Universidad Acción Pro Educación y Cultura

Decanato de Ingeniería e Informática

Escuela de Ingeniería

Tesis de Grado para Optar por el Título de:
Ingeniero Electrónico Mención Comunicaciones

PROPUESTA DE UN SISTEMA DE SEGURIDAD PARA PANELES BLINDADOS MEDIANTE EL USO DE LA CPU BEAGLEBONE BLACK, REPÚBLICA DOMINICANA, AÑO 2015.

Sustentante:

Br. Denisse Margarita Nin Mateo 2001-1675

Asesor:

Ing. Luis Pérez Méndez

.

Los conceptos emitidos en el Presente trabajo, es de la exclusiva Responsabilidad de quien sustenta.

Distrito Nacional
República Dominicana

Julio 2015

RESUMEN

El proyecto que se mostrará a continuación, se desarrolló como proceso de investigación enfocado en un sistema de seguridad para los paneles blindados que colocan las compañías distribuidora de energía eléctrica, mostrando las irregularidades de seguridad que pueda mostrar dichos paneles y permitiendo el control del acceso a los mismos; se implementó un CPU, el cual permite tener un sistema de alarma y monitoreo de los paneles en distintas localidades.

Para poder llevar a cabo dicho proyecto, se realizó una búsqueda de los avances, aplicaciones y temáticas relacionadas con los sistemas embebidos, la programación en distintos lenguajes para así lograr los objetivos planteados. También fue necesario realizar el estudio del hardware que permitió llevar a cabo la asociación de los diferentes elementos conformantes y así lograr el funcionamiento e implementación del dispositivo.

Este proyecto, está conformado por diferentes áreas de la tecnología que requieren del uso de herramientas como JavaScript, C#, y la tarjeta electrónica BeagleBone Black. La primera herramienta es un lenguaje de programación orientada a objetos muy parecida a la sintaxis del lenguaje de programación C. La segunda herramienta es el lenguaje de marcado para la elaboración de aplicaciones y programas. El BeagleBone Black es un sistema embebido que

mediante programación se puede obtener aplicaciones como la de seguridad y acceso como se plantea. Por otra parte, se tendrá un circuito de interfaz entre los componentes del sistema y el BeagleBone Black que permitirá controlar las entradas y salidas del sistema para el manejo de los distintos componentes.

Finalmente después de haber terminado la aplicación del hardware y el desarrollo del software, se mostrarán los resultados obtenidos que se llevaron a cabo.

Palabras clave: C#, JavaScript, Aplicaciones, Software, Hardware, BeagleBone Black.

DEDICATORIAS

A Dios por derramar su bendición sobre mí y llenarme de su fuerza y su bondad en los momentos difíciles.

A mis amores eternos:

Mi madre Denia Mateo: Motor y eje principal de mi familia. Gracias por tu amor, apoyo incondicional, devoción y confianza.

Mi padre Mario Nin: Por haberme acompañado en los 15 primeros años de mi vida. Seguirá siendo para mí un honor ser tu hija. Gracias por tu amor donde quiera que estés. Los amo!!

A mis hermanos Mario y Teurys

A mis tías y tíos: En especial a Tía Julita por acompañar a mi madre en los momentos difíciles y ser su apoyo incondicional.

A mis primos **Manny Fresh y Nabo** que más que primos son los hermanos pequeños que la vida me regaló.

A mi querido y siempre amor!!

A mi Basha!!

AGRADECIMIENTOS

A Dios siempre. Infinitas Gracias al Todopoderoso por haberme dado la sabiduría y el entendimiento para poder llegar al final de mi carrera, por proveerme de todo lo necesario para salir adelante.

A la Universidad APEC

A mis maestros de la carrera por sus conocimientos, consejos, confianza y formación en el área de la Ingeniería. De manera especial a los ingenieros Emín Rivera, Porfirio Sánchez, Pérez Burdier, Edward Báez, José Mesa y en especial a mi asesor de tesis: Luis Pérez Méndez, clave en la realización de este trabajo. También a los profesores Rafael Joa y Ricardo Peguero.

A mis compañeros de trabajo y amigos

Que siempre estuvieron pendientes a lo largo de este proceso, brindando su apoyo incondicional. En especial al Ing. Mario Montgomery y el Lic. Víctor Ventura.

Gracias a esas personas maravillosas que me acompañaron en una parte de mi vida, llenándola de felicidad, amor y consejos. Y en especial a Cuchito por haber formado parte de este proyecto!

Por todo lo que significa concluir este trabajo y cerrar ciclos en mi vida y a la espera de abrir nuevos horizontes…….

Denisse Margarita

TABLA DE CONTENIDOS

RESUMEN ***2***

DEDICATORIAS ..... ***iv***

AGRADECIMIENTOS ..... ***v***

TABLA DE CONTENIDOS ..... ***vi***

LISTADO DE TABLAS ..... ***ix***

LISTADO DE FIGURAS ..... ***x***

LISTADO DE ANEXOS Y O APÉNDICES ..... ***xii***

ANEXOS ..... ***xii***

APÉNDICES ..... ***xii***

GLOSARIO ..... ***xiii***

INTRODUCCION ..... ***xvi***

CAPÍTULO 1. SISTEMA DE ALARMA Y MONITOREO ***18***

1.1 Sistemas de alarma, estructuras y características **18**

1.1.1 Sistema de alarma. 18

1.1.2 Estructura de un sistema de alarma. 19

1.1.3 Características. 21

1.1.3.1 Sensores, características y tipos. 23

1.1.3.2 Tipos de sensores. 27

Interruptores de Proximidad 28

Interruptores limitswitch o final de carrera 28

Interruptor Magnético 29

Detectores de Movimiento 31

Sensores Contra Incendio 34

- Detectores de Humo 34

Funcionamiento de Detectores de Humo Fotoeléctricos 38

- Detector de Humo Fotoeléctrico por Dispersión de Luz 38
- Detector de Humo Fotoeléctrico por Oscurecimiento 39

Sensores de Temperatura 39

Sensores de Vibración 41

Acelerómetro 41

Sensores de velocidad 42

1.2 Definición y características de sistemas de monitoreo 44
1.2.1 Central de monitoreo. 44
1.2.2 Características de una central de monitoreo. 47

1.3 Microcomputadores, estructura y características 48
1.3.1 Microcomputadores. 48

1.4 Sistemas embebidos 56

1.5 BeagleBone Black, teoría y aplicaciones 64

1.6 Interfaz gráfica, características 68

CAPÍTULO 2. DISEÑO DEL SISTEMA DE ALARMA Y MONITOREO DEL PROYECTO 72

2.1 Estructura de los paneles blindados, criterios para selección del área donde se implementaran el sistema 72
2.1.1 Ubicación (localización) de los paneles. 72
2.1.2 Anclaje del panel. 73
2.1.3 Obra civil. 74
2.1.4 Especificaciones técnicas de instalación para los paneles tipo **A** (socket) y tipo **B** (bottom). 75
2.1.5 Especificación para la construcción de paneles. 77
2.1.6 Especificaciones técnicas para construcción de paneles blindados tipo **C**. 78
2.1.7 Llavines para los paneles. 84
2.1.8 Pintura de panel. 85
2.1.9 Identificación del panel. 85
2.1.10 Instalación del sistema. 87
2.1.11 Sistema de respaldo de energía. 87

2.2 Selección de los componentes que estructuraran el sistema 88
2.2.1 Interruptores magnéticos. 88
2.2.2 Sensores de vibración. 90
2.2.2 Cerradura eléctrica. 91

2.3 Integración de los componentes del sistema con la herramienta BeagleBone Black 94
2.3.1 Proceso de diseño del circuito impreso. 95
2.3.2 Diseño del circuito impreso de interfaz. 97

2.4 Diseño de la interfaz gráfica 100

2.5 Implementación del sistema 104

CAPÍTULO 3.VALIDACIÓN DEL DISEÑO DEL SISTEMA DE ALARMA Y MONITOREO 107

3.1. Descripción de las pruebas del sistema 107
3.1.1. Pruebas del hardware. 107
3.1.2 Pruebas del programa. 113

Creación de localidades de Paneles 116
Validación de la conexión..... 119
Códigos de validación de accesos..... 120
Visualización de estados..... 122

3.2 Resultados de las pruebas.....123

3.3 Comparación de las pruebas con los cálculos del diseño.....124

CONCLUSIONES.....125

RECOMENDACIONES.....127

BIBLIOGRAFICAS.....128

APÉNDICE.....133

ANEXOS 187

LISTADO DE TABLAS

TABLA 1: CARACTERÍSTICAS DE BEAGLEBONE BLACK 66
TABLA 2: MUESTRA LA LEYENDA DE LOS PINES DONDE LOS COMPONENTES DEL SISTEMA ESTÁN CONECTADOS. 112

LISTADO DE FIGURAS

FIGURA 1: ESTRUCTURA DE UN SISTEMA DE ALARMA. 21
FIGURA 2: ESQUEMA FIN DE CARRERA. 29
FIGURA 3: ESTRUCTURA DEL REED SWITCH OFF 30
FIGURA 4: ESTRUCTURA DEL REED SWITCH ON 30
FIGURA 5: ESTRUCTURA DEL REED SWITCH 31
FIGURA 6: DETECTOR DE MOVIMIENTO. 32
FIGURA 7: DIAGRAMA DE UN SENSOR FOTOELÉCTRICO. 32
FIGURA 8: EFECTO FOTOELÉCTRICO EN UNIÓN P-N[1] 34
FIGURA 9: EMISIÓN DE LOS RAYOS DENTRO DE UNA CÁMARA DE IONIZACIÓN. 35
FIGURA 10: MOVIMIENTO DE LAS PARTÍCULAS AL SER POLARIZADA. 36
FIGURA 11: PARTÍCULAS EN COLISIÓN EN CÁMARA DE IONIZACIÓN CON EL AIRE IONIZADO. 36
FIGURA 12: PARTES DE UN DE DETECTOR DE CÁMARA DOBLE. 37
FIGURA 13: CÁMARA DE DETECCIÓN CON PARTÍCULAS DE COMBUSTIÓN. 38
FIGURA 14: TRAYECTORIA DEL HAZ DE LUZ Y SU INCIDENCIA EN LA PRESENCIA DE HUMO PARA ACTIVAR LA ALARMA. 39
FIGURA 15: ACCIONAMIENTO DE LA ALARMA DE INCENDIO POR LA INTERRUPCIÓN DEL HAZ DE LUZ. 39
FIGURA 16: ACELERÓMETRO PIEZO-ELÉCTRICO 42
FIGURA 17: ACELERÓMETRO PIEZO-ELÉCTRICO 44
FIGURA 18: ESTRUCTURA DE UNA CENTRAL DE MONITOREO. 46
FIGURA 19: MICROCOMPUTADOR. 56
FIGURA 20: EJEMPLO DE UN BOARD EMBEBIDO. 58
FIGURA 21: VISIÓN DE LOS SISTEMAS EMBEBIDOS AL 2020 61
FIGURA 22: TARJETA BEAGLEBONE BLACK. 67
FIGURA 23: DIAGRAMA EN BLOQUE DEL BEAGLEBONE BLACK. 68
FIGURA 24: BASE DE CONCRETO DEL PANEL 74
FIGURA 25: SEÑALIZACIÓN DE LOS PANELES. 76
FIGURA 26: PRESENTACIÓN EXTERIOR DEL PANEL. 79
FIGURA 27: PRESENTACIÓN PANELES TIPO A, B Y C. 80
FIGURA 28: ESQUEMA DE CONSTRUCCIÓN PANEL TIPO A. 81
FIGURA 29: ESQUEMA DE CONSTRUCCIÓN PANEL TIPO B. 81
FIGURA 30: ESQUEMA DE CONSTRUCCIÓN PANEL TIPO A. 82
FIGURA 31: ESQUEMA DE CONSTRUCCIÓN PANEL TIPO B. 83
FIGURA 32: ESQUEMA DE CONSTRUCCIÓN PANEL TIPO C. 83
FIGURA 33: ESQUEMA DE CONSTRUCCIÓN PANEL TIPO C. 84
FIGURA 34: LLAVINES DE LOS PANELES 84
FIGURA 35: SENSOR MAGNÉTICO DE PUERTAS COMERCIALES ENROLLABLES. 89
FIGURA 36: SENSOR MAGNÉTICO DE PUERTAS COMERCIALES ENROLLABLES. 89

FIGURA 37: SENSOR DE VIBRACIÓN. 90
FIGURA 38: CERRADURA ELÉCTRICA. 92
FIGURA 39: ESQUEMÁTICO PRINCIPAL DEL CIRCUITO IMPRESO. 96
FIGURA 40: ESQUEMÁTICO DE LA INTERCONEXIÓN DE COMPONENTES DE ENTRADA. 97
FIGURA 41: DISEÑO PCB DEL CIRCUITO DE INTERFAZ. 99
FIGURA 42: MUESTRA EL ENTORNO VISUAL DEL CLOUD9. 101
FIGURE 43: CONEXIÓN DEL BEAGLEBONE BLACK A LA FUENTE DE ALIMENTACIÓN 108
FIGURA 44: PÁGINA DE CONFIGURACIÓN DE PUERTOS DEL ROUTER D-LINK DIR-601 110
FIGURA 45: CIRCUITO DE INTERFAZ ENTRE EL BEAGLEBONE Y LOS COMPONENTES DEL SISTEMA. 111
FIGURA 46: ABRIENDO LA APLICACIÓN. 114
FIGURA 47: ENTORNO PRINCIPAL DEL PROGRAMA. RUBRO SECTOR. 115
FIGURA 48: SEÑALANDO BOTÓN AGREGAR. 117
FIGURA 49: VENTANA DE CREACIÓN DE PUNTO REMOTO. 118
FIGURA 50: SEÑALIZACIÓN DE BOTONES DE MODIFICAR Y ELIMINAR. 119
FIGURA 51: VENTANA DE CONEXIÓN EXITOSA. 120
FIGURA 52: VENTANA DE CONEXIÓN ERRÓNEA. 120
FIGURA 53: VENTANA DE AUTORIZACIÓN DEL CÓDIGO DE ACCESO. 121
FIGURA 54: VENTANA HISTÓRICO DE EVENTOS. 122
FIGURA 55: REPORTE DE EVENTOS. 124

LISTADO DE ANEXOS Y O APÉNDICES

ANEXOS

ANEXO 1: ANTEPROYECTO DE GRADO. 187
ANEXO 2: FORMULARIO DE SOLICITUD DE APROBACIÓN DE TEMA DE TRABAJO DE GRADO. 207
ANEXO 3: ESQUEMÁTICO BEAGLEBONE BLACK 209
ANEXO 4: DISEÑO COMPLETO CIRCUITO DE INTERFAZ 221
ANEXO 5: SISTEMA COMPLETO Y ARMADO DEL BEAGLEBONE BLACK 225

APÉNDICES

APÉNDICE A: CÓDIGO DEL BEAGLEBONE BLACK 133
APÉNDICE B: CÓDIGO DEL ENTORNO PRINCIPAL DE LA APLICACIÓN 141
APÉNDICE C: CÓDIGO DEL ESTILO DE LETRAS Y FIGURA DEL INTERFAZ GRÁFICA 155

GLOSARIO

1) **Aplicación:** Es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.

2) **Control Electrónico:** Consiste en la verificación de si una entidad (una persona, ordenador, etc...) solicitando acceso a un recurso tiene los derechos necesarios para hacerlo.

3) **Base de datos:** Una base de datos es una colección de datos organizados y estructurados según un determinado modelo de información que refleja no sólo los datos en sí mismos, sino también las relaciones que existen entre ellos.

4) **C#:** Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO(ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

5) **JavaScript:** Al igual que HTML, JavaScript es un lenguaje de programación que se puede utilizar para construir sitios Web y para hacerlos más interactivos. El lenguaje JavaScript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico.

6) **Interfaz:** Conjunto de métodos para lograr interactividad entre un usuario y una computadora.

7) **Corriente eléctrica:** Se da el nombre de corriente eléctrica a un desplazamiento de electrones a lo largo de un conductor entre cuyos extremos se aplica una diferencia de potencial.

8) **Router:** Conocido como enrutador o encaminador de paquetes, y españolizado como router es un dispositivo que proporciona conectividad a nivel de red o nivel tres en el modelo OSI.

9) **Servidor:** Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final.

10) **Breaker de Carga:** Componente de protección termo magnético que despeja las fallas por sobre corriente en la acometida, es decir en los conductores que van desde el panel hacia la carga del cliente.

11) **Breaker Principal o Mainbreaker:** Componente de protección termo magnético que protege el resto de la instalación de las fallas en ese tramo de la acometida principal y las barras del panel.

12) **Panel Porta Medidores:** Compartimiento auto soportado provisto de protección general, barraje, cubículo para medidores y breakers de protección y/o corte en el que se pueden instalar cuatro o más medidores.

13) **Panel:** Tablero, caja de metal o policarbonato, destinado a albergar medidores de energía, complementariamente posee dispositivo de protección de acometida principal y dispositivos de protección de acometidas de carga.

14) **Clientes en Baja Tensión (B.T.):** Son aquellos clientes conectados a las redes de tensión igual o inferior a 600 Voltios.

15) **Consumo:** Cantidad de energía utilizada por un suministro, puede ser con lectura real o acordada con el Cliente (Dependerá de la potencia y el tiempo de uso).

INTRODUCCION

Un sistema de seguridad conjuga un grupo de elementos conexos cuyo fin u objetivo primordial es instaurar cierto nivel de protección frente a riesgos o delitos que puedan perjudicar de forma negativa la integridad de una cierta población y/o bien.

Un sistema electrónico de alarmas pertenece a un plan de contingencia, debido a sus características, versatilidad, costo razonable y sobre todo a la adaptabilidad de nuevas tecnologías tanto en el campo de la electrónica, como en las telecomunicaciones y el manejo de la información.

El contenido de dicho trabajo está basado, en una investigación de campo y aplicación para resolver el problema de seguridad con que cuentan los Paneles Blindados que las distribuidoras eléctricas instalan a sus clientes para proteger los medidores y los posibles fraudes que se comenten en el mismo.

Por tal razón, se ha planteado un sistema de alarma utilizando la CPU BeagleBone Black como sistema embebido que permitirá realizar la aplicación de sistema de alarma. En dicho trabajo mostraremos las características de sistemas embebidos como el BeagleBone Black,

los componentes del sistema de alarma, las centrales de monitoreo y la aplicación de este conjunto de elementos.

CAPÍTULO 1. SISTEMA DE ALARMA Y MONITOREO

1.1 Sistemas de alarma, estructuras y características

1.1.1 Sistema de alarma.

El objetivo de un sistema de seguridad es la detección cualquier situación de riesgo que se presente en un determinado ambiente. Estos eventos pueden variar desde la detección de un intruso hasta el reporte del inicio de un incendio. Un sistema de seguridad no significa únicamente la detección de algún problema determinado, sino también, un evento como respuesta que logre poner sobre aviso a las personas correspondientes, ya sea el administrador del sistema o alguna empresa dedicada a la solución de estos problemas.

Todo sistema de seguridad, debe emitir una alarma que alerte a la persona responsable de cualquier desperfecto ocurrido en la empresa o en el hogar. Esta alarma, representada por cualquier tipo señal, sea sonido, imagen o texto, debe de llegar también a una central de control (la cual se detallará más adelante) ya sea dentro de la misma empresa o alguna empresa contratada externamente, para que se puedan tomar las medidas como llamar a la policía, a una agencia de seguridad o simplemente determinar que la situación no es de gravedad.

El sistema debe tomar en cuenta el tipo de situación que se presente para poder determinar si es necesario que se involucre a una empresa dedicada a la seguridad o simplemente requiere una solución sencilla que pueda llevar a cabo cualquier usuario.

Estos sistemas deben de ser completamente operables, confiables, que se les pueda proporcionar un fácil mantenimiento y que cuenten con algún sistema de medición que permita mantener el control de su funcionalidad. Además debe de operar de acuerdo a las tecnologías más recientes en el mercado, ya que esto dará confiabilidad al usuario y garantizará una mayor seguridad. (Lugo, 2001)

1.1.2 Estructura de un sistema de alarma.

Un sistema de alarma lo conforma un grupo de dispositivos interconectados que envían información a un panel principal donde se procesan todas las informaciones recibidas de cada uno de ellos para realizar las acciones necesarias dependiendo de lo que suceda en el entorno. Toda la integración de cada uno de los dispositivos, hacen que el sistema sea más confiable y más eficiente. “(ver fig.1)”Tiene como objeto componentes científicos, administrativos y tecnológicos que están enlazados por diferentes procesos de comunicación.

En general un sistema de alarma está conformado fundamentalmente por los siguientes elementos:

- **Detección** los sensores que monitorean el área donde están instaladas producen una señal eléctrica, sonora, visual cuando se ven perturbados de su condición de operación normal.

- **Control o panel principal**, Recibe las señales provenientes de los sensores, para procesarlas y determinar las fallas ocurridas.

- **Comunicación**, los cuales sirven de interfaz entre el usuario y el área que se quiere proteger.

- **Control y accionamiento**, realiza una actividad que ayuda a controlar las anomalías detectadas por el sistema, aunque estos no siempre solucionan el problema por sí solo, pero si ayudan a evitar mayores efectos.

Figura 1: Estructura de un Sistema de Alarma.
Fuente: itics.co

1.1.3 Características.

Para que un sistema de alarma sea eficiente y cumpla con los requerimientos para el cual se instala se debe tener ciertas características que ayudan a que la implementación del mismo sea lo más seguro posible. A continuación se muestran algunas de las características:

- Los sistemas de alarma deben contar con la emisión de alguna señal ya sea sonora, grafica o textual que alerte a la persona responsable y a su vez este comunicada con alguna central de monitoreo.

- El sistema debe contener los dispositivos necesarios que garanticen la mayor protección en los diferentes escenarios, por lo que es sumamente importante tomar en cuenta cualquier tipo de situación que pudiese presentarse.

- Estos sistemas deben de ser completamente operables, confiables, de fácil mantenimiento.

- Deben operar de acuerdo a las tecnologías más recientes en el mercado, ya que esto dará confiabilidad al usuario y garantizara una mayor seguridad.

- Es importante que los sistemas de seguridad cuenten con una interfaz amigable para el usuario. Planos de las ubicaciones de los dispositivos de seguridad o alguna representación visual del área a proteger.

- Se debe considerar un sistema de protección para que ninguna persona pueda manipularlo para fines ajenos a este.

- Considerar el respaldo de energía por fallas en el sistema eléctrico externo para evitar la pérdida de información.

1.1.3.1 Sensores, características y tipos.

Los sensores constituyen el principal medio de enlace entre los procesos industriales y los circuitos electrónicos encargados de controlarlos o monitorearlos. Se examinará los conceptos generales relacionados con los sensores eléctricos, haciendo énfasis en los principales tipos disponibles para aplicaciones industriales, sus principios de funcionamiento y la interpretación de sus características estáticas y dinámicas.

Los sensores o transductores, en general, son dispositivos que transforman una cantidad física cualquiera, por ejemplo la temperatura, en otra cantidad física equivalente, digamos un desplazamiento mecánico. Se hace referencia principalmente a los sensores eléctricos, es decir aquellos cuya salida es una señal eléctrica de corriente o voltaje, codificada en forma análoga o digital. Los sensores posibilitan la comunicación entre el mundo físico y los sistemas de control, tanto eléctricos como electrónicos, utilizándose extensivamente en todo tipo de procesos industriales y no industriales para propósitos de monitoreo, control y procesamiento.

En un sentido más amplio, el uso de los sensores no se limita solamente a la medición o la detección DC cantidades físicas. También pueden ser empleados para medir o detectar propiedades químicas y biológicas. Asimismo, la salida no siempre tiene que ser una señal eléctrica. Por ejemplo, muchos termómetros utilizan como sensor una lámina vi metálica, formada por dos metales con diferentes coeficientes de dilatación, la cual produce un desplazamiento (señal mecánica) proporcional a la temperatura (señal térmica).

En la actualidad, se utilizan en gran cantidad de aplicaciones y prácticamente en todos los sectores, desde los sensores industriales, utilizados para el control de procesos de fabricación, hasta en la medicina, donde se usan para la monitorización de parámetros propios del cuerpo humano.

La evolución en la electrónica de los últimos años ha propiciado una reducción de sus dimensiones, un menor consumo y la posibilidad de comunicarse de forma inalámbrica.

Características de los sensores

Entre las características técnicas destacables de un sensor se encuentran:

Rango de medida: dominio en la magnitud de medida en el que puede aplicarse el sensor.

Precisión: Es el error de medida máximo esperado.

Sensibilidad de un sensor: Relación entre la variación de la magnitud de salida y la variación magnitud entrada.

Rapidez de repuesta: Puede ser un tiempo fijo o depender de cuánto varié la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.

Derivas: Son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura, u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.

Repetitividad: Error esperado al repetir varias veces la misma medida.

Hay aspectos a cerca de los sensores que es necesario tener en cuenta:

- La naturaleza de la señal que el sensor – transductor genera: voltaje, rango de amplitud, respuesta en frecuencia, precisión necesaria, determinan el tipo de acondicionamiento de señal, convertidor A/D y cualquier otro hardware a utilizar.

- La influencia de las señales de ruido así como los efectos de carga del hardware de adquisición de datos sobre el sensor.

- La calibración del sensor con respecto a la variable física. Si la respuesta del sensor a los cambios de la variable física es lineal o no. Una calibración mal hecha va a producir mediciones erróneas.

- La interdependencia entre los distintos componentes del sistema de adquisición de datos, por ejemplo un sensor muy bueno, con un pobre convertidor A/D no sirve de casi nada.

- La precisión del sensor, esto es la capacidad de medir el mismo valor repetidas veces en idénticas condiciones.

- El tiempo de respuesta del sensor, es decir, el tiempo requerido para responder a un cambio brusco de la variable que está siendo sensada.

- El coeficiente de temperatura del sensor, el cual viene dado por el cambio que se produce en la respuesta del sensor debido al cambio en la temperatura a la cual se encuentra, por ejemplo el aumento en las corrientes de fuga y el voltaje offset de un amplificador, el aumento de la corriente en la oscuridad de un fotodiodo.

- La histéresis de un sensor, la cual se define como la dependencia de la salida del sensor de la respuesta anterior. Esta es muy común en sistemas magnéticos y mecánicos.

Existen varias formas de clasificar los sensores, por ejemplo se pueden clasificar por el principio físico de funcionamiento (inductivo, capacitivo, termoeléctrico o resistivo etc.), por la variable física medida (temperatura, presión, posición etc. por la capacidad de generar energía (activos) o de necesitar de un circuito de excitación (pasivos). (UNET, s.f)

1.1.3.2 Tipos de sensores.

En la actualidad existe una diversidad de sensores para aplicaciones de sistemas de alarmas y monitoreo como son los sensores de movimiento, sensores de contacto, sensores detectores de humo, de quebradura de vidrio y detectores de fase que conforman el sistema de alarma para abarcar las distintas áreas que se necesita proteger.

Clasificación de los sensores:

- De acción mecánica o de contacto
- De acción magnética
- De acción capacitiva

- Accionados por luz
- De acción ultrasónica
- De acción neumática

Dentro de estos sensores se encuentra:

Interruptores de Proximidad

Los sensores de proximidad son los más comunes y asequibles para la solución de detección de objetos que no se pueden tocar. A continuación los más comunes:

Interruptores limitswitch o final de carrera

Los interruptores de proximidad, también llamados limitswitch o final de carrera, son interruptores que al ser accionado mecánicamente, operan contactos eléctricos. Estos determinan la presencia, ausencia, paso y posicionamiento de un objeto, transformando un movimiento mecánico en una señal eléctrica.

Su funcionamiento está basado en un pequeño interruptor mecánico que requiere un contacto físico y una pequeña fuerza de acción para cerrar los contactos. Se colocan al final del recorrido de un elemento móvil. Internamente pueden contener interruptores normalmente abiertos (NA), cerrados (NC) o conmutadores dependiendo de la operación que cumplan al

ser accionados“(ver fig.2)”. Pueden trabajar a tensiones elevadas y son inmunes a la electricidad estática ya que no tienen imanes.

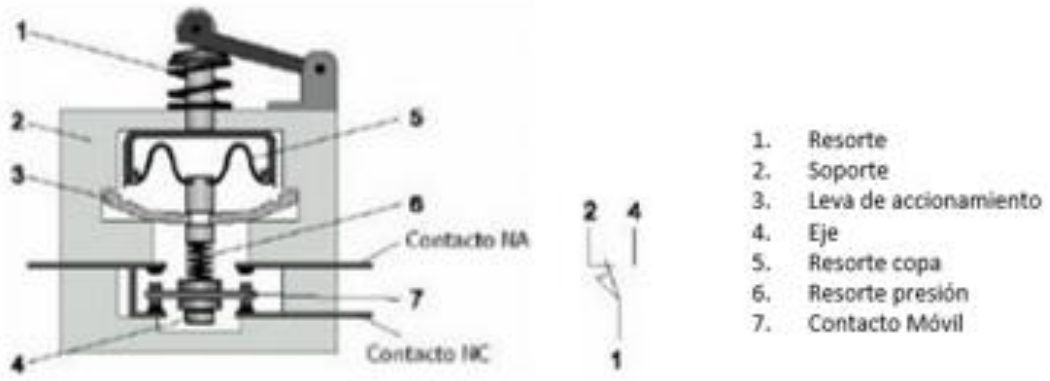


Figura 2: Esquema fin de carrera.
Fuente:http://es.slideshare.net/krloscr3/trabajo-sobre-sensores-para-uso-industrial

Interruptor Magnético

Los Reed Switches se usan en una gran cantidad de mercados y aplicaciones.

A continuación enumeramos algunas de ellas aplicadas a la electrónica del hogar, utilizados como sensores para:

- Detección del nivel de líquidos en sistemas de calefacción y humidificadores
- Detección de presión en los sensores de nivel de calentadores de agua
- Detección de posición en sensores de sistemas de seguridad, teléfonos móviles, en los interruptores de cepillos de dientes eléctricos y máquinas de afeitar

El **reed switch** se compone de dos filamentos ferromagnéticos (recubiertos con rodio o iridio) distanciados entre sí y sellados herméticamente en un tubo de cristal, el cual se llena con un gas inerte para prevenir la activación de los contactos. “(ver fig. 3)”.

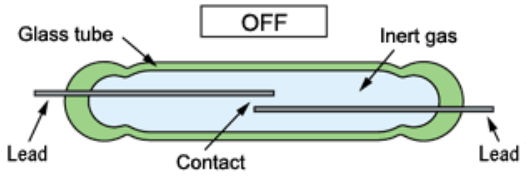


Figura 3: Estructura del Reed Switch off
Fuente: http://www.cika.com/productos/oki/

Se activa con el campo magnético de una bobina energizada o de un imán permanente que induce los polos norte (N) y sur (S) en los contactos, originando que éstos se cierren. Al remover el campo magnético, la elasticidad de los contactos ocasiona que se abra el circuito. “(ver fig.4)”.

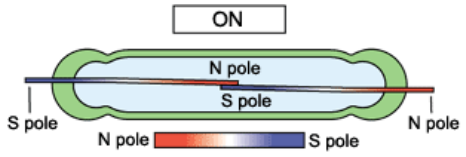


Figura 4: Estructura del Reed Switch On
Fuente: http://www.cika.com/productos/oki/

El reed switch de conmutación, presenta tres contactos de nombres COM (común), N.C. (normal cerrado) y N.O. (normal abierto). Sin un campo magnético presente, el circuito se encuentra mecánicamente establecido a través del par COM-N.C. Al acercar un imán sucede

que el COM es atraído hacia el contacto N.O. El N.C. no se ve afectado puesto que no es de material ferromagnético.“(ver fig.5)”.(Oki Sensor Device, 1996)

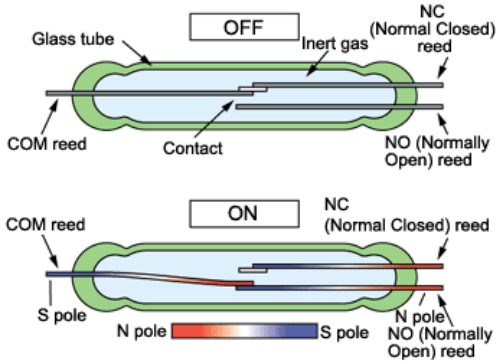


Figura 5: Estructura del Reed Switch
Fuente:http://www.cika.com/productos/oki/

Detectores de Movimiento

Los detectores de movimiento son interruptores electrónicos accionados por un sensor de rayos infrarrojos pasivos sensible a las radiaciones de calor que emiten las personas en movimiento. El detector de movimiento activa los servicios conectados (p. ej. el encendido de una lámpara) cuando detecta movimiento en el área de cobertura, la desactivación se efectúa con un retardo de tiempo regulable por el usuario. (ELECTRY PERRY, s.f)

Figura 6: Detector de Movimiento.
Fuente: www.brielco.net

Sensores Fotoeléctricos

Los sensores fotoeléctricos usan un haz de luz para detectar la presencia o la ausencia de un objeto. Esta tecnología es una alternativa ideal a sensores de proximidad inductivos cuando se requieren distancias de detección largas o cuando el ítem que se desea detectar no es metálico. Nuestros sensores fotoeléctricos satisfacen aplicaciones que se encuentran en muchas industrias, tales como manejo de materiales, empaquetado, procesamiento de alimentos y transporte. Para sus aplicaciones especiales que requieren un conector o cable no estándar, podemos personalizar las conexiones del sensor.(Allen-Bradley , 2015)

Diagrama de un sensor fotoeléctrico

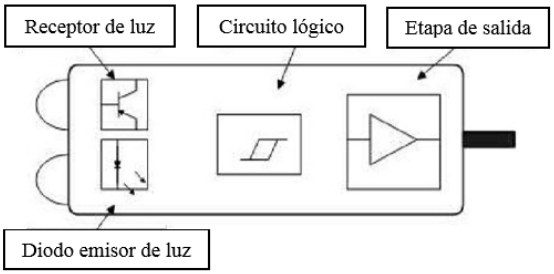


Figura 7: Diagrama de un sensor fotoeléctrico.
Fuente: http://es.scribd.com/doc/7853459/SENSORES-FOTOELECTRICOS#scribd

Su principio de funcionamiento está basado en la generación de un haz luminoso por parte de un foto emisor, que se proyecta bien sobre un foto receptor, o bien sobre un dispositivo reflectante. La interrupción o reflexión del haz por parte del objeto a detectar, provoca el cambio de estado de la salida de la fotocélula.(Cabrera, 2008).

Los sensores fotoeléctricos, tienen como función principal, la detección de todo tipo de objetos. Funcionan mediante la detección de un cambio en la cantidad de luz que es reflejada o bloqueada por el objeto que se desea detectar (objetivo). El cambio de luz puede ser producido por la presencia o ausencia del objetivo, o bien como resultado de un cambio de tamaño, forma, reflexividad o color del objetivo.(Automation, 2012)

El efecto fotovoltaico permite obtener una tensión eléctrica que es función de la intensidad de la radiación incidente sobre un material fotoconductor. Esto se da porque se ioniza una zona donde hay una barrera de potencial. Al poner en contacto un semiconductor **p** con uno tipo **n**, debido al movimiento térmico algunos electrones pasan de la zona **n** a la zona **p**, esto ocasiona que en una banda alrededor de la zona de contacto de los dos materiales sea eléctricamente neutra y esta parte neutra se opone a el paso de los electrones de un material a otro. Esto es similar a lo que pasa en la zona de unión de un Diodo semiconductor.

En los materiales fotoconductores este equilibrio puede ser roto por medio de radiación, visible o no, solo se requiere que la energía supere el ancho de banda bajo la acción del campo eléctrico en la zona de unión. “(ver fig. 8)”. La radiación incidente hace que aumente el número de huecos en el material **p**, lo que llega a generar un potencial en los extremos del material.(Rojas, 2008)

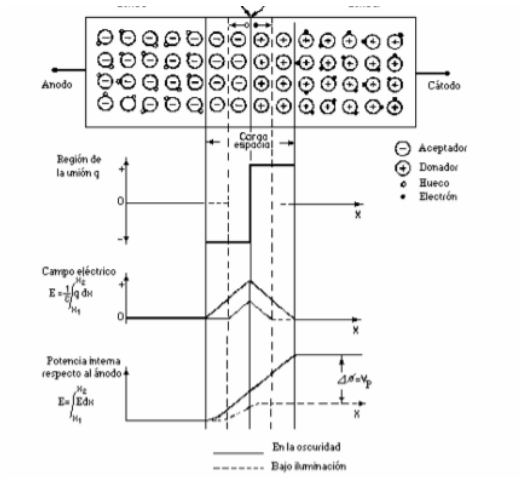


Figura 8: Efecto fotoeléctrico en unión p-n[1]
Fuente: www.sportsmobileforum.com

Sensores Contra Incendio

Estos son dispositivos de seguridad que detecta la presencia de un incendio y emite una señal acústica y eléctrica avisando del peligro. Atendiendo al método de detección que usan pueden ser de varios tipos:

- Detectores de Humo

Existen dos tipos básicos de detectores de humo en uso actualmente: los detectores por ionización y los detectores fotoeléctricos. Las cámaras de los sensores tienen diferentes principios de funcionamiento para detectar las partículas de combustión visibles o invisibles liberadas en un incendio.

Típicamente, una cámara de ionización consiste en dos placas cargadas eléctricamente y un material radioactivo (generalmente Americio 241) para ionizar el aire entre las placas "(ver fig.9)".El material radioactivo emite partículas que entran en colisión con las moléculas en el aire, desalojando los electrones de su órbita.

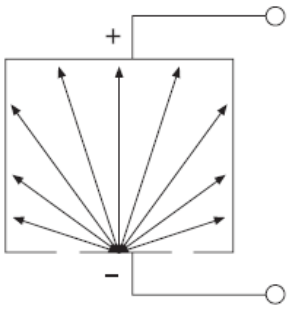


Figura 9: Emisión de los rayos dentro de una cámara de ionización.
Fuente: www.systemsensor.ca/es/docs/guides/A05-1046.pdf

Esto causa que esas moléculas se conviertan en iones cargados positivamente, y las moléculas que ganaron electrones se conviertan en iones negativos. Los iones positivos son atraídos a la placa de polaridad negativa, y los iones negativos a la placa de polaridad positiva "(ver fig.10)". De esta manera, la ionización genera una pequeña corriente que es medida por un circuito electrónico conectado a las placas (esta es la condición "normal" del detector).

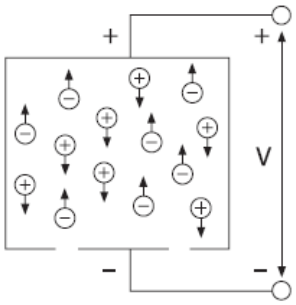


Figura 10: Movimiento de las partículas al ser polarizada.
Fuente: www.systemsensor.ca/es/docs/guides/A05-1046.pdf

Las partículas liberadas en la combustión son mucho más grandes que las moléculas de aire ionizadas. Cuando ingresan a la cámara de ionización, entran en colisión con las moléculas de aire ionizadas y se combinan con ellas "(ver fig.11)", como resultado de lo cual algunas partículas se cargan positivamente y otras negativamente. A medida que continúan combinándose, cada partícula grande se convierte en un punto de recombinación y así la cantidad total de iones en la cámara será menor. Al mismo tiempo, la corriente medida por el circuito también disminuirá y cuando sea inferior a un valor predeterminado, se generará una condición de alarma.

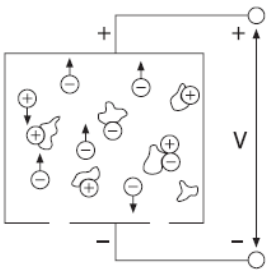


Figura 11: Partículas en colisión en cámara de ionización con el aire ionizado.
Fuente:www.systemsensor.ca/es/docs/guides/A05-1046.pdf

La humedad ambiente y la presión atmosférica influyen en el valor de la corriente de la cámara y crean un efecto similar al causado por el ingreso de las partículas de combustión. Para compensar la influencia de la humedad y la presión atmosférica, se creó la cámara doble de ionización.

En un detector de cámara doble, una cámara es utilizada para detección y está abierta al aire externo "(ver fig.12)", por lo cual en ella hay presencia de humedad ambiente, presión atmosférica y partículas liberadas por combustión. La otra cámara suministra un valor de referencia o comparación, ya que es afectada solamente por la humedad y la presión, ya que las partículas de combustión no pueden ingresar por los orificios de pequeño tamaño de esta cámara.

