



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Sistema embebido para recibir
alertas de riesgo codificadas
con el protocolo SAME**

TESIS

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Manuel Rodrigo Rábade García

DIRECTORA DE TESIS

Dra. Rocío Alejandra Aldeco Pérez



Ciudad Universitaria, Cd. Mx., 2023

Dedicatoria

Para mis padres, Martha Leticia y Manuel Luis.

Agradezco su infinito apoyo e incansable entrega. Este logro nunca hubiera sido posible sin su constante esfuerzo y sacrificio.

Con todo mi cariño y gratitud les dedico este trabajo de grado, como un testimonio de su influencia en mi vida y formación. Su legado de esfuerzo, compromiso y amor perdurará en cada éxito que alcance en el futuro.

Eternamente agradecido,

Manuel Rodrigo.

Agradecimientos

A Karla Ballesteros, mi compañera de vida y apoyo incondicional. Por su amor que ha sido mi fuente de fortaleza a lo largo de este proceso. Su trayectoria académica ha iluminado constantemente mi camino, inspirándome a alcanzar mis metas con determinación y pasión.

A la Dra. Rocío Aldeco-Pérez, mi directora de tesis. Por su excepcional orientación durante el desarrollo de este trabajo. Su colaboración y guía experta fueron fundamentales en la culminación exitosa de este proyecto.

Al M.C. Alejandro Velazquez Mena. Por su constante apoyo, valiosa orientación y su destacada labor como gran profesor durante mi trayectoria académica. Su ayuda e inspiración han sido fundamentales para alcanzar mis metas.

A los distinguidos profesores de la Facultad de Ingeniería. Su compromiso con nuestra educación y su profundo saber han enriquecido mi formación profesional y humana de manera invaluable. Agradezco su constante apoyo y guía durante mi trayectoria académica.

A nuestra querida Universidad. Por proporcionarme un entorno que ha impulsado mi crecimiento personal y académico, alentándome a superarme constantemente y a contribuir de manera significativa a un mundo mejor.

A todos los involucrados en la implementación del Sistema de Alerta Sísmica Mexicano. Por su arduo trabajo y dedicación en la protección de la vida y el patrimonio de la comunidad. Su labor en el desarrollo y mantenimiento de este sistema vital ha contribuido significativamente a la seguridad y bienestar de todos.

Con profunda gratitud,

Manuel Rodrigo Rábade García.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Manuel Rodrigo Rábade García.

Ciudad Universitaria, Cd. Mx., 2023

Resumen

El protocolo SAME (del inglés *Specific Area Message Encoding* o Codificación de Mensaje de Área Específica) es utilizado para codificar las alertas de riesgo transmitidas por el Sistema de Alerta Sísmica Mexicano, entre otras organizaciones.

Por otra parte, un sistema embebido es una forma de computadora diseñada para llevar a cabo funciones específicas en el control y la operación de equipos en una amplia variedad de aplicaciones, desde electrodomésticos y automóviles hasta dispositivos médicos e industriales.

Este trabajo de tesis tiene como objetivo analizar el protocolo SAME y aplicar el diseño de sistemas embebidos en la creación de un receptor de alertas de riesgo. En este proyecto se prioriza la interoperabilidad, adaptabilidad, diseño abierto y coste moderado, en contraste con las limitaciones presentes en los receptores de alerta disponibles en el mercado. El propósito último de esta investigación es mejorar el aprovechamiento y la cobertura de los sistemas de alerta existentes.

El marco teórico se compone de una investigación del protocolo SAME y un estudio de los sistemas embebidos, incluyendo su metodología de diseño. Para el desarrollo de este proyecto, primero se presenta el proceso de diseño, construcción y pruebas de un receptor de alertas. En segundo lugar, basándose en un análisis de los resultados obtenidos durante las pruebas, se presentan una serie de mejoras al sistema desarrollado. Por último, se exponen las conclusiones y las posibles direcciones futuras para esta investigación.

Como resultado de este trabajo, se desarrolló un sistema embebido capaz de recibir alertas de riesgo transmitidas mediante el protocolo SAME y difundirlas a través de medios visuales, auditivos o electrónicos. Además, se logra documentar las bases técnicas para el desarrollo de productos o servicios que amplíen el aprovechamiento y la cobertura de los sistemas de alerta de riesgos.

Palabras clave

Protocolo SAME, sistemas embebidos, receptor de alertas, alerta sismica.

Índice general

Introducción	15
Capítulo 1. Protocolo SAME	18
1.1 Historia	18
1.2 Radio del Clima NOAA	20
1.3 Especificación del protocolo SAME	22
1.3.1 Formato digital	22
1.3.2 Elementos	24
1.3.2.1 Encabezado	25
1.3.2.2 Tono de alarma	28
1.3.2.3 Mensaje de voz	28
1.3.2.4 Fin de mensaje	28
1.3.3 Ejemplos de mensajes	29
1.4 Receptores	32
Capítulo 2. Sistemas Embebidos	36
2.1 Definición	36
2.1.1 Aplicaciones	37
2.2 Componentes	39
2.2.1 Microprocesador	39
2.2.2 Memoria	41
2.2.3 Perifericos	42
2.2.4 Software	45
2.2.5 Algoritmos	48
2.3 Diseño	48
2.3.1 Especificación de producto	50
2.3.2 Distribución de software y hardware	53
2.3.3 Iteración y refinamiento de la distribución	54
2.3.4 Diseño de software y hardware	55
2.3.5 Pruebas de producto y lanzamiento	60
2.3.6 Mantenimiento y actualizaciones	62

Capítulo 3. Diseño del receptor de alertas	63
3.1 Especificación de producto	63
3.1.1 Requerimientos operacionales	63
3.1.2 Plan de mantenimiento	65
3.1.3 Análisis de factibilidad	65
3.1.4 Plan de administración de la ingeniería del sistema	68
3.2 Distribución de software y hardware	70
3.2.1 Hardware	71
3.2.2 Software	73
3.3 Iteración y refinamiento de la distribución	74
3.4 Diseño detallado de software y hardware	78
3.4.1 Hardware	79
3.4.2 Software	85
3.5 Integración de software y hardware	91
3.5.1 Inspección de hardware	92
3.5.2 Combinación de software y hardware	93
3.6 Pruebas de producto y lanzamiento	94
3.6.1 Pruebas estáticas de caja negra	94
3.6.2 Pruebas estáticas de caja blanca	98
3.6.3 Pruebas dinámicas de caja negra	98
3.6.4 Pruebas dinámicas de caja blanca	99
3.7 Mantenimiento y actualizaciones	103
Capítulo 4. Resultados y mejoras	107
4.1 Resultados de las pruebas	107
4.2 Datos recabados	108
4.3 Mensajes faltantes	109
4.3.1 Falta de energía	112
4.3.2 Mensajes ignorados	114
4.3.3 Sin razón aparente	115
4.4 Mensajes sin audio	115
4.5 Mejoras propuestas	117
4.5.1 Batería de respaldo para el banco de pruebas	117
4.5.2 Procesar encabezados recibidos solo dos veces	118
4.5.3 Reproducir el audio de alerta sísmica	119
4.6 Prueba en simulacro	119
4.7 Ejemplos de configuración	122

Conclusiones	127
Apéndice A. Código de eventos SAME	130
Apéndice B. Diseño de hardware	134
B.1 Configuración de conectores	135
B.2 Diseño de la placa electrónica	138
B.3 Listado de componentes	144
B.4 Fabricación de prototipos	146
Apéndice C. Diseño de software	149
C.1 Diagramas UML	150
C.2 Protocolo de comunicación	157
Apéndice D. Integración de software y hardware	164
D.1 Instalación del bootloader	165
D.2 Instalación de software	168
D.3 Pruebas unitarias	171
D.4 Pruebas de integración	178
Apéndice E. Pruebas de producto y lanzamiento	182
Apéndice F. Integración del receptor de alertas	187
Bibliografía	190

Índice de acrónimos y abreviaturas

AC	Del inglés <i>Alternating Current</i> o Corriente alterna
AFSK	Del inglés <i>Audio Frequency Shift Keying</i> o Modulación de audio por desplazamiento de frecuencia
AM	Amplitud modulada
ANSI	Del inglés <i>American National Standards Institute</i> o Instituto nacional estadounidense de estándares
ASCII	Del inglés <i>American Standard Code for Information Interchange</i> o Código estándar estadounidense para el intercambio de información
BDM	Del inglés <i>Background Debug Mode</i> o Modo depuración en segundo plano
CAN	Del inglés <i>Controller Area Network</i> o Red de área de controlador
CIRES	Centro de Instrumentación y Registro Sísmico
CMM	Del inglés <i>Capability Maturity Model</i> o Modelo de madurez de capacidades
CMOS	Del inglés <i>Complementary Metal-Oxide-Semiconductor</i> o semiconductor complementario de óxido metálico
CPU	Del inglés <i>Central Processing Unit</i> o Unidad central de procesamiento
CTA	Del inglés <i>Consumer Technology Association</i> o Asociación de Consumidores de Tecnología
DC	Del inglés <i>Direct Current</i> o Corriente directa
DTMF	Del inglés <i>Dual-Tone Multi-Frequency</i> o Marcación por tonos
E/S	Entrada/Salida

EAS	Del inglés <i>Emergency Alert System</i> o Sistema de alerta de emergencias
ECAD	Del inglés <i>Electronic Computer Aided Design</i> o Diseño electrónico asistido por computadora
EDA	Del inglés <i>Electronic Design Automation</i> o Diseño electrónico automatizado
EOM	Del inglés <i>End Of Message</i> o Fin de mensaje
EPROM	Del inglés <i>Erasable Programmable Read-Only Memory</i> o Memoria de solo lectura, programable y borrable
ESP	Del inglés <i>Electronic Stability Program</i> o Control de estabilidad electrónico
ESSA	Del inglés <i>Environmental Science Services Administration</i> o Servicio de Ciencias Ambientales
EUA	Estados Unidos de América
FCC	Del inglés <i>Federal Communications Commission</i> o Comisión Federal de Comunicaciones
FIPS	Del inglés <i>Federal Information Processing Standard</i> o Estándares Federales de Procesamiento de la Información
FM	Frecuencia Modulada
FSK	Del inglés <i>Frequency Shift Keying</i> o Modulación por desplazamiento de frecuencia
I2C	Del inglés <i>Inter-Integrated Circuit</i> o Circuito Inter-Integrado
ICSP	Del inglés <i>In-Circuit Serial Programming</i> o Programación Serie en Circuito
IDE	Del inglés <i>Integrated Development Environment</i> o Entorno de desarrollo integrado

IEEE	Del inglés <i>Institute of Electrical and Electronics Engineers</i> o Instituto de ingenieros eléctricos y electrónicos
ISO	Del inglés <i>International Organization for Standardization</i> o Organización Internacional de Normalización
JTAG	Del inglés <i>Joint Test Action Group</i> o Grupo de acción conjunta de pruebas
KB	Kilobyte
L/MR	Del inglés <i>Low/Medium Frequency</i> o Frecuencia baja/media
LCD	Del inglés <i>Liquid Crystal Display</i> o Pantalla de cristal líquido
LED	Del inglés <i>Light-Emitting Diode</i> o Diodo emisor de luz
LIN	Del inglés <i>Local Interconnected Network</i> o Red local interconectada
NIC	Del inglés <i>National Information Center</i> o Centro de Información Nacional
NOAA	Del inglés <i>National Oceanic and Atmospheric Administration</i> o Administración Nacional Oceánica y Atmosférica
NWEMs	Del inglés <i>Non-Weather Emergency Messages</i> o Alertas sin relación a la meteorología
NWR	Del inglés <i>NOAA Weather Radio</i> o Radio del clima NOAA
NWS	Del inglés <i>National Weather Service</i> o Servicio Meteorológico Nacional
PCBA	Del inglés <i>Printed Circuit Board Assembly</i> o Placa ensamblada de circuito impreso
PCB	Del inglés <i>Printed Circuit Board</i> o Placa de circuito impreso
PCM	Del inglés <i>Powertrain Control Module</i> o Módulo de control del tren motriz

PDA	Del inglés <i>Personal Digital Assistant</i> o Asistente digital personal
PEP	Del inglés <i>Primary Entry Point</i> o Punto de entrada principal
PTU	Del inglés <i>Peripheral Touch Controller</i> o Controlador de periférico táctil
PWM	Del inglés <i>Pulse-Width Modulation</i> o Modulación por ancho de Pplsos
RAM	Del inglés <i>Random Access Memory</i> o Memoria de acceso aleatorio
RF	Radiofrecuencia
RMT	Del inglés <i>Required Monthly Test</i> o Test mensual obligatorio
ROM	Del inglés <i>Read-Only Memory</i> o Memoria de solo lectura
RTOS	Del inglés <i>Real-Time Operating System</i> o Sistema operativo en tiempo real
RWT	Del inglés <i>Required Weekly Test</i> o Test semanal obligatorio
SAME	Del inglés <i>Specific Area Message Encoding</i> o Codificación de Mensaje de Área Específica
SASMEX-CDMX	Sistema de Alerta Sísmica para la Ciudad de México
SASMEX	Sistema de Alerta Sísmica Mexicano
SASO	Sistema de Alerta Sísmica para la Ciudad de Oaxaca
SAS	Sistema de Alerta Sísmica
SD	Del inglés <i>Secure Digital</i> o Estándar SD
SDSAS	Sistemas de Difusión Secundaria para el Alertamiento Sísmico
SEMP	Del inglés <i>System Engineering Management Plan</i> o Plan de administración de la ingeniería del sistema
SERCOM	Del inglés <i>Serial Communication Module</i> o Módulo de comunicaciones seriales

SGIRPC	Secretaría de Gestión Integral de Riesgos y Protección Civil
SPI	Del inglés <i>Serial Peripheral Interface</i> o Interfaz de periféricos en serie
TV	Televisión
UML	Del inglés <i>Unified Modeling Language</i> o Lenguaje unificado de modelado
USART	Del inglés <i>Universal Synchronous/Asynchronous Receiver/Transceiver</i> o Receptor/transceptor universal síncrono/asíncrono
USB	Del inglés <i>Universal Serial Bus</i> o Bus universal en serie
VHF	Del inglés <i>Very High Frequency</i> o Frecuencia muy alta
W-FM	Del inglés <i>Wide-FM</i> o FM de Banda ancha
WAT	Del inglés <i>Warning Alarm Tone</i> o Tono de alarma de advertencia
WB	Del inglés <i>Weather Band</i> o Banda del clima

Introducción

México está situado en una de las regiones del mundo con más actividad sísmica debido a la interacción entre las placas de Norteamérica, Cocos, Pacífico, Rivera y del Caribe.

Aunque las zonas de mayor sismicidad se concentran en la costa occidental del país, la Ciudad de México debe ser considerada como una zona sísmica debido a las condiciones del subsuelo del Valle de México. (Servicio Geológico Mexicano, 2017)

En respuesta a este riesgo, el Centro de Instrumentación y Registro Sísmico (CIRES) con el auspicio de la Secretaría de Obras del Gobierno del Distrito Federal desarrolló en 1989 el Sistema de Alerta Sísmica (SAS) para advertir el efecto de los sismos de gran magnitud provenientes de Guerrero. (CIRES, 2005)

Con el fin de compartir el conocimiento oportuno del peligro sísmico se conformó en 2005 el Sistema de Alerta Sísmica Mexicano (SASMEX) por iniciativa de los gobiernos de Oaxaca, Ciudad de México y la Secretaría de Gobernación.

El SASMEX se conforma por el Sistema de Alerta Sísmica para la Ciudad de México (SASMEX-CDMX), que se encuentra en operación desde 1991 y el Sistema de Alerta Sísmica para la Ciudad de Oaxaca (SASO) brinda servicio desde 2003.

Gracias al apoyo de la autoridad del Centro Histórico de la Ciudad de México, SASMEX logró en 2008 la instalación de transmisores VHF para difundir la alerta sísmica en la Banda del Clima (en inglés *Weather Band*) empleando el protocolo para la codificación de mensajes de área específica, mejor conocido como protocolo SAME (del inglés *Specific Area Message Encoding* o Codificación de Mensaje de Área Específica).

El protocolo SAME fue desarrollado en Estados Unidos de América por la Administración Nacional Oceánica y Atmosférica (en inglés *National Oceanic and Atmospheric Administration*, NOAA) para codificar las alertas transmitidas por la Radio del Clima NOAA (en inglés *NOAA Weather Radio*, NWR). (Weather Radios Direct, 2011)

Además, el protocolo SAME ha sido adoptado por otros sistemas de alertamiento como el Sistema de Alerta de Emergencias de EUA (en inglés *Emergency Alert System*, EAS), Meteoradio Canadá y el SASMEX.

Por otra parte, uno de los avances más sorprendentes de las últimas décadas ha sido el ascenso de las computadoras a una posición predominante en la vida humana. Actualmente hay más computadoras en casas y oficinas que personas viviendo o trabajando en ellas, sin embargo, muchas de estas computadoras no son reconocidas como tales.

Así tenemos que, un sistema embebido es una computadora diseñada para desempeñar una función específica dentro de un sistema mecánico, eléctrico o de cualquier otro tipo. Un buen ejemplo de donde se puede encontrar un sistema embebido es un horno de microondas. Aunque es usado por miles de millones de personas en el mundo, pocos conocen que hay un procesador y software involucrado en la preparación de su comida o cena. (Massa y Barr, 2006)

En contraste con los sistemas embebidos, las computadoras personales son diseñadas para que después de su fabricación se elija su función dentro de una muy amplia gama de opciones. Por esta razón muchas personas usan el término computadora de propósito general para clarificar su distinción con los sistemas embebidos. (ITS, 1996)

Es importante mencionar que una computadora de propósito general es una interfaz para muchos sistemas embebidos. Por ejemplo, un módem es un sistema embebido con su propio procesador y software cuyo propósito específico es enviar y recibir datos digitales mediante una línea telefónica. (Ganssle, 2008)

Aunque la existencia de un sistema embebido puede pasar desapercibida por su usuario, en muchos casos también es posible construir un dispositivo funcionalmente equivalente que no contenga hardware ni software. Sin embargo, un sistema embebido ofrece mayor flexibilidad y generalmente es más fácil, barato y eficiente que un sistema similar sin él.

Un ejemplo de dispositivo mejorado por la incorporación de sistemas embebidos son los automóviles. Gracias a los sistemas embebidos, un automóvil moderno es más económico, emite menos contaminación, es más seguro y brinda mayor comodidad a sus usuarios. (Cox, 2009; RACV, 2004)

También, existen productos imposibles de diseñar sin los sistemas embebidos, un ejemplo son las lámparas eléctricas que permiten su control desde un teléfono inteligente. (Keeping, 2016)

Así estos ejemplos muestran que es posible diseñar un sistema embebido para construir un receptor de alertas de riesgo transmitidas utilizando el protocolo SAME y difundirlas por medios visuales, auditivos o electrónicos.

Aunque al día de hoy existen receptores de alertas de riesgo, estos traen consigo las siguientes limitantes.

1. No son interoperables. La incapacidad de conectarlos con otros sistemas lo que dificulta su integración en sistemas de alertamiento más amplios.
2. No son adaptables. Son dispositivos totalmente integrados en los que no se pueden reemplazar sus componentes para adaptarlos a situaciones específicas.
3. Son de código y diseño cerrado. La imposibilidad de reproducirlos libremente y la falta de información detallada sobre su funcionamiento interno puede dificultar el mantenimiento y la mejora continua de estos sistemas.
4. Son de alto costo. El costo relativamente elevado y su rigidez puede dificultar su adopción en regiones con recursos limitados.

En vista de esto, el objetivo de este trabajo es analizar el protocolo SAME y aplicar la metodología del diseño de sistemas embebidos en la construcción de un receptor de alertas de riesgo que permita incrementar el aprovechamiento y cobertura de los sistemas de alertamiento.

Los objetivos específicos son:

1. Diseñar un sistema receptor de alertas de riesgo codificadas con el protocolo SAME.
2. Fabricar un prototipo del receptor diseñado y evaluar su funcionamiento.
3. Documentar el proceso anterior para el desarrollo de productos o servicios que incrementen el aprovechamiento de los sistemas de alertamiento de riesgos.

Este documento está organizado de la siguiente manera. En el Capítulo 1 y el Capítulo 2 se establece el marco teórico con una investigación del protocolo SAME y el estudio de los sistemas embebidos incluyendo su metodología de diseño. A lo largo del Capítulo 3 se presenta el diseño y construcción de un receptor de alertas de riesgo aplicando la metodología descrita en el Capítulo 2. En el Capítulo 4 se analizan y discuten los resultados de las pruebas del prototipo construido en el Capítulo 3 y se presentan una serie de mejoras basadas en el análisis de los resultados obtenidos. Finalmente, en el Capítulo 5 se presentan las conclusiones y trabajo futuro de esta tesis.

Capítulo 1. Protocolo SAME

El protocolo SAME, del inglés *Specific Area Message Encoding* o en español Codificación de Mensaje de Área Específica, es un formato digital para transmitir información sobre el tipo y lugar de amenazas a la seguridad de aquellos cercanos a riesgos como clima severo u otros peligros. Su mayor ventaja es que mejora significativamente la selección y distribución automática de alertas de riesgo. (NOAA, 2011, p. 3)

El objetivo del protocolo SAME es posibilitar la selección de alertas para usuarios de la Radio del Clima NOAA (NWR) y participantes del Sistema de Alerta de Emergencias de EUA (EAS) a medida que la precisión y número de advertencias públicas aumentan en las áreas cubiertas por la NWR.

En este capítulo se estudiará:

- La historia de los sistemas de alertamiento en EUA y el protocolo SAME.
- El medio principal para la transmisión de alertas codificadas con el protocolo SAME: la Radio del Clima NOAA.
- Los elementos y formato de mensajes codificados con el protocolo SAME.
- Las opciones disponibles para sintonizar la Banda del Clima e interpretar información codificada con el protocolo SAME.

1.1 Historia

La Oficina de Meteorología de los EUA inició en 1960 la radiodifusión de información meteorológica marina por medio de mensajes de voz en dos estaciones de radio VHF en Chicago y Nueva York. (Nelson, 2002, p. 56)

Estas dos estaciones fueron originalmente construidas para la radiodifusión de información meteorológica aeronáutica y se volvieron obsoletas por la migración de la radiodifusión aeronáutica a la banda L/MR (*Low or Medium Frequency*). El éxito de este experimento llevó a la expansión de la red de radiodifusión al público en general durante los años 1960 y principios de los 1970.

En 1970, la Oficina de Meteorología se convierte en el Servicio Meteorológico Nacional (*National Weather Service*, NWS) y el Servicio de Ciencias Ambientales (*Environmental Science Services Administration*, ESSA) se convierte en la NOAA. (National Weather Service, 2015)

En Octubre de 1971, el NWS anuncia:

“Como característica adicional, la radio meteorológica del NWS puede encender receptores de radio especialmente diseñados por medio de una señal de tono. Esta señal se transmite a 1050 Hz con una duración de tres a cinco segundos antes de anuncios de condiciones climáticas peligrosas. La señal de tono alerta a las escuelas, hospitales, iglesias y otros lugares de reunión, unidades de servicios públicos, fuerzas de emergencia, y medios de comunicación para estar preparados para mensajes meteorológicos críticamente importantes.” (NWS, 1971 citado por Nelson, 2002, p. 62)

El tono de 1050 Hz se llegaría a conocer como el WAT (*Warning Alarm Tone*) y a pesar de su gran utilidad tenía la gran desventaja que era impráctico e inaceptable para los medios de comunicación retransmitir automáticamente cualquier mensaje de voz precedido por él.

Asumiendo que los anuncios precedidos por el WAT fueran críticos no había forma de que el equipo de recepción determinará la correcta y completa recepción de mensajes. Además, por el largo alcance de los radios VHF del NWS, se presentaba un serio problema de usabilidad ya que era posible captar alertas falsas en el sentido de que no eran relevantes por ser destinadas a zonas muy distantes del receptor.

En 1974 se designa oficialmente como Radio del Clima NOAA (*NOAA Weather Radio*, NWR) a la red de 55 transmisores en VHF-FM de información meteorológica con los que contaba el NWS. (Weigel, 1974 citado por Nelson, 2002, p. 62)

En 1979 Lawrence Krudwig, gerente de sistemas de campo del NWS, envía a la sede central del Servicio Nacional de Meteorología una propuesta para codificar información adjunta a los mensajes del NWR con la finalidad de resolver las limitaciones del WAT. (Nelson, 2002, p. 78)

Krudwig recibe fondos del NWS en 1983 para llevar a cabo un proyecto exploratorio de su propuesta que culminó en 1985 con un programa piloto en Kansas City utilizando tonos analógicos en un sistema multifrecuencia (*Dual-Tone Multi-Frequency*, DTMF) para transmitir información codificada con los mensajes de NWR a dos instalaciones piloto pertenecientes a transmisores de TV.

A finales de los años 1980 Krudwig comienza las pruebas con un nuevo formato digital utilizando la Modulación por Desplazamiento de Frecuencia o FSK (del inglés *Frequency Shift Keying*). Este sistema evolucionó en lo que hoy es conocido como el protocolo SAME y fue formalmente adoptado en 1988 por el NWS para su implementación nacional en los EUA. (NOAA, 2011, p. 3)

El protocolo SAME fue adoptado por la Comisión Federal de Comunicaciones (*Federal Communications Commission*, FCC) en 1997 para su uso en el Sistema de Alerta de Emergencias (EAS) con el objetivo transmitir información adicional a medios de comunicación y al público en general incluyendo alertas sin relación a la meteorología (*Non-Weather Emergency Messages*, NWEMs). (NOAA, 2011, p. 3)

Además, el protocolo SAME ha sido adoptado por el Servicio Meteorológico de Canadá para su sistema Meteoradio Canadá en 2004 (Meteorological Service of Canada, 2004) y por el Sistema de Alerta Sísmica Mexicana (SASMEX) en 2008 (Centro de Instrumentación y Registro Sísmico, 2005) que es de nuestro interés en esta tesis.

1.2 Radio del Clima NOAA

La Radio del Clima NOAA (*NOAA Weather Radio All Hazards*, NWR) es un conjunto de estaciones de radio que transmiten continuamente alertas, advertencias y otra información de riesgos las 24 horas al día y los 7 días de la semana (NOAA, s.f. c). NWR es un servicio público de la NOAA compuesto por 1025 transmisores que cubren el 97% de la población de EUA. (NOAA, s.f. b)

Los transmisores de NWR tienen una potencia de entre 5 y 1000 watts con un alcance de hasta 40 millas en terrenos planos (NOAA, s.f. a). Para escuchar una estación de NWR se requiere de un receptor especial de radio capaz de sintonizar la Banda del Clima que abarca de los 162.400 MHz a los 162.550 MHz.

La Radio del Clima es una emisión de tipo 16K0F3E, es decir, emplea la modulación W-FM (*Wide-FM* o FM de Banda Ancha) con 16.0 kHz de ancho de banda y transmite un canal analógico de voz. (U.S. Department of Homeland Security, 2011). En la Tabla 1 se muestran los canales y frecuencias de esta radio.

Tabla 1. Canales y frecuencias de NWR. (United States Coast Guard, 2018)

Canal NOAA	Frecuencia	Canal de Alerta Pública
WX1	162.550 MHz	7
WX2	162.400 MHz	1
WX3	162.475 MHz	4
WX4	162.425 MHz	2
WX5	162.450 MHz	3
WX6	162.500 MHz	5
WX7	162.525 MHz	6
WX8	161.650 MHz	—
WX9	161.775 MHz	—
WX10	163.275 MHz	—

Estados Unidos, Canadá y México operan estaciones de Radio del Clima en los canales WX1 a WX7. Los canales WX8 y WX9 son utilizados en algunas zonas de Canadá para la transmisión de informes de meteorología marina. WX10 se encuentra fuera de uso y era utilizado por el NWS para coordinarse en caso de un corte de energía. (NOAA, 2015)

La numeración de los canales de Radio del Clima según el esquema NOAA (WX1, WX2, etc.) indica el orden en que se establecieron y se está volviendo menos popular ante el esquema de Alerta Pública que indica el orden de las frecuencias. Muchos receptores de NWR confunden los esquemas de numeración aplicando un etiquetado inconsistente. (United States Coast Guard, 2018)

Como en cualquier transmisión de radio, es posible recibir más de uno de los canales de Radio del Clima dependiendo de la ubicación del receptor, potencia del transmisor y características geográficas de la región. Se recomienda que la elección de canal a sintonizar se lleve a cabo por frecuencia en lugar de canal debido a los rotulados inconsistentes entre diferentes modelos de receptores.

1.3 Especificación del protocolo SAME

La especificación del protocolo SAME está contenida en la directiva “NOAA *Weather Radio (NWR) All Hazards Specific Area Message Encoding (SAME)*” publicada en febrero de 2007 por el NWS. En esta sección se tomará como referencia principal esta directiva para caracterizar los elementos y describir los formatos empleados por el protocolo SAME.

Un mensaje transmitido utilizando el protocolo SAME consiste de cuatro posibles elementos en el siguiente orden:

1. Encabezado
2. Tono de alarma o señal de atención
3. Mensaje de voz
4. EOM (*End of Message* o fin de mensaje)

Entre cada elemento de un mensaje y mensajes diferentes se transmite un segundo de audio vacío. Los elementos 1 y 4 son obligatorios en toda transmisión mientras que los elementos 2 y 3 son opcionales de acuerdo al tipo y aplicación del mensaje.

A continuación se describe el formato de los elementos digitales de un mensaje SAME y más adelante se analizan los detalles de cada elemento.

1.3.1 Formato digital

El encabezado y EOM son los elementos digitales de un mensaje SAME. Se transmiten en el canal de audio de NWR utilizando la Modulación de Audio por Desplazamiento de Frecuencia o AFSK (del inglés *Audio Frequency Shift Keying*). La transmisión se realiza a por lo menos un 80% de modulación con una desviación mínima de ± 4.0 kHz y máxima de ± 5.0 kHz.

La técnica de modulación AFSK permite representar información digital por medio de cambios en la frecuencia de un tono audible, dando como resultado una señal codificada apta para ser transmitida por radio o teléfono. La técnica AFSK se diferencia de la FSK en que la modulación se lleva a cabo en la banda base. Con la señal modulada por AFSK se modula una onda portadora de RF utilizando una técnica convencional como AM o FM para su transmisión. (Frenzel, 2012)

El mensaje de datos codificados y voz se transmite mediante NWR utilizando la preacentuación estándar para las transmisiones VHF-FM. La preacentuación tiene como objetivo reducir el ruido de fondo causado por el ancho de banda en las transmisiones W-FM. Ya que el ruido de fondo es más notable en el extremo agudo del espectro del audio transmitido por VHF-FM, la preacentuación aumenta el nivel de las frecuencias de agudos en el transmisor con la intención de que en el receptor se atenúen correspondientemente para reestablecer el equilibrio. (Electronics Notes, s.f.)

Las siguientes definiciones surgen a partir de que en el protocolo SAME un periodo de bit equivale a 1.92 ms con una desviación de ± 1 us:

1. La velocidad de datos es de 520.83 bits por segundo.
2. El cero lógico es 1562.5 Hz (tres periodos de bit).
3. El uno lógico es 2083.3 Hz (cuatro periodos de bit).
4. El periodo de los bits de señal y espacio es de 1.92 ms por igual.

La transmisión de datos en el protocolo SAME es un proceso asíncrono muy similar a la comunicación serial asíncrona y sus principales características son las que a continuación se describen.

- La sincronización de bit y byte se logra por medio de un preámbulo transmitido antes de cada encabezado o EOM.
- El preámbulo tiene una longitud de 16 bytes cada uno con el valor AB en hexadecimal o 10101011 en binario.
- Las partes digitales son transmitidas tres veces y el receptor está obligado a implementar corrección de paridad columnar, es decir, elegir el mejor byte de cada transmisión para construir un mensaje a interpretar.
- La información es codificada en bytes compuestos de 8 bits utilizando el código ASCII (*American Standard Code for Information Interchange*) de 7 bits. (INCITS, 2002)
- La transmisión de cada byte empieza por el bit de menos significativo y el bit más significativo siempre es cero.

1.3.2 Elementos

Como se mencionó anteriormente, los elementos del protocolo SAME son: Encabezado, Tono de alarma, mensaje de voz y EOM. Los elementos se pueden representar simbólicamente y se muestran en la Tabla 2. En las siguientes secciones se describe a mayor detalle estos elementos.

Tabla 2. Representación simbólica de un mensaje SAME. (NOAA, 2011, p. A-2)

Elemento	Representación simbólica
Encabezado	[Preambulo] ZCZC-ORG-EEE-PSSCCC-PSSCCC+TTTT-JJJHHMM-LLLLLLLLL- [Pausa de 1 segundo] [Preambulo] ZCZC-ORG-EEE-PSSCCC-PSSCCC+TTTT-JJJHHMM-LLLLLLLLL- [Pausa de 1 segundo] [Preambulo] ZCZC-ORG-EEE-PSSCCC-PSSCCC+TTTT-JJJHHMM-LLLLLLLLL- [Pausa de 1 a 3 segundos]
Tono de alarma	[Tono de 1050 Hz por 8 a 10 segundos] (opcional) [Pausa de 3 a 5 segundos]
Mensaje de voz	[Audio del mensaje] (opcional) [Pausa de 1 a 3 segundos]
EOM (Fin de mensaje)	[Preambulo] NNNN [Pausa de 1 segundo] [Preambulo] NNNN [Pausa de 1 segundo] [Preambulo] NNNN

1.3.2.1 Encabezado

Contiene la información digital del mensaje y su objetivo principal es dar a conocer al equipo receptor el tipo de alerta y las áreas geográficas afectadas. Cada encabezado transmitido no debe exceder los 268 bytes y su formato es el siguiente:

ZCZC-ORG-EEE-PSSCCC-PSSCCC+TTTT-JJJHHMM-LLLLLLLLL-

El encabezado está dividido en bloques de datos por el carácter “-” (guión) excepto antes del tiempo de validez del mensaje que está precedido por el carácter “+” (signo más). La definición de cada bloque es:

ZCZC

Indica el inicio del encabezado y su contenido son los valores ASCII de la cadena ZCZC.

ORG

Indica el autor del mensaje, en la Tabla 3 se listan los posibles códigos para este bloque.

Tabla 3. Código para autores de mensajes SAME. (NOAA, 2011, p. A-3)

Código	Autor
EAS	Radiodifusora o sistema de cable
CIV	Autoridad civil
WXR	Servicio Meteorológico Nacional (NWS)
PEP	Sistema de estaciones PEP (<i>Primary Entry Point</i>)

EEE

Identifica el tipo de evento que genera la alerta. Puede ser enviado con o sin tono de alerta y mensaje de voz además que puede funcionar como código de control para algunas funciones del sistema de NWR. En el Apéndice A se listan los códigos de eventos SAME.

PSSCCC

El código de área geográfica identifica la zona afectada por el mensaje. Cada código identifica de manera única a una zona geográfica. Un mensaje SAME puede contener hasta 31 códigos de área geográfica.

El dígito P permite dividir el área definida por el código CCC para el caso de áreas muy amplias, con características geográficas particulares o con cambios bruscos de elevación, clima o población. Un valor de P igual a cero indica que el evento aplica a toda el área y puede existir hasta 9 divisiones. Algunos ejemplos de divisiones se listan en la Tabla 4.

Tabla 4. Ejemplos de divisiones de áreas geográficas.

P	2 divisiones	3 divisiones	5 divisiones	9 divisiones
0	Area A completa	Area B completa	Area C completa	Area D completa
1		Montaña arriba de X elevación	Noroeste area C	Noroeste area D
2		Montaña entre X y Z elevación		Centro norte area D
3		Valle por debajo de Z elevación	Noreste area C	Noreste area D
4	Oeste area A			Centro oeste area D
5			Centro area C	Centro area D
6	Este area A			Centro este area D
7			Suroeste area C	Suroeste area D
8				Centro sur area D
9			Sureste area C	Sureste area D

El código SS indica el estado, territorio o área marina de EUA al que aplica el evento transmitido. La lista de códigos estatales y territoriales para EUA está definida en el listado de códigos estatales FIPS (*Federal Information Processing Standard*) mientras que los códigos de zonas marinas están definidos por la NOAA.

El código CCC identifica a un municipio, provincia o área metropolitana de EUA al que el mensaje está dirigido. Un código 000 cubre a todo el estado o área definida por el código SS. La lista de códigos municipales para EUA está definida en el FIPS 6-4 (*Federal Information Processing Standard Publication 6-4*).

Otros países que han adoptado el protocolo SAME implementan el código de área geográfica de diferentes maneras. En Canadá los seis dígitos del código PSSCCC emplean el código de región de pronóstico definidos por el Servicio Meteorológico de Canadá. En México los códigos de estado-municipio son definidos por el SASMEX de acuerdo a su cobertura en algunos municipios de los estados de Guerrero, Oaxaca, Morelia, Estado de México y Ciudad de México.

TTTT

Indica el tiempo en el cual expira el mensaje recibido expresado como una diferencia a partir del momento de su emisión y en intervalos de 15 minutos hasta una hora o en intervalos de 30 minutos desde una hora hasta seis horas, por ejemplo, +0015-, +0030-, +0045-, +0100-, +0430- ó +0600-. Este bloque de datos siempre es precedido por el carácter "+". Cuando sumamos esta diferencia de tiempo a la hora de emisión del mensaje se obtiene el momento cuando el mensaje ya no es válido y debe ser eliminado de los sistemas receptores.

Es importante notar que el periodo de validez de un mensaje no siempre es el mismo que el de la duración de un evento. Para eventos de corta duración como un tornado o tormenta, el tiempo de expiración y duración de la amenaza serán casi iguales. Para eventos con una duración más larga como un huracán o tormenta de nieve que pueden durar varias horas o incluso días, el tiempo de validez solo aplica para el mensaje y no debe ser interpretado como la duración de la amenaza.

JJJHHMM

Informa la fecha y hora de emisión del mensaje. JJJ indica la fecha ordinal (día) del año con ceros a la izquierda mientras que HHMM indica la hora y minuto en formato de 24 horas, con ceros a la izquierda y utilizando la zona horaria UTC.

LLLLLLLL

Indica el emisor del mensaje por medio de la matrícula del transmisor reemplazando el carácter "-" con "/", por ejemplo: WABC/FM para WABC-FM, KLOX/NWS para una estación de NWS o XGDF/002 en la Ciudad de México.

1.3.2.2 Tono de alarma

El propósito del tono de alarma o señal de alerta es indicar al receptor que se prepare para difundir un mensaje de voz. En receptores particulares generalmente implica encender su amplificador para emitir el mensaje de voz a un volumen suficiente. En transmisores de radio o televisión la recepción de la señal de alerta implica interrumpir su programación y preparar la retransmisión del mensaje de voz.

El tono de alarma, si es transmitido, se envía entre 1 y 3 segundos después de la tercera repetición del encabezado del mensaje. La frecuencia del WAT en NWR es de 1060 Hz (\pm 0.3%) y tiene una duración de 8 a 10 segundos.

En el caso de la aplicación del protocolo SAME en el EAS donde los mensajes son difundidos por radio y televisión, la señal de alerta consiste de una combinación de ondas senoidales de 853 y 960 Hz.

Gracias al encabezado del mensaje los receptores que lo interpretan tienen la capacidad de ignorar el tono de alarma si la alerta no está dirigida a su zona geográfica o discriminar ciertos eventos. Por medio del WAT receptores sin capacidad de interpretar el encabezado pueden seguir difundiendo mensajes sin ser obsoletos.

1.3.2.3 Mensaje de voz

Si es transmitido, el audio del mensaje debe empezar entre 3 y 5 segundos después de la transmisión de la tercera repetición del encabezado o posterior al WAT, lo que ocurra al final.

El audio del mensaje debe oscilar entre el 20% y 80% de modulación con pausas ocasionales cercanas al 0% y picos alcanzando, pero sin exceder, el 100% de modulación. La longitud del mensaje de voz no debe exceder los dos minutos.

1.3.2.4 Fin de mensaje

Al igual que el encabezado, se repite tres veces y su contenido son los valores ASCII de la cadena NNNN.

1.3.3 Ejemplos de mensajes

La aplicación más común del protocolo SAME es la difusión de alertas de riesgo. A continuación se muestra la representación simbólica de un mensaje de alerta con tono de alarma debido a un tornado (evento TOR) para los municipios de Wood, Fulton y el noroeste de Henry ubicados en el estado de Ohio, EUA. El mensaje tiene una validez de 30 minutos y fue emitido el día 159 a las 1829 UTC por el transmisor KCLE del sistema NWR:

[Preambulo]

ZCZC-WXR-TOR-039173-039051-139069+0030-1591829-KCLE/NWS-

[Pausa de 1 segundo]

[Preambulo]

ZCZC-WXR-TOR-039173-039051-139069+0030-1591829-KCLE/NWS-

[Pausa de 1 segundo]

[Preambulo]

ZCZC-WXR-TOR-039173-039051-139069+0030-1591829-KCLE/NWS-

[Pausa de 1 a 3 segundos]

[Tono de 1050 Hz por 8 a 10 segundos]

[Pausa de 3 a 5 segundos]

[Mensaje de voz informando la amenaza de tornado]

[Pausa de 1 a 3 segundos]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

El protocolo SAME también puede ser aplicado a la transmisión de mensajes que no son críticos. A continuación se muestra la representación simbólica de un mensaje sin tono de alarma con información meteorológica (evento SPS):

[Preambulo]

ZCZC-WXR-SPS-039173-039051-139069+0030-1591829-KCLE/NWS-

[Pausa de 1 segundo]

ZCZC-WXR-SPS-039173-039051-139069+0030-1591829-KCLE/NWS-

[Preambulo]

ZCZC-WXR-SPS-039173-039051-139069+0030-1591829-KCLE/NWS-

[Pausa de 1 segundo]

[Preambulo]

[Pausa de 1 a 3 segundos]

[Mensaje de voz compartiendo el pronóstico meteorológico]

[Pausa de 1 a 3 segundos]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

En Ciudad de México, el SASMEX transmite cada tres horas un mensaje de prueba con el evento RWT y en caso de alerta sísmica transmite un mensaje con el evento EQW. Para ambos tipos de evento, usualmente el código de área geográfica indica que aplica para todas las zonas posibles y la información de la fecha y hora de emisión del mensaje no es correcta.

A continuación se muestra la representación simbólica correspondiente a una recepción de la prueba periódica:

[Preambulo]

ZCZC-CIV-RWT-000000+0300-1311431-XGDF/002-

[Pausa de 1 segundo]

[Preambulo]

ZCZC-CIV-RWT-000000+0300-1311431-XGDF/002-

[Pausa de 1 segundo]

[Preambulo]

ZCZC-CIV-RWT-000000+0300-1311431-XGDF/002-

[Pausa de 1 a 3 segundos]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

Y en seguida se muestra la representación simbólica de una alerta sísmica recibida:

[Preambulo]

ZCZC-CIV-EQW-000000+0001-1311420-XGDF/001-

[Pausa de 1 segundo]

[Preambulo]

ZCZC-CIV-EQW-000000+0001-1311420-XGDF/001-

[Pausa de 1 segundo]

[Preambulo]

ZCZC-CIV-EQW-000000+0001-1311420-XGDF/001-

[Pausa de 1 a 3 segundos]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

[Pausa de 1 segundo]

[Preambulo]

NNNN

1.4 Receptores

Existen diferentes tipos de receptores de alertas de riesgo codificadas con el protocolo SAME, por ejemplo:

- Receptores de grado profesional para radiodifusoras de radio y televisión o agencias públicas que son responsables de retransmitir mensajes o reaccionar ante alertas.
- Radios de sobremesa alimentados por AC, usualmente con una batería de respaldo.
- Radios portátiles alimentados por baterías para excursionistas, navegantes y kits de emergencia.
- Radios portátiles de manivela que no requieren alimentación de AC o batería, especialmente diseñados para su uso en kits de emergencia.
- Receptores integrados como función auxiliar de otro dispositivo, como radios de banda civil, televisiones portátiles, radios FM o autoestereos.

- Módulos receptores electrónicos, como el Silicon Labs Si4707, diseñados para su integración en nuevos dispositivos y equipos.

Desde Abril de 2004 todos los radios vendidos como certificados para alertas públicas (*Public Alert-certified*) deben poder sintonizar la Banda del Clima, interpretar el protocolo SAME y cumplir cierto criterio de desempeño, como lo especifica el estándar CEA-2009 de la Asociación de Consumidores de Tecnología o CTA. (Consumer Technology Association, 2008).

En la especificación del protocolo SAME se enlistan una serie de recomendaciones y consideraciones para receptores de NWR que a continuación mencionaremos.

Uso del divisor en el código de área geográfica

El dígito cero en el divisor de área geográfica implica que el mensaje está destinado a toda o una parte no específica del área. Su uso en el receptor implica que el usuario desea ser alertado de eventos en cualquier división de su área geográfica.

Si un receptor está configurado para alertar de mensajes destinados a una porción de un área geográfica (por ejemplo 129139), el receptor debe alertar también de mensajes destinados a toda el área geográfica (por ejemplo 029139).

Por otra parte, si el receptor está configurado para alertar de mensajes destinados a toda un área geográfica (por ejemplo 029139), el receptor debe alertar de todos los mensajes destinados a cualquier división del área geográfica (por ejemplo 129139, 229139 ó 329139).

