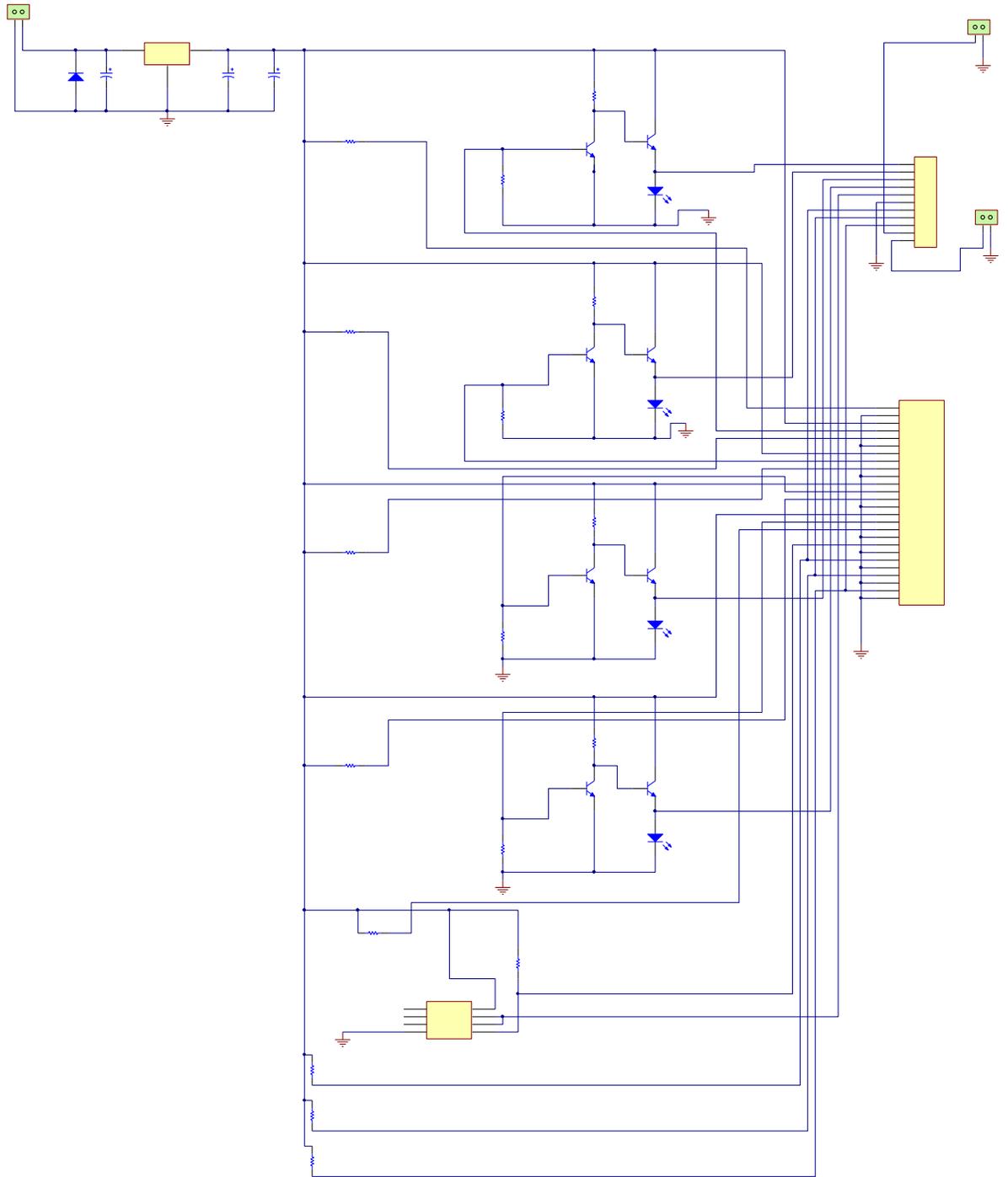


Diagrama Etapa de Control.