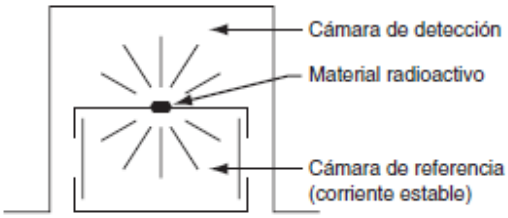


Figura 12: Partes de un de detector de cámara doble.
Fuente: www.systemsensor.ca/es/docs/guides/A05-1046.pdf

El circuito electrónico mide y compara la corriente de ambas cámaras. Como los cambios de humedad y presión atmosférica afectan por igual a ambas cámaras, la variación en una se compensa con la variación en la otra. Cuando las partículas de la combustión ingresan a la cámara de detección, la corriente disminuye y se produce un desfasaje de valores de corriente entre las dos cámaras, que es captado por el circuito de medición "(ver fig.13)". Hay varios factores que pueden influir en la detección de una cámara ionizada: polvo, condensación de

humedad, corrientes fuertes de aire e incluso insectos minúsculos, que podrían variar la medición del circuito como si fueran partículas de combustión.

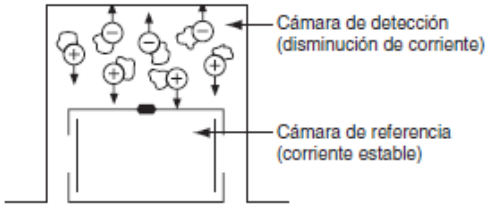


Figura 13: Cámara de detección con partículas de combustión.
Fuente: www.systemsensor.ca/es/docs/guides/A05-1046.pdf

Funcionamiento de Detectores de Humo Fotoeléctricos

El humo generado en un incendio bloquea u oscurece el medio en el que se propaga un haz de luz. También puede dispersar la luz cuando ésta se refleja y refracta en las partículas humo. Los detectores fotoeléctricos están diseñados para utilizar estos efectos a fin de detectar la presencia de humo.

- Detector de Humo Fotoeléctrico por Dispersión de Luz

La mayoría de los detectores de humo fotoeléctricos tienen cobertura localizada (puntual) y funcionan con el principio de dispersión de luz. El haz de un diodo emisor de luz (LED) incide en un área donde no puede ser captado bajo condiciones normales por un foto sensor, que generalmente es un fotodiodo "(ver fig.14)". Cuando hay presencia de humo en la trayectoria del haz, la luz incide sobre las partículas de humo y se refleja sobre el foto sensor, que al recibir la luz genera una señal "(ver fig.14)".

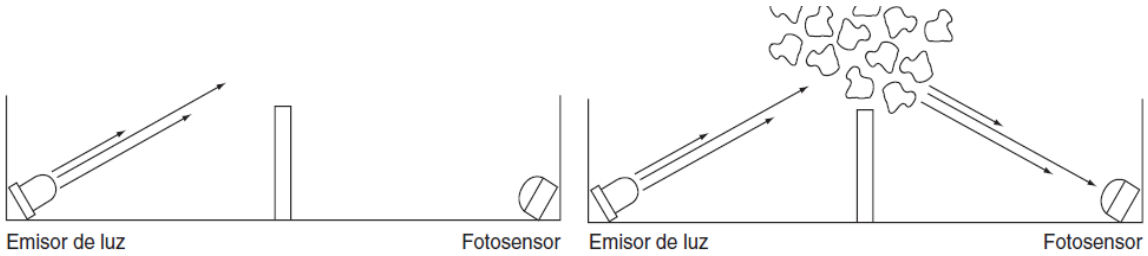


Figura 14: Trayectoria del haz de luz y su incidencia en la presencia de humo para activar la alarma.
Fuente: www.systemsensor.ca/es/docs/guides/A05-1046.pdf

- Detector de Humo Fotoeléctrico por Oscurecimiento

Este tipo de detector también utiliza un emisor de luz y un elemento foto sensor, tal como sería un fotodiodo. Cuando las partículas de humo bloquean parcialmente la trayectoria del haz de luz, se reduce la intensidad de luz recibida por el foto sensor. Esta variación es captada por un circuito electrónico que, al llegar al valor pre calibrado genera una señal de iniciación de alarma. Generalmente, los detectores por oscurecimiento utilizan un haz de luz que barre el área a proteger"(ver fig.15)".(System Sensor, 2004)

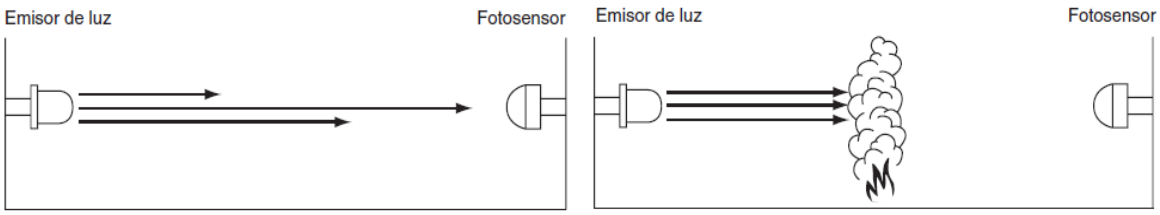


Figura 15: Accionamiento de la alarma de incendio por la interrupción del haz de luz.
Fuente: www.systemsensor.ca/es/docs/guides/A05-1046.pdf

Sensores de Temperatura

La temperatura es el parámetro físico más común que se mide en una aplicación electrónica, incluso en muchos casos en que el parámetro de interés no es la temperatura, ésta se ha de

medir para incluir indirectamente su efecto en la medida deseada. La diversidad de sus aplicaciones ha condicionado igualmente una gran proliferación de dispositivos sensores y transductores, desde la sencilla unión bimetálica de los termostatos, hasta los dispositivos semiconductores más complejos.

Tipos de Sensores de Temperatura

- **Termopares:**

Los termopares utilizan la tensión generada en la unión de dos metales en contacto térmico, debido a sus distintos comportamientos eléctricos.

- **Resistivos:**

Lo constituyen las RTD (Resistance Temperature Detector) o PT100 basadas en la dependencia de la resistividad de un conductor con la temperatura, están caracterizadas por un coeficiente de resistividad positivo PTC (Positive Termal Coefficient). También lo son las NTC (Negative Termal Coefficient), que se llaman termistores y están caracterizadas por un coeficiente de temperatura negativo.

- **Semiconductores:**

Se basan en la variación de la conducción de una unión p-n polarizada directamente.

Sensores de Vibración

Los sensores de vibración son acelerómetros que permiten trabajar en frecuencias altas, que es donde se produce principalmente la vibración.

Estas frecuencias y en consecuencia estos acelerómetros se utilizan para diferentes aplicaciones, pero hay una que destaca sobre el resto, y es el control preventivo o mantenimiento preventivo en maquinaria.

Estos sensores de vibración en combinación con el acondicionador o equipo de adquisición pertinente, nos permiten llevar a cabo una instalación de sensores para el registro y posterior análisis de las vibraciones, ver tendencias de desgaste o problemas de funcionamiento de la maquinaria, debido a un mayor nivel de vibración en ciertas frecuencias.(SENSINC, 2015)

Acelerómetro

Los acelerómetros o sensores de aceleración, están pensados para realizar una medida de aceleración o vibración, proporcionando una señal eléctrica según la variación física, en este caso la variación física es la aceleración o la vibración. Es el más común de los sensores de vibración.

Vienen en una variedad de diseños, y pueden detectar una amplia gama de diferentes vibraciones. Una de las versiones más populares de acelerómetro es el sensor piezoeléctrico, estos se someten a un esfuerzo mecánico.(SENSINC, 2015)

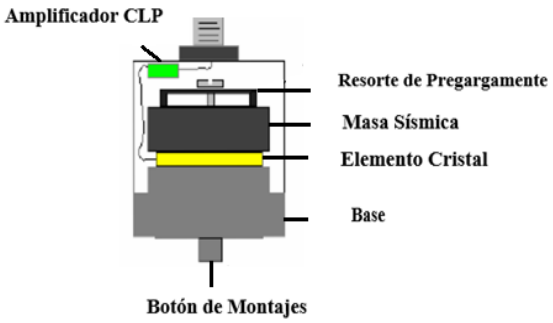


Figura 16: Acelerómetro Piezo-Eléctrico
Fuente: https://prezi.com/pswdl_-08vat/sensores-de-vibracion/

Mediante la medición de la cantidad de carga eléctrica que los acelerómetros piezoeléctricos emiten, se hace posible determinar la cantidad de vibración que pasa en la conexión.

Sensores de velocidad

El sensor de velocidad, mide la corriente que genera la bobina y es proporcional a la velocidad del movimiento. Se utiliza para el funcionamiento del velocímetro, la respuesta del ABS, dirección asistida, etc. Está hecho con una bobina de alambre más un imán, colocados de forma que al moverse el cárter, el imán permanece sin moverse. Se crea un movimiento relativo en el campo magnético y la bobina provoca una corriente que está en proporción a la velocidad del movimiento. Es auto generador, no necesita de aditamentos electrónicos para

funcionar. Posee una impedancia de salida eléctrica baja, que lo hace casi insensible a la inducción del ruido.

Detecta magnitudes químicas o físicas, a las que se les llama variables de instrumentación y las transforma en variables eléctricas. Está continuamente en contacto con la variable de instrumentación y adapta la señal que mide, para que sea interpretada por otro dispositivo Por ejemplo el sensor de velocidad detecta la variable eléctrica producida por la bobina. La variable de instrumentación detectada en este caso, es el movimiento del cárter. (Gómez, 2013)

Las variables de instrumentación son: aceleración, temperatura, intensidad lumínica, distancia, inclinación, desplazamiento, presión, fuerza, torsión, ph, etc. Una magnitud eléctrica sería las señales que se pueden detectar por otros dispositivos como: la capacitación eléctrica en un sensor de humedad, la resistencia eléctrica en una RTD (Resistance Temperature Detector), la tensión eléctrica en un termopar, la corriente eléctrica en un foto trasmisor, etc.

A la propiedad que tiene de convertir la forma de energía en otra diferente, se le han buscado diferentes aplicaciones, en la industria aeroespacial, Industria automotriz, Medicina, Robótica, etc.

Figura 17: Acelerómetro Piezo-Eléctrico
Fuente: https://prezi.com/pswdl_-08vat/sensores-de-vibracion/

1.2 Definición y características de sistemas de monitoreo

1.2.1 Central de monitoreo.

Las centrales de monitoreo son unidades que permiten el control y prevención de distintos eventos. Las centrales de monitoreo en los sistemas de alarma, el objetivo fue dar aviso lo más claro posible a los hechos imprevistos y no cotejados que estaban ocurriendo en una propiedad o equipo que se esté verificando. Los avisos que las centrales monitorean con el paso del tiempo se han extendido a los detectores de fuego y con la llegada de los sistemas microprocesados, a todo tipo de señales y sucesos, desde escapes de gas, control de ingreso de personal, entrada en funcionamiento de maquinarias automáticas, hasta el telecomando de equipos, luces o sistemas de riego.

Obviamente, que para que esto tuviera efecto alguien debía oír estas señales y actuar en consecuencia, es entonces donde aparecen las primeras centrales de monitoreo de eventos, las cuales por una cuestión de costos su implementación se basaba exclusivamente en la comunicación por radioenlace punto a punto. Estos enlaces tuvieron sus comienzos en los destacamentos de policía y en las entidades financieras y bancarias a sus principales clientes.

En la medida que se fueron perfeccionando los sistemas de comunicaciones, el monitoreo de alarmas se convirtió en un servicio más popular, y esto debido en gran medida a que por utilizarse la telefonía urbana para el enlace entre el objetivo y la central, el costo final descendió notablemente, haciéndose accesible a viviendas particulares y pequeños comercios.

El objetivo de una estación de monitoreo, como su nombre lo indica, es monitorear los eventos detectados por el sistema de alarma instalado y tomar una acción pertinente. Esta acción puede ser: avisar a la policía de un robo detectado, contactar a un técnico para la reparación del sistema o bien a un supervisor de guardia para corregir algún problema en la línea de montaje de una fábrica, es decir, el monitoreo aumenta el nivel de seguridad y productividad de una propiedad protegida haciendo de enlace entre el propietario y las autoridades pertinentes o la comunicación a distancia entre las máquinas y los usuarios “(ver fig.18)”.

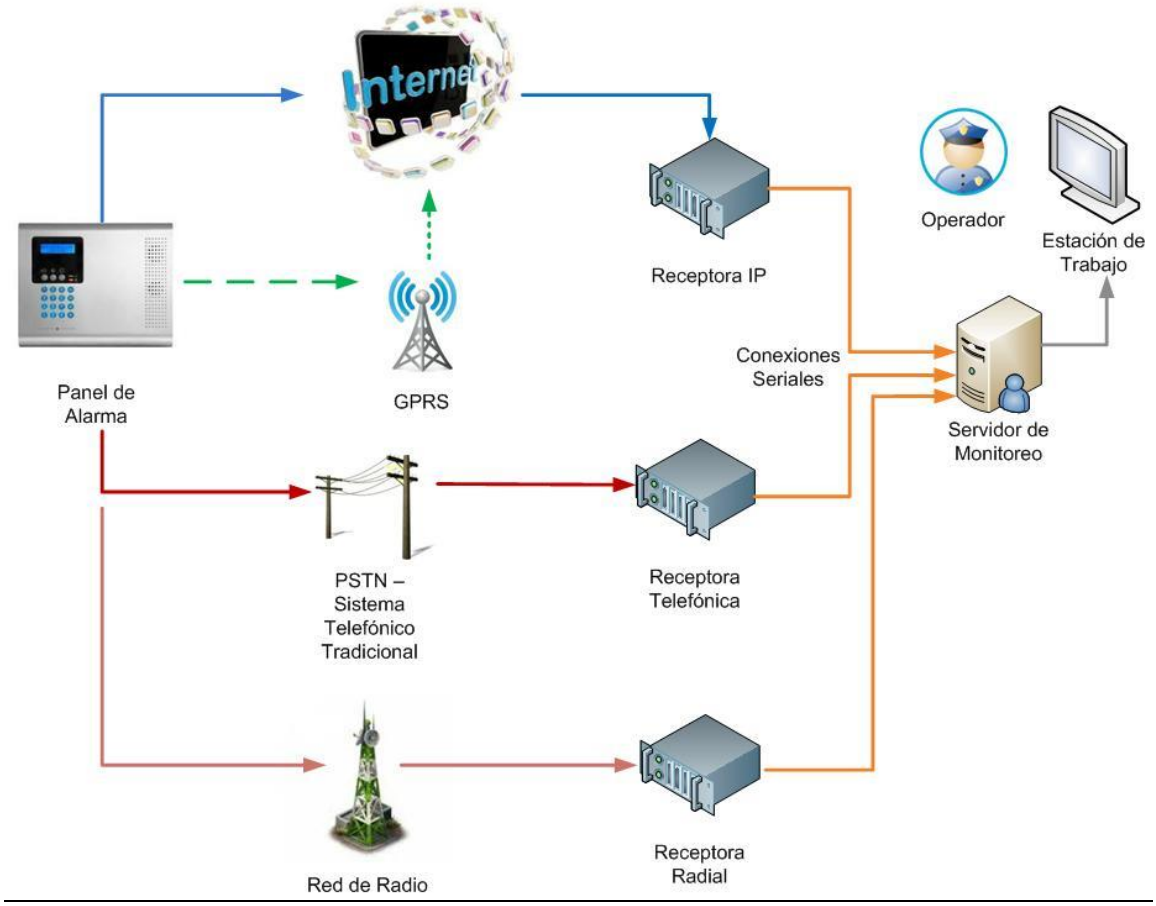


Figura 18: Estructura de una Central de Monitoreo.
Fuente: tecnoseguro.com/tutoriales/alarma

Mediante la utilización de un programa adecuado, la empresa de monitoreo registra en un archivo histórico los eventos recibidos del usuario y cuando aparecen en la pantalla del operador, éste procede según una lista de acciones previamente pactadas. Los datos del usuario son confidenciales y sólo se transmiten a las autoridades policiales u otras, los que sean relevantes para actuar en consecuencia. Dar cuenta de los pormenores del usuario no es una función de las centrales de monitoreo, sino asistir y brindar ayuda lo más eficazmente posible.

1.2.2 Características de una central de monitoreo.

Para considerar que se tiene una central de monitoreo es necesario cumplir una serie de características básicas que permiten el reconocimiento y estructuración básica de una central de monitoreo. Las características más importantes que deben ser implementadas para una central de monitoreo se resumen en las siguientes partes:

- Sistema de monitoreo robusto y con entorno amigable, fácil de usar y con herramientas adaptadas a todas las necesidades, además, que contemple todas las características del servicio de monitoreo.

- Servidores e internet, el primero para tener un banco de datos con gran capacidad de almacenamiento de información y el segundo para que los usuarios puedan comunicarse de forma rápida y eficiente con el servidor.

- Poseer la cantidad de personal necesaria para cubrir los avisos de alarma generada diariamente.

- Seguridad y confidencialidad de los datos.

- Múltiple plataformas de backup, así también, múltiple redundancia de los servidores instalados en internet.

- Respaldo de energía ilimitado.

1.3 Microcomputadores, estructura y características

1.3.1 Microcomputadores.

Los microcomputadores o microordenadores son computadores de una menor escala que los computadores de escritorios y portátiles, con un diseño específico que dependerá para qué función o funciones fueron creados, al hacer estos nos permite integrar los componentes necesarios en un solo circuito impreso.

Entre sus características podemos señalar:

- Espacio, al tener reducidas dimensiones que en lo general no exceden las 10” de largo y 6” de ancho, permite a estos microcomputadores instalarse en ubicaciones donde el factor del espacio sea importante, también permite la fácil portabilidad de los mismos.

- Bajo consumo eléctrico, generalmente los microcomputadores pueden ser alimentados por una fuente externa entre 1 watts a 5 watts.

Configuración básica de un micro computador:

- Hardware básico
- Software básico
- Dispositivos de entrada/salida

Hardware Básico

Tarjeta madre (Motherboard): La tarjeta madre es el componente más importante de un computador. Es el dispositivo que funciona como la plataforma o circuito principal de una computadora, integra y coordina todos los sus demás elementos. También es conocida como placa base, placa central, placa madre, tarjeta madre o Board.

La tarjeta madre es un tablero que contiene todos los conectores que se necesitan para conectar las demás tarjetas del computador. Una tarjeta madre alberga los conectores del procesador, memoria RAM, BIOS, puertas en serie, puertas en paralelo, expansión de la memoria, pantalla, teclado, disco duro, enchufes. Una vez que la tarjeta madre ha sido equipada con estos los elementos que se han mencionado, se le llama “Chipset” o conjunto de procesadores.

La tarjeta madre debe realizar básicamente las siguientes tareas:

- Conexión física
- Administración, control y distribución de energía eléctrica
- Comunicación de datos
- Temporización
- Sincronismo
- Control y monitoreo

Para que la placa base cumpla con su cometido, lleva instalado un software muy básico denominado BIOS.

Unidad Central de Procesamiento (CPU):Es la parte de la computadora donde se manipulan los símbolos, los números, letras y datos, además controla y dirige la operación de la computadora, ejecuta funciones de procesamiento, y se le conoce generalmente como procesador.

Procesa o manipula los datos almacenados en memoria; puede recuperar información desde memoria (esta información son datos o instrucciones de programas). También puede almacenar los resultados de estos procesos en memoria para su uso posterior.

El procesador o CPU, lleva a cabo una gran variedad de cálculos, comparaciones numéricas y transferencias de datos como respuesta a las peticiones de los programas que están siendo ejecutados en memoria. La CPU controla las operaciones básicas del ordenador enviando y recibiendo señales de control, direcciones de memoria y datos de un lugar a otro de la computadora a través de un grupo de canales llamados BUS.

La Unidad Central de Proceso está constituida internamente por la Unidad de Control (Control Unit UC), la Unidad Aritmética y lógica (Arithmetc-LogicUnit UAL) y la Unidad de Memoria comúnmente llamada memoria principal o almacenamiento primario y su función es la de guardar los datos en registros que leen y escriben de las celdas de memoria, llevan y traen datos entre celdas de memoria y registros especiales, decodifican y ejecutan las instrucciones de un programa.

Memorias

- Memoria principal o de proceso RAM: A esta memoria se le denomina, entre otros calificativos, memoria de acceso libre, más conocida como memoria principal RAM (por sus siglas en inglés: Random Access Memory) y cuya misión es la de identificar y clasificar en forma ordenada la información que recibe, procesa y emite. La memoria RAM también se suele denominar de escritura/lectura, donde el usuario puede escribir o leer indistintamente, así como almacenar los resultados intermedios.

Una característica importante de la memoria RAM que deberá tener muy en cuenta es que se mantiene activa, siempre y cuando este alimentada eléctricamente. Al faltarle la electricidad a la cual está conectada la computadora o bien al desconectar o apagar la computadora, el contenido de la memoria RAM se borra, se pierde y es por esta característica que también se le acostumbra denominar memoria volátil.

- Memoria ROM: Denominada así por sus siglas en inglés (Read Only Memory), que significa memoria de solo lectura, es la principal característica de este tipo de memoria. En consecuencia, los datos grabados en la misma solo se puede leer y no pueden ser modificados por el usuario. La memoria ROM contiene una serie de programas y datos, incluidos por el fabricante del equipo, en donde vienen grabados todas las órdenes para que, al encender el equipo, se chequeen todas las conexiones y sus partes buscando anomalías de las mismas. Al mismo tiempo busca y carga desde los dispositivos de almacenamiento (disquete o disco duro) el sistema operativo. La memoria ROM en realidad es un software funcional incluido dentro de la tarjeta madre.

- Memoria Caché: Se conoce con el nombre de Cache Primario (nivel 1 o L1), la cual, al estar incluida dentro del CPU, corre la velocidad del mismo. Como regla general se dice que mientras más grande sea este cache, más rápido será el CPU. La denominada memoria cache secundaria, Nivel 2 o L2, viene para ser colocada en la

tarjeta madre y al ser una memoria externa su velocidad es más lenta que la del Nivel 1.

- Memoria Expandida: La memoria expandida se crea en el disco duro. Al ser una memoria externa, el CPU realiza la comunicación a una velocidad menor a como lo hace con la memoria cache, pera a una velocidad superior a como la realiza con el resto de la memoria auxiliar del disco duro.

- Memoria Auxiliar: Podemos considerar como memorias auxiliares a cualquier dispositivo que almacene datos, información o programa fuera de la memoria principal de procesos (memoria RAM). Entre las más generalizadas se encuentran los discos flexibles o duros y las cintas magnéticas.

Interfaz y Puertos

Una interfaz es el medio físico que permite la conexión entre el CPU y un periférico. Para hacer efectiva dicha conexión deberá existir otro conector que permita e acople con la interfaz. Este último conector, que se encuentra asociado con el CPU, se le denomina puerto.

Las interfaz y los puertos se clasifican de acuerdo a la forma como se transmiten los datos a través de ellos. Las posibles transmisiones son dos:

- Transmisión Serial: Los bits de cada dato o símbolo se transmiten uno detrás de otro en un mismo cable. Es decir, los datos viajan (se transmiten) bit a bit y, en consecuencia, se necesitara de un solo cable.

- Transmisión Paralela: En la transmisión paralela todos los bit de un carácter o símbolos viajan al mismo tiempo, por lo tanto, la transmisión paralela necesitara de ocho cables para la transmisión de datos pues, como recordara, un carácter equivale a 8 bits y, en esta transmisión se envía un carácter a la vez lo que significa 8 bits simultáneamente.

Software Básico

Sistema Operativo: Un Sistema operativo (SO) es un software que actúa de interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar un computador. Es responsable de gestionar, coordinar las actividades y llevar a cabo el intercambio de los recursos y actúa como estación para las aplicaciones que se ejecutan en la máquina, siendo este la parte más importante de un ordenador.

Programa: En el área de la computación, se entiende como programa a una secuencia lógica de instrucciones, dadas en un determinado lenguaje, que indican a la computadora el proceso que debe realizar con los datos suministrados. Por lo tanto, por medio de los lenguajes se obtienen programas que permitan a la computadora realizar diferentes tareas o funciones.

- **Programas Funcionales:** Es el conjunto de programas cuyo principal objetivo es facilitar al máximo una explotación racional de un equipo de computación, como la memoria ROM.

- **Programas Específicos:** Los programas específicos son los que se aplican para resolver determinadas necesidades del usuario de la computadora. Estas necesidades pueden ser: científicas, técnicas, de gestión, disfrute de juegos, etc.

Dispositivos de Entrada/Salida

Dispositivos de Entrada: Solo son capaces de suministrar información al sistema ordenador.

Dispositivos de Salida: Dispositivos que están preparados para sacar al exterior los resultados obtenidos por la CPU.

Los dispositivos de entrada/salida comúnmente denominados periféricos. Un periférico es un dispositivo electrónico físico que se conecta o acopla a una computadora, pero no forma parte del núcleo básico (CPU, memoria, placa madre, alimentación eléctrica) de la misma.

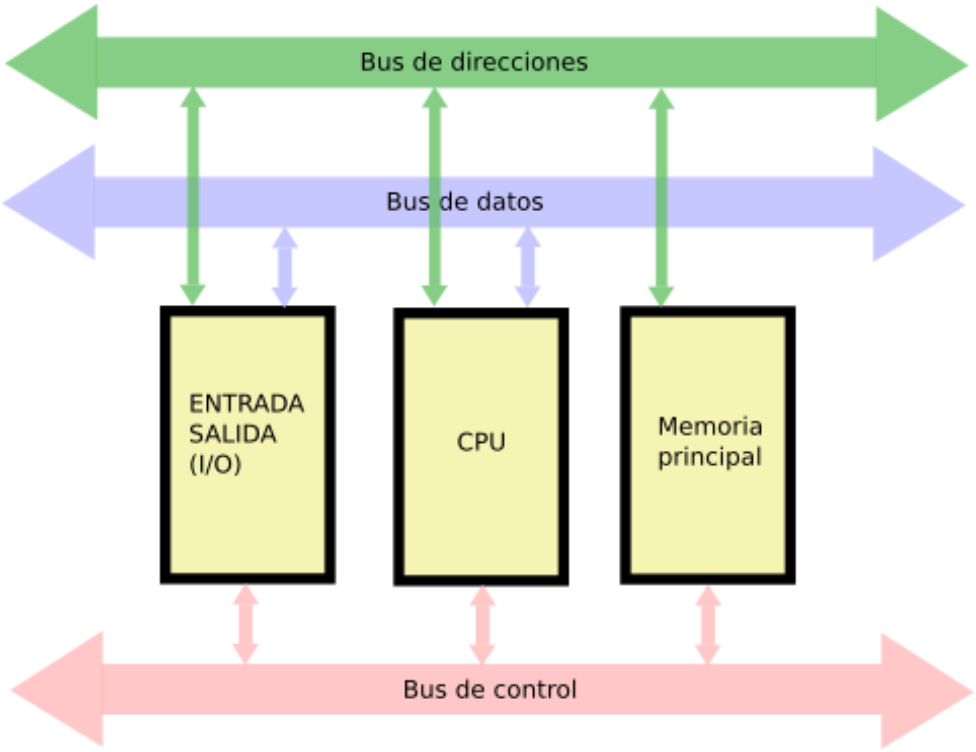


Figura 19: Microcomputador.
Fuente: Arquitectura de Computadoras: 2.-Organización de un microcomputador (estructura de Von Newman). (2008.).

1.4 Sistemas embebidos

Un sistema embebido consiste en un sistema de computación cuyo hardware y software están específicamente diseñados y optimizados para resolver un problema concreto eficientemente. El término "embebido" (también se le conoce como “empotrado”) hace referencia al hecho que la electrónica o el sistema electrónico de control es una parte integral del sistema en que se encuentra. La característica principal que diferencia a los “embebidos” de los demás

sistemas electrónicos es que, por estar insertados dentro del dispositivo que controlan, están sujetos en mayor medida a cumplir requisitos de tamaño, fiabilidad, consumo y coste, y su existencia puede no ser aparente. (Según: *"Estudio de Prospectiva: Tendencias y aplicaciones de los Sistemas Empotrados" del Observatorio de Prospectiva Tecnológica Industrial).* Algunos ejemplos de Sistemas Embebidos son los sistemas de información integrados en automóviles, trenes o aviones, y controladores de procesos en sistemas de producción industrial.

La importancia que están adquiriendo los Sistemas Embebidos es indiscutible. Estos Sistemas aportan valor añadido a los productos y, cada vez más, son los responsables de las mejoras introducidas en términos de innovación y competitividad.

Por lo general los sistemas embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador incorporado sobre el mismo o bien, utilizando algún compilador específico, suelen utilizarse lenguajes como C y C++.

¿Cómo están construidos los Sistemas Embebidos?

Los Sistemas Embebidos suelen tener en una de sus partes una computadora con características especiales conocida como microcontrolador que viene a ser el cerebro del sistema. Este no es más que un microprocesador que incluye interfaz de entrada/salida en el

mismo chip. Normalmente estos sistemas poseen una interfaz externa para efectuar un monitoreo del estado y hacer un diagnóstico del sistema.

Figura 20: Ejemplo de un Board Embebido.
Fuente: http://beagleboard.org/

Por lo general, los Sistemas Embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo, o también, utilizando los compiladores específicos que utilizan lenguajes como C o C++ y en algunos casos, cuando el tiempo de respuesta de la aplicación no es un factor crítico, también pueden usarse lenguajes interpretados como Java.

Hardware

Normalmente un sistema embebido se trata de un módulo electrónico alojado dentro de un sistema de mayor entidad (‘host’ o anfitrión) al que ayuda en la realización tareas tales como el procesamiento de información generada por sensores, el control de determinados actuadores, etc. El núcleo de dicho módulo lo forma al menos una CPU en cualquiera de los formatos conocidos:

- Microprocesador.

- Microcontrolador de 4, 8, 16 o 32 bits.

- DSP de punto fijo o punto flotante.

- Diseño a medida ‘custom’ tales como los dispositivos FPGA

El módulo o tarjeta, además puede haber sido desarrollado para satisfacer una serie de requisitos específicos de la aplicación a la que está dirigido. Entre éstos, podemos citar:

- Tamaño: por lo general deberá ser reducido, aunque también es posible que se desee que adopte un formato estándar: PC-104, Eurocard, etc.

- Margen de temperatura especifico del ámbito de aplicación:

- Gran consumo (0ºC hasta 70ºC) o Industrial y automoción. Márgenes de temperatura hasta 125ºC
- Aerospacial
- Militar
- Electromedicina

- Consumo de energía: En aplicaciones en las que es necesario el empleo de baterías, se buscará minimizar éste.

- Robustez mecánica: Existen aplicaciones donde los dispositivos sufren un alto nivel de vibraciones, golpes bruscos, etc. En el diseño se deberá tener en cuenta dicha posibilidad.

- Coste: No es lo mismo diseñar un producto a medida con pocas unidades que diseñar un producto para el competitivo mercado del gran consumo. La calibración de los costes es esencial y es tarea de los ingenieros de diseño.

Software

En lo que se refiere al software, se tendrán requisitos específicos según la aplicación. En general para el diseño de un SE no se dispone de recursos ilimitados sino que la cantidad de memoria será escasa, la capacidad de cálculo y dispositivos externos será limitada, etc.

Podemos hablar de las siguientes necesidades:

-Trabajo en tiempo real.

- Optimizar al máximo los recursos disponibles.

- Disponer de un sistema de desarrollo específico para cada familia de microprocesadores empleados.

- Programación en ensamblador, aunque en los últimos años, los fabricantes o empresas externas han mejorado la oferta de compiladores que nos permiten trabajar en lenguajes de alto nivel, tales como C. - etc.

El empleo de un sistema operativo determinado o el no empleo de éste dependerá del sistema a desarrollar y es una de las principales decisiones que habrá que tomar en la fase de diseño del SE. Así, en el caso de decidirse por el empleo de microcontroladores y DSP, por lo general no se usará sistema operativo mientras que si se emplea algún micro del tipo ARM, PowerPC, Intel X86, etc. sí que lo llevará. La decisión dependerá de los requisitos del sistema, tanto técnicos como económicos.

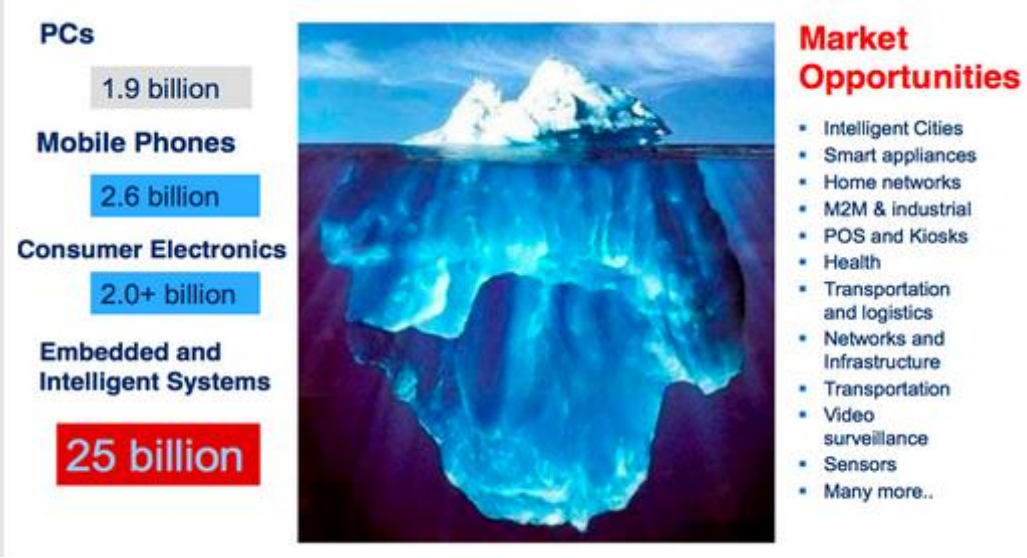


Figura 21: Visión de los sistemas Embebidos al 2020
Fuente: semanticwebbuilder.org.mx

Ventajas

- Posibilidad de utilización de sistemas operativos potentes que ya realizan numerosas tareas: comunicaciones por redes de datos, soporte gráfico, concurrencia con lanzamiento de threads, etc. Estos sistemas operativos pueden ser los mismos que para PC compatibles (Linux, Windows, MS-DOS) con fuertes exigencias en hardware o bien ser una versión reducida de los mismos con características orientadas a los PC embebidos.

- Al utilizar los Sistemas Embebidos, se pueden encontrar fácilmente herramientas de desarrollo de software potentes, así como numerosos programadores que las dominan, dada la extensión mundial de las aplicaciones para computadoras compatibles.
- Reducción en el precio de los componentes hardware y software debido a la gran cantidad de computadoras en todo el mundo.

Tecnología Java

La tecnología **Java es cada vez más utilizada en sistemas embebidos avanzados,** debido a sus capacidades inherentes de soporte de red, optimización de dispositivos y procesador de datos. La mayoría de las características de la plataforma Java SE pueden ser ahora empleadas para el desarrollo embebido, gracias a la capacidad cada vez mayor del nuevo hardware disponible en el mercado.

Seguridad

Al utilizar Sistemas Embebidos en productos complejos se debe pensar en la seguridad de la información contenida en el dispositivo, pues esa información será transmitida por redes privadas e Internet. El diseño de un producto que incorpora sistemas embebidos generalmente está orientado a minimizar los costos y maximizar la confiabilidad, por lo que se deben incluir funciones criptográficas, diseño de protocolos y consultoría en análisis y verificación así como servicios de pruebas de seguridad y evaluaciones específicas.

Características de los Sistemas Embebidos

Las principales características de un Sistema Embebido son el bajo costo y consumo de potencia. Dado que muchos sistemas embebidos son concebidos para ser producidos en miles o millones de unidades, el costo por unidad es un aspecto importante a tener en cuenta en la etapa de diseño.

Un **Sistema Embebido** está conformado por un microprocesador y un software que se ejecuta sobre él mismo. Sin embargo, este software necesita un lugar donde pueda guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o

ROM, la cual cierta cantidad es utilizada por el Sistema Embebido.

1.5 BeagleBone Black, teoría y aplicaciones

El BeagleBone Black Lanzado el 23 de abril de 2013 a un precio de 45 dólares. Entre otras diferencias, aumenta la memoria RAM de 512 MB, el reloj del procesador a 1 GHz, y añade HDMI y 2 GB de memoria flash eMMC. El BEAGLEBONE Negro también viene con Linux kernel 3.8, actualizado de kernel de Linux de la BEAGLEBONE original de 3.2, lo que permite la BEAGLEBONE Negro aprovechar Directo Rendering Manager (DRM).

BEAGLEBONE Negro Revisión C (lanzado en 2014) aumentó el tamaño de la memoria flash 4 GB. Esto le permite a la nave con Debian GNU / Linux instalado

El microcomputador BeagleBone Black es parte elemental de este proyecto, desarrollado por Texas Instruments y lanzado el 23 de Abril de 2013 (Nuxat) es la segunda generación tras el BeagleBone, surgió como respuesta de Texas Instruments al éxito del microcomputador similar Raspberry pi. BeagleBone cuenta con una comunidad de

desarrolladores que ha permitido descubrir diferentes aplicaciones de este microcomputador haciéndolo cada día más reconocido y utilizado en proyectos de ingeniería.

El BeagleBone Black, es la última placa desarrollada por BeagleBoard.org y al igual que sus predecesoras, está diseñado para hacer frente a la Comunidad Open Source, los primeros adquirientes, y cualquier persona interesada en un procesador basado en ARM de bajo costo Cortex-A8. Se ha equipado con un conjunto mínimo de características para que el usuario pueda experimentar el poder del procesador y no pretende ser una plataforma de desarrollo completa ya que muchas de las características y las interfaces proporcionadas por el procesador no son accesibles desde el BeagleBone Back a través del soporte en el board de algunas interfaces. No es un producto completo diseñado para hacer cualquier función particular. Es una base para la experimentación y el aprendizaje de cómo programar el procesador y el acceso a los periféricos mediante la creación de su propio software y hardware.

También ofrece acceso a muchas de las interfaces y permite el uso de Add-On llamadas capas, para añadir muchas combinaciones diferentes de características. Un usuario también puede desarrollar su propia placa o añadir sus propios circuito.

BeagleBone Black	
Processor	AM3358 ARM Cortex-A8
Maximum Processor	1GHz
Analog Pins	7
Digital Pins	65 (3.3V)
Memory	512MB DDR3 (800MHz x 16), 2GB (4GB on Rev C) on-board storage
USB	HS USB 2.0 Client Port, LS/FS/HS USB 2.0 Host Port
Video	microHDMI, cape add-ons
Audio	microHDMI, cape add-ons
Supported Interfaces	4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2xCAN Bus, 4 Timers
Price	$49

Tabla 1: Características de BeagleBone Black
Fuente: beagleboard.org/boards

- Procesador: AM335x 1GHz ARM® Cortex-A8
- Memoria RAM: 512MB DDR3
- Tarjeta Gráfica: SGX530 aceleradora 3D
- Vídeo: 1280 x 1024
- Salida de Vídeo/Audio: Micro HDMI
- Tarjeta de red: Ethernet 10/100Mbps conexión RJ-45 Jack
- Puerto host USB 2.0 tipo A
- Puerto cliente mini-USB 2.0
- 2 x 46 Entrada/Salida de tipo: 65 digitales, 7 analógicas, 4 serie, 2 SPI, 2 I2C, 8 PWM, 4 timers
- Memoria interna de 2GB eMMC y ampliable por el puerto USB o por tarjeta micro SD

Sin duda la principal ventaja del de BeagleBone Black, es la compatibilidad de varios sistemas operativos: Android, Debian, Cloud9 IDE y por supuesto Totalmente compatible con cualquier distribución de Linux.