Discriminación por pares de eventos y códigos de área geográfica

Para aplicaciones industriales, comerciales y del sistema EAS, cualquier alertamiento o proceso de activación debe estar sujeto a coincidir con un par específico de evento y código de área geográfica. Por medio de una búsqueda en una tabla de eventos y códigos de área (por ejemplo: [TOR\033001], [TOR\033005], [FFW\033011], [CFW\033005], [SMW\075711], [SMW\075709]) un receptor de gama alta puede discriminar efectivamente los mensajes captados.

Equipo para consumidores finales puede poseer dos tablas en lugar de pares (por ejemplo: [TOR, FFW, CFW, SMW] y [033001, 033005]) por que el interés o la necesidad de alertamiento estará acotado a uno o máximo dos códigos de área geográfica reduciendo las activaciones del receptor de un 80 a 90%.

Corrección de errores

Para un buen desempeño del receptor, es recomendado que un algoritmo de software confirme que al menos dos de los tres encabezados recibidos son idénticos antes de proceder con interpretar el mensaje. Si esta prueba falla, se debe realizar un intento de reconstrucción del mensaje de bit por bit, es decir, comparar los bits con los de la misma posición en los encabezados y aceptar como válido el que aparezca en dos de tres recepciones.

Fin del mensaje

Un EOM se puede considerar válido si el receptor detecta el preámbulo seguido de al menos una N pero de preferencia dos N.

Múltiples mensajes

Los receptores de la Radio del Clima (con y sin funciones SAME) deben ser capaces de recibir e informar de mensajes a la velocidad que son transmitidos. Mensaje de alta prioridad no deben degradar mensajes anteriores o causar que sus contenido se pierda.

Mensajes de texto producidos por la información del encabezado

Dispositivos que generen un mensaje de texto en lenguaje simple deben prestar especial atención al tiempo de expiración del mensaje para no caer en el error de informar que este es el tiempo de duración del evento. Por ejemplo, se debe informar "... se espera que el evento dure por lo menos 6 horas o más..." y si el receptor cuenta con un reloj en tiempo real "... se espera que el evento dure por lo menos hasta las hh:mm o más...".

Alimentación externa DC

Los receptores deben estar equipados con un conector apropiado para ser alimentados por una fuente de energía de 12V DC como el sistema eléctrico de autos, camiones o botes. También aunque es conveniente una batería de 9V DC interna de respaldo en caso de fallas en el suministro de energía, esta no podría alimentar al receptor por más de unas pocas horas.

Reconocimiento del código de autor

Para asegurarse que los propietarios de receptores reciban el máximo de protección posible del sistema, los receptores deben procesar mensajes con cualquier código de autor.

Prueba de receptor

Existen dos eventos para probar los sistema de Radio del Clima y el protocolo SAME:

- RWT: *Required Weekly Test* o Test Semanal Obligatorio
- RMT: *Required Monthly Test* o Test Mensual Obligatorio

En EUA el RWT normalmente ocurre los Miércoles entre 10:00 am y 12:00 pm hora local con algunas variaciones, por ejemplo, si hay amenaza de clima peligroso cercano o al momento de la prueba se pospone al próximo día con buen clima. Los RMT normalmente coinciden con las pruebas semanales reemplazando el código en el mensaje correspondiente.

En la Ciudad de México el RWT se transmite cada 3 horas todos los días del año empezando a las 2:45 am y no se implementa el RMT.

Cualquier otro mensaje destinado al área geográfica programada en el dispositivo automáticamente debe tomarse en cuenta como una prueba exitosa. Todos los receptores del protocolo SAME deben responder a estos tests de alguna manera y se proponen dos métodos:

1. Mostrar un indicador de que la unidad recibió la prueba correctamente. Aunque el mensaje de prueba especifica un tiempo de expiración de 15 minutos el indicador de prueba debe mantenerse activado por lo menos 12 horas para que el propietario del receptor pueda observar el resultado de la prueba.
2. Notificar al usuario cuando la unidad sospeche un error. Si el receptor no ha recibido ningún mensaje destinado a su área geográfica en una ventana de tiempo (por ejemplo 8 o 9 días), la unidad debe mostrar un indicador para su revisión.

Estos dos métodos se basan en que el receptor capte un mensaje válido de tipo RWT, RMT u otro coincidiendo al menos un código de área geográfica con los programados en el receptor. Solo reconociendo la recepción de cualquier mensaje no es suficiente para confirmar la prueba de la unidad.

Si existiera una alarma externa conectada al receptor (como luces, sirena, altavoz, etc) se debe proporcionar la funcionalidad para probarla por lo menos de forma manual.

Capítulo 2. Sistemas Embebidos

Un sistema embebido, o en inglés *embedded system*, es un sistema de computación aplicado que se distingue de otros tipos de sistemas de computación como las computadoras personales o supercomputadoras. La definición de sistema embebido es fluida y difícil de precisar ya que constantemente evoluciona por los avances de la tecnología y disminución de costos.

En este capítulo se estudiará:

- Las definiciones tradicionales de sistema embebido.
- Los principales componentes de los sistemas embebidos.
- El proceso de diseño para los sistemas embebidos.

2.1 Definición

Ya que en los últimos años los sistemas embebidos han superado muchas de sus descripciones tradicionales, a continuación se mencionan las más comunes junto a su razonamiento y un breve análisis de su validez en la actualidad. (Noergaard, 2005, p. 5)

En comparación con una computadora personal, los sistemas embebidos están limitados en su hardware y software.

Esta definición es verdadera para un gran número de sistemas embebidos. Las limitantes de hardware significa que cuentan con poco poder de cómputo, tienen un bajo consumo de energía o poseen memoria reducida entre muchas otras posibles limitantes. En el software las limitantes en comparación con una computadora personal son, por ejemplo, que no cuentan con un sistema operativo o tienen uno muy básico, las aplicaciones deben construirse a baja escala o cuentan menos capas de abstracción en su código.

Sin embargo, en la actualidad microprocesadores o software que típicamente encontramos en una computadora personal han sido adaptados para utilizarse en sistemas embebidos complejos.

Un sistema embebido está diseñado para realizar una función específica.

La mayoría de los sistemas embebidos están diseñados principalmente para una función específica. Sin embargo, en la actualidad existen sistemas embebidos que permiten realizar una amplia variedad de funciones sin relación a la principal pero igual de importantes. Un ejemplo son los sistemas embebidos parte de las televisiones inteligentes que permiten navegar por Internet, leer el correo electrónico o mostrar fotografías además de proyectar la televisión.

Algunos dispositivos llamados sistemas embebidos, como los teléfonos inteligentes o asistentes personales (PDAs), no son sistemas embebidos.

Existe controversia si se debe clasificar como embebidos a ciertos sistemas computacionales que solo poseen algunas características de los sistemas embebidos. Algunos creen que denominar como embebidos a sistemas complejos, como los teléfonos inteligentes, es producto de profesionales especializados en mercadotecnia o ventas. En realidad los ingenieros en sistemas embebidos están divididos respecto a si se deben clasificar como embebidos o no.

Ya sea que las definiciones tradicionales sigan evolucionando o un nuevo campo de la computación incluya a estos sistemas de computación complejos, por ahora y a lo largo de este trabajo se clasificará como embebido a estos sistemas de computación.

Un sistema embebido es diseñado con requerimientos de alta calidad y confiabilidad en comparación con otros sistemas de cómputo.

Algunos tipos de sistemas embebidos tienen estrictos requerimientos de calidad y confiabilidad. Por ejemplo, si la computadora del motor de un automóvil falla en una autopista o un dispositivo médico deja de funcionar durante una cirugía, existen serias consecuencias para sus usuarios. Sin embargo, para sistemas embebidos como los que se pueden encontrar en las televisiones, videojuegos o teléfonos inteligentes, una falla es un inconveniente pero sin consecuencias que pongan en riesgo la vida de sus usuarios.

2.1.1 Aplicaciones

Productos en casi cualquier segmento de mercado cuentan con sistemas embebidos. En la actualidad es difícil encontrar algún dispositivo electrónico cuyo funcionamiento no esté basado en algún sistema embebido.

En la Tabla 5 se presenta un listado de aplicaciones actuales de los sistemas embebidos agrupadas por el segmento de mercado al que pertenecen.

Tabla 5. Ejemplos de sistemas embebidos y su segmento de mercado (Noergaard, 2005, p. 5).

Mercado	Sistemas embebidos
Automotriz	Sistema de encendido
	Módulo de control del tren motriz o PCM
	Sistema de control de estabilidad o ESP
	Sistemas de entretenimiento y navegación
Electrónica de consumo	Sistemas de entretenimiento (televisores, decodificadores, etc.)
	Electrodomésticos (refrigeradores, microondas, etc.)
	Teléfonos inteligentes y cámaras fotográficas
	Sistemas de domótica y automatización
	Juegos y juguetes
Industrial	Sistemas robóticos para asistir a la manufactura
	Sistemas de control para motores o actuadores
	Operación de sistemas neumáticos o hidráulicos
	Control de calidad automatizado
Comunicaciones	Ruteadores, switches o hubs
	Puntos de acceso o repetidoras inalámbricas
	Equipo para telecomunicaciones celulares
Negocios	Fotocopiadoras
	Impresoras
	Conmutadores
Seguridad	Sistemas de circuito cerrado
	Sistemas de control de acceso

2.2 Componentes

En un sistema embebido podemos encontrar cinco componentes principales: (1) microprocesador, (2) memoria, (3) periféricos, (4) software y (5) algoritmos. (Heath, 2003, p. 8). A continuación definimos cada uno de estos componentes.

2.2.1 Microprocesador

El microprocesador es la principal unidad funcional de un sistema embebido y su propósito es procesar instrucciones y datos. Un dispositivo electrónico posee al menos un microprocesador principal que actúa como el dispositivo de control central y puede contener microprocesadores secundarios bajo el control del microprocesador principal. (Noergaard, 2005, p. 129).

El principal criterio de selección para un microprocesador es si este puede proveer la capacidad de procesamiento necesario para realizar las tareas del sistema embebido. Parece obvio pero frecuentemente ocurre que las tareas del sistema embebido son subestimadas en tamaño o complejidad. Otros criterios para la elección de un procesador son:

- Costo: Tanto del procesador individual como de los componentes auxiliares para su funcionamiento, por ejemplo, fuente de alimentación o sistema de enfriamiento.
- Disponibilidad en el mercado: Debe estar disponible en el volumen necesario y durante la vida útil de la producción del dispositivo.
- Software: Disponibilidad y costos de herramientas como ensambladores, compiladores, depuradores o emuladores.

Una alternativa a los microprocesadores son los microcontroladores: sistemas autocontenidos que agrupan un microprocesador, memoria y periféricos en un solo circuito integrado.

Con el empleo de microcontroladores los sistemas embebidos pueden simplificarse y ser más confiables ya que requieren menos componentes en sus placas electrónicas reduciendo posibles puntos de falla. Además el consumo energético de componentes agrupados en un circuito integrado típicamente es menor al de los mismos componentes por separado a nivel de placa electrónica.

En la Figura 1 se comparan los sistemas embebidos basados en microprocesadores y microcontroladores. Podemos apreciar que los elementos son los mismos y como el microcontrolador los agrupa en una unidad autocontenida.

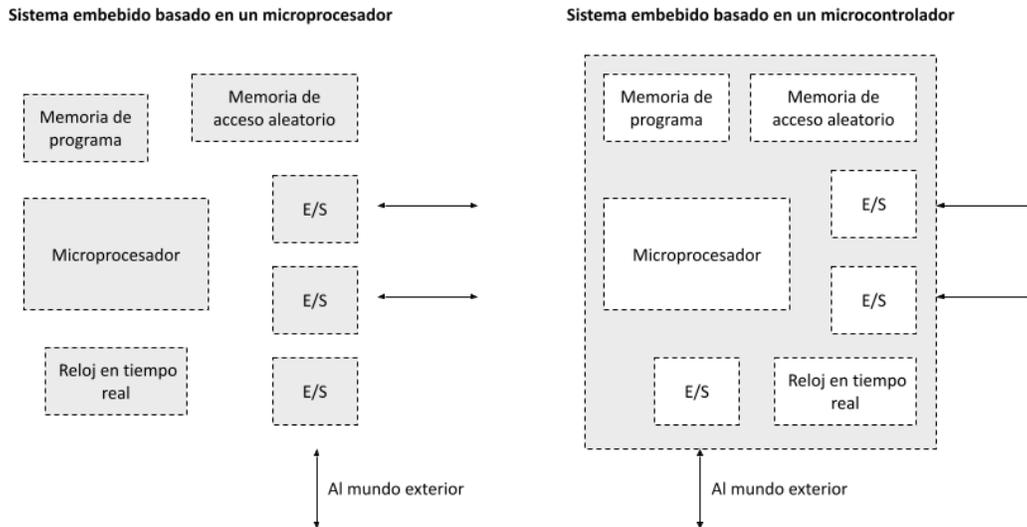


Figura 1. Microcontrolador vs. microprocesador. (Berger, 2002, p. 34)

Con menos componentes y menores requerimientos energéticos un microcontrolador puede dar como resultado un dispositivo más pequeño y económico. Por otra parte, con los microcontroladores hay menos flexibilidad en añadir, cambiar o eliminar funcionalidad ya que los componentes integrados no pueden cambiarse con la misma facilidad que si hubieran sido implementados a nivel de placa electrónica.

Comercialmente los microcontroladores y microprocesadores se clasifican de acuerdo al ancho de su bus de datos. Microchip (s.f. e), uno de los principales fabricantes de componentes para sistemas embebidos, utiliza las siguientes categorías para su línea de microcontroladores y microprocesadores:

Microcontroladores de 8 bits

- Son el punto de entrada más fácil para el desarrollo en microcontroladores.
- Diseñados para aplicaciones de control en tiempo real.
- Bajo costo, bajo consumo energético y alta confiabilidad.

Microcontroladores de 16 bits

- Simplicidad de un microcontrolador de 8 bits con mayor desempeño y capacidad.
- Diseñados para aplicaciones de control, conversión de energía o seguridad.
- Buen balance entre costo, consumo energético y desempeño.

Microcontroladores de 32 bits

- Conectividad integrada: CAN, USB, Ethernet, etc.
- Periféricos para interfaces humanas como gráficos y pantallas táctiles.
- Bajo consumo energético y alto desempeño para aplicaciones multihilo.

Microprocesadores

- Siguiendo el paso para las aplicaciones que superan las capacidades de los microcontroladores.
- Soluciones escalables para comunicaciones y almacenamiento seguros.
- Soporte para RTOS (sistemas operativos en tiempo real) y Linux.

2.2.2 Memoria

La memoria es una parte importante de cualquier sistema embebido y es fuertemente influenciada por el diseño de software al mismo tiempo que indica cómo el software es diseñado, escrito y desarrollado.

Según Heath (2003, p. 8), la memoria tiene dos principales funciones en los sistemas embebidos:

Almacena el software del sistema embebido.

El software almacenado puede ser el programa completo o rutinas de inicialización para obtener el programa completo de otra fuente ya sea dentro o fuera del sistema embebido. Este tipo de almacenamiento se implementa con memoria no volátil ya sea tipo ROM (memoria de solo lectura) o EPROM (memoria de sólo lectura eléctricamente programable).

Almacena datos durante la operación del sistema embebido como variables, resultados parciales o información de estatus.

El software requiere memoria para almacenar variables y administrar estructuras de datos como pilas o listas. Normalmente la memoria requerida para variables y estructuras de datos es menor a la necesaria para el programa de software. Este tipo de almacenamiento se implementa con memoria volátil tipo RAM (memoria de acceso aleatorio).

Siendo más cara la memoria RAM que la ROM o EPROM, muchos microcontroladores tienen pequeñas cantidades de almacenamiento volátil y su software debe estar diseñado para minimizar su uso. Esta situación también requiere el uso de compiladores que produzcan programas cuyo código se ejecute directamente desde una memoria de almacenamiento.

Comercialmente la cantidad de memoria en un microcontrolador guarda una relación con el tamaño de su bus de datos y su aplicación. En la Tabla 6 se presentan los rangos comerciales para el tamaño de memoria en microcontroladores de acuerdo al tamaño de su bus de datos.

Tabla 6. Memoria volátil y de programa por categoría de microcontrolador. (Microchip, s.f. e)

Categoría	Memoria de programa	Memoria de acceso aleatorio
Microcontroladores de 8 bits	0.5 - 384 KB	0.032 - 32 KB
Microcontroladores de 16 bits	4 - 512 KB	0.5 - 48 KB
Microcontroladores de 32 bits	8 - 2048 KB	2 - 512 KB

2.2.3 Periféricos

Un sistema embebido debe comunicarse con el mundo exterior y eso se lleva a cabo con periféricos. Un periférico es cualquier cosa aparte del procesador que se comunique con él. Como el procesador tiene acceso directo a la memoria, esta no cuenta como periférico. (White, 2011, p. 149).

Los periféricos de entrada normalmente son asociados con sensores que miden el magnitudes externas para calcular las operaciones de control que el sistema embebido debe llevar a cabo. De esta manera podemos modelar a los sistemas embebidos como un proceso de tres etapas: datos e información entran a la primera etapa, son procesados en la segunda etapa y tienen salida en la tercera etapa. (Heath, 2003, p. 9).

Si este modelo es aplicado a un controlador de motor, los datos e información de entrada sería la velocidad del motor, su potencia y la velocidad requerida por el operador. La etapa de salida consistiría en una señal PWM (modulación por ancho de pulsos) para controlar la potencia entregada al motor y por lo tanto su velocidad además de un indicador para mostrar la velocidad. La etapa intermedia sería el software que procesa las entradas y ajusta la salida para alcanzar la velocidad requerida del motor.

Según Noergaard (2005, p. 268) las categorías más comunes de periféricos son:

- Entrada (desde switches o botones hasta teclados, ratones o controles remoto).
- Salida y gráficos (LEDs, pantallas alfanuméricas o monitores).
- Almacenamiento (memorias externas y discos ópticos o magnéticos)
- Comunicaciones y Redes (interfaces SPI, I2C, Ethernet o Wireless).
- Depuración (interfaces JTAG, BDM o puertos seriales/paralelos).
- Tiempo real y misceláneos (temporizadores, relojes, contadores o convertidores analogico-digital y digital-analógico).

Los periféricos de un sistema embebido pueden ser tan simples como un circuito electrónico básico conectado al procesador principal o complejos circuitos con diversos subsistemas.

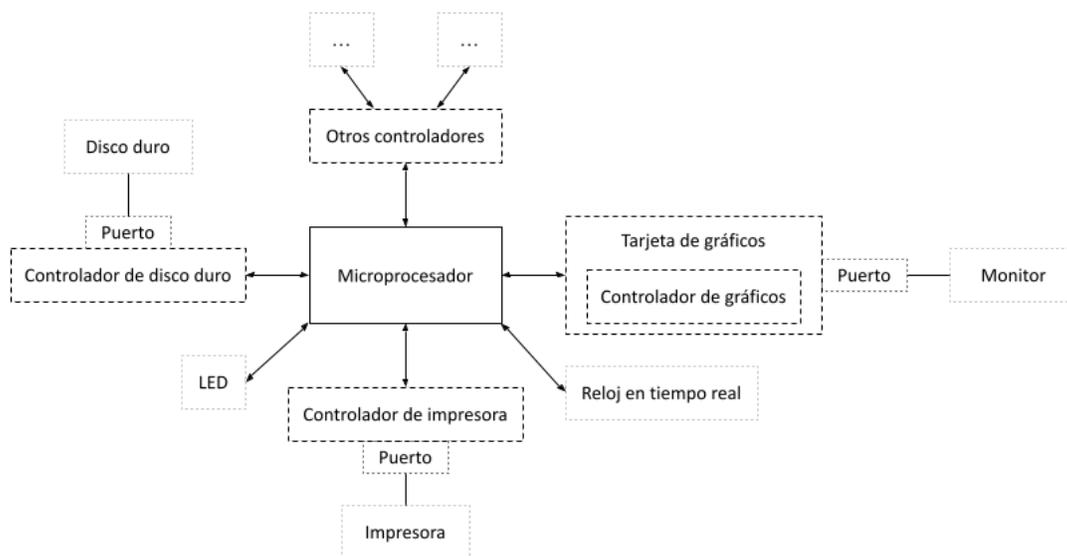


Figura 2. Periféricos de un sistema embebido. (Noergaard, 2005, p. 255)

Los microcontroladores integran un conjunto de periféricos de comunicación e interfaces para periféricos específicos. Acorde con la categoría del microcontrolador se pueden integrar desde periféricos simples con interfaces básicas o periféricos complejos con interfaces de alto desempeño, como se muestra en la Figura 2.

Por otro lado, en una misma categoría de microcontroladores existen modelos que integran muy pocos dispositivos y otros que integran gran cantidad de ellos. Algunas de las opciones de periféricos que ofrece Microchip (s.f. a, b, c, d) por categoría de microcontrolador son:

Microcontroladores de 8 bits

- Convertidores analógico-digital y digital-analógico de 8 a 12 bits
- Generadores y capturadores PWM de 10 o 16 bits
- Interfaces de comunicación CAN (*Controller Area Network*), LIN (*Local Interconnected Network*), I2C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*), USART (*Universal Asynchronous Receiver Transceiver*) o USB (*Universal Serial Bus*).
- Interfaces para pantallas LCD y sensores táctiles.

Microcontroladores de 16 bits

- Convertidores analógico-digital y digital-analógico de 10 a 16 bits
- Generadores y capturadores PWM de 16 o 32 bits
- Interfaces de comunicación CAN, LIN, I2C, I2S (*Inter-IC Sound Controller*), SPI, USART o USB.
- Interfaces para pantallas LCD, gráficas y sensores táctiles.

Microcontroladores de 32 bits

- Convertidores analógico-digital y digital-analógico de 10 a 48 canales
- Generadores y capturadores PWM de 4 a 24 canales
- Interfaz de comunicación SERCOM (*Serial Communication Module*) que puede actuar como un periférico I2C, SPI o USART.
- Interfaces de comunicación CAN, Ethernet, I2S, USB o SD (*Secure Digital*).
- Interfaces para sensores CMOS (captura de imágenes).
- Interfaces para pantallas gráficas y PTU (*Peripheral Touch Controller*).
- Módulos de seguridad como TrustZone (separación de recursos confiables o no confiables), TrustRAM (almacenamiento seguro de llaves) o SecureBoot (auténtica la memoria de programa al arranque del sistema).

2.2.4 Software

Los componentes de software en un sistema embebido abarca la tecnología que añade valor al sistema y define lo que hace y que tan bien lo hace (Heath, 2003, p. 10).

Como se muestra en la Figura 3, Noergaard (2005, p. 309) divide el software de un sistema embebido en dos categorías:

- El *software de sistema* es cualquiera que soporta las aplicaciones, como controladores de dispositivo, sistemas operativos o software intermedio (middleware).
- El *software de aplicación* es el de nivel superior que define la función y propósito del sistema embebido y que se encarga de la interacción con usuarios.

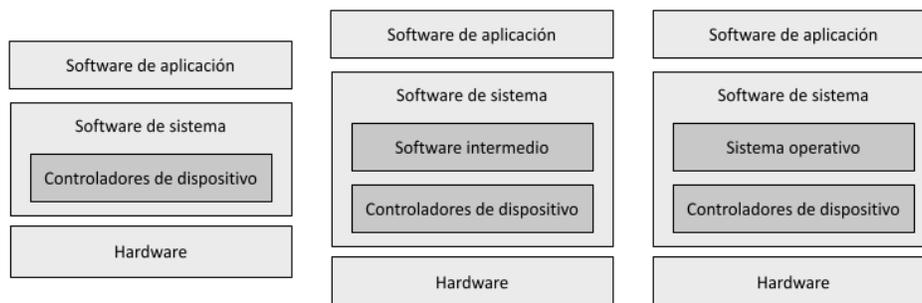


Figura 3. Software de un sistema embebido. (Noergaard, 2005, p. 311)

A continuación se describen los componentes de software más comunes en los sistemas embebidos: controladores de dispositivo, sistemas operativos, software intermedio y software de aplicación.

Controladores de dispositivo

La mayoría de sistemas embebidos requiere software que lleve a cabo una inicialización y gestión de su hardware. El software que directamente interconecta y controla el hardware se le conoce como *controlador de dispositivo*. Todos los sistemas embebidos que requieren software tienen, por lo menos, controladores de dispositivo en la capa de software de sistema.

Los controladores de dispositivo son las bibliotecas de software que inicializa el hardware y administran el acceso a él por otras capas de software. Los controladores de dispositivo son el enlace entre el hardware y el sistema operativo, software de intermedio o de aplicación.

Existen dos tipos de controladores de dispositivo: específicos de una arquitectura o genéricos (Noergaard, 2005, p. 312).

- Un controlador de dispositivo específico de una arquitectura administra el hardware que está integrado en el procesador principal. Ejemplos de controladores de dispositivo específicos de una arquitectura son los que inicializan y habilitan la memoria integrada o la unidad de punto flotante del microprocesador.
- Un controlador de dispositivo genérico administra hardware que está en la placa de un sistema embebido pero que no está integrado con el procesador principal. Controladores de dispositivo genéricos incluyen código que inicializa y administra el acceso al resto de principales componentes del sistema embebido incluyendo buses de comunicación (I2C, CAN, SPI, etc), almacenamiento (memorias Flash, EEPROM o discos) y dispositivos de entrada/salida (pantallas, teclados, etc).

Sistema operativo

El sistema operativo es un conjunto de bibliotecas de software que cumple dos propósitos:

- Proveer una capa de abstracción para independizar el software del hardware y facilitar el desarrollo del software de aplicación o intermedio.
- Administrar los diversos sistemas de hardware y recursos de software para asegurarse que el sistema embebido opera eficiente y confiablemente.

No todos los sistemas embebidos cuentan con un sistema operativo y este varían en componentes pero todos los sistemas operativos poseen un elemento en común: el kernel o núcleo. (Noergaard, 2005, p. 384)

El núcleo es el componente con las principales funciones del sistema operativo y puede proveer todas o una combinación de las siguientes funciones:

1. Administración de procesos. El sistema operativo administra por medio de procesos el resto de software del sistema embebido. Una subfunción típicamente encontrada en la administración de procesos es la administración de interrupciones y detección de errores. Múltiples interrupciones o excepciones generadas por diversos procesos deben ser administradas eficientemente para que puedan ser manejadas correctamente.

2. Administración de memoria. El espacio de memoria en un sistema embebido es compartido por todos los diferentes procesos, por lo que el acceso y la distribución de porciones del espacio de memoria debe ser administrado. Una subfunción del administrador de memoria es la administración de seguridad del sistema que impide que el sistema embebido sufra trastornos en partes sensibles y lo mantiene seguro de software hostil o mal escrito.
3. Administración del sistema de E/S. Los dispositivos de entrada y salida también necesitan ser compartidos entre diversos procesos y de igual manera que con la memoria el acceso y distribución de dispositivos necesita ser administrada. Por medio del administrador del sistema de E/S, la administración del sistema de archivos puede proveer un método para almacenar y administrar datos en forma de archivos.

Software intermedio

El software intermedio es aquel que generalmente media entre el software de aplicación y el núcleo o controladores de dispositivo. También el software intermedio puede intermediar entre diferentes software de aplicación. Específicamente el software intermedio es una capa de abstracción utilizada principalmente en sistemas embebidos con dos o más aplicaciones con el fin de proveer flexibilidad, seguridad, portabilidad, conectividad, intercomunicación y/o mecanismos de interoperabilidad entre aplicaciones. (Noergaard, 2005, p. 446)

Una de las principales ventajas de usar software intermedio es que permite reducir la complejidad de aplicaciones centralizando código que tradicionalmente sería redundante en diferentes software de aplicación. Sin embargo, al utilizar software intermedio se integra una sobrecarga que puede reducir la escalabilidad y desempeño del sistema embebido. En resumen, el software intermedio impacta al sistema embebido en todos sus niveles.

Software de aplicación

El software de aplicación es la última capa del software de un sistema embebido y es dependiente, administrado y ejecutado por el software de sistema. El software de aplicación define qué producto es el sistema embebido ya que su funcionalidad es el propósito del dispositivo y lleva a cabo la mayoría de interacciones con usuarios. (Noergaard, 2005, p. 447)

Las aplicaciones de un sistema embebido pueden ser agrupadas de acuerdo a si son específicas para un mercado (como un receptor de televisión digital) o de propósito general (como un navegador de páginas web).

2.2.5 Algoritmos

Los algoritmos son el componente clave del software que permite a un sistema embebido comportarse de la manera en que lo hace. Los algoritmos pueden ser desde simple procesamiento lógico-matemático hasta complejos modelos del entorno del sistema embebido para interpretar la información de sensores y así generar señales de control. (Heath, 2003, p. 10).

Con la tecnología digital de hoy en día como reproductores de audio y video portátiles, la mayoría de los algoritmos que codifican digitalmente datos análogos son definidos por organismos de normalización. Mientras que esta estandarización puede significar que no es importante la selección de algoritmos, la realidad es diferente. La implementación y elección correcta de los algoritmos es importante por que puede permitir la misma funcionalidad en hardware más económico y ya que la mayoría de sistemas embebidos son diseñados para ser comercialmente exitosos, este proceso es muy importante.

2.3 Diseño

Diferentes modelos pueden utilizarse para describir el proceso de diseño de un sistema embebido. La mayoría de estos modelos están basados en uno o son una combinación de los siguientes modelos de desarrollo.

- El modelo *big-bang*, en donde esencialmente no hay planeación o procesos antes y durante el desarrollo de un sistema.
- El modelo *programar-y-corregir*, en donde los requerimientos del producto son definidos pero no hay procesos formales antes de iniciar el desarrollo.
- El modelo *cascada*, donde hay un proceso para desarrollar un sistema en pasos, donde el resultado de un paso fluye al siguiente.
- El modelo *espiral*, donde el proceso para desarrollar un sistema se repite incorporando la retroalimentación obtenida en iteraciones previas.

En comparación con el desarrollo de software para una sistema de propósito general, el diseño de un sistema embebido implica que software y hardware son diseñados en paralelo. Según Berger (2002, p. 17) el proceso de diseño de un sistema embebido cuenta con las siguientes siete fases, que además se muestran en la Figura 4.

1. Especificación de producto.
2. Distribución de software y hardware.
3. Iteración y refinamiento de la distribución.
4. Diseño de software y hardware.
5. Integración de software y hardware.
6. Pruebas de producto y lanzamiento.
7. Mantenimiento y actualizaciones.

Las implicaciones del proceso de diseño simultáneo de hardware y software afecta considerablemente como los sistemas embebidos son diseñados. Además, la economía y realidad de los requerimientos de diseño con frecuencia obliga a tomar decisiones antes de que los diseñadores puedan considerar los pros y contras de estas elecciones.

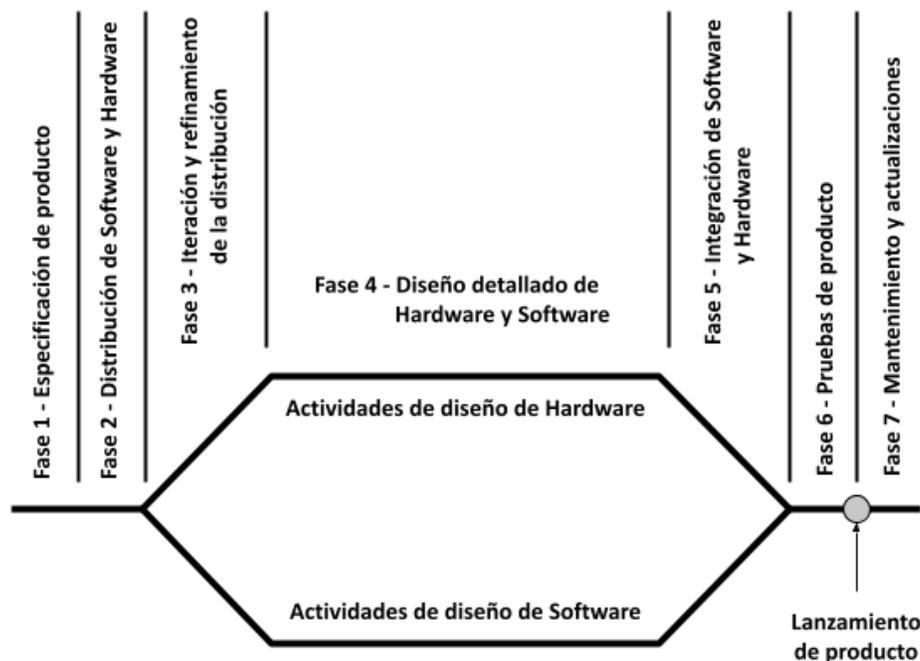


Figura 4. Proceso de diseño de sistemas embebidos. (Berger, 2002, p. 17)

2.3.1 Especificación de producto

El diseño de sistemas embebidos debe considerar el diseño del producto correcto. Los requerimientos y las especificaciones son componentes muy importantes en el desarrollo de cualquier sistema. El análisis de requerimientos es el primer paso en el proceso de diseño de un sistema embebido y es donde los requerimientos de los usuarios finales deben ser clarificados y documentados para generar la especificación de producto correspondiente (Tran, 1999).

Existe una clara diferencia entre requerimientos y especificaciones.

- Un requerimiento es una función que necesita un usuario para resolver un problema o lograr un objetivo.
- Una especificación es un documento que especifica, en una manera completa, precisa y verificable, los requerimientos, diseño, comportamiento o cualquier otra característica de un sistema, y con frecuencia, los procedimientos para determinar si estas provisiones se han cumplido.

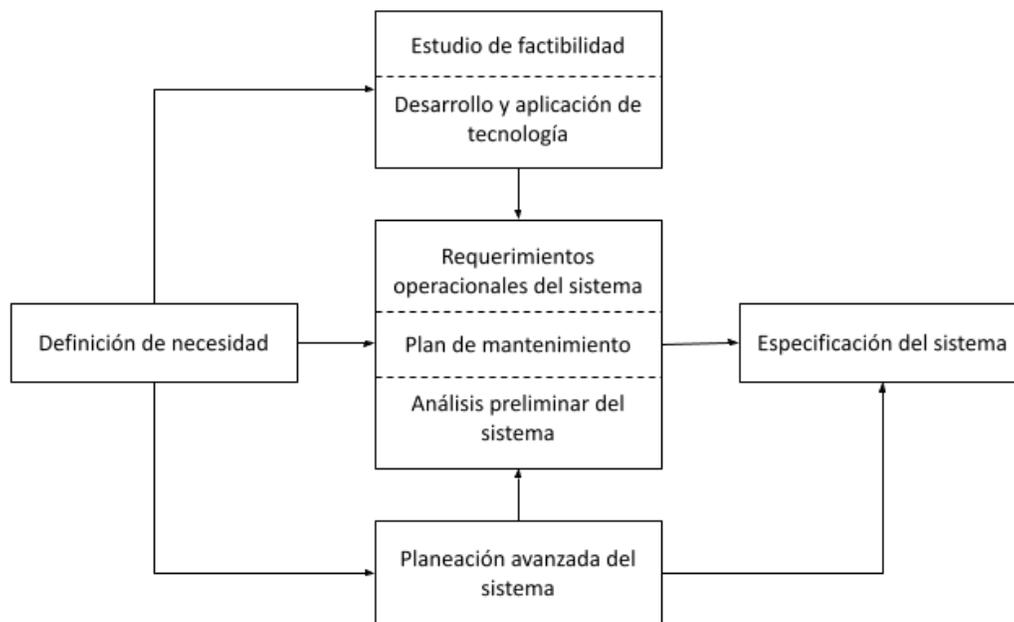


Figura 5. Proceso para definir la especificación del sistema embebido.

(Blanchard y Fabrycky, 1990, citado por Tran, 1999)

Como se muestra en la Figura 5, para establecer los requerimientos de sistema el primer paso es definir una necesidad. Usualmente, un individuo u organización identifica la necesidad de un producto o función y entonces se desarrolla un nuevo sistema o se modifica uno existente para cumplir con el requerimiento.

Con una necesidad claramente definida se debe hacer un estudio de factibilidad para analizar los posibles diferentes acercamientos técnicos para satisfacerla. El estudio de factibilidad inicia con definir los requerimientos operacionales del sistema y estos deben incluir:

- **Definición de misión.** Identificación de la misión operativa primaria del sistema en adición a misiones secundarias alternativas.
- **Parámetros físicos y de desempeño.** Definición de las características de operación o funciones del sistema.
- **Requerimientos de uso.** Anticipación de los usos del sistema.
- **Distribución operacional.** Identificación de los requerimientos de transporte y movilidad. Incluye cantidad de equipo, personal, etc y localización geográfica.
- **Ciclo de vida operacional.** Anticipación del tiempo que el sistema estará en uso operacional.
- **Factores de efectividad.** Especificaciones numéricas de requerimientos del sistema. Incluye factor de costo-efectividad, tiempo entre mantenimientos (MTBM), tasa de fallas, horas perdidas por mantenimiento, etc.
- **Ambiente.** Definición del ambiente en donde se espera opere el sistema.

La tendencia al definir requerimientos de sistema es solo cubrir las áreas relacionadas con desempeño descuidando las áreas relacionadas con soporte o mantenimiento. Esto significa que se da más importancia a solo una parte del sistema y no en su totalidad. Es esencial tomar en consideración el sistema completo cuando se definen los requerimientos del sistema. El plan de mantenimiento básicamente describe el ambiente de soporte en cual se supone el producto existirá.

Después de que los requerimientos operacionales del sistema y el plan de mantenimiento estén definidos, se debe llevar a cabo el análisis preliminar del sistema para determinar qué acercamiento para el desarrollo del sistema se debe adoptar. Usualmente el siguiente proceso es aplicado:

1. **Definir el problema.** El primer paso siempre empieza con clarificar los objetivos, definir los asuntos relevantes y limitar el problema para que pueda ser estudiado efectivamente.
2. **Identificar alternativas factibles.** Todas las alternativas deben considerarse para asegurarse que el mejor acercamiento es elegido.
3. **Seleccionar el criterio de evaluación.** El criterio para el proceso de evaluación puede variar considerablemente así que se debe elegir al más apropiado.
4. **Aplicar técnicas de modelación.** Un modelo o serie de modelos deben ser utilizados.
5. **Generar datos de entrada.** Los requerimientos para datos de entrada válidos deben ser especificados.
6. **Manipular el modelo.** Después que los datos son recolectados el modelo puede ser empleado. El análisis después de emplear el modelo puede llevar a recomendar algún tipo de acción.

Después del análisis preliminar del sistema se debe llevar a cabo la planeación avanzada del sistema. Los resultados de esta planeación serán definidos como requerimientos técnicos incluidos en la especificación del producto o requerimientos administrativos incluidos en un plan de administración del proyecto. Eso se puede ver en la Figura 6.

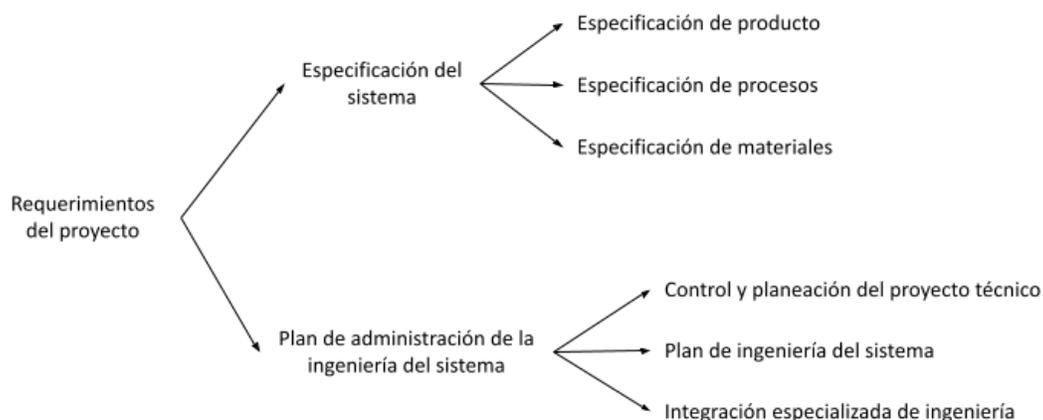


Figura 6. Documentación del proyecto. (Blanchard y Fabrycky, 1990, citado por Tran, 1999)

La especificación del producto incluye información de los requerimientos operacionales, plan de mantenimiento y estudio de factibilidad. El plan de administración de la ingeniería del sistema o SEMP (del inglés *System Engineering Management Plan*) se compone de tres secciones:

- **Control y planeación del proyecto técnico.** Describe las tareas del proyecto que tienen que planearse y desarrollarse para alcanzar los objetivos de la ingeniería del sistema como el plan de la estructura del proyecto, organización, manejo de riesgos, etc.
- **Plan de ingeniería del sistema.** Describe como el proceso de la ingeniería del sistema aplica a los requerimientos del proyecto.
- **Integración especializada de ingeniería.** Describe los principales requerimientos a nivel de sistema en las áreas de especialidad de ingeniería como confiabilidad, mantenibilidad, control de calidad, etc.

2.3.2 Distribución de software y hardware

Ya que el diseño de un sistema embebido involucra componentes de hardware y software, alguien debe decidir qué parte del problema será resuelto en hardware y que parte en software. (Berger, 2002, p. 22)

Si se generaliza el concepto de algoritmo a los pasos requeridos para implementar un diseño, se puede concebir al algoritmo como una combinación de componentes de hardware y software. En algunos casos se puede implementar un algoritmo en software, en hardware o por una combinación de ambos.

La decisión de distribución es un complejo problema de optimización. Muchos sistemas embebidos deben poseer las siguientes características:

- Precio sensible.
- Desempeño de vanguardia.
- No estándar.
- Competitivos en el mercado.
- Propietarios.

Estos requerimientos conflictivos hacen difícil crear un diseño óptimo para el sistema embebido. El algoritmo de distribución depende en que microprocesador se implemente el diseño y se pueden elegir entre varios cientos de microprocesadores, microcontroladores y núcleos a la medida. La elección de CPU impacta la decisión de distribución, que impacta a las decisiones de herramientas y así sucesivamente.

Dado este espacio de posibles elecciones, el diseñador o equipo de diseño debe confiar en su experiencia para lograr un diseño óptimo. Por otra parte, la solución del problema generalmente es evidente y significa que una solución válida (posiblemente motivada por una restricción completamente diferente) está frecuentemente muy cerca de la posible mejor solución. De cualquier manera, las restricciones son las que usualmente encaminan el proceso de decisión de los ingenieros en la distribución de software y hardware.

2.3.3 Iteración y refinamiento de la distribución

Esta parte del proceso de diseño de un sistema embebido representa un área borrosa entre la implementación y la distribución de hardware-software donde los caminos del software y hardware divergen. Esta fase representa el trabajo temprano de diseño antes de que los equipos de hardware y software trabajen independientemente. (Berger, 2002, p. 24)

El diseño del sistema embebido es todavía variable en esta fase. Aunque los mayores bloques del sistema han sido distribuidos entre los componentes de hardware y de software, queda mucha libertad de acción para mover esos límites conforme las restricciones de diseño son entendidas y modeladas.

Los diseñadores de hardware pueden usar herramientas de simulación, como simuladores de arquitectura, para modelar el desempeño de un sistema de procesador y memoria. Los diseñadores de software probablemente ejecutan pruebas de rendimiento en computadoras monoplaca que utilizan el microprocesador elegido. A estas computadoras monoplaca se les conoce también como placas de evaluación porque permiten evaluar el desempeño de un microprocesador ejecutando código de prueba en él. La placa de evaluación también provee un ambiente conveniente para el diseño y depuración de software hasta que el sistema de hardware real esté disponible.

2.3.4 Diseño de software y hardware

Teniendo clara la distribución de hardware y software los programadores e ingenieros pueden implementar el sistema embebido que se ajuste a los requerimientos del producto. Para el desarrollo e integración de los diversos componentes de software y hardware de un sistema embebido es importante entender las herramientas de desarrollo disponibles para la implementación de sistemas embebidos.

El entorno de desarrollo para un sistema embebido consta de dos elementos: un *sistema destino* (el sistema embebido en proceso de diseño) y un *sistema anfitrión* (una computadora personal donde el sistema embebido es desarrollado). El sistema destino y el sistema anfitrión están conectados por un medio de transmisión ya sea serial, USB u otro. También herramientas como grabadores de memorias o programadores de microcontroladores pueden ser usadas en conjunto con otras herramientas de desarrollo para interconectar el sistema anfitrión y destino.

Las herramientas clave para el diseño de un sistema embebido pueden ubicarse en el sistema destino, en el sistema anfitrión o ser autónomas. Estas herramientas podemos categorizarlas en utilidades, traductores y depuración. Las utilidades son herramientas que facilitan el desarrollo de software y hardware, como editores de texto, sistemas de control de versiones o software para captura de esquemáticos. Las herramientas de traducción convierten código desarrollador en el sistema anfitrión para ser ejecutado en el sistema destino. Por último, las herramientas de depuración permiten rastrear errores y verificar el sistema embebido. (Noergaard, 2005, p. 541)

Diseño electrónico asistido por computadora

También conocido como EDA (del inglés *Electronic Design Automation*) o ECAD (del inglés *Electronic Computer Aided Design*) es una categoría de herramientas enfocadas en el diseño y producción de sistemas electrónicos, abarcando desde el desarrollo de circuitos integrados hasta placas electrónicas. El software de diseño asistido por computadora se ejecuta en un sistema anfitrión y para los sistemas embebidos se enfoca en facilitar dos procesos:

1. Captura de esquemático: es la tarea en la cual el diagrama eléctrico de un circuito es creado con la ayuda de software especializado. Algunas herramientas para la edición de esquemáticos permiten simular el circuito capturado.