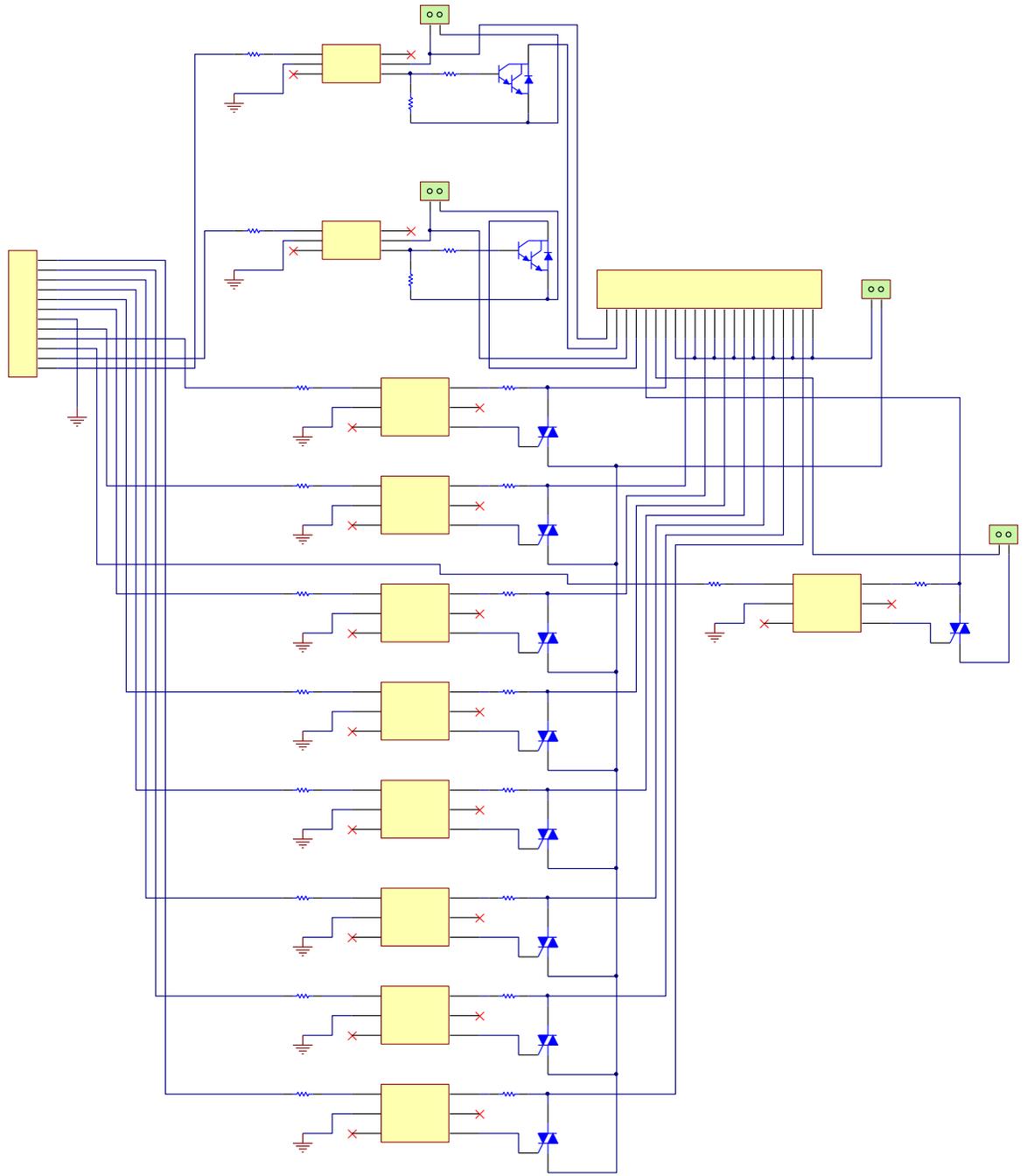


Diagrama Etapa de Potencia.