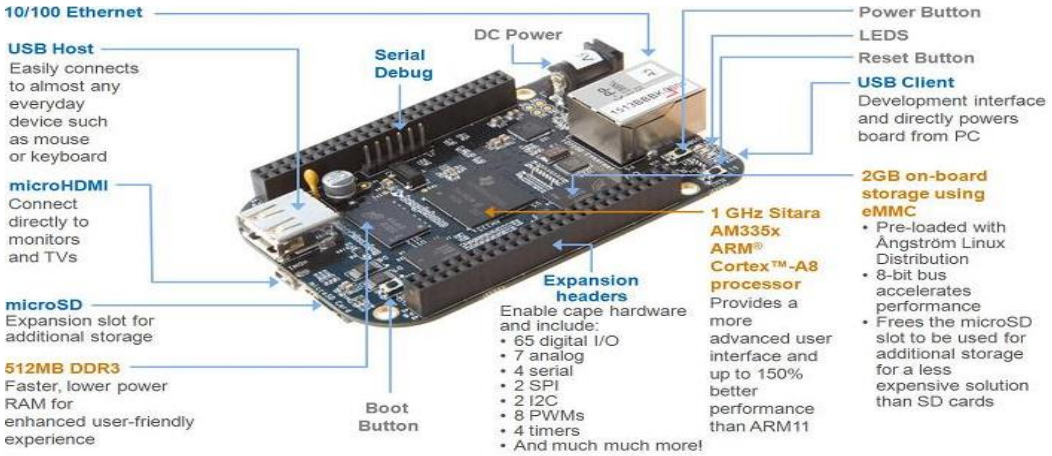


Figura 22: Tarjeta BeagleBone Black.
Fuente: Texas Instruments

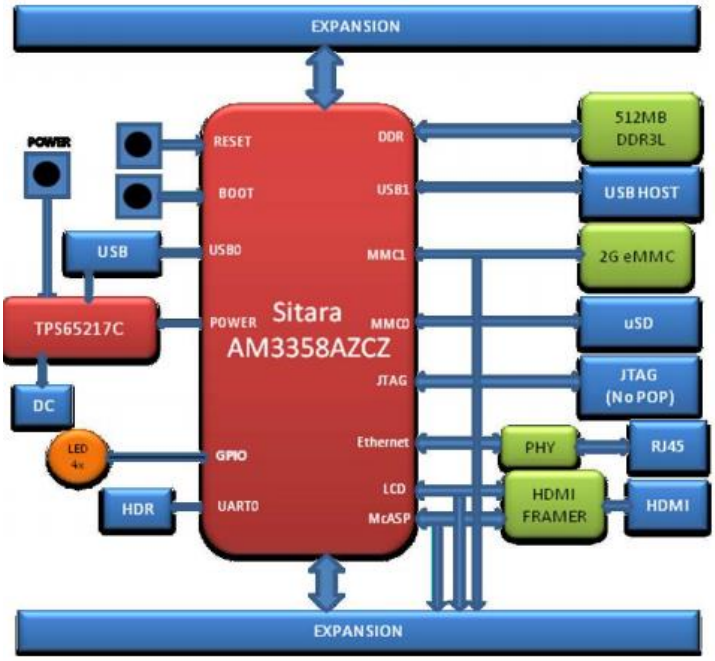


Figura 23: Diagrama en bloque del BeagleBone Black.
Fuente: Coley, G. (2013). BeagleBone Black System Reference Manual.

1.6 Interfaz gráfica, características

La interfaz gráfica de usuario, conocida también como GUI. Es un programa informático que actúa de tal forma que el usuario se puede comunicar con la máquina, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

Utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina.

Las interfaces gráficas surgen de la necesidad de hacer los ordenadores más accesibles para el uso de los usuarios comunes.

Pequeños dispositivos móviles como los teléfonos inteligentes suelen utilizar los elementos WIMP con diferentes metáforas de unificación, debido a las restricciones en el espacio y los dispositivos de entrada disponibles.

Algunos sistemas operativos de pantalla táctil utilizan la clase de interfaces gráficas de usuario llamado post-WIMP. Estos estilos de interacción con el apoyo de más de un dedo en contacto con una pantalla, que permite acciones como presionar y girar, y que no están respaldadas por un puntero y un ratón.

El estilo más común de una interfaz gráfica entre hombre-computadora es la de WYSIWYG, la cual manipula directamente una interfaz de usuario basada en iconos. En una interfaz WYSIWYG la representación gráfica con la cual los usuarios interactúan en la pantalla es esencialmente la misma imagen creada por la aplicación.

Muchas de las aplicaciones y editores graficas tienen algún componente WYSIWYG; principalmente aquellas aplicaciones gráficas. Lo que nosotros vemos en la pantalla es el resultado de nuestra aplicación en papel.

En los sistemas informáticos, la relación humano-computadora se realiza por medio de la interfaz, que se podría definir como mediador. Cuando existen dos sistemas cualesquiera que se deben comunicar entre ellos la interfaz será el mecanismo, el entorno o la herramienta que hará posible dicha comunicación.

Podríamos definir básicamente dos tipos de interfaces:

- La interfaz física: un ratón y un teclado que sirven para introducir y manipular datos en nuestro ordenador.

- La interfaz virtual o interfaz gráfica (GUI) que permite, mediante iconos (cursor + objetos gráficos metafóricos), interactuar con los elementos gráficos convirtiendo al ser humano en usuario de la aplicación.

Estas dos mediaciones son relaciones del tipo entrada de datos (*input*). Al igual que tenemos una entrada, necesitamos algo que facilite la salida de datos (*output*), para esto tenemos, por ejemplo, la pantalla de la computadora, donde se visualizan estas interfaz gráficas, o la impresora, donde se imprimen los datos.

Para diseñar una interfaz es necesario pasar por cuatro etapas:

1. Análisis de requerimientos del producto, análisis de las tareas. Conocimiento del usuario. Generación de posibles metáforas y análisis del tipo de diálogo. Revisión.
2. Generación de prototipos virtuales (*layouts*) o físicos para investigar desde lo general hasta el detalle. Desarrollo de la aplicación, del sitio o del sistema.
3. Planificación (desarrollo del plan, definición de las medidas, selección de participantes, formación de observadores, preparación de los materiales). Test (prueba piloto, test con usuarios).
4. Conclusión (análisis de los datos, elaboración del informe, resultados y recomendaciones). Comparación con estándares (internos y/o externos), versiones anteriores del mismo producto y productos competidores. Verificación de las diferencias. Generación de nuevas metas.

CAPÍTULO 2. DISEÑO DEL SISTEMA DE ALARMA Y MONITOREO DEL PROYECTO

2.1 Estructura de los paneles blindados, criterios para selección del área donde se implementaran el sistema

Un panel blindado debe reunir una estructura, ciertos criterios y estandarización al momento de ser construido, instalado, y/o rehabilitado en fincas y edificaciones verticales, donde se requieran más de (4) equipos de medida, o en el que se haga necesario centralizar diferentes puntos de medida. Formalizando los niveles de seguridad con los que se debe construir, rehabilitar e instalar alimentados desde baja tensión BT a niveles de tensión menores o iguales a 600V, que garanticen los niveles de seguridad física adecuados a clientes y técnicos de la empresa.

2.1.1 Ubicación (localización) de los paneles.

Para la ubicación e instalación de los referidos paneles de medidores se recomienda ubicarlo de la siguiente manera:

A. Protegido, de modo que garantice el acceso a técnicos y lectores de la empresa Distribuidora de Electricidad, donde no se vea mayormente afectado por las inclemencias del tiempo u otras condiciones que puedan ocasionar deterioro de cualquier índole en el tiempo.

B. En lugares donde esté a salvo de ser golpeado por vehículos.

C. En lugares donde no obstruya el tránsito peatonal.

D. No debe ser empotrado.

E. En un punto de fácil acceso que los técnicos de la empresa pueda realizar su labor sin dificultad, asegurando que las puertas queden con el suficiente espacio para que un operario pueda retroceder en caso de emergencia.

2.1.2 Anclaje del panel.

El panel deberá estar anclado en su base y a una pared en su parte posterior, mediante tarugos tipo de anclaje de expansión de ¼” x 3” (pulgadas) o clavos a presión de similar longitud.

No se permitirán armarios cuya base este rasante con el piso.

Los armarios deberán ser instalados sobre una base de 9’’ (pulgadas) de altura construida en block de concreto.

2.1.3 Obra civil.

En los casos donde se requiera obra civil, ésta deberá realizarse según las siguientes características:

Los paneles reubicados en espacios abiertos, donde no existe una pared que los pueda soportar en su parte posterior, requieren de una caseta construida a partir de una base de block de por lo menos 9’’ (pulgadas) de altura, sobre cual se levantarán los muros en blocks de las dimensiones requeridas por el panel a instalar, a su vez el muro deberá poseer columnas con varillas de 3/8’’ fundidas sobre los mismos orificios de los blocks, por lo menos cada 32’’. Las columnas que fijan el muro también deben quedar amarradas a la torta de modo que se garantice la firmeza de la estructura; de igual forma la caseta deberá poseer un techo fundido para proteger al panel de la intemperie (“Ver Fig.24”).

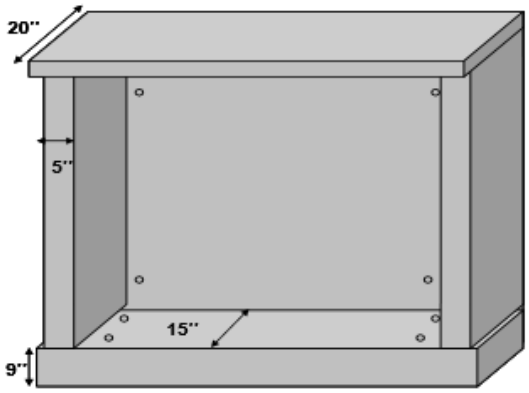


Figura 24: Base de Concreto del panel
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

2.1.4 Especificaciones técnicas de instalación para los paneles tipo **A** (socket) y tipo **B** (bottom).

Señalizaciones:

Todos los paneles deberán tener en su parte externa señales gráficas y textuales donde se especifique:

A. Peligro de riesgo eléctrico

B. No maniobrar

C. No acceder sin permiso de la empresa distribuidora

D. Requerimiento de uso de equipo de protección personal

E. Señal ética según disposición de la gerencia de Seguridad Industrial

Figura 25: Señalización de los paneles.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

Puesta a Tierra de Paneles

- Para garantizar la seguridad todos los paneles deberán poseer en sus carcasas un conector de 100 amperios con espacio para 2 conductores No 1/0, en material bimetálico, para conectar la línea de tierra, el cual deberá estar localizado en la estructura del módulo del breaker principal.

- El cable que aterriza un panel (conexión varilla panel) debe ser de cobre, desnudo No 2 AWG.

- La varilla de tierra debe ser de cobre, de diámetro 5/8” x 6 pies de longitud.

- Se utilizara conector tipo cuña o conector de ajuste con tornillo confeccionado en cobre para unir la varilla de tierra al conductor de tierra.

- La tierra también se deberá conectar obligatoriamente al neutro del panel en cable No 6 AWG como mínimo.

2.1.5 Especificación para la construcción de paneles.

Especificaciones Técnicas para Construcción de Paneles Blindados Tipo A y B:

Los paneles tipo A y B se construyen para albergar grupos de medidores tipo socket, bottom y sus protecciones. Esencialmente son armarios construidos en lámina tratada o galvanizada de calibre No. 14 que constan de tres módulos independientes, y separados físicamente por las láminas internas del panel como son el módulo del breaker principal, el módulo de medidores y el módulo de breaker de carga.

Al interior conllevan un breaker de protección general, un barraje principal, los zócalos o base que albergan los medidores, los breaker de protección y el cableado interno. Los paneles Tipo Bottom conllevan además una puerta interna con sistema de cierre de seguridad y mica de policarbonato ultravioleta (UV).

Cada módulo posee un sistema de cierre de seguridad con puertas que van enmarcadas dentro de un marco metálico que hace parte constitutiva del mismo armario.

2.1.6 Especificaciones técnicas para construcción de paneles blindados tipo **C**.

Los paneles tipo C son estructuras sencillas construidas en láminas de acero que albergan de 3 a 6 medidores tipo zócalo (240 ó 120 V), dispuestos en sentido vertical o distribuidos en forma rectangular y las protecciones para las acometidas de carga; a diferencia de los paneles para edificios descritos en la primera parte de este manual, estos no contienen barras alimentadoras ni breaker principal. Esencialmente son armarios construidos en lámina tratada o galvanizada de calibre No. 14 que constan de una sola puerta asegurada con perno que posee visores UV y una malla de protección en la parte posterior. Al interior, este tipo de paneles contiene los zócalos que albergan los medidores, los breaker de protección de acometidas, barra de neutros y el cableado interno.

Los paneles blindados tipo C se alimentan mediante acometida que proviene directamente del transformador o de la red Baja Tensión e ingresan por la parte inferior o lateral inferior del panel, lugar por donde también salen las acometidas de carga.

Figura 26: Presentación exterior del panel.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

Figura 27: Presentación Paneles Tipo A, B y C.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

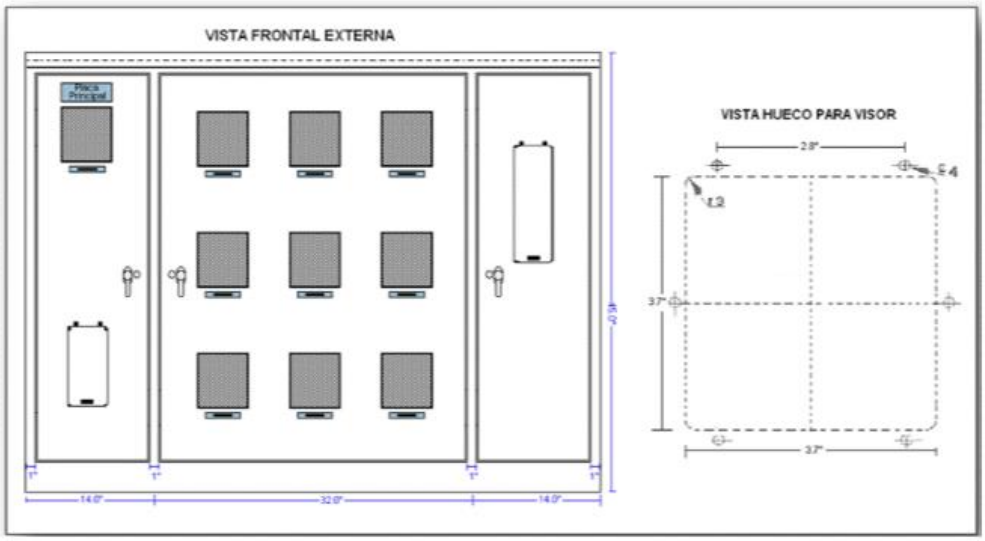


Figura 28: Esquema de Construcción Panel Tipo A.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

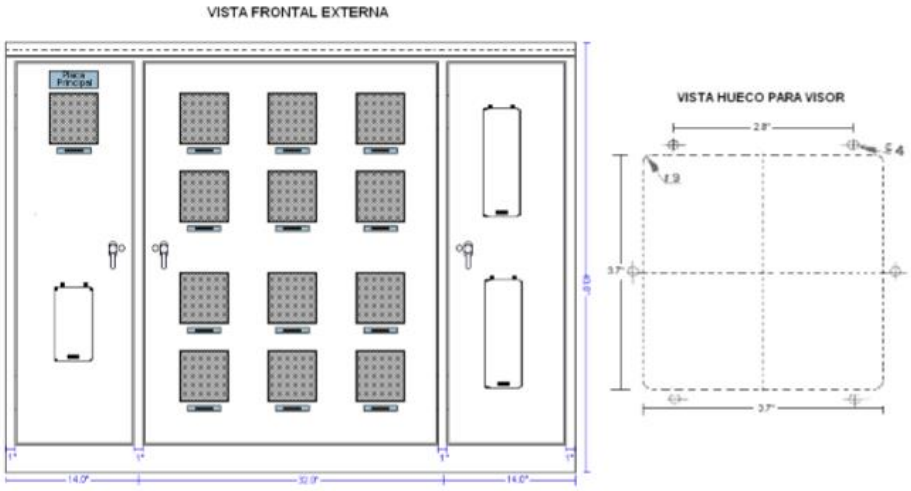


Figura 29: Esquema de Construcción Panel Tipo B.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

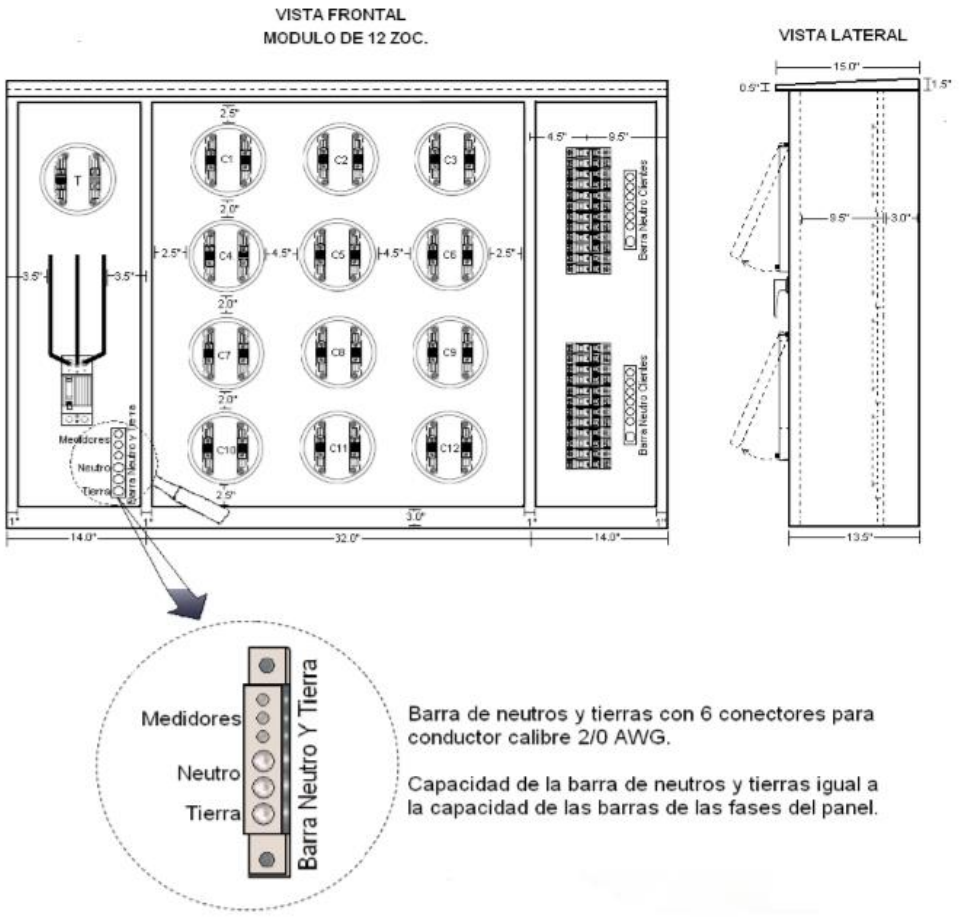


Figura 30: Esquema de Construcción Panel Tipo A.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

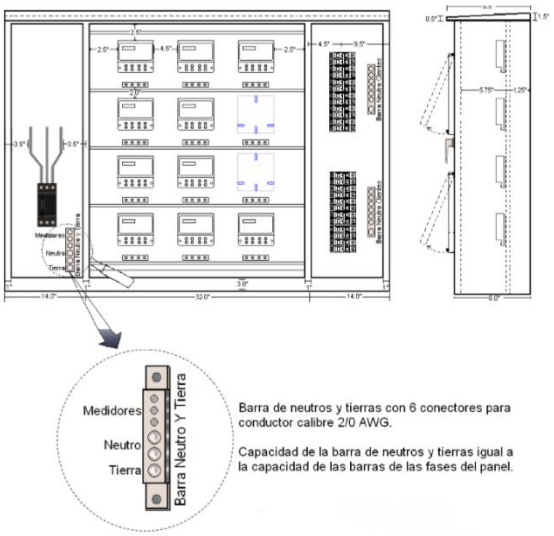


Figura 31: Esquema de Construcción Panel Tipo B.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

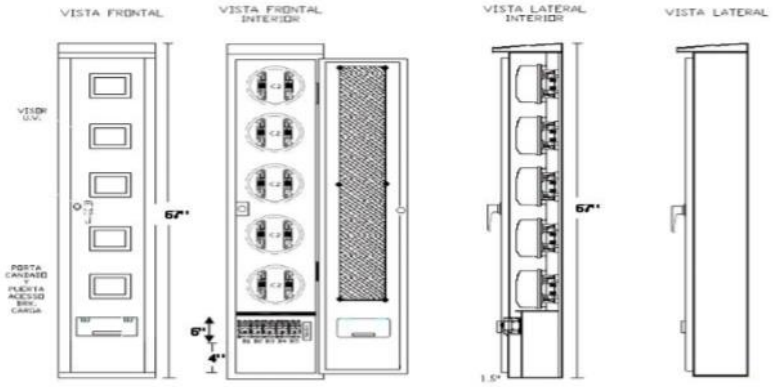


Figura 32: Esquema de Construcción Panel Tipo C.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

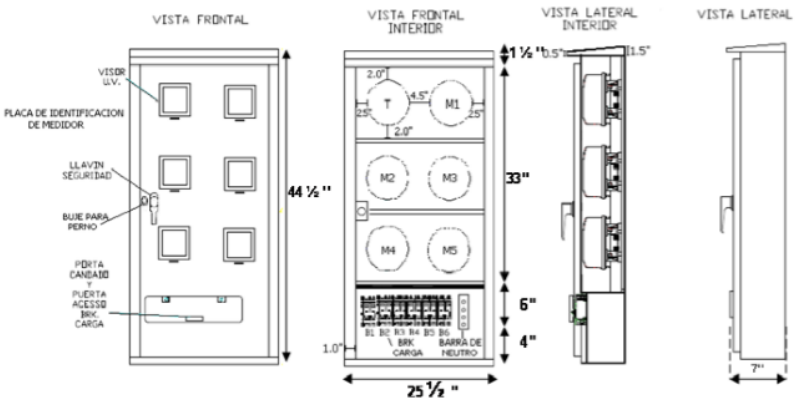


Figura 33: Esquema de Construcción Panel Tipo C.
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

2.1.7 Llavines para los paneles.

Cada panel cuenta con tres o más llavines para cada uno de sus tres módulos. Las llaves son especiales para cada llavín y su función es insertar una barra de acero (Perno tipo torpedo) en cada uno de los llavines instalados para estos fines. Los detalles sobre las medidas del perno, el casquillo porta perno y el cierre, se muestran en la figura No. 34. La manija de cada puerta debe ser asegurada desde el interior sin tornillos a la vista, cada llavín debe tener unas barras y placa retenedora que dificulta el doblado del cierre.

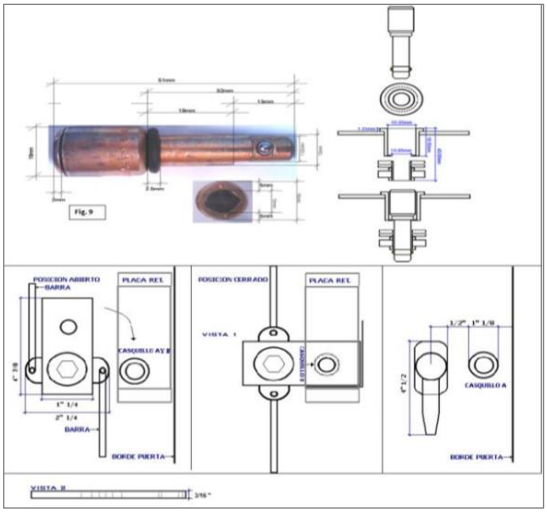


Figura 34: Llavines de los Paneles
Fuente: Manual de construcción e instalación de paneles blindados para medidores de energía

2.1.8 Pintura de panel.

La pintura debe ser de color Beige RAL 7032 o Gris ANSI 61. En pintura electrostática o pintura al horno, con una capa mínima de recubrimiento de 100µm en la parte exterior y 70µm en la parte interior. La misma debe garantizar una adherencia mínima de todas y de cada una de las capas de 400 libras/ pulg².

No se deben ejecutar soldaduras después de pintado el panel para evitar deterioro de la pintura y corrosión de la misma.

2.1.9 Identificación del panel.

Los paneles deben tener tres tipos de identificación:

A. Placa de Identificación General:

- Es una placa de 5'' X 2,5'' pulgadas (Fig. 21) que se instala en la parte superior izquierda del panel sobre la puerta del compartimiento del Breaker principal, la placa de identificación general se debe instalar remachada en sus cuatro esquinas, el material será acero inoxidable o aluminio tintado de azul con letras grabadas que contengan la siguiente información:

- Logo de EDE Este o Número asignado al panel o Numero de finca (NIF) o Dirección o Sector o Tensión de alimentación del panel o Teléfono de emergencia ante averías o fraude en el panel.

B. Placas para Identificación de Medidores:

- Cada servicio debe tener una identificación, en lámina de aluminio o acero inoxidable de 2’’ X ½’’ grabada sobre la parte superior de cada visor que se localiza en la puerta de los medidores y remachada en dos (2) de sus extremos.

C. Placas para Identificación de los Breaker de Carga:

- Cada servicio debe de tener una identificación del apartamento o local en lámina de aluminio o acero inoxidable de 1½’’ X ½’’ grabada al lado de los breaker de carga y remachada en dos (2) de sus extremos.

D. Datos de placa característica del fabricante:

- Los paneles deberán tener su placa interna con la información general del fabricante y del panel, la misma debe colocarse en la parte superior izquierda del módulo del interruptor principal. La información mínima necesaria que debe contener la placa característica es la siguiente: nombre del fabricante, número de serie y fecha de fabricación.(Edeeste, 2013)

2.1.10 Instalación del sistema.

El sistema de alarma basado en BeagleBone Black puede ser instalado en cualquier panel ubicado en el territorio nacional, ya que para su instalación no requiere modificaciones en la estructura del panel blindado. Solo es necesario colocar los sensores en las puertas y en la estructura y la caja que contendrá el sistema con el BeagleBone Black.

Este sistema está dirigido a cualquier panel blindado de cualquiera de las distribuidoras, por la fácil instalación y los requerimientos mínimos para su instalación como son: la energía eléctrica y el internet.

2.1.11 Sistema de respaldo de energía.

El sistema de alarma de los penales blindados está suministrado de energía por una línea directa desde los puntos antes de las medidores de los medidores dentro del panel, pero no siempre se puede contar que la energía estará las 24 horas, por tal razón y previniendo cualquier tipo de falla eléctrica el sistema estará suministrado de energía alternativa por un UPS de la marca APC modelo BR1500G, el cual suministrara la energía por 16 horas al sistema en caso de que allá cualquier fallo en el suministro principal.

Características de UPS:

- APC Back-UPS Pro

- 865 Watts.
- 1500 VA.
- Entrada 120V /Salida 120V.
- Interfaz Puerto USB.
- Modelo de tanda extendida.

2.2 Selección de los componentes que estructuraran el sistema

El sistema de seguridad de los paneles blindados de las distribuidoras de energía eléctrica estará monitoreado por los sensores mencionados en el capítulo1. El cual estará colocados en las puertas de los paneles y laterales para proteger el acceso a los mismos. A continuación mostraremos los distintos componentes que seleccionamos para la protección del panel.

2.2.1 Interruptores magnéticos.

La principal área a cubrir son los accesos al panel, estos estarán monitoreados con interruptores magnéticos mencionados en el capítulo 1, también conocidos como sensores magnéticos, estos constan de dos contactos de un interruptor magnético que están en un tubo de vidrio. Cuando un imán se aproxima al interruptor, las lengüetas magnéticas se atraen y cierran así los contactos del interruptor.

De este tipo de sensores existen varios tipos dependiendo el lugar específico en que serán colocados, por ejemplo las puertas metálicas en espiral usan el interruptor de la Figura 35 y los de puertas y ventanas usan los de la Figura 36.

Figura 36: Sensor Magnético de Puertas comerciales enrollables.

Figura 35: Sensor Magnético de Puertas comerciales enrollables.

Fuente: seamtrack.es

El sensor mostrado en la Figura 37 es parecido al sensor utilizado en el sistema de alarma este será colocado en las puertas de los paneles, los mismos estarán conectados al sistema de interfaz que luego enviara la señal al BeagleBone Black, la función básica del mismo es determinar si hay una violación al panel cuando el sistema de alarma está cerrado, indicar si la puerta del panel blindado ha sido abierta.

Estos paneles por las ubicaciones donde son colocados no cuentan con visibilidad de los contratistas o empleados de las distribuidoras esto hace que los mismos tengan poca seguridad, los propietarios de las distribuidoras no tiene forma de conocer si los mismo han sido violados a menos que se vaya a realizar algún tipo de trabajo técnico en dicho panel blindado.

2.2.2 Sensores de vibración.

El sistema de alarma cuenta con sensores de vibración mencionados en el capítulo1, los cuales son colocados en las estructuras metálicas de los paneles o áreas sensibles a vibraciones donde se quiera activar las alarmas en caso de que intenten remover estas estructuras metálicas.

El sensor a utilizar es un sensor de vibración de la marca Honeywell modelo 11WH, el cual permite ajustar el grado de sensibilidad de la vibración en la parte con la que hace el contacto "(ver fig.37)". Este cuando el sistema de alarma este cerrado y monitoree una vibración con el objeto el cual el sensor este haciendo contacto, internamente chocan unas placas en el interior del sensor mandando una serie de pulsos inmediatamente el contacto comience a vibrar. La misma llega al sistema de interfaz con el BeagleBone Black.

Figura 37: Sensor de Vibración.
Fuente: Honeywell.com

Las especificaciones técnicas del mismo son:

Rango de medición	0 ... 25,4 mm/s
Rango de frecuencia	10 Hz ... 1 kHz
Salida	4 ... 20 mA
Resolución	0,13 mm/s rms
Reproducibilidad	±1 %
Rango de temperatura	-40 ... +85 ºC
Tensión de alimentación	12 ... 30 VDC
Tiempo de respuesta (dentro del 2 % del valor)	<15 s

2.2.2 Cerradura eléctrica.

Las cerraduras eléctricas poseen un sistema de traba de puertas y aberturas con un funcionamiento electromecánico, que fortalece la estructura de modo único. A diferencia de las cerraduras convencionales, vulnerables en muchos aspectos, este tipo de cierre identifica a los usuarios habilitados y audita las acciones, creando registros de ingresos y egresos, y programando permisos o restricciones horarias para determinadas personas.

Figura 38: Cerradura Eléctrica.
Fuente: locktrackrd.com

El cierre de la abertura se realiza con un bloqueo mecánico, muy seguro y resistente a agravios. El más completo funciona mediante electro-imanes, haciendo un cierre total de la puerta y activándose según la programación determinada. Como el sistema depende del suministro eléctrico, es conveniente contar con un generador o con abastecimiento de emergencias alternativo, generalmente incluido en el paquete.

Hay alternativas de cerraduras eléctricas más económicas y versátiles, las cuales funcionan con chapas y contrachapas electrónicas, o también cilindros eléctricos de cerraduras. Éstas se alimentan de la electricidad y de baterías para casos de cortes de energía.

FUNCIONES Y BENEFICIOS DE LAS CERRADURAS ELÉCTRICAS

La cerradura eléctrica puede contar con paneles de registro de huellas digitales o con botoneras de teclas en las que se ha de introducir el código de usuario. Otros más avanzados tienen lectores de chips, barras magnéticas o códigos de barras (cercanos o a distancia, y hasta con control remoto) que la persona habilitada llevará consigo, en su tarjeta de identificación u otro elemento.

Estas lecturas permiten la programación del sistema, de acuerdo a horarios y personal específico, para definir el ingreso a ciertas áreas. También, existe la posibilidad de modificar la definición de seguridad cada cierto tiempo, evitando así inconvenientes por el extravío o el olvido de los sistemas, o también su decodificación por parte de invasores.

Las cerraduras eléctricas son bastante más costosas que las convencionales, pero el gasto se ve amortizado en poco tiempo gracias a la seguridad.

2.3 Integración de los componentes del sistema con la herramienta BeagleBone Black

La integración de los componentes del sistema de alarma mencionados en el capítulo anterior son componentes de entrada y salida, el BeagleBone Black cuenta con una serie de pines entrada y salida los cuales nos sirven para interconectar estos componentes, pero por la características de algunos de estos pines en niveles de consumo, potencia, corriente o voltajes y por la característica de sockets que tiene el BeagleBone Black para poner sus puertos a disposición del usuario, es necesario tener una tarjeta de acoplo con las mismas características a nivel de los sockets o pines de salida del BeagleBone Black y que permita al usuario de manera rápida y sencilla interconectar los componentes de entrada y salida.

Para la realización de esta tarjeta o Circuito impreso se tomó en cuenta factores como:

- Potencia.
- Voltajes del BeagleBone Black.
- Consumo de Corriente.
- Contactos abiertos para uso general.
- Señales digitales.
- Ergonomía.

La placa de circuito impreso fue diseñada con un programa de diseño de circuitos impresos llamado Altium Designer 13, el cual nos permitió esquematizar tanto nuestras entradas como salidas para poder obtener el resultado deseado a su vez dicho programa permitió la simulación de dicha interfaz para comprobar que el circuito a diseñarse representara los resultados esperados.

2.3.1 Proceso de diseño del circuito impreso.

En el programa de diseño antes mencionado fue creado el proyecto llamado PANEL BLINDADO BEAGLEBONE, esta es la representación esquemática de los componentes del circuito de interfaz con el BeagleBone Black, la cual contiene los dispositivos de interfaz entre nuestros componentes del sistema y el BeagleBone Black. En la "Figura 39" se muestra la hoja principal de esquemático la cual contiene los puertos de entrada y salida del BeagleBone Black, ya que la interconexión de los pines tiene que ser simétrica a los pines del BeagleBone Black para poder interconectarse ya que la disposición de los mismo no puede ser variada y es necesario que estos caigan de manera similar y a su vez el pin out del BeagleBone Black debe preservarse para no cometer errores que puedan afectar el funcionamiento del mismo. Este mismo esquemático cuenta con el relay o accionador que activará la cerradura electrónica para abrir la puerta del panel blindado, también se puede ver los bloques de los otros esquemáticos que componen el proyecto y los componentes electrónicos los cuales tiene el circuito.

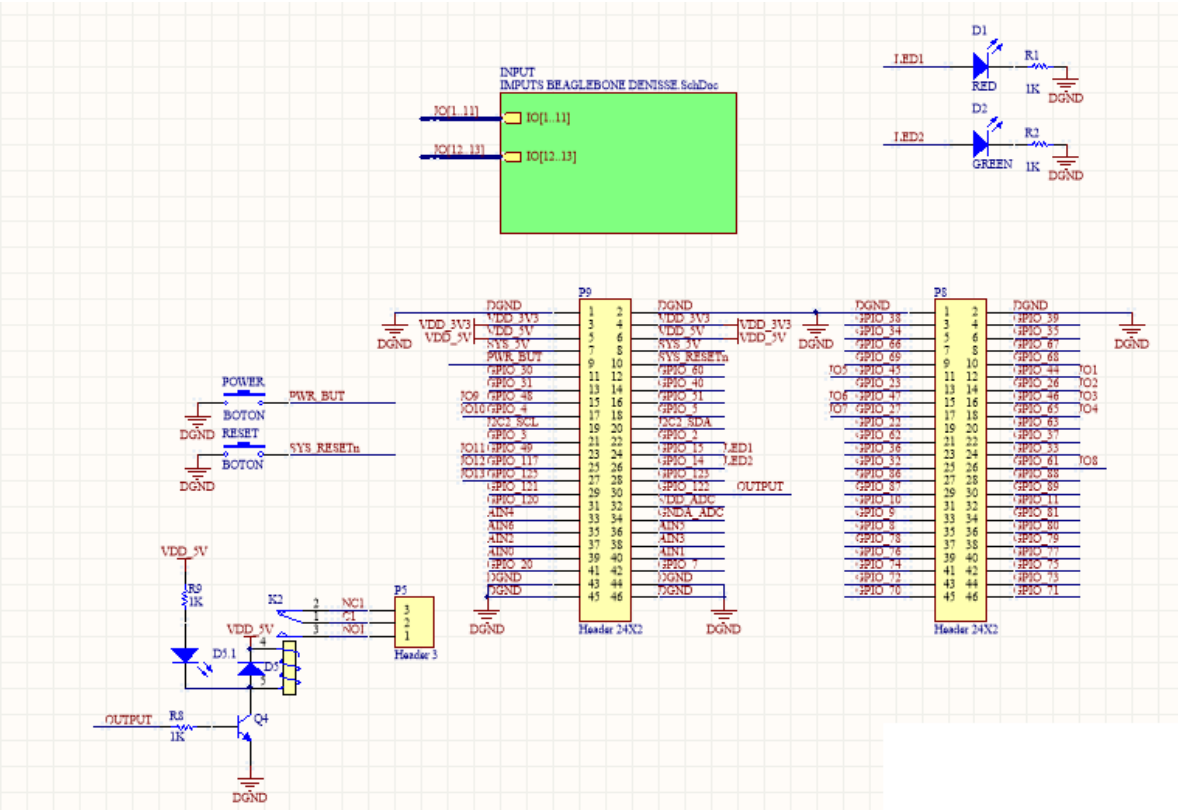


Figura 39: Esquemático Principal del Circuito Impreso.

Los botones que se muestran en el esquemático designados por los nombre POWER y RESET son una extensión de los botones de encendido y reseteo los cuales contiene el BeagleBone Black pero como el circuito impreso de interfaz se coloca sobre el BeagleBone Black no se podrá tener acceso a esos botones y es necesario que el usuario lo tenga a fácil acceso para su manejo sin tener que remover el circuito de interfaz.

Como el ideal es manejar entradas que se interconecten con el BeagleBone la “Figura 40” muestra dos entradas digitales utilizadas para conectar los componentes de entrada del sistema como son los sensores. Y cuenta con el conector para el teclado que permitirá

manejar la clave para el acceso al panel, cada botón se interconecta con el BeagleBone Black para realizar su validación.

Todas las entradas digitales están referenciadas a 3.3V que es el voltaje de trabajo del BeagleBone Black aunque el mismo también tiene una entrada referenciada a los 5V DC, ya que la alimentación del mismo se hace por la vía del USB o una fuente externa de 5V.

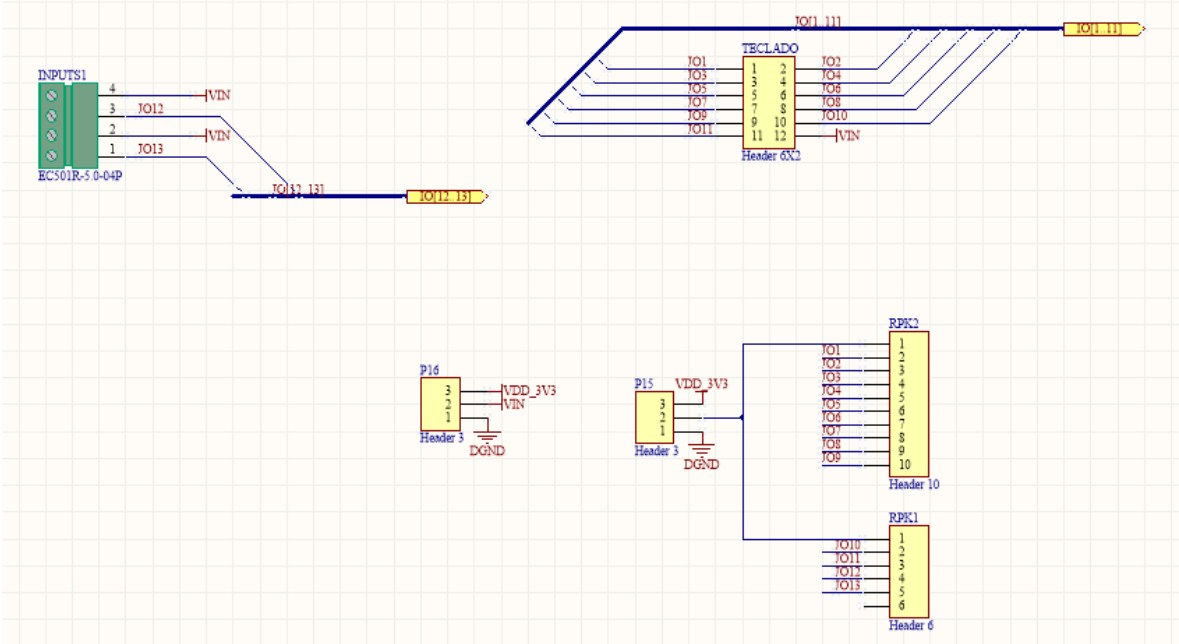


Figura 40: Esquemático de la interconexión de componentes de entrada.