2. Diseño de placas electrónicas: es el proceso en el cual se transforma un diagrama eléctrico en un circuito impreso o PCB (del inglés *Printed Circuit Board*). Esta tarea da como resultado los archivos e información necesaria para fabricar un PCBA (del inglés *Printed Circuit Board Assembly*).

Editor de texto o entorno de desarrollo integrado

El código fuente es normalmente escrito en un sistema anfitrión con ayuda de un editor de texto o un entorno de desarrollo integrado mejor conocido como IDE (del inglés *Integrated Development Environment*).

Un IDE es una colección de herramientas, incluido un editor de texto, en una misma interfaz de usuario. Mientras un editor de texto puede utilizarse para escribir cualquier tipo de código, sin importar el lenguaje o plataforma, un IDE es específico para una plataforma y en algunos casos es provisto por una fábrica de hardware (junto con placas de evaluación compatibles), comercializador de sistema operativo o lenguaje de programación (Java, C, etc). (Noergaard, 2005, p. 542)

Herramientas de traducción

Después que el código fuente ha sido escrito necesita ser traducido en código máquina que pueda ejecutar el sistema embebido. Como se muestra en la Figura 7, conforme Massa y Barr (2006) el proceso de convertir código fuente en una imagen binaria ejecutable involucra tres diferentes pasos:

1. Cada archivo con código fuente debe ser compilado o ensamblado en un archivo objeto.
2. Todos los archivos objeto resultantes del primer paso deben ser vinculados para producir un solo archivo objeto llamado el programa reubicable.
3. Las direcciones de memoria física deben ser asignadas según los desfases relativos dentro del programa reubicable en un proceso llamado reubicación.

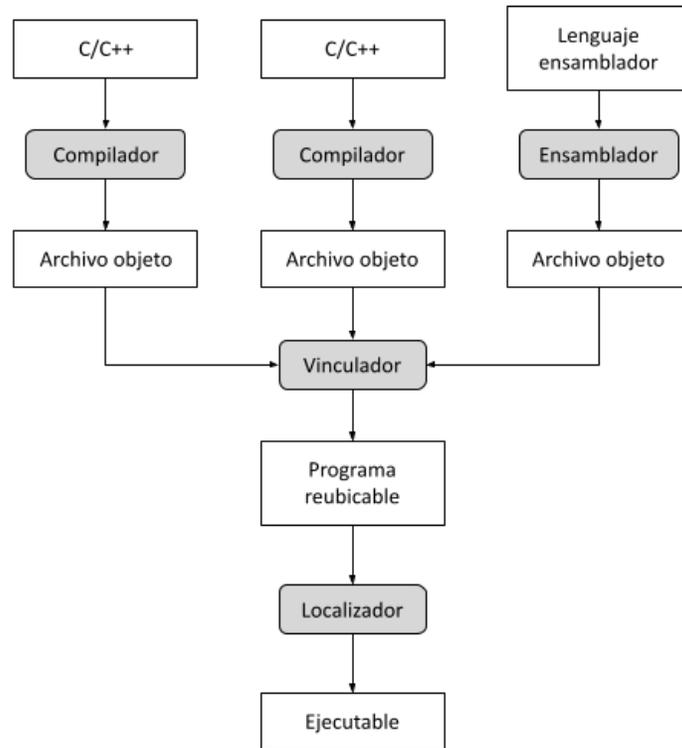


Figura 7. Proceso de traducción de software de un sistema embebido. (Massa y Barr, 2006)

Las herramientas necesarias para este proceso normalmente se ejecutan en un sistema anfitrión y en muchas ocasiones están integradas en el entorno de desarrollo integrado.

El preprocesamiento es un paso opcional que ocurre antes de la compilación del código fuente. Su propósito es organizar y reestructurar el código fuente para facilitar su traducción o interpretación.

Un compilador es un programa de computadora que traduce el código fuente en una secuencia de enunciados o instrucciones en un lenguaje de computadora llamado código objeto o un lenguaje intermedio como ensamblador.

Se le llama compilador cruzado o cross-compiler si el código fuente es traducido a código objeto en una plataforma diferente a la del sistema destino. Para los sistemas embebidos el uso de un compilador cruzado es obligatorio por que en algunas plataformas como los microcontroladores es inconveniente o imposible llevar a cabo un proceso de compilación.

El código resultante del proceso de compilación debe ser combinado de una manera específica antes de poder ser ejecutado en el sistema destino. Al resultado del proceso de vinculación y reubicación se le conoce como ejecutable binario.

Por último, el ejecutable binario debe ser transmitido al sistema destino desde el sistema anfitrión por medio del medio de transmisión que los comunica.

Herramientas de depuración

La depuración es principalmente la tarea de localizar y reparar errores en el sistema embebido. Las herramientas de depuración son una combinación de dispositivos independientes, en el sistema anfitrión y/o en el sistema destino. Las herramientas de depuración más comunes para sistemas embebidos son:

- Interfaz JTAG (del inglés *Joint Test Action Group*). Es un estándar abierto para observar sin obstrucciones la ejecución de software en un sistema embebido. Permite configurar puntos de ruptura para detener la operación de un sistema embebido bajo ciertas condiciones. Además permite leer y escribir registros del procesador, direcciones de memoria RAM o puertos de entrada/salida. Es implementado por medio de un puerto en el sistema destino y un software de depuración en el sistema anfitrión interconectados por una interfaz serial.
- Osciloscopio. Es un instrumento de visualización electrónico para la representación gráfica de señales eléctricas que varían en el tiempo. Permite monitorear dos o más señales simultáneamente y se puede configurar para capturar una señal a partir de una condición específica. Permite verificar la operación de un circuito eléctrico por medio del análisis de señales en buses de comunicación o puertos de entrada/salida.
- Analizador lógico. Es un instrumento que captura múltiples señales digitales para su representación visual e interpretación. Permite capturar los cambios en señales en buses o puertos de entrada/salida para verificar que el software de un sistema embebido opera según lo planeado.
- Multímetro. Es un instrumento que mide diversos parámetros de operación de un circuito como voltaje, resistencia, corriente, etc.

También existe un conjunto de técnicas de depuración manuales que no requieren de equipo especializado, son efectivas y simples pero son más intrusivas que otras herramientas y pueden dificultar la depuración de sistemas en tiempo real. Algunas de estas técnicas son:

- Sentencias de impresión. Consiste en insertar sentencias de impresión para obtener por un puerto de comunicación el valor de variables, registros, etc durante la operación del sistema embebido. Permite verificar la ejecución del código fuente del sistema embebido.
- Volcados. Consiste en volcar información en algún tipo de estructura de almacenamiento durante la operación del sistema embebido. Permite analizar el contenido de la memoria durante la operación del sistema.
- Despliegue rápido. Consiste en apagar y encender LEDs o presentar información en pantallas LCD. Es similar a las sentencias de impresión pero más rápido y menos intrusivo.
- Puertos de salida. Consiste en activar y desactivar puertos de salida en varios puntos de la ejecución del software en el sistema embebido. Se puede consultar mediante un osciloscopio o analizador de señales y así determinar si la operación del sistema es la deseada.

Mientras que algunas de estas herramientas son intrusivas y pueden afectar la operación del sistema embebido otras capturan pasivamente su comportamiento sin obstaculizar su operación. Depurar un sistema embebido usualmente requiere una combinación de estas herramientas para abarcar todos los diferentes tipos de problemas que pueden surgir durante el proceso de desarrollo.

2.3.5 Integración de software y hardware

La fase de integración de software y hardware debe utilizar herramientas especiales para gestionar su complejidad. El proceso de integración de software embebido es un ejercicio de depuración y descubrimiento. Descubrimiento es un término apto porque el equipo de software descubre si en realidad entendió la especificación de hardware provista. (Berger, 2002, p. 25)

El *santo grial* del diseño de sistemas embebidos es que al combinar el primer prototipo de hardware con el software de aplicación, los controladores de dispositivos y el sistema operativo, el diseño funcione perfectamente sin necesidad de modificar la placa electrónica, rediseñar el hardware o reescribir el software.

El escenario anterior es poco probable por diversas razones. Por una parte, la naturaleza en tiempo real de los sistemas embebidos da como resultado un comportamiento complejo y no determinístico que solo puede ser analizado conforme ocurre. Intentar modelar o simular con precisión el comportamiento de un sistema embebido puede tomar mucho más tiempo que la vida útil del producto desarrollado. A pesar de esto, la habilidad del diseñador de sistemas embebidos para encontrar errores en etapas tempranas del proceso de diseño se incrementa conforme se mejoran las herramientas de modelado.

2.3.5 Pruebas de producto y lanzamiento

Las pruebas de producto son especialmente importantes cuando el mal desempeño de un sistema embebido tiene consecuencias de vida o muerte. Mientras que podemos lidiar con las fallas de una computadora personal no podemos permitirnos una falla del software que controla el sistema de emergencia de una central nuclear. Por esta razón, las pruebas y requerimientos de confiabilidad de un sistema embebido son mucho más rigurosos que los de una aplicación para una computadora personal. (Berger, 2002, p. 28)

Sin embargo, las pruebas son mucho más que asegurarse que el software no falla en un momento crítico. Ya que los sistemas embebidos tienen muchas restricciones de diseño para alcanzar sus metas de costos, las pruebas también deben determinar si el desempeño del sistema es el óptimo. Esto es especialmente relevante si el código fue escrito en un lenguaje de alto nivel o el equipo de programación consiste de múltiples desarrolladores. Mientras que en una computadora personal una fuga de memoria puede pasar desapercibida, en un sistema embebido que opera sin detenerse durante varias semanas una fuga de memoria puede ser catastrófica.

Las técnicas de pruebas se pueden agrupar en cuatro modelos: pruebas de caja negra estáticas, pruebas de caja negra dinámicas, pruebas de caja blanca estáticas y pruebas de caja blanca dinámicas. En la Tabla 7 se definen estos modelos de pruebas en una matriz de acuerdo a la visibilidad que tienen del sistema (de caja negra o caja blanca) y el estado del sistema durante las pruebas (estáticas o dinámicas).

Las pruebas de caja negra ocurren cuando no hay visibilidad del funcionamiento interno del sistema. Las pruebas de caja negra se basan en los requerimientos operacionales del sistema al contrario de las pruebas de caja blanca donde se tiene acceso al código, diagramas y demás documentación del sistema. Las pruebas estáticas se llevan a cabo cuando el sistema no está operando al contrario de las dinámicas que se llevan a cabo con el sistema en operación. (Noergaard, 2005, p. 563)

Tabla 7. Matriz de modelos de pruebas. (Noergaard, 2005, p. 564)

	Pruebas de caja negra	Pruebas de caja blanca
Pruebas estáticas	<p>Prueban la especificación del producto:</p> <ol style="list-style-type: none"> 1. A alto nivel buscando problemas, omisiones o descuidos (por ejemplo, simulando ser el usuario final, investigando estándares existentes, explorando soluciones similares, etc). 2. A bajo nivel asegurando su integridad, precisión, exactitud, consistencia, relevancia, viabilidad, etc. 	<p>Metódicamente repasar el hardware y software para encontrar errores.</p>
Pruebas dinámicas	<p>Requieren una definición de las funciones del software y hardware. Estas pruebas incluyen:</p> <ul style="list-style-type: none"> • <i>Pruebas de datos</i>: verificar la información en las entradas y salidas de usuario. • <i>Pruebas de condiciones límite</i>: comprobar la ejecución del sistema al borde de sus límites operacionales. • <i>Pruebas de entrada</i>: introducir datos inválidos o vacíos al sistema. • <i>Pruebas de estado</i>: probar los estados y transiciones entre estados del software. 	<p>Probar el sistema operando mientras se examina código, diagramas, etc.</p> <p>Basados en el conocimiento a detalle de la operación del sistema comprobar directamente detalles de alto y bajo nivel, estructuras de datos y vaciados de memoria.</p> <p>Buscar errores en referencias o declaraciones de datos, errores en cálculos, comparaciones erróneas, errores de control de flujo, parámetros de subrutinas equivocadas, errores de entrada/salida, etc.</p>

Dentro de cada uno de estos modelos, se pueden clasificar las pruebas en:

- Pruebas unitarias o de modelo: son pruebas incrementales de elementos individuales del sistema.
- Pruebas de compatibilidad: son pruebas de que los elementos del sistema no causan problemas a otros elementos.
- Pruebas de integración: son pruebas incrementales de integración de elementos.
- Pruebas de sistema: son pruebas del sistema embebido completo con todos sus elementos integrados.
- Pruebas de manufactura: aseguran la fabricación del sistema no causa nuevos errores.

De todos estos modelos y tipos de pruebas se puede definir un conjunto efectivo de casos de prueba para verificar que el sistema cumple las especificaciones y requerimientos del producto. Cabe mencionar que también existen metodologías específicas para ciertos tipos de pruebas así como procesos de prueba completos definidos en diversos estándares de pruebas industriales, como la norma ISO 9000 para el aseguramiento de la calidad, el CMM (del inglés *Capability Maturity Model*) o la norma ANSI/IEEE 829 para el software y documentación de prueba.

El proceso de pruebas es crucial como cualquier otro paso del diseño de un sistema embebido y es la última línea de defensa entre el lanzamiento de un producto sin problemas y un desastre financiero importante.

2.3.6 Mantenimiento y actualizaciones

Después de la puesta en operación de un sistema embebido ciertas tareas deben llevarse a cabo como capacitación a usuarios, soporte técnico, actualizaciones, mantenimiento o corrección de errores por nombrar algunas. Las responsabilidades del equipo de ingeniería duran hasta el fin de la vida útil del dispositivo y no cuando el sistema embebido ha sido puesto a operar en campo. (Noergaard, 2005, p. 566)

Capítulo 3. Diseño del receptor de alertas

El diseño de sistemas embebidos es un campo amplio y de auge en los últimos años. Este requiere de un conjunto de diversas habilidades incluyendo el diseño eléctrico, desarrollo de firmware, diseño de placas electrónicas e integración de sistemas.

A lo largo de este capítulo se aplicarán los conocimientos adquiridos en la Facultad de Ingeniería junto con la investigación documental y metodología expuesta en este trabajo para la construcción de un receptor de alertas de riesgo

En este capítulo se documenta:

- La aplicación de la metodología para diseño de sistemas embebidos descrita en el Capítulo 2.
- La fabricación y operación de un receptor de alertas.
- Las bases técnicas para el desarrollo de productos o servicios que aprovechen los sistemas de alertamiento de riesgos que utilizan el protocolo SAME.

3.1 Especificación de producto

El propósito de este trabajo de tesis es analizar el protocolo SAME y aplicar el diseño de sistemas embebidos para la creación de un receptor de alertas de riesgo. Partiendo de este propósito podemos definir la necesidad que satisface el sistema embebido a diseñar como:

Difundir las alertas de riesgo emitidas por las autoridades para proteger vidas y bienes.

3.1.1 Requerimientos operacionales

En esta sección se desarrollan los requerimientos operacionales del receptor de alertas que aplican de acuerdo al propósito de este trabajo de tesis.

Misión

Para satisfacer la necesidad planteada, la misión primaria del sistema embebido es:

Recibir alertas de riesgo emitidas mediante el protocolo SAME y difundirlas por medios visuales, auditivos o electrónicos.

Las misiones secundarias del sistema embebido son:

1. Analizar el protocolo SAME y los sistemas de alertamiento que lo usan.
2. Documentar las bases técnicas para el desarrollo de productos o servicios que incrementen el aprovechamiento y la cobertura de los sistemas de alertamiento de riesgos.

Parámetros físicos y de desempeño

Para que el sistema embebido cumpla las misiones requeridas sus funciones deben incluir:

1. Sintonizar un canal de la Banda del clima para captar alertas de riesgo transmitidas por las autoridades.
2. Procesar los mensajes SAME recibidos..
3. De acuerdo al tipo de evento recibido y su cobertura geográfica transmitir la alerta por medios visuales, auditivos o eléctricos.
4. Transmitir digitalmente los mensajes recibidos para su análisis.

Requerimientos de uso

El objetivo de las alertas emitidas por las autoridades es reducir las afectaciones que pueden sufrir vidas y bienes en hogares, oficinas, comercios o industria ante ciertos eventos. Algunos casos de uso del receptor de alertas son:

1. Notificar de alertas recibidas audiovisualmente.
 - a. Integrándose en sistemas de audio ambiental o de alarma.
 - b. En nuevos sistemas con capacidades personalizadas.
2. Interconectarse con equipo en oficinas o comercios (como puertas automáticas, elevadores, escaleras mecánicas, etc) para alistarse ante una posible evacuación.
3. Preparar automáticamente maquinaria y equipo industrial de peligros y así reducir afectaciones a sus operadores o a las máquinas mismas.
4. Retransmitir la alerta en medios digitales para amplificar su alcance y permitir su análisis.

Factores de efectividad

- La tasa de mensajes SAME captados exitosamente durante el periodo de prueba debe ser de un 99%.
- El costo de un receptor de alertas debe ser igual o menor al de alternativas comerciales con prestaciones equivalentes.
- El tiempo de mantenimiento programado al sistema debe ser igual o menor a una hora al año.

Ambiente

El sistema embebido debe operar en un entorno protegido de los elementos naturales como la lluvia y viento y ser capaz de soportar el rango de temperatura habitual en México.

3.1.2 Plan de mantenimiento

Como es habitual en los sistemas embebidos electrónicos el mantenimiento se espera sea mínimo. Para esto, en la sección 3.7 se desarrolla un plan de mantenimiento preventivo y correctivo para cada caso de uso del receptor de alertas.

Se debe contemplar en el diseño que el sistema embebido genere un tipo de alerta si detecta errores en su funcionamiento como la falta de mensajes de prueba periódicos recibidos con éxito. Una alerta de este tipo dispararía un evento de mantenimiento correctivo del sistema embebido.

3.1.3 Análisis de factibilidad

El propósito del estudio de factibilidad es asegurar que con los medios técnicos disponibles el producto a diseñar va a satisfacer la necesidad planteada.

Para este proyecto, satisfacer la necesidad técnicamente es factible ya que existen varios productos comerciales que emiten una alerta al recibir las advertencias de riesgos transmitidas por las autoridades. En esta sección se identifican y evalúan diferentes alternativas para el diseño del receptor de alertas.

Por otra parte, la aportación de este trabajo de tesis es documentar las bases técnicas para el desarrollo de productos o servicios que incrementen el aprovechamiento y la cobertura de los sistemas de alertamiento de riesgos. Por esta razón se puede considerar este trabajo de tesis en su totalidad como un estudio de factibilidad técnica.

Ciclo de vida operacional

El receptor de alertas deberá tener una vida operativa de varios años como habitualmente sucede con los sistemas embebidos.

Definición del problema

Identificando el desafío más importante del sistema embebido podemos definir el problema como:

Captar la Radio del clima y procesar los mensajes SAME recibidos.

Identificar alternativas factibles

Existen tres posibles fórmulas tecnológicas para en un sistema embebido sintonizar la radio del clima y procesar los mensajes SAME recibidos:

- A. Circuito de componentes discretos para sintonizar radio FM y un algoritmo de software en un microcontrolador para procesar los mensajes SAME capturados.
- B. Circuito integrado para sintonizar radio FM y un algoritmo de software en un microcontrolador para procesar los mensajes SAME capturados.
- C. Circuito integrado para sintonizar la radio del clima y decodificar el protocolo SAME.

Seleccionar el criterio de evaluación

Siendo el propósito del sistema embebido proteger vidas y bienes, los criterios a evaluar las posibles soluciones técnicas, en orden de importancia, son:

1. *Desempeño*: que la solución tecnológica cumpla su función.
2. *Confiabilidad*: la solución técnica con la menor probabilidad de que sus funciones se interrumpen.
3. *Costo*: la solución tecnológica con el menor costo posible.

Aplicar técnicas de modelación

Una matriz de decisión es el modelo elegido para evaluar las alternativas técnicas para resolver el problema planteado.

Según Cortesi (2010), una matriz de decisión permite identificar, analizar y calificar las fortalezas de las relaciones entre conjuntos de información. Es frecuente su uso en evaluación de posibles soluciones a un problema y para seleccionar productos o servicios por sus cualidades.

Habiendo identificado las alternativas técnicas y el criterio de evaluación los siguientes pasos para aplicar la metodología de las matrices de decisión es asignar pesos a los criterios de evaluación, definir un sistema de calificación y calificar cada alternativa respecto a los criterios de evaluación.

El último paso de esta técnica de modelación es multiplicar la calificación de cada criterio de evaluación por el peso del mismo y sumar el resultado por cada alternativa. La alternativa con la mayor suma de calificaciones es la más adecuada para resolver el problema en términos de los criterios de evaluación dispuestos.

Generar datos de entrada

De acuerdo al modelo de las matriz de decisión, los datos de entrada son las calificaciones de cada alternativa técnica y los pesos de los criterios de evaluación. Los requerimientos para que los datos de entrada sean válidos son:

- Para los criterios de evaluación: pesos numéricos del 1 al 3 de menor a mayor importancia.
- Para las calificaciones: pesos numéricos del 1 al 3 según si la calificación es pobre, promedio o buena.

En la Tabla 8 se muestran las calificaciones correspondientes a las alternativas técnicas respecto a los criterios de evaluación junto a sus pesos numéricos entre paréntesis.

Tabla 8. Matriz de calificaciones de las alternativas tecnológicas.

	Opción A	Opción B	Opción C
Desempeño (3)	Promedio (2)	Promedio (2)	Bueno (3)
Confiabilidad (2)	Pobre (1)	Promedio (2)	Buena (3)
Costo (1)	Promedio (2)	Promedio (2)	Pobre (1)

Manipular el modelo

Multiplicando el peso de cada criterio de evaluación por las calificaciones de cada posible fórmula tecnológica para diseñar el producto, se obtienen los resultados de la Tabla 9.

Tabla 9. Matriz de decisión de las alternativas tecnológicas con resultados.

	Opción A	Opción B	Opción C
Desempeño	$2 * 3 = 6$	$2 * 3 = 6$	$3 * 3 = 9$
Confiabilidad	$1 * 2 = 2$	$2 * 2 = 4$	$3 * 2 = 6$
Costo	$2 * 1 = 2$	$2 * 1 = 2$	$1 * 1 = 1$
Total	10	12	16
Posición	3	2	1

Como resultado de aplicar el modelo de las matrices de decisión, en este trabajo se elige la fórmula tecnológica C para el diseño del sistema embebido: Circuito integrado para sintonizar la radio del clima y decodificar el protocolo SAME.

3.1.4 Plan de administración de la ingeniería del sistema

El Plan de administración de la ingeniería del sistema (SEMP) describe los procesos técnicos, organización, canales de comunicación, organigrama o cualquier otro recurso para lograr el objetivo del proceso de diseño del producto. El SEMP provee la estructura para realizar el trabajo apropiado para completar el ciclo de vida del producto y la información necesaria para evaluar el progreso técnico. (Solberg, 2018)

Para el desarrollo del receptor de alertas de riesgos emplearemos la metodología descrita en el Capítulo 2 y que consta de siete pasos:

1. Especificación de producto.
2. Distribución de software y hardware.
3. Iteración y refinamiento de la distribución.
4. Diseño de software y hardware.
5. Integración de software y hardware.
6. Pruebas de producto y lanzamiento.
7. Mantenimiento y actualizaciones.

En la Tabla 10 se muestra el plan de resultados esperados de cada etapa del proyecto.

Tabla 10. Plan de ingeniería para el desarrollo del receptor de alertas.

Etapas	Progreso
Especificación de producto	Requerimientos operacionales, plan de mantenimiento, análisis de factibilidad y SEMP
Distribución de software y hardware	Arquitectura del sistema
Iteración y refinamiento de la distribución	Prototipo experimental de hardware y software
Diseño de software y hardware	Manufactura de hardware y programación de software
Integración de software y hardware	Receptor de alertas operativo
Pruebas de producto y lanzamiento	Pruebas estáticas, dinámicas, de campo, y de mantenimiento del receptor de alertas
Mantenimiento y actualizaciones	Monitoreo de desempeño, mantenimiento preventivo-correctivo y retroalimentación al diseño del hardware-software

En términos de organización del tiempo, en la Tabla 11 se muestra el cronograma planeado para el diseño del receptor de alertas. Las siete etapas de la metodología se ubican en las filas de la tabla y las columnas representan los meses siendo las celdas marcadas en color negro la agenda planeada para llevar a cabo la etapa correspondiente.

Tabla 11. Cronograma de etapas de diseño del receptor de alertas.

Etapa	Mes														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Especificación de producto	■														
Distribución de software y hardware	■														
Iteración y refinamiento de la distribución		■													
Diseño de software y hardware			■	■	■										
Integración de software y hardware						■									
Pruebas de producto y lanzamiento							■	■	■	■	■	■	■		
Mantenimiento y actualizaciones													■	■	■

La etapa con más duración es la de pruebas con el propósito de obtener la mayor cantidad de datos posibles de la operación del prototipo construido, seguida del diseño que involucra tiempos de manufactura y logística. Al final, en la etapa de mantenimiento y actualizaciones se contempla el análisis y retroalimentación del proceso de diseño a partir de los resultados de la etapa de pruebas.

3.2 Distribución de software y hardware

Por medio de una propuesta de arquitectura para el sistema en esta sección se documenta la distribución inicial del software y hardware. Además, el diagrama de arquitectura del sistema será una útil herramienta durante el diseño de software y hardware.

Los sistemas embebidos dependen críticamente de su hardware. Si el hardware es inestable, el software parecerá poco confiable y con errores. En este proyecto se diseñará primero el hardware para reforzar la idea de que el producto depende de las características del hardware respecto a desempeño y confiabilidad.

3.2.1 Hardware

El elemento clave del sistema embebido a diseñar es el circuito integrado para sintonizar la radio del clima y decodificar el protocolo SAME. El Si4707 en palabras de su fabricante Silicon Laboratories (2017) es *el primer receptor de radio del clima que incluye un procesador específico para decodificar el protocolo SAME* y será la pieza clave de nuestro sistema embebido.

Las principales características del Si4707 son:

- Capta el radio del clima (162.4 a 162.55 MHz).
- Procesador de mensajes SAME.
- Detección de tono de alerta a 1050 Hz.
- Sintetizador de frecuencia con VCO integrado.
- Sintonización de radio del clima digital.
- Rejo de referencia programable.
- Interfaz de comunicación I2C, SPI o 3-wire.
- Voltaje de alimentación de 2.7 a 5.5 V.

La documentación provista por el fabricante es:

- Si4707-B20: Hoja de especificaciones del Si4707.
- AN332: Guía de programación para la familia Si47XX.
- AN383: Guías para antenas, esquemáticos, disposición y diseño de circuitos para la familia Si47xx

De acuerdo al Diagrama de Bloques Funcional (*Functional Block Diagram*) de la hoja de especificaciones Si4707-B20 (Silicon Laboratories, 2008, p. 1), se deben tomar en cuenta las siguientes interfaces y módulos en la arquitectura de hardware:

1. Entrada de antena FM
2. Salida de audio estéreo
3. Fuente de energía de 2.7 a 5.5 V
4. Interfaz de control I2C, SPI o 3-wire.

Partiendo de lo anterior y analizando la especificación de producto, en la arquitectura de hardware también se deben considerar las siguientes interfaces y módulos:

1. Microcontrolador de 8 bits por su accesibilidad, confiabilidad y ser ideales para aplicaciones de control en tiempo real.
2. Interfaz I2C, SPI o 3-wire para comunicación con el circuito integrado Si4707.
3. Interfaz USART para comunicación con otros dispositivos electrónicos.
4. Múltiples puertos digitales de salida.

A partir de las interfaces y módulos que requieren el circuito integrado Si4707 y el microcontrolador de acuerdo a la especificación de producto, se propone la arquitectura de hardware descrita en la Figura 8.

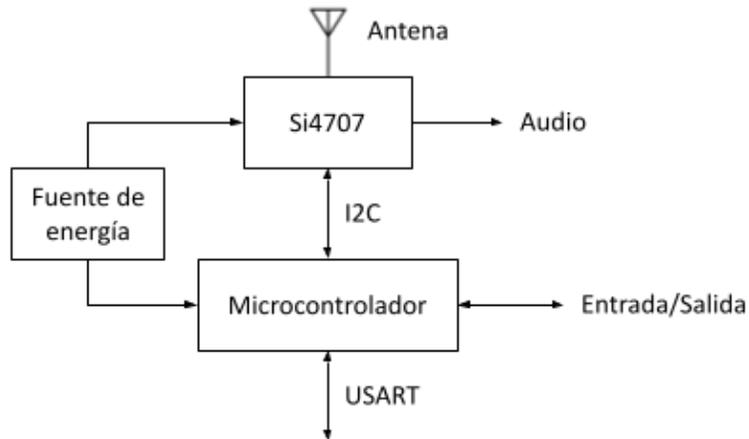


Figura 8. Arquitectura de hardware.

Por último, se elige operar el sistema a 3.3 V por ser conveniente para integrarlo con interfaces eléctricas-electrónicas o sistemas de respaldo de energía tanto modernos como convencionales.

3.2.2 Software

Analizando la especificación del producto y la arquitectura de hardware, se propone la arquitectura de software descrita en la Figura 9.

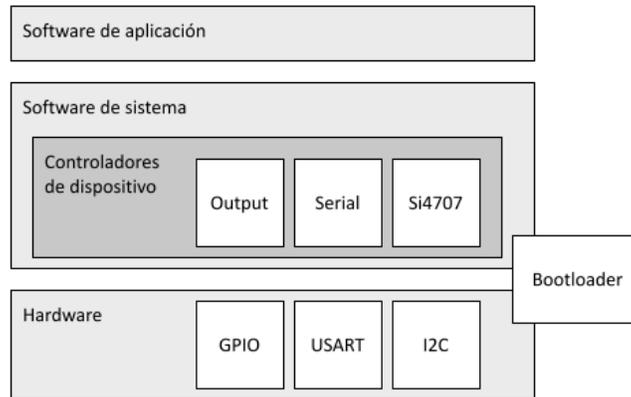


Figura 9. Arquitectura de software.

Es importante notar que en la arquitectura de software no se plantea utilizar un sistema operativo y software intermedio debido a que el sistema embebido tiene solo una aplicación.

Los componentes de software que propone la arquitectura son:

- Output: Facilita a la aplicación accionar dispositivos eléctricos de alerta.
- Serial: Permite a la aplicación comunicarse con otros dispositivos electrónicos.
- Si4707: Hace posible la interacción con el circuito integrado receptor de mensajes SAME.

Además, existe una pieza de código llamada *Bootloader* que permite la rápida descarga de programas en el microcontrolador sin utilizar programador externo. La descarga se lleva a cabo por el puerto USART del microcontrolador y puede funcionar con convertidores USART-USB. (Wiki-Robotics, 2007)

3.3 Iteración y refinamiento de la distribución

Para evaluar el desempeño de la distribución propuesta se construirá un prototipo del sistema embebido por medio de tarjetas de desarrollo. El prototipo también proveerá un ambiente conveniente para el diseño y depuración de software hasta que el sistema de hardware real esté disponible.

Se plantea utilizar las siguientes tarjetas de desarrollo para construir el prototipo del sistema embebido:

SparkFun Weather Band Receiver Breakout

La tarjeta receptora de radio del clima de SparkFun integra un circuito integrado Si4707 y un amplificador de audio en un circuito impreso. Su voltaje de alimentación son 3.3 V, incluye un conector de audio y expone los pines de la interfaz I2C del Si4707. (SparkFun, s.f. b)

Arduino Pro Mini 328 - 3.3V/8MHz

La tarjeta de desarrollo Arduino Pro Mini está basada en el microcontrolador ATmega328 de 8 bits, con una velocidad de 8 MHz, 32 kB de memoria de programa y 2 kB de memoria RAM. Cuenta con 14 pines de entradas o salidas digitales, 6 entradas analógicas y un botón de reinicio. Opera a 3.3 V e incluye un conector de 6 pines para emplear un cable FTDI para suministrar energía y establecer comunicación con el circuito. (SparkFun, s.f. a)

Las características de estas tarjetas de desarrollo que justifican su elección son:

Proveen las funciones necesarias del sistema embebido

De acuerdo a los parámetros físicos y de desempeño definidos en la sección 3.1.1 (requerimientos operacionales), la tarjeta receptora de radio del clima puede llevar a cabo las siguientes funciones:

1. Sintonizar un canal de la Banda del clima para captar alertas de riesgo transmitidas por las autoridades.
2. Procesar los mensajes SAME recibidos.

Y la tarjeta de desarrollo Arduino Pro Mini puede llevar a cabo las siguientes operaciones con los mensajes SAME recibidos:

1. De acuerdo al tipo de evento recibido y su cobertura geográfica transmitir la alerta por medios visuales, auditivos o eléctricos.
2. Transmitir digitalmente todos los mensajes recibidos para su análisis.

Son hardware de fuente abierta (OSHW en inglés)

El diseño de ambas tarjetas está disponible públicamente para que cualquier persona lo pueda estudiar, modificar, distribuir, materializar y vender. Gracias a estas libertades se pueden facilitar subsecuentes etapas del proceso de diseño del sistema embebido. Además, ambas tarjetas las acompaña software libre que podemos utilizar para agilizar la construcción del prototipo y cómo inicios del software del sistema embebido.

En la Figura 10 se muestra el diagrama esquemático del prototipo que representa la interconexión de las tarjetas de desarrollo elegidas para su construcción.

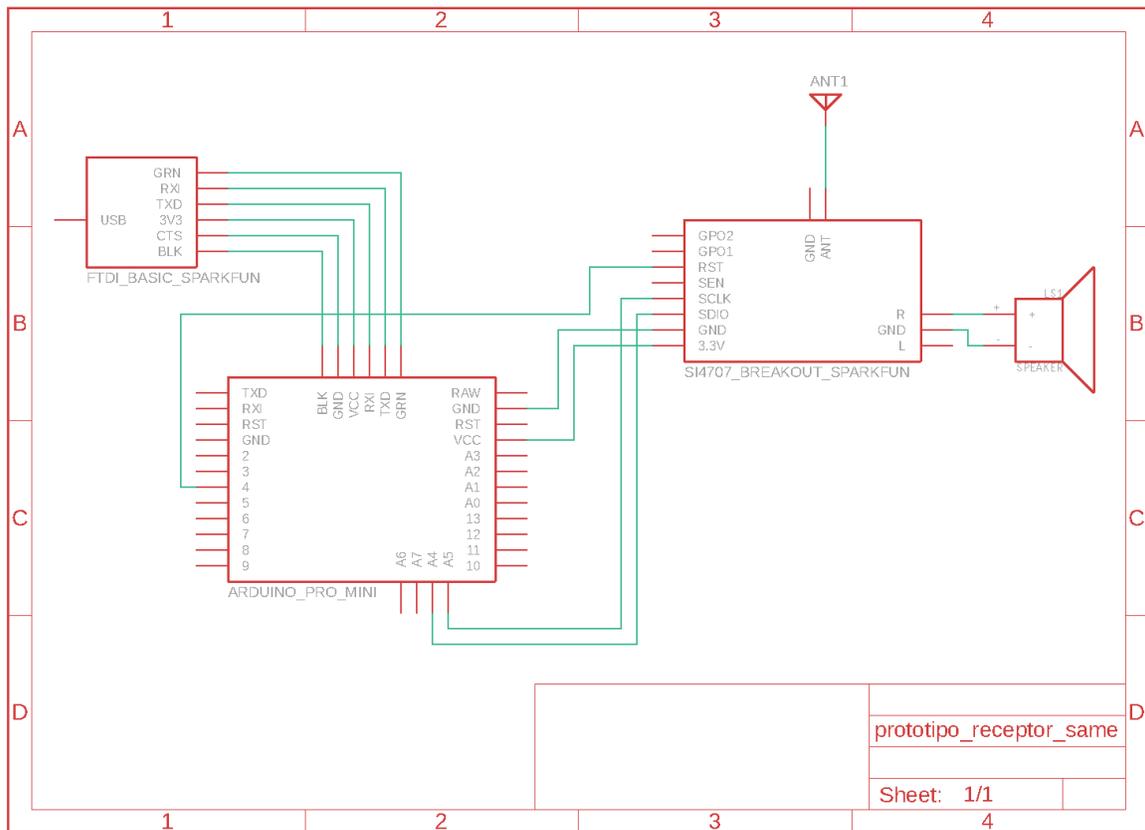


Figura 10. Diagrama esquemático del prototipo receptor de alertas SAME.

El prototipo se construirá en un tablero de circuitos y se diagnosticara por medio de comunicación serie en el puerto USB de una computadora personal. Para el desarrollo de software se utilizará el ambiente Arduino combinado con su Bootloader en el microcontrolador del sistema embebido.

En la Figura 11 se presenta el prototipo construido, los módulos en el tablero de circuitos son (de izquierda a derecha): adaptador USART-USB, Arduino Pro Mini y SparkFun Weather Band Receiver Breakout.

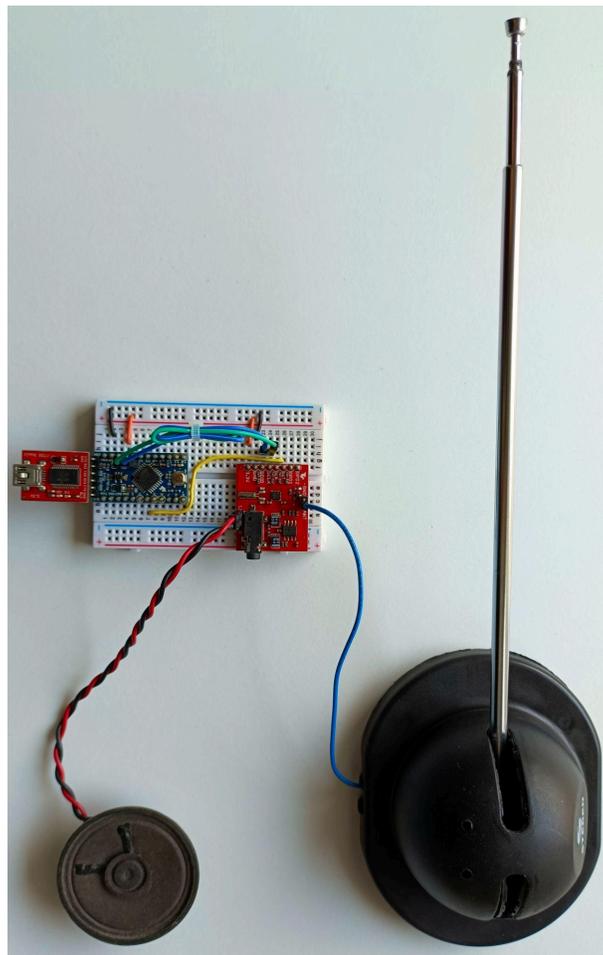


Figura 11. Prototipo del sistema embebido.

El software de aplicación del prototipo está basado en el código de demostración de la tarjeta receptora de radio del clima de SparkFun de Jim Lindblom (2012) y mejorado por Derek Chafin (2014). El software hace uso de las bibliotecas del proyecto Arduino para los controladores de dispositivos Output, Serial e I2C.

Los archivos fuente del diagrama esquemático y software del prototipo pueden encontrarse en la carpeta *prototipo* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

El resultado de las pruebas de recepción de mensajes SAME se lista a continuación (capturado por medio del puerto serial del prototipo y el puerto USB de una computadora personal):

```
Si4707 Weather Band Example Code Menu
```

```
Weather Band: (162.4 - 162.55 MHz)
```

```
=====
```

- u) Fine tune up (+2.5kHz)
- d) Fine tune down (-2.5kHz)
- U) Coarse tune up (+25kHz)
- D) Coarse tune down (-25kHz)
- r) Get RSSI
- s) Get SNR
- o) Get Frequency offset
- f) Get Tune Frequency
- m) Mute audio output
- M) Un-mute audio output
- +) Increase volume
-) Decrease volume
- h) Re-print this menu

```
Successfully connected to Si4707
```

```
Tune Success!
```

```
Frequency = 162.5500 MHz
```

```
Frequency offset = -2
```

```
RSSI = 7
```

```
SNR = 14
```

```
SAME state = Preamble detected
```

```
SAME state = Receiving SAME header message
```

```
SAME state = SAME header message complete
```

```
SAME message = -CIV-RWT-000000+0300-832300-XDIF/005-...!.
```

```
SAME state = Preamble detected
```

```
SAME state = Receiving SAME header message
```

```
SAME state = SAME header message complete
```

```
SAME message = -CIV-RWT-000000+0300-832300-XDIF/005-...!.
```

```
SAME state = Preamble detected
SAME state = Receiving SAME header message
SAME state = SAME header message complete
SAME message = -CIV-RWT-000000+0300-832300-XDIF/005-.....
SAME state = Preamble detected
SAME state = End of message
SAME state = Preamble detected
SAME state = End of message
SAME state = Preamble detected
SAME state = End of message
```

Con estos resultados se concluye que la arquitectura de software y hardware propuesta cumple con la misión de recibir mensajes SAME y por lo tanto es un punto de partida adecuado para el desarrollo del sistema embebido.

3.4 Diseño detallado de software y hardware

En esta sección se concreta el diseño del sistema embebido a partir de la distribución software y hardware. Las herramientas de desarrollo elegidas para la implementación del sistema embebido son:

- Diseño electrónico asistido por computadora: EAGLE 9.3
- Editor de texto: EMACS 24
- Bibliotecas de software y herramienta de traducción: Arduino 1.8
- Sistema de control de versiones: Git 2.7
- Transferencia de programas entre anfitrión y destino: AVRDUDE 6.3
- Herramientas de depuración: osciloscopio, multímetro y adaptador USART-USB

Los criterios de selección de las herramientas de software son que tienen la capacidad necesaria para el diseño del sistema embebido y que están disponibles de manera gratuita o son software libre.

Para la selección de las herramientas de hardware el criterio de elección fue su accesibilidad y costo además de su capacidad para el desarrollo del sistema embebido.

En las siguientes subsecciones se describe el proceso de diseño del hardware y software del sistema embebido. El diseño del hardware y software se puede consultar a detalle en los Apéndices B y C. Además, todo el diseño e implementación del receptor de alertas puede encontrarse en repositorio de Git del proyecto que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

3.4.1 Hardware

Analizando el diagrama de arquitectura de hardware existen dos importantes secciones del sistema pendientes de detallar: energía y entrada-salida.

Para satisfacer los requerimientos operacionales del sistema embebido respecto a la transmisión de alertas e interconexión con otros equipos, se propone que el diseño del sistema embebido incorpore las siguientes interfaces de entrada-salida:

- Salida de audio para sistemas de difusión de audio.
- Puerto de comunicación serie para la configuración del sistema embebido, la retransmisión de mensajes recibidos o integración en redes de sistemas embebidos.
- Salidas digitales para la integración con módulos o sistemas eléctricos.
- Relevador AC/DC para controlar otros dispositivos eléctricos.
- Salidas para leds externos de estado.
- Entrada para botón de prueba.

Para proveer la energía necesaria para los componentes del sistema embebido se propone contar con entradas de energía de 12V y 5V para integrarse tanto en sistemas basados en baterías de plomo como en sistemas más modernos. También el sistema contará con una entrada secundaria de 3.3 V con la desventaja de si se utiliza el sistema embebido no podrá operar el relevador AC/DC.

El proceso para el diseño detallado del hardware se divide en dos etapas: captura de diagrama esquemático y diseño de placa electrónica. Concluido el proceso de diseño se llevará a cabo la fabricación de receptores para la etapa de integración de hardware y software.

Captura de diagrama esquemático

Para organizar la captura del esquemático el diseño del hardware se dividirá en ocho bloques de acuerdo a su componente principal o función en el sistema.

Como puede verse en la Figura 12 y 13, el esquemático se dividirá en dos hojas tamaño A4 con un sistema de coordenadas cartesiano como método para referenciar secciones. Cada hoja posee cuatro filas marcadas con letras de la A a la D y seis columnas numeradas del uno al seis.

Los bloques del sistema y su ubicación en el diagrama esquemático son:

1. Microcontrolador: hoja 1, coordenadas C1 a D2.
2. Receptor mensajes SAME: hoja 1, coordenadas A3 a B6.
3. Amplificador de audio: hoja 1, coordenadas C4 a D6.
4. Conectores de programación, comunicación, interconexión y energía auxiliar: hoja 1, coordenadas A1 a B2.
5. Fuente de energía: hoja 2, coordenadas A2 a A5.
6. Relevador AC/DC: hoja 2, coordenadas C5 a C6.
7. Leds internos: hoja 2, C1 a C2.
8. Conectores para leds y botón externo: hoja 2, coordenadas C3 a C4.

El diseño de los bloques del microcontrolador, receptor de mensajes SAME y amplificador de audio toma como referencia el diagrama esquemático de las tarjetas de evaluación usadas en la sección 3.3.

El diseño del relevador AC/DC contempla un transistor bipolar tipo NPN en un arreglo tal que alimenta la bobina del relevador con una señal de 5V de acuerdo a una salida digital del microcontrolador de 3.3 V.

La fuente de energía es un diseño de dos etapas. La primera etapa es una fuente de regulación conmutada en una configuración que provee 5 V y 600 mA a partir de una entrada en el rango de 6 a 36 V (Microchip, 2015). La segunda etapa es una fuente de regulación lineal que provee 3.3 V y 300 mA (Texas Instruments, 2015).

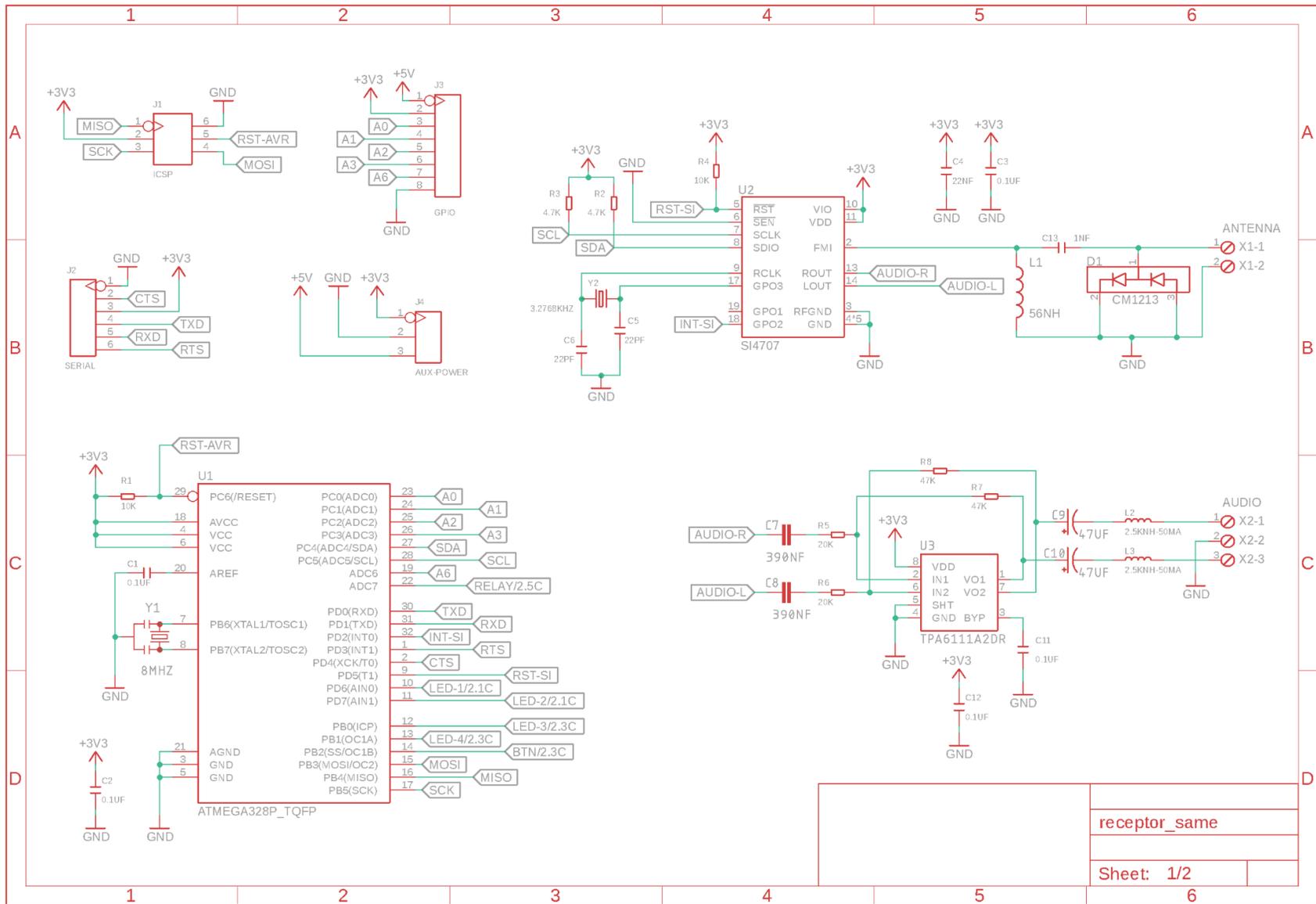


Figura 12. Hoja 1 del diagrama esquemático.

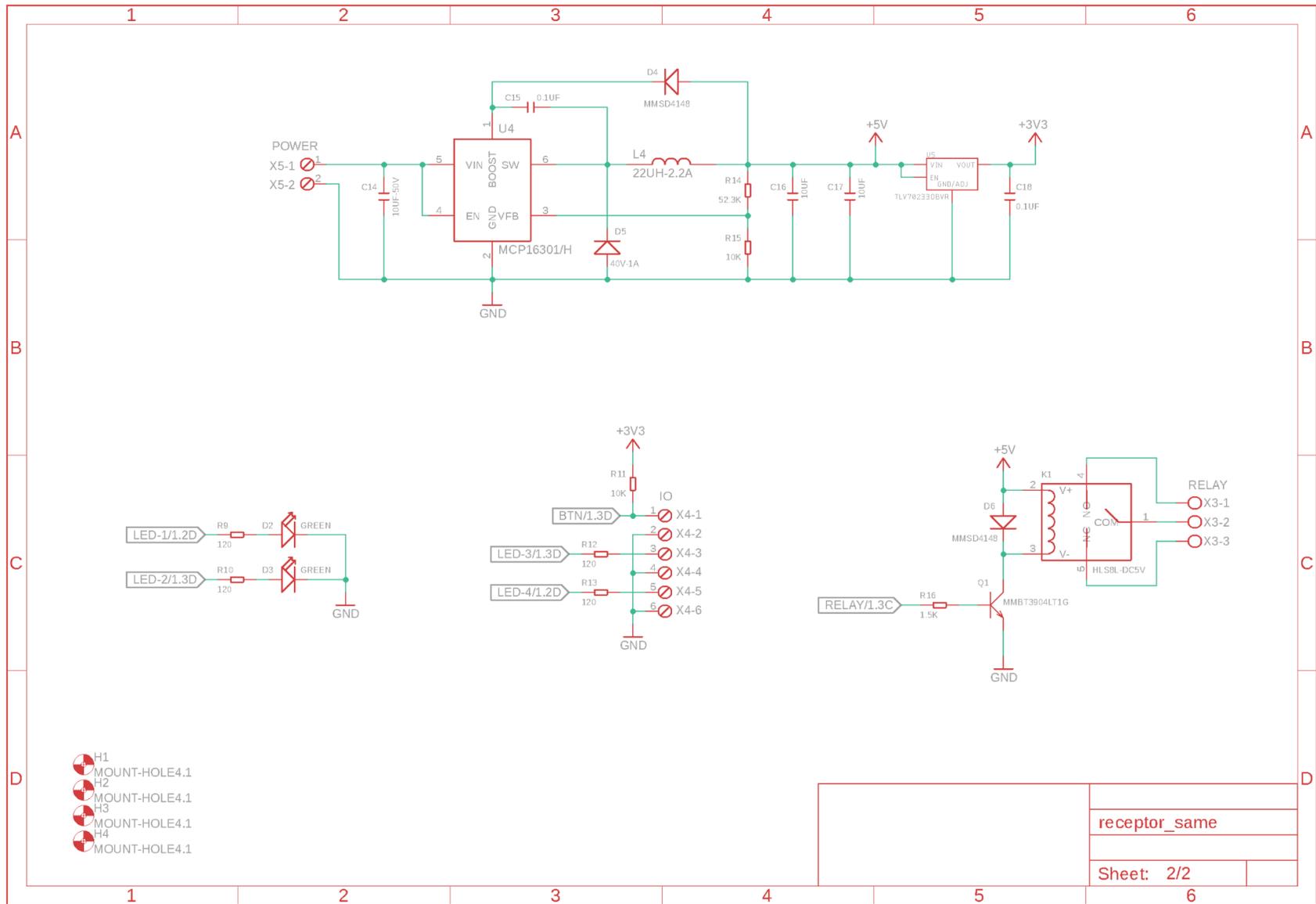


Figura 13. Hoja 2 del diagrama esquemático.

Diseño de placa electrónica

Partiendo del esquemático capturado se diseñó una placa electrónica de 61 x 79 mm con puntos de montaje de 4 mm en sus cuatro esquinas. Los principales parámetros de diseño para la placa electrónica serán:

1. 2 caras
2. Tamaño de perforación mínimo de 0.3 mm
3. Espaciado y ancho de pista mínimo de 6 mil

En la Figura 14 se expone una representación de la placa electrónica diseñada que muestra en color rojo la cara superior de cobre, en color azul la cara inferior de cobre, en color verde las vías y agujeros pasantes para componentes, y color en gris las impresiones para sobre la máscara de soldadura.

En el perímetro de la placa se colocaron los conectores de tipo borne designados X y pines, designados en el circuito y lista de partes como X y J respectivamente. Las funciones de cada conector son:

- J1: Puerto de programación ICSP
- J2: Puerto de comunicación serie
- J3: Puerto de expansión analogico digital
- J4: Entrada de energía auxiliar
- X1: Antena receptora
- X2: Salida de audio
- X3: Relevador
- X4: Leds y botón externos
- X5: Entrada de energía

La configuración de cada conector se puede consultar en el Apéndice B. Además de los conectores en el perímetro hay un par de LEDs D1 y D2 para indicar el estado del sistema, durante el diseño del software se definirá su labor específica.

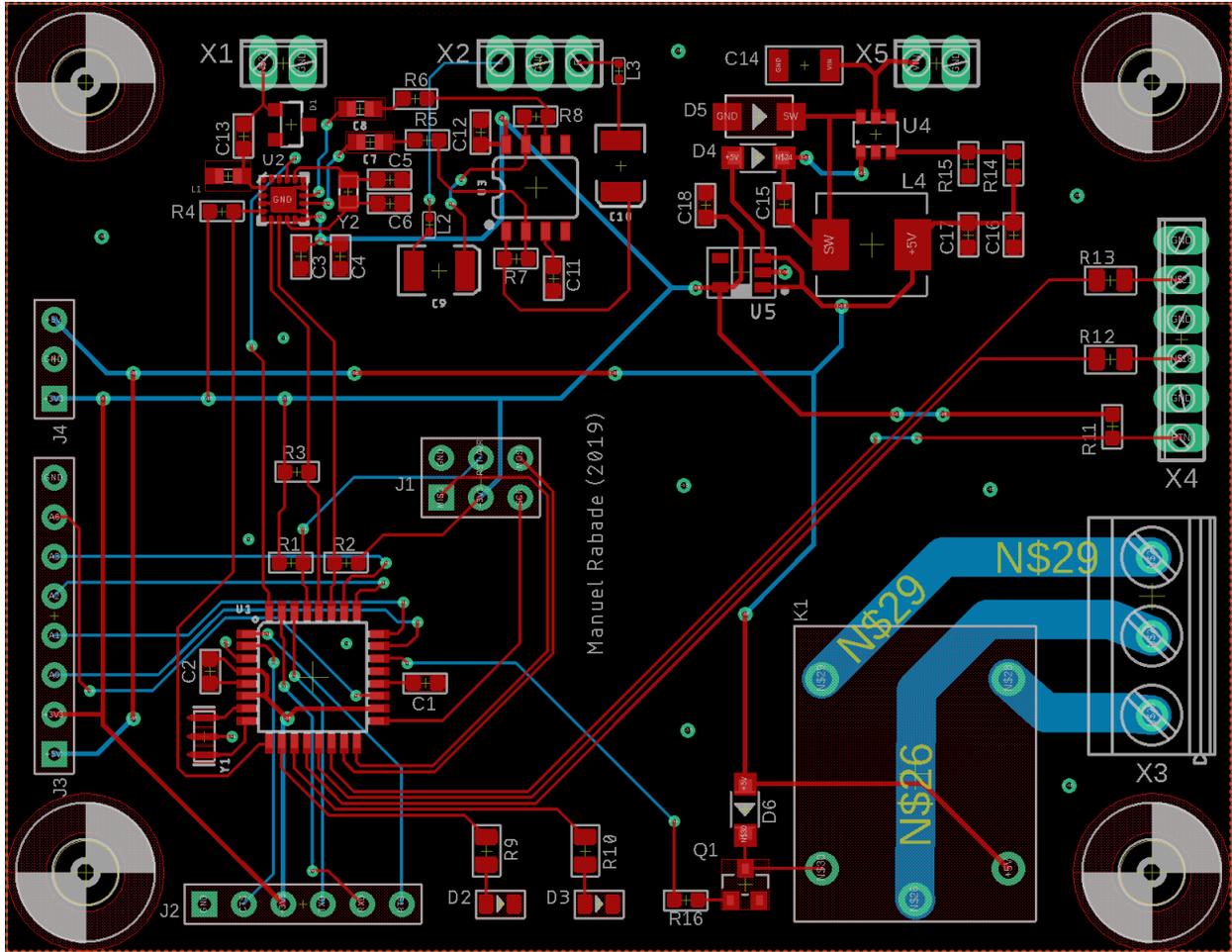


Figura 14. Placa electrónica.

En la placa electrónica por cercanía con sus respectivos conectores los bloques del sistema se ubican en:

- Microcontrolador: inferior izquierda.
- Receptor mensajes SAME: superior izquierda.
- Amplificador de audio: superior centro.
- Fuente de energía: superior derecha.
- Relevador AC/DC: inferior derecha.

En el Apéndice B se puede consultar el diseño de todas las capas de la placa electrónica.

Fabricación de prototipos

Como proveedor de manufactura de placa electrónica y ensamble de componentes se elige el servicio *Fusion PCB & PCB Assembly & Flexible PCB* (Seeed Studio, s.f.) siendo los criterios para su elección los que a continuación se mencionan.

- Costo atractivo por una oferta de ensamble gratis en la primer orden de cinco piezas
- Biblioteca de componentes universales con símbolos y huellas.
- Proceso integral de fabricación de circuito impreso y ensamble de componentes.
- Tiempo de entrega de siete días hábiles para placas electrónicas con solo componentes de la biblioteca de partes universales.

Para el ensamble de componentes se registró en el atributo MPN de cada elemento del diagrama esquemático su número de parte de la biblioteca de componentes universales de Seeed Studio. Para componentes no disponibles en la biblioteca universal de Seeed Studio se registró el número de parte del distribuidor en el atributo MPN y la dirección de internet del componente en el catálogo del distribuidor en el atributo URL. (Seeed Studio, 2019)

Terminado el diseño de la placa electrónica se generaron todos los archivos necesarios para su fabricación según las instrucciones del fabricante incluyendo la lista de componentes electrónicos o BOM. (Seeed Studio, 2019)

El proyecto fue remitido a fabricación el 26 de mayo de 2019 y debido a que el diseño utiliza componentes no disponibles en la biblioteca de partes universales el tiempo de producción se estimó en 20 días hábiles. Se pueden consultar los detalles de la orden de fabricación en el Apéndice B.

3.4.2 Software

El proceso elegido para la construcción del software es el modelo de desarrollo en cascada. Domínguez (2017) señala “El modelo en cascada es un proceso de desarrollo secuencial en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra”.

El modelo de desarrollo en cascada se compone de cinco fases:

1. **Requisitos de software.** Especifica los requerimientos funcionales del sistema sin entrar en detalles técnicos.
2. **Diseño.** Describe la estructura interna del software y las relaciones entre las entidades que lo componen.
3. **Implementación.** Programación de los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior.
4. **Verificación.** Confirmar que todos los componentes del sistema funcionan correctamente y cumplen los requisitos.
5. **Instalación y mantenimiento.** Instalación de la aplicación en el sistema y comprobación de que funciona correctamente. Modificación del producto para corregir errores, mejorar rendimiento o añadir funcionalidad.

A continuación se aplica este proceso dentro del modelo para desarrollo de sistemas embebidos tomando como base la especificación del producto definida en la sección 3.1.

Requisitos de software

De acuerdo a los parámetros físicos y de desempeño previamente definidos, el sistema debe ser capaz de:

1. Sintonizar un canal de la Banda del clima para captar alertas de riesgo transmitidas por las autoridades.
2. Procesar los mensajes SAME recibidos y filtrar los dirigidos al usuario o bienes.
3. De acuerdo al tipo de evento recibido y su cobertura geográfica transmitir la alerta por medios visuales, auditivos o eléctricos.
4. Transmitir digitalmente los mensajes recibidos para su análisis.

Por otra parte, de la especificación del protocolo SAME estudiada en la sección 1.4 de este trabajo, las recomendaciones y consideraciones que aplican al software del receptor son:

- Uso del divisor en el código de área geográfica
- Discriminación por pares de eventos y códigos de área geográfica
- Corrección de errores
- Recepción de múltiples mensajes
- Prueba de receptor

Diseño

Partiendo de la arquitectura de software y elementos propuestos en la sección 3.2.3, los componentes de software son:

- **Command:** Facilita la recepción de datos por medio del puerto USART del sistema embebido. Interpreta y valida la información recibida en comandos y argumentos.
- **Config:** Provee métodos de obtención y establecimiento de parámetros de operación para administrar la configuración del sistema embebido. Sincroniza la configuración del sistema en la memoria volátil y no volátil del microcontrolador.
- **IO:** Configura los puertos de entrada y salida para los indicadores, botón y relevador del sistema. Simplifica la operación de los indicadores, botón y relevadores.
- **Si4707:** Controlador que permite la interacción con el circuito integrado receptor de mensajes SAME.

Gracias al hardware y software libre empleado en la sección 3.3, se puede tomar como base los componentes de código utilizados en el prototipo del sistema embebido.

Para la documentación de los controladores de dispositivo y el diseño de la capa de aplicación, se utilizará el lenguaje unificado de modelado (del inglés *Unified Modeling Language*, UML). Se emplearán los siguientes tipos de diagramas UML por su particular relevancia en el diseño de sistemas embebidos (Apvrille, 2012):

1. Diagrama de clases
2. Diagrama de casos de uso
3. Diagrama de máquina de estados
4. Diagrama de secuencia
5. Diagrama de estructura compuesta

En el Apéndice C se encuentran los diagramas UML correspondientes al diseño del software.

Implementación

El código de programa, resultado de esta fase del proceso de desarrollo de software, se puede consultar en la carpeta *software/ReceptorSame* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

En esta etapa se definió e implementó la funcionalidad de los LEDs indicadores de estado. En la Tabla 12 se muestran las posibles condiciones según el modo de operación de cada LED.

Tabla 12. Condición del receptor según los LEDs indicadores de estado

Led 1	Led 2	Estado
Encendido	Encendido	Iniciando
Parpadeo rápido	Apagado	Error
Parpadeo lento	Apagado	Esperando mensaje
Parpadeo lento	Encendido	Esperando mensaje con prueba periódica vigente
Parpadeo rápido	Parpadeo rápido	Alerta

Durante esta fase también se definió un estándar para transmitir información por medio del puerto USART del sistema embebido. El protocolo de comunicación se encuentra documentado en el Apéndice C.

Verificación

Para confirmar que todos los componentes del software funcionan correctamente y satisfacen los requerimientos especificados se emplean dos técnicas: pruebas unitarias y pruebas de integración.

Las *pruebas unitarias* son un tipo de pruebas de software en el cual unidades individuales de código son probadas para determinar si se comportan de acuerdo a su diseño. Una unidad de código en programación procedural puede ser un módulo, función o procedimiento y en programación orientada a objetos una clase o método. Escribiendo pruebas unitarias para las unidades más pequeñas de código y acumulandolas se pueden comprobar exhaustivamente aplicaciones complejas. (Hamill, 2004, p. 2)

En la Tabla 13 se describen las pruebas unitarias implementadas de acuerdo al componente individual que verifican, su propósito y sus casos de prueba.

Tabla 13. Pruebas unitarias implementadas.

Pruebas	Propósito	Casos de prueba
TestCommand	Permite verificar manualmente la recepción e interpretación de comandos por la clase Command.	Lee buffer del puerto serial hasta detectar un comando recibido completo.
		Devuelve el tipo de comando recibido y si es válido.
		Determina si el comando recibido incluye un argumento.
		Devuelve el argumento recibido como cadena de texto, entero y byte.
TestConfig	Verifica automáticamente la obtención y establecimiento de la configuración del receptor de alertas por medio de la clase Config.	Informa el tamaño de la configuración en memoria y su versión
		Carga de la memoria flash y vuelca la configuración.
		Altas, bajas, listado y búsqueda de códigos de área.
		Altas, bajas, listado y búsqueda de códigos de eventos.
TestIO	Prueba la operación de los puertos de entrada-salida del prototipo por medio de la clase IO. Con ayuda de un circuito externo permite comprobar el correcto funcionamiento del Hardware (ver Figura 18).	Reporta si el botón externo es presionado
		Enciende el led 1, apaga el led 2 y activa el relevador.
		Apagar el led 1, enciende el led 2 y desactiva el relevador.
		Titila lento el led 1, titila rápido el led 2 y desactiva el relevador.
		Titila rápido el led 1, titila lento el led 2 y desactiva el relevador.
		Activa en los leds el patrón de inicio, error, espera, espera con prueba válida y alerta.
		Activa y desactiva el reproductor de audio externo.

El código de las pruebas unitarias correspondiente a las unidades de código definidas en el diseño del software, se puede consultar en las carpetas *software/TestCommand*, *software/TestConfig* y *software/TestIO* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

Las *pruebas de integración* son una extensión lógica de las pruebas unitarias. En su forma más simple, combinan dos unidades de código previamente verificadas para probar la interfaz entre ellas. (Lin, 2012) Las pruebas de integración permiten comprobar que todos los elementos del software funcionan juntos correctamente y evalúan si un sistema cumple con sus requerimientos funcionales. (ISO/IEC/IEEE, 2017, p. 229)

Para implementar las pruebas de integración se desarrolló un objeto simulado que imita el comportamiento del controlador de software del receptor de mensajes SAME, permitiendo la prueba del software en diferentes escenarios de operación.

En la Tabla 14 se indican los escenarios que verifican las pruebas de integración junto con el mensaje que se recibe con la simulación implementada.

Tabla 14. Escenarios de las pruebas de integración implementadas.

Escenario de prueba	Mensaje simulado
Falla circuito integrado receptor	—
Mensaje de prueba	-CIV-RWT-000000+0300-832300-XDIF/005- -CIV-RMT-000000+0300-832300-XDIF/005-
Mensaje de alerta con voz sin tono ASQ	-WXR-SPS-039173-039051-139069+0030-1591829-KCLE/NWS-
Mensaje de alerta con tono ASQ	-WXR-TOR-039173-039051-139069+0030-1591829-KCLE/NWS-
Mensaje con solo dos encabezados	-CIV-RWT-000000+0300-832300-XDIF/005-
Mensaje de alerta sísmica	-CIV-EQW-000000+0005-832326-XDIF/005-

El código correspondiente al objeto simulado y las pruebas de integración , se puede consultar en los archivos *MockRadio.cpp* y *MockRadio.h* de la carpeta *software/ReceptorSame* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

Instalación y mantenimiento

Mientras el sistema de hardware real está disponible se utilizará el prototipo del sistema embebido construido en la sección 3.3 para comprobar que el software funciona correctamente.

La instalación del software en el prototipo se llevará a cabo a nivel de hardware por el puerto USART del microcontrolador conectado a un convertidor USART-USB y a nivel de software por medio del ambiente Arduino en el sistema anfitrión combinado con su Bootloader en el sistema destino.

Para comprobar que el software funciona correctamente se emplean sentencias de impresión como técnica de depuración durante el desarrollo y para obtener el resultado de las pruebas unitarias y de integración del sistema embebido. En el siguiente capítulo se muestran los resultados de estas pruebas.

3.5 Integración de software y hardware

Recibido el hardware fabricado se da inicio a su integración con el software. En la Figura 15 se muestra la placa electrónica fabricada al momento de recibirla.

Como se menciona en la sección 2.3.5, el proceso de integración es un ejercicio de depuración y descubrimiento, en este caso iniciando por inspeccionar el hardware fabricado para luego combinarlo con el software de aplicación.



Figura 15. Placa electrónica con componentes.

3.5.1 Inspección de hardware

Durante la exploración del hardware se detectaron errores de diseño que se describen a continuación para su posterior corrección.

1. Se interrumpió la conexión entre el pin 3 del puerto de comunicación serie J2 y la señal de 3.3 V y se conectó a la señal de 5 V.
2. Se interrumpió la pista de RESET proveniente del pin 6 del puerto de comunicación serie J2 y se agregó un condensador de 0.1 uF.

La inspección de hardware culmina con la instalación del bootloader en el microcontrolador, proceso que se documenta en el Apéndice D.1.

3.5.2 Combinación de software y hardware

Operando correctamente el hardware fabricado se procede a instalar el software de aplicación y verificar su funcionamiento. Para la combinación de software y hardware se seguirán los procedimientos descritos en la sección 3.4.2 empezando por los de verificación del software y terminando con la instalación del software de aplicación.

La instalación de software en el prototipo se documenta en el Apéndice D.2. La verificación del hardware por medio de las pruebas unitarias y de integración se documentan en los Apéndices D.3 y D.4 respectivamente.

Resultado de las pruebas de verificación y la instalación del software de aplicación se detectaron y corrigieron los siguientes errores.

1. Se cambió el pin RELAY de puerto del microcontrolador de ADC7 a PD3 y se ajustó la configuración del software correspondiente.
2. Se invirtió la posición del diodo D6.

En la Figura 16 y 17 se exhiben las modificaciones a la placa, en alambre amarillo estan los nuevos puentes y se indica en círculos rojos los cortes a pistas de cobre y el diodo invertido.

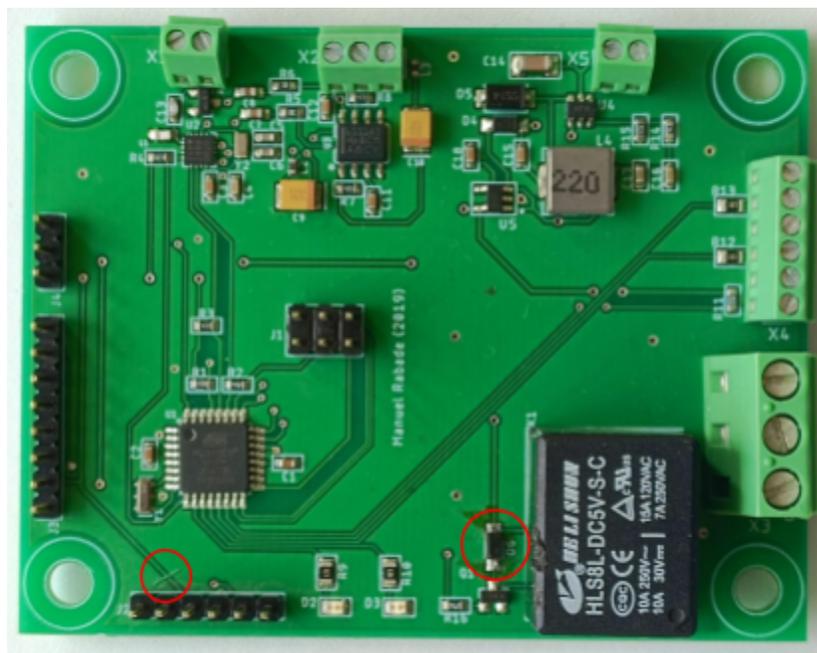


Figura 16. Vista superior de correcciones en la placa electrónica.

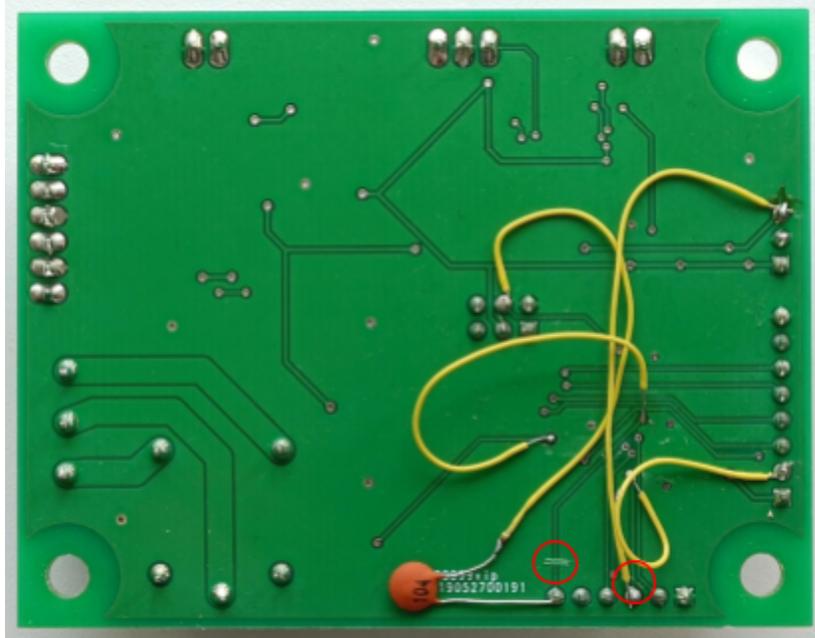


Figura 17. Vista inferior de correcciones en la placa electrónica.

3.6 Pruebas de producto y lanzamiento

A continuación se listan las pruebas aplicadas al receptor de alertas y sus resultados, agrupadas según las técnicas descritas en la sección 2.3.6.

3.6.1 Pruebas estáticas de caja negra

Estas pruebas tienen el objetivo de verificar que el receptor de alertas cumple la especificación de producto descrita en la sección 3.1. A continuación se analiza el porqué del cumplimiento de los requerimientos funcionales de nuestro sistema.

1. La misión principal de *recibir alertas de riesgo emitidas mediante el protocolo SAME y difundirlas por medios visuales, auditivos o electrónicos* se cumple exitosamente como se demostró durante la integración de software y hardware más sin embargo más adelante se comprueba durante el proceso de pruebas dinámicas.
2. La misión secundaria de *analizar el protocolo SAME y los sistemas de alertamiento que lo usan* se lleva a cabo en el capítulo 1 de este trabajo.
3. La misión secundaria de *documentar las bases técnicas para el desarrollo de productos o servicios que incrementen el aprovechamiento y la cobertura de los sistemas de alertamiento de riesgos* se realiza a lo largo de este capítulo.

Cumpliendo la misión primaria y secundarias, podemos concluir que tanto los parámetros físicos y de desempeño como los requerimientos de uso del sistema embebido se satisfacen debidamente.

Para verificar si el receptor de alertas cumple los factores de efectividad y ciclo de vida operacional, a continuación se analiza y evalúa cada parámetro de acuerdo a la especificación de producto.

Costo

De acuerdo con el Apéndice B.4, el costo de fabricación de 5 unidades fue de \$224 USD más \$1,102 MXN por impuestos de importación siendo el costo unitario de cada receptor de \$1,116 MXN.

Por otra parte, de acuerdo a los montos indicados en la Tabla 15, el costo promedio de receptores comerciales con prestaciones equivalentes es de \$2,413 MXN.

Tabla 15. Receptores de alertas con prestaciones equivalentes

Modelo	Costo MXN
Midland WR120	\$1,250
Midland WR300	\$2,390
Receptor Sarmex	\$3,599

A partir de los costos unitarios calculados podemos concluir que el receptor cumple la condición de que su precio debe ser igual o menor al de alternativas comerciales con prestaciones similares.

Cabe mencionar que el receptor fabricado carece de ciertas características presentes en las alternativas comerciales que no son partes de sus requerimientos operaciones como un gabinete plástico, batería de respaldo integrada, altavoz, pantalla, botones de configuración, etc.

Es importante aclarar que los costos del receptor de alertas incluyen un importante descuento del fabricante. Eliminando el descuento de fabricación y proyectando los costos de importación el costo unitario de cada receptor fabricado sería de \$3,374 MX.

Tiempo entre mantenimiento programado

Según los requerimientos operacionales, el ambiente donde operará el receptor de alertas será un entorno protegido de los elementos naturales. Por otro lado, los componentes que integran al receptor de alertas son electrónicos o electromecánicos. A partir de estas dos premisas podemos concluir que el mantenimiento preventivo del receptor de alertas debe ser mínimo y alcanzar períodos de al menos un año.

En la sección 3.7 se especifican los procedimientos de mantenimiento preventivo que se deben realizar para asegurar el correcto funcionamiento del receptor de alertas. Realizando el mantenimiento preventivo al cabo de un año de operación, de acuerdo a la Tabla 16, el tiempo total suma 10 minutos, duración inferior a la meta establecida de una hora al año.

Tabla 16. Tiempos de mantenimiento preventivo realizados al receptor de alertas

Tarea	Tiempo	Notas
Inspección visual	3 minutos	Se detectó polvo acumulado en la placa electrónica
Prueba de recepción	1 minuto	Led indicador de recepción de mensaje de prueba encendido
Prueba de alerta	1 minuto	Al presionar el botón de prueba se activaron los leds en modo alerta, se encendió el audio y se activó el relevador correctamente
Prueba de batería	—	El banco de pruebas no cuenta con batería de respaldo
Ajuste de instalación	5 minutos	Limpieza de polvo

Tasa de mensajes SAME captados exitosamente

Para comprobar que el receptor de alertas cumple el requerimiento de que la tasa de mensajes SAME captados exitosamente durante el periodo de prueba debe ser de un 99% utilizaremos los datos recabados durante las pruebas dinámicas de caja blanca.

En el periodo del 1 de abril de 2021 al 31 de marzo de 2022 el receptor captó 2,755 mensajes SAME con los siguientes eventos:

- 2,741 mensajes RWT
- 14 mensajes EQW

Conociendo de antemano que los eventos RWT se transmiten 8 veces al día en la Ciudad de México, las tasa de mensajes de prueba recibidos es:

$$(2741 / (8 * 365)) * 100 = 93.9\%$$

Los 14 mensajes EQW captados que suponen alertas públicas de sismo para la Ciudad de México corresponden a 5 eventos:

1. 21 de junio de 2021 a las 11:30
2. 7 de septiembre de 2021 a las 20:48
3. 19 de septiembre de 2021 a las 11:30
4. 3 de marzo de 2022 a las 8:41
5. 18 de marzo de 2022 a las 13:54

Consultando el Histórico del Sistema de Alerta Sísmica Mexicano (Centro de Instrumentación y Registro Sísmico, s.f) en el periodo del 1 de abril de 2021 al 31 de abril de 2022 se reportan 3 alertas públicas de sismo para la Ciudad de México:

1. 7 de septiembre de 2021 a las 20:47
2. 3 de marzo de 2022 a las 08:41
3. 18 de marzo de 2022 a las 13:54

Sabiendo de antemano que el 21 de junio de 2021 y 19 de septiembre de 2021 a las 11:30 se llevaron a cabo simulacros que incluyeron la transmisión de la alerta pública de sismo en la Ciudad de México, la tasa de mensajes de alerta recibidos fue del 100% en el periodo de prueba.

Así podemos concluir que se supera la meta de una tasa de 99% de mensajes de alerta captados exitosamente y aunque la tasa de mensajes de prueba recibidos no alcanzó el objetivo, se da por concluido con éxito este grupo de pruebas.

En el capítulo 4, se analizarán los resultados de recepción de mensajes de prueba para intentar esclarecer por qué no se alcanzó la meta propuesta en este tipo de mensajes.

3.6.2 Pruebas estáticas de caja blanca

En esta etapa se revisó cuidadosamente el diseño de hardware y software documentado en la sección 3.4 para encontrar errores. Durante la búsqueda se descubrieron y corrigieron los siguientes errores:

1. Tres comparaciones erróneas de la bandera de recepción de mensajes de prueba mensual.
2. Valores invertidos en los argumento de las funciones para apagar y encender el audio del receptor.

3.6.3 Pruebas dinámicas de caja negra

Durante la etapa de verificación del software en la sección 3.4.2 se emplearon las técnicas de pruebas unitarias y pruebas de integración para confirmar que todos los componentes del software funcionan correctamente y satisfacen los requisitos del software además del funcionamiento del prototipo de hardware. Dentro de las pruebas unitarias y de integración se contemplaron las pruebas propuestas por el modelo de pruebas dinámicas de caja negra.

Las pruebas unitarias incluyen las pruebas de datos (verificar la información en las entradas y salidas) y pruebas de entrada (introducir datos inválidos o vacíos al sistema) en las siguientes interfaces de entrada-salida:

- Salida de audio.
- Puerto de comunicación serie.
- Salidas digitales.
- Relevador AC/DC.
- Salidas para leds externos.
- Entrada para botón de prueba.

Las pruebas de integración contemplan las pruebas de estado (transiciones entre estados del software) y pruebas de condiciones límite (ejecución del sistema al borde de sus límites operacionales) de acuerdo al diseño del software documentado en la sección 3.4.2 de este trabajo.

Los diagramas UML derivados del diseño de software se encuentran en el Apéndice C.1. El código de las pruebas unitarias y de integración se puede consultar en la carpeta *software* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

3.6.4 Pruebas dinámicas de caja blanca

Para verificar que el sistema opera correctamente basándose en el conocimiento a detalle de su diseño, se integró el receptor de alertas con una computadora para recolectar los registros de su operación y buscar errores en datos o cálculos, comparaciones erróneas, errores de control de flujo, parámetros, errores de entrada/salida, etc.

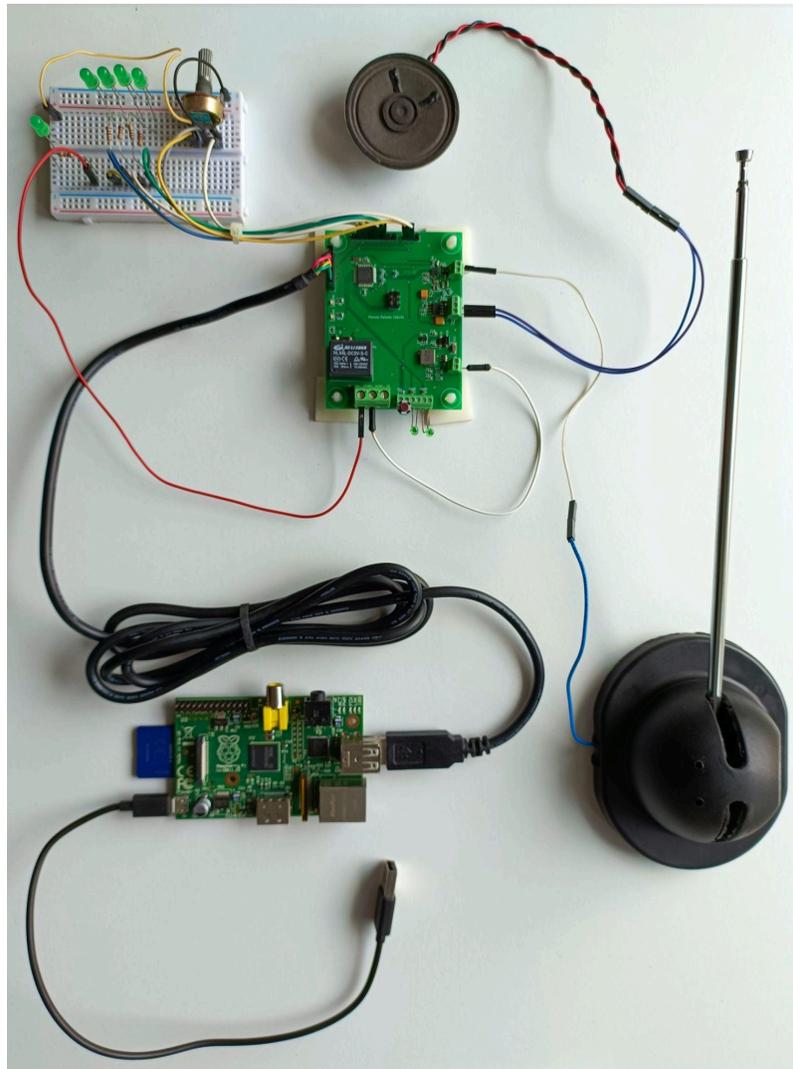


Figura 18. Sistema para pruebas dinámicas de caja blanca.

En la Figura 18 se muestra el sistema para pruebas dinámicas de caja blanca. Al centro se encuentra el receptor de alertas, en la parte superior izquierda está un tablero de circuitos construido para probar las entradas-salidas del receptor de alertas y en la parte inferior izquierda se ubica la mini computadora para recolectar los registros de operación durante las pruebas dinámicas.

La computadora elegida para recolectar los registros es una *Raspberry Pi* con un convertidor USART-USB que le permite recibir la información generada por el receptor de alertas y almacenarla en su memoria SD para su análisis. En el Apéndice E se describe la configuración del sistema operativo de la computadora para recolectar los registros de operación.

Analizando los registros de recepción de un mensaje de prueba, podemos comprobar que el receptor capta e interpreta el mensaje correctamente. A continuación se muestran dichos registros.

```
2022-06-26,11:45:01,1656261901.5076096,SAME_PRE_DET
2022-06-26,11:45:01,1656261901.6218686,SAME_HDR_DET
2022-06-26,11:45:02,1656261902.3224835,"SAME_HDR_RDY,1"
2022-06-26,11:45:04,1656261904.452356,SAME_PRE_DET
2022-06-26,11:45:04,1656261904.6277263,SAME_HDR_DET
2022-06-26,11:45:05,1656261905.266389,"SAME_HDR_RDY,2"
2022-06-26,11:45:07,1656261907.4674582,SAME_PRE_DET
2022-06-26,11:45:07,1656261907.5810804,SAME_HDR_DET
2022-06-26,11:45:08,1656261908.2707438,"SAME_HDR_RDY,3"
2022-06-26,11:45:08,1656261908.3634589,"SAME_MESSAGE,
-CIV-RWT-000000+0300-832300-XDIF/005- . . . . Q@"
2022-06-26,11:45:08,1656261908.3708994,SAME_RWT
2022-06-26,11:45:08,1656261908.3832698,SAME_EOM_DET
2022-06-26,11:45:10,1656261910.1219087,SAME_PRE_DET
2022-06-26,11:45:10,1656261910.3476996,SAME_EOM_DET
2022-06-26,11:45:12,1656261912.0384247,SAME_PRE_DET
2022-06-26,11:45:12,1656261912.2031474,SAME_EOM_DET
2022-06-26,11:45:13,1656261913.8936677,SAME_PRE_DET
2022-06-26,11:45:14,1656261914.1204956,SAME_EOM_DET
```

Por otro lado, analizando los registros de recepción de un mensaje de alerta podemos comprobar que el receptor capta e interpreta el mensaje correctamente pero encontramos una variación importante con la especificación descrita en la sección 1.3.

Según los registros recabados, se recibe el mensaje de alerta 3 veces y ninguno contiene mensaje de voz ya que inmediatamente después de cada mensaje se recibe el fin del mensaje.

```
2022-03-03,08:41:19,1646318479.7413568,SAME_PRE_DET
2022-03-03,08:41:19,1646318479.9059904,SAME_HDR_DET
2022-03-03,08:41:20,1646318480.5450633,"SAME_HDR_RDY,1"
2022-03-03,08:41:20,1646318480.7181363,SAME_PRE_DET
2022-03-03,08:41:20,1646318480.8320622,SAME_HDR_DET
2022-03-03,08:41:21,1646318481.5320325,"SAME_HDR_RDY,2"
2022-03-03,08:41:21,1646318481.7075624,SAME_PRE_DET
2022-03-03,08:41:21,1646318481.8211234,SAME_HDR_DET
2022-03-03,08:41:22,1646318482.5113442,"SAME_HDR_RDY,3"
2022-03-03,08:41:22,1646318482.6025028,
    "SAME_MESSAGE,-CIV-EQW-000000+0005-832326-XDIF/005-...H."
2022-03-03,08:41:22,1646318482.6189742,SAME_ALERT_ON
2022-03-03,08:41:22,1646318482.6273596,AUDIO_ON
2022-03-03,08:41:22,1646318482.6354125,RELAY_ON
2022-03-03,08:41:22,1646318482.6517386,SAME_EOM_DET
2022-03-03,08:41:22,1646318482.666511,AUDIO_OFF
2022-03-03,08:41:22,1646318482.6740036,RELAY_OFF
2022-03-03,08:41:22,1646318482.6856492,SAME_ALERT_OFF
2022-03-03,08:41:23,1646318483.1151667,SAME_PRE_DET
2022-03-03,08:41:23,1646318483.351819,SAME_EOM_DET
2022-03-03,08:41:25,1646318485.3818288,SAME_PRE_DET
2022-03-03,08:41:25,1646318485.5564318,SAME_HDR_DET
2022-03-03,08:41:26,1646318486.1853862,"SAME_HDR_RDY,1"
2022-03-03,08:41:26,1646318486.3589892,SAME_PRE_DET
2022-03-03,08:41:26,1646318486.5347984,SAME_HDR_DET
2022-03-03,08:41:27,1646318487.1733625,"SAME_HDR_RDY,2"
2022-03-03,08:41:27,1646318487.3467965,SAME_PRE_DET
2022-03-03,08:41:27,1646318487.4607763,SAME_HDR_DET
```

Mensaje de alerta 1

Fin de mensaje

2022-03-03,08:41:28,1646318488.1609406,"SAME_HDR_RDY,3"
2022-03-03,08:41:28,1646318488.2435648,
"SAME_MESSAGE,-CIV-EQW-000000+0005-832326-XDIF/005- .@. . ." **Mensaje de alerta 2**
2022-03-03,08:41:28,1646318488.2585003,SAME_ALERT_ON
2022-03-03,08:41:28,1646318488.2730596,AUDIO_ON
2022-03-03,08:41:28,1646318488.2803884,RELAY_ON
2022-03-03,08:41:28,1646318488.2927287,SAME_EOM_DET **Fin de mensaje**
2022-03-03,08:41:28,1646318488.3059452,AUDIO_OFF
2022-03-03,08:41:28,1646318488.3209665,RELAY_OFF
2022-03-03,08:41:28,1646318488.3376045,SAME_ALERT_OFF
2022-03-03,08:41:28,1646318488.7561357,SAME_PRE_DET
2022-03-03,08:41:28,1646318488.9924078,SAME_EOM_DET
2022-03-03,08:41:31,1646318491.021864,SAME_PRE_DET
2022-03-03,08:41:31,1646318491.19731,SAME_HDR_DET
2022-03-03,08:41:31,1646318491.835689,"SAME_HDR_RDY,1"
2022-03-03,08:41:32,1646318492.0002162,SAME_PRE_DET
2022-03-03,08:41:32,1646318492.1746354,SAME_HDR_DET
2022-03-03,08:41:32,1646318492.8144035,"SAME_HDR_RDY,2"
2022-03-03,08:41:32,1646318492.9879205,SAME_PRE_DET
2022-03-03,08:41:33,1646318493.1637816,SAME_HDR_DET
2022-03-03,08:41:33,1646318493.8024356,"SAME_HDR_RDY,3"
2022-03-03,08:41:33,1646318493.883001,
"SAME_MESSAGE,-CIV-EQW-000000+0005-832326-XDIF/005-" **Mensaje de alerta 3**
2022-03-03,08:41:33,1646318493.8996122,SAME_ALERT_ON
2022-03-03,08:41:33,1646318493.9138627,AUDIO_ON
2022-03-03,08:41:33,1646318493.9209754,RELAY_ON
2022-03-03,08:41:33,1646318493.9333868,SAME_EOM_DET **Fin de mensaje**
2022-03-03,08:41:33,1646318493.9467692,AUDIO_OFF
2022-03-03,08:41:33,1646318493.9617512,RELAY_OFF
2022-03-03,08:41:33,1646318493.9783897,SAME_ALERT_OFF
2022-03-03,08:41:34,1646318494.458556,SAME_PRE_DET
2022-03-03,08:41:34,1646318494.6335387,SAME_EOM_DET
2022-03-03,08:41:36,1646318496.261302,SAME_PRE_DET

Más adelante, en el capítulo 4, se analizará el comportamiento del receptor ante los mensajes de alerta para confirmar su correcta operación.