2.3.2 Diseño del circuito impreso de interfaz.

Un circuito impreso, tarjeta de circuito impreso o PCB (del inglés printed circuit board), es una superficie constituida por caminos o pistas de material conductor laminadas sobre una

base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de los caminos conductores, y sostener mecánicamente por medio de la base, un conjunto de componentes electrónicos. Los caminos son generalmente de cobre mientras que la base se fabrica de resinas de fibra de vidrio reforzada (la más conocida es la FR4), cerámica, plástico, teflón o polímeros como la baquelita.

Este circuito impreso cumple con una serie de características fundamentales para su funcionamiento, como son aspectos de funcionamiento y estética. Dentro del funcionamiento están aspectos como el tamaño de las pistas dependiendo de la corriente que manejan, componentes que van colocados cercanos por la función que realizan, interconexión adecuada de dispositivos etc.; por otro lado está la estética del circuito que no es más que la forma y tamaño que este puede tener para su aspecto final. En la "Figura 41" muestra el diseño del circuito impreso, este cuenta con los conectores de interconexión entre la placa de circuito impreso y el BeagleBone que necesariamente tienen que tener la disposición, tamaño y características de los pines los cuales tiene el BeagleBone Black.

Otro factor determinante es la disposición de las entradas y las salidas en los extremos de la placa para estar más cerca con los dispositivos que maneja. Un factor crítico es el tamaño de las pistas del circuito que en el caso de las pistas de alimentación y las de los relés tiene que contar con un tamaño mayor por la corriente que pasara por las mismas.

También cuenta con el terminal que conecta el teclado numérico que permitirá la validación de los pines de acceso al panel.

En definitiva la “figura 41” muestra todos los componentes esquematizados anteriormente dispuestos para que el usuario pueda trabajar con el PCB.

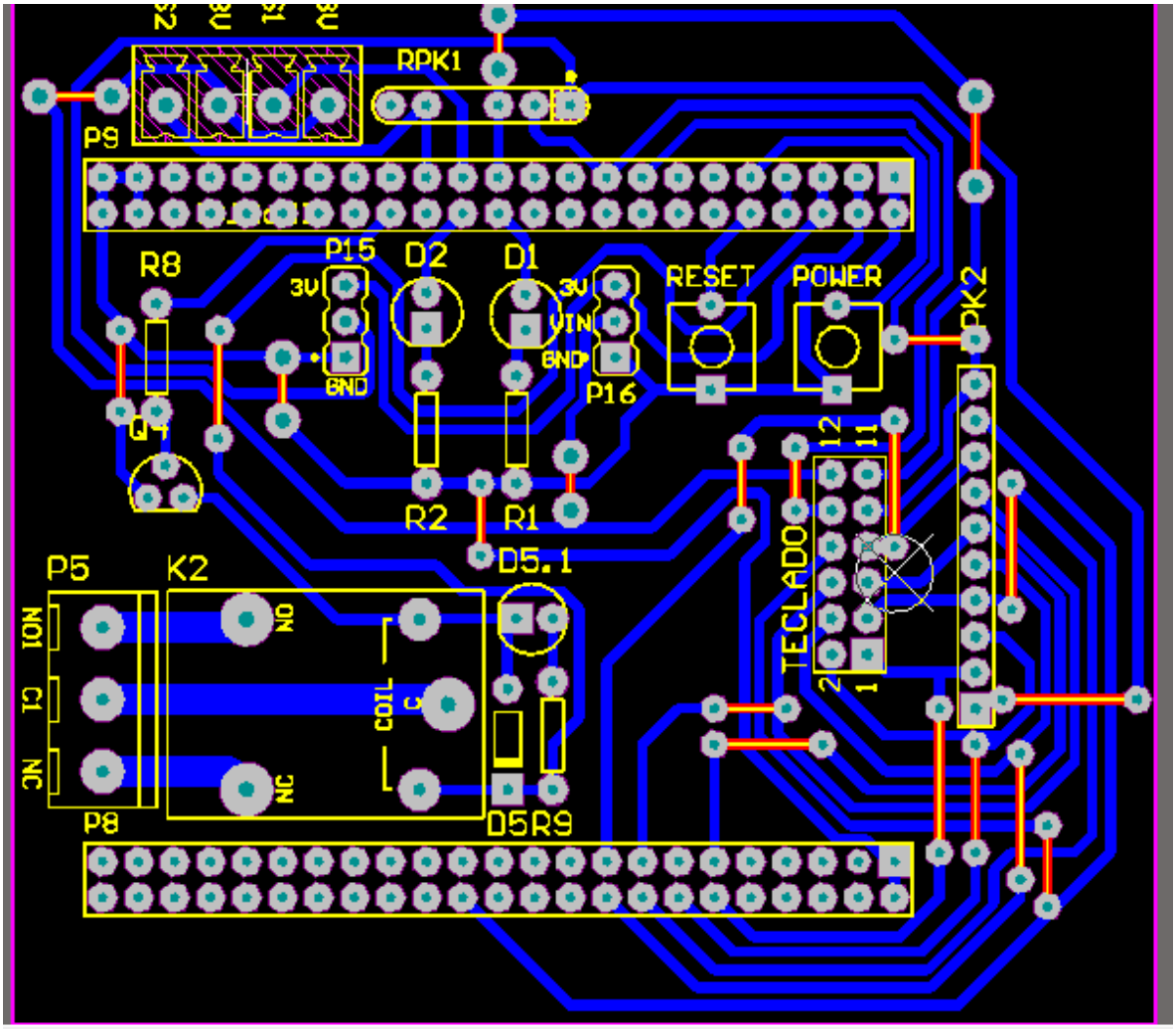


Figura 41: Diseño PCB del Circuito de Interfaz.

2.4 Diseño de la interfaz gráfica

La realización de la interfaz gráfica del sistema de alarma con BeagleBone Black está basada en un programa computación con el cual un usuario administrador podrá observar en tiempo real las alarmas del sistema y poder llevar un record con los reportes de las alarmas activadas.

El BeagleBone Black internamente contiene un entorno de desarrollo de aplicaciones JavaScript y Node.js, así como HTML, CSS, PHP, Java, Ruby y otros 23 lenguajes de programación. Esto permite programar en los diferentes lenguajes de programación el BeagleBone Black “(ver fig.42)”.

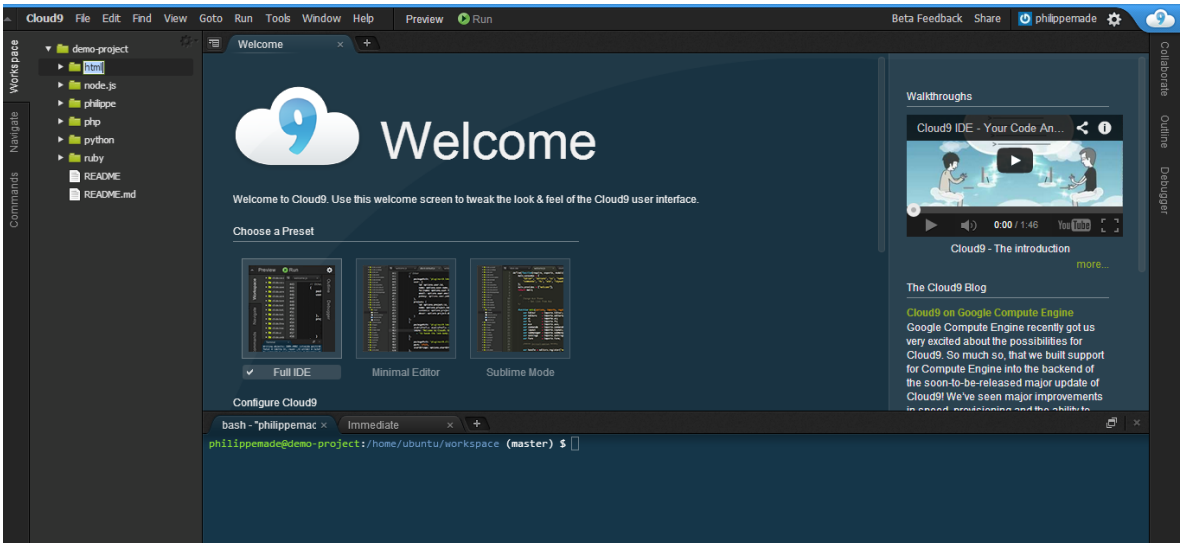


Figura 42: Muestra el entorno visual del Cloud9.
Fuente: clou9.com

El BeagleBone Black, además de contener el Cloud9 como herramienta de desarrollo de aplicaciones, internamente posee librerías orientadas al uso de los recursos del mismo, así como el manejo de los pines entradas y salidas, las entradas analógicas, el manejo de las pantallas Gráficas y otros módulos que este contiene. Específicamente para el manejo del hardware, al momento de hacer la comunicación de los pines entrada y salida se utilizó la librería Bonescript está permite a través de la programación JavaScript una comunicación más fácil y optima con los pines del BeagleBone Black.

El lenguaje de programación principal que se utilizó para desarrollar el sistema de alarma es el JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos,

imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets).

Este lenguaje integrado con las librerías internas del BeagleBone Black nos permite mediante la programación conocer los estados de los sensores conectados mediante el hardware de interfaz del BeagleBone y los sensores, sirenas y luces que contiene el sistema de alarma mencionado en el capítulo anterior.

El procedimiento para accesar a los pines es el siguiente:

El BeagleBone contiene 65 pines entrada-salida, esto quiere decir que esos pines pueden hacer ambas funciones dependiendo de cómo sea declarado dentro del programa principal, para colocar un pin como entrada o salida utilizando la librería de Bonescript el procedimiento se muestra a continuación:

varBonscript = require('bonescript'); Se declara la librería del Bonescript para el uso de la misma dentro del programa.

Bonscript.pinMode('USR0', Bonscript.OUTPUT)/ Bonscript.pinMode('USR0', Bonscript.INPUT): Con la siguiente sentencia se está colocando el Pin USR0 como salida en la primera parte o como entrada en la segunda parte. La sintaxis de la sentencia está compuesta por:

Nombre de la librería= Bonescript

La instrucción que permite el manejo de los pines para determinar si son entrada o salida= PinMode.

Dentro del paréntesis se declara el pin del BeagleBone que se desea colocar como salida o entrada, luego separado por una coma (,) se coloca el nombre de la librería con el estado del pin que se desea ya sea entrada o salida, refiriéndose a estas con sus nombres en el lenguaje inglés Input – Output.

Para el uso ya directamente del pin para escribir o leer, un uno lógico que en el caso del BeagleBone Black son 3.3V DC o un 0 lógico que son 0V DC es necesario del uso de la siguiente Sintaxis:

- Bonescript.digitalWrite(pin, estado)
- Bonescript.digitalRead(pin, x)

El Bonescript es el nombre de la librería mencionado anteriormente, el uso de digital Write es para escribir al pin deseado del BeagleBone Black, mientras que digitalRead es para leer

un pin del BeagleBone Black luego dentro del paréntesis se coloca el pin deseado para ambos casos. Para la escritura se coloca si lo que se desea es el 1 lógico conocido como HIGH en el lenguaje de programación o el 0 lógico conocido como Low para que este pueda poner el estado deseado por el programador. En el caso de las entradas la x dentro de la sintaxis puede ser cualquier etiqueta que nos permita hacer una función de lectura, para conocer el estado de este pin.

Estas instrucciones son las que ayudan dentro del programa principal, desarrollar la interfaz entre el BeagleBone Black y los componentes del sistema.

2.5 Implementación del sistema

Para la implementación del sistema de alarma en los paneles blindados se debe tener en cuenta la localidad donde estará instalado la central de monitoreo y hacer levantamiento del área para la adquisición e instalaciones de los equipos de monitoreo que se van a utilizar. Definir cómo debe conformarse el equipo de seguridad con la debida experiencia en dicho ámbito. Definir procedimientos y normas de respuestas a dichas alarmas generadas.

Se debe verificar cada una de las localidades remotas para así evaluar en cada una de ellas cuales y en qué lugares serán instalados los dispositivos de seguridad. Es necesario realizar

una evaluación de riesgo profundo para determinar los puntos críticos y peores escenarios que puedan ocurrir en los distintos puntos.

Es importante que el lugar donde se instalen los detectores sean los más idóneos para contar con la máxima anticipación posible de alarma.

Se diseñará un esquema de cada panel que mostrará los lugares que pueden ser instalados los dispositivos, tomando en cuenta, donde pueden ser los lugares más apropiados para que no ocasionen falsa alarma o bien no poder detectar alguna anomalía. Es importante y necesario consultar los planos de instalación del fabricante ya que estos indican el tendido y las conexiones de cables en forma especialmente clara para cumplir con los requisitos de supervisión eléctrica de circuitos. Cuanto más rápido se pueda determinar la ubicación de la alarma activada, más rápido se podrá responder a la misma.

Antes de poner el sistema en servicio, se realizaran pruebas a los circuitos para verificar que no haya ningún tipo de anomalía como cortocircuitos a tierra, cortocircuito entre conductores ni circuitos abiertos. Después de instalar todos los detectores y probar que el sistema no tenga problemas de conexión.

Por último es necesario crear un registro de mantenimiento para cuidar que siempre el sistema de alarma esté en buenas condiciones de funcionamiento. Este se puede hacer de la siguiente manera:

- Capacitar el personal que le estará dando mantenimiento al panel.

- Llevar un registro de mantenimiento de los sensores en el cual se asiente la fecha de mantenimiento, prueba y limpieza de cada sensor del sistema.

- Conservar un archivo completo de toda la información pertinente al sistema de alarma, en un lugar de acceso inmediato. Dicho archivo debería contener las especificaciones técnicas y las instrucciones de instalación de detectores, panel de control, dispositivos auxiliares y diagramas.
- Conservar registros y datos sobre las localidades donde estén instalados los paneles y modificaciones de todos los sistemas electromecánicos del edificio que puedan interferir con el sistema de alarma. Entre esos datos se tendría que incluir esquemas de tendidos de conductores y fechas de implementación de los cambios, para poder encontrar rápidamente la solución de problemas que puedan estar relacionados con esos trabajos.

- Mantenerse en contacto con los equipos para verificar la conexión con la red de datos móviles.

CAPÍTULO 3.VALIDACIÓN DEL DISEÑO DEL SISTEMA DE ALARMA Y MONITOREO

3.1. Descripción de las pruebas del sistema

Para realizar las pruebas del sistema, se tiene que tomar en cuenta que como su nombre lo dice, es un sistema compuesto por varias partes y en este capítulo serán explicadas cada parte por separado para la mejor comprensión de las mismas.

3.1.1. Pruebas del hardware.

El BeagleBone Black posee dos formas de alimentación de voltaje:

1. USB 5V DC

2. DC-Jack

Estas formas alimentan con 5V DC el dispositivo, para este caso se alimentará por la opción 2 con el DC-Jack conectado a una fuente de 5V DC a 2Amp que estará conectado a un suministro eléctrico conectado directamente a la línea de voltaje que suministra las Distribuidora dentro del panel blindado, ya que el circuito de interfaz descrito en capítulos

anteriores necesita de este voltaje de alimentación para que los componentes electrónicos como led, relés, funcionen.

Lo principal es conectar el BeagleBone Black con la fuente de alimentación de 5V y verificar que este encienda correctamente“(ver fig.43)”.

Figure 43: Conexión del BeagleBone Black a la Fuente de alimentación

Luego de estar conectado y encendido se procede a conectar el BeagleBone Black dentro de una red local con conexión a internet, esto lo podemos lograr conectándolo vía Ethernet o utilizando un adaptador de red inalámbrico USB para este sistema se utilizará una conexión Ethernet por los beneficios de seguridad y rapidez que permite este tipo de conexión, el router al cual el BeagleBone Black esté conectado necesita tener el Puerto 8888 abierto, ya que este es el puerto de conexión el servicio Web generado por el BeagleBone Black para este ejemplo utilizamos un router D-Link Dir-601 el cual para colocar el puerto necesario se deberá seguir los siguientes pasos:

Para acceder a las configuraciones del router es necesario utilizar un computador conectado al router, se coloca la dirección IP del router que viene pre configurado de fábrica, Este pide validar el usuario administrador, luego de que se está dentro buscamos la opción servidor virtual.

Este router se puede configurar como servidor virtual, de forma que los usuarios remotos que accedan a la web o a servicios FTP a través de la dirección IP pública puedan ser automáticamente redireccionados a servidores locales en la red LAN. La función de cortafuegos del router filtra paquetes no reconocidos para proteger la red LAN, de forma que todos los ordenadores conectados en red con el router sean invisibles al resto del mundo. El usuario puede hacer que algunos de los ordenadores LAN sean accesibles desde Internet activando la opción Virtual Server.

En función del servicio solicitado, el router redirecciona la solicitud de servicio externo al servidor adecuado dentro de la red LAN. El router también es capaz de redireccionar puertos, lo cual significa que el tráfico de entrada a un puerto concreto se puede redireccionar a otro puerto del ordenador servidor.

Antes de realizar estas funciones debemos conocer el IP de nuestro BeagleBone Black para así colocar el IP donde se va a redirecionar el IP público cuando se solicite la entrada desde el exterior.

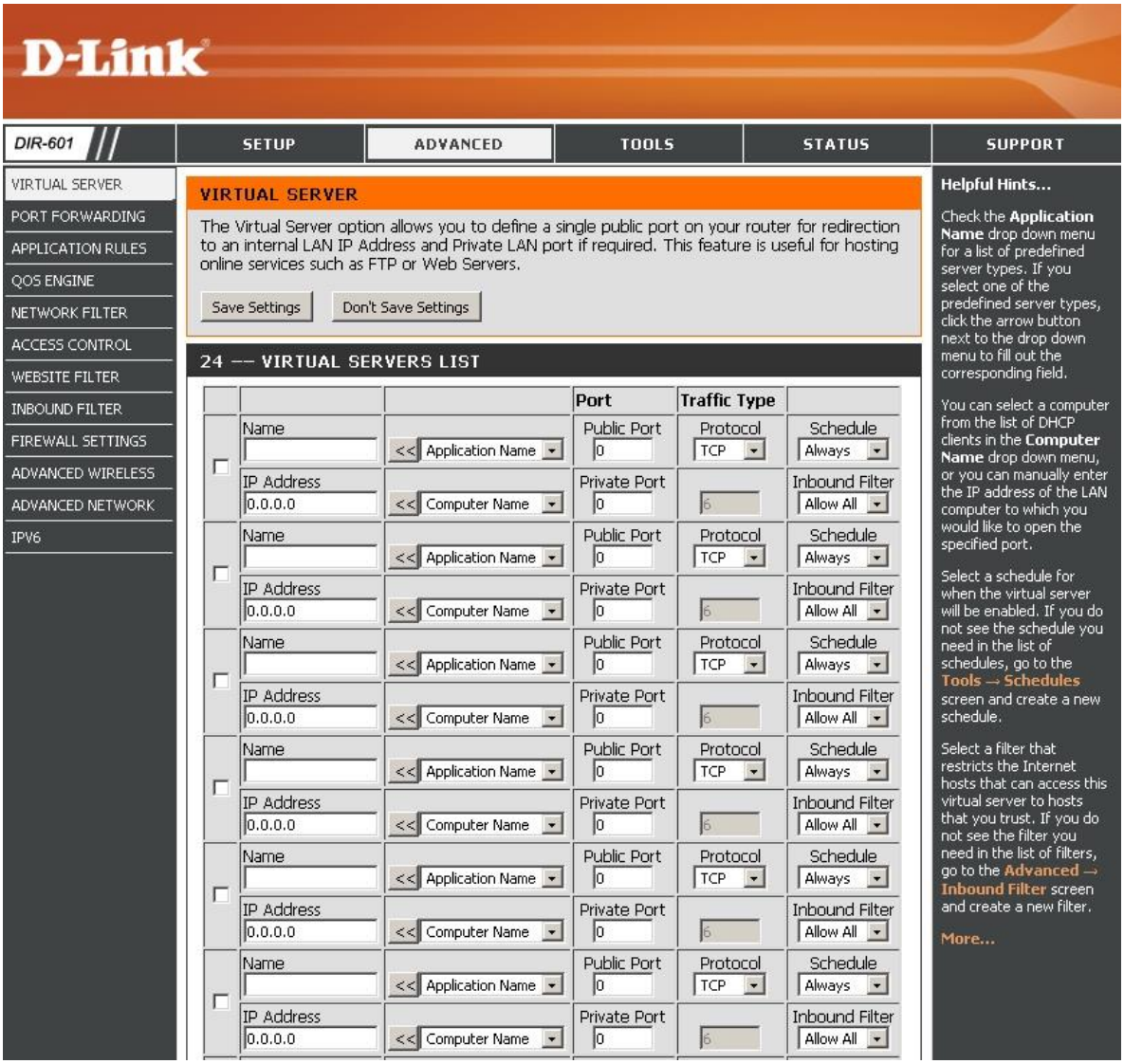


Figura 44: Página de configuración de puertos del Router D-Link Dir-601

Después de abrir los puertos y conociendo el IP del BeagleBone Black ya se encuentra listo para proceder a entrar desde el exterior a nuestro BeagleBone Black.

Para el sistema de alarma el IP es 192.168.0.100:8888. Así se accede de manera local, si es para acceder de manera remota es necesario colocar el IP pública: El puerto.

Cuando el BeagleBone Black está en la red local con conexión a internet con su IP y Puertos colocados y abiertos se podrá proceder a conectar nuestro Circuito de Interfaz. Figura 45.

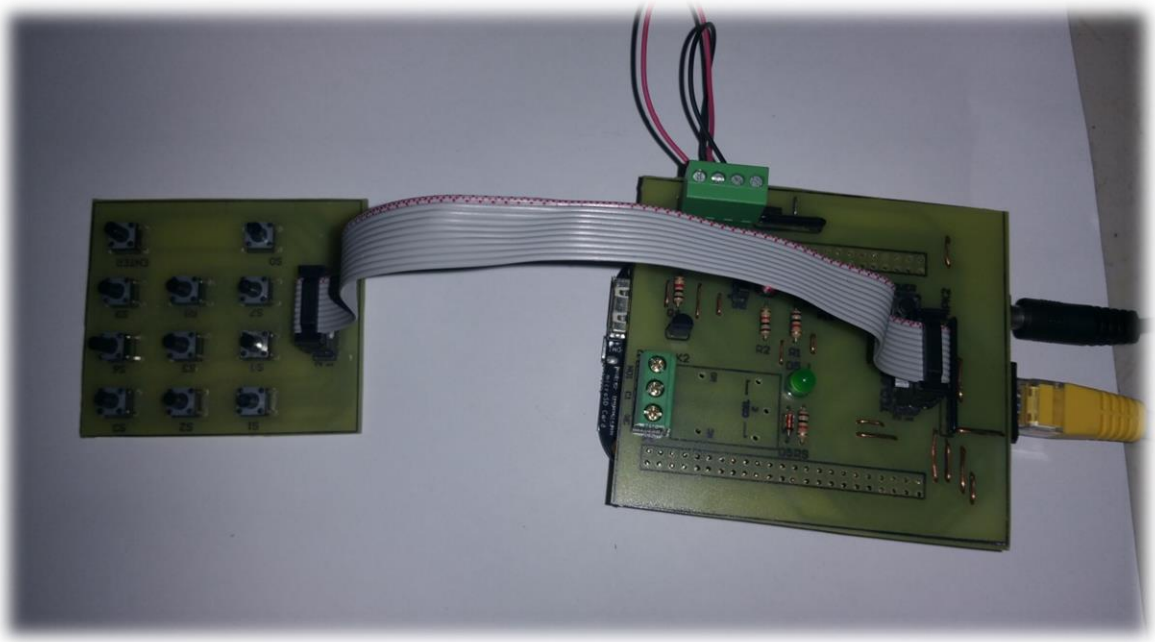


Figura 45: Circuito de interfaz entre el BeagleBone y los componentes del sistema.

El circuito de la "figura 45" permite interconectar los componentes del sistema entre los que se encuentran los distintos sensores, teclado y la cerradura eléctrica. Los cuales componen los elementos principales de monitoreo. Dicho circuito PCB contiene 2 entradas digitales referenciadas a 3.3 V DC, un puerto para conectar el teclado y 1 salidas manejadas por relés. El mismo tiene los pulsadores de encendido y reseteo del BeagleBone Black, ya que el

BeagleBone Black está debajo de esta placa. La disposición de los compontes del sistema en el Circuito de interfaz está detallado para su rápida comprensión “(Ver Tabla 2)”.

CIRCUITO DE INTERFAZ DEL BEAGLEBONE BLACK		
COMPONENTE	**PIN BEAGLE**	**ESTADO**
SENSOR MAGNETICO	GPIO125/P9-27	ENTRADA
SENSOR DE VIBRACION	GPIO117/P9-25	ENTRADA
LED INDICADOR VERDE	GPIO14/P9-26	SALIDA
LED INDICADOR ROJO	GPIO15/P9-24	SALIDA
CERRADURA ELECTRICA	GPIO122/P9-30	SALIDA
BOTON TECLADO 1	GPIO44/P8-12	ENTRADA
BOTON TECLADO 2	GPIO26/P8-14	ENTRADA
BOTON TECLADO 3	GPIO46/P8-16	ENTRADA
BOTON TECLADO 4	GPIO65/P8-16	ENTRADA
BOTON TECLADO 5	GPIO45/P8-11	ENTRADA
BOTON TECLADO 6	GPIO47/P8-15	ENTRADA
BOTON TECLADO 7	GPIO27/P8-17	ENTRADA
BOTON TECLADO 8	GPIO61/P8-26	ENTRADA
BOTON TECLADO 9	GPIO48/P9-15	ENTRADA
BOTON TECLADO 0	GPIO4/P9-17	ENTRADA
BOTON TECLADO ENTER	GPIO49/P9-23	ENTRADA

Tabla 2: Muestra la leyenda de los pines donde los componentes del sistema están conectados.

Conociendo la designación de pines, se procede a interconectar todos los componentes del sistema en el circuito de interfaz para verificar que todos funcionan correctamente. Las pruebas que se realizan estas, verifican si los sensores se accionan al momento de encontrar anomalías, que la cerradura eléctrica encienda cuando la señal es enviada y los botones del teclado de validación estén funcionando correctamente. Una vez todo listo y conectado se

procede a realizar las pruebas con el servidor Web, ya que este es el que contiene todo el programa el cual permitirá controlar el BeagleBone Black y los componentes del sistema.

3.1.2 Pruebas del programa.

El programa computacional, que controla el monitoreo del sistema de alarma, es la pantalla de interfaz con los usuarios administradores del servidor del BeagleBone Black, esta pantalla mostrara todas las características, estados y eventos de los componentes del sistema. Para abrir dicho programa se corre el ejecutable del mismo que será colocado en cada computadora de las personas que estarán monitoreando los paneles. A dicho programa le ha llamado Real time Locker Manager (RLM). (Ver Apéndice A y B)

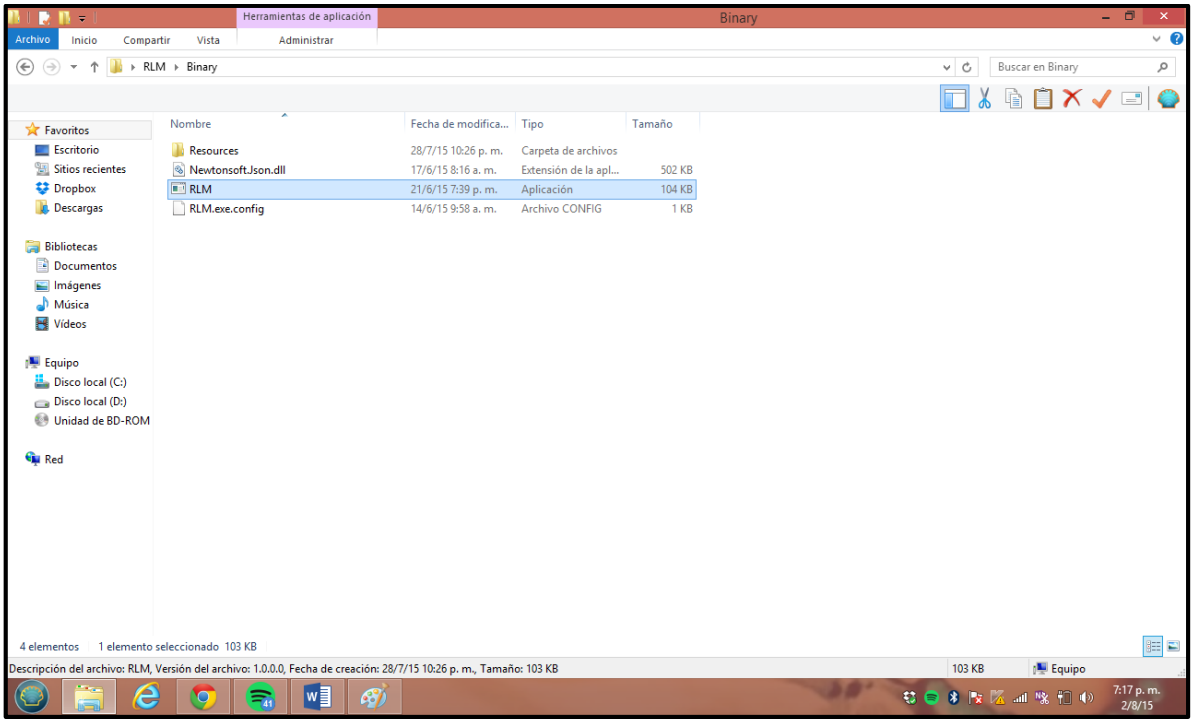


Figura 46: Abriendo la aplicación.

En la "figura 46" se muestra como abre el programa colocado en el computador local que manejara los distintos paneles. Este programa se encarga de hacer el llamado al servidor dentro del BeagleBone Black.

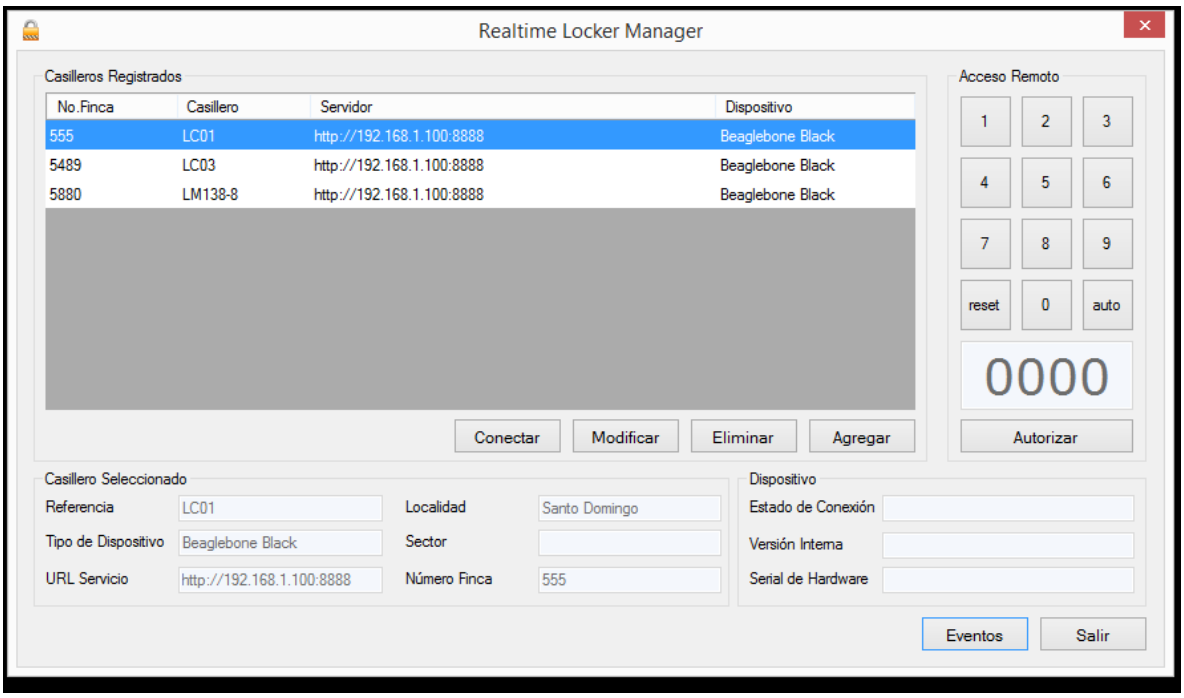


Figura 47: Entorno principal del programa. Rubro sector

La "figura 47" muestra el entorno del programa para las pruebas a nivel de programa que realizará, estarán conexión con el BeagleBone Black, la creación de paneles, validación de los sensores y la validaciones de pines de acceso. Ver Apéndices B y C, muestran todo el código detrás de la interfaz completa del programa.

La ventana principal del programa el usuario puede observar una tabla donde están las distintas localidades de paneles ya enlazados al programa, también puede ver todas las informaciones de la localidad, que dispositivo está instalado y las referencias del mismo. Cuenta a su vez con un teclado numérico para las autorizaciones de acceso al panel y poder ver las alertas del sistema.

Creación de localidades de Paneles

Para crear una localidad remota de un panel blindado, es necesario darle un clic al botón "Agregar" en la pantalla principal del programa. "(ver fig.48)". Esta opción nos despliega una ventana donde ingresamos todas las características del dispositivo remoto (BeagleBone Black) en el panel blindado. "(ver fig.49"). Entre las características que el usuario del programa ingresa están:

Referencia Casillero: Aquí el usuario del programa, coloca una referencia de donde está ubicado el panel, específicamente puede ser la referencia de la zona dentro del sistema eléctrico nacional se encuentra dicho panel.

Tipo de dispositivo: Esta opción permite seleccionar el tipo de dispositivo al cual se conecta el programa, para nuestro caso nuestro dispositivo remoto es el BeagleBone Black.

URL Servicio: Esta es la opción donde se coloca la IP pública de internet y el puerto del dispositivo que previamente fue configurado.

Localidad: Permite colocar el condominio, residencial o apartamento donde el panel blindado y el dispositivo remoto está ubicado.

Sector: Esta opción permite colocar el sector donde está ubicado dicho panel.

Finca: Propiedad rural o urbana que determina el emplazamiento, o la situación de uno o más servicios de carácter permanente o eventual.

Número de Finca (NIF): Este es el número por el cual los paneles están identificado dentro del sistema eléctrico nacional.

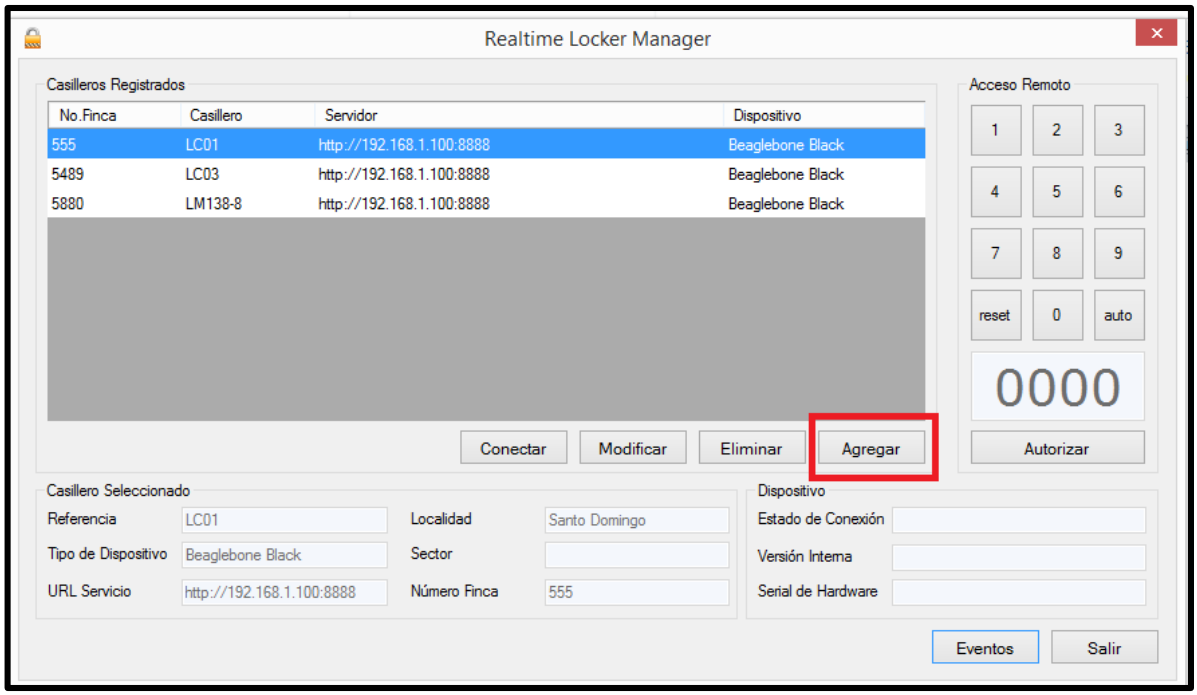


Figura 48: Señalando botón agregar.

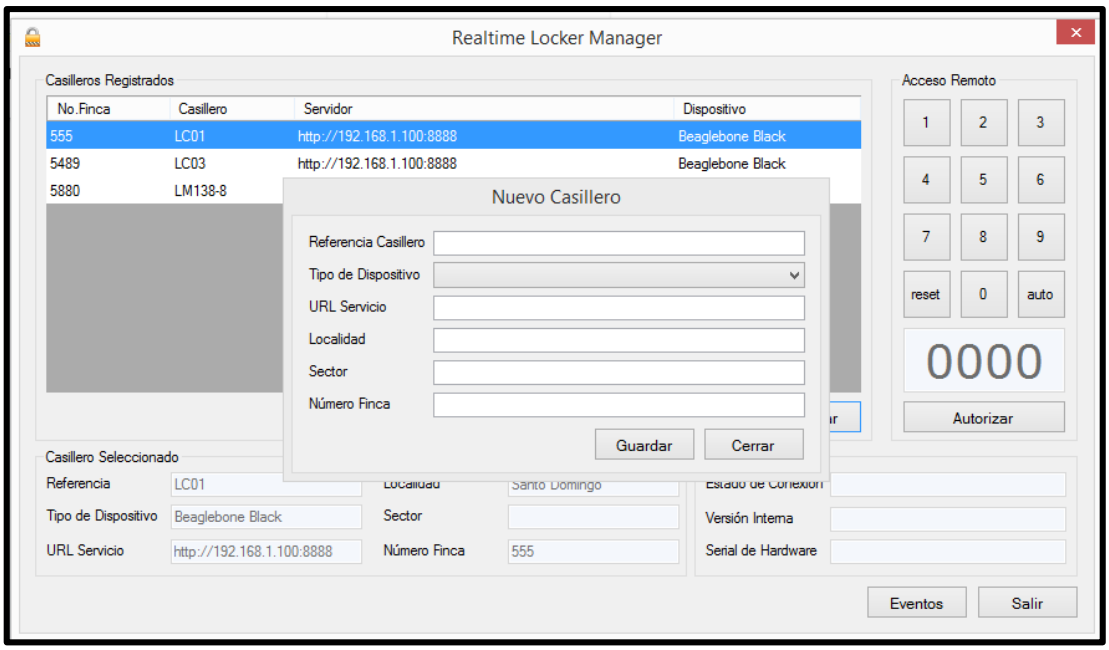


Figura 49: Ventana de Creación de Punto remoto.

Luego de colocada la información del panel, se procede a guardarla y ya en la ventana principal del programa se puede observar dicho panel agregado correctamente. Luego de esto, se procede a conectar al dispositivo remoto utilizando el botón de Conectar en la ventana principal del programa.