3.7 Mantenimiento y actualizaciones

Una vez puesto en operación el receptor de alertas se considera que una vez al año se realicen las siguientes pruebas y procedimientos de mantenimiento preventivo.

1. **Inspección visual.** Verificar visualmente el buen estado de la instalación del sistema embebido y corregir cualquier defecto.
2. **Prueba de recepción.** Verificar que los mensajes de prueba periódicos han sido recibidos exitosamente por el sistema embebido.
3. **Prueba de alerta.** Verificar el correcto funcionamiento de los mecanismos para la alerta, ya sean audiovisuales o electrónicos.
4. **Prueba de batería.** Si la fuente de alimentación del sistema embebido cuenta con batería de respaldo de energía:
 - a. Verificar visualmente que la batería no sufre deformaciones o fugas.
 - b. Verificar que la fuente de alimentación conmuta entre el suministro de energía principal y la batería de respaldo.
 - c. Verificar que la fuente de alimentación recarga la batería.
5. **Ajuste de instalación.** Corregir cualquier defecto, daño o desgaste de cualquier elemento mecánico o eléctrico del sistema como su cerradura, gabinete, soportes o cableado.

Por otro lado, derivado del análisis de resultados llevado a cabo en el capítulo 4, se llevó a cabo una actualización del receptor de alertas para:

1. Corregir el error que descarta alertas cuyo encabezado se reciba solo dos veces en lugar de tres.
2. Al recibir un código de alerta EQW, reproducir el audio característico de alerta sísmica durante 60 segundos.

A continuación se describen las mejoras llevadas a cabo, tanto en el hardware como en el software del prototipo realizado.

3.7.1 Actualización de Hardware

Para añadir al prototipo la capacidad de reproducir el audio de alerta sísmica al recibir un código EQW, se integró el módulo ISD1820 que se muestra en la Figura 19.



Figura 19. Módulo ISD1820 para reproducción de audio.

Experimentando con el funcionamiento del módulo ISD1820, se determinó grabar un ciclo del audio de la alerta sísmica para luego interconectarse con el receptor de alertas como se muestra en la Figura 20.

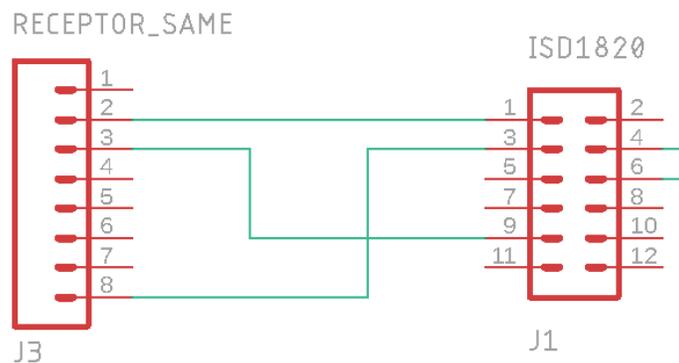


Figura 20. Interconexión del módulo ISD1820 con el receptor de alertas.

En la siguiente sección, se programara la integración entre el módulo ISD1820 y el receptor de alertas para reproducir el audio según sea necesario.

3.7.2 Actualización de Software

Para abordar las mejoras necesarias en el código de software se utilizará la práctica de desarrollo guiado por pruebas que consiste de los siguientes pasos.

1. Escribir las pruebas de software necesarias y verificar que las nuevas pruebas fallan.
2. Implementar el software que hace que la prueba se cumpla satisfactoriamente.
3. Ajustar el código según sea necesario comprobando que las pruebas siguen cumpliéndose correctamente.

El propósito del desarrollo de software guiado por pruebas es garantizar que el producto desarrollado cumple con los requisitos establecidos. (Hamilton, 2020)

Para empezar con la actualización de software, se desarrollaron las siguientes pruebas de integración que nos permitirán replicar las situaciones en las que el receptor requiere ajustes.

1. La recepción de un mensaje de prueba de solo dos encabezados simulando los registros de operación mostrados en el apartado 4.3.2.
2. La recepción de un mensaje de alerta con código EQW simulando los registros de operación ejemplificados en la sección 4.4.

Posterior a la ejecución de estas pruebas, se confirmó que para estos dos casos se comportan igual a los registros de operación estudiados en la sección 4. En los casos en donde se dió un comportamiento diferente, se procedió a corregir el error que ignoraba los mensajes de alerta con solo dos encabezados recibidos de manera exitosa. Los casos en los que efectivamente el receptor requirió ajustes se presentan y analizan en el Capítulo 4.

Además, tomando en cuenta las consideraciones respecto al audio de alerta sísmica que se describen en la sección 4.5.3, para la integración del módulo ISD1820 se implementó:

- Un nuevo parámetro de configuración para indicar si el receptor debe seguir lo indicado en la Norma Técnica NT-SGIRPC-SDSAS-001-2021 con sus pruebas unitarias correspondientes.
- El sistema de control para activar la reproducción de audio en el módulo ISD1820 con sus respectivas pruebas unitarias.

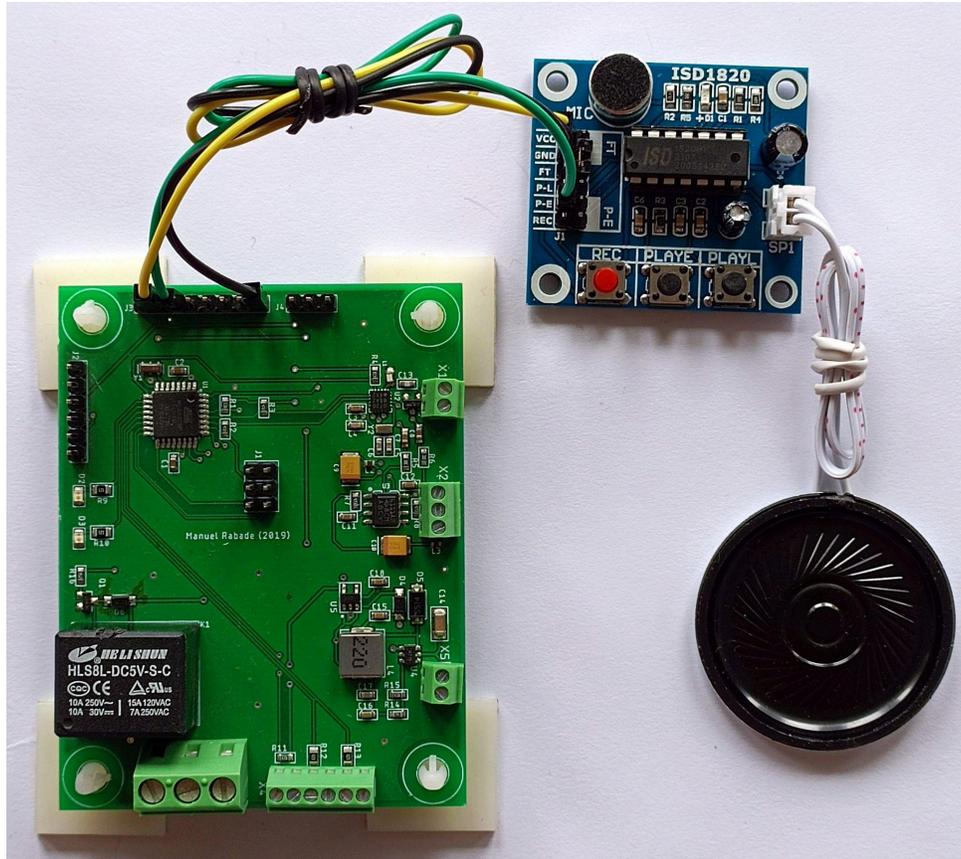


Figura 21. Receptor de alertas y módulo ISD1820.

Una vez programadas estas pruebas unitarias y la funcionalidad necesaria para cumplirlas, se implementó la funcionalidad en el programa principal para satisfacer las pruebas de integración desarrolladas al comienzo del proceso de actualización de software. En la Figura 21 se muestra a la izquierda el receptor de alertas y a la derecha el módulo ISD1820 interconectados en su configuración final.

Por último y para concluir con la actualización del software del receptor de alertas se incorporaron estos cambios a la documentación y código de los Apéndices D y F de este trabajo.

Capítulo 4. Resultados y mejoras

El objetivo de este capítulo es mostrar los resultados de este trabajo mientras se evalúa el diseño del receptor de alertas por medio de un análisis de los resultados obtenidos durante las pruebas de producto.

Como parte de la evaluación del receptor de alertas, también se verificó su funcionamiento durante un simulacro de alerta sísmica. Los detalles de esta prueba se encuentran relatados en este mismo capítulo.

Por último, a modo de cierre global de este trabajo, se exponen posibles configuraciones del receptor de alertas para su empleo en diferentes escenarios.

En este capítulo se presenta:

- El resumen de las pruebas llevadas a cabo durante el diseño del receptor con su resultado.
- A partir de los datos recabados durante el periodo de evaluación, el estudio de los resultados no satisfactorios para descubrir su causa origen.
- Las mejoras necesarias para que el receptor de alertas alcance las metas propuestas.
- La prueba del receptor durante el simulacro de alerta sísmica del 19 de abril de 2023.
- Tres ejemplos de configuración del receptor y el ensamble de uno de ellos.

4.1 Resultados de las pruebas

De acuerdo a la sección 3.6, las pruebas llevadas a cabo con resultado positivo fueron:

- Comprobamos que el receptor de alertas cumple con su misión principal de recibir y difundir alertas de riesgo transmitidas usando el protocolo SAME.
- En los capítulos 1 y 3 de este trabajo se cumplieron las misiones secundarias de analizar el protocolo SAME y documentar las bases técnicas para el desarrollo de tecnología que incrementen su aprovechamiento
- Se cumplen tanto los parámetros físicos y de desempeño como los requerimientos de uso planteados en la especificación de producto documentada en la sección 3.1.
- Los costos del prototipo fabricado son iguales o menores al de alternativas comerciales con prestaciones similares.

- El tiempo de mantenimiento programado anual ensayado es de 10 minutos, menor al límite de una hora al año.
- El prototipo captó todos los mensajes SAME de alerta transmitidos superando el objetivo de una tasa de recepción del 99% de mensajes.
- Se detectaron y corrigieron dos errores durante la revisión del diseño de hardware y software.
- En la sección 3.4.2 se construyeron un conjunto de pruebas unitarias y de integración para confirmar que tanto los componentes, el software en conjunto y el prototipo de hardware funcionan correctamente y satisfacen los requisitos del software.
- El prototipo capta e interpreta correctamente los mensajes SAME de prueba recibidos.

Por otro lado, las pruebas en las que se obtuvo un resultado diferente al esperado fueron:

- La tasa de mensajes SAME de prueba captados fue del 94% y no alcanzó el objetivo de un 99% de mensajes recibidos correctamente.
- El prototipo capta e interpreta correctamente los mensajes SAME de alerta pero según los registros de operación se repiten 3 veces y sin espacio para audio como se esperaba por ser alertas sísmicas.

Para descubrir la causa origen de los resultados no satisfactorios y retroalimentar el proceso de diseño llevado a cabo el capítulo 3 de este trabajo, en las siguientes secciones:

1. Estudiaremos a fondo los registros de operación para descubrir por qué no se recibieron los mensajes de prueba faltantes.
2. Investigaremos de nueva cuenta los estándares y regulaciones para determinar el funcionamiento esperado del receptor ante los mensajes de alerta.

4.2 Datos recabados

En el diseño de software y hardware se contempló aprovechar el puerto serie del receptor de alertas para enviar información de su funcionamiento además de la configuración del sistema embebido y retransmisión de mensajes recibidos.

Durante las pruebas dinámicas de caja blanca del 1 de abril de 2021 al 31 de marzo de 2022, se recabaron registros de operación del prototipo con ayuda de una computadora externa. Cada mensaje está acompañado de la fecha y hora en que se registró y se encuentran en archivos de texto nombrados por el día en que se recolectaron.

El significado de los mensajes de operación del receptor se encuentra documentado a detalle en el Apéndice C.2 y podemos catalogarlos en:

1. **Estatus del receptor:** encendido, configuración del receptor, comunicación con el circuito integrado para sintonizar la radio del clima, prueba de alarma y validez de mensajes de prueba SAME.
2. **Recepción de mensajes SAME:** detección de tono ASQ, preámbulo, encabezado o fin de mensaje SAME; recepción de mensajes de alerta o prueba y discriminación de mensajes por código de area o evento.
3. **Actividad de periféricos de salida:** encendido o apagado de audio, activación del relevador o del botón de usuario.

Con las marcas de tiempo y los mensajes recabados, podemos estudiar el comportamiento del receptor de alertas respecto a su estatus, recepción de mensajes y acción de periféricos durante la etapa de evaluación.

Por la extensión de los registros de operación durante el periodo de prueba (81,209 líneas en total), en el Apéndice E se reproduce un día representativo del periodo de prueba.

4.3 Mensajes faltantes

A partir del estudio de los registros de operación durante las pruebas dinámicas de caja blanca descritas en la sección 3.6.4, la tasa de mensajes SAME recibidos fue:

- 100% de mensajes de alerta recibidos
- 93.9% de mensajes de prueba recibidos

Cumplida la meta de mensajes de alerta recibidos, a continuación examinaremos los registros de operación respecto a los mensajes de prueba sabiendo de antemano que se deben recibir 8 mensajes por día a las 2:45, 5:45, 8:45, 11:45, 14:45, 17:45, 20:45 y 23:45 horas.

Primeramente, en la Tabla 17 exponemos el número de mensajes de prueba recibidos por día durante el periodo de verificación del 1 de abril de 2021 al 31 de marzo de 2022.

Tabla 17. Mensajes de prueba recibidos por día durante el periodo de verificación.

Mensajes recibidos	Días	Porcentaje
9	2	0.5%
8	295	80.8%
7	41	11.2%
6	8	2.2%
5	2	0.5%
4	4	1.1%
3	0	0%
2	1	0.3%
1	0	0%
0	12	3.3%

Mientras que el 80.8% de días del periodo de verificación se recibieron correctamente todos los mensajes de prueba, 15.3% de días se recibieron de 2 a 7 mensajes y 3.3% de días no se recibieron mensajes.

Sobresale que dos días se recibieron más mensajes de los esperado: el 21 de noviembre de 2021 se captó un mensaje extra a las 2:44 y el 19 de enero de 2022 se captaron dos mensajes a las 11:45.

Por otro lado, en la Tabla 18 indicamos el número de mensajes de prueba captados por la hora de recepción.

Tabla 18. Mensajes de prueba recibidos por hora durante el periodo de verificación.

Mensajes recibidos	Hora	Porcentaje
1	2:44	0.3%
340	2:45	93.2%
1	3:00	0.3%
337	5:45	92.3%
344	8:45	94.2%
346	11:45	94.8%
1	14:44	0.3%
344	14:45	94.2%
342	17:45	93.7%
342	20:45	93.7%
1	23:44	0.3%
342	23:45	93.7%

Observando en conjunto el número de mensajes de prueba recibidos por hora y por día podemos comprobar que los registros recabados durante el periodo de verificación tienen sentido con las transmisiones de prueba efectuadas en la Ciudad de México.

Partiendo de los análisis anteriores, en la Tabla 19 se resume la cantidad y tasa de mensajes recibidos y faltantes tomando en cuenta los mensajes sobrantes.

Tabla 19. Mensajes de prueba esperados, recibidos, sobrantes y faltantes.

Mensajes esperados	2,920	100%
Mensajes recibidos	2,739	93.8%
Mensajes sobrantes	2	0.07%
Mensajes faltantes	181	6.2%

Respecto a los 181 mensajes de prueba no captados, analizando a detalle los registros de operación del prototipo podemos deducir tres escenarios que dan cuenta por qué no se recibieron.

4.3.1 Falta de energía

Cuando se interrumpen abruptamente los registros de operación por un periodo de tiempo substancial y a continuación se captura el indicador de encendido, podemos deducir que el receptor de alertas experimentó una falla en el suministro de energía eléctrica.

Por ejemplo, el 31 de julio de 2021 el último mensaje captado y registrado fue a las 14:45:

```
2021-07-31,14:45:01,1627760701.3999512,SAME_PRE_DET
2021-07-31,14:45:01,1627760701.5644817,SAME_HDR_DET
2021-07-31,14:45:02,1627760702.265705,"SAME_HDR_RDY,1"
2021-07-31,14:45:04,1627760704.4053617,SAME_PRE_DET
2021-07-31,14:45:04,1627760704.5186076,SAME_HDR_DET
2021-07-31,14:45:05,1627760705.219313,"SAME_HDR_RDY,2"
2021-07-31,14:45:07,1627760707.348583,SAME_PRE_DET
2021-07-31,14:45:07,1627760707.523335,SAME_HDR_DET
2021-07-31,14:45:08,1627760708.163171,"SAME_HDR_RDY,3"
2021-07-31,14:45:08,1627760708.254538,
    "SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-....."
2021-07-31,14:45:08,1627760708.2620356,SAME_RWT
2021-07-31,14:45:08,1627760708.2730422,SAME_EOM_DET
2021-07-31,14:45:10,1627760710.0635216,SAME_PRE_DET
2021-07-31,14:45:10,1627760710.3000858,SAME_EOM_DET
2021-07-31,14:45:11,1627760711.917875,SAME_PRE_DET
2021-07-31,14:45:12,1627760712.1548173,SAME_EOM_DET
2021-07-31,14:45:13,1627760713.7839618,SAME_PRE_DET
2021-07-31,14:45:14,1627760714.0096927,SAME_EOM_DET
```

Los registros de operación continúan de nuevo el 2 de agosto a las 8:53 con un encendido seguido de la recepción de un mensaje de prueba a las 11:45:

2021-08-02,08:53:10,1627912390.350261,STARTUP
2021-08-02,08:53:10,1627912390.3616052,"CONFIG_VERSION,16"
2021-08-02,08:53:10,1627912390.624779,SI4707_OK
2021-08-02,08:53:10,1627912390.637701,"MUTE,1"
2021-08-02,08:53:10,1627912390.652746,"VOLUME,10"
2021-08-02,08:53:10,1627912390.8606012,"TUNE_OK,7"
2021-08-02,08:53:10,1627912390.865925,RUN
2021-08-02,10:32:50,1627918370.957889,SAME_PRE_DET
2021-08-02,10:32:51,1627918371.19443,SAME_EOM_DET
2021-08-02,10:32:52,1627918372.8742192,SAME_PRE_DET
2021-08-02,10:32:53,1627918373.1108203,SAME_EOM_DET
2021-08-02,10:32:54,1627918374.738338,SAME_PRE_DET
2021-08-02,10:32:54,1627918374.9657767,SAME_EOM_DET
2021-08-02,11:45:01,1627922701.504798,SAME_PRE_DET
2021-08-02,11:45:01,1627922701.617013,SAME_HDR_DET
2021-08-02,11:45:02,1627922702.3176384,"SAME_HDR_RDY,1"
2021-08-02,11:45:04,1627922704.4585829,SAME_PRE_DET
2021-08-02,11:45:04,1627922704.6221495,SAME_HDR_DET
2021-08-02,11:45:05,1627922705.2621295,"SAME_HDR_RDY,2"
2021-08-02,11:45:07,1627922707.4635532,SAME_PRE_DET
2021-08-02,11:45:07,1627922707.5757432,SAME_HDR_DET
2021-08-02,11:45:08,1627922708.2760675,"SAME_HDR_RDY,3"
2021-08-02,11:45:08,1627922708.358588,
 "SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-....."
2021-08-02,11:45:08,1627922708.3734972,SAME_RWT
2021-08-02,11:45:08,1627922708.3822799,SAME_EOM_DET
2021-08-02,11:45:10,1627922710.1164374,SAME_PRE_DET
2021-08-02,11:45:10,1627922710.3417885,SAME_EOM_DET
2021-08-02,11:45:12,1627922712.0327682,SAME_PRE_DET
2021-08-02,11:45:12,1627922712.259561,SAME_EOM_DET
2021-08-02,11:45:13,1627922713.8883746,SAME_PRE_DET
2021-08-02,11:45:14,1627922714.125125,SAME_EOM_DET

Durante el periodo de prueba hay cinco períodos sin mensajes recibidos que siguen ese patrón y suman 120 mensajes faltantes:

1. Del 31 de julio de 2021 después de las 14:45 y hasta el 2 de agosto de 2021 a las 8:53.
2. Del 11 de agosto de 2021 después de las 17:45 y hasta el 23 de agosto de 2021 a las 17:17.
3. Del 7 de noviembre de 2021 después de las 20:45 y hasta el 8 de noviembre a las 12:01.
4. El 4 de diciembre de 2021 después de las 2:45 y hasta las 11:25.
5. Del 28 de diciembre de 2021 después de las 17:45 y hasta el 29 de diciembre de 2021 a las 6:17.

Las causas más probables de la falta de energía son fallas generales en el suministro de electricidad o desconexiones del cable de alimentación de energía del receptor por accidente o por aseo de la habitación donde se ubicaba el banco de pruebas.

4.3.2 Mensajes ignorados

Cuando en lugar de recibir tres encabezados sólo se recibieron dos seguidos del fin de mensaje el receptor ignoró el mensaje capturado. Por ejemplo el 13 de abril del 2021 a las 17:45 no se procesó el mensaje de prueba por que se recibieron solo dos encabezados:

```
2021-04-13,17:45:01,1618353901.5931165,SAME_PRE_DET
2021-04-13,17:45:01,1618353901.707077,SAME_EOM_DET
2021-04-13,17:45:04,1618353904.4258883,SAME_PRE_DET
2021-04-13,17:45:04,1618353904.6018205,SAME_HDR_DET
2021-04-13,17:45:05,1618353905.3028526,"SAME_HDR_RDY,1"
2021-04-13,17:45:07,1618353907.442126,SAME_PRE_DET
2021-04-13,17:45:07,1618353907.6160212,SAME_HDR_DET
2021-04-13,17:45:08,1618353908.256189,"SAME_HDR_RDY,2"
2021-04-13,17:45:10,1618353910.0970743,SAME_PRE_DET
2021-04-13,17:45:10,1618353910.38495,SAME_EOM_DET
2021-04-13,17:45:12,1618353912.0141604,SAME_PRE_DET
2021-04-13,17:45:12,1618353912.2504969,SAME_EOM_DET
2021-04-13,17:45:13,1618353913.8683727,SAME_PRE_DET
2021-04-13,17:45:14,1618353914.1046448,SAME_EOM_DET
```

Siguiendo este patrón 49 mensajes de prueba no se interpretaron. Con casi total seguridad la causa de esta omisión es un error en el algoritmo de software que interpreta los mensajes recibidos.

4.3.3 Sin razón aparente

Los restantes 12 mensajes de prueba faltantes no encajan con alguno de los patrones anteriores. Posibles causas de la falta de estos mensajes son que no se emitieron o la antena del receptor se desconectó probablemente por acomodo y limpieza del escritorio donde se encontraba operando el banco de pruebas.

Para concluir el análisis de mensajes de prueba faltantes, en la Tabla 20 se recapitula el número y tasa de mensajes por cada escenario que intenta aclarar su falla de recepción.

Tabla 20. Mensajes de prueba faltantes por causa probable.

Mensajes faltantes	181	100%
Falta de energía	120	66.3%
Mensajes ignorados	49	27.1%
Sin razón aparente	12	6.6%

4.4 Mensajes sin audio

Retomando el examen a los registros de operación durante las pruebas dinámicas de caja blanca en la sección 3.6.4, se observó que el receptor funciona como era esperado al recibir los mensajes de prueba pero no funciona como era esperado al recibir los mensajes de alerta.

Según lo estudiado en la sección 1.3, un mensaje de alerta con audio, se compone de 4 elementos: encabezado, tono de alarma, mensaje de audio y fin de mensaje. Partiendo de la información contenida en el encabezado en combinación con el tono de alarma, el receptor debe prepararse para difundir el mensaje de voz a volumen suficiente. La longitud del mensaje de voz no debe exceder los 2 minutos y al recibir el fin de mensaje el receptor debe considerar el mensaje de voz finalizado.

Por otra parte, durante el periodo de prueba del 1 de abril de 2021 al 31 de marzo de 2022 se recibieron 14 mensajes de alerta correspondientes a 5 eventos distribuidos como se muestra en la Tabla 21.

Tabla 21. Mensajes de alerta recibidos por evento.

Evento	Mensajes
21 de junio de 2021 a las 11:30	2
7 de septiembre de 2021 a las 20:48	3
19 de septiembre de 2021 a las 11:30	3
3 de marzo de 2022 a las 8:41	3
18 de marzo de 2022 a las 13:54	3

En contraste con lo esperado, además de replicarse el mensaje de alerta por evento ninguno de los mensajes de alerta recibidos durante el periodo de prueba tienen espacio de un mensaje de voz como se muestra a continuación.

```

2022-03-03,08:41:19,1646318479.7413568,SAME_PRE_DET
2022-03-03,08:41:19,1646318479.9059904,SAME_HDR_DET
2022-03-03,08:41:20,1646318480.5450633,"SAME_HDR_RDY,1"
2022-03-03,08:41:20,1646318480.7181363,SAME_PRE_DET
2022-03-03,08:41:20,1646318480.8320622,SAME_HDR_DET
2022-03-03,08:41:21,1646318481.5320325,"SAME_HDR_RDY,2"
2022-03-03,08:41:21,1646318481.7075624,SAME_PRE_DET
2022-03-03,08:41:21,1646318481.8211234,SAME_HDR_DET
2022-03-03,08:41:22,1646318482.5113442,"SAME_HDR_RDY,3"
2022-03-03,08:41:22,1646318482.6025028,
    "SAME_MESSAGE,-CIV-EQW-000000+0005-832326-XDIF/005-...H."
2022-03-03,08:41:22,1646318482.6189742,SAME_ALERT_ON
2022-03-03,08:41:22,1646318482.6273596,AUDIO_ON
2022-03-03,08:41:22,1646318482.6354125,RELAY_ON
2022-03-03,08:41:22,1646318482.6517386,SAME_EOM_DET
2022-03-03,08:41:22,1646318482.666511,AUDIO_OFF
2022-03-03,08:41:22,1646318482.6740036,RELAY_OFF
2022-03-03,08:41:22,1646318482.6856492,SAME_ALERT_OFF

```

2022-03-03,08:41:23,1646318483.1151667,SAME_PRE_DET

2022-03-03,08:41:23,1646318483.351819,SAME_EOM_DET

2022-03-03,08:41:25,1646318485.3818288,SAME_PRE_DET

2022-03-03,08:41:25,1646318485.5564318,SAME_HDR_DET

Investigando de nueva cuenta los estándares aplicables, tanto los ya estudiados en el capítulo 1 como nuevos, se encontró que en la Gaceta Oficial de la Ciudad de México del 28 de septiembre de 2021 se publicó la Norma Técnica NT-SGIRPC-SDSAS-001-2021 titulada "Sistemas de Difusión Secundaria para el Alertamiento Sísmico".

El apartado 9.14.4 de la Norma Técnica NT-SGIRPC-SDSAS-001-2021 indica:

"Cuando el receptor reciba y reconozca el código EQW que emite el transmisor de VHF, el receptor deberá reaccionar en forma automática con la reproducción del sonido oficial característico del SASMEX-CDMX en un tiempo menor o igual al indicado en el punto 9.10.2., ver Estándares para receptores de Alerta Pública ANSI-CEA-2009-B o actualmente la ANSI-CTA-2009-B¹.

El equipo deberá activar el sonido de alerta sísmica única y exclusivamente para el código EQW, para cualquier otro código deberá cumplir con lo que marca la ANSI-CTA-2009-B." (Gaceta Oficial de la Ciudad de México, 2021)

Con lo que se explica la razón de que los mensajes de alerta sísmica recibidos no tienen espacio para un mensaje de audio como se esperaba.

4.5 Mejoras propuestas

Derivado del análisis realizado en la sección 4.2 y 4.3 de este capítulo, a continuación se presentan una serie de propuestas para solucionar las áreas de oportunidad encontradas durante las pruebas del receptor de alertas.

4.5.1 Batería de respaldo para el banco de pruebas

Como se describió en la sección 3.4.1, el diseño de hardware del receptor no incluye una batería de respaldo ya que se considera su integración con sistemas basados en baterías de plomo por lo que se incluye una entrada de energía de 12V.

¹ https://www.techstreet.com/standards/cta-2009-b-r2016?product_id=1815434

Pero derivado de que el 66% de los mensajes de prueba no recibidos durante la etapa de pruebas se debió a fallas en el suministro de energía, se propone integrar un sistema de alimentación ininterrumpida a la computadora que recolecta los registros de operación del receptor.

Con un sistema de alimentación ininterrumpida se esperaría que la computadora del banco de pruebas también registre el estado del abastecimiento de energía eléctrica. De esta manera, en caso de una falla prolongada en el suministro de energía podríamos conocer de antemano que el receptor estaba funcionando antes de que se agotara la batería de respaldo.

Esta mejora, junto con una segunda etapa de pruebas dinámicas, queda pendiente para desarrollarse en un trabajo posterior.

4.5.2 Procesar encabezados recibidos solo dos veces

Durante el diseño de software en la sección 3.4.2, se tuvo en cuenta que para un buen desempeño del receptor es recomendado un algoritmo que considere al menos dos de los tres encabezados recibidos son idénticos o realice un intento de reconstrucción bit por bit para aceptar como válido el mensaje recibido.

Ya que el circuito integrado receptor de mensajes SAME incluye el algoritmo para combinar múltiples encabezados para incrementar la confianza de la recepción, en la etapa de diseño e implementación del software se consideró utilizarlo en conjunto con un temporizador para detectar la ausencia de tres encabezados.

En la implementación de software al excederse el tiempo de espera de los tres encabezados de mensaje y si se tuvieran más de un encabezado, se interpretaría el mensaje recibido precedido de un aviso de que se superó el tiempo de espera.

Con este análisis, podemos concluir que existe un error en el diseño y/o implementación del software que impide interpretar los mensajes con menos de tres encabezados recibidos.

En la sección 3.7 se corrige este bug como parte de las actualizaciones del diseño del receptor de alertas.

4.5.3 Reproducir el audio de alerta sísmica

Para que el receptor sea útil ante las alertas sísmicas emitidas en la Ciudad de México es necesario incorporar la reproducción del audio oficial de alerta sísmica al recibir una alerta de terremoto.

Debido a que los cambios al estándar SAME presentados en la Norma Técnica NT-SGIRPC-SDSAS-001-2021 fueron publicados después del diseño del receptor de alertas y durante la etapa de pruebas, el proyecto no cuenta con la capacidad de reproducir audio.

Analizando los puntos relevantes al de audio de las alertas en la Norma Técnica NT-SGIRPC-SDSAS-001-2021, se debe considerar:

- Privilegiar la alerta sísmica sobre cualquier otro tipo de mensaje (numeral 9.14.2).
- Cuando el receptor reciba el código EQW deberá reproducir de forma automática el sonido oficial en menos de 5 segundos (numerales 9.10.2 y 9.14.4).
- Se debe activar el sonido de alerta sísmica con duración igual a 60 segundos a máximo volumen (numeral 9.14.1).

Por lo anterior, se propone incorporar un circuito reproductor de audio adicional que brinde al prototipo la funcionalidad requerida con su contraparte correspondiente de ajustes en el software del receptor.

En la sección 3.7 se integra esta característica al receptor, tanto en su hardware como en su software. Sin embargo, queda pendiente para desarrollarse en un trabajo posterior integrar esta funcionalidad en el diseño base del receptor de alertas.

4.6 Prueba en simulacro

El 19 de abril de 2023, se llevó a cabo el Primer Simulacro Nacional 2023 con el propósito de fomentar la cultura de la Protección Civil en la población y contribuir con el fortalecimiento de las capacidades de reacción ante la eventualidad de una emergencia o desastre. (Atlas Nacional de Riesgos, 2023)

Como parte del simulacro, en la Ciudad de México a las 11:00 am se transmitió la alerta sísmica, tanto por el SASMEX utilizando el protocolo SAME como en los altavoces públicos, radiodifusoras y canales de televisión. Este simulacro representó una oportunidad para poner a prueba el receptor en condiciones cercanas a un escenario de emergencia real.

Para llevar a cabo esta prueba, se incorporaron los componentes esenciales de un sistema de alertamiento en un banco de pruebas compuesto por el receptor de alertas con su antena y el módulo ISD1820.

Además, se incorporó una mezcladora de señales conectada a un amplificador de audio con bocina, una fuente de energía ininterrumpida de 12V y una computadora portátil para capturar los registros de operación del receptor. A continuación en la Figura 22 se presenta el banco de pruebas construido para esta prueba.

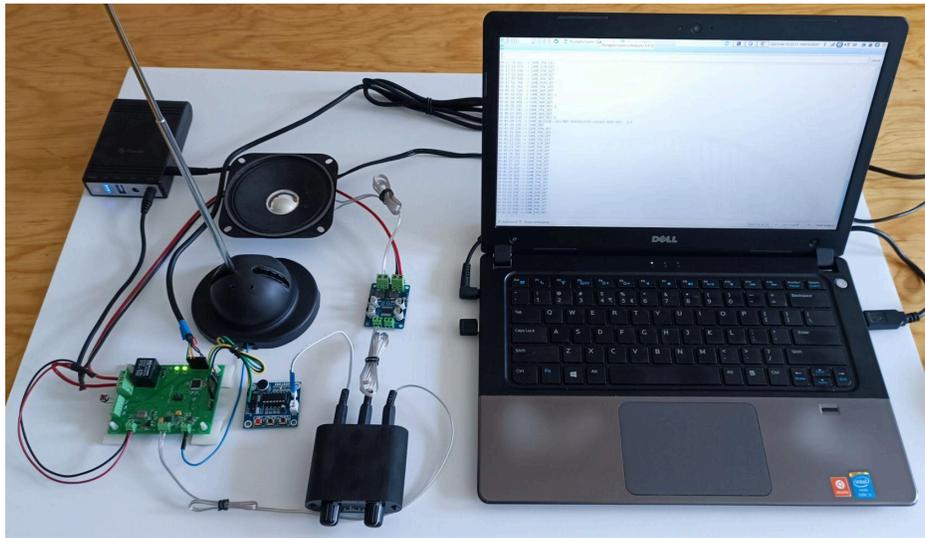


Figura 22. Banco de pruebas para simulacro de alerta sísmica.

A las 11 de la mañana del día señalado se procedió a grabar un video detallado del funcionamiento del receptor durante el simulacro. Los resultados de esta prueba demostraron que el receptor de alertas de riesgos funcionó de acuerdo a las expectativas y requisitos establecidos durante su diseño y desarrollo. El sistema fue capaz de recibir, procesar y transmitir la alerta sísmica de manera correcta.

El video que documenta el funcionamiento del receptor de alertas durante el simulacro del 11 de abril de 2023 y, además demuestra la funcionalidad de su interfaz serial, está disponible en https://youtu.be/aFHcrUs_PhW.

Los registros de operación del receptor durante el simulacro se listan a continuación. Inicialmente, el primer mensaje activa la reproducción del audio oficial de la alerta sísmica. Posteriormente, se reciben dos mensajes consecutivos que, dado que la reproducción del audio está activa, son pasados por alto. Finalmente, la reproducción del sonido de la alerta sísmica se detiene automáticamente un minuto después de haber comenzado.

11:00:01.032 -> SAME_PRE_DET	Inicio mensaje 1
11:00:01.164 -> SAME_HDR_DET	
11:00:01.829 -> SAME_HDR_RDY,1	Encabezado 1
11:00:01.994 -> SAME_PRE_DET	
11:00:02.127 -> SAME_HDR_DET	
11:00:02.791 -> SAME_HDR_RDY,2	Encabezado 2
11:00:02.956 -> SAME_PRE_DET	
11:00:03.089 -> SAME_HDR_DET	
11:00:03.785 -> SAME_HDR_RDY,3	Encabezado 3
11:00:03.818 -> SAME_MESSAGE, -CIV-EQW-000000+0005-832326-XDIF/005-	Mensaje de alerta 1
11:00:03.884 -> CDMX_ALERT_ON	Inicio audio
11:00:03.884 -> RELAY_ON	alerta sísmica
11:00:03.884 -> SAME_EOM_DET	
11:00:04.381 -> SAME_PRE_DET	
11:00:04.580 -> SAME_EOM_DET	Fin mensaje 1
11:00:06.638 -> SAME_PRE_DET	Inicio mensaje 2
11:00:06.804 -> SAME_HDR_DET	
11:00:07.467 -> SAME_HDR_RDY	Encabezado 1
11:00:07.632 -> SAME_PRE_DET	
11:00:07.765 -> SAME_HDR_DET	
11:00:08.461 -> SAME_HDR_RDY	Encabezado 2
11:00:08.627 -> SAME_PRE_DET	
11:00:08.760 -> SAME_HDR_DET	
11:00:09.424 -> SAME_HDR_RDY	Encabezado 3
11:00:09.523 -> SAME_EOM_DET	
11:00:10.020 -> SAME_PRE_DET	
11:00:10.219 -> SAME_EOM_DET	
11:00:12.275 -> SAME_PRE_DET	Inicio mensaje 3
11:00:12.441 -> SAME_HDR_DET	
11:00:13.104 -> SAME_HDR_RDY	Encabezado 1
11:00:13.269 -> SAME_PRE_DET	

11:00:13.402 -> SAME_HDR_DET	
11:00:14.098 -> SAME_HDR_RDY	Encabezado 2
11:00:14.264 -> SAME_PRE_DET	
11:00:14.397 -> SAME_HDR_DET	
11:00:15.061 -> SAME_HDR_RDY	Encabezado 3
11:00:15.691 -> SAME_PRE_DET	
11:00:15.923 -> SAME_EOM_DET	
11:00:17.581 -> SAME_PRE_DET	
11:00:17.780 -> SAME_EOM_DET	
11:00:19.438 -> SAME_PRE_DET	
11:00:19.637 -> SAME_EOM_DET	Fin mensaje 3
11:01:03.876 -> RELAY_OFF	Fin audio
11:01:03.876 -> CDMX_ALERT_OFF	alerta sismica

Estos resultados respaldan la hipótesis de que el receptor de alertas de riesgos es una herramienta efectiva para recibir las alertas de riesgo emitidas mediante el protocolo SAME para difundirlas por medios visuales, auditivos o electrónicos, y proporcionan una base para las conclusiones y recomendaciones finales de este trabajo.

4.7 Ejemplos de configuración

Profundizando en su adaptabilidad, a continuación se presentan diferentes configuraciones del receptor de alertas para su empleo en diferentes escenarios.

Para una casa u oficina pequeña es ideal una configuración con un sistema de respaldo de energía miniatura y un amplificador de audio pequeño como el que se muestra diseñado en la Figura 23.

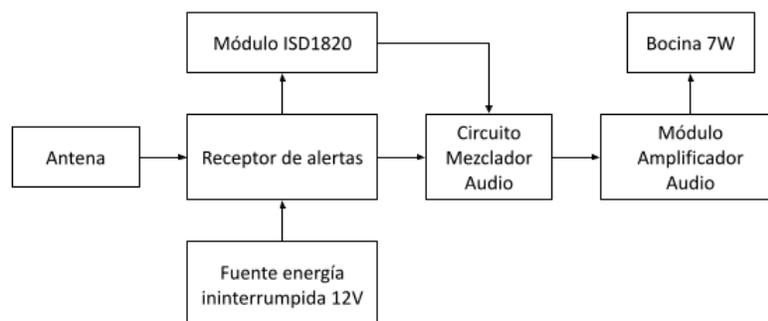


Figura 23. Configuración del receptor de alertas para una casa u oficina pequeña.

Para un mayor efecto en la difusión, se puede integrar con sistema de respaldo de energía basado en baterías de plomo, amplificador de audio de alta potencia y una señal luminosa de alerta, como el que se muestra diseñado en la Figura 24.

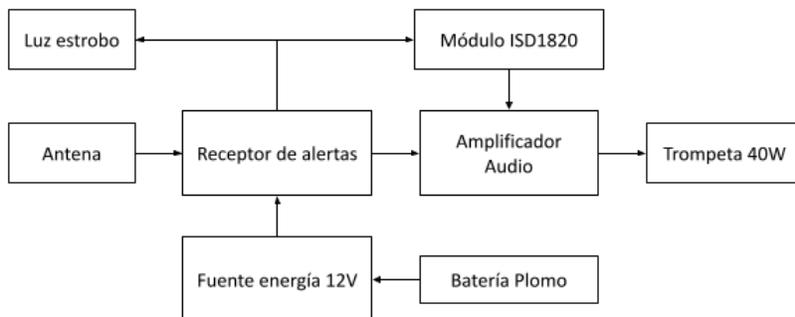


Figura 24. Configuración del receptor de alertas para un mayor efecto en la difusión.

A manera de cierre, se integró el receptor según la configuración mostrada en la Figura 22 para ser utilizado en una casa, comercio u oficina pequeña. En la Figura 25 se muestra a la izquierda el receptor de alertas con su antena extendida y a la derecha su fuente de energía con batería de respaldo.



Figura 25. Prototipo de receptor de alertas para una casa u oficina pequeña.

En la Figura 26 se muestra el interior del gabinete, se encuentra el receptor de alertas al centro, el amplificador de audio en la parte inferior y el módulo ISD1820 con el circuito mezclador de audio en el lateral izquierdo. En la tapa del gabinete está la bocina y en la parte superior los leds de estatus y el botón de prueba.

La lista completa de materiales utilizados en esta aplicación se encuentra en el Apéndice F.



Figura 26. Interior del prototipo de receptor de alertas para una casa u oficina pequeña.

Según los resultados presentados, se puede concluir que el prototipo ha demostrado ser funcional y cumplir con los objetivos establecidos en la especificación de producto, aunque, como cualquier producto, todavía hay áreas que necesitan ser mejoradas. En este capítulo se analizaron los resultados a detalle para proponer un conjunto de mejoras al receptor de alertas para abordar las áreas de oportunidad detectadas.

Las mejoras propuestas incluyen la integración de un sistema de alimentación ininterrumpida, la corrección de un error en la implementación del software, y la incorporación de un circuito reproductor de audio para reproducir el audio oficial de alerta sísmica. Estas mejoras incrementarán la fiabilidad y la utilidad del receptor de alertas en el futuro.

Aunque algunas mejoras quedan pendientes para trabajos posteriores, la implementación de las mejoras propuestas ayudará a garantizar que el receptor de alertas funcione de manera efectiva en situaciones de emergencia.

También, para confirmar el funcionamiento del receptor, se examinó su desempeño durante el Primer Simulacro Nacional 2023 durante el cual se transmitió el aviso de terremoto mediante el protocolo SAME. El sistema funcionó conforme a las expectativas y especificaciones, recibiendo, procesando y transmitiendo de manera precisa la alerta sísmica. Los registros de operación confirmaron un funcionamiento sin contratiempos, incluyendo la reproducción del audio oficial y la correcta interacción con mensajes consecutivos. Estos resultados respaldan nuestra hipótesis inicial y establecen una sólida base para las conclusiones y recomendaciones finales de este trabajo.

Por último, se presentaron diferentes configuraciones del receptor, demostrando su adaptabilidad a diferentes contextos. También se expuso la construcción de una configuración para casa u oficina pequeña, dando cuenta de una posible aplicación del receptor de alertas.

Conclusiones

México es un país con altos niveles de sismicidad por lo que en 2005 se creó el Sistema de Alerta Sísmica Mexicano (SASMEX). Este sistema, a pesar de alertar unos minutos antes de un sismo, al día de hoy no existe en cada uno de los hogares de los ciudadanos mexicanos. Esto debido a la falta de interoperabilidad además del diseño cerrado y el alto costo de los receptores de alertas de riesgo existentes. Por esta razón, en este trabajo de tesis proponemos el uso de sistemas embebidos para la construcción de un receptor de alertas de riesgo efectivo, costeable, de diseño abierto y fácil de mantener.

Así, se diseñó y fabricó un sistema embebido para recibir y difundir alertas de riesgo codificadas con el protocolo SAME. Para lograrlo se estudió el protocolo SAME y se aplicó la metodología de diseño de sistemas embebidos creada por Berger (2002) para fabricar un receptor de alertas de riesgos. Esto requirió de crear un diseño tanto del hardware como del software a usar, implementar estos diseños, probarlos y mejorarlos. Se creó un primer prototipo que se evaluó y posteriormente se mejoró. Todo este proceso se documentó en esta tesis.

El resultado de este trabajo es un sistema embebido capaz de recibir las alertas de riesgo emitidas mediante el protocolo SAME para difundirlas por medios visuales, auditivos o electrónicos. Para cumplir con esta tarea las funciones del sistema incluyen las siguientes.

1. Sintonizar la Banda del Clima para captar alertas de riesgo transmitidas.
2. Procesar los mensajes de alerta codificados con el protocolo SAME.
3. Filtrar los mensajes recibidos por tipo de evento o cobertura geográfica.
4. Comunicar la alerta por medios visuales, auditivos o eléctricos.
5. Transmitir la información digital del mensaje recibido.

Por medio de estas funcionalidades, el receptor es apto para cumplir los requerimientos de uso listados a continuación.

1. Notificar de alertas recibidas audiovisualmente, ya sea integrándose en sistemas existentes (como de audio ambiental o alarma) como en nuevos sistemas.
2. Interconectarse con equipo en oficinas o comercios (como puertas automáticas, elevadores, escaleras mecánicas, etc) para alistarse ante una posible evacuación.
3. Preparar automáticamente maquinaria y equipo industrial de peligros y así reducir afectaciones a sus operadores o a las máquinas mismas.

4. Retransmitir la alerta en medios digitales para amplificar su alcance y permitir su análisis.

Este sistema embebido se evaluó por medio de tres factores: el porcentaje de mensajes captados exitosamente, el costo del receptor y el tiempo de mantenimiento al sistema. En estos tres factores se llegó a los niveles propuestos.

Como parte de este proceso de evaluación, se implementaron mejoras en el diseño original que nos permitieron efectivamente llegar a los niveles propuestos.

Derivado de este proceso, se corrigió un error en el algoritmo de recepción que impedía procesar mensajes con solo dos encabezados recibidos en lugar de tres. También se incluyó la reproducción del audio oficial de alerta sísmica al recibir una alerta de terremoto en lugar de transmitirse como un mensaje de voz.

Dicho lo anterior, podemos afirmar que efectivamente es posible construir un receptor de alertas de riesgos codificadas con el protocolo SAME aplicando la metodología de diseño de sistemas embebidos que funcione correctamente, sea costeable y de fácil mantenimiento.

Además, el receptor diseñado durante este trabajo tiene las siguientes características que lo distinguen de otras alternativas:

1. Es interoperable: Tiene la capacidad de conectarse con otros sistemas de difusión más amplios o ya existentes.
2. Se puede adaptar: Tanto su amplificador de audio, como antena y fuente de alimentación de energía son componentes externos que pueden ser elegidos de acuerdo a diferentes necesidades y contextos de uso.
3. Su código y diseño es abierto: Se puede mejorar y reproducir libremente, permitiendo la mejora continua del receptor de alertas y ampliando su alcance.
4. Costo asequible: Comparado con alternativas comerciales, su costo es competitivo y sus prestaciones equivalentes o superiores.

Es importante mencionar que uno de los retos más importantes de este trabajo fue la planeación de tiempos. La fabricación de los prototipos llevó considerablemente más tiempo del planeado en el cronograma del plan de administración de la ingeniería propuesto. Específicamente la adquisición de componentes especializados tomó varias semanas extras a

la fábrica, lo cual incrementó el tiempo total de diseño considerablemente. A pesar de esto, el objetivo descrito en este trabajo de tesis se cumplió.

Como resultado de las pruebas y evaluaciones realizadas al prototipo construido, se propone como trabajo futuro actualizar el diseño del receptor para considerar la norma NT-SGIRPC-SDSAS-001-2021 desde la especificación de producto y así certificar al receptor ante la Secretaría de Gestión Integral de Riesgos y Protección Civil (SGIRPC) como sistema de difusión secundaria autorizado para el alertamiento sísmico. Cumpliendo con esta certificación, el receptor de alertas podría satisfacer los requerimientos de normas como la NOM-001-STPS-2008 "Edificios, locales, instalaciones y áreas en los centros de trabajo - Condiciones de seguridad" en la que se determina es deber de las empresas hacer las instalaciones de trabajo un lugar seguro, o la misma NT-SGIRPC-SDSAS-001-2021 que también indica que los establecimientos mercantiles e industriales tanto públicos como privados, deben contar con la recepción de avisos del sistema de alerta sísmica.

Además, se propone mejorar el diseño de hardware del receptor incluyendo una batería de respaldo para garantizar el correcto funcionamiento de este incluso cuando el servicio eléctrico fallé.

Concluimos que con este trabajo, se ha logrado documentar las bases técnicas para el desarrollo de productos o servicios que incrementen el aprovechamiento y la cobertura de los sistemas de alertamiento de riesgos.

Apéndice A. Código de eventos SAME

En este apartado, se enlistan los códigos de eventos definidos por la NOAA (2011) de acuerdo a su ámbito. Primero, en la Tabla A.1 se encuentran los códigos relacionados con fenómenos meteorológicos y en las Tabla A.2 y A.3 los que no tienen relación con el clima. Por último en la Tabla A.4 y A.5 se encuentran los códigos administrativos y de control del NWS.

Según la NOAA (2011, p. A-15), con la excepción de los códigos TOR (Aviso de tornado), SVR (Aviso de tormenta severa), EVI (Evacuación inmediata) y BLU (Alerta azul) más los códigos de control del NWS, el tercer carácter significa:

- W del inglés *Warning* o Aviso, para sucesos que suponen una amenaza significativa y la probabilidad de que se presenten son altas y el tiempo de ocurrencia relativamente corto.
- A del inglés *Watch* o Vigilancia, para sucesos que cumplen la clasificación de aviso pero la probabilidad de que se presenten o el tiempo de ocurrencia son inciertos.
- E del inglés *Emergency* o Emergencia, para sucesos que indirectamente pueden provocar situaciones peligrosas.
- S del inglés *Statement* o Comunicado, para mensajes de seguimiento a un aviso, vigilancia o emergencia.

Tabla A.1. Códigos de fenómenos meteorológicos (NOAA, 2011, p. A-12).

Evento	Código
Aviso de ventisca	BZW
Vigilancia de inundaciones costeras	CFA
Aviso de inundaciones costeras	CFW
Aviso de tormenta de polvo	DSW
Advertencia de viento extremo	EWV
Vigilancia de inundaciones repentinas	FFA
Aviso de inundaciones repentinas	FFW
Comunicado de inundaciones repentinas	FFS
Vigilancia de inundación	FLA

Evento	Código
Aviso de inundación	FLW
Advertencia de inundación	FLS
Vigilancia de vientos fuertes	HWA
Aviso de vientos fuertes	HWW
Vigilancia de huracán	HUA
Aviso de huracán	HUW
Comunicado de huracán	HLS
Vigilancia de tormenta severa	SVA
Aviso de tormenta severa	SVR
Advertencia de mal clima	SVS
Aviso marítimo especial	SMW
Aviso meteorológico especial	SPS
Advertencia de tormenta de nieve	SQW
Advertencia de marejada ciclónica	SSW
Vigilancia de marejada ciclónica	SSA
Vigilancia de tornado	TOA
Aviso de tornado	TOR
Vigilancia de tormenta tropical	TRA
Aviso de tormenta tropical	TRW
Vigilancia de tsunami	TSA
Aviso de tsunami	TSW
Vigilancia de tormenta de nieve	WSA
Aviso de tormenta de nieve	WSW

Tabla A.2. Códigos de sucesos nacionales sin relación con el clima (NOAA, 2011, p. A-13).

Evento	Código
Anuncio de medidas de emergencia	EAN
Mensaje del Centro Nacional de Información (NIC) de EUA	NIC
Prueba periódica nacional	NPT
Prueba mensual obligatoria	RMT
Prueba semanal obligatoria	RWT

Tabla A.3. Códigos de sucesos estatales o locales sin relación con el clima (NOAA, 2011, p. A-14).

Evento	Código
Mensaje administrativo	ADR
Vigilancia de avalancha	AVA
Aviso de avalancha	AVW
Alerta azul	BLU
Emergencia de rapto de menores	CAE
Aviso de peligro civil	CDW
Mensaje de emergencia civil	CEM
Aviso de terremoto	EQW
Evacuación inmediata	EVI
Aviso de fuego	FRW
Aviso de materiales peligrosos	HMW
Aviso de las autoridades de la ley	LEW
Emergencia de área local	LAE
Interrupción del servicio telefónico del 911	TOE
Aviso de riesgo nuclear	NUW

Evento	Código
Aviso de peligro radiológico	RHW
Aviso de refugio en el lugar	SPW
Aviso de actividad volcánica	VOW

Tabla A.4. Códigos de sucesos administrativos (NOAA, 2011, p. A-14).

Evento	Código
Anuncio de mensaje en la red	NMN
Aviso de Práctica/Demostración	DMO

Tabla A.5. Códigos de control del NWR (NOAA, 2011, p. A-14).

Evento	Código
Transmisor de portadora apagado	TXF
Transmisor de portadora encendido	TXO
Transmisor de respaldo activado	TXB
Transmisor principal activado	TXP

Apéndice B. Diseño de hardware

A continuación, se presentan detalles del hardware del receptor de alertas diseñado en la sección 3.4.1.

El apartado B.1 enumera las funciones de los pines de cada uno de los conectores del circuito impreso.

En el apartado B.2 se presentan imágenes correspondientes a cada una de las capas del circuito impreso a fabricar y en el apartado B.3 se listan los componentes para ensamble del receptor de alertas.

El apartado B.4 contiene la orden de fabricación de los circuitos impresos y el ensamble de componentes.

Además, los archivos fuente con el diseño del circuito impreso y ensamble de componentes se encuentran en la carpeta *hardware* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

B.1 Configuración de conectores

Tabla B.1.1. Configuración del conector J1: Puerto de programación ICSP.

Pin	Función
1	MISO
2	3.3V
3	SCK
4	MOSI
5	Reset
6	GND

Tabla B.1.2. Configuración del conector J2: Puerto de comunicación serie.

Pin	Función
1	GND
2	CTS
3	3.3 V
4	TXD
5	RXD
6	RTS

Tabla B.1.3. Configuración del conector J3: Puerto de expansión analógico digital.

Pin	Función
1	5 V
2	3.3 V
3	A0
4	A1
5	A2
6	A3
7	A6
8	GND

Tabla B.1.4. Configuración del conector J4: Entrada de energía auxiliar.

Pin	Función
1	3.3 V
2	GND
3	5 V

Tabla B.1.5. Configuración del conector X1: Antena receptora.

Pin	Función
1	Antena
2	GND

Tabla B.1.6. Configuración del conector X2: Salida de audio.

Pin	Función
1	Canal audio derecho
2	GND
3	Canal audio izquierdo

Tabla B.1.7. Configuración del conector X3: Relevador.

Pin	Función
1	Contacto normalmente abierto
2	Contacto común
3	Contacto normalmente cerrado

Tabla B.1.8. Configuración del conector X4: Leds y botón externos.

Pin	Función
1	Entrada pull-down para botón de prueba normalmente abierto
2	GND
3	Salida para led 1 de 20 mW máximo
4	GND
5	Salida para led 2 de 20 mW máximo
6	GND

Tabla B.1.9. Configuración del conector X5: Entrada de energía.

Pin	Función
1	Entrada de 6 a 36 V de corriente continua
2	GND

B.2 Diseño de la placa electrónica

Las dos primeras figuras en seguida representan visualmente el diseño final de los circuitos impresos.

Por otro lado, las figuras posteriores muestran representaciones visuales de los archivos tipo Gerber y NC que fueron entregados para la fabricación de los circuitos impresos.

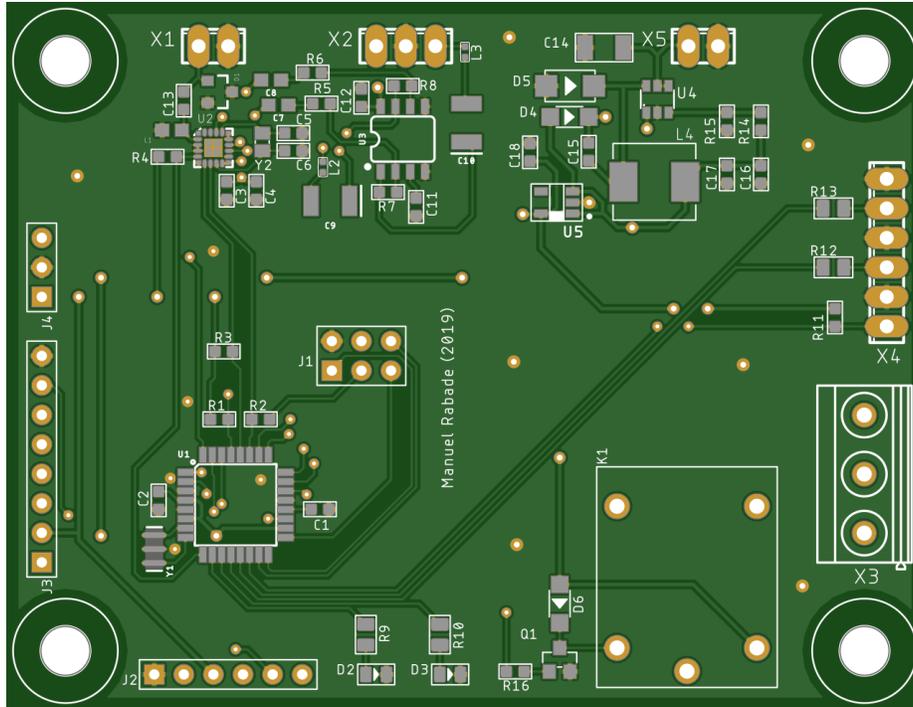


Figura B.2.1. Vista superior de la placa electrónica.

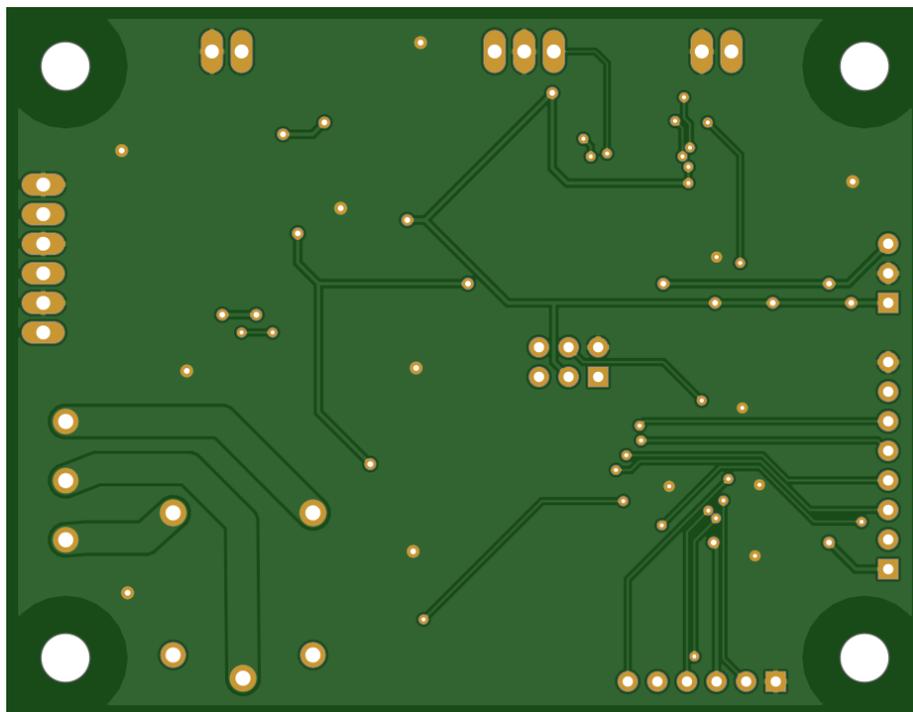


Figura B.2.2. Vista inferior de la placa electrónica.

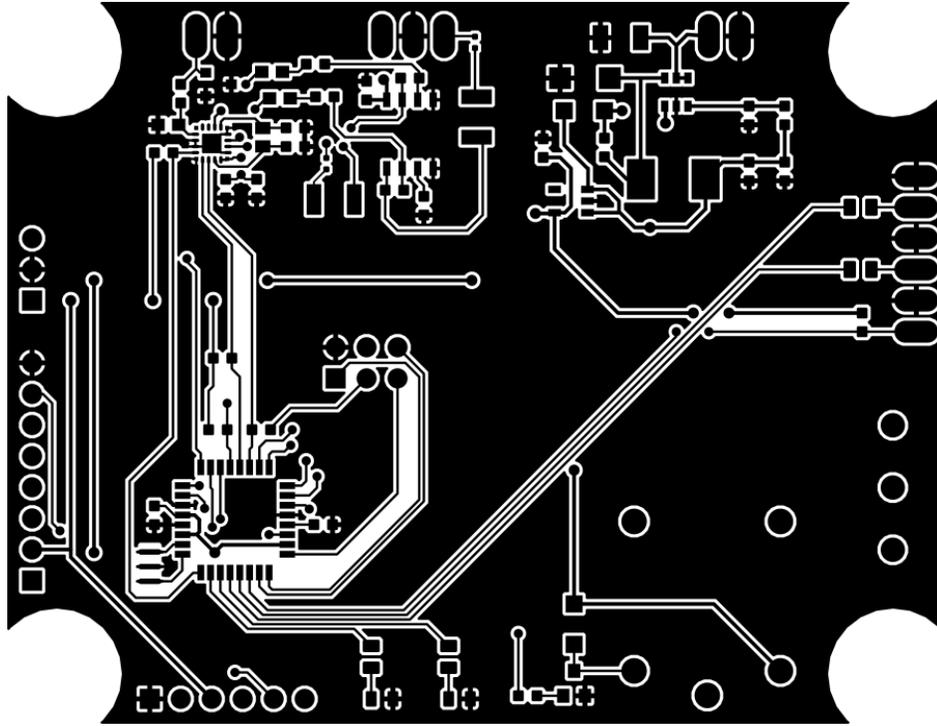


Figura B.2.3. Diseño de la capa de cobre superior.

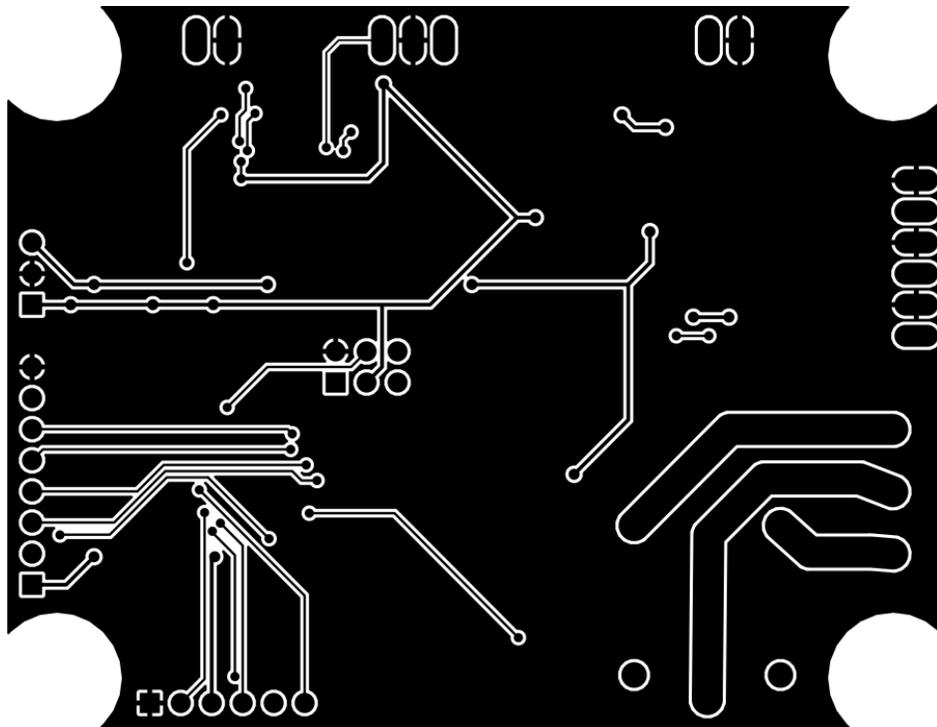


Figura B.2.4. Diseño de la capa de cobre inferior.

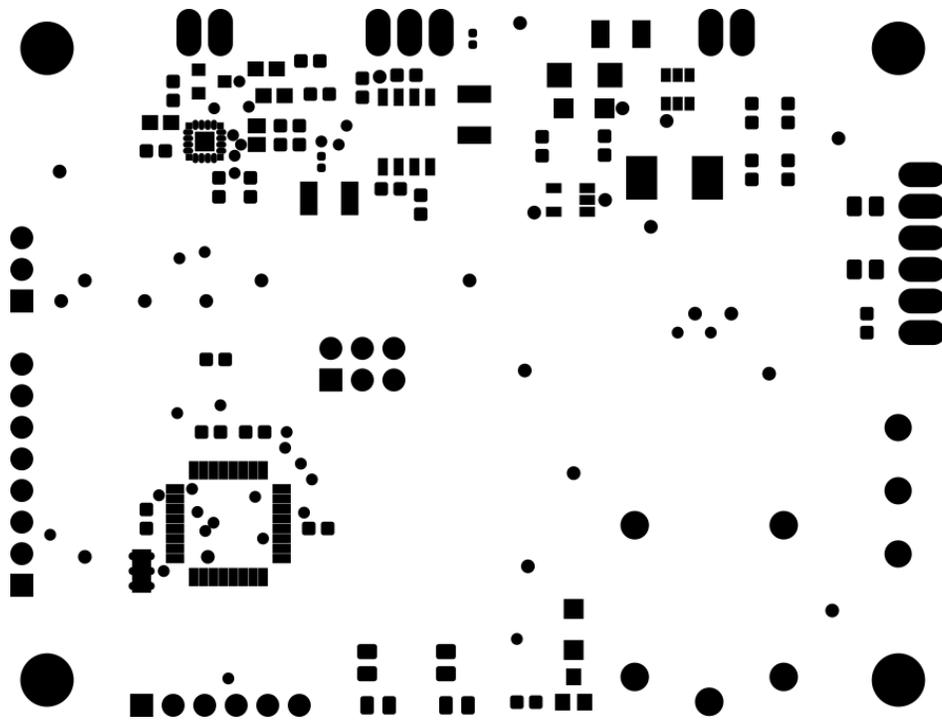


Figura B.2.5. Diseño de la máscara de soldadura superior.

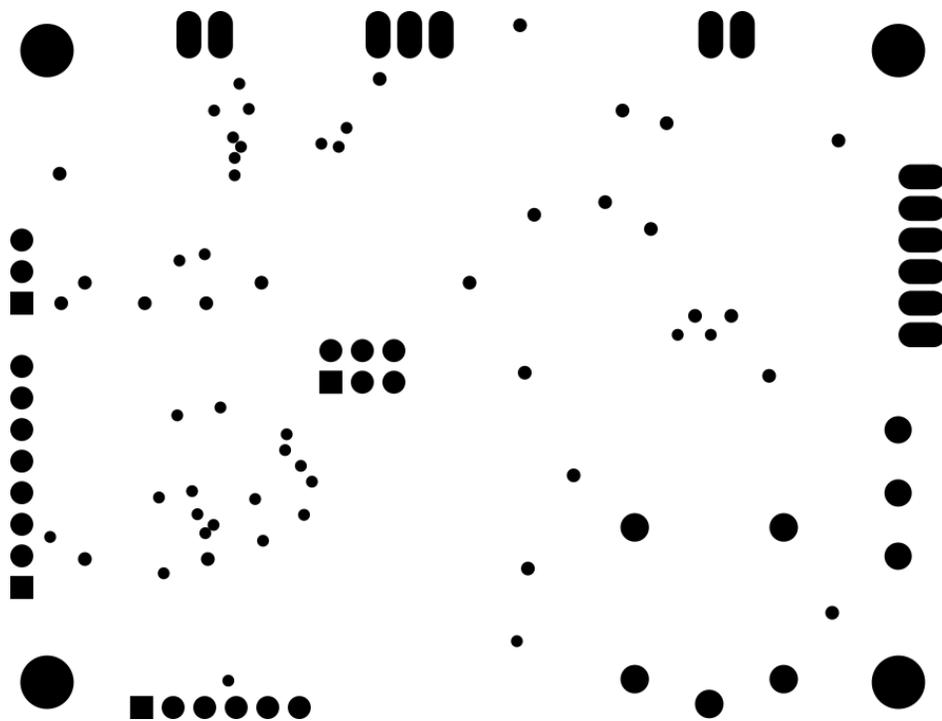


Figura B.2.6. Diseño de la máscara de soldadura inferior.



Figura B.2.7. Diseño de la pasta para soldar superior.

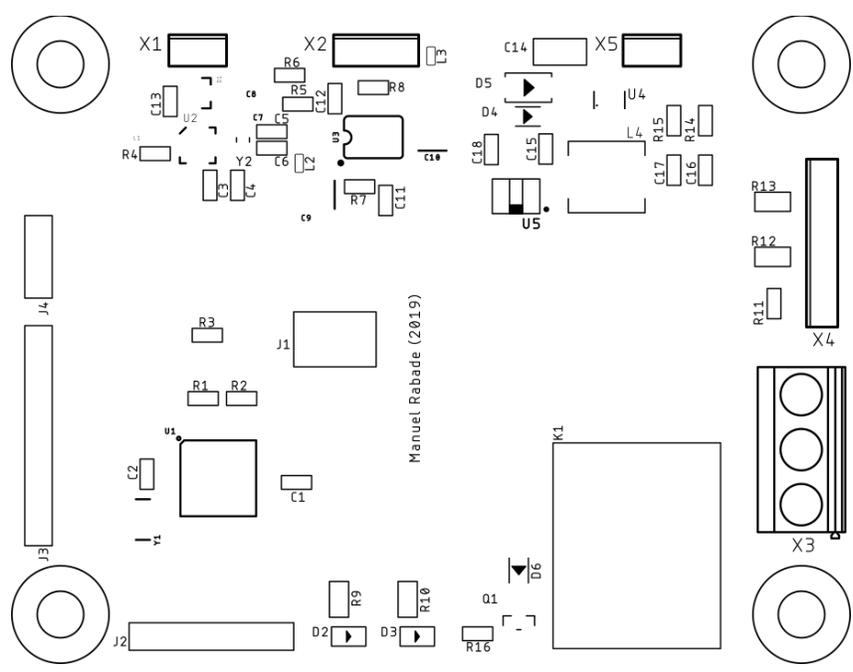


Figura B.2.8. Diseño de la serigrafía superior.

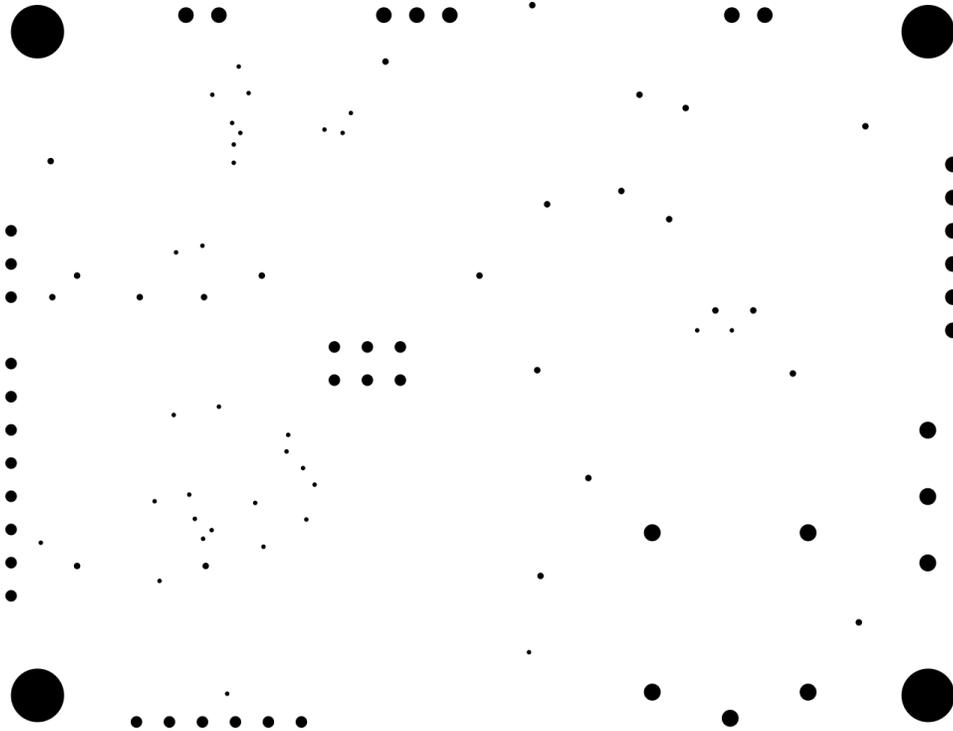


Figura B.2.9. Diseño de perforaciones.

B.3 Listado de componentes

En seguida se enlistan los componentes para fabricar el receptor de alertas. El número de parte corresponde al código de cada componente en la biblioteca del fabricante de los PCBA y si el componente no estuviera disponible se usa el código del distribuidor DigiKey con el URL de la parte en su sitio web.

Tabla B.3.1. Listado de componentes del receptor de alertas.

Parte	Número de parte	Cantidad	URL
C1, C2, C3, C11, C12, C15, C18	CC0603KRX7R9BB104	7	
R16	RC0603JR-071K5L	1	
R1, R4, R11, R15	RC0603JR-0710KL	4	
C16, C17	CC0603MRX5R5BB106	2	
C14	CL31A106KBHNNNE	1	
R9, R10, R12, R13	RC0805JR-07120RL	4	
C13	GRM1885C1H102JA01D	1	
L2, L3	BLM18BD252SN1D	2	
R5, R6	RC0603FR-0720KL	2	
C4	CC0603KRX7R9BB223	1	
C5, C6	CC0603JRNPO9BN220	2	
L4	SHC0603-220M	1	
Y2	FC-12M	1	
C7, C8	CC0603KRX7R6BB394	2	
R2, R3	RC0603JR-074K7L	2	
D5	1N5819	1	
R7, R8	RC0603JR-0747KL	2	
C9, C10	TAJB476K010RNJ	2	
R14	WR06X5232FTL	1	
L1	SDCL1608C56NJTDF	1	

Parte	Número de parte	Cantidad	URL
Y1	CSTCE8M00G55-R0	1	
X1	1725656	1	
U1	ATMEGA328P-AU	1	
X2	1725669	1	
J4	P125-1103A1BS116A1	1	
D1	CM1213A-01SO	1	https://www.digikey.com/product-detail/en/on-semiconductor/CM1213A-01SO/CM1213A-01SOOSCT-ND/3462265
J3	P125-1108A1BS116A1	1	
D2, D3	17-215SYGC/S530-E2/TR8	2	
K1	HLS8L-DC5V-S-C	1	
J1	P125-12A1BS116A1	1	
X4	1725698	1	
U4	MCP16301T-I/CHY	1	
Q1	MMBT3904LT1G	1	
D4, D6	MMSD4148	2	
X5	1725656	1	
X3	1715734	1	
J2	P125-1106A1BS116A1	1	
U2	SI4707-B20-GM	1	https://www.digikey.com/product-detail/en/silicon-labs/SI4707-B20-GM/336-2147-ND/2686997
U5	TLV70233DBVR	1	
U3	TPA6111A2DR	1	https://www.digikey.com/product-detail/en/texas-instruments/TPA6111A2DR/296-10846-1-ND/381452

B.4 Fabricación de prototipos

En seguida, se muestra la orden de fabricación de los circuitos impresos y ensamble de componentes del receptor de alertas.

5/26/2019 Free Assembly for 5 PCBs

Free Assembly for 5 PCBs. No Set-up Cost, No Assembly Cost, No Operation Fee. FREE Shipping [Order Now](#)

seeed
The IoT Hardware Enabler

Fusion PCB Assembly PCB/PCBA Premium PCB/PCBA Free Assembly for 5 PCBs DPL 3D Printing PCB Layout Kitting Service More Support Centre English

Free Assembly for 5 PCBs

Seeed Fusion is committed to making hardware manufacturing more accessible and affordable. In this special event, we are sponsoring you to try our PCB assembly service practically for free! Just purchase the materials, and the rest is free. Maximum 5 pieces, one offer per customer.

Maximum 5 pieces Free Assembly Starts from Seeed or others
Free Setup Cost Free Shipping Free operation fee

PCB

[Add Gerber Files](#)

Only ZIP or RAR files accepted. Maximum size 20MB.

1143285_receptor_same_pcb_20190527.zip [Gerber Viewer](#)

Base Material Aluminum Flexible Boards

No. of Layers

PCB Dimensions * * Units in mm 61.0 mm * 78.7mm

PCB Quantity

No. of Different Designs [Example](#)

PCB Thickness * Units in mm

PCB Color

Surface Finish

Minimum Solder Mask Dam

Copper Weight

Minimum Drill Hole Size

Trace Width / Spacing

Blind or Buried Vias

Plated Half-holes / Castellated Holes

Impedance Control

PCB Cost

USD\$4.90

Base Material	FR-4 TG130
No. of Layers	2 layers
PCB Dimensions	61.0mm * 78.7mm
PCB Quantity	10
No. of Different Designs	1
PCB Thickness	1.60mm
PCB Color	Green
Surface Finish	HASL
Minimum Solder Mask Dam	0.4mm
Copper Weight	1oz.
Minimum Drill Hole Size	0.3mm
Trace Width / Spacing	6/6 mil
Blind or Buried Vias	No
Plated Half-holes / Castellated Holes	No
Impedance Control	No

PCBA Cost

USD\$219.14

Setup Cost	\$0.00 \$50.00
Consumptive Material Cost	USD\$15.00
Assembly Cost	\$0.00 \$255.31
Component Cost	USD\$204.14
Operation Fee	\$0.00 \$148.07

Sub-Total **USD\$224.04** ~~USD\$677.42~~

Production Time 20 Working Days

Weight 0.24kg

Shipping Calculated at Checkout

[Add to Cart](#)

Your final payment is subject to review for the component(s) that our system cannot match.

PCB Assembly More than 15,000 in-stock components and FREE express shipping all over the world

[Contact Support](#)

https://www.seeedstudio.com/free-assembly-for-5-pcbs.html?utm_source=facebookPCB&utm_medium=cpc&utm_campaign=freesas... 1/4

Figura B.4.1 Página 1 de la orden de fabricación y ensamble de componentes.

Add Assembly Drawing & Pick and Place File

Only ZIP or RAR files accepted. Maximum size 20MB.

1143286_receptor_same_pcba_2019_0527.zip

- [How to Export PCB Pick and Place/XY Files?](#)
- [How to Export PCB Assembly Drawings/Fabrication Files?](#)
- [Seed Fusion PCB Assembly Service Quick Questions.](#)

PCBA Qty

Add BOM File

Only CSV, XLS or XLSX files accepted.

receptor_same_bom.xlsx

How to shorten the PCBA production time without paying more? Check out Seeed's 15,000+ in-stock OPL Now

There are 39 component(s) in your BOM

	Designator	MPN	SeeedSKU	Description	Quantity	Waste quantity	Unit Price	Amount	Procurement Time
<input checked="" type="checkbox"/>	R7, R8	RC0603JR-0747KL	301010084	SMD RES 47K-5%-1/10W	2 * 5	5	0.01	0.1500	1-2 days
<input checked="" type="checkbox"/>	R16	RC0603JR-071KSL	301010114	SMD RES 1.5K-5%-1/10W	1 * 5	5	0.01	0.1000	1-2 days
<input checked="" type="checkbox"/>	R5, R6	RC0603FR-0720KL	301010288	SMD RES 20K-1%-1/10W	2 * 5	5	0.01	0.1500	1-2 days
<input checked="" type="checkbox"/>	R2, R3	RC0603JR-074K7L	301010290	SMD RES 4.7K-5%-1/10W	2 * 5	5	0.01	0.1500	1-2 days
<input checked="" type="checkbox"/>	R1, R4, R11, R15	RC0603JR-0710KL	301010293	SMD RES 10K-5%-1/10W	4 * 5	5	0.01	0.2500	1-2 days
<input checked="" type="checkbox"/>	C4	CC0603KRX7R9BB223	302010095	SMD CAP Ceramic 22nF-50V-10%-X7R;0603	1 * 5	5	0.01	0.1000	1-2 days
<input checked="" type="checkbox"/>	C1, C2, C3, C11, C12, C15, C18	CC0603KRX7R9BB104	302010138	SMD CAP Ceramic 100nF-50V-10%-X7R	7 * 5	7	0.01	0.4200	1-2 days
<input checked="" type="checkbox"/>	C5, C6	CC0603JRNPO9BN220	302010143	SMD CAP Ceramic 22pF-50V-5%-NPO	2 * 5	5	0.01	0.1500	1-2 days
<input checked="" type="checkbox"/>	C16, C17	CC0603MRX5R5BB106	302010286	SMD CAP Ceramic 10uF-6.3V-20%-X5R	2 * 5	5	0.03	0.4500	1-2 days
<input checked="" type="checkbox"/>	C13	GRM1885C1H102JA01D	302010341	SMD CAP Ceramic;1nF-50V-5%-COG	1 * 5	5	0.02	0.2000	1-2 days
<input checked="" type="checkbox"/>	C9, C10	TAJB476K010RNJ	302020020	SMD CAP Tantalum 47uF-10V 10%	2 * 5	5	0.09	1.3500	1-2 days
<input checked="" type="checkbox"/>	L2, L3	BLM18BD252SN1D	303030020	SMD Ferrite Bead 2.5k-50mA	2 * 5	5	0.01	0.1500	1-2 days
<input checked="" type="checkbox"/>	D4, D6	MMSD4148	304010007	SMD Diode Switching 100V-200mA	2 * 5	1	0.05	0.5500	1-2 days
<input checked="" type="checkbox"/>	D5	1N5819	304020027	SMD Diode Schottky 40V-1A	1 * 5	1	0.01	0.0600	1-2 days
<input checked="" type="checkbox"/>	D2, D3	17-215SYGJ/S530-E2/TR8	304090057	SMD LED Clear-Green	2 * 5	2	0.03	0.3600	1-2 days
<input checked="" type="checkbox"/>	Q1	MMBT3904LT1G	305010013	SMD Transistors NPN 40V 200mA	1 * 5	1	0.01	0.0600	1-2 days
<input checked="" type="checkbox"/>	Y2	FC-12M	306010020	SMD Crystal 32.768KHz-12.5pF-20ppm-90k	1 * 5	1	0.33	1.9800	1-2 days
<input checked="" type="checkbox"/>	U1	ATMEGA328P	31001004	SMD IC MCU;AVR;8-Bit;3	1 * 5	0	1.28	6.4000	1-2 days

Contact Support

https://www.seeedstudio.com/free-assembly-for-5-pcb.html?utm_source=facebookPCB&utm_medium=cpc&utm_campaign=freesas... 2/4

Figura B.4.2 Página 2 de la orden de fabricación y ensamble de componentes.

	-AU	9	2KB;16MHz;1.8-5.5V						
<input checked="" type="checkbox"/>	K1	HLS8L-DC5V-S-C	315030000	DIP Power Relay 5V-7A	1 * 5	1	0.23	1.3800	1-2 days
<input checked="" type="checkbox"/>	J1	P125-12A1BS116A1	320020077	DIP Black Male Header	1 * 5	1	0.01	0.0600	1-2 days
<input checked="" type="checkbox"/>	J2	P125-1106A1BS116A1	320020078	DIP Black Male Header VERT	1 * 5	1	0.01	0.0600	1-2 days
<input checked="" type="checkbox"/>	J4	P125-1103A1BS116A1	320020079	DIP Black Male Header VERT	1 * 5	1	0.01	0.0600	1-2 days
<input checked="" type="checkbox"/>	J3	P125-1108A1BS116A1	320020080	DIP Black Male Header VERT	1 * 5	1	0.02	0.1200	1-2 days
<input checked="" type="checkbox"/>	L4	SHC0603-220M	303010525	SMD Inductor 22uH-2 0%-0.139R-2.2A,2P-L7.1 *W6.6*H3.0mm	1 * 5	5	0.10	1.0000	1-2 days
<input checked="" type="checkbox"/>	R9, R10, R12, R13	RC0805JR-07120RL	301010351	SMD RES 120R-5%-1/8 W,0805	4 * 5	5	0.01	0.2500	1-2 days
<input checked="" type="checkbox"/>	C14	CL31A106KBHNNNE	508000001	CAP CER 10UF 50V XSR 1206	1 * 5	5	0.74	7.4000	15-20 days
<input checked="" type="checkbox"/>	C7, C8	CC0603KRX7R6BB394	508000001	0603 390nF ±10% 10V X7R	2 * 5	1	0.07	0.7700	1-2 days
<input checked="" type="checkbox"/>	R14	WR06X523ZFTL	508000001	贴片电阻 0603 52.3KΩ ±1% 1/10W ±100ppm/°C	1 * 5	1	0.02	0.1200	1-2 days
<input checked="" type="checkbox"/>	L1	SDCL1608C56NJTDF	508000001	0603 56nH ±5% DCR=75 0mΩ Q=12@100MHz 30 0mA SRF=1.1GHz 应用: 射频电路	1 * 5	1	0.02	0.1200	1-2 days
<input checked="" type="checkbox"/>	Y1	CSTCE8M00G55-R0	508000001	8MHz 33pF ±0.5%	1 * 5	1	0.16	0.9600	1-2 days
<input checked="" type="checkbox"/>	X5,X1	1725656	508000001	间距2.54mm 2位 6A 160V 螺钉式 直插 配线26-20 AWG(0.14-0.5)mm ²	2 * 5	1	0.55	6.0500	1-2 days
<input checked="" type="checkbox"/>	X2	1725669	508000001	固定接线端子 3P 2.54mm 90DEG	1 * 5	1	2.58	15.4800	15-20 days
<input checked="" type="checkbox"/>	D1	CM1213A-01SO	508000001	ESD 抑制器/TVS 二极管 1-ch, 8KV/1pF Ultra	1 * 5	1	0.47	2.8200	15-20 days
<input checked="" type="checkbox"/>	X4	1725698	508000001	固定接线端子 6P 2.54mm 90DEG	1 * 5	1	4.96	29.7600	15-20 days
<input checked="" type="checkbox"/>	U4	MCP16301T-I/CHY	508000001	IC REG BUCK ADJ 0.6A SOT23-6	1 * 5	0	0.73	3.6500	1-2 days
<input checked="" type="checkbox"/>	X3	1715734	508000001	间距5.08mm 3位 17.5A 4 00V 可拼接 螺钉式 直插 配线26-14AWG(0.14-1.5) mm ² 可与两位(1715721) 拼成任意PIN位	1 * 5	1	0.42	2.5200	1-2 days
<input checked="" type="checkbox"/>	U2	SI4707-B20-GM	508000001	RF RX WB 162.4-162.55 MHz 20QFN	1 * 5	1	18.56	111.3600	15-20 days
<input checked="" type="checkbox"/>	U5	TLV70233DBVR	508000001	IC REG LDO 3.3V 0.3A SOT23-5	1 * 5	0	0.14	0.7000	1-2 days
<input checked="" type="checkbox"/>	U3	TPA6111A2DR	508000001	音频放大器 150mW Stereo Power	1 * 5	1	1.12	6.7200	15-20 days

Contact Support

https://www.seedstudio.com/free-assembly-for-5-pcbs.html?utm_source=facebookPCB&utm_medium=cpc&utm_campaign=freesas... 3/4

Figura B.4.3 Página 3 de la orden de fabricación y ensamble de componentes.

Apéndice C. Diseño de software

En este apartado, se expone el diseño del software del receptor de alertas elaborado en la sección 3.4.2.

El apartado C.1 presenta los diagramas UML del diseño de software receptor de alertas.

El protocolo de comunicación para operar el receptor de alertas por su puerto serial se describe en el apartado C.2.

El código fuente del programa principal, bibliotecas y pruebas tanto unitarias como de integración se encuentran en la carpeta *software* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same>.

C.1 Diagramas UML

Como se mencionó al inicio de la sección 3.4.2, para diseñar y documentar el software del receptor de alertas se emplearán los diagramas de clases, casos de uso, máquina de estados, secuencia y estructura compuesta por su relevancia en el diseño e implementación de sistemas embebidos (Apvrille, 2012).

Partiendo de los requisitos del software, se elaboró el diagrama de casos de uso presentado en la Figura C.1.1.