En dado caso, de ser necesario, el programa le permite modificar o borrar las informaciones de los puntos remotos ya creados. “(Ver Figura 50)”

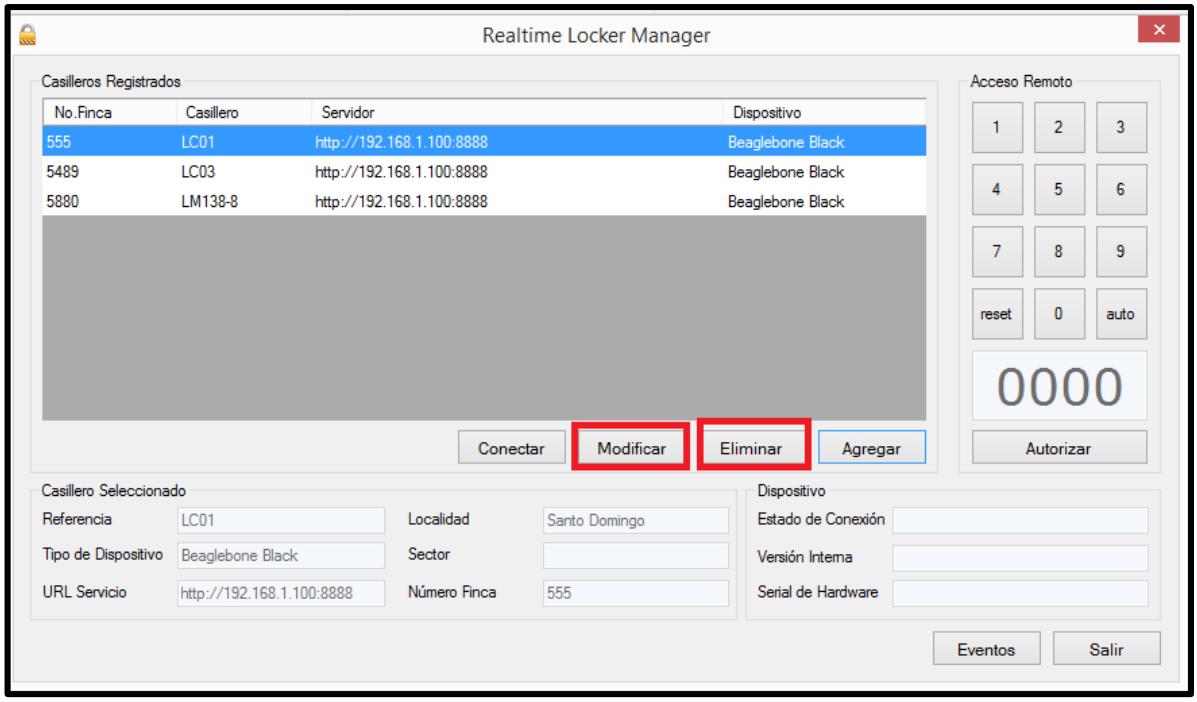


Figura 50: Señalización de Botones de Modificar y Eliminar.

Validación de la conexión

Luego de presionar el botón de conectar el programa comprueba si existe conexión entre el BeagleBone Black remoto en el Panel Blindado y el programa. Si la validación es correcta y existe una conexión aparece en pantalla una ventana confirmando la conexión. “(Ver Figura 51)”. En el caso contrario, que por alguna razón no exista una conexión correcta aparecerá en pantalla una ventana indicando que no hubo conexión. “(Ver Figura 52)”

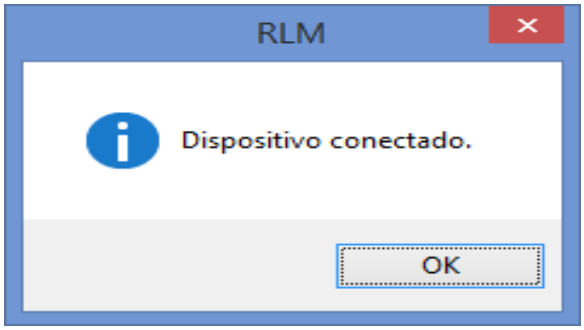


Figura 51: Ventana de Conexión exitosa.

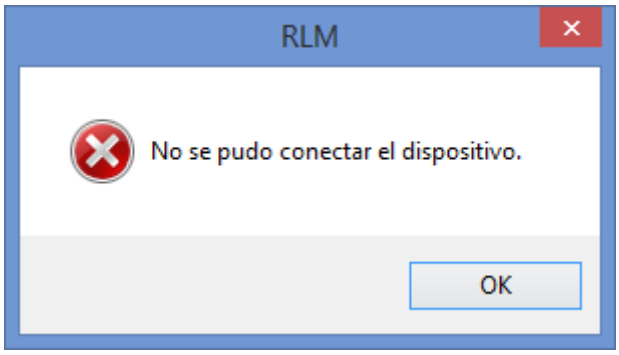


Figura 52: Ventana de Conexión errónea.

Códigos de validación de accesos

Luego de estar conectados satisfactoriamente el usuario puede visualizar en tiempo real lo que está pasando con el panel, si algún empleado o contratista de las distribuidoras de energía necesiten dar algún mantenimiento dentro del panel o realizar cualquier tipo de medición, tendrá que comunicarse con la central de monitoreo y el usuario del programa mediante el teclado numérico que contiene el programa puede generarle un código aleatorio o introducir uno manualmente y este lo envía al BeagleBone Black remoto para su validación, luego de

enviado, el usuario del programa le suministra al empleado o contratista de la Distribuidora el número de acceso que tiene que digitar en el teclado para poder abrir dicho panel. "(ver fig.53)".

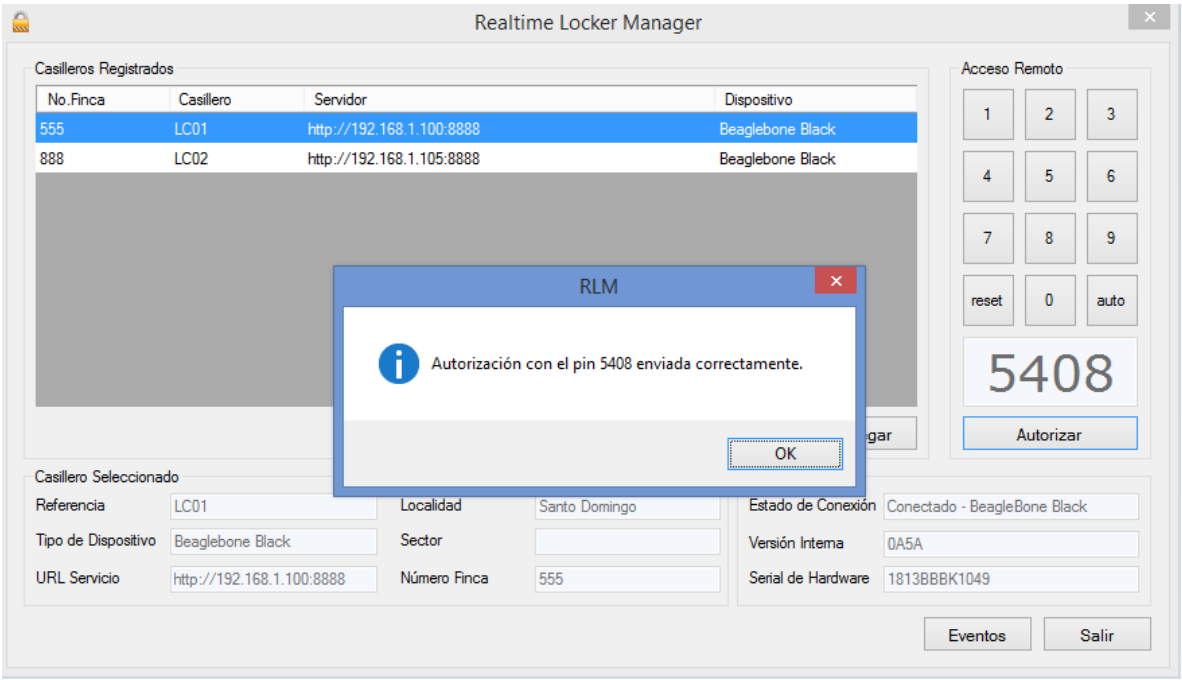


Figura 53: Ventana de autorización del código de acceso.

Cuando la persona frente al panel introduce el pin de acceso provisto por la central el panel procede abrirse y todas las informaciones de acceso y alarmas son desplegadas en la pantalla de eventos del programa, cuando la puerta del panel se cierra ya el código suministrado en invalidado, en caso de un nuevo ingreso el empleado o contratista tendrá que volver a repetir el proceso.

Visualización de estados

Figura 54: Ventana histórico de eventos.

La "figura 54"muestra los eventos generados por el BeagleBone Black, todas las alarmas y actividades que suceden en el punto remoto, entre los eventos que se pueden visualizar en dicha pantalla se encuentran:

Pin Válido: Cuando el pin de acceso es digitado correctamente.

Pin Incorrecto: Cuando el pin de acceso es digitado incorrectamente. Y en caso de que alguien intente accesar al panel colocando pines de accesos incorrectos.

Violación de Integridad física: Cuando el sensor magnético o de vibración detectan cualquier tipo de violación o anomalía.

3.2 Resultados de las pruebas

Los resultados de las pruebas realizadas fueron totalmente favorables, ya que se probaron cada uno de los componentes del sistema (sensores y teclado) y como estos reaccionaban con su conexión al BeagleBone Black.

El resultado de la primera prueba fue cuando se comprobó la conexión del punto remoto con el programa en la central de monitoreo, se demostró que había conexión, se abrió la puerta del panel sin el pin de acceso y a su vez se trató de forzar el panel blindado provocando una violación de seguridad, los parámetros para los cuales está diseñado estos componentes proporcionaron la respuesta adecuada y pudiéndose visualizar esta información en el histórico de eventos del programa.

Uno de los resultados fundamentales del sistema es el almacenamiento de eventos generados, ya que es allí donde el usuario lleva una bitácora de las alarmas generadas. En la "figura 55" muestra la sección de eventos donde podemos visualizar el evento, la fecha y hora en las cuales aconteció algún tipo de evento en el dispositivo remoto. Esta información es almacenada en el programa de manera automática.

Dentro de las pruebas realizadas se envió un código desde el programa al dispositivo remoto para que el empleado o contratista lo coloque, cuando este lo colocó, se comprobó que tanto el LED verde que indica que el pin es correcto como la cerradura que abre la puerta del panel se activaron, luego que ese realizó el mantenimiento y al cerrar la puerta intentó entrar con el mismo código de acceso provisto previamente pero no pudo accesar ya que el código se invalida inmediatamente la puerta es cerrada. En este último caso se comprobó que el LED rojo y el histórico de eventos se accionaron mostrando que el pin fue incorrecto.

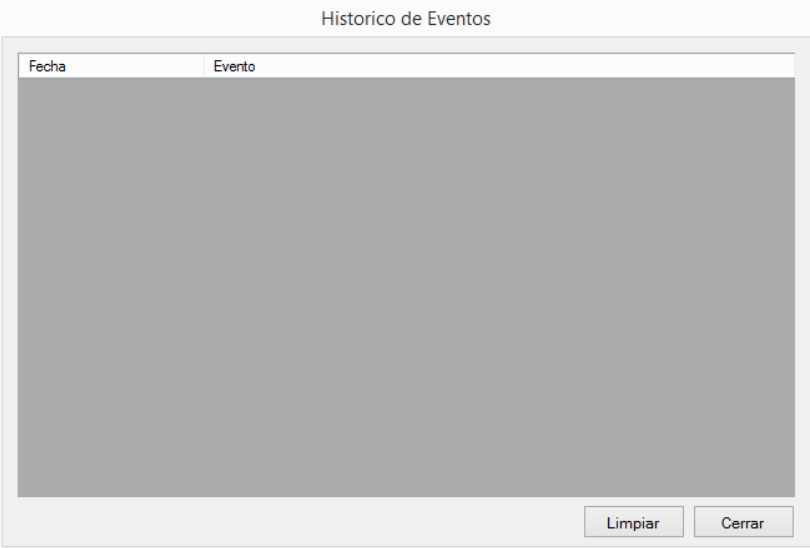


Figura 55: Reporte de Eventos.

3.3 Comparación de las pruebas con los cálculos del diseño

En base a los logros obtenidos por las pruebas realizadas, se concluye, que comparando las mediciones de cada sensor dentro del sistema y los resultados de las señales enviadas desde el teclado, los cálculos realizados dieron como resultado una conexión exitosa.

CONCLUSIONES

La creación de un sistema de alarma y monitoreo para los paneles blindados de las compañías Distribuidoras de Electricidad con un dispositivo como el BeagleBone Black, es una forma innovadora de llevar a cabo dicho sistema. La característica de ser un hardware totalmente abierto e independiente, donde, el usuario mediante la programación puede manejar las características internas del dispositivo.

El sistema de alarma y monitoreo de manera general verificó cada sensor, los pines de acceso y la apertura de la cerradura y mostró a través del programa los estados en tiempo real, mostrando las características y generando reportes almacenados en el programa, permitió así al usuario ver las alarmas generadas ,en caso de no estar presente en el momento.

Los sistemas de seguridad remotos son la tendencia del momento ya que permite el monitoreo desde diferentes lugares y mantener siempre vigilado las propiedades y bienes preciados de las personas. Contribuyendo así, en que no tiene que existir una persona física verificando en el momento lo que está sucediendo en cada localidad. El sistema eléctrico nacional cuenta

con una gran cantidad de paneles blindados y tener un personal que se encargue de verificar cada uno independiente es una tarea difícil y costosa.

Sistemas embebidos como el BeagleBone Black, permiten el desarrollo de innumerables aplicaciones, siendo el caso que las herramientas o partes que se le pueden agregar al sistema de alarmas podrían hacer dicho Sistema de Alarma y Monitoreo, una central donde se pueda visualizar cualquier paramento dentro y fuera del panel blindado.

RECOMENDACIONES

Los sistemas embebidos de gran potencia como el BeagleBone Black, permite manejar innumerables aplicaciones. En el sistema de alarma y monitoreo, se puede agregar características que harían el sistema más completo, entre estas características están:

- Monitorear los medidores eléctricos para conocer los consumos de las viviendas.
- Sensar y mostrar dentro del programa factores climatológicos.
- Mostrar en el programa parámetros de los equipos de medición dentro del panel blindado como son: Voltajes a los que los equipos están trabajando, temperatura de los equipos, entre otros.
- Medir los voltajes de la línea de energía que suministra los medidores del panel.
- En caso de que exista más de un generador eléctrico controlar horas de uso y hacer transferencia entre estos.
- Controlar la iluminación interior y exterior de la localidad donde está instalado.
- Controlar el consumo de energía de algunos equipos dentro de la localidad donde estén instalados, los cuales se puedan apagar en horas que no se encuentre personal presente o que estén en desuso.
- Enviar las alarmas por correo electrónico o mensajes al celular del encargado de seguridad.

Estos son algunas características que se podrían manejar con el BeagleBone Black el cual mostraría a través del programa todas estas informaciones y generaría reportes para que el usuario administrador pueda manejar este tipo de información.

BIBLIOGRAFICAS

Libros e informes:

1) Bayon. R (1978). La protección contra incendios en la construcción, primera edición.

2) Bolton. W (s.f.) Mecatrónica, sistemas de control electrónico en ingeniería mecánica y eléctrica, segunda edición.

3) Coley, G. (Abril 11, 2013). BeagleBone Black System Reference Manual. USA: Revisión A5.2.

4) Edeeste. (2013). MANUAL DE CONSTRUCCION E INSTALACION DE PANELES BLINDADOS PARA MEDIDORES DE ENERGIA . Santo Domingo, República Dominicana.

5) Heath, Steve (2003). Embedded systems design. EDN series for design engineers (en inglés) (2 edición). Newnes. p. 2.

6) Lavi, J, Kudish, J. (2005). Systems modelling & requirements specification using ECSAM: an analysis method or embedded & computer-based systems". Innovations Syst Softw Eng. 1: 100–115.

7) MARWEDEL, P. (2003). Embedded system design. Kluwer Academic Publishers University of Dortmund. Alemania. 2003.

8) Michael Barr. «Embedded Systems Glossary». NetrinoTechnical Library. Consultado el 21-04-2007.

9) Payas, R. (2003). Sensores y acondicionadores de señal, cuarta edición.

10) Williams, E. A. (2007). ENGINEERING HANDBOOK (10ma ed.). National Association of Broadcasters.

11) Zumbahlen, H. (2008). Linear Circuit Design Handbook, Burlington, Linacre House.

Páginas Web:

1) Alegsa. L. (s.f.). ¿Cuál es la definición de Periférico (informática)? Recuperado de http://www.alegsa.com.ar/Dic/periferico.php

2) Allen-Bradley . (2015). Rockwell Automation.

3) Alzate, D. (2010). Control y medida de nivel de líquido con señales de ultrasonido. Recuperado de: http://repositorio.utp.edu.co/dspace/bitstream/11059/1774/4/6298A478.pdf

4) Arian. (s.f). Control e Instrumentacion, Rev. B. Recuperado de: http://www.arian.cl/downloads/nt-002.pdf

5) Automation, R. (2012). Sensores fotoeléctricos PHOTOSWITCH®. Rockwell Automation, 8.

6) Cabrera, M. (10 de Noviembre de 2008). Scribd. Obtenido de http://es.scribd.com/doc/7853459/SENSORES-FOTOELECTRICOS#scribd

7) Cardenas, H. (2007), Proceso de Coalision en Partículas Elementales. Recuperado de: http://virtual.uptc.edu.co/drupal/files/84.pdf

8) Cardozo, J. (2012, Marzo 29). concepto de informática: TARJETA MADRE. Recuperado de: http://jeisoncardo.blogspot.com/2012/03/tarjeta-madre.html

9) Carrasco, M. (2013). BeagleBone Black, primeros pasos | Internet de las Cosas. Recuperado de: http://www.internetdelascosas.cl/2013/05/20/beaglebone-black-primeros-pasos/

10) Castro, A. (2008), Sensores utilizados en la Automatización Industrial. Recuperado de: http://eie.ucr.ac.cr/uploads/file/proybach/pb0813t.pdf

11) Cervigón, C. (s.f.). Interfaz gráficas de usuario. Recuperado de: http://www.fdi.ucm.es/profesor/jpavon/poo/tema6resumido.pdf

12) ELECTRY PERRY. (s.f). Direct Industry. Obtenido de http://www.directindustry.es/prod/perry-electric/product-65713-500850.html

13) Felici, S. (s.f.). Sistemas operativos. En FUNDAMENTOS DE TELEMATICA. Recuperado de: http://informatica.uv.es/it3guia/FT/cap5-ssoo-ft.pdf

14) Gómez, J. D. (28 de mayo de 2013). Prezi. Obtenido de https://prezi.com/jy9ou6s3w7fi/untitled-prezi/

15) González Palacio, L., & Urrego Giraldo, G. (2008, Julio 13). MODELO DE REQUISITOS PARA SISTEMAS EMBEBIDOS. Revista Ingenierías Universidad Medellín, 7(13), 111-127. Recuperado de: http://www.redalyc.org/articulo.oa?id=75071308

16) Herrera. H. (30 marzo, 2012). Estructura de una Central de Monitoreo de Alarmas; topologías y tendencias. Recuperado de: www.tecnoseguro.com

17) IdosE. (s.f.). Sistemas Embebidos. Recuperado de: http://www.idose.es/sistemas-embebidos

18) Lezama, A (2001). Modelado de dispositivos para un sistema de seguridad. Recuperado de: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/lezama_l_a/capitulo1.pdf

19) Lugo, L. (2001). catarina.udlap. Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/lezama_l_a/capitulo1.pdf

20) Mañas Carbonell, M. (2013, Enero 10). Interfaz gráfica de usuario (GUI) | Fundéu BBVA. Recuperado de: http://www.fundeu.es/escribireninternet/interfaz-grafica-de-usuario-gui/

21) Moermond, J (2010). Pneumatic Cylinder Sensor. Recuperado de: http://www.designworldonline.com/choose-your-best-pneumatic-cylinder-sensor-here/

22) Moreno, E. (2006). Sensor de movimiento por ultrasonido. Recuperado de: http://proton.ucting.udg.mx/materias/ET201/modulo_11/2006B/sensor_movimiento_ultrasonido.pdf

23) Oki Sensor Device. (1996). Cika ELECTRONICA. Obtenido de http://www.cika.com/productos/oki/

24) Pérez Hernández, G. (2007). Interfaz de usuario. Recuperado de: http://www.escet.urjc.es/~gperez/Interfaz/practicas/InterfazU1.pdf

25) Raspberry para torpes. (s.f.). Recuperado de: http://raspberryparatorpes.net/tag/beaglebone-black.html

26) Reed Switch info. (s.f.). Recuperado de: http://reed-switch-info.com/index.html

27) Rico, L., & Rodriguez, G. (s.f.). Evolución de la microcomputadora. Recuperado desde http://computacionequipo16.wikispaces.com/

28) Rockwell automation (2012). Sensores fotoeléctricos. Recuperado de: http://literature.rockwellautomation.com/idc/groups/literature/documents/br/photo-br001_-es-p.pdf

29) Rojas, A. A. (Julio de 2008). Escuela de Ingeniería Eléctrica Universidad de Costa Rica. Obtenido de http://eie.ucr.ac.cr/uploads/file/proybach/pb0813t.pdf

30) Ruiz, J., Martin, J. M., & Lladó, R. (s.f.) Termómetros de Resistencia. Recuperado de: https://www.epsem.upc.edu/escola/departaments/fisica-aplicada/documents/fona_fisic_eng2/robleff2electro/termistor.pdf

31) SENSINC. (28 de Julio de 2015). Sensores-de-medida. Obtenido de http://www.sensores-de-medida.es/sensing_sl/MEDIDA-DE-VIBRACIONES_151/Sensores-de-vibraci%C3%B3n_290/

32) SENSINC. (28 de JULIO de 2015). Sensores-de-medida. Obtenido de http://www.sensores-de-medida.es/sensing_sl/SENSORES-Y-TRANSDUCTORES_35/Aceler%C3%B3metros---Sensores-de-aceleraci%C3%B3n_49/

33) System Sensor. (2004). System Sensor. Obtenido de http://www.systemsensor.ca/es/docs/guides/A05-1046.pdf

34) Turmo, E (1988). Detectores de humo. Recuperado de: http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/201a300/ntp_215.pdf

35) UNET. (s.f). UNET. Obtenido de http://www.unet.edu.ve/~ielectro/sensores.pdf

36) Unidad Central de Proceso. (s.f). Recuperado de: http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_terminados/PolilibroFC/Unidad_II/Unidad%20II_2.htm

37) Venkat, K. (2010), Glass breakage detector uses one microcontroller. Recuperado de: http://www.edn.com/design/systems-design/4363638/Glass-breakage-detector-uses-one-microcontroller

APÉNDICE

Apéndice A: Código del BeagleBone Black

```
var core = require('bonescript');
var server = require('http').createServer(handler);
var fs = require('fs');
var currentPin = "";
var lastChar = null;
var pressedCounter = 0;
var isNewEvent = false;
var validating = false;
var doorOpen = false;
var canEvent = true;
var doorOpenLastState = null;
var doorOpenStateChanged = null;
var pinIsValid = false;
var canInvalidatePin = false;

core.pinMode('P8_12', core.INPUT);
core.pinMode('P8_14', core.INPUT);
core.pinMode('P8_16', core.INPUT);
core.pinMode('P8_18', core.INPUT);
core.pinMode('P8_11', core.INPUT);
core.pinMode('P8_15', core.INPUT);
core.pinMode('P8_17', core.INPUT);
core.pinMode('P8_26', core.INPUT);
core.pinMode('P9_17', core.INPUT);
core.pinMode('P9_23', core.INPUT);

core.pinMode('P9_27', core.INPUT);

core.pinMode("P9_26", core.OUTPUT, 7, 'pulldown', 'fast');
core.pinMode("P9_24", core.OUTPUT, 7, 'pulldown', 'fast');
//core.pinMode("P9_25", core.OUTPUT, 7, 'pulldown', 'fast');
core.pinMode("P9_30", core.OUTPUT, 7, 'pulldown', 'fast');

core.digitalWrite('P9_26', core.LOW);
core.digitalWrite('P9_24', core.LOW);
```

```
//core.digitalWrite('P9_25', core.LOW);
core.digitalWrite('P9_30', core.LOW);

setInterval(check, 15);

function GenerateAlarm() {
    if (canEvent && doorOpen && doorOpenStateChanged) {
        canEvent = false;
        doorOpenStateChanged = false;
        var timer = setTimeout(function() {
            canEvent = true;
            console.log("event logger active")
        }, 1000);

        AddNewEvent("Violacion de integridad fisica", "WARNING", function() {
            SetLedState(false);
        });
    }
}

function SetLedState(ok) {
    if (ok != undefined && ok != null) {
        if (ok) {
            core.digitalWrite('P9_24', core.HIGH);
            core.digitalWrite('P9_26', core.LOW);
        } else {
            core.digitalWrite('P9_26', core.HIGH);
            core.digitalWrite('P9_24', core.LOW);
        }
    } else {
        core.digitalWrite('P9_26', core.LOW);
        core.digitalWrite('P9_24', core.LOW);
    }
}


function KeyPressed(ch) {
    if (doorOpen) {
        return;
    }
    if (pressedCounter > 10) {
        pressedCounter = 0;
        lastChar = "-";
    }
    if (lastChar != ch) {
```

```
        lastChar = ch;
        if(ch == null) {
            if (validating == false) {
                validating = true;
                GetCurrentPin(function (pin) {
                    if (pin == "" || pin != currentPin){
                        AddNewEvent("Pin incorrecto, acceso denegado.", "ERROR", function() {
                            validating = false;
                                                                    pinIsValid = false;
                            SetLedState(false);
                        });
                    } else {
                        pinIsValid = true;
                        AddNewEvent("Pin válido", "OK", function() {
                            ClearPin(function() {
                                validating = false;

                                                                    //core.digitalWrite('P9_25',
core.HIGH);
                                                                    core.digitalWrite('P9_30',
core.HIGH);
                                                                    setTimeout(function() {

            //core.digitalWrite('P9_25', core.LOW);

            core.digitalWrite('P9_30', core.LOW);
                                                                    }, 5000);

                                SetLedState(true);
                            });
                        });
                    }
                });
            }
        } else {
            pinIsValid = false;
            SetLedState(null);
            currentPin += ch;
        }
    }
    if (currentPin.length > 4) {
        currentPin = currentPin.substring(1);
    }
}
```

```
function check(){
    pressedCounter++;

    core.digitalRead("P8_12", function(x){
        if (x.value == 1) {
            KeyPressed("1");
        }
    });
    core.digitalRead("P8_14", function(x){
        if (x.value == 1) {
            KeyPressed("2");
        }
    });
    core.digitalRead("P8_16", function(x){
        if (x.value == 1) {
            KeyPressed("3");
        }
    });
    core.digitalRead("P8_18", function(x){
        if (x.value == 1) {
            KeyPressed("4");
        }
    });
    core.digitalRead("P8_11", function(x){
        if (x.value == 1) {
            KeyPressed("5");
        }
    });
    core.digitalRead("P8_15", function(x){
        if (x.value == 1) {
            KeyPressed("6");
        }
    });
    core.digitalRead("P8_17", function(x){
        if (x.value == 1) {
            KeyPressed("7");
        }
    });
    core.digitalRead("P8_26", function(x){
        if (x.value == 1) {
            KeyPressed("8");
        }
    });
    /*core.digitalRead("P9_17", function(x){
```

```
        if (x.value == 1) {
            console.log("9")
        }
    });*/
    core.digitalRead("P9_17", function(x){
        if (x.value == 1) {
            KeyPressed("0");
        }
    });
    core.digitalRead("P9_23", function(x){
        if (x.value == 1) {
            KeyPressed(null);
        }
    });
    /*core.digitalRead("P9_25", function(x){
        if (x.value == 0) {
            GenerateAlarm();
        }
    });*/
    core.digitalRead("P9_27", function(x){
        doorOpen = x.value == 0;
        if (doorOpenLastState != doorOpen) {
            doorOpenStateChanged = true;
            console.log("state changed")
        }
        doorOpenLastState = doorOpen;

                if (doorOpenStateChanged) {
                        if (doorOpen && pinIsValid == false) {
                                GenerateAlarm();
                        }
                        if (pinIsValid && doorOpen) {
                                canInvalidatePin = true;
                        }
                        if (pinIsValid && doorOpen == false && canInvalidatePin) {
                            pinIsValid = false;
                            canInvalidatePin = false;
                        }
                }
    });
};

function AddNewEvent(title, remarks, complete) {
    console.log(title);
    GetEventsFile(function(data) {
```

```
        var events = JSON.parse(data);
        events.push({
            Date: new Date(),
            Title: title,
            Remarks: remarks != null ? remarks : ""
        });
        core.writeTextFile("Resources/Data/Events.json", JSON.stringify(events), function() {
            complete();
            isNewEvent = true;
        });
    });
}

function GetCurrentPin(complete) {
    core.readTextFile("Resources/Data/CurrentPin.json", function(x) {
        if (complete != undefined) {
            complete(x.data.trim());
        }
    });
}

function GetEventsFile(complete) {
    core.readTextFile("Resources/Data/Events.json", function(x) {
        if (complete != undefined) {
            complete(x.data);
        }
    });
}

function SetPin(parameters, complete) {
    core.writeTextFile("Resources/Data/CurrentPin.json", parameters.pin, function() {
        complete(true);
    });
};

function ClearEvents() {
    core.writeTextFile("Resources/Data/Events.json", JSON.stringify([]), function() {
    });
}

function ClearPin(complete) {
    core.writeTextFile("Resources/Data/CurrentPin.json", "", function() {
        complete();
    });
}
```

```
function GetInfo(parameters, complete) {
    core.getPlatform(function(x) {
        var beagle = {
            system: "RLM - v1.0.5",
            name: x.name,
            version: x.version,
            serial: x.serialNumber
        };

        complete(beagle);
    });
};

function handler(req, res) {
    if (req.method == 'POST') {
        var input = "";
        req.on('data', function (data) {
            input += data;
            if (input.length > 1e6) {
                req.connection.destroy();
            }
        });

        req.on('end', function () {
            var json = JSON.parse(input);

            if (req.url == "/Command") {
                res.writeHead(200, {'Content-Type': 'application/json'});
                var actionName = json.action;

                switch(actionName) {
                    case "ClearEvents":
                        ClearEvents();
                        return res.end(JSON.stringify(true));
                    break;
                    case "IsNewEvent":
                        var result = isNewEvent;
                        isNewEvent = false;
                        return res.end(JSON.stringify(result));
                    break;
                    case "GetEvents":
                        GetEventsFile(function(result) {
                            return res.end(result);
                        });
```

```
                break;
                case "GetInfo":
                    GetInfo(json.parameters, function(result) {
                        return res.end(JSON.stringify(result));
                    });
                break;
                case "SetPin":
                    SetPin(json.parameters, function(result) {
                        return res.end(JSON.stringify(result));
                    });
                break;
                default:
                    res.writeHead(500);
                    return res.end('Remote Locker Manager - Invalid Command');
                break;
            }
        }
    });
} else {
    if (req.url == "/Serial") {
        core.getPlatform(function(x) {
            res.writeHead(200, {'Content-Type': 'application/json'});

            var beagle = {
                system: "RLM - v1.0.5",
                name: x.name,
                version: x.version,
                serial: x.serialNumber
            };

            return res.end(JSON.stringify(beagle));
        });
    } else {
        res.writeHead(500);
        return res.end('Remote Locker Manager - Invalid Request');
    }
}
}

server.listen(8888);
```

Apéndice B: Código del Entorno Principal de la aplicación

```
using RLM.Middleware;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;


namespace RLM {

public partial class EventsView : Form {


private Locker _locker;

private Timer _timer;

private bool _fetchingData = false;

private bool _refreshData;
```

```
public EventsView(Locker locker) {
InitializeComponent();

        _locker = locker;
        _timer = new Timer();
        _timer.Tick += EventChange;
        _timer.Interval = 150;
        _timer.Enabled = true;
      }

public void EventChange(object sender, EventArgs e) {
if (_fetchingData == false) {
            _fetchingData = true;
            _locker.IsNewEvent((isNewEvent) => {
if (isNewEvent) {
                _refreshData = true;
              }
              _fetchingData = false;
            });
        }

if (_refreshData) {
RefreshData();
            _refreshData = false;
```

```
            }
        }

public void RefreshData() {
            _locker.GetEvents((result) => {
if (result != null) {
BeginInvoke(new Action(() => {
                    grdEvents.DataSource = result.OrderByDescending(d => d.Date).ToList();
}));
                }
            });
        }
private void btnClose_Click(object sender, EventArgs e) {
            _timer.Enabled = false;
Close();
        }

private void EventsView_Load(object sender, EventArgs e) {
            grdEvents.AutoGenerateColumns = false;
RefreshData();
        }

private void btnClear_Click(object sender, EventArgs e) {
            btnClear.Enabled = false;
```

```
            _locker.ClearEvents((result) => {
BeginInvoke(new Action(() => {
                    btnClear.Enabled = true;
}));
if (result) {
                    _refreshData = true;
                }
            });


        }
    }
}
using Newtonsoft.Json;
using RLM.Middleware;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Dynamic;
using System.IO;
using System.Linq;
using System.Net;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace RLM {
public partial class MainForm : Form {
private List<Locker> _lockers;
private Locker _selectedLocker;
private Timer _timer;
private bool _isNewEventFetching = false;
public MainForm() {
InitializeComponent();
        _lockers = new List<Locker>();
        _timer = new Timer();
        _timer.Tick += EventChange;
        _timer.Interval = 100;
        _timer.Enabled = true;
      }

private void Main_Load(object sender, EventArgs e) {
        dgLockers.AutoGenerateColumns = false;

RefreshLockersView();
```

```
if (dgLockers.SelectedRows.Count > 0) {
SelectLocker((Locker)dgLockers.SelectedRows[0].DataBoundItem);
        }
    }

public void EventChange(object sender, EventArgs e) {
if (_isNewEventFetching == false && _selectedLocker != null &&
_selectedLocker.Connectabla) {
            _isNewEventFetching = true;
            _selectedLocker.IsNewEvent((isNewEvent) => {
if (isNewEvent) {
var events = new EventsView(_selectedLocker);
                _timer.Enabled = false;
events.ShowDialog();
                _timer.Enabled = true;
              }
              _isNewEventFetching = false;
            });
        }
    }

private void SelectLocker() {
if (dgLockers.SelectedRows.Count > 0) {
SelectLocker((Locker)dgLockers.SelectedRows[0].DataBoundItem);
        }
```

```
        }

private void SelectLocker(Middleware.Locker locker) {
if (locker != null) {
            _selectedLocker = locker;
            txtLockerName.Text = locker.Name;
            txtDeviceUrl.Text = locker.URL;
            txtDeviceType.Text = locker.Type;
            txtLocation.Text = locker.Location;
            txtArea.Text = locker.Area;
            txtCode.Text = locker.Code;
        } else {
            _selectedLocker = null;
            txtLockerName.Text = "";
            txtDeviceUrl.Text = "";
            txtDeviceType.Text = "";
            txtLocation.Text = "";
            txtArea.Text = "";
            txtCode.Text = "";
        }
        txtDeviceState.Text = "";
        txtSerialNumber.Text = "";
        txtInternalVersion.Text = "";
    }
```

```
private void RefreshLockersView() {
            _lockers = Global.GetLockers();
            dgLockers.DataSource = _lockers;
SelectLocker();
        }

private void btnEvents_Click(object sender, EventArgs e) {
if (_selectedLocker != null && _selectedLocker.Connectabla) {
var events = new EventsView(_selectedLocker);
                _timer.Enabled = false;
events.ShowDialog();
                _timer.Enabled = true;
            } else {
MessageBox.Show("Dispositivo desconectado.", "RLM", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
            }
        }

private void btnExit_Click(object sender, EventArgs e) {
Close();
        }

private void btnAdd_Click(object sender, EventArgs e) {
var dialog = new LockerEditor();
```

```
dialog.ShowDialog();

if (dialog.Data != null) {
            _lockers.Add(dialog.Data);
SaveLockers();
        }
    }

private void SaveLockers() {
Global.SetLockers(_lockers);
RefreshLockersView();
    }

private void btnDelete_Click(object sender, EventArgs e) {
if (dgLockers.SelectedRows.Count > 0) {

var response = MessageBox.Show("Está seguro que desea eliminar este item?", "RLM",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (response == System.Windows.Forms.DialogResult.Yes) {
                _lockers.Remove((Locker)dgLockers.SelectedRows[0].DataBoundItem);
SaveLockers();
            }
        }
    }
```

```
private void btnEdit_Click(object sender, EventArgs e) {
EditSelectedLocker();
        }

private void EditSelectedLocker() {
if (dgLockers.SelectedRows.Count > 0) {
var index = dgLockers.SelectedRows[0].Index;

var dialog = new LockerEditor(_lockers[index]);
dialog.ShowDialog();

if (dialog.Data != null) {
_lockers[index] = dialog.Data;
SaveLockers();
                }
            }
        }

private void dgLockers_RowEnter(object sender, DataGridViewCellEventArgs e) {
if (dgLockers.SelectedRows.Count > 0) {
SelectLocker((Locker)dgLockers.SelectedRows[0].DataBoundItem);
            }
        }
```

```
#region Keynboard Events
private void btn1_Click(object sender, EventArgs e) {
AddNumber(1);
}

private void btn2_Click(object sender, EventArgs e) {
AddNumber(2);
}

private void btn3_Click(object sender, EventArgs e) {
AddNumber(3);
}

private void btn4_Click(object sender, EventArgs e) {
AddNumber(4);
}

private void btn5_Click(object sender, EventArgs e) {
AddNumber(5);
}

private void btn6_Click(object sender, EventArgs e) {
AddNumber(6);
```

```
        }

private void btn7_Click(object sender, EventArgs e) {
AddNumber(7);
        }

private void btn8_Click(object sender, EventArgs e) {
AddNumber(8);
        }

private void btn9_Click(object sender, EventArgs e) {
AddNumber(9);
        }

private void btn0_Click(object sender, EventArgs e) {
AddNumber(0);
        }
        #endregion

private void AddNumber(int number) {
if (txtPin.Text.Length == 4) {
            txtPin.Text = txtPin.Text.Substring(1);
          }
          txtPin.Text = txtPin.Text + number.ToString();
```

```
        }

private void btnReset_Click(object sender, EventArgs e) {
            txtPin.Text = "0000";
        }

private void btnAuto_Click(object sender, EventArgs e) {
var rnd = new Random();
var value = rnd.Next(0, 9999);

if (value < 10) {
                txtPin.Text = "000" + value.ToString();
            } else if (value < 100) {
                txtPin.Text = "00" + value.ToString();
            } else if (value < 1000) {
                txtPin.Text = "0" + value.ToString();
            } else {
                txtPin.Text = value.ToString();
            }
        }

private void btnAuthorize_Click(object sender, EventArgs e) {
if (_selectedLocker != null && txtSerialNumber.Text.Length > 0 && txtPin.Text.Length
== 4 && txtPin.Text != "0000") {
if (_selectedLocker.SendPin(txtPin.Text)) {
```

```
MessageBox.Show("Autorización con el pin " + txtPin.Text + " enviada correctamente.",
"RLM", MessageBoxButtons.OK, MessageBoxIcon.Information);

            } else {

MessageBox.Show("Error enviando autorización.", "RLM", MessageBoxButtons.OK,
MessageBoxIcon.Error);

            }

        }

    }


private void btnConnect_Click(object sender, EventArgs e) {

        btnConnect.Enabled = false;

if (_selectedLocker != null) {

var deviceData = _selectedLocker.ConnectDevice();

            btnConnect.Enabled = true;

if (deviceData != null) {

MessageBox.Show("Dispositivo conectado.", "RLM", MessageBoxButtons.OK,
MessageBoxIcon.Information);

                txtDeviceState.Text = "Conectado - " + deviceData.name;

                txtSerialNumber.Text = (string)deviceData.serial;

                txtInternalVersion.Text = (string)deviceData.version;

            } else {

                txtDeviceState.Text = "Desconectado";

                txtSerialNumber.Text = "";

                txtInternalVersion.Text = "";

MessageBox.Show("No se pudo conectar el dispositivo.", "RLM",
MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
            }
        }
    }

private void dgLockers_CellMouseDoubleClick(object sender,
DataGridViewCellMouseEventArgs e) {
EditSelectedLocker();
    }

  }
}
```