De acuerdo con la arquitectura de software y elementos propuestos en la sección 3.2.3, se diseñaron las clases descritas en el diagrama de clases expuesto en la Figura C.1.2.

El programa principal está representado por medio de diagramas de máquina de estados. En la Figura C.1.3 están descritas sus dos funciones principales: *setup* para la inicialización y *loop* para el bucle infinito. En la Figura C.1.4 se detallan los bloques que componen al bucle infinito.

En la Figura C.1.5 por medio de un diagrama de secuencia se ejemplifica la obtención de los elementos de un mensaje SAME captado por el receptor, desde el programa principal hasta el bus I2C pasando por el controlador de dispositivo del circuito integrado Si4707.

Para concluir, en la Figura C.1.6 con un diagrama de estructura compuesta se presenta la interacción entre los módulos construidos y el software de aplicación.

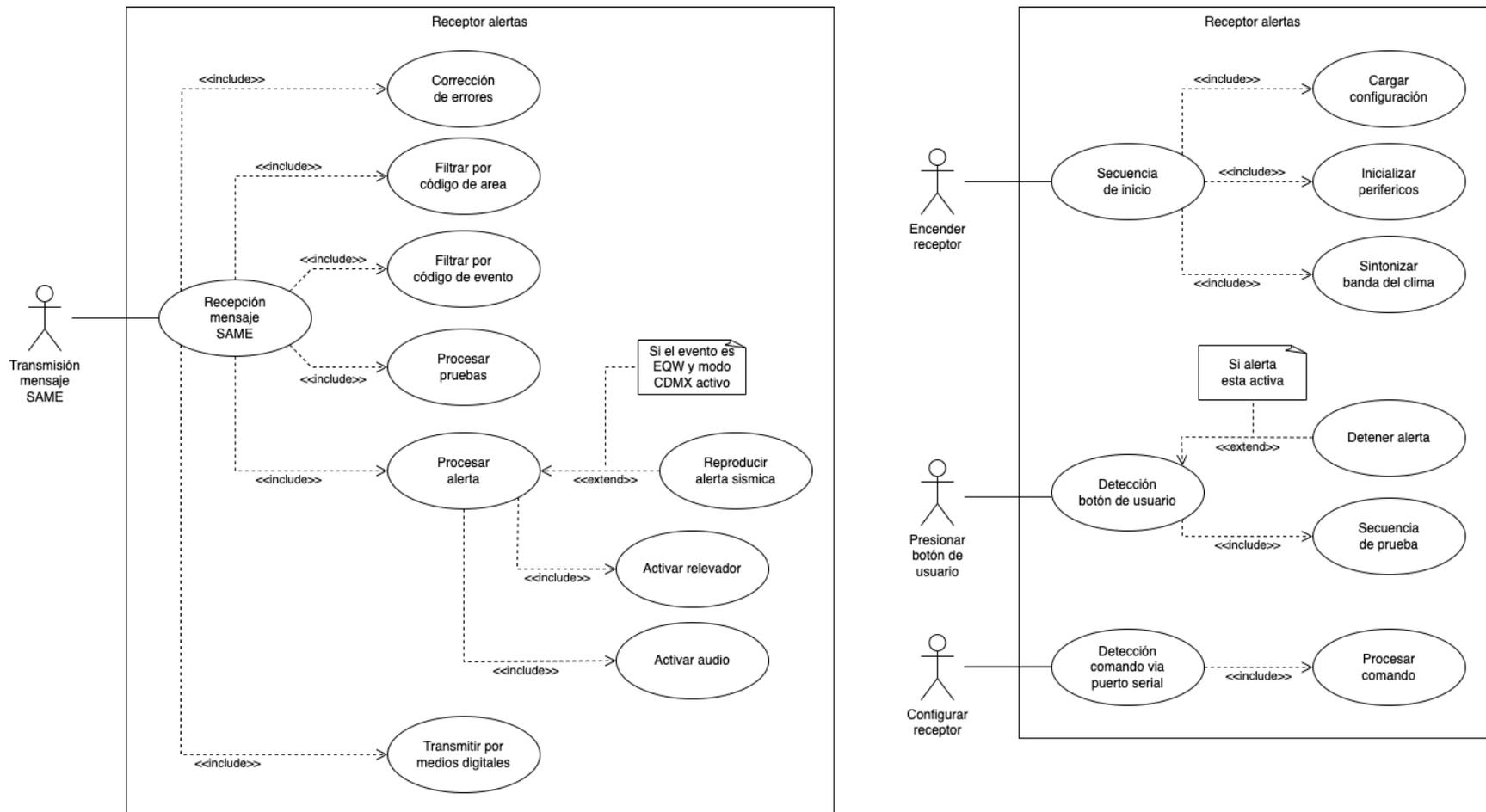


Figura C.1.1 Diagrama de casos de uso.

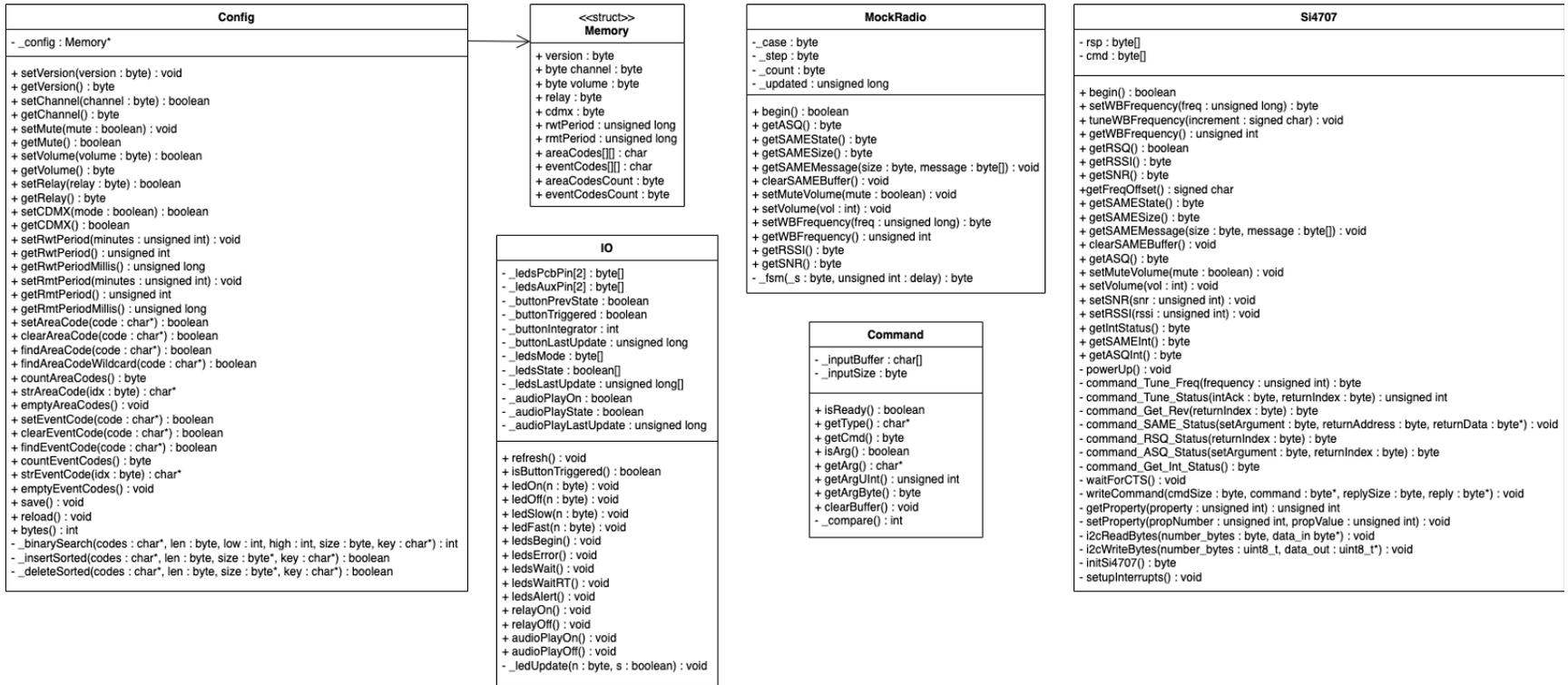


Figura C.1.2 Diagrama de clases.

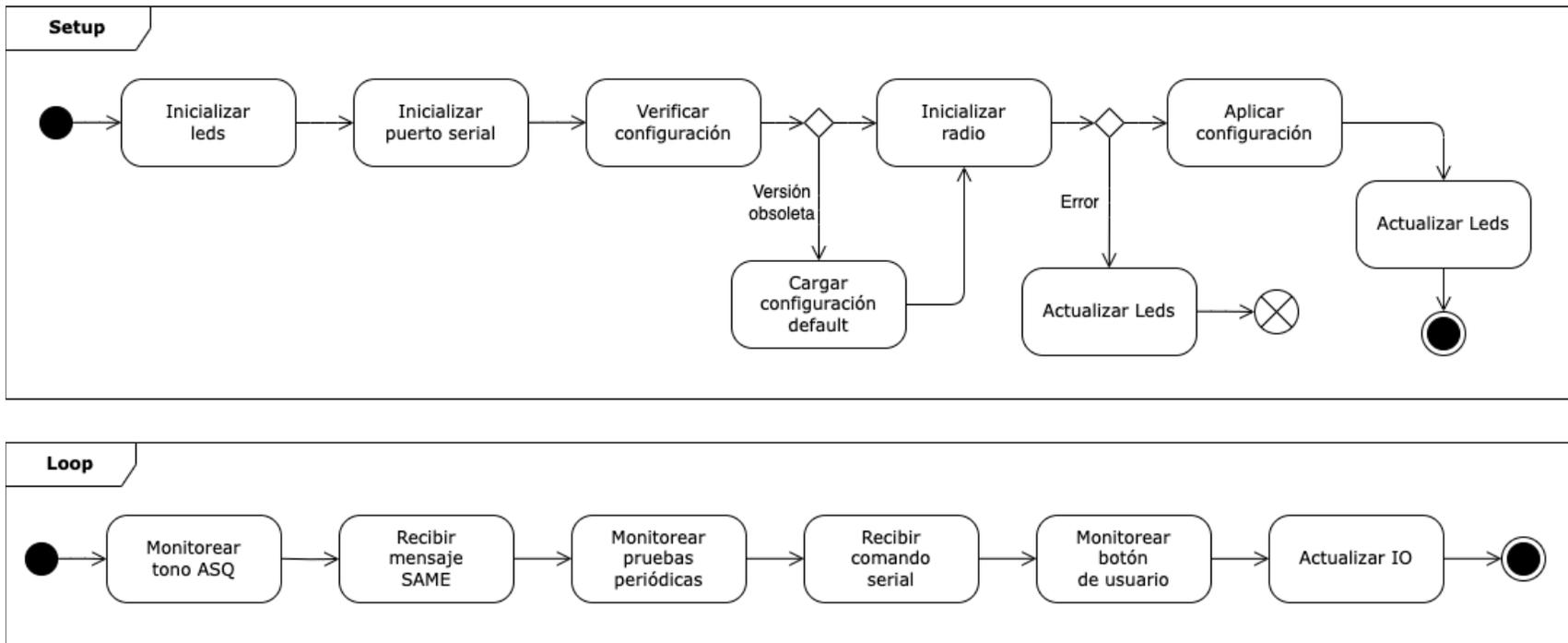


Figura C.1.3 Diagrama de máquina de estados del programa principal.

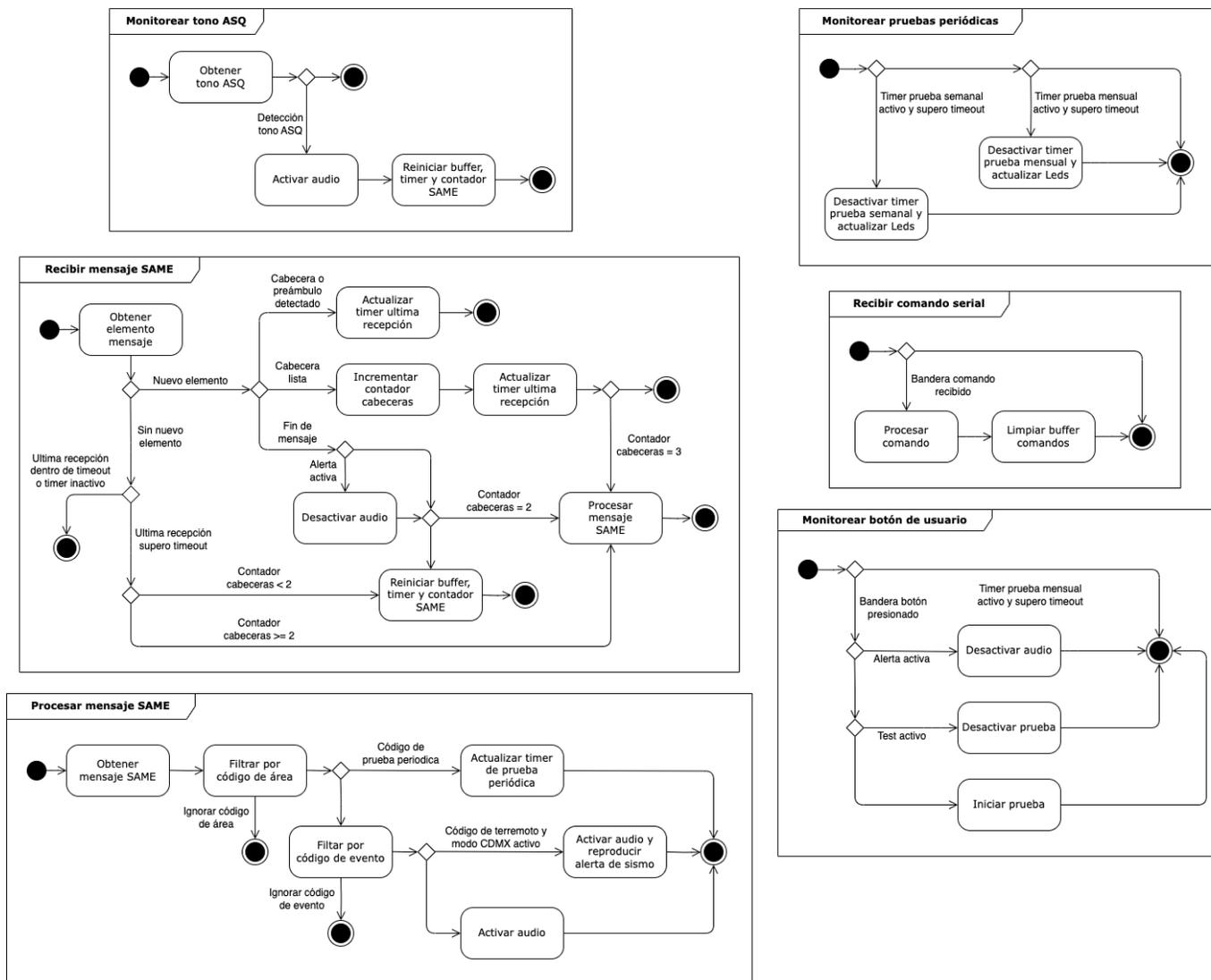


Figura C.1.3 Diagrama de máquina de estados de bloques del bucle infinito.

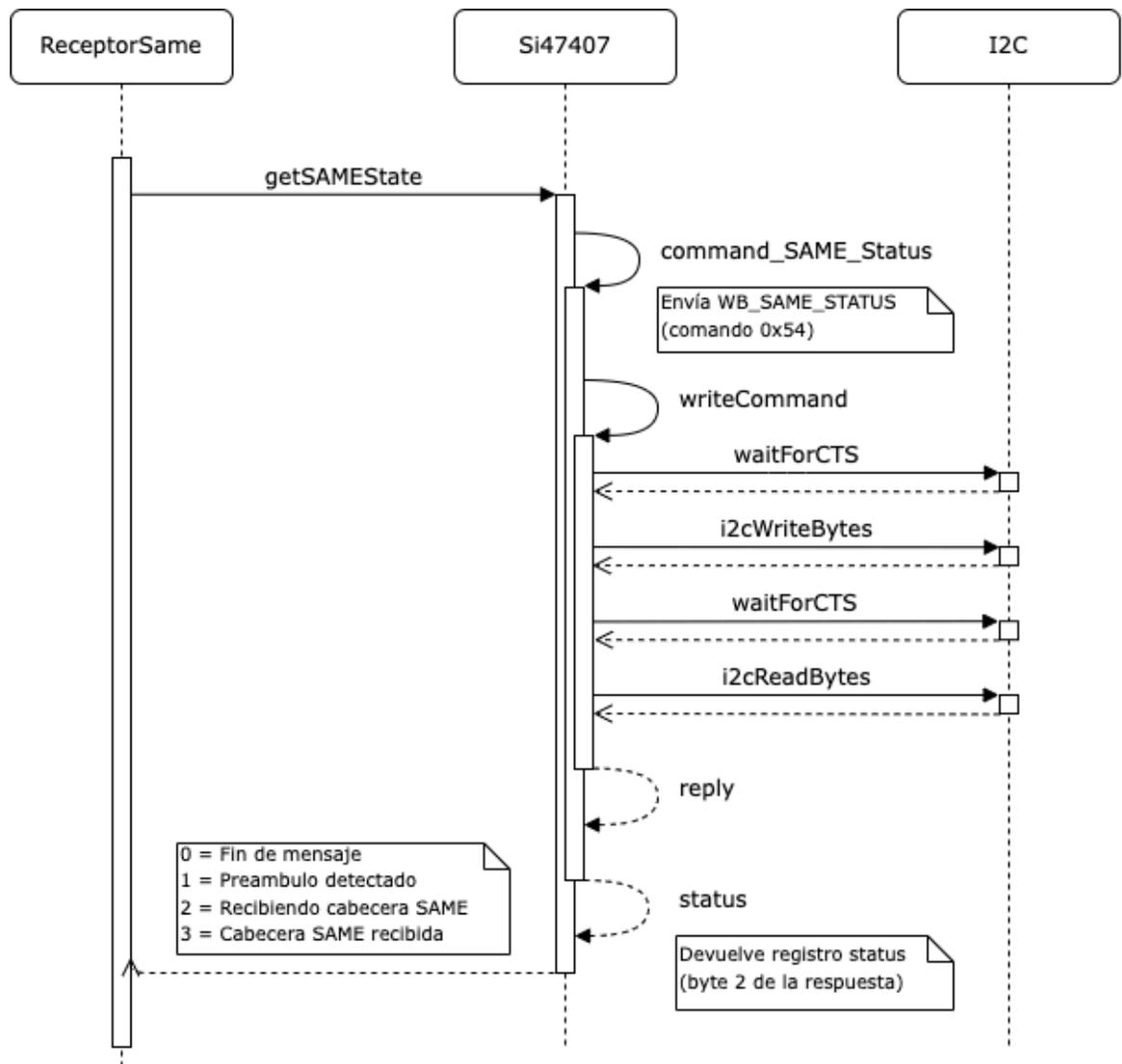


Figura C.1.3 Diagrama de secuencia de obtención de elementos de un mensaje SAME.

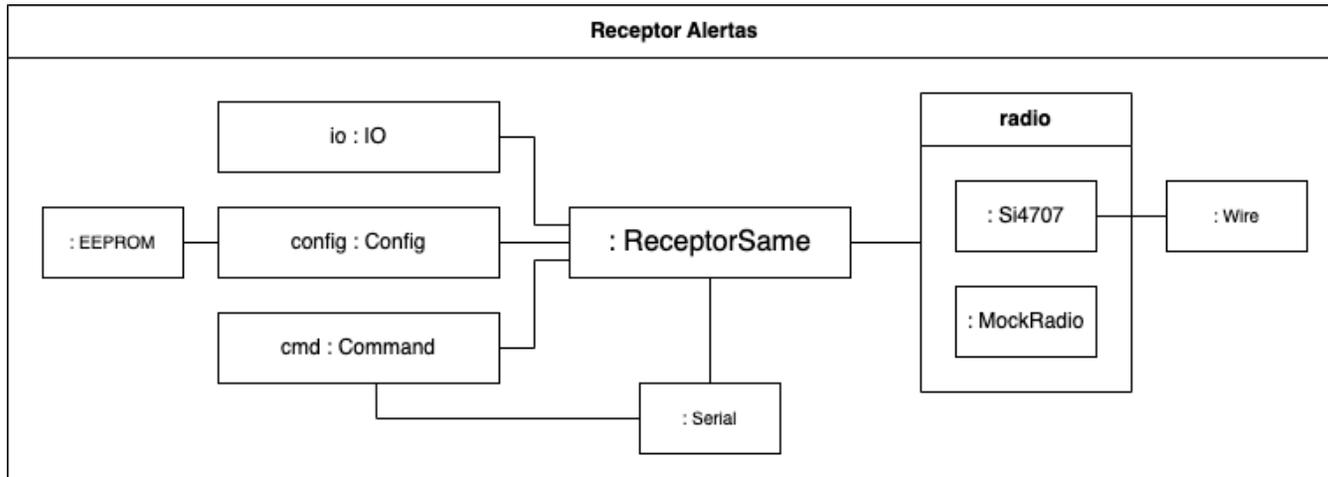


Figura C.1.3 Diagrama de estructura compuesta del software del receptor de alertas.

C.2 Protocolo de comunicación

La configuración del puerto serie del receptor de alertas es de 9600 bits por segundo, 8 bits de datos, sin paridad y un bit de parada.

En la Tabla C.2.1 se enumeran los comandos disponibles para controlar el receptor de alertas y ajustar su configuración.

En la Tabla C.2.2 se presenta la lista de reportes que puede generar el receptor de alertas durante su operación .

Todos los mensajes son terminados por un carácter de retorno de carro (código ASCII 13 ó \r) seguido de un carácter de nueva línea (ASCII 10 ó \n).

Tabla C.2.1 Comandos del receptor de alertas.

Mensaje	Descripción
AREA_ADD, [código_área]	Añade el código a la lista de códigos de área para alertar. El parámetro <i>código_área</i> debe ser una cadena de 6 caracteres numéricos.
AREA_DEL, [índice]	Elimina el código de la lista de códigos de área para alertar. El parámetro <i>índice</i> debe ser una cadena de caracteres numéricos entre 0 y 19.
CDMX, [modo_cdmx]	Configura el modo de alerta sísmica para Ciudad de México. El parámetro <i>modo_cdmx</i> debe ser un carácter 0 ó 1.
CHANNEL, [canal]	Configura el canal a sintonizar. El parámetro <i>canal</i> debe ser un carácter numérico entre 1 y 7.
DEFAULTS	Aplica la configuración por defecto.

Mensaje	Descripción
DUMP	<p>Solicita la configuración del receptor de alertas. Devuelve los reportes:</p> <p>CONFIG_VERSION CONFIG_CHANNEL CONFIG_MUTE CONFIG_VOLUME CONFIG_RELAY CONFIG_CDMX CONFIG_RWT_PERIOD CONFIG_RMT_PERIOD CONFIG_AREA_CODES CONFIG_AREA_CODE CONFIG_EVENT_CODES CONFIG_EVENT_CODE</p>
EVENT_ADD,[código_evento]	<p>Añade el código a la lista de códigos de evento para alertar. El parámetro <i>código_evento</i> debe ser una cadena de 3 caracteres.</p>
EVENT_DEL,[índice]	<p>Elimina el código de la lista de códigos de evento para alertar. El parámetro <i>índice</i> debe ser una cadena de caracteres numéricos entre 0 y 19.</p>
FREQUENCY	<p>Solicita la frecuencia del canal sintonizado.</p>
MUTE,[silencio]	<p>Configura el silencio del audio. El parámetro <i>silencio</i> debe ser un carácter 0 ó 1.</p>
QUALITY	<p>Solicita la calidad de la señal recibida.</p>
RELAY,[modo_relevador]	<p>Configura el modo de operación del relevador. El parámetro <i>modo_relevador</i> debe ser un carácter numérico entre 0 y 2 que significa:</p> <p>0: Relevador desactivado 1: Relevador automático (activado al dispararse una alerta y desactivado al final de un mensaje SAME). 2: Relevador activado automáticamente al activarse una alerta y desactivado manualmente por medio del botón de usuario.</p>
RELOAD	<p>Carga la configuración de la memoria no volátil.</p>
RMT_PERIOD,[periodo_rmt]	<p>Configura el tiempo de validez en minutos para la prueba periódica mensual. El parámetro <i>periodo_rmt</i> debe ser una cadena de caracteres numéricos entre 0 y 65,535.</p>

Mensaje	Descripción
RWT_PERIOD, [periodo_rwt]	Configura el tiempo de validez en minutos para la prueba periódica semanal. El parámetro <i>periodo_rwt</i> debe ser una cadena de caracteres numéricos entre 0 y 65,535.
SAVE	Guarda la configuración en la memoria no volátil.
VOLUME, [volumen]	Configura el volumen del audio. El parámetro <i>volumen</i> debe ser una cadena de caracteres numéricos entre 0 y 10.

Tabla C.2.2 Reportes del receptor de alertas.

Mensaje	Descripción
AREA_ADD_ERROR	Error al agregar código a la lista de códigos de área para alertar. Puede ser que el código de área ya exista en la lista.
AREA_ADD_OK, [código_área]	Código añadido a la lista de códigos de área para alertar.
AREA_DEL_ERROR	Error al eliminar código de la lista de códigos de área para alertar.
AREA_DEL_OK, [código_área]	Código eliminado de la lista de códigos de área para alertar.
ASQ_OFF	Tono de alarma ASQ desactivado.
ASQ_ON	Tono de alarma ASQ activado.
AUDIO_OFF	Audio desactivado.
AUDIO_ON	Audio activado.
BUTTON_TRIGGERED	Botón de usuario presionado.
CDMX, [modo_cdmx]	Modo de alerta sísmica para Ciudad de México.
CDMX_ALERT_OFF	Sonido oficial de la alerta sísmica de la Ciudad de México desactivado.

Mensaje	Descripción
CDMX_ALERT_ON	Sonido oficial de la alerta sísmica de la Ciudad de México activado.
CDMX_ERROR	Error al configurar modo CDMX.
CHANNEL_ERROR	Error al configurar canal.
CONFIG_AREA_CODE, [índice], [código_área]	Lista de códigos de área para alertar.
CONFIG_AREA_CODES, [conteo_códigos_area]	Conteo de códigos en la lista de códigos de área para alertar.
CONFIG_CDMX, [modo_cdmx]	Configuración del modo de alerta sísmica para Ciudad de México.
CONFIG_CHANNEL, [canal]	Canal configurado.
CONFIG_DEFAULTS	Configuración por defecto establecida.
CONFIG_EVENT_CODE, [índice], [codigo_evento]	Lista de códigos de evento para alertar.
CONFIG_EVENT_CODES, [conteo_codigos_evento]	Conteo de códigos en la lista de códigos de evento para alertar.
CONFIG_MUTE, [silencio]	Configuración de silencio.
CONFIG_RELAY, [modo_relevador]	Configuración de modo de relevador.
CONFIG_RMT_PERIOD, [periodo_rmt]	Configuración del tiempo de validez en minutos para prueba periódica mensual.
CONFIG_RWT_PERIOD, [periodo_rwt]	Configuración del tiempo de validez en minutos para prueba periódica semanal.
CONFIG_VERSION, [version]	Versión de configuración. Si la versión guardada en la EEPROM es diferente a la esperada por el firmware se establecerá la configuración por defecto.
CONFIG_VERSION, [version]	Versión de la configuración.
CONFIG_VOLUME, [volumen]	Configuración de nivel de volumen.

Mensaje	Descripción
EVENT_ADD_ERROR	Error al agregar código a la lista de códigos de evento para alertar. Puede ser que el código de evento ya exista en la lista.
EVENT_ADD_OK,[codigo_evento]	Código añadido a la lista de códigos de evento para alertar.
EVENT_DEL_ERROR	Error al eliminar código de la lista de códigos de evento para alertar.
EVENT_DEL_OK,[codigo_evento]	Código eliminado de la lista de códigos de evento para alertar.
FREQUENCY,[frecuencia]	Frecuencia del canal sintonizado.
MEM_RELOAD	Configuración recargada de memoria no volátil.
MEM_SAVE	Configuración almacenada en la memoria no volátil.
MUTE,[silencio]	Indicador de audio silenciado.
MUTE_ERROR	Error al configurar silencio.
QUALITY,[rssi],[snr]	Indicador de fuerza de la señal recibida (RSSI) y relación señal-ruido (SNR) del canal sintonizado.
RELAY,[modo_relevador]	Modo de relevador.
RELAY_ERROR	Error al configurar modo de relevador.
RELAY_OFF	Relevador desactivado.
RELAY_ON	Relevador activado.
RMT_PERIOD,[periodo_rmt]	Tiempo de validez de la prueba periódica mensual en minutos.
RMT_PERIOD_ERROR	Error al configurar tiempo de validez para la prueba periódica mensual.
RUN	Inicio del bucle infinito del receptor de alertas.
RWT_PERIOD,[periodo_rwt]	Tiempo de validez de la prueba periódica semanal en minutos.
RWT_PERIOD_ERROR	Error al configurar tiempo de validez para la prueba periódica semanal.

Mensaje	Descripción
SAME_ALERT_OFF	Alerta SAME desactivada.
SAME_ALERT_OFF	Alerta SAME desactivada.
SAME_ALERT_ON	Alerta SAME activada.
SAME_EMPTY	Mensaje SAME vacío.
SAME_EOM_DET	Fin de mensaje SAME detectado.
SAME_HDR_DET	Encabezado SAME detectado.
SAME_HDR_RDY, [conteo_encabezados]	Encabezado SAME disponible, incluye el conteo de encabezados recibidos.
SAME_IGNORE_AREA	Mensaje SAME ignorado por código de área.
SAME_IGNORE_EVENT	Mensaje SAME ignorado por código de evento.
SAME_MESSAGE, [mensaje]	Mensaje SAME recibido.
SAME_PRE_DET	Preambulo SAME detectado.
SAME_RMT	Prueba periódica mensual recibida.
SAME_RMT_TIMEOUT	Tiempo de espera agotado para recibir la prueba periódica mensual.
SAME_RWT	Prueba periódica semanal recibida.
SAME_RWT_TIMEOUT	Tiempo de espera agotado para recibir la prueba periódica semanal.
SAME_TIMEOUT	Tiempo de espera agotado para recibir mensaje SAME.
SI4707_ERROR	Falla al inicializar circuito integrado Si4707.
SI4707_OK	Inicialización de circuito integrado Si4707 exitosa.
STARTUP	Arranque del receptor de alertas.
TEST_ALERT_OFF	Prueba de alerta desactivada.
TEST_ALERT_ON	Prueba de alerta activada.
TUNE_ERROR, [cana1]	Error al sintonizar el canal.

Mensaje	Descripción
TUNE_OK, [canal]	Canal sintonizado con éxito.
UNKNOWN_COMMAND	Comando desconocido.
VOLUME, [volumen]	Nivel de volumen.
VOLUME_ERROR	Error al configurar volumen.

Apéndice D. Integración de software y hardware

A continuación, se documenta el proceso de integración llevado a cabo en la sección 3.5.

Primero, en el apartado D.1 se presenta el proceso para instalar el bootloader en el hardware fabricado seguido en el apartado D.2 del proceso para la instalación de software.

Por último, en el apartado D.3 se transcriben los resultados obtenidos de las pruebas unitarias y de integración por medio del puerto serial del hardware.

D.1 Instalación del bootloader

Como se menciona al inicio de la sección 3.4, hemos elegido utilizar el software AVRDUDE 6.3 para la transferencia de programas. Sin embargo, en lugar de utilizarlo directamente, nos apoyaremos en el entorno Arduino para llevar a cabo las transferencias.

Además, para la instalación específica del bootloader, requerimos un programador ICSP y emplearemos el AVR Pocket Programmer para esta tarea.

Primero, es crucial verificar que las opciones adecuadas estén configuradas en el ambiente Arduino para nuestro hardware. Bajo el menú *Tools* se debe verificar se encuentra seleccionado:

- Board: "Arduino Pro or Pro Mini"
- Processor: "ATmega328P (3.3V, 8 MHz)"
- Programmer: "USBtinyISP"

Estas opciones están señaladas con una flecha verde en la Figura D.1.1.

A continuación, conectar el programador al puerto de programación ICSP (conector J1) del receptor de alertas y para instalar el bootloader seleccionar la opción *Burn Bootloader* del menú *Tools*. Esta opción está marcada con una flecha de color rojo en la Figura D.1.1.

Una vez completado el proceso de instalación del bootloader se muestra en la barra de estado del ambiente Arduino el mensaje *Done burning bootloader*. En la Figura D.1.2, este mensaje se destaca con un recuadro rojo.

Por último, en la Figura D.1.2, en la parte inferior del ambiente Arduino, se muestra el resultado de la ejecución del software AVRDUDE para instalar el bootloader.

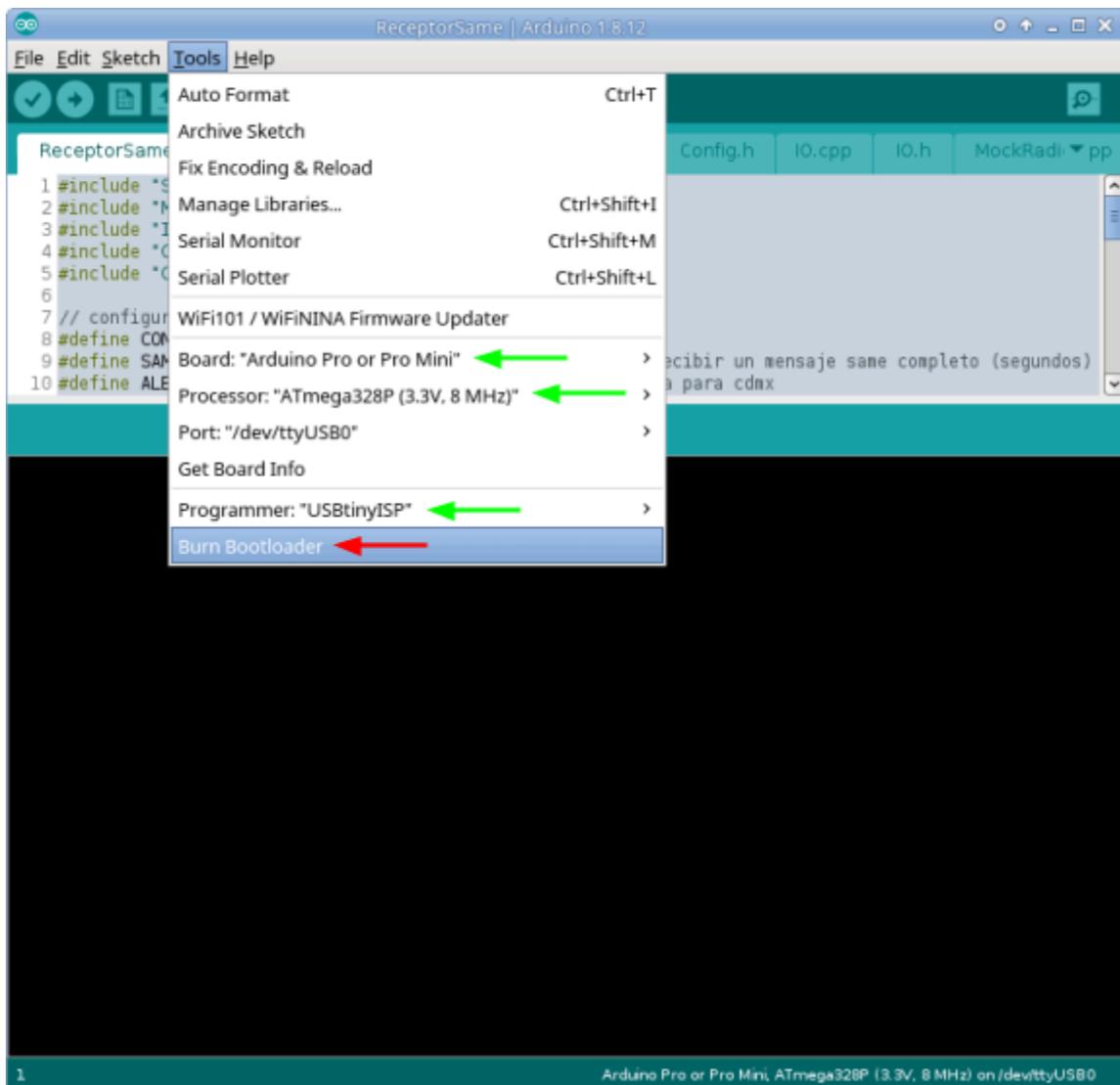


Figura D.1.1. Preparativos e instalación del bootloader.

```
ReceptorSame | Arduino 1.8.12
File Edit Sketch Tools Help
ReceptorSame Command.cpp Command.h Config.cpp Config.h IO.cpp IO.h MockRadio.cpp
1 #include "Si4707.h"
2 #include "MockRadio.h"
3 #include "IO.h"
4 #include "Command.h"
5 #include "Config.h"
6
7 // configuración
8 #define CONFIG_VERSION 0x12 // versión memoria
9 #define SAME_TIMEOUT 6000 // tiempo de espera máximo para recibir un mensaje sane completo (segundos)
10 #define ALERT_CDMX_MILLIS 60000 // tiempo de duración alerta para cdmx

Done burning bootloader.
avrdude: input file /home/manuel/.arduino15/packages/arduino/hardware/avr/1.8.2/bootloaders/atmega/ATmegaBOOT_v1.5_3.1a.hex
avrdude: reading on-chip flash data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 32652 bytes of flash verified
avrdude: reading input file "0x0F"
avrdude: writing lock (1 bytes):

Writing | ##### | 100% 0.01s

avrdude: 1 bytes of lock written
avrdude: verifying lock memory against 0x0F:
avrdude: load data lock data from input file 0x0F:
avrdude: input file 0x0F contains 1 bytes
avrdude: reading on-chip lock data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of lock verified

avrdude done. Thank you.

Arduino Pro or Pro Mini on /dev/ttyUSB0
```

Figura D.1.2. Resultado de la instalación del bootloader.

D.2 Instalación de software

Para instalar un programa en el receptor de alertas, ya sea el software de aplicación, las pruebas unitarias o de integración, se lleva a cabo el mismo procedimiento utilizando el ambiente Arduino.

Primero, se debe conectar el receptor de alertas al sistema anfitrión por medio de su puerto USART con un convertidor USART-USB y abrir el *Sketch* correspondiente al software que queremos cargar. En la Figura D.2.1 se encuentra cargado el software de aplicación *ReceptorSame*.

Para compilar e instalar el software elegido, se debe dar click en el botón Upload, en la Figura D.2.1 destacado con un recuadro rojo.

Al terminar el proceso de compilación e instalación del software se muestra en la barra de estado del ambiente Arduino el mensaje *Done uploading*. En la Figura D.2.2, este mensaje se destaca con un recuadro rojo.

Por último, en la Figura D.2.2, en la parte inferior del ambiente Arduino, se muestra el resultado de la ejecución del software AVRDUDE para instalar el software de aplicación.

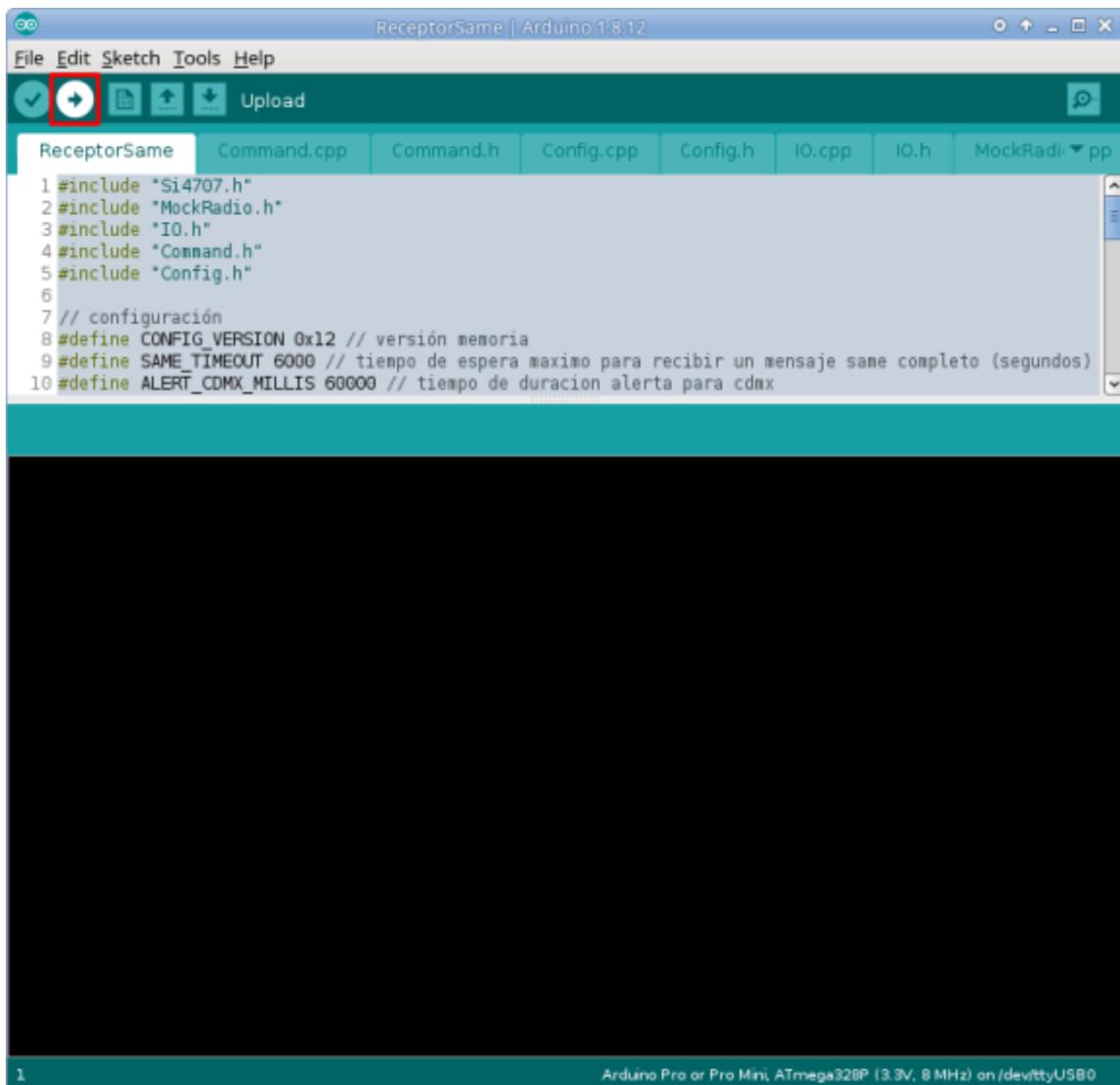


Figura D.2.1. Preparativos para la instalación del software de aplicación.

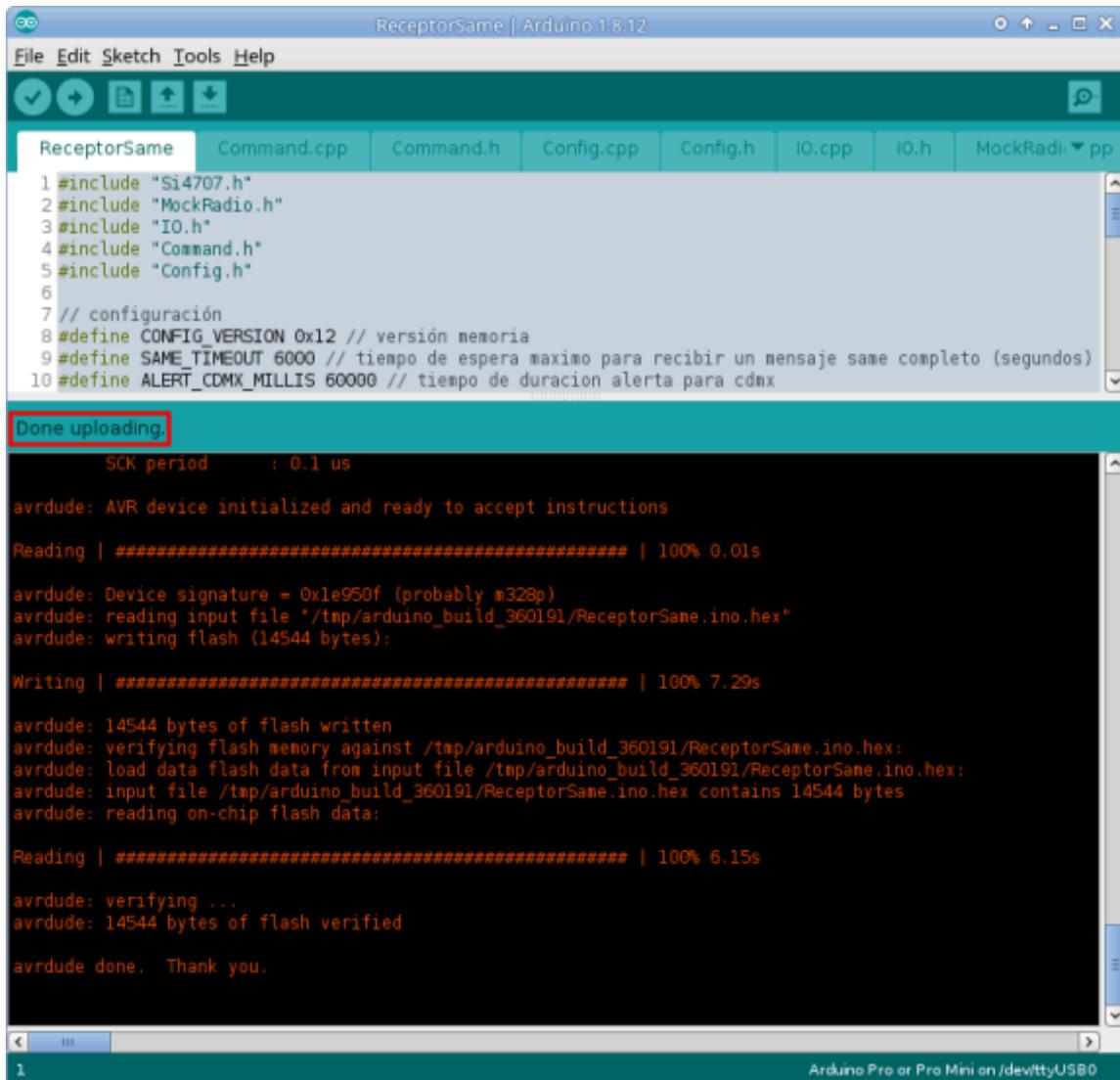


Figura D.2.2. Resultado de la instalación del software de aplicación.