Apéndice C: Código del Estilo de letras y figura del Interfaz gráfica

```
namespace RLM {
partial class MainForm {
    /// <summary>
    /// Required designer variable.
    /// </summary>
private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
```

```
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
protected override void Dispose(bool disposing) {
if (disposing && (components != null)) {
components.Dispose();
            }
base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
private void InitializeComponent() {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(MainForm));
            this.dgLockers = new System.Windows.Forms.DataGridView();
            this.grpLockers = new System.Windows.Forms.GroupBox();
            this.btnConnect = new System.Windows.Forms.Button();
            this.btnEdit = new System.Windows.Forms.Button();
            this.btnDelete = new System.Windows.Forms.Button();
            this.btnAdd = new System.Windows.Forms.Button();
```

```
this.groupBox1 = new System.Windows.Forms.GroupBox();
this.txtLocation = new System.Windows.Forms.TextBox();
this.label7 = new System.Windows.Forms.Label();
this.txtLockerName = new System.Windows.Forms.TextBox();
this.txtCode = new System.Windows.Forms.TextBox();
this.label5 = new System.Windows.Forms.Label();
this.label3 = new System.Windows.Forms.Label();
this.txtDeviceUrl = new System.Windows.Forms.TextBox();
this.txtArea = new System.Windows.Forms.TextBox();
this.label4 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.txtDeviceType = new System.Windows.Forms.TextBox();
this.label1 = new System.Windows.Forms.Label();
this.groupBox2 = new System.Windows.Forms.GroupBox();
this.btnAuto = new System.Windows.Forms.Button();
this.btnAuthorize = new System.Windows.Forms.Button();
this.btn9 = new System.Windows.Forms.Button();
this.btn6 = new System.Windows.Forms.Button();
this.btn3 = new System.Windows.Forms.Button();
this.btn0 = new System.Windows.Forms.Button();
this.btnReset = new System.Windows.Forms.Button();
this.btn8 = new System.Windows.Forms.Button();
this.btn7 = new System.Windows.Forms.Button();
this.txtPin = new System.Windows.Forms.TextBox();
```

```
this.btn5 = new System.Windows.Forms.Button();
this.btn4 = new System.Windows.Forms.Button();
this.btn2 = new System.Windows.Forms.Button();
this.btn1 = new System.Windows.Forms.Button();
this.btnExit = new System.Windows.Forms.Button();
this.btnEvents = new System.Windows.Forms.Button();
this.groupBox3 = new System.Windows.Forms.GroupBox();
this.txtDeviceState = new System.Windows.Forms.TextBox();
this.label6 = new System.Windows.Forms.Label();
this.txtSerialNumber = new System.Windows.Forms.TextBox();
this.label8 = new System.Windows.Forms.Label();
this.txtInternalVersion = new System.Windows.Forms.TextBox();
this.label9 = new System.Windows.Forms.Label();
this.Code = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.Locker = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.Server = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.Device = new System.Windows.Forms.DataGridViewTextBoxColumn();
((System.ComponentModel.ISupportInitialize)(this.dgLockers)).BeginInit();
this.grpLockers.SuspendLayout();
this.groupBox1.SuspendLayout();
this.groupBox2.SuspendLayout();
this.groupBox3.SuspendLayout();
this.SuspendLayout();
//
```

```
// dgLockers
//
this.dgLockers.AllowUserToAddRows = false;
this.dgLockers.AllowUserToDeleteRows = false;
this.dgLockers.AllowUserToResizeRows = false;
this.dgLockers.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.dgLockers.BorderStyle = System.Windows.Forms.BorderStyle.None;
this.dgLockers.CellBorderStyle =
System.Windows.Forms.DataGridViewCellBorderStyle.None;
this.dgLockers.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgLockers.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[]
{
this.Code,
this.Locker,
this.Server,
this.Device});
this.dgLockers.EditMode =
System.Windows.Forms.DataGridViewEditMode.EditProgrammatically;
this.dgLockers.Location = new System.Drawing.Point(9, 19);
this.dgLockers.MultiSelect = false;
this.dgLockers.Name = "dgLockers";
```

```
this.dgLockers.ReadOnly = true;

this.dgLockers.RowHeadersBorderStyle =
System.Windows.Forms.DataGridViewHeaderBorderStyle.None;

this.dgLockers.RowHeadersVisible = false;

this.dgLockers.RowHeadersWidthSizeMode =
System.Windows.Forms.DataGridViewRowHeadersWidthSizeMode.DisableResizing;

this.dgLockers.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

this.dgLockers.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;

this.dgLockers.ShowCellErrors = false;

this.dgLockers.ShowCellToolTips = false;

this.dgLockers.ShowEditingIcon = false;

this.dgLockers.Size = new System.Drawing.Size(655, 239);

this.dgLockers.TabIndex = 0;

this.dgLockers.CellMouseDoubleClick += new
System.Windows.Forms.DataGridViewCellMouseEventHandler(this.dgLockers_CellMous
eDoubleClick);

this.dgLockers.RowEnter += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dgLockers_RowEnter);

//

// grpLockers

//

this.grpLockers.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));
```

```
this.grpLockers.Controls.Add(this.btnConnect);
this.grpLockers.Controls.Add(this.btnEdit);
this.grpLockers.Controls.Add(this.btnDelete);
this.grpLockers.Controls.Add(this.btnAdd);
this.grpLockers.Controls.Add(this.dgLockers);
        this.grpLockers.Location = new System.Drawing.Point(12, 12);
        this.grpLockers.Name = "grpLockers";
        this.grpLockers.Size = new System.Drawing.Size(673, 297);
        this.grpLockers.TabIndex = 1;
        this.grpLockers.TabStop = false;
        this.grpLockers.Text = "Casilleros Registrados";
        //
        // btnConnect
        //
        this.btnConnect.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
        this.btnConnect.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.btnConnect.Location = new System.Drawing.Point(316, 264);
        this.btnConnect.Name = "btnConnect";
        this.btnConnect.Size = new System.Drawing.Size(82, 27);
        this.btnConnect.TabIndex = 1;
        this.btnConnect.Text = "Conectar";
        this.btnConnect.UseVisualStyleBackColor = true;
```

```
this.btnConnect.Click += new System.EventHandler(this.btnConnect_Click);

//

// btnEdit

//

this.btnEdit.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.btnEdit.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.btnEdit.Location = new System.Drawing.Point(405, 264);

this.btnEdit.Name = "btnEdit";

this.btnEdit.Size = new System.Drawing.Size(82, 27);

this.btnEdit.TabIndex = 1;

this.btnEdit.Text = "Modificar";

this.btnEdit.UseVisualStyleBackColor = true;

this.btnEdit.Click += new System.EventHandler(this.btnEdit_Click);

//

// btnDelete

//

this.btnDelete.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.btnDelete.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.btnDelete.Location = new System.Drawing.Point(494, 264);

this.btnDelete.Name = "btnDelete";
```

```
this.btnDelete.Size = new System.Drawing.Size(82, 27);

this.btnDelete.TabIndex = 1;

this.btnDelete.Text = "Eliminar";

this.btnDelete.UseVisualStyleBackColor = true;

this.btnDelete.Click += new System.EventHandler(this.btnDelete_Click);

//

// btnAdd

//

this.btnAdd.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.btnAdd.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.btnAdd.Location = new System.Drawing.Point(583, 264);

this.btnAdd.Name = "btnAdd";

this.btnAdd.Size = new System.Drawing.Size(82, 27);

this.btnAdd.TabIndex = 1;

this.btnAdd.Text = "Agregar";

this.btnAdd.UseVisualStyleBackColor = true;

this.btnAdd.Click += new System.EventHandler(this.btnAdd_Click);

//

// groupBox1

//

this.groupBox1.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)
```

```
            | System.Windows.Forms.AnchorStyles.Right)));
this.groupBox1.Controls.Add(this.txtLocation);
this.groupBox1.Controls.Add(this.label7);
this.groupBox1.Controls.Add(this.txtLockerName);
this.groupBox1.Controls.Add(this.txtCode);
this.groupBox1.Controls.Add(this.label5);
this.groupBox1.Controls.Add(this.label3);
this.groupBox1.Controls.Add(this.txtDeviceUrl);
this.groupBox1.Controls.Add(this.txtArea);
this.groupBox1.Controls.Add(this.label4);
this.groupBox1.Controls.Add(this.label2);
this.groupBox1.Controls.Add(this.txtDeviceType);
this.groupBox1.Controls.Add(this.label1);
            this.groupBox1.Location = new System.Drawing.Point(12, 315);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(524, 103);
            this.groupBox1.TabIndex = 2;
            this.groupBox1.TabStop = false;
this.groupBox1.Text = "Casillero Seleccionado";
            //
            // txtLocation
            //
            this.txtLocation.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
```

```
this.txtLocation.BackColor = System.Drawing.SystemColors.InactiveBorder;
        this.txtLocation.Enabled = false;
        this.txtLocation.Location = new System.Drawing.Point(380, 19);
        this.txtLocation.Name = "txtLocation";
        this.txtLocation.Size = new System.Drawing.Size(138, 20);
        this.txtLocation.TabIndex = 1;
        //
        // label7
        //
        this.label7.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
        this.label7.AutoSize = true;
        this.label7.Location = new System.Drawing.Point(277, 21);
        this.label7.Name = "label7";
        this.label7.Size = new System.Drawing.Size(53, 13);
        this.label7.TabIndex = 0;
        this.label7.Text = "Localidad";
        //
        // txtLockerName
        //
        this.txtLockerName.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
        this.txtLockerName.BackColor = System.Drawing.SystemColors.InactiveBorder;
```

```
this.txtLockerName.Enabled = false;
this.txtLockerName.Location = new System.Drawing.Point(109, 19);
this.txtLockerName.Name = "txtLockerName";
this.txtLockerName.Size = new System.Drawing.Size(154, 20);
this.txtLockerName.TabIndex = 1;
//
// txtCode
//
this.txtCode.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
this.txtCode.BackColor = System.Drawing.SystemColors.InactiveBorder;
this.txtCode.Enabled = false;
this.txtCode.Location = new System.Drawing.Point(380, 73);
this.txtCode.Name = "txtCode";
this.txtCode.Size = new System.Drawing.Size(138, 20);
this.txtCode.TabIndex = 1;
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(6, 21);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(59, 13);
this.label5.TabIndex = 0;
```

```
this.label5.Text = "Referencia";
//
// label3
//
this.label3.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(277, 75);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(73, 13);
this.label3.TabIndex = 0;
this.label3.Text = "Número Finca";
//
// txtDeviceUrl
//
this.txtDeviceUrl.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.txtDeviceUrl.BackColor = System.Drawing.SystemColors.InactiveBorder;
this.txtDeviceUrl.Enabled = false;
this.txtDeviceUrl.Location = new System.Drawing.Point(109, 73);
this.txtDeviceUrl.Name = "txtDeviceUrl";
this.txtDeviceUrl.Size = new System.Drawing.Size(154, 20);
this.txtDeviceUrl.TabIndex = 1;
```

```
//
// txtArea
//
this.txtArea.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
this.txtArea.BackColor = System.Drawing.SystemColors.InactiveBorder;
this.txtArea.Enabled = false;
this.txtArea.Location = new System.Drawing.Point(380, 45);
this.txtArea.Name = "txtArea";
this.txtArea.Size = new System.Drawing.Size(138, 20);
this.txtArea.TabIndex = 1;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(6, 75);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(70, 13);
this.label4.TabIndex = 0;
this.label4.Text = "URL Servicio";
//
// label2
//
```

```
this.label2.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(277, 47);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(38, 13);

this.label2.TabIndex = 0;

this.label2.Text = "Sector";

//

// txtDeviceType

//

this.txtDeviceType.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.txtDeviceType.BackColor = System.Drawing.SystemColors.InactiveBorder;

this.txtDeviceType.Enabled = false;

this.txtDeviceType.Location = new System.Drawing.Point(109, 45);

this.txtDeviceType.Name = "txtDeviceType";

this.txtDeviceType.Size = new System.Drawing.Size(154, 20);

this.txtDeviceType.TabIndex = 1;

//

// label1

//

this.label1.AutoSize = true;
```

```
this.label1.Location = new System.Drawing.Point(6, 47);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(97, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Tipo de Dispositivo";
//
// groupBox2
//
this.groupBox2.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Right)));
this.groupBox2.Controls.Add(this.btnAuto);
this.groupBox2.Controls.Add(this.btnAuthorize);
this.groupBox2.Controls.Add(this.btn9);
this.groupBox2.Controls.Add(this.btn6);
this.groupBox2.Controls.Add(this.btn3);
this.groupBox2.Controls.Add(this.btn0);
this.groupBox2.Controls.Add(this.btnReset);
this.groupBox2.Controls.Add(this.btn8);
this.groupBox2.Controls.Add(this.btn7);
this.groupBox2.Controls.Add(this.txtPin);
this.groupBox2.Controls.Add(this.btn5);
this.groupBox2.Controls.Add(this.btn4);
this.groupBox2.Controls.Add(this.btn2);
```

```
this.groupBox2.Controls.Add(this.btn1);
        this.groupBox2.Location = new System.Drawing.Point(700, 12);
        this.groupBox2.Name = "groupBox2";
        this.groupBox2.Size = new System.Drawing.Size(150, 297);
        this.groupBox2.TabIndex = 3;
        this.groupBox2.TabStop = false;
        this.groupBox2.Text = "Acceso Remoto";
        //
        // btnAuto
        //
        this.btnAuto.Location = new System.Drawing.Point(101, 159);
        this.btnAuto.Name = "btnAuto";
        this.btnAuto.Size = new System.Drawing.Size(40, 40);
        this.btnAuto.TabIndex = 0;
        this.btnAuto.Text = "auto";
        this.btnAuto.UseVisualStyleBackColor = true;
        this.btnAuto.Click += new System.EventHandler(this.btnAuto_Click);
        //
        // btnAuthorize
        //
        this.btnAuthorize.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
```

```
this.btnAuthorize.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.btnAuthorize.Location = new System.Drawing.Point(9, 264);

this.btnAuthorize.Name = "btnAuthorize";

this.btnAuthorize.Size = new System.Drawing.Size(132, 27);

this.btnAuthorize.TabIndex = 1;

this.btnAuthorize.Text = "Autorizar";

this.btnAuthorize.UseVisualStyleBackColor = true;

this.btnAuthorize.Click += new System.EventHandler(this.btnAuthorize_Click);

//

// btn9

//

this.btn9.Location = new System.Drawing.Point(101, 113);

this.btn9.Name = "btn9";

this.btn9.Size = new System.Drawing.Size(40, 40);

this.btn9.TabIndex = 0;

this.btn9.Text = "9";

this.btn9.UseVisualStyleBackColor = true;

this.btn9.Click += new System.EventHandler(this.btn9_Click);

//

// btn6

//

this.btn6.Location = new System.Drawing.Point(101, 67);

this.btn6.Name = "btn6";

this.btn6.Size = new System.Drawing.Size(40, 40);
```

```
this.btn6.TabIndex = 0;
this.btn6.Text = "6";
this.btn6.UseVisualStyleBackColor = true;
this.btn6.Click += new System.EventHandler(this.btn6_Click);
//
// btn3
//
this.btn3.Location = new System.Drawing.Point(101, 21);
this.btn3.Name = "btn3";
this.btn3.Size = new System.Drawing.Size(40, 40);
this.btn3.TabIndex = 0;
this.btn3.Text = "3";
this.btn3.UseVisualStyleBackColor = true;
this.btn3.Click += new System.EventHandler(this.btn3_Click);
//
// btn0
//
this.btn0.Location = new System.Drawing.Point(55, 159);
this.btn0.Name = "btn0";
this.btn0.Size = new System.Drawing.Size(40, 40);
this.btn0.TabIndex = 0;
this.btn0.Text = "0";
this.btn0.UseVisualStyleBackColor = true;
this.btn0.Click += new System.EventHandler(this.btn0_Click);
```

```
//
// btnReset
//
this.btnReset.Location = new System.Drawing.Point(9, 159);
this.btnReset.Name = "btnReset";
this.btnReset.Size = new System.Drawing.Size(40, 40);
this.btnReset.TabIndex = 0;
this.btnReset.Text = "reset";
this.btnReset.UseVisualStyleBackColor = true;
this.btnReset.Click += new System.EventHandler(this.btnReset_Click);
//
// btn8
//
this.btn8.Location = new System.Drawing.Point(55, 113);
this.btn8.Name = "btn8";
this.btn8.Size = new System.Drawing.Size(40, 40);
this.btn8.TabIndex = 0;
this.btn8.Text = "8";
this.btn8.UseVisualStyleBackColor = true;
this.btn8.Click += new System.EventHandler(this.btn8_Click);
//
// btn7
//
this.btn7.Location = new System.Drawing.Point(9, 113);
```

```
this.btn7.Name = "btn7";
this.btn7.Size = new System.Drawing.Size(40, 40);
this.btn7.TabIndex = 0;
this.btn7.Text = "7";
this.btn7.UseVisualStyleBackColor = true;
this.btn7.Click += new System.EventHandler(this.btn7_Click);
//
// txtPin
//
this.txtPin.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.txtPin.BackColor = System.Drawing.SystemColors.InactiveBorder;
this.txtPin.Enabled = false;
this.txtPin.Font = new System.Drawing.Font("Verdana", 27.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtPin.ForeColor = System.Drawing.Color.Navy;
this.txtPin.Location = new System.Drawing.Point(10, 206);
this.txtPin.Name = "txtPin";
this.txtPin.Size = new System.Drawing.Size(130, 52);
this.txtPin.TabIndex = 1;
this.txtPin.Text = "0000";
this.txtPin.TextAlign = System.Windows.Forms.HorizontalAlignment.Center;
//
```

```
// btn5
//
this.btn5.Location = new System.Drawing.Point(55, 67);
this.btn5.Name = "btn5";
this.btn5.Size = new System.Drawing.Size(40, 40);
this.btn5.TabIndex = 0;
this.btn5.Text = "5";
this.btn5.UseVisualStyleBackColor = true;
this.btn5.Click += new System.EventHandler(this.btn5_Click);
//
// btn4
//
this.btn4.Location = new System.Drawing.Point(9, 67);
this.btn4.Name = "btn4";
this.btn4.Size = new System.Drawing.Size(40, 40);
this.btn4.TabIndex = 0;
this.btn4.Text = "4";
this.btn4.UseVisualStyleBackColor = true;
this.btn4.Click += new System.EventHandler(this.btn4_Click);
//
// btn2
//
this.btn2.Location = new System.Drawing.Point(55, 21);
this.btn2.Name = "btn2";
```

```
this.btn2.Size = new System.Drawing.Size(40, 40);
this.btn2.TabIndex = 0;
this.btn2.Text = "2";
this.btn2.UseVisualStyleBackColor = true;
this.btn2.Click += new System.EventHandler(this.btn2_Click);
//
// btn1
//
this.btn1.Location = new System.Drawing.Point(9, 21);
this.btn1.Name = "btn1";
this.btn1.Size = new System.Drawing.Size(40, 40);
this.btn1.TabIndex = 0;
this.btn1.Text = "1";
this.btn1.UseVisualStyleBackColor = true;
this.btn1.Click += new System.EventHandler(this.btn1_Click);
//
// btnExit
//
this.btnExit.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
this.btnExit.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.btnExit.Location = new System.Drawing.Point(769, 425);
this.btnExit.Name = "btnExit";
```

```
this.btnExit.Size = new System.Drawing.Size(82, 27);

this.btnExit.TabIndex = 1;

this.btnExit.Text = "Salir";

this.btnExit.UseVisualStyleBackColor = true;

this.btnExit.Click += new System.EventHandler(this.btnExit_Click);

//

// btnEvents

//

this.btnEvents.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));

this.btnEvents.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.btnEvents.Location = new System.Drawing.Point(680, 425);

this.btnEvents.Name = "btnEvents";

this.btnEvents.Size = new System.Drawing.Size(82, 27);

this.btnEvents.TabIndex = 1;

this.btnEvents.Text = "Eventos";

this.btnEvents.UseVisualStyleBackColor = true;

this.btnEvents.Click += new System.EventHandler(this.btnEvents_Click);

//

// groupBox3

//

this.groupBox3.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
```

```
this.groupBox3.Controls.Add(this.txtDeviceState);
this.groupBox3.Controls.Add(this.label6);
this.groupBox3.Controls.Add(this.txtSerialNumber);
this.groupBox3.Controls.Add(this.label8);
this.groupBox3.Controls.Add(this.txtInternalVersion);
this.groupBox3.Controls.Add(this.label9);
        this.groupBox3.Location = new System.Drawing.Point(542, 315);
        this.groupBox3.Name = "groupBox3";
        this.groupBox3.Size = new System.Drawing.Size(308, 103);
        this.groupBox3.TabIndex = 2;
        this.groupBox3.TabStop = false;
        this.groupBox3.Text = "Dispositivo";
        //
        // txtDeviceState
        //
        this.txtDeviceState.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
        this.txtDeviceState.BackColor = System.Drawing.SystemColors.InactiveBorder;
        this.txtDeviceState.Enabled = false;
        this.txtDeviceState.Location = new System.Drawing.Point(109, 19);
        this.txtDeviceState.Name = "txtDeviceState";
        this.txtDeviceState.Size = new System.Drawing.Size(190, 20);
        this.txtDeviceState.TabIndex = 1;
```

```
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(6, 21);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(105, 13);
this.label6.TabIndex = 0;
this.label6.Text = "Estado de Conexión ";
//
// txtSerialNumber
//
this.txtSerialNumber.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.txtSerialNumber.BackColor = System.Drawing.SystemColors.InactiveBorder;
this.txtSerialNumber.Enabled = false;
this.txtSerialNumber.Location = new System.Drawing.Point(109, 73);
this.txtSerialNumber.Name = "txtSerialNumber";
this.txtSerialNumber.Size = new System.Drawing.Size(190, 20);
this.txtSerialNumber.TabIndex = 1;
//
// label8
//
```

```
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(6, 75);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(97, 13);
this.label8.TabIndex = 0;
this.label8.Text = "Serial de Hardware";
//
// txtInternalVersion
//
this.txtInternalVersion.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.txtInternalVersion.BackColor = System.Drawing.SystemColors.InactiveBorder;
this.txtInternalVersion.Enabled = false;
this.txtInternalVersion.Location = new System.Drawing.Point(109, 47);
this.txtInternalVersion.Name = "txtInternalVersion";
this.txtInternalVersion.Size = new System.Drawing.Size(190, 20);
this.txtInternalVersion.TabIndex = 1;
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(6, 49);
this.label9.Name = "label9";
```

```
this.label9.Size = new System.Drawing.Size(78, 13);
this.label9.TabIndex = 0;
this.label9.Text = "Versión Interna";
//
// Code
//
this.Code.DataPropertyName = "Code";
this.Code.HeaderText = "No.Finca";
this.Code.Name = "Code";
this.Code.ReadOnly = true;
//
// Locker
//
this.Locker.DataPropertyName = "Name";
this.Locker.HeaderText = "Casillero";
this.Locker.Name = "Locker";
this.Locker.ReadOnly = true;
//
// Server
//
this.Server.AutoSizeMode =
System.Windows.Forms.DataGridViewAutoSizeColumnMode.Fill;
this.Server.DataPropertyName = "URL";
this.Server.HeaderText = "Servidor";
this.Server.Name = "Server";
```

```
            this.Server.ReadOnly = true;
            //
            // Device
            //
            this.Device.DataPropertyName = "Type";
this.Device.HeaderText = "Dispositivo";
            this.Device.Name = "Device";
            this.Device.ReadOnly = true;
            this.Device.Width = 150;
            //
            // MainForm
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(863, 463);
this.Controls.Add(this.groupBox2);
this.Controls.Add(this.groupBox3);
this.Controls.Add(this.groupBox1);
this.Controls.Add(this.btnEvents);
this.Controls.Add(this.grpLockers);
this.Controls.Add(this.btnExit);
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;
            this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.MaximizeBox = false;
```

```
            this.MinimizeBox = false;
            this.Name = "MainForm";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "Realtime Locker Manager";
            this.Load += new System.EventHandler(this.Main_Load);
            ((System.ComponentModel.ISupportInitialize)(this.dgLockers)).EndInit();
this.grpLockers.ResumeLayout(false);
this.groupBox1.ResumeLayout(false);
this.groupBox1.PerformLayout();
this.groupBox2.ResumeLayout(false);
this.groupBox2.PerformLayout();
this.groupBox3.ResumeLayout(false);
this.groupBox3.PerformLayout();
this.ResumeLayout(false);

}        #endregion
private System.Windows.Forms.DataGridView dgLockers;
private System.Windows.Forms.GroupBox grpLockers;
private System.Windows.Forms.Button btnEdit;
private System.Windows.Forms.Button btnDelete;
private System.Windows.Forms.Button btnAdd;
private System.Windows.Forms.GroupBox groupBox1;
private System.Windows.Forms.TextBox txtLockerName;
private System.Windows.Forms.Label label5;
```

```
private System.Windows.Forms.TextBox txtDeviceUrl;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.TextBox txtDeviceType;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.GroupBox groupBox2;
private System.Windows.Forms.Button btnAuto;
private System.Windows.Forms.Button btnAuthorize;
private System.Windows.Forms.Button btn9;
private System.Windows.Forms.Button btn6;
private System.Windows.Forms.Button btn3;
private System.Windows.Forms.Button btn0;
private System.Windows.Forms.Button btnReset;
private System.Windows.Forms.Button btn8;
private System.Windows.Forms.Button btn7;
private System.Windows.Forms.TextBox txtPin;
private System.Windows.Forms.Button btn5;
private System.Windows.Forms.Button btn4;
private System.Windows.Forms.Button btn2;
private System.Windows.Forms.Button btn1;
private System.Windows.Forms.Button btnConnect;
private System.Windows.Forms.Button btnExit;
private System.Windows.Forms.Button btnEvents;
private System.Windows.Forms.GroupBox groupBox3;
private System.Windows.Forms.TextBox txtDeviceState;
```

```
private System.Windows.Forms.Label label6;
private System.Windows.Forms.TextBox txtSerialNumber;
private System.Windows.Forms.Label label8;
private System.Windows.Forms.TextBox txtInternalVersion;
private System.Windows.Forms.Label label9;
private System.Windows.Forms.TextBox txtLocation;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.TextBox txtCode;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox txtArea;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.DataGridViewTextBoxColumn Code;
private System.Windows.Forms.DataGridViewTextBoxColumn Locker;
private System.Windows.Forms.DataGridViewTextBoxColumn Server;
private System.Windows.Forms.DataGridViewTextBoxColumn Device;
}
}
```

ANEXOS

Anexo 1: Anteproyecto de Grado.

TÍTULO DEL TEMA

PROPUESTA DE UN SISTEMA DE SEGURIDAD PARA PANELES BLINDADOS MEDIANTE EL USO DE LA CPU BEAGLEBONE BLACK, REPÚBLICA DOMINICANA, AÑO 2015.

1. INTRODUCCIÓN

Un sistema de seguridad integrado es el conjunto de elementos y dispositivos electrónicos que interconectados tienen la tarea de interactuar o reaccionar a ciertos cambios en el entorno donde se encuentran. El trabajo que se presenta está enfocado con la reducción de pérdidas de energía que presentan las Distribuidoras de Empresas Eléctricas, ocasionadas por acciones de terceros en los paneles blindados. Estas Empresas Eléctricas no solo se enfocan en la calidad del servicio sino también en la seguridad del mismo.

El mercado eléctrico Dominicano ha realizado importantes esfuerzos para combatir el hurto de energía. Sin embargo, las Distribuidoras de Empresas Eléctricas presentan cada vez mayores costos asociados a dichos programas de reducción de pérdidas, ocasionados por incrementos en los casos detectados (clientes hurtadores), y la reincidencia de los mismos.

Existen individuos que se dedican al hurto de energía, delito que es considerado como un ahorro económico y/o un ingreso adicional. El hurto de energía es actualmente considerado un problema económico y social, ya que afecta la rentabilidad de las Distribuidoras de Empresas Eléctricas, al incrementarse sus costos de operación.

Hoy en día el desarrollo de los sistemas basados en microcomputadores ha permitido la creación de un sin número de equipos y software que permiten manejar la seguridad física. Este proyecto estará enfocado en la utilización de sistemas embebidos para lograr la

integración de todos los dispositivos de seguridad, entre los sistemas embebidos existentes en el mercado se optará por la CPU Beaglebone Black debido a sus características.

2. JUSTIFICACIÓN

Los paneles blindados es una alternativa para asegurar físicamente los equipos de medida, impidiendo de esta forma su posible manipulación por parte de desaprensivos.

Estos paneles blindados que poseen las Distribuidoras de Empresas Eléctricas permiten la protección de los medidores de energía colocado en las viviendas de nuestro país. (Edeeste, 2014)

En la actualidad para los fines que fueron ideados y materializados dichos paneles han resultado un dolor de cabeza para las Distribuidoras de Empresas Eléctricas ya que el acceso ilegal a estos paneles es una de las mayores causas de pérdidas que tienen las Distribuidoras por las cantidades de violaciones que sufren los mismos y donde los usuarios modifican los medidores a su favor para pagar menos en sus facturas de energía eléctrica.

La Distribuidora de Energía Eléctrica EDEESTE en su informe de Gestión del año 2013, en sus resultados acumulados muestra que tuvo un 37.3% de pérdidas de energía desde agosto 2012-Mayo 2013, presentando 79,934 casos de fraudes e irregularidades normalizados, estos

fraudes incluyen violación de los paneles (Manipulación y alteración de los registros de consumo de energía), conexión directas, entre otros. (Edeeste, 2013)

La Distribuidora cuenta con alrededor de 3,000 paneles blindados en su zona de concesión que representan una facturación anual que sobrepasa los RD$1000MM de pesos dominicanos.

Casos como los anteriores demuestran que existe la necesidad de un mejor sistema de seguridad para proteger a los usuarios ya que están corriendo peligro al maniobrar dichos paneles y a las Distribuidoras de Empresas Eléctricas para no tener Pérdidas. Un problema como este es considerado como un delito por ende no solo estamos presentando una solución a las prestadoras de este servicio sino también una solución al país.

3. PLANTEAMIENTO DEL PROBLEMA

En la actualidad Edeeste cuenta con más de 600,000 suministros, donde el 10% promedio de los mismos están bajo la instalación de paneles y el 50 % de estos paneles poseen estructura blindada.

El propósito que se pretende con este proyecto es controlar y determinar quién vulnera estas instalaciones a través de las bondades que hoy en día nos ofrece la tecnología de última generación. Aunque la infraestructura de manera física es blindada, aun las intenciones fraudulentas la llevan a la realidad por la facilidad de acceso. El porcentaje de las pérdidas promedio en los paneles rondan el 30%, un número preocupante para la sostenibilidad financiera de la Distribuidora de Empresa Eléctrica.

Estos párrafos se antepuso a la idea que antes tenía haciendo referencia a información de terceros. Es la información actual de la empresa.

"Las condiciones de vida cambian continuamente, y si bien anteriormente eran contados los abonados que consideraban la necesidad de reducir su factura de electricidad, hoy debido a las crecientes aplicaciones de este fluido y a los mayores conocimientos que tales aplicaciones les proporcionan, consideran mayormente la conveniencia de reducir este capítulo de gastos del presupuesto doméstico, unido al lamentable concepto de que el hurto de corriente no constituye delito vergonzoso; esto hace que EN ESTOS ÚLTIMOS TIEMPOS el fraude eléctrico se cometa en tal proporción, que las empresas empiezan a

considerar la necesidad de atacarlo resueltamente por todos los medios a su alcance". (Porras, 2010)

Este comentario puede resultar muy normal y pudo ser una noticia de ayer en cualquier diario, pero no, fue escrito en el prólogo del libro "FRAUDES EN LOS MEDIDORES ELÉCTRICOS", en 1936 por el español Manuel Gabarro. Según lo anterior, siempre a través de los tiempos en que ha existido la energía y se ha cobrado este servicio, han existido clientes que no quieren pagar todo o parte de lo que consumen; estas personas se han ideado muchas formas para evadir los pagos y a través de una serie de etapas han empleado diversas "técnicas" para intervenir las acometidas y los medidores, "técnicas" que como dije antes las realiza él mismo o paga para que un electricista o uno que se cree electricista las realice; esta iniciativa puede partir del mismo cliente o incitada por otros.

En un principio cuando las comercializadoras de energía no ejercían ningún control sobre las pérdidas, el fraude los realizaba el mismo cliente o usuario y en algunas ocasiones pagaba para que un técnico los hiciera. A esto le llamo La primera etapa, los fraudes eran muy sencillos o "caseros" y se limitaban a instalar servicios directos, a soltar o correr el ancla de tensión cuando era exterior, aislar el ancla con esmalte o cinta, o a retirarla completamente, también ya se atrevían a invertir la entrada por salida y a instalar puentes entre la entra entrada y la salida. Pero las comercializadoras de energía comienzan a ver el perjuicio que esta modalidad le está causando a sus finanzas e inicia los programas de reducción de pérdidas; estos programas se limitaban al envió al terreno de cuadrillas que con solo realizar una inspección visual ubicaban las modalidades de fraude antes mencionadas.

3.1 DELIMITACIÓN EN TIEMPO Y ESPACIO

La investigación se llevará en uno de los panales blindados de la Empresa Distribuidora de Electricidad del Este, S. A. (EDEEste). La zona de concesión de dicha Distribuidora abarca desde la acera Este de la Ave. Máximo Gómez hasta la Provincia La Altagracia (Higüey) incluyendo Monte Plata, Hato Mayor, Seibo, Villa Mella y Santo Domingo Norte. Debido a que se puede elaborar un planteamiento claro de las necesidades de seguridad de los paneles ubicados dentro de la zona de concesión. Específicamente algunos sectores donde están instalados los paneles blindados.

El diseño e implementación se realizará durante el período Mayo – Agosto de 2015 porque se considera un tiempo prudente para su ejecución conceptual y física del proyecto.

4. OBJETIVOS

4.1 General

- Diseñar un sistema de alarma y monitoreo para mejorar la seguridad de los paneles blindados de la Empresa Distribuidora de Electricidad del Este, S. A. (EDEEste).

4.2 Específicos

- Identificar la problemática que se presenta a nivel de seguridad los paneles blindados.

- Diseñar un programa de interfaz web donde la Empresa Distribuidora de Electricidad pueda visualizar las alertas.
- Seleccionar sensores movimiento, magnéticos y de vibración en los alrededores de las instalaciones físicas.
- Seleccionar la estructura computacional que va hacer el análisis y envío de los datos.
- Evaluar las prioridades de seguridad en las ubicaciones donde los paneles son instalados.
- Determinar las medidas a implementarse de acuerdo a los resultados obtenidos del análisis y evaluación realizados dentro de las ubicaciones de los paneles y las informaciones de la Empresa Distribuidora de Electricidad.
- Garantizar la seguridad de la inversión por parte de la Empresa Distribuidora de Electricidad propiedad privada de la misma.

5. MARCO TEORICO REFERENCIAL

5.1 MARCO TEÓRICO

Al paso de los años el desarrollo de sistemas embebidos ha permitido el uso de herramientas independientes y autónomas que permiten múltiples aplicaciones de automatización a nivel cotidiano.

En República Dominicana los sistemas de seguridad son un hecho y existen sistemas usando microcomputadores los cuales ejercen aplicaciones específicas, a su vez existen compañías que ofrecen el servicio a las alertas que emiten estos sistemas de alarma.

Debido a lo anterior, es de gran importancia que los distintos paneles blindados instalados en la zona de concesión antes indicada, tengan sistemas de alarmas de forma centralizada para que estas sean monitoreadas por operadores a distancia desde la oficina principal de la Empresa Distribuidora de Electricidad y así evitar o dar respuestas rápidas a cualquier evento que pueda representar una amenaza para dichos paneles.

Con este minicomputador se han desarrollado una gran variedad de proyectos de distintas aplicaciones incluyendo sistemas de seguridad, aunque no se registran diseño de sistemas para paneles blindados con este minicomputador. Según sus características con relación a un computador común y corriente, este sería de gran ventaja para los sistemas de seguridad en los paneles blindados por su fácil reemplazo y ergonómico tamaño además de su potente procesamiento para aplicaciones diversas.

En el territorio dominicano la violación de los paneles blindados se realiza con gran facilidad por fallas en la seguridad tales son los casos que enumeremos a continuación:

1. Residencial. Dulce María
2. Issfapol (1)
3. Residencial Francia I - Panel A

4. Residencial Francia I - Panel B
5. Residencial Hermes Primera
6. Residencial HLC I - Vista Hermosa
7. Condominio. ZN - Eugenio María De Hostos (1)
8. Residencial Charles de Gaulle (2)
9. Av. Faro a Colón No. 9 - Faro a Colón (1)
10. Av. San Martin No. 57 - Villa Consuelo (2)
11. Duarte Edificio 35 - Los Multis
12. Faro a Colón Edificio 10.
13. Faro A. Colón - Plaza Boulevard
14. Prados Cachón de La Rubia
15. Jardines del Norte - Residencial Doña Ivonne I
16. Colinas Del Este - Vista Hermosa
17. Profesional Monte Cristi Co. - San Carlos
18. Residencial. JP - Alma Rosa I (1)
19. Trinitaria No. 23 - San Carlos (2)
20. Plaza Enriquillo - Urbanización Mallen (Antes)

EL ROBO DE ENERGÍA PROVOCA PÉRDIDAS DE US$500 MM AL AÑO.

Así lo afirmó el titular de la Procuraduría General Adjunta para el Sistema Eléctrico (PGASE), Moisés Ferrer, tras indicar que en el país las pérdidas por el robo de la energía es de, aproximadamente, 500 millones de dólares al año.

En la entrada de esta dependencia de la Procuraduría, la bienvenida es auspiciada por más de 20 transformadores incautados a algunas de unas tres mil personas que fueron sometidas a la Justicia el pasado año por atentar contra el sistema energético. La entidad también posee parte de su evidencia en el techo de su local, expuesta a las condiciones climáticas y a los malhechores que quieran hurtarlas o manipularlas. (Periódico El Caribe, 2013).

¿Cómo realizan los fraudes?

Moisés Ferrer cuenta que pontear los medidores y conectarse al sistema de manera directa son prácticas cavernícolas de robarse la luz, que se combatieron con tecnología. “Luego llegaron los medidores digitales, y llegó el llamado borrado de lectura, a esto se sumó los busters (un componente grande hecho con alambre de cobre e imán, que limitaba el conteo de energía en un 50% o 75%), y la cajita mágica, que eran los mismo busters pero más sofisticados”, detalla Ferrer.

En un caso curioso y hasta jocoso, citó que hace un año descubrieron la utilización de los controles digitales y/o controles universales de Televisión para el robo de energía. “Por medio de una clave lo ponían a que funcionaran para paralizar los medidores tecnológicos digitales”, dijo.

Usando esta tecnología las personas detenían su consumo en el momento deseado y volvían a encender el medidor cuando les parecía. También señaló que las grandes edificaciones y/o empresas camuflan las conexiones o utilizan paneles blindados para evadir y/o reducir los pagos. Citó entre los problemas que agudizan este flagelo "la complicidad" con personal subcontratista, que en ocasiones incitan a los ciudadanos a cometer fraude, y la negligencia, esta última porque a veces las personas solicitan el servicio y ante la dilación de las Edes, deciden instalarlo por cuenta propia.

LAS PÉRDIDAS DE ENERGÍA Y SU IMPACTO EN EL DÉFICIT ELÉCTRICO

Las Pérdidas Eléctricas se clasifican en Técnicas y No Técnicas.

En la lógica más elemental de cualquier negocio, las pérdidas se producen cuando los ingresos son menores que los egresos. Con el negocio de la electricidad sucede exactamente lo mismo; los ingresos de las empresas distribuidoras (EDEs) son menores al valor total de la energía consumida por los usuarios. Esto se debe a que un alto porcentaje de la energía consumida no es facturada y por tanto no se refleja en los ingresos. (Periódico Listín Diario, 2014)

Las pérdidas eléctricas se clasifican en técnicas y no técnicas. Las primeras están vinculadas con la energía que se pierde en el proceso de transformación, medición y transporte de la

electricidad a través del uso de materiales y equipos. Desde el punto de vista de las leyes físicas es imposible reducirla a cero por ciento las pérdidas técnicas. En el caso de las segundas, las pérdidas no técnicas, estás tienen su origen en ineficiencias de carácter administrativo, comercial y en el uso indebido o robo de la electricidad.

En términos reales es posible reducirlas a su mínima expresión si se logra cobrar toda la energía que se consume.

El Informe de Desempeño de la CDEEE de febrero del 2014 señala, que del total de la energía servida por las tres (3) distribuidoras de electricidad se pierde en promedio el 33%, de la siguiente manera, Edenorte pierde el 34.3%, Edesur el 29.2% y Edeeste el 36.1%. De este porcentaje de energía perdida, las razones técnicas son mucho menores que las no técnicas en términos porcentuales.

Si comparamos las cifras oficiales de pérdida de energía (33%) con las ofrecidas en el Informe de Indicadores de Desarrollo Mundial 2014 (World Development Indicators) del Banco Mundial nos daremos cuenta que la República Dominicana ocupa el cuarto lugar entre los países que más perdida de energía tienen de 147 países evaluados.

Sólo Botswana (56%), El Congo (46%) y Haití (55%) nos superan en ese renglón.

De acuerdo a dicho estudio, los países de bajos ingresos tienen un promedio de pérdidas de energía de 15% y los países de ingreso medio en cuya categoría fue clasificada República Dominicana tienen un promedio de pérdidas de 9%. Es decir, que para alcanzar el valor promedio de nuestra categoría como país, tendríamos que reducir 24 puntos porcentuales respecto al estado actual.

¿Cuánto es el monto del déficit de las empresas distribuidoras por concepto de las pérdidas? Según cifras oficiales, a diciembre del 2013 las empresas distribuidoras compraron energía por un monto de US$1,971.4 millones de dólares y de esa energía lograron facturar US$1,544.5 millones de dólares, de los cuales US$1,474 millones de dólares se tradujeron en cobros. Como se puede observar, el déficit de las EDEs generado por la energía dejada de cobrar fue de US$497.4 millones de dólares o su equivalente de RD$21,636 millones de pesos en el año 2013. Este monto del déficit equivale a la cantidad de dinero que pierde el Estado por concepto de energía consumida que no se factura y representa el 25% de la energía comprada a las empresas generadoras.

El déficit financiero de las EDEs originado por la pérdida de la energía han llevado a las autoridades a limitar la oferta sobre la base de la no rentabilidad al satisfacer la demanda completa. En términos llanos, lo que queremos señalar es que dentro de la lógica financiera de las distribuidoras, mientras más energía suministran más pérdida de dinero tienen.

No es lo mismo perder el 33% de 100 unidades, que perder el 33% de 80 unidades; en la primera pierdes 33 unidades del total servido y en la segunda pierdes 26.4 unidades. Esta simple lógica es que le da origen a los circuitos 24 horas, aquellos clasificados de esa manera porque reciben electricidad constante sobre la base de tener una alta tasa de pago de servicio.

Los bajos ingresos mensuales de las EDEs les generan retrasos con sus suplidores, además que las limita para realizar las inversiones necesarias para superar la situación anteriormente descrita.

Se requiere invertir cuantiosos recursos en la compra de materiales, equipos, y blindaje de las redes de distribución, así como en la compra e instalación de medidores eléctricos.

Estimaciones realizadas por organismos internacionales indican que el orden de magnitud de las inversiones anuales necesarias se encuentra por encima de los US$100 millones de dólares.

Luego de realizadas dichas inversiones, hay que monitorear de manera permanente los circuitos de distribución, para hacer sostenibles los logros que se vayan alcanzado.

¿Son los clientes de las distribuidoras morosos o mala paga?

Las cifras demuestran que no, ya que las cobranzas en febrero del 2014 promediaron un 98% y en los últimos años han estado por encima de 85%. Esto se traduce como una muy alta propensión al pago, donde casi la totalidad de las personas que recibieron una factura eléctrica la pagaron de alguna u otra forma. Queda evidenciado que el gran reto para reducir las pérdidas no técnicas está en aumentar la facturación convirtiendo la mayor cantidad de consumidores en clientes.

Indudablemente, todos debemos contribuir para que las empresas distribuidoras sean rentables. Es un reto que nos involucra a todos y especialmente a las autoridades que dirigen esas empresas, quienes deberán seguir fortaleciendo los aspectos gerenciales, técnicos, comerciales y su vinculación social con la población, de suerte tal que continúen reduciendo drásticamente las pérdidas de energía.

6. HIPOTESIS

El desarrollo de un sistema de alarma y monitoreo para la protección de los paneles blindados de la Compañía Distribuidora de Energía EDEESTE.

7. DISEÑO METODOLÓGICO: Metodología y técnicas de investigación.

A continuación presentamos los distintos métodos, técnicas y tipo de investigación que se utilizarán durante el desarrollo de este trabajo.

7.1 MÉTODO

El método que se pretende utilizar durante esta investigación de orden deductivo porque se partirá de datos ya aceptados como válidos para llegar a una solución.

7.2. TÉCNICAS

Las técnicas que se utilizarán son entrevistas, observación y la recopilación documental. Debido a esto se usarán diferentes fuentes de obtención de los datos y métodos de recolección de datos, las cuales se describen a continuación.

7.2.1 La entrevista

Se entrevistaran a: Directores, Gerentes y personal de seguridad física SSFF de la Empresa Distribuidora de Electricidad para determinar sus inquietudes a nivel de seguridad, y conocer sus opiniones sobre el impacto que tendría el proyecto.

7.2.2 Fuentes bibliográficas

Se utilizarán, principalmente, investigaciones ya documentadas, revistas y base de datos en línea en la Internet y estadísticos. Así como también artículos publicados por periódicos locales e internacionales. Además, enciclopedias y diccionarios.

7.2.3 La observación

Se pretende visitar paneles blindados instalados en distintas localidades Empresa Distribuidora de Electricidad y realizar las observaciones que nos permitan la obtención de datos y análisis detallado.

8. ESQUEMA PRELIMINAR DEL TRABAJO DE GRADO

Carátula.
Dedicatorias.
Agradecimientos.
Índice.
Introducción.

Capítulo 1. Sistema de alarma y monitoreo

1.1 Sistemas de alarma, estructuras y características.

1.2 Definición y características de sistemas de monitoreo.

1.3 Microcomputadores, estructura y características.

1.4 Sistemas Embebidos.

1.5 Beaglebone Black, teoría y aplicaciones.

1.6 Interfaz gráfica, características.

Capítulo 2. Diseño del sistema de alarma y monitoreo del proyecto.

2.1 Estructura de los paneles blindados, criterios para selección del área donde se implementará el sistema.

2.2 Selección de los componentes que estructuraran el sistema.

2.3 Integración de los componentes del sistema con la herramienta BeagleBone.

2.4 Diseño de la interfaz gráfica.

2.5 Implementación del sistema.

Capítulo 3. Validación del diseño del sistema de alarma y monitoreo.

3.1 Descripción de las pruebas del sistema.

3.2 Resultados de las pruebas.

3.3 Comparación de las pruebas con los cálculos del diseño.

Conclusión.

Recomendaciones.

Bibliografía.

Anexos.

FUENTES DE DOCUMENTACION

Página oficial de Beaglebone: http://beagleboard.org/Products/BeagleBone

Modelo de requisitos para sistemas embebidos. (n.d.). Recuperado de

Sistemas Embebidos. (n.d.).

http://www.idose.es/sistemas-embebidos

http://www.redalyc.org/articulo.oa?id=75071308

Bibliografía

Edeeste. (3 de Julio de 2013). *Edeeste.* Obtenido de Edeeste: http://www.edeeste.com.do/wp-content/uploads/2013/07/3.-EDEEste-Resultados-Gestin-May-2013-Hotel-Hilton-080613.pdf.

Edeeste. (Marzo de 2014). *IADB.* Obtenido de IADB: http://www.iadb.org/projectDocument.cfm?id=37281441

Periódico El Caribe. (29 de Enero de 2013). *El robo de energía provoca pérdidas de US$500 MM al año* , pág. 5.

Porras, H. O. (2010). Las Pérdidas de Energía, Enfoque Operativo. En *Detectar* (pág. 93). Colombia: Cartoprint tda.

Periódico Listín Diario. (6 de junio de 2014). Las pérdidas de energía y su impacto en el déficit eléctrico.

ANEXO 2: FORMULARIO DE SOLICITUD DE APROBACIÓN DE TEMA DE TRABAJO DE GRADO.

UNAPEC
UNIVERSIDAD APEC

DECANATO DE INGENIERÍA E INFORMÁTICA
ESCUELA DE INGENIERÍA

PROPUESTA DE UN SISTEMA DE SEGURIDAD PARA PANELES BLINDADOS MEDIANTE EL USO DE LA CPU BEAGLEBONE BLACK, REPUBLICA DOMINICANA, AÑO 2015.

Anteproyecto de trabajo de grado para optar por el título de Ingeniero Electrónico
Mención Telecomunicaciones.

Sustentante:

Asesor:

Docente investigador.

DEPARTAMENTO DE SOCIALES
PROYECTO DE TRABAJO DE GRADO
APROBADO
FIRMA
FECHA 13/4/15

Santo Domingo, D. N. República Dominicana
Abril de 2015

ANEXO 3: ESQUEMÁTICO BEAGLEBONE BLACK

REV	Description	DATE	BY
A4A	Initial production Release.	11/19/2012	GC
A5	On the initial production release the processors were to be found incorrect as supplied by TI. Parts while marked AM3359 were actually AM3352. This revision uses the correct parts.	1/2/2013	GC
A5A	1. Deleted R29-R44 from the LCD lines. 2. Added 47pf capacitors C156-C173 to LCD data lines to ground. 3. Changed schematic revision to A5A. 4. Changed a few footprints after PCB update for above changes. 5. Added access point for the battery function of the TPS65217C. 6. Added Ferrite beads in series with LED power and 5V power rail of the USB host connector. Required to pass FCC/CE testing due to noise emissions on that pin. 7. Added power button to enable sleep, wakeup, power down and power up features on the system. 8. Added Modification to add 100K ohm resistor to ground to prvent crosstalk when serial cable is not plugged in.	4/1/2013	GC

PAGE NO.	SCHEMATIC PAGE
1	COVER PAGE
2	POWER MANAGEMENT
3	PROCESSOR 1 OF 3, JTAG HEADER
4	PROCESSOR 2 OF 3, UAB PORTS
5	PROCESSOR 3 OF 3
6	LED, CONFIGURATION AND BUTTON
7	DDR3 MEMORY
8	eMMC FLASH
9	10/100 ETHERNET
10	HDMI FRAMER
11	EXP CONN, uSD

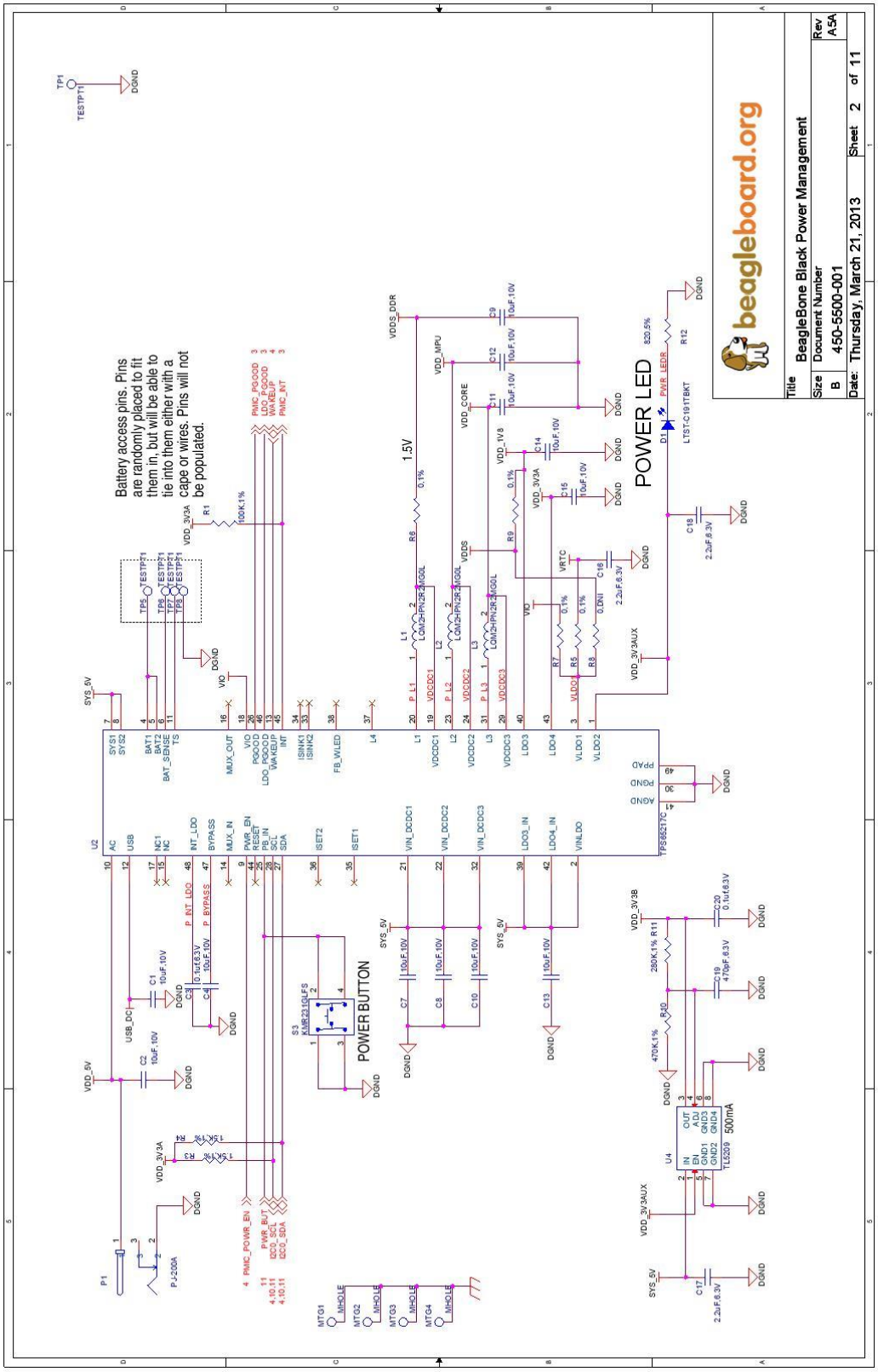
NOTE: PCB Revision for this board is Rev B3.

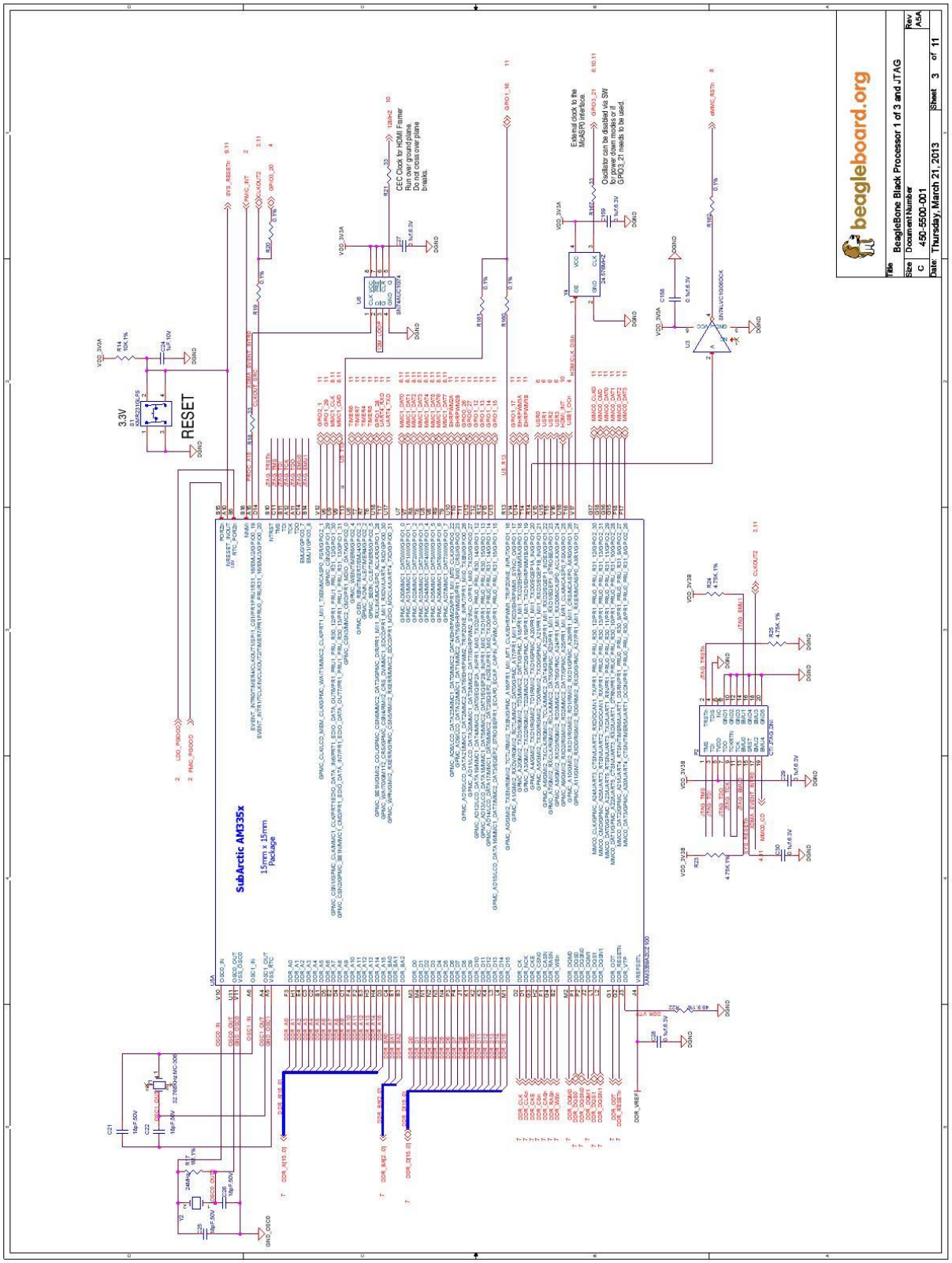
This schematic is *NOT SUPPORTED* and DOES NOT constitute a reference design. Only "community" support is allowed via resources at BeagleBoard.org/discuss.

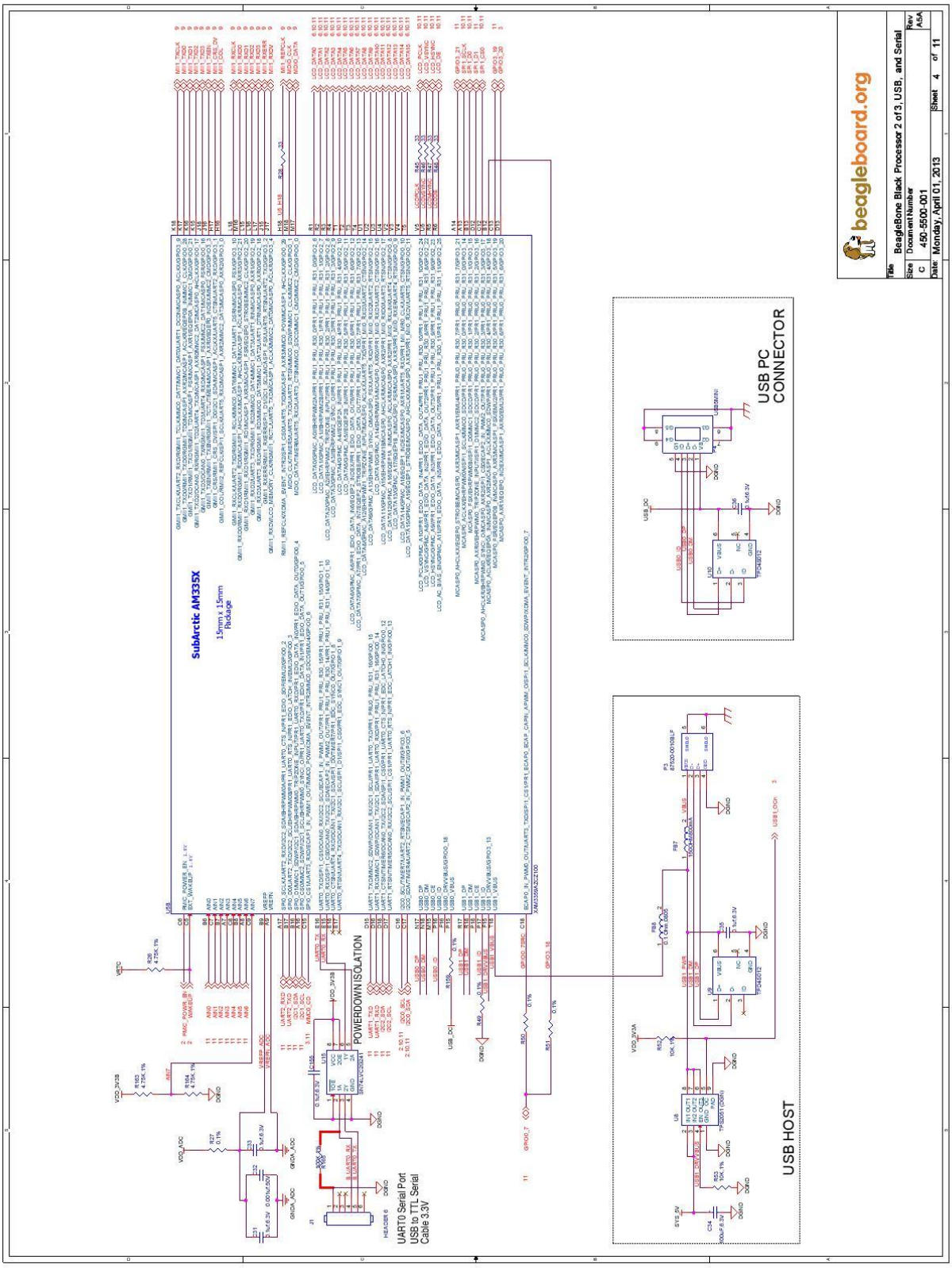
THERE IS NO WARRANTY FOR THIS DESIGN , TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE DESIGN "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE DESIGN IS WITH YOU. SHOULD THE DESIGN PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

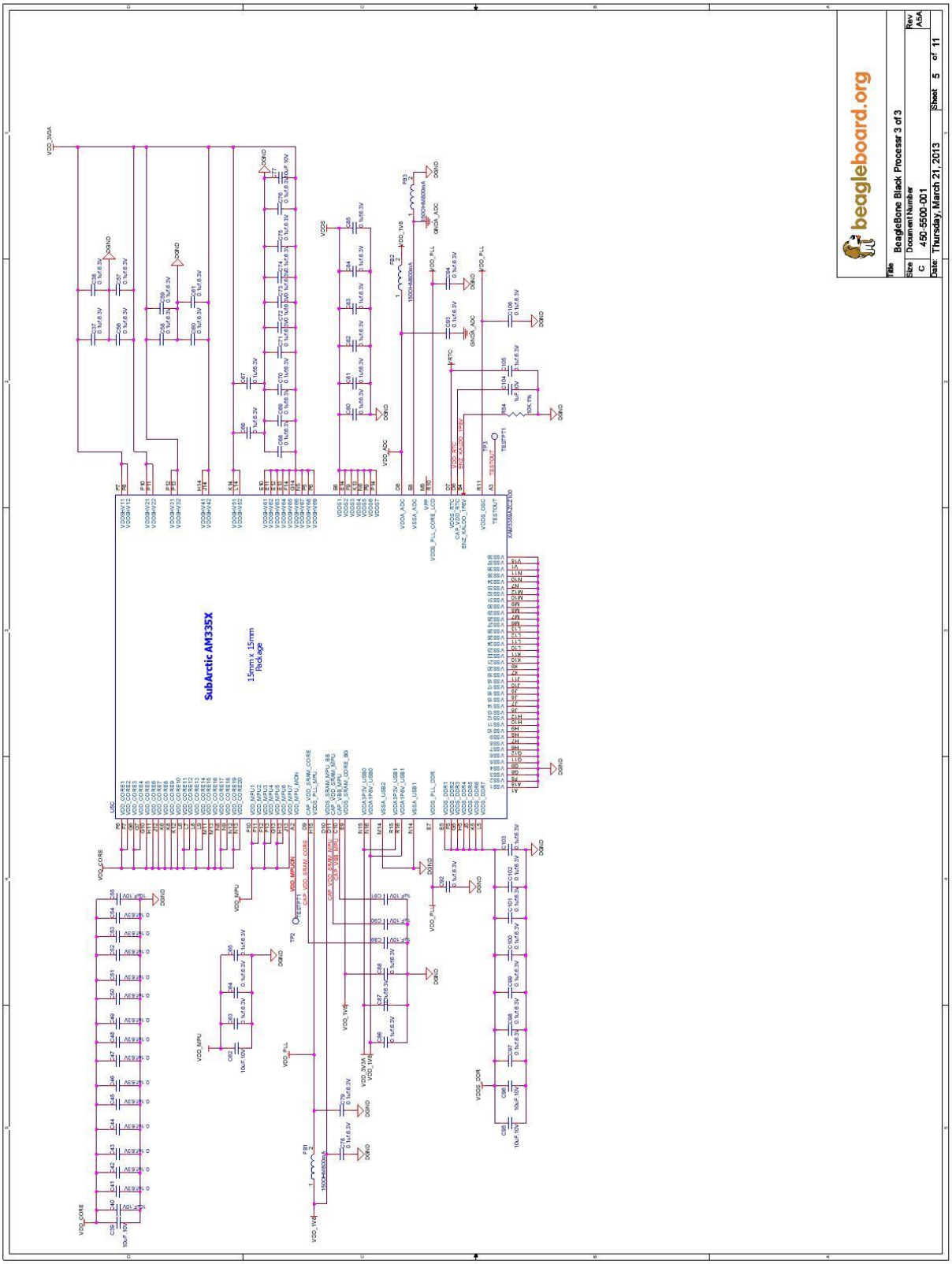
beagleboard.org

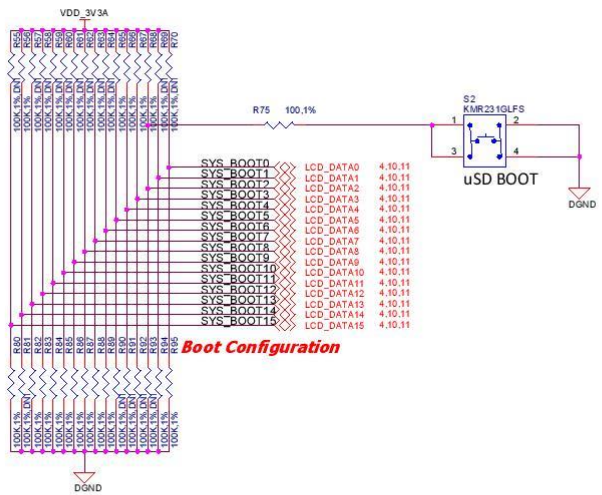
Title	BeagleBone Black Cover Page
Size B	Document Number 450-5500-001
Rev	A5A
Date:	Monday, April 01, 2013
Sheet	1 of 11

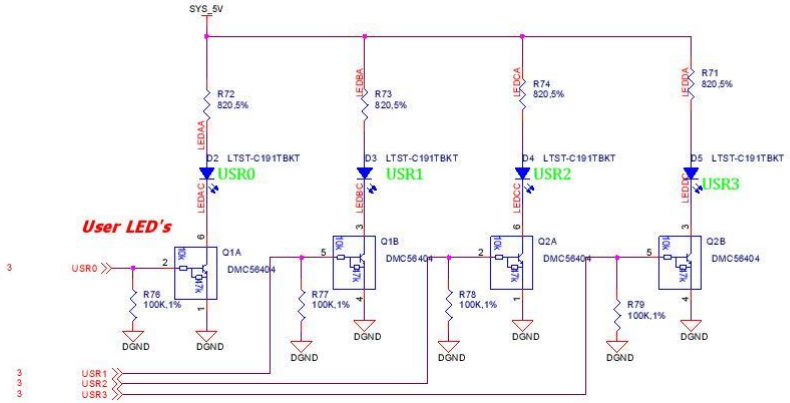

Battery access pins. Pins are randomly placed to fit them in, but will be able to tie into them either with a cape or wires. Pins will not be populated.
POWER BUTTON
POWER LED
1.5V
500mA
beagleboard.org
Title
BeagleBone Black Power Management
Size
Document Number
Rev
B
450-5500-001
A5A
Date: Thursday, March 21, 2013
Sheet 2 of 11


beagleboard.org
Title BeagleBone Black Processor 1 of 3 and JTAG
Size Document Number Rev A5A
C 450-5500-001
Date: Thursday, March 21, 2013 Sheet 3 of 11
SubArctic AM335x
15mm x 15mm Package
RESET
3.3V
CEC Clock for HDMI Framer
Run over ground plane.
Do not cross over plane breaks.
External clock to the McASP0 interface.
Oscillator can be disabled via SW to power down modes or if GPIO3_21 needs to be used.


beagleboard.org
Title
BeagleBone Black Processor 2 of 3, USB, and Serial
Size
C
Document Number
450-5500-001
Rev
A5A
Date: Monday, April 01, 2013
Sheet 4 of 11
SubArctic AM335X
15mm x 15mm
Package
USB PC
CONNECTOR
USB HOST
POWERDOWN ISOLATION
UART0 Serial Port
USB to TTL Serial
Cable 3.3V


beagleboard.org
Title: BeagleBone Black Processor 3 of 3
Size: C
Document Number: 450-5500-001
Rev: A5A
Date: Thursday, March 21, 2013
Sheet 5 of 11
SubArctic AM335X
15mm x 15mm Package

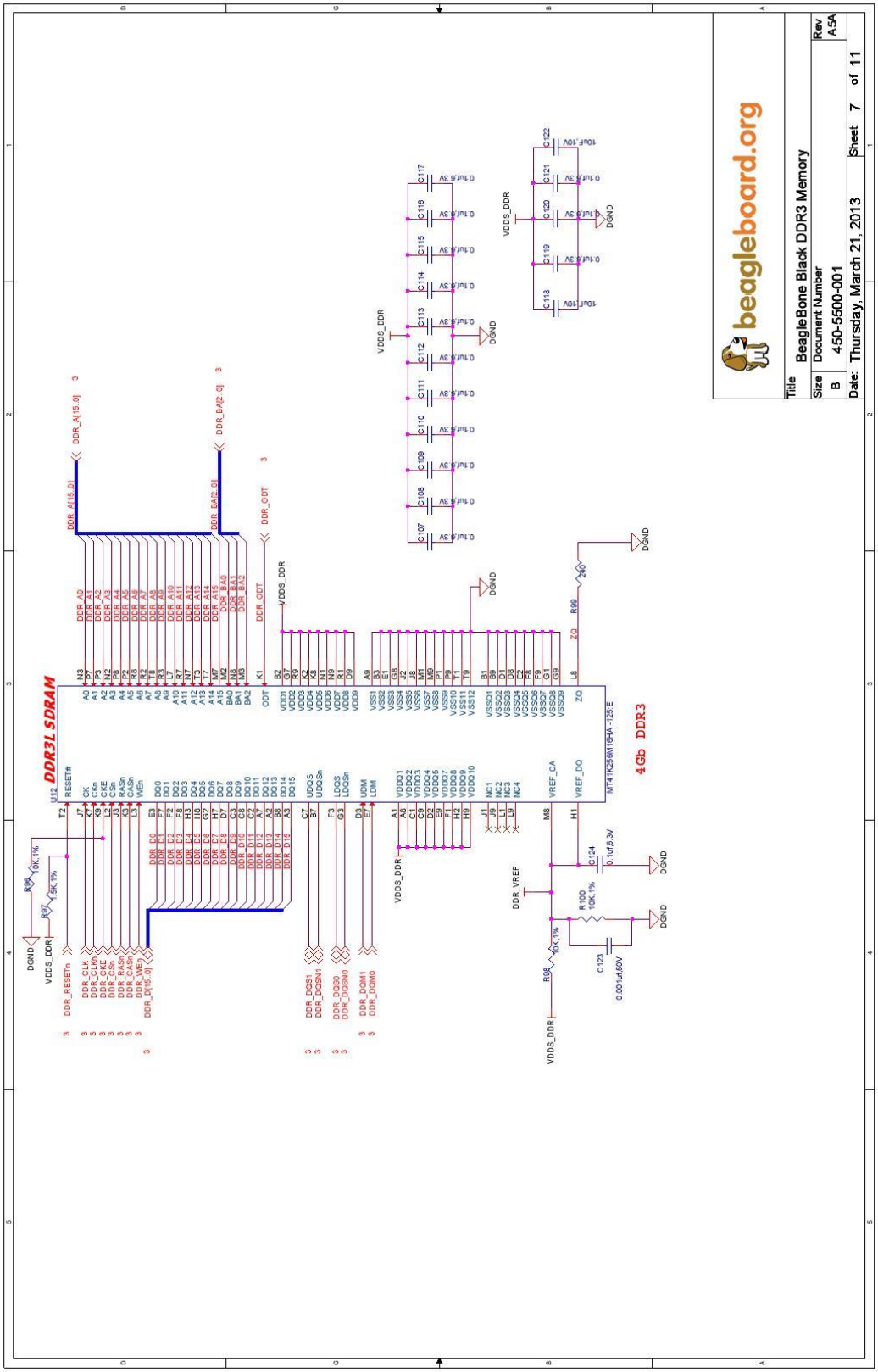


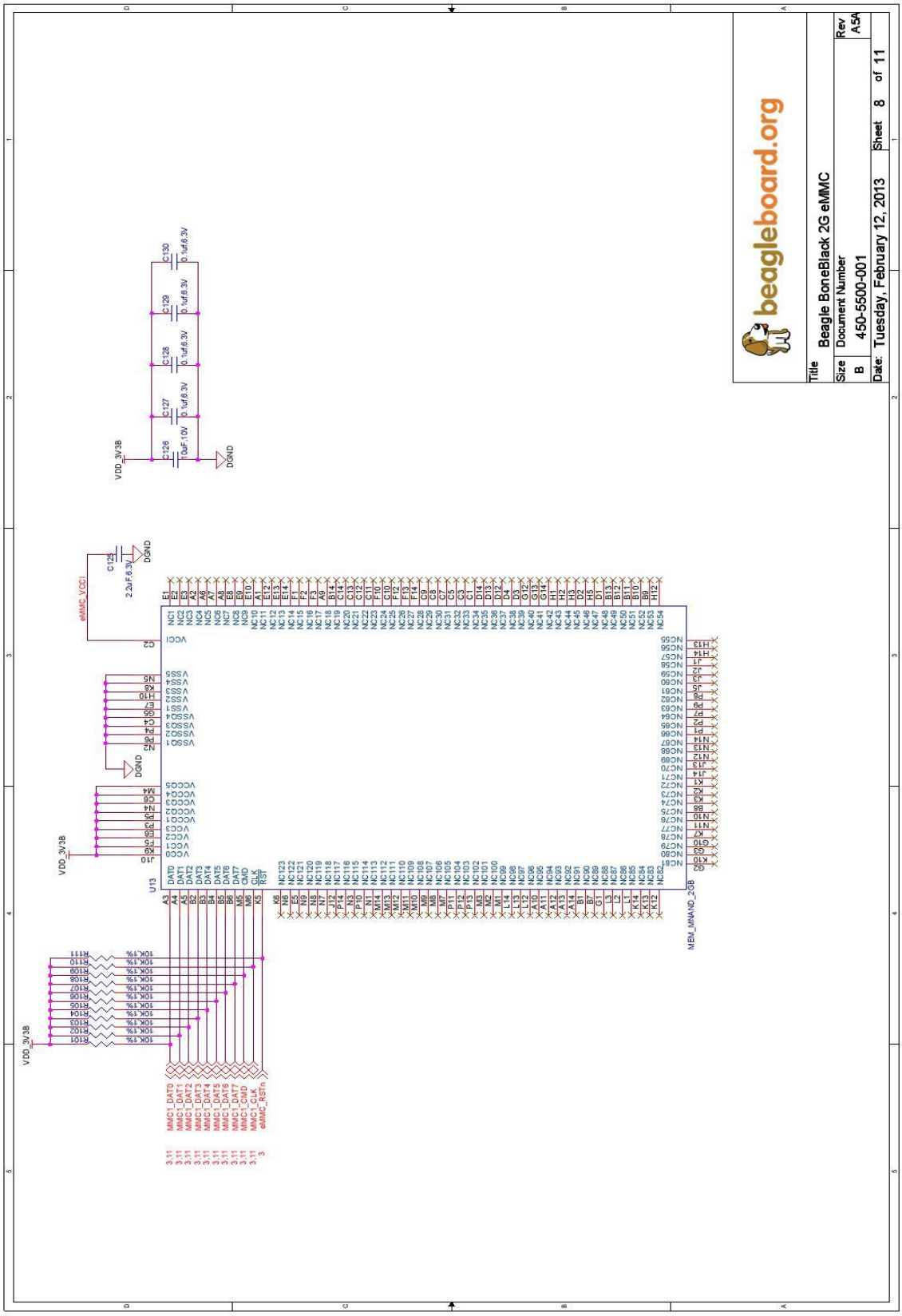


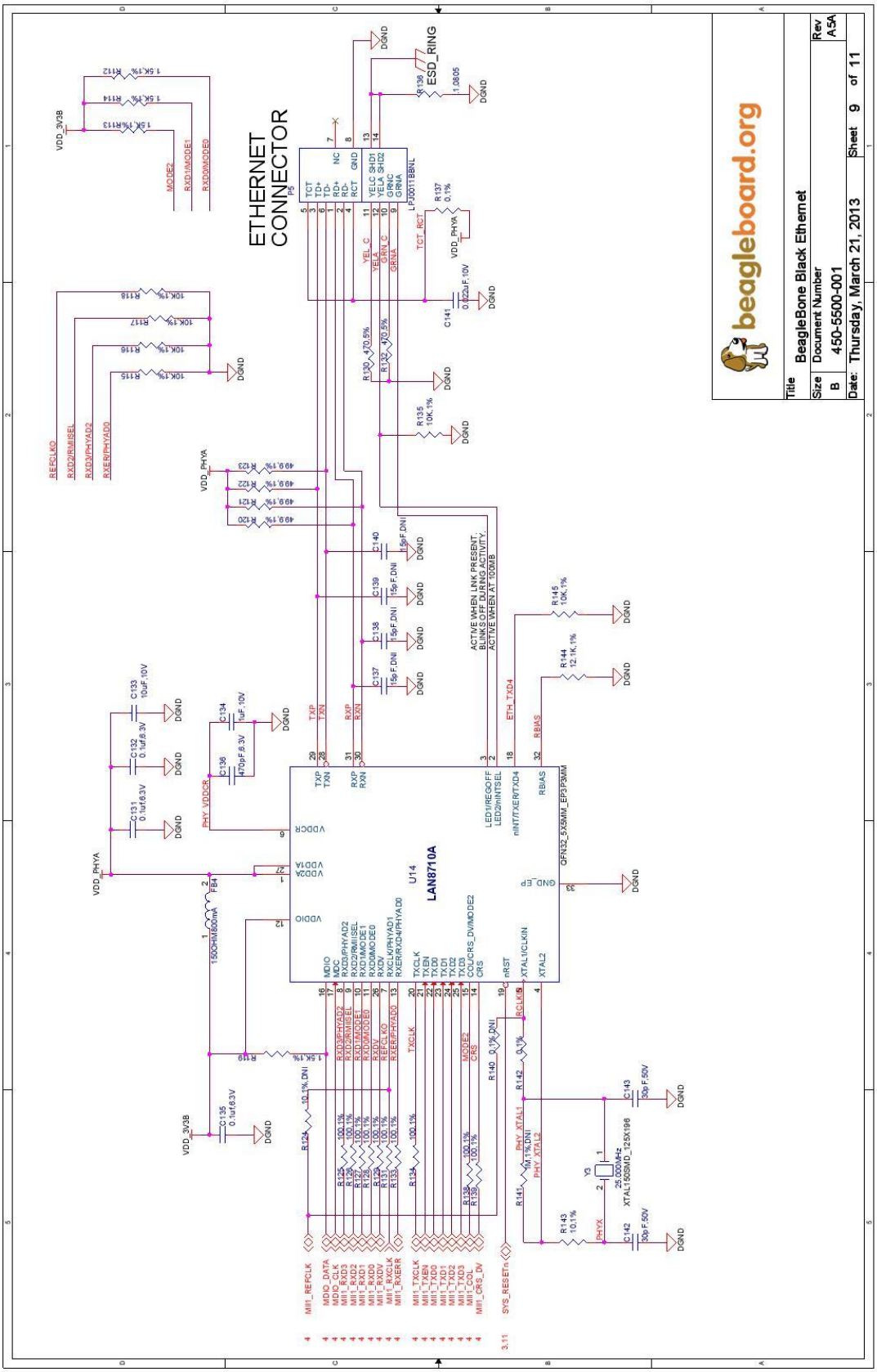
SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11100b	MMC1	MMC0	UART0	USB0[5]
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11000b	SPI0	MMC0	USB0[5]	UART0