D.3 Pruebas unitarias

En este apartado, se presentan los resultados de ejecutar las pruebas unitarias en el hardware fabricado. Las pruebas se instalan en el receptor de alertas de manera similar al software de aplicación como se describe en el apartado D.2.

Dependiendo del diseño de la prueba, esta puede ser de tipo automática o manual y se comprueban principalmente con el resultado obtenido por medio del puerto serial del receptor de alertas.

Se pueden encontrar más detalles sobre la implementación de las pruebas unitarias en la Sección 3.4 y específicamente en la Tabla 13.

TestCommand

El objetivo de esta prueba unitaria es verificar manualmente la recepción e interpretación de comandos. En la Tabla D.3.1 se presentan cuatro ejemplos para esta prueba.

Tabla D.3.1. Resultados de la prueba unitaria TestCommand.

Prueba	Resultado
TEST	isReady getType: TEST (4) getCmd: 0 ---
TEST,ARG	isReady getType: TEST (4) getCmd: 0 isArg getArg: ARG (3) getArgUInt: 0x0 getArgByte: 0x0 ---
TEST,171	isReady getType: TEST (4) getCmd: 0 isArg getArg: 171 (3) getArgUInt: 0xAB getArgByte: 0xAB ---
TEST,57584	isReady getType: TEST (4) getCmd: 0 isArg getArg: 57584 (5) getArgUInt: 0xE0F0 getArgByte: 0xF0 ---

TestConfig

Esta prueba unitaria verifica la obtención y establecimiento de la configuración del receptor de alertas. En la Tabla D.3.2 se exponen los resultados de esta prueba.

Tabla D.3.2. Resultados de la prueba unitaria TestConfig.

Prueba	Resultado
Inicio	SETUP
Tamaño y versión de configuración	MEM_BYTES, 195 MEM_VERSION, 18
Carga de la memoria no volátil y volcado de configuración.	MEM_RELOAD DUMP_START MEM_VERSION, 18 MEM_CHANNEL, 7 MEM_MUTE, 1 MEM_VOLUME, 10 MEM_RELAY, 1 MEM_CDMX, 1 MEM_RWT_PERIOD, 0 MEM_RMT_PERIOD, 0 DUMP_AREA_CODES, 0 DUMP_EVENT_CODES, 0 DUMP_END
Altas, bajas, listado y búsqueda de códigos de área.	TEST_AREA_CODES_START SET_AREA, 555555 AREA_SET SET_AREA, 333333 AREA_SET SET_AREA, 222222 AREA_SET SET_AREA, 444444 AREA_SET SET_AREA, 000000 AREA_SET SET_AREA, 567890 AREA_SET SET_AREA, 333333 AREA_NOT_SET DUMP_AREA_CODES, 6 DUMP_AREA_CODE, 0, 000000 DUMP_AREA_CODE, 1, 222222 DUMP_AREA_CODE, 2, 333333

Prueba	Resultado
	DUMP_AREA_CODE,3,444444 DUMP_AREA_CODE,4,555555 DUMP_AREA_CODE,5,567890 FIND_AREA,555555 AREA_FOUND FIND_AREA, AREA_NOT_FOUND FIND_AREA,111111 AREA_NOT_FOUND FIND_AREA,033333 AREA_FOUND FIND_AREA,900000 AREA_FOUND FIND_AREA,088888 AREA_NOT_FOUND CLEAR_AREA,444444 AREA_CLEAR CLEAR_AREA,0 AREA_NOT_CLEAR CLEAR_AREA,666666 AREA_NOT_CLEAR DUMP_AREA_CODES,5 DUMP_AREA_CODE,0,000000 DUMP_AREA_CODE,1,222222 DUMP_AREA_CODE,2,333333 DUMP_AREA_CODE,3,555444 DUMP_AREA_CODE,4,567555 TEST_AREA_CODES_END
Altas, bajas, listado y búsqueda de códigos de eventos.	TEST_EVENT_CODES_START SET_EVENT,YZ- EVENT_SET SET_EVENT,MNO EVENT_SET SET_EVENT,STU EVENT_SET SET_EVENT,PQR EVENT_SET SET_EVENT,VWX EVENT_SET SET_EVENT,JKL EVENT_SET SET_EVENT,GHI EVENT_SET SET_EVENT,DEF

Prueba	Resultado
	EVENT_SET SET_EVENT,ABC EVENT_SET SET_EVENT,123 EVENT_SET SET_EVENT,789 EVENT_SET SET_EVENT,456 EVENT_SET SET_EVENT,VWX EVENT_NOT_SET DUMP_EVENT_CODES,12 DUMP_EVENT_CODE,0,123 DUMP_EVENT_CODE,1,456 DUMP_EVENT_CODE,2,789 DUMP_EVENT_CODE,3,ABC DUMP_EVENT_CODE,4,DEF DUMP_EVENT_CODE,5,GHI DUMP_EVENT_CODE,6,JKL DUMP_EVENT_CODE,7,MNO DUMP_EVENT_CODE,8,PQR DUMP_EVENT_CODE,9,STU DUMP_EVENT_CODE,10,VWX DUMP_EVENT_CODE,11,YZ- FIND_EVENT,JKL EVENT_FOUND FIND_EVENT, EVENT_NOT_FOUND FIND_EVENT,ASD EVENT_NOT_FOUND CLEAR_EVENT,ABC EVENT_CLEAR CLEAR_EVENT, EVENT_NOT_CLEAR CLEAR_EVENT,QWE EVENT_NOT_CLEAR DUMP_EVENT_CODES,11 DUMP_EVENT_CODE,0,123 DUMP_EVENT_CODE,1,456 DUMP_EVENT_CODE,2,789 DUMP_EVENT_CODE,3,DEF DUMP_EVENT_CODE,4,GHI DUMP_EVENT_CODE,5,JKL DUMP_EVENT_CODE,6,MNO DUMP_EVENT_CODE,7,PQR

Prueba	Resultado
	DUMP_EVENT_CODE,8,STU DUMP_EVENT_CODE,9,VWX DUMP_EVENT_CODE,10,YZ- TEST_EVENT_CODES_END
Ajustes en los parámetros de volumen.	TEST_VOLUME_START TEST_VOLUME,11,false DUMP_VOLUME,10,0 TEST_VOLUME,11,true DUMP_VOLUME,10,1 TEST_VOLUME,44,true DUMP_VOLUME,10,1 TEST_VOLUME,22,false DUMP_VOLUME,10,0 TEST_VOLUME,22,true DUMP_VOLUME,10,1 TEST_VOLUME_END

TestIO

El propósito de esta prueba unitaria es verificar el funcionamiento de los puertos de entrada y salida del receptor de alertas. Los resultados de esta prueba se presentan en la Tabla D.3.3, mientras que en la Figura 18 se muestra el circuito externo utilizado para validar estos resultados.

Tabla D.3.3. Resultados de la prueba unitaria TestIO.

Prueba	Resultado
Inicio	SETUP
Enciende el led 1, apaga el led 2 y activa el relevador.	LED_1_ON, LED_2_OFF, RELAY_ON
Apagar el led 1, enciende el led 2 y desactiva el relevador.	LED_1_OFF, LED_2_ON, RELAY_OFF
Titila lento el led 1, titila rápido el led 2 y desactiva el relevador.	LED_1_SLOW, LED_2_FAST, RELAY_OFF
Titila rápido el led 1, titila lento el led 2 y desactiva el relevador.	LED_1_FAST, LED_2_SLOW, RELAY_OFF
Activa en los leds el patrón de inicio, error, espera, espera con prueba válida y alerta.	LEDS_BEGIN LEDS_ERROR LEDS_WAIT LEDS_WAIT_RT LEDS_ALERT
Activa y desactiva el reproductor de audio externo.	AUDIO_PLAY_ON AUDIO_PLAY_OFF
Reporta si el botón externo es presionado	BUTTON_TRIGGERED

D.4 Pruebas de integración

Finalmente, se presentan los resultados obtenidos al llevar a cabo las pruebas de integración, las cuales evalúan el desempeño del receptor de alertas.

Los resultados presentados más adelante corresponden a capturas del puerto serial del receptor de alertas ejecutando su software de aplicación con un controlador de dispositivo que simula el comportamiento del circuito integrado Si4707 frente a diferentes escenarios y mensajes.

En la Sección 3.4 y específicamente en la Tabla 14 se pueden encontrar más detalles sobre la implementación de las pruebas de integración.

Falla circuito integrado receptor

```
STARTUP
CONFIG_VERSION,18
SI4707_ERROR
```

Mensaje de prueba periódica semanal

```
STARTUP
CONFIG_VERSION,18
SI4707_OK
MUTE,1
VOLUME,10
TUNE_OK,7
RUN
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,1
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,2
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,3
SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-.....
SAME_RWT
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
```

SAME_PRE_DET
SAME_EOM_DET

Mensaje de alerta con voz sin tono ASQ

STARTUP
CONFIG_VERSION,18
SI4707_OK
MUTE,1
VOLUME,10
TUNE_OK,7
RUN
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,1
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,2
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,3
SAME_MESSAGE, -WXR-SPS-039173-039051-139069+0030-1591829-KCLE/NWS-
SAME_ALERT_ON
AUDIO_ON
RELAY_ON
SAME_EOM_DET
AUDIO_OFF
RELAY_OFF
SAME_ALERT_OFF
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET

Mensaje de alerta con tono ASQ

STARTUP
CONFIG_VERSION,18
SI4707_OK
MUTE,1
VOLUME,10
TUNE_OK,7
RUN
SAME_PRE_DET
SAME_HDR_DET

SAME_HDR_RDY,1
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,2
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,3
SAME_MESSAGE, -WXR-TOR-039173-039051-139069+0030-1591829-KCLE/NWS-
SAME_ALERT_ON
AUDIO_ON
RELAY_ON
ASQ_ON
ASQ_OFF
SAME_EOM_DET
AUDIO_OFF
RELAY_OFF
SAME_ALERT_OFF
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET

Mensaje con solo dos encabezados

STARTUP
CONFIG_VERSION,18
SI4707_OK
MUTE,1
VOLUME,10
TUNE_OK,7
RUN
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,1
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,2
SAME_PRE_DET
SAME_EOM_DET
SAME_MESSAGE, -CIV-RWT-000000+0300-832300-XDIF/005-.....
SAME_RWT
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET

Mensaje de alerta sísmica

STARTUP
CONFIG_VERSION,18
SI4707_OK
MUTE,1
VOLUME,10
TUNE_OK,7
RUN
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,1
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,2
SAME_PRE_DET
SAME_HDR_DET
SAME_HDR_RDY,3
SAME_MESSAGE, -CIV-EQW-000000+0005-832326-XDIF/005-.....
CDMX_ALERT_ON
RELAY_ON
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
SAME_PRE_DET
SAME_EOM_DET
RELAY_OFF
CDMX_ALERT_OFF

Apéndice E. Pruebas de producto y lanzamiento

En seguida, se documenta la configuración necesaria en la computadora encargada de recolectar los registros de operación para las pruebas dinámicas de caja blanca. Los registros obtenidos con este dispositivo son fundamentales para este trabajo, ya que forman la base de los resultados analizados en el Capítulo 4.

Como se mencionó en la sección 3.6.4, se ha seleccionado una *Raspberry Pi* para recolectar la información, específicamente el modelo B versión 2.0, con el sistema operativo Raspbian GNU/Linux 10.

En cuanto al hardware, la computadora requiere un convertidor USART-USB para recibir la información generada por el receptor de alertas. Para el software, además de la configuración básica de red y zona horaria del sistema operativo, se requiere el intérprete del lenguaje de programación *Python* versión 3.

El código fuente del programa y scripts necesarios para recolectar los registros de operación del receptor de alertas se encuentran en la carpeta *logger* del repositorio de Git de este trabajo que está disponible en <https://gitlab.com/manuel-rabade/receptor-alertas-same> y a continuación se describe el propósito y funcionamiento de cada archivo.

El programa *log.py* debe ubicarse en la carpeta */home/pi* y es el encargado de recibir los datos a través del convertidor USART-USB, añadir la fecha y hora de recepción al inicio de cada registro y guardarlos en un archivo de texto nombrado según la fecha de arribo.

En el archivo *rc.local*, se muestra la configuración necesaria para ejecutar el archivo *log.py* en segundo plano al inicio del sistema operativo, utilizando el programa *screen*. Esta configuración se debe ubicar en el archivo */etc/rc.local*. El programa *screen* permite la ejecución de procesos en segundo plano y su recuperación al primer plano desde cualquier sesión con los permisos adecuados.

El script *screen.sh* ilustra el uso del programa *screen* para llevar al proceso *log.py* al primer plano y consultar su estado.

Por último, se presentan los registros de operación recopilados utilizando esta configuración durante un día representativo del periodo de prueba. Estos registros están extraídos del archivo *2021-09-26.log*, correspondiente al 26 de septiembre de 2021.

2021-09-26,02:45:01,1632642301.4060724,SAME_PRE_DET
2021-09-26,02:45:01,1632642301.5770922,SAME_HDR_DET
2021-09-26,02:45:02,1632642302.2774923,"SAME_HDR_RDY,1"
2021-09-26,02:45:04,1632642304.4078286,SAME_PRE_DET
2021-09-26,02:45:04,1632642304.5817761,SAME_HDR_DET
2021-09-26,02:45:05,1632642305.2822034,"SAME_HDR_RDY,2"
2021-09-26,02:45:07,1632642307.4228241,SAME_PRE_DET
2021-09-26,02:45:07,1632642307.5351403,SAME_HDR_DET
2021-09-26,02:45:08,1632642308.226539,"SAME_HDR_RDY,3"
2021-09-26,02:45:08,1632642308.318507,
"SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-....."
2021-09-26,02:45:08,1632642308.3259108,SAME_RWT
2021-09-26,02:45:08,1632642308.3378992,SAME_EOM_DET
2021-09-26,02:45:10,1632642310.127368,SAME_PRE_DET
2021-09-26,02:45:10,1632642310.363132,SAME_EOM_DET
2021-09-26,02:45:11,1632642311.9921796,SAME_PRE_DET
2021-09-26,02:45:12,1632642312.219233,SAME_EOM_DET
2021-09-26,02:45:13,1632642313.8478174,SAME_PRE_DET
2021-09-26,02:45:14,1632642314.0742328,SAME_EOM_DET
2021-09-26,05:45:01,1632653101.4414108,SAME_PRE_DET
2021-09-26,05:45:01,1632653101.5555806,SAME_HDR_DET
2021-09-26,05:45:02,1632653102.244983,"SAME_HDR_RDY,1"
2021-09-26,05:45:04,1632653104.3842702,SAME_PRE_DET
2021-09-26,05:45:04,1632653104.5603952,SAME_HDR_DET
2021-09-26,05:45:05,1632653105.2609332,"SAME_HDR_RDY,2"
2021-09-26,05:45:07,1632653107.4001791,SAME_PRE_DET
2021-09-26,05:45:07,1632653107.5140662,SAME_HDR_DET
2021-09-26,05:45:08,1632653108.2034304,"SAME_HDR_RDY,3"
2021-09-26,05:45:08,1632653108.2952394,
"SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-....."
2021-09-26,05:45:08,1632653108.3023841,SAME_RWT
2021-09-26,05:45:08,1632653108.3152556,SAME_EOM_DET
2021-09-26,05:45:10,1632653110.1037781,SAME_PRE_DET
2021-09-26,05:45:10,1632653110.3418806,SAME_EOM_DET
2021-09-26,05:45:11,1632653111.9695861,SAME_PRE_DET
2021-09-26,05:45:12,1632653112.1967552,SAME_EOM_DET
2021-09-26,05:45:13,1632653113.8243067,SAME_PRE_DET
2021-09-26,05:45:14,1632653114.0517857,SAME_EOM_DET
2021-09-26,08:45:01,1632663901.8976374,SAME_PRE_DET
2021-09-26,08:45:02,1632663902.0730898,SAME_HDR_DET
2021-09-26,08:45:02,1632663902.7116086,"SAME_HDR_RDY,1"
2021-09-26,08:45:04,1632663904.8512232,SAME_PRE_DET
2021-09-26,08:45:05,1632663905.0267897,SAME_HDR_DET
2021-09-26,08:45:05,1632663905.7168093,"SAME_HDR_RDY,2"
2021-09-26,08:45:07,1632663907.856263,SAME_PRE_DET
2021-09-26,08:45:08,1632663908.030606,SAME_HDR_DET
2021-09-26,08:45:08,1632663908.670337,"SAME_HDR_RDY,3"

2021-09-26,08:45:08,1632663908.7513254,
"SAME_MESSAGE, -CIV-RWT-000000+0300-832300-XDIF/005-....."
2021-09-26,08:45:08,1632663908.766033, SAME_RWT
2021-09-26,08:45:08,1632663908.7813632, SAME_EOM_DET
2021-09-26,08:45:10,1632663910.5708463, SAME_PRE_DET
2021-09-26,08:45:10,1632663910.7980273, SAME_EOM_DET
2021-09-26,08:45:12,1632663912.4257965, SAME_PRE_DET
2021-09-26,08:45:12,1632663912.6616561, SAME_EOM_DET
2021-09-26,08:45:14,1632663914.2803056, SAME_PRE_DET
2021-09-26,08:45:14,1632663914.5169044, SAME_EOM_DET
2021-09-26,11:45:01,1632674701.5388327, SAME_PRE_DET
2021-09-26,11:45:01,1632674701.652227, SAME_HDR_DET
2021-09-26,11:45:02,1632674702.3530369, "SAME_HDR_RDY,1"
2021-09-26,11:45:04,1632674704.482091, SAME_PRE_DET
2021-09-26,11:45:04,1632674704.657402, SAME_HDR_DET
2021-09-26,11:45:05,1632674705.295789, "SAME_HDR_RDY,2"
2021-09-26,11:45:07,1632674707.4362147, SAME_PRE_DET
2021-09-26,11:45:07,1632674707.610701, SAME_HDR_DET
2021-09-26,11:45:08,1632674708.310957, "SAME_HDR_RDY,3"
2021-09-26,11:45:08,1632674708.3933723,
"SAME_MESSAGE, -CIV-RWT-000000+0300-832300-XDIF/005-..@.."
2021-09-26,11:45:08,1632674708.4084399, SAME_RWT
2021-09-26,11:45:08,1632674708.4170964, SAME_EOM_DET
2021-09-26,11:45:10,1632674710.1507761, SAME_PRE_DET
2021-09-26,11:45:10,1632674710.376535, SAME_EOM_DET
2021-09-26,11:45:12,1632674712.0675845, SAME_PRE_DET
2021-09-26,11:45:12,1632674712.2928886, SAME_EOM_DET
2021-09-26,11:45:13,1632674713.9220936, SAME_PRE_DET
2021-09-26,11:45:14,1632674714.1587334, SAME_EOM_DET
2021-09-26,14:45:01,1632685501.4217894, SAME_PRE_DET
2021-09-26,14:45:01,1632685501.5977864, SAME_HDR_DET
2021-09-26,14:45:02,1632685502.2980711, "SAME_HDR_RDY,1"
2021-09-26,14:45:04,1632685504.43761, SAME_PRE_DET
2021-09-26,14:45:04,1632685504.5511777, SAME_HDR_DET
2021-09-26,14:45:05,1632685505.2517164, "SAME_HDR_RDY,2"
2021-09-26,14:45:07,1632685507.4421623, SAME_PRE_DET
2021-09-26,14:45:07,1632685507.556113, SAME_HDR_DET
2021-09-26,14:45:08,1632685508.1943502, "SAME_HDR_RDY,3"
2021-09-26,14:45:08,1632685508.2768776,
"SAME_MESSAGE, -CIV-RWT-000000+0300-832300-XDIF/005-....."
2021-09-26,14:45:08,1632685508.2912626, SAME_RWT
2021-09-26,14:45:08,1632685508.3053653, SAME_EOM_DET
2021-09-26,14:45:10,1632685510.0963113, SAME_PRE_DET
2021-09-26,14:45:10,1632685510.3327475, SAME_EOM_DET
2021-09-26,14:45:12,1632685512.0117137, SAME_PRE_DET
2021-09-26,14:45:12,1632685512.1875565, SAME_EOM_DET
2021-09-26,14:45:13,1632685513.866546, SAME_PRE_DET
2021-09-26,14:45:14,1632685514.042239, SAME_EOM_DET

2021-09-26,17:45:01,1632696301.5296326,SAME_PRE_DET
2021-09-26,17:45:01,1632696301.6944702,SAME_HDR_DET
2021-09-26,17:45:02,1632696302.3952885,"SAME_HDR_RDY,1"
2021-09-26,17:45:04,1632696304.5344334,SAME_PRE_DET
2021-09-26,17:45:04,1632696304.648483,SAME_HDR_DET
2021-09-26,17:45:05,1632696305.3483875,"SAME_HDR_RDY,2"
2021-09-26,17:45:07,1632696307.4787047,SAME_PRE_DET
2021-09-26,17:45:07,1632696307.6539497,SAME_HDR_DET
2021-09-26,17:45:08,1632696308.292757,"SAME_HDR_RDY,3"
2021-09-26,17:45:08,1632696308.384348,
"SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-@ .D"
2021-09-26,17:45:08,1632696308.3918417,SAME_RWT
2021-09-26,17:45:08,1632696308.4042969,SAME_EOM_DET
2021-09-26,17:45:10,1632696310.1925497,SAME_PRE_DET
2021-09-26,17:45:10,1632696310.4293349,SAME_EOM_DET
2021-09-26,17:45:12,1632696312.049186,SAME_PRE_DET
2021-09-26,17:45:12,1632696312.2851305,SAME_EOM_DET
2021-09-26,17:45:13,1632696313.912836,SAME_PRE_DET
2021-09-26,17:45:14,1632696314.14048,SAME_EOM_DET
2021-09-26,20:45:01,1632707101.869585,SAME_PRE_DET
2021-09-26,20:45:01,1632707101.98317,SAME_HDR_DET
2021-09-26,20:45:02,1632707102.7350342,"SAME_HDR_RDY,1"
2021-09-26,20:45:04,1632707104.824363,SAME_PRE_DET
2021-09-26,20:45:04,1632707104.9884217,SAME_HDR_DET
2021-09-26,20:45:05,1632707105.751203,"SAME_HDR_RDY,2"
2021-09-26,20:45:07,1632707107.768513,SAME_PRE_DET
2021-09-26,20:45:07,1632707107.941692,SAME_HDR_DET
2021-09-26,20:45:08,1632707108.6424046,"SAME_HDR_RDY,3"
2021-09-26,20:45:08,1632707108.7264307,
"SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-D...@HM."
2021-09-26,20:45:08,1632707108.7392921,SAME_RWT
2021-09-26,20:45:08,1632707108.7552896,SAME_EOM_DET
2021-09-26,20:45:10,1632707110.4834936,SAME_PRE_DET
2021-09-26,20:45:10,1632707110.720341,SAME_EOM_DET
2021-09-26,20:45:12,1632707112.40054,SAME_PRE_DET
2021-09-26,20:45:12,1632707112.6369433,SAME_EOM_DET
2021-09-26,20:45:14,1632707114.2551625,SAME_PRE_DET
2021-09-26,20:45:14,1632707114.4914436,SAME_EOM_DET
2021-09-26,23:45:01,1632717901.4176857,SAME_PRE_DET
2021-09-26,23:45:01,1632717901.5916078,SAME_HDR_DET
2021-09-26,23:45:02,1632717902.2306056,"SAME_HDR_RDY,1"
2021-09-26,23:45:04,1632717904.422291,SAME_PRE_DET
2021-09-26,23:45:04,1632717904.5466352,SAME_HDR_DET
2021-09-26,23:45:05,1632717905.3500137,"SAME_HDR_RDY,2"
2021-09-26,23:45:07,1632717907.3772423,SAME_PRE_DET
2021-09-26,23:45:07,1632717907.5510154,SAME_HDR_DET
2021-09-26,23:45:08,1632717908.1908891,"SAME_HDR_RDY,3"

2021-09-26,23:45:08,1632717908.284901,
"SAME_MESSAGE,-CIV-RWT-000000+0300-832300-XDIF/005-...\$......-"
2021-09-26,23:45:08,1632717908.2924745,SAME_RWT
2021-09-26,23:45:08,1632717908.3028772,SAME_EOM_DET
2021-09-26,23:45:10,1632717910.0934987,SAME_PRE_DET
2021-09-26,23:45:10,1632717910.3301299,SAME_EOM_DET
2021-09-26,23:45:11,1632717911.9484,SAME_PRE_DET
2021-09-26,23:45:12,1632717912.1847985,SAME_EOM_DET
2021-09-26,23:45:13,1632717913.8124204,SAME_PRE_DET
2021-09-26,23:45:14,1632717914.039811,SAME_EOM_DET

Apéndice F. Integración del receptor de alertas

En este apartado, se detallan los materiales y conexiones necesarias para incorporar el receptor de alertas en una configuración apta para su uso en un entorno doméstico o comercial pequeño.

Como se describe en la sección 4.7, el receptor de alertas se ha integrado en un dispositivo tal como se ilustra en las Figuras 25 y 26, siguiendo la configuración propuesta en la Figura 23.

Debido a la falta de disponibilidad de un módulo prefabricado en el mercado, es necesario diseñar y construir el circuito mezclador de audio para combinar las señales de audio del receptor de alertas y el módulo ISD1820, encargado de reproducir el sonido de la alerta sísmica. En la Figura F.1 se muestra el diagrama esquemático correspondiente.

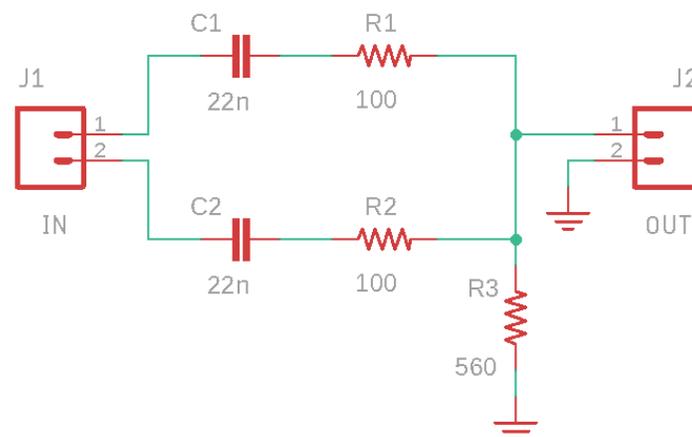


Figura F.1. Diagrama esquemático del circuito mezclador de audio.

En la Tabla F.2 se detallan los materiales necesarios para la construcción de esta configuración, con proveedor y número de parte para algunos de los componentes.

Tabla F.2. Lista de materiales del prototipo integrado para casa, comercio u oficina pequeña.

Cantidad	Descripción	Proveedor	Parte
1	Gabinete 155Hx179Wx100D mm	IDE Electric	ELT171
1	Antena telescópica 80 cm	—	—
1	Bocina 8 ohms 7 W	Steren	BOC-400
1	Mini No Break para módem	Steren	NB-050
1	Módulo grabador y reproductor de voz ISD1820	—	—
1	Módulo amplificador de audio TPA3118	—	—
1	Switch miniatura push normalmente abierto	Steren	AU-102R
2	Led de 5 mm color verde	AG Electronica	LED-G5D-AD-R30
2	Portaled 5 mm	AG Electronica	PORTALED2-5MM
1	Conector hembra invertido 2.5 mm para gabinete	—	—
1	Eliminador de 12V @ 2A	Steren	ELI-1240
16	Espaciador nylon 3 mm	Mouser	749-908-3MM
16	Tornillo cabeza de queso 2.5x12 mm con tuerca hexagonal	—	—
4	Tornillo cabeza de gota 5x10 mm con tuerca hexagonal	—	—
4	Círculo de goma antideslizante 10mm	—	—
3	Cables jumper F/F 0.1x4"	—	—
—	Cable de control 2 vías calibre 22	—	—
—	Cable duplex 2 vías calibre 22	—	—
—	Cable calibre 22	—	—

Para el propósito del prototipo, el circuito mezclador de audio se ensambló en una placa fenólica perforada utilizando la técnica de wire wrap con soldadura. La Tabla F.3 detalla los componentes utilizados en su construcción.

Tabla F.3. Lista de componentes del circuito mezclador de audio.

Parte	Descripción	Proveedor	Parte
C1, C2	Capacitor cerámico 22nF @ 50V	AG Electronica	CC-.022/50V-WELSON
R1, R2	Resistencia de carbón 100 ohms ¼W	AG Electronica	RC-100E/1/4
R3	Resistencia de carbón 560 ohms ¼W	AG Electronica	RC-560E/1/4
J1, J2	Conector de bloque con terminales de tornillo 2 pines 1mm	—	—
—	Placa fenólica perforada 4.5x4.5 cm	Steren	155

Finalmente, la Figura F.4 presenta el diagrama de conexiones que ilustra todos los módulos y componentes utilizados en el prototipo, diseñado para su empleo en hogares u oficinas pequeñas.

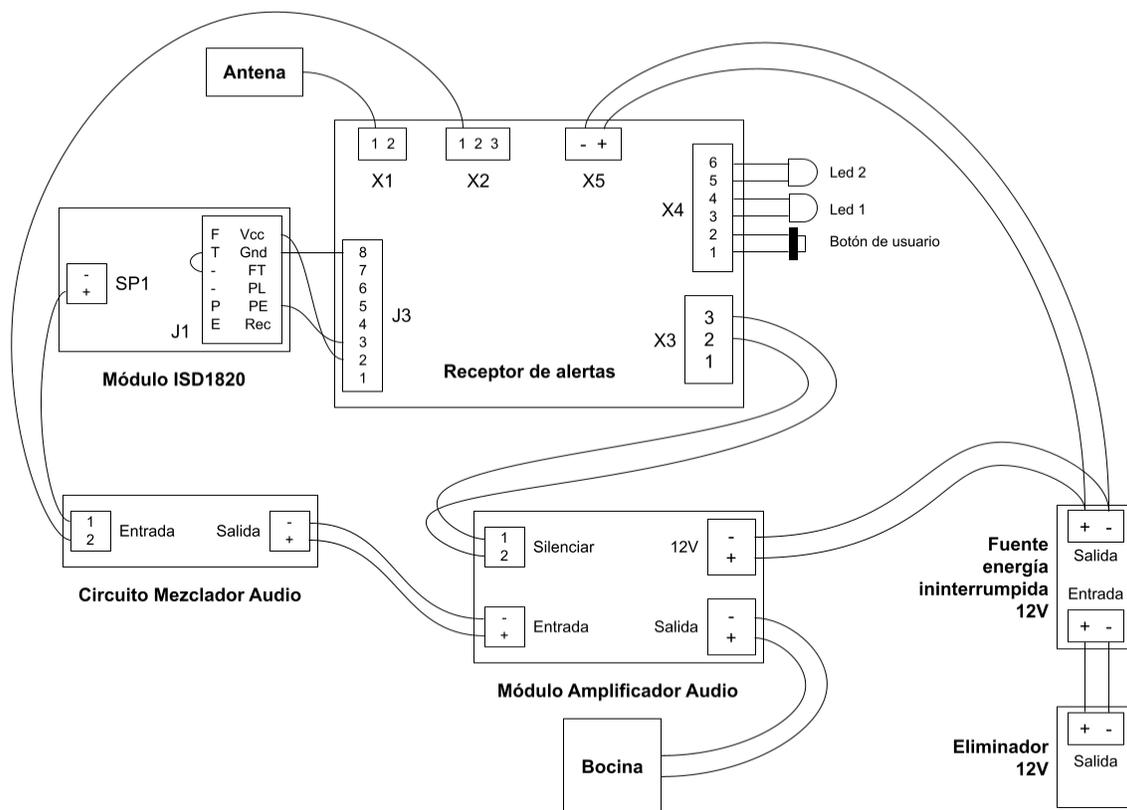


Figura F.4. Conexiones del prototipo integrado para casa, comercio u oficina pequeña.

Bibliografía

Atlas Nacional de Riesgos (2023). 1er Simulacro Nacional 2023. Recuperado el 13 de septiembre de 2023 de

<http://www.atlasmnacionalderiesgos.gob.mx/portal/1ersimulacronacional2023/>

Berger, A. (2002). Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. Kansas, USA: CMP Media.

Blanchard y Fabrycky. (1990) Systems Engineering and Analysis, New Jersey, USA: Prentice Hall

Centro de Instrumentación y Registro Sísmico (2005). Descripción CIRES. Recuperado el 10 de marzo de 2018 de http://cires.mx/cires_es.php

Centro de Instrumentación y Registro Sísmico (2005). Información sobre el SASMEX. Recuperado el 10 de marzo de 2018 de http://cires.org.mx/sasmex_es.php

Centro de Instrumentación y Registro Sísmico (s.f.). Histórico del Sistema de Alerta Sísmica Mexicano. Recuperado el 10 de julio de 2022 de http://www.cires.org.mx/sasmex_historico_n.php

Consumer Technology Association (2008). Public Alert: Delivers Emergency All-Hazard Warnings Everywhere All the Time. Recuperado el 31 de marzo de 2018 de https://web.archive.org/web/20151028090125/http://www.ce.org/CorporateSite/media/Standards-Media/PA_Whitepaper_9-3-08.pdf

Cortesi, G. (2010). Decision Matrix/Selection Matrix. Recuperado el 18 de Febrero de 2019 de <https://www.albany.edu/~gc227838/ist611/decision.pdf>

Cox, R. (2009). GM Emission Control Project Center - I Was There. Recuperado el 24 de abril de 2018 de

https://history.gmheritagecenter.com/wiki/index.php/GM_Emission_Control_Project_Center_-_I_Was_There

Derek Chafin (2014). Si4707 Weather Band Radio Receiver Library. Recuperado el 27 de Abril de 2019 de <https://github.com/infomaniac50/Arduino-Si4707-Library>

Electronic Notes (s.f.). VHF Broadcast FM Pre-emphasis & De-emphasis. Recuperado el 18 de marzo de 2019 de

<https://www.electronics-notes.com/articles/audio-video/broadcast-audio/vhf-fm-pre-de-emphasis.php>

Frenzel, L. (2012). Understanding Modern Digital Modulation Techniques. Recuperado el 19 de marzo de 2018 de

<http://www.electronicdesign.com/communications/understanding-modern-digital-modulation-techniques>

Gaceta Oficial de la Ciudad de México (2021). Acuerdo por el que se expide la Norma Técnica NT-SGIRPC-SDSAS-001-2021, Sistemas de Difusión Secundaria para el Alertamiento Sísmico.

Recuperado el 12 de noviembre de 2022 de <https://www.proteccioncivil.cdmx.gob.mx/storage/app/uploads/public/63a/0c8/348/63a0c8348338b903747009.pdf>

Ganssle, J. (2008). What's Embedded?. Recuperado el 24 de abril de 2018 de <https://www.embedded.com/electronics-blogs/break-points/4026846/What-s-Embedded->

Hamill, Paul (2004). Unit Test Frameworks: Tools for High-Quality Software Development. California, USA: O'Reilly Media.

Hamilton, T (2020). What is Test Driven Development (TDD)? Recuperado el 15 de enero de 2023 de <https://www.guru99.com/test-driven-development.html>

Heath, S. (2003). Embedded Systems Design. Oxford, UK: Elsevier.

Institute for Telecommunication Sciences (1996). Glossary of Telecommunication Terms. Recuperado el 24 de abril de 2018 de https://www.its.bldrdoc.gov/fs-1037/dir-017/_2452.htm

International Committee for Information Technology Standards (2002). 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII). New York, USA: American National Standards Institute.

ISO/IEC/IEEE (2017). ISO/IEC/IEEE 24765:2017 Systems and software engineering Vocabulary. New York, USA: Institute of Electrical and Electronics Engineers.

Jim Lindblom (2012). SparkFun Si4707 Weather Band Radio Firmware. Recueprado el 27 de Abril de 2019 de https://github.com/sparkfun/Si4707_Breakout/tree/master/firmware

Keeping, S. (2016). ZigBee-Powered Smart Lighting Solutions. Recuperado el 24 de abril de 2018 de <https://www.digikey.com/en/articles/techzone/2016/apr/zigbee-powered-smart-lighting-solutions>

Lin, Nai-Wei (2012). Integration Testing. Recuperado el 25 de Junio de 2021 de <https://www.cs.ccu.edu.tw/~naiwei/cs5812/st8.pdf>

Ludovic Apvrille (2012). UML for Embedded Systems Introduction. Recuperado el 30 de noviembre de 2019 de http://soc.eurecom.fr/UMLEmb/Course_UML_0_Introduction.pdf

Massa, A. y Barr, M. (2006). Programming Embedded Systems, 2nd Edition. Recuperado el 17 de mayo de 2022 de <https://learning.oreilly.com/library/view/programming-embedded-systems/0596009836/ch04.html>

Meteorological Service of Canada (2004). Weatheradio Network. Recuperado el 11 de marzo de 2018 de http://www.ec.gc.ca/media_archive/press/2004/040107_b_e.htm

Microchip (2015). MCP16301/H High-Voltage Input Integrated Switch Step-Down Regulator. Recuperado el 22 de junio de 2019 de <http://ww1.microchip.com/downloads/en/devicedoc/20005004d.pdf>

Microchip (s.f. a). 16-bit PIC Microcontroller Peripheral Integration. Recuperado el 20 de Enero de 2019 de <http://ww1.microchip.com/downloads/en/DeviceDoc/30010109E.pdf>

Microchip (s.f. b). 32-bit PIC and SAM Microcontrollers Peripheral Integration. Recuperado el 20 de Enero de 2019 de <http://ww1.microchip.com/downloads/en/DeviceDoc/60001455D.pdf>

Microchip (s.f. c). 8-bit AVR Microcontrollers Peripheral Integration. Recuperado el 20 de Enero de 2019 de <http://ww1.microchip.com/downloads/en/DeviceDoc/30010135D.pdf>

Microchip (s.f. d). 8-bit PIC Microcontroller Peripheral Integration. Recuperado el 20 de Enero de 2019 de <http://ww1.microchip.com/downloads/en/DeviceDoc/30010068F.pdf>

Microchip (s.f. e). Microcontrollers, Digital Signal Controllers and Microprocessors. Recuperado el 19 de Enero de 2019 de <https://www.microchip.com/design-centers/microcontrollers>

National Weather Service (1971). Operations of the National Weather Service, October 1971.

National Weather Service (2015). History of the National Weather Service. Recuperado el 11 de marzo de 2018 de <https://www.weather.gov/timeline>

Nelson, W.C. (2002). American Warning Dissemination and NOAA Weather Radio. Tesis de Maestría por la Universidad de Carolina del Norte. Recuperado el 11 de marzo de 2018 de <http://cliffedits.com/thesis/thesis.html>

NOAA (2011). NOAA Weather Radio (NWR) All Hazards Specific Area Message Encoding (SAME). Recuperado el 10 de marzo de 2018 de <http://www.nws.noaa.gov/directives/sym/pd01017012curr.pdf>

NOAA (2015). National Weather Service Marine Products via NOAA Weather Radio. Recuperado el 18 de marzo de 2018 de <http://www.nws.noaa.gov/os/marine/wxradio.htm>

NOAA (s.f. a). County Coverage Listings by State and for Marine Zones. Recuperado el 16 de marzo de 2018 de http://www.nws.noaa.gov/nwr/coverage/county_coverage.html

NOAA (s.f. b). Emergency Warnings for People with Hearing Loss. Recuperado el 11 de marzo de 2018 de <http://www.nws.noaa.gov/nwr/info/shhh.html>

NOAA (s.f. c). NOAA Weather Radio All Hazards. Recuperado el 11 de marzo de 2018 de <http://www.nws.noaa.gov/nwr/>

Noergaard, T. (2005). Embedded Systems Architecture. Oxford, UK: Elsevier.

Pablo Domínguez (2017). Gestiona tu proyecto de desarrollo: En qué consiste el modelo en cascada. Recuperado el 18 de noviembre de 2019 de <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>

Royal Automobile Club of Victoria (2004). Effectiveness of ABS and Vehicle Stability Control Systems. Recuperado el 24 de abril de 2018 de https://www.monash.edu/_data/assets/pdf_file/0020/218270/racv-abs-braking-system-effectiveness.pdf

Secretaría del Trabajo y Previsión Social (2008). Norma Oficial Mexicana NOM-001-STPS-2008. Edificios, locales, instalaciones y áreas en los centros de trabajo - Condiciones de seguridad. Recuperado el 22 de marzo de 2023 de <https://www.stps.gob.mx/bp/secciones/dgsst/normatividad/normas/nom-001.pdf>

Seed Studio (2019). How do I prepare the Bill of Materials (BOM) file for Seed Fusion PCBA Orders?. Recuperado el 31 de agosto de 2019 de <http://support.seedstudio.com/knowledgebase/articles/1886734-how-do-i-prepare-the-bill-of-materials-bom-file>

Seed Studio (2019). How to generate Gerber and Drill files from Eagle. Recuperado el 31 de agosto de 2019 de <http://support.seedstudio.com/knowledgebase/articles/1176949-how-to-generate-gerber-and-drill-files-from-eagle>

Seed Studio (s.f.). Fusion PCB & PCB Assembly & Flexible PCB. Recuperado el 28 de agosto de 2019 de <https://www.seedstudio.com/fusion.html>

Servicio Geológico Mexicano (2017). Sismología de México. Recuperado el 10 de marzo de 2018 de <https://www.sgm.gob.mx/Web/MuseoVirtual/Riesgos-geologicos/Sismologia-de-Mexico.html>

Silicon Laboratories (2008). Si4707-B20. Recuperado el 31 de marzo de 2019 de <https://www.silabs.com/documents/public/data-sheets/Si4707-B20.pdf>

Silicon Laboratories (2014). Si47xx Antenna, Schematic, Layout, and Design Guidelines. Recuperado el 31 de marzo de 2019 de <https://www.silabs.com/documents/public/application-notes/AN383.pdf>

Silicon Laboratories (2014). Si47XX Programming Guide. Recuperado el 31 de marzo de 2019 de <https://www.silabs.com/documents/public/application-notes/AN332.pdf>

Silicon Laboratories (2017). Si4707 Weather Band Radio Receiver. Recuperado el 31 de Marzo de 2019 de <https://www.silabs.com/products/audio-and-radio/fm-radios/si4707-radio-receiver>

Solberg, A. (2018). Recuperado el 23 de Marzo de 2019 de <https://www.nasa.gov/consortium/SystemsEngineeringManagementPlanTechnicalContent>

Sparkfun (s.f. a). Arduino Pro Mini 328 - 3.3V/8MHz. Recuperado el 16 de abril de 2019 de <https://www.sparkfun.com/products/11114>

Sparkfun (s.f. b). SparkFun Weather Band Receiver Breakout - Si4707. Recuperado el 15 de abril de 2019 de <https://www.sparkfun.com/products/11129>

Texas Instruments (2015). TLV702 300-mA, Low-IQ, Low-Dropout Regulator. Recuperado el 22 de junio de 2019 de <http://www.ti.com/lit/ds/symlink/tlv702.pdf>

Tran, E. (1999). Requirements & Specifications. Recuperado el 16 de Marzo de 2019 de https://users.ece.cmu.edu/~koopman/des_s99/requirements_specs/

U.S. Department of Homeland Security (2011). National Interoperability Field Operations Guide Version 1.4. Recuperado el 18 de marzo de 2018 de <https://www.dhs.gov/xlibrary/assets/nifog-v1-4-resized-for-pda-viewing.pdf>

United States Coast Guard (2018). U.S. VHF Channel Information. Recuperado el 11 de marzo de 2018 de <https://www.navcen.uscg.gov/?pageName=mtvhf>

Weather Radios Direct (2011). The History of NOAA Weather Radio. Recuperado el 10 de marzo de 2018 de <http://weatherradios.com/blog/the-history-of-noaa-weather-radio>

Weigel, E. (1974). NOAA Weather Radio: Stormalert. Brochure from the National Weather Service.

White, E. (2011). Making Embedded Systems. California, USA: O'Reilly Media.

Wiki-Robotics (2007). PIC Bootloader. Recuperado el 9 de abril de 2019 de http://www.learobotics.com/wiki/index.php?title=PIC_Bootloader