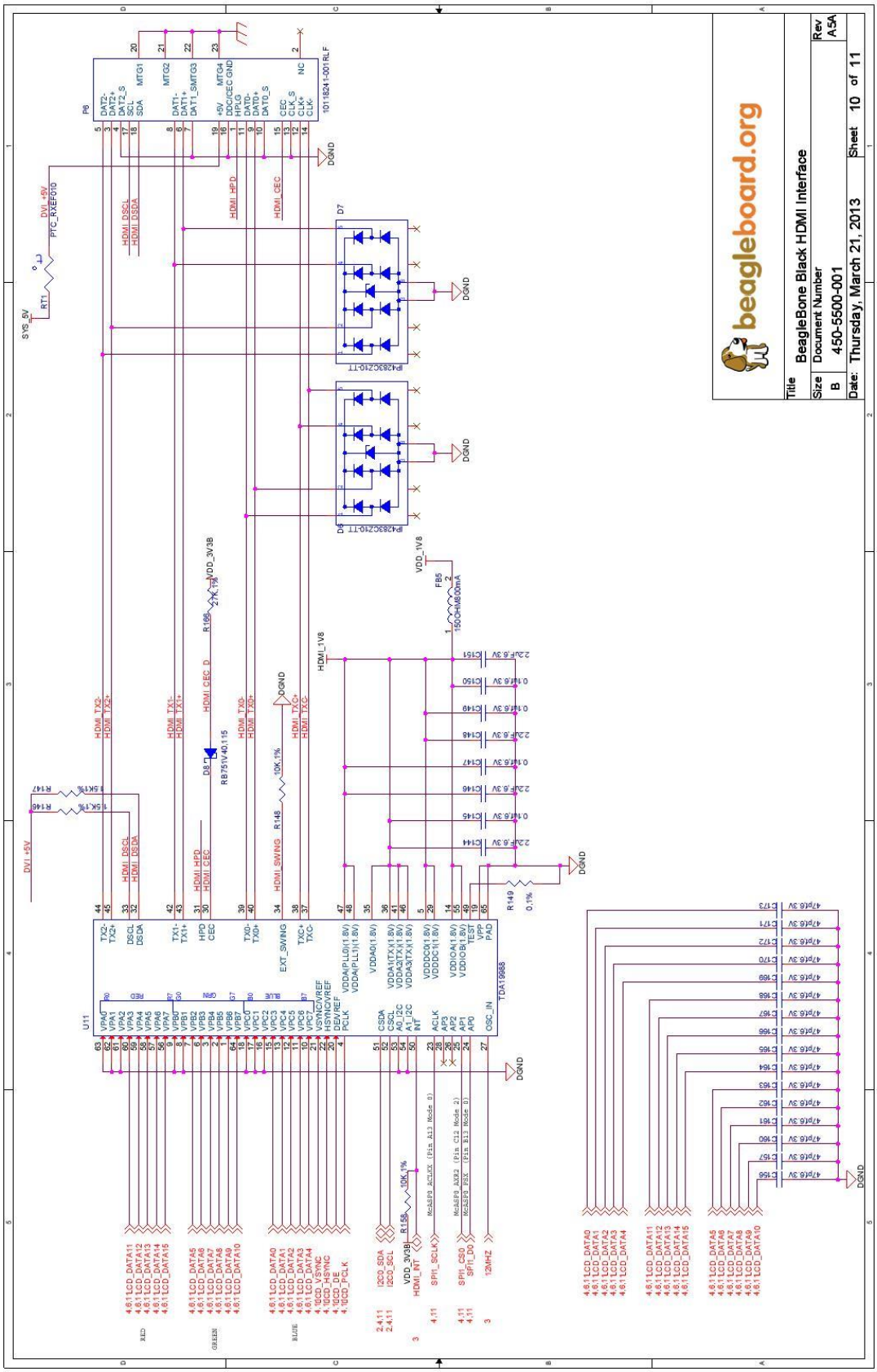
beagleboard.org

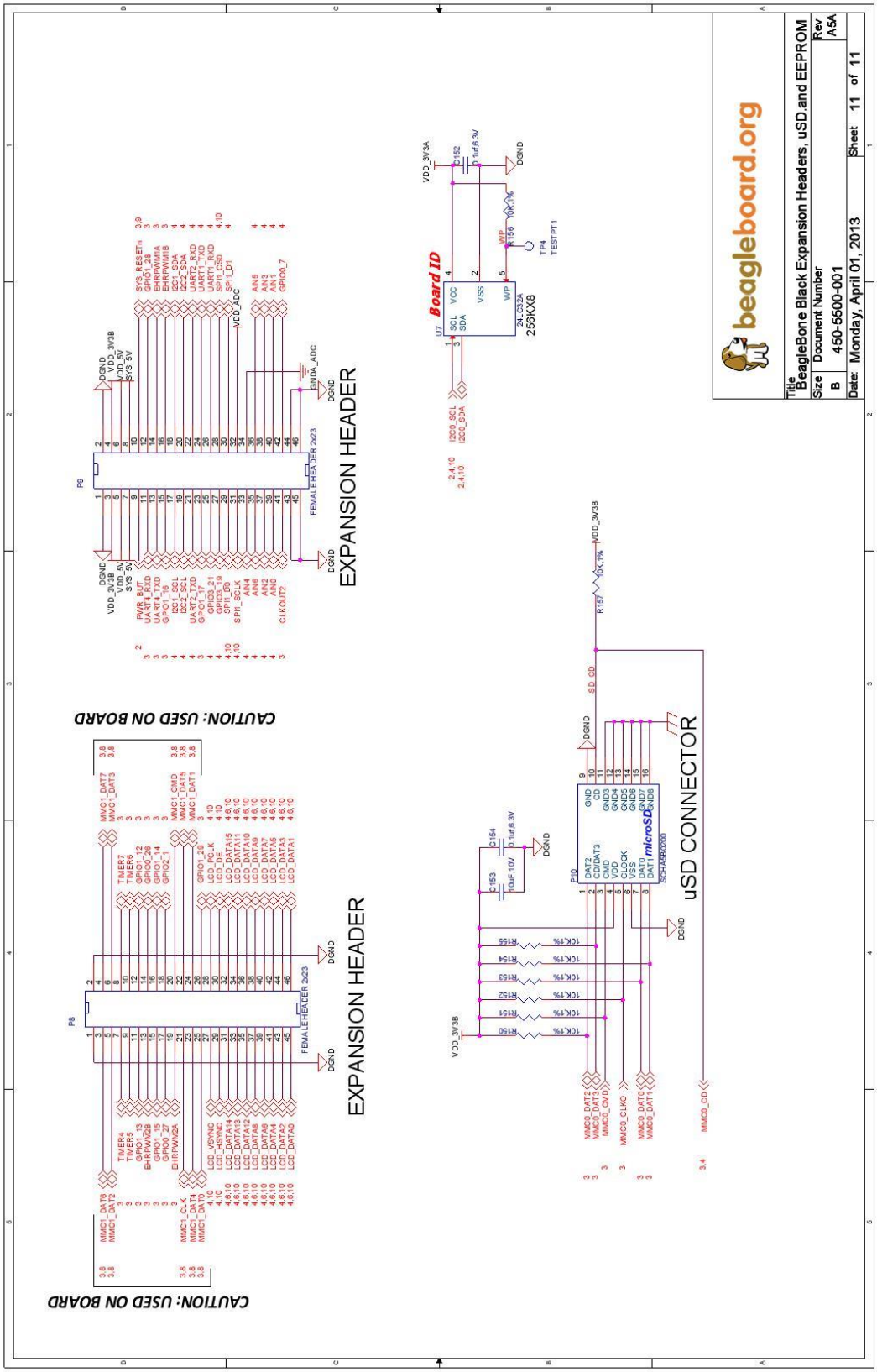
Title	BeagleBone Black LED, Configuration, and Reset
Size B	Document Number 450-5500-001 Rev A5A
Date: Thursday, March 21, 2013	Sheet 6 of 11


beagleboard.org
Title
BeagleBone Black DDR3 Memory
Size
Document Number
Rev
B
450-5500-001
A5A
Date: Thursday, March 21, 2013
Sheet 7 of 11
DDR3L SDRAM
4Gb DDR3
MT41K256M16HA-125:E
U12
VDDS_DDR
DGND
DDR_A[15..0]
DDR_BA[2..0]
DDR_ODT
DDR_RESETn
DDR_CLK
DDR_CLKn
DDR_CKE
DDR_CSn
DDR_RASn
DDR_CASn
DDR_WEn
DDR_D[15..0]
DDR_DQS1
DDR_DQSN1
DDR_DQS0
DDR_DQSN0
DDR_DQM1
DDR_DQM0
DDR_VREF
R96
0K,1%
R97
1.5K,1%
R98
0K,1%
R100
10K,1%
R99
240
C123
0.001uf,50V
C124
0.1uf,6.3V
C107
C108
C109
C110
C111
C112
C113
C114
C115
C116
C117
C118
C119
C120
C121
C122
0.1uf,6.3V
10uF,10V

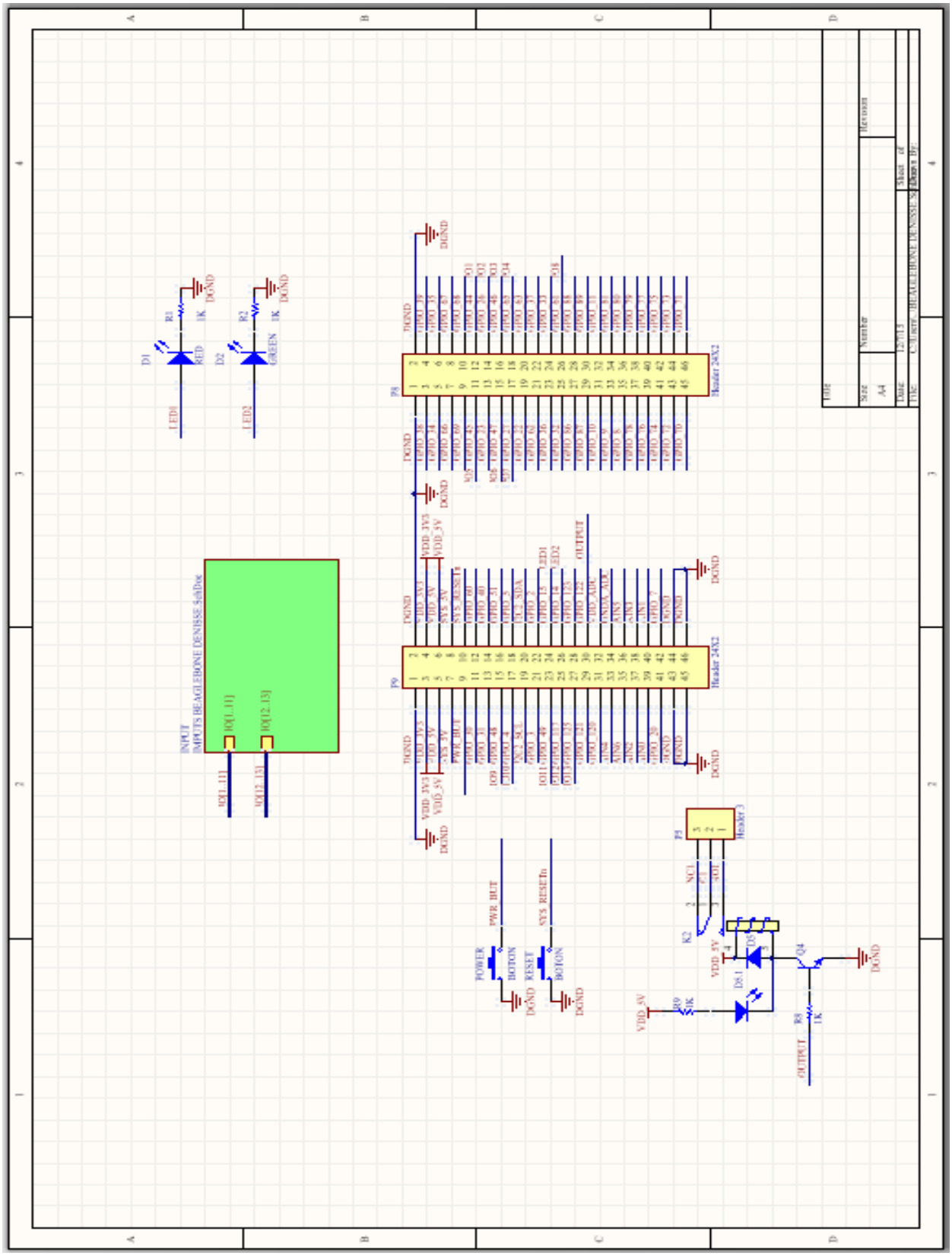

beagleboard.org
Title
Beagle BoneBlack 2G eMMC
Size
B
Document Number
450-5500-001
Rev
A5A
Date: Tuesday, February 12, 2013
Sheet 8 of 11
U13
MEM_MNAND_2GB
VDD_3V3B
DGND
eMMC_VCC1
C125
2.2uF,6.3V
C126
10uF,10V
C127
C128
C129
C130
0.1uf,6.3V
MMC1_DAT0
MMC1_DAT1
MMC1_DAT2
MMC1_DAT3
MMC1_DAT4
MMC1_DAT5
MMC1_DAT6
MMC1_DAT7
MMC1_CMD
MMC1_CLK
eMMC_RSTn

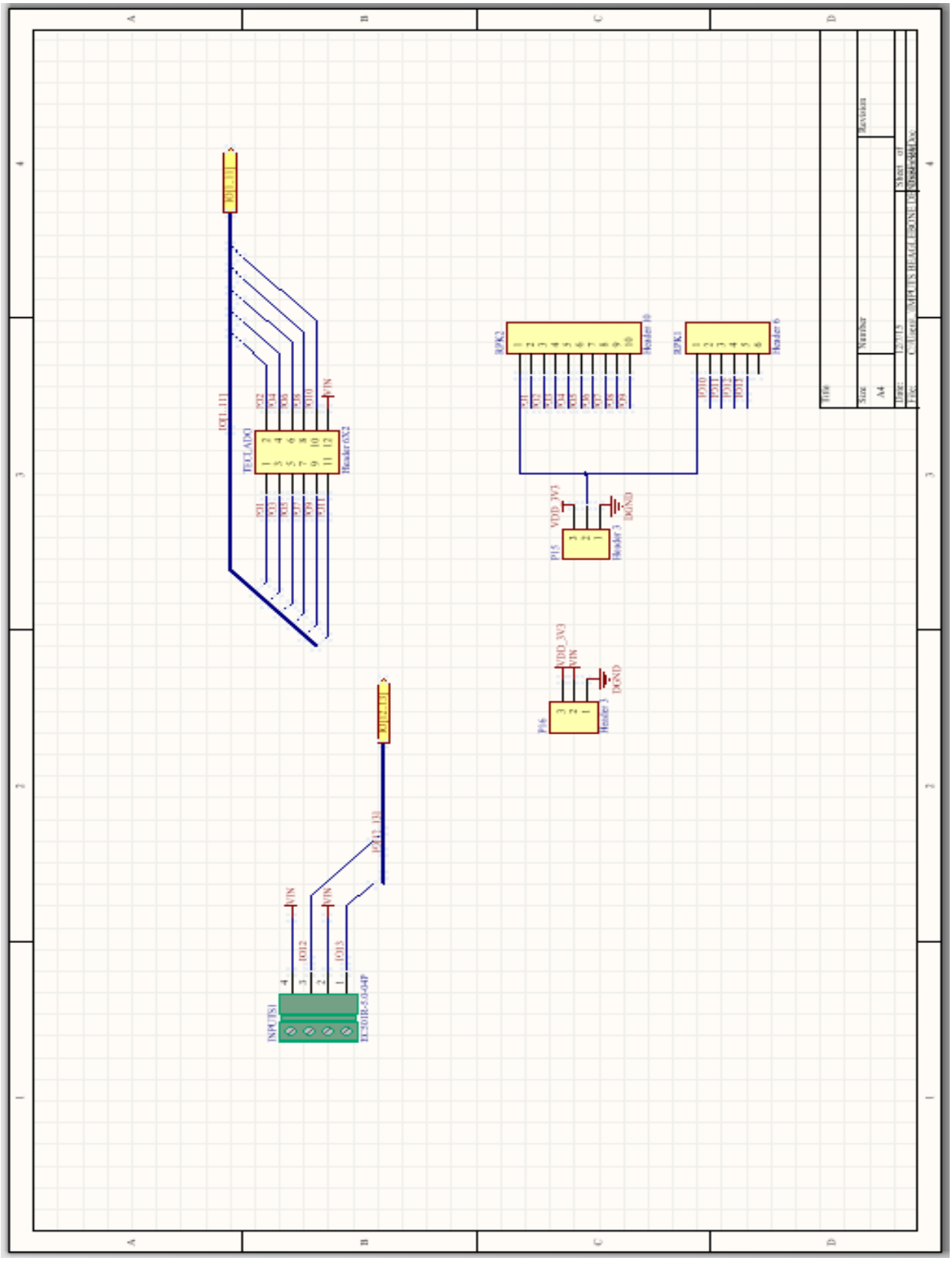

ETHERNET CONNECTOR
P9
LP2001TBBNL
ESD_RING
LAN8710A
U14
QFN32_5X5MM_EP3P3MM
beagleboard.org
Title
BeagleBone Black Ethernet
Size
Document Number
Rev
A5A
B
450-5500-001
Date: Thursday, March 21, 2013
Sheet 9 of 11

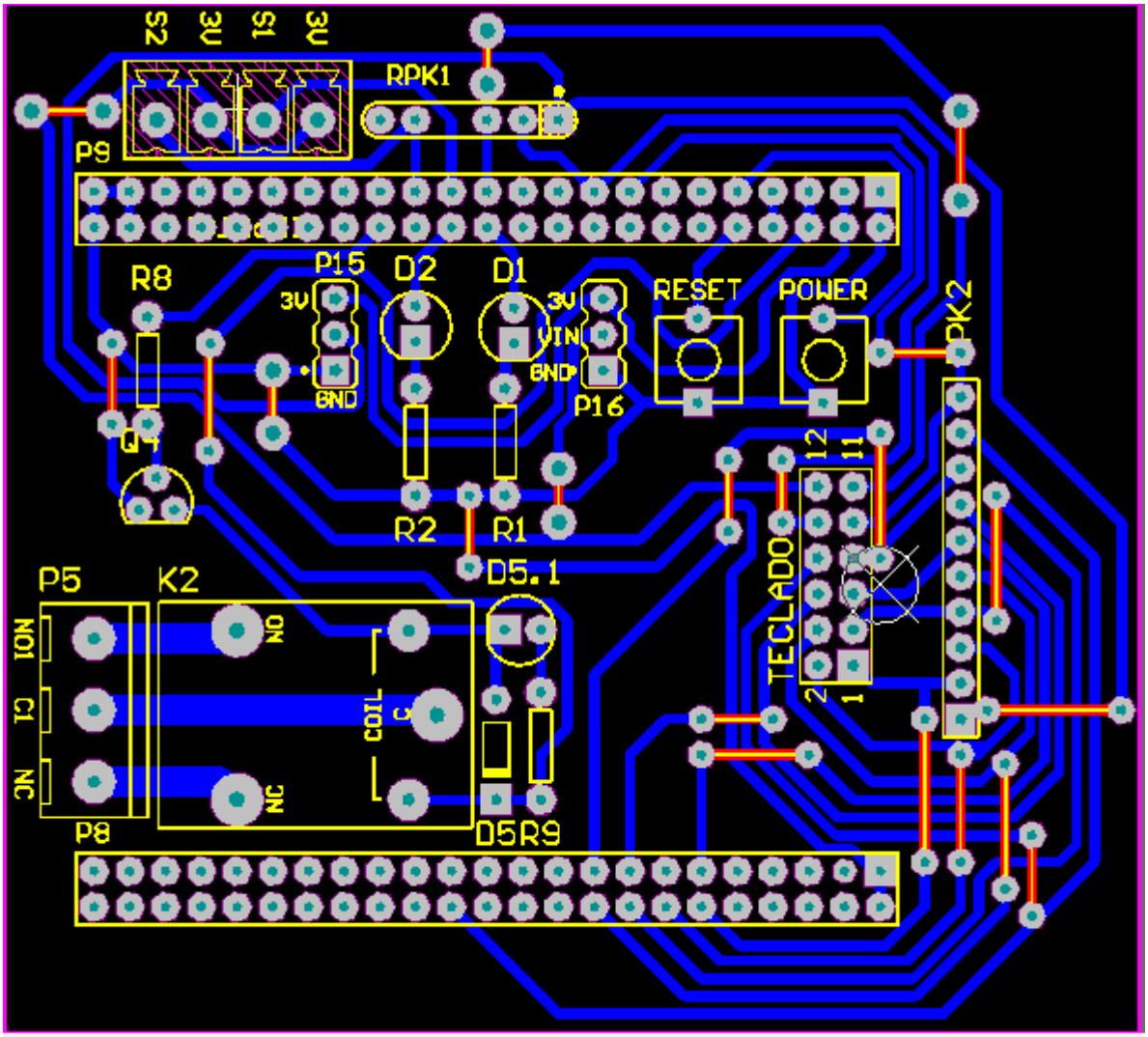

beagleboard.org
Title BeagleBone Black HDMI Interface
Size Document Number Rev
B 450-5500-001 A5A
Date: Thursday, March 21, 2013 Sheet 10 of 11
U11
TDA19988
P6
101182A1-001RLF
D7
D8
IP4283CZ10-TT
RB751V40,115
DGND
DVI +5V
SYS_5V
VDD_1V8
VDD_3V3B
HDMI_1V8
HDMI_DSCL
HDMI_DSDA
HDMI_HPD
HDMI_CEC
HDMI_CEC_D
HDMI_SWING
HDMI_INT
HDMI_TX2-
HDMI_TX2+
HDMI_TX1-
HDMI_TX1+
HDMI_TX0-
HDMI_TX0+
HDMI_TXC+
HDMI_TXC-
PTC_RXEF010
FB5
150OHM600mA
R147
15K1%
R146
15K1%
R158
21K1%
R148
10K1%
R149
0.1%
R156
10K1%
4.6.1LCD_DATA11
4.6.1LCD_DATA12
4.6.1LCD_DATA13
4.6.1LCD_DATA14
4.6.1LCD_DATA15
4.6.1LCD_DATA5
4.6.1LCD_DATA6
4.6.1LCD_DATA7
4.6.1LCD_DATA8
4.6.1LCD_DATA9
4.6.1LCD_DATA10
4.6.1LCD_DATA0
4.6.1LCD_DATA1
4.6.1LCD_DATA2
4.6.1LCD_DATA3
4.6.1LCD_DATA4
4.0CD_VSYNC
4.0CD_HSYNC
4.0CD_DE
4.0CD_PCLK
RED
GREEN
BLUE
2.4I1I2C_SDA
2.4I1I2C_SCL
HDMI_INT
SPI1_SCLK
SPI1_CS0
SPI1_D0
12MHZ


EXPANSION HEADER
EXPANSION HEADER
uSD CONNECTOR
Board ID
CAUTION: USED ON BOARD
CAUTION: USED ON BOARD
beagleboard.org
Title
BeagleBone Black Expansion Headers, uSD and EEPROM
Size
Document Number
Rev
B
450-5500-001
A5A
Date: Monday, April 01, 2013
Sheet 11 of 11

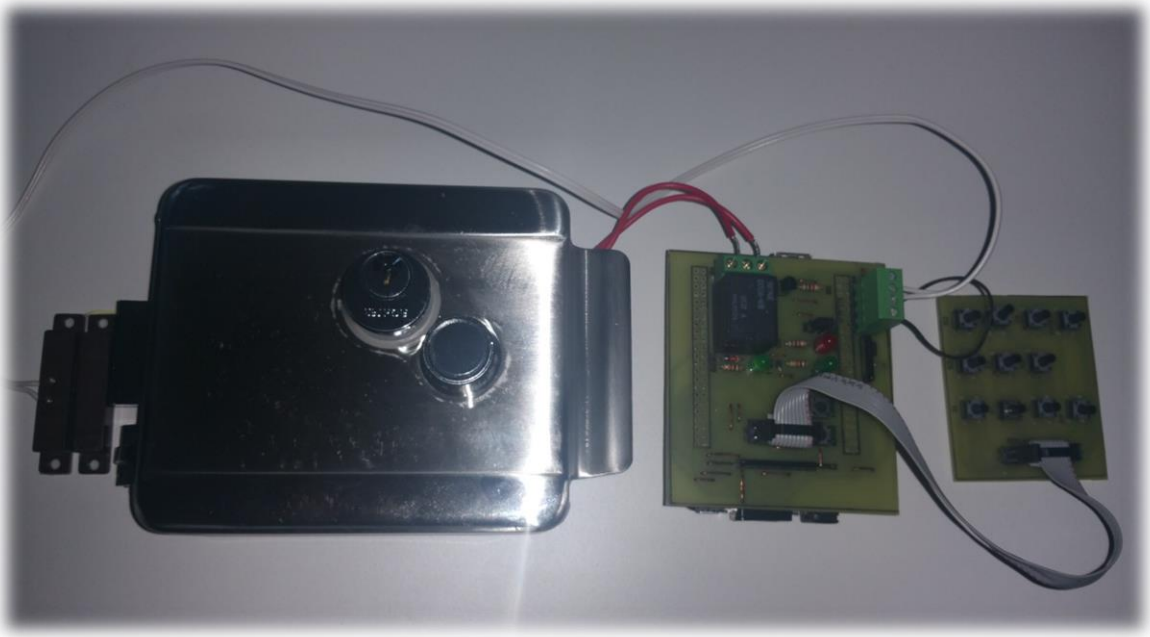
Anexo 4: DISEÑO COMPLETO CIRCUITO DE INTERFAZ

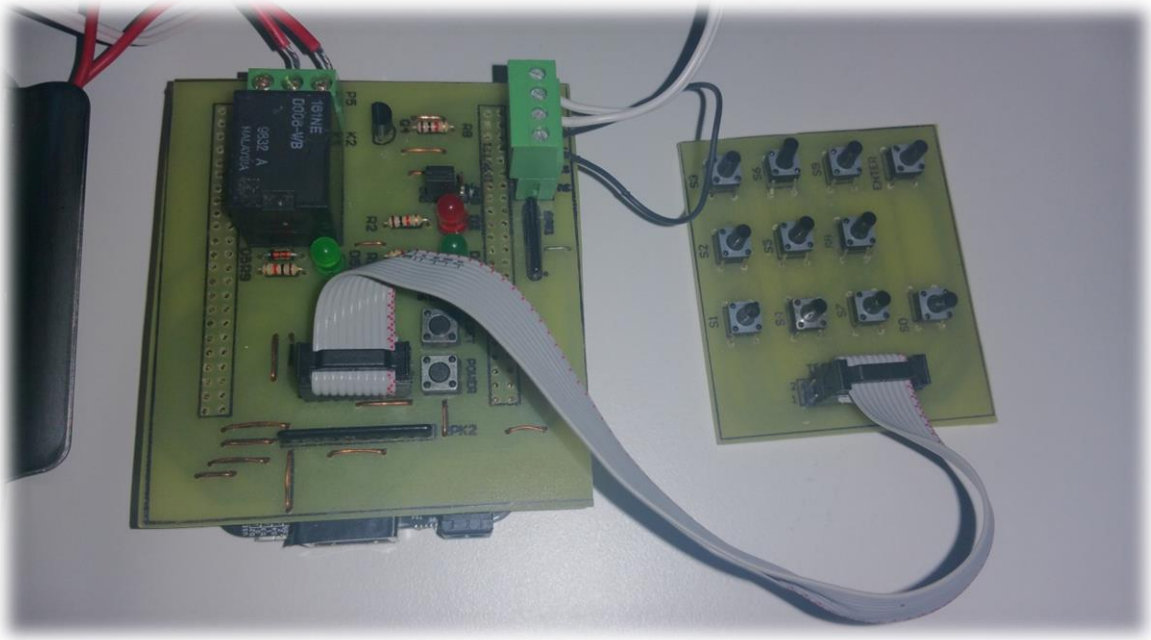

INPUT
MPU TS [BEAGLEBONE DENISSE.SchDoc
IO[1..11]
IO[12..13]
D1
RED
R1
1K
LED1
D2
GREEN
R2
1K
LED2
P8
Header 24X2
P9
Header 24X2
P3
Header 3
POWER
BOTON
RESET
BOTON
PWR_BUT
SYS_RESETn
K2
D5
D5.1
Q4
R8
1K
R9
1K
OUTPUT
VDD_5V
DGND
Title
Size
A4
Number
Revision
Date
12/7/13
Sheet of
File
C:\Users\...\BEAGLEBONE DENISSE.SchDoc
Drawn By:

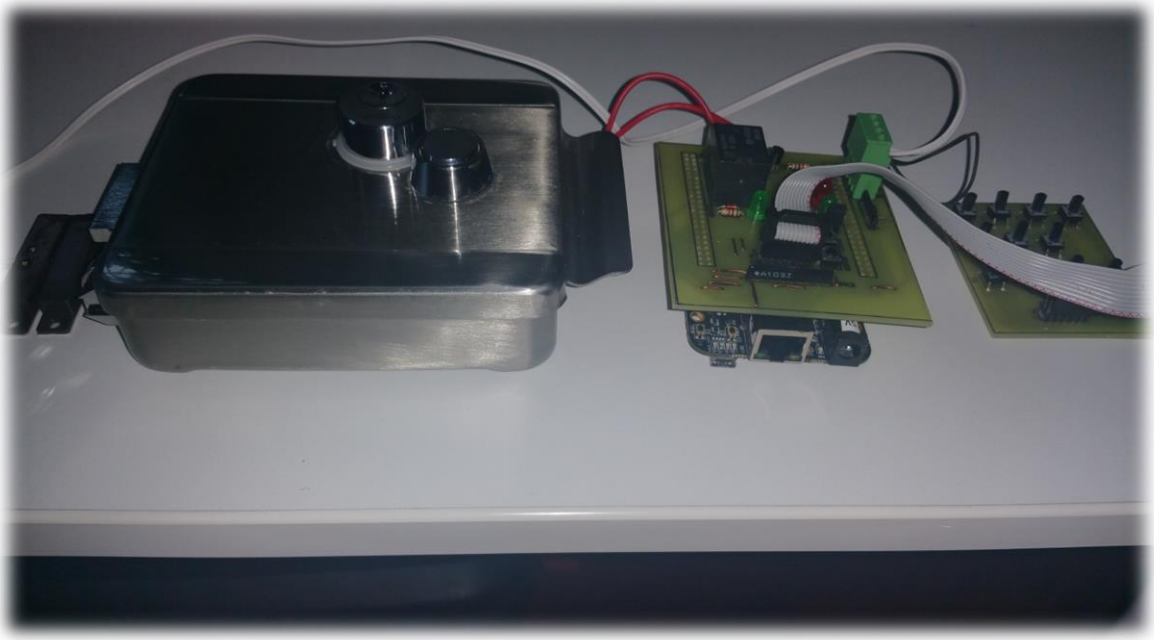


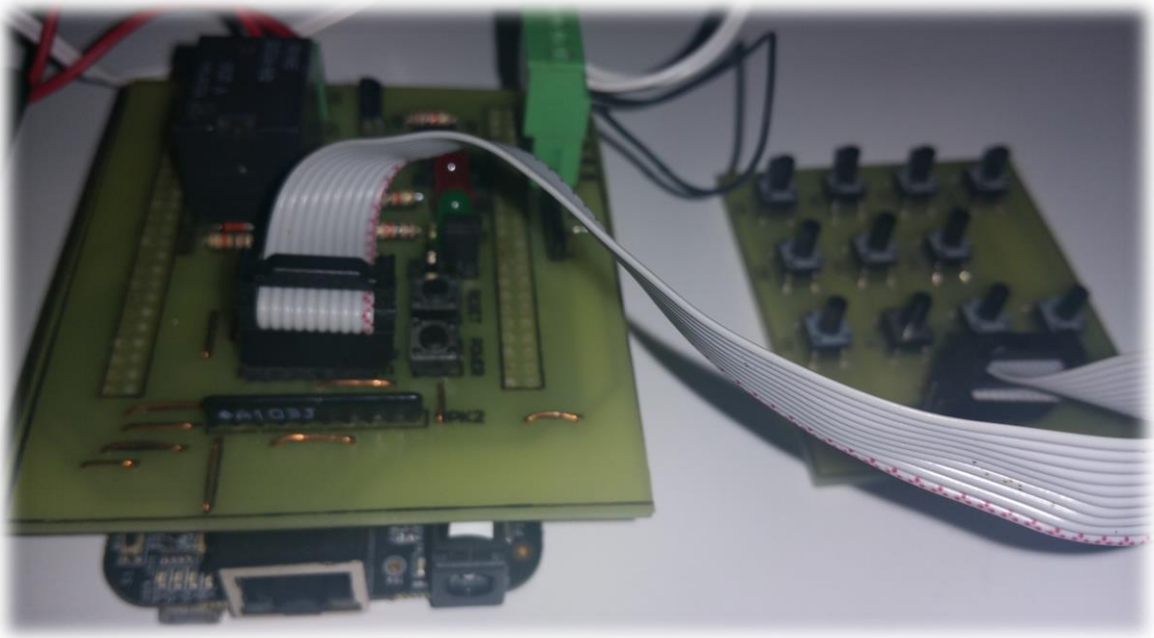

S2
3V
S1
3V
RPK1
P9
R8
P15
3V
GND
D2
D1
3V
VIN
GND
P16
RESET
POWER
PK2
Q1
R2
R1
D5.1
P5
K2
NO1
C1
NC1
NO
COIL
C
NC
P8
D5R9
TECLADO
12
11
2
1

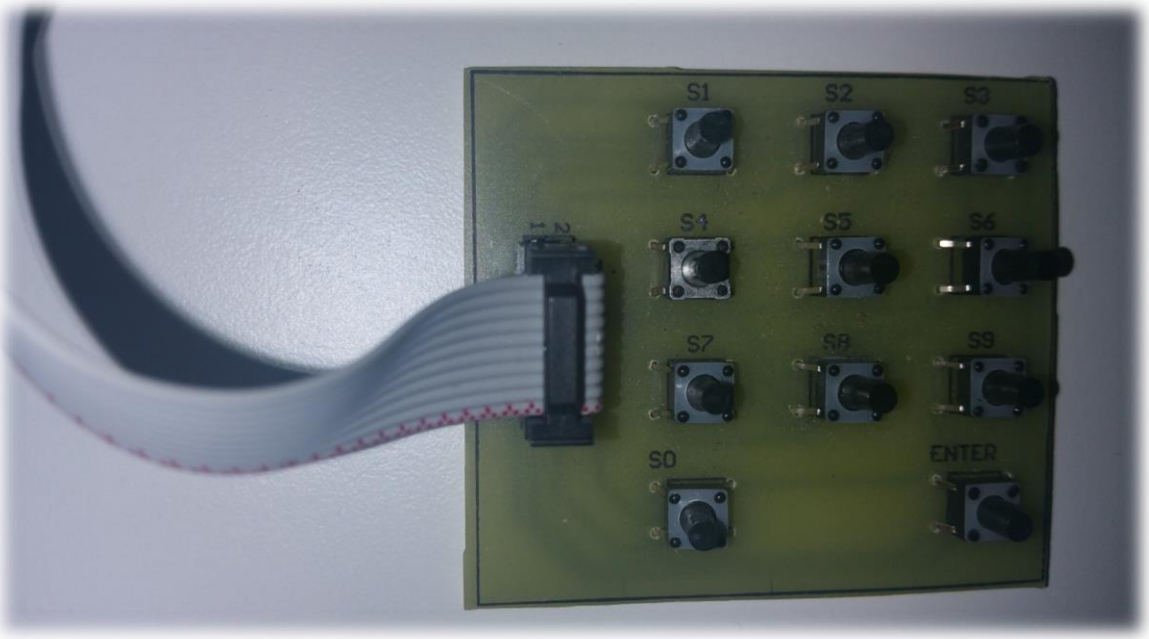
Anexo 5: SISTEMA COMPLETO Y ARMADO DEL BEAGLEBONE BLACK

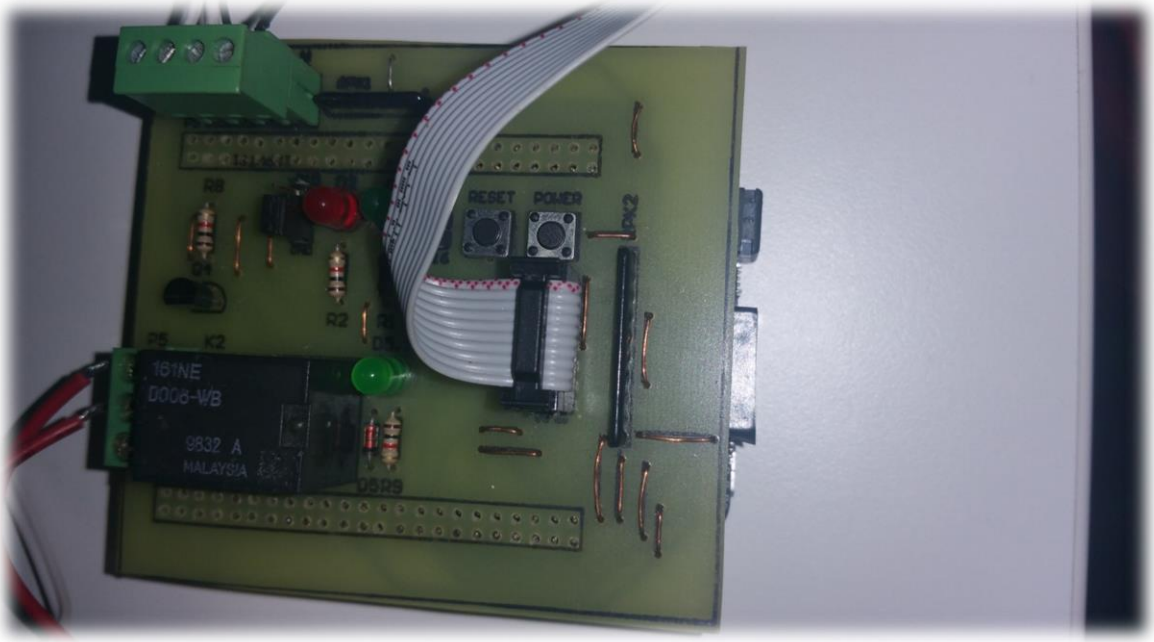
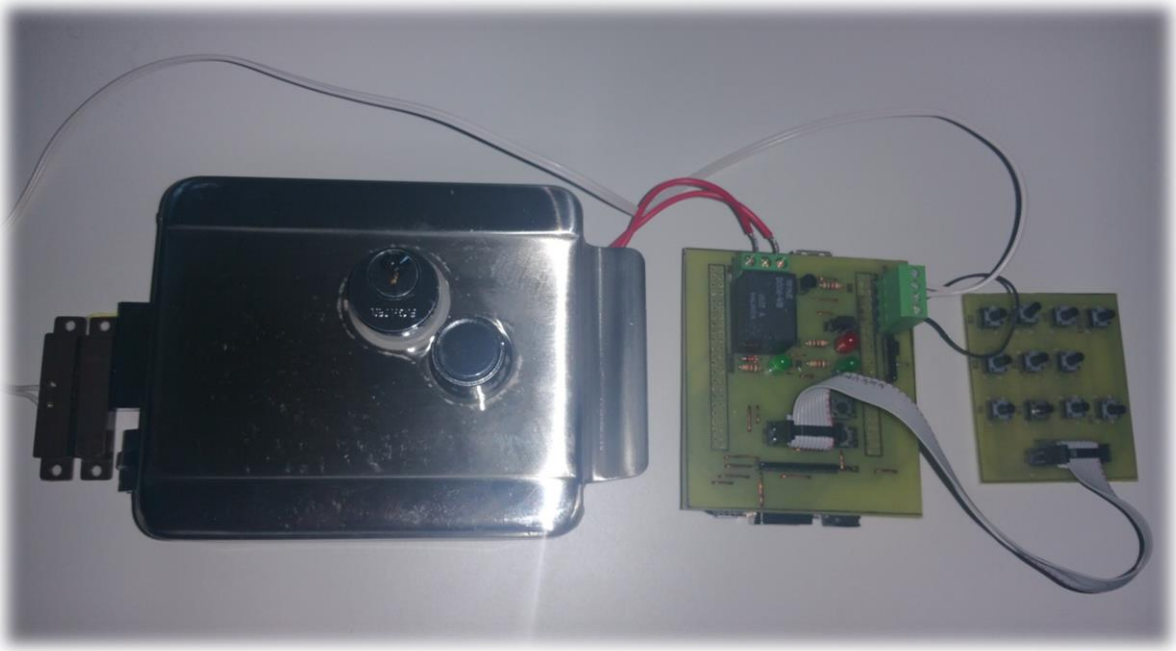









S1
S2
S3
S4
S5
S6
S7
S8
S9
S0
ENTER


RESET
POWER
161NE
D006-WB
9832 A
MALAYSIA