Combinador de Información Primaria y Secundaria para Extractor Digital de Datos de Radar en Sistemas de Vigilancia

Autor: Ignacio A. Montamat

Director: Giorgio M. Caranti

Mayo de 2015

El presente trabajo contiene reserva de información. Ante requerimiento de mayores detalles, comunicarse con el autor a nachomontamat@gmail.com.

RESUMEN

Un Extractor Digital de Datos de Radar (E.D.D.R.) es un subsistema -de *hardware* y *software*- del Sistema de Radar, que permite la representación visual de las aeronaves detectadas. A nivel de *software*, puede considerarse al E.D.D.R. compuesto de una serie de aplicaciones interactuantes: los procesadores de información primaria y secundaria, la aplicación de asociación primaria y secundaria (o Combinador) y la aplicación de seguimiento de objetivos (o *Tracker*, en inglés).

En el presente trabajo se muestra el proceso de rediseño, implementación y validación de la aplicación de *software* de asociación de información primaria y secundaria de un E.D.D.R. perteneciente a un Sistema de Radar militar de vigilancia de largo alcance de la Fuerza Aérea Argentina (FAA).

Palabras clave: Combinador, radar, vigilancia, plot, software, asterix.

ÍNDICE DE CONTENIDOS

CAPÍTULO I

Introducción

Tanto en aplicaciones civiles como militares, al observar el espacio aéreo por sistema de radar, comúnmente se hace uso de dos tipos de sensores: primario y secundario[1]. Estos, por su diseño, revelan características de distinta naturaleza acerca de las aeronaves detectadas.

El sensor primario extrae posición acimutal, rango y, en el caso de radares 3D, elevación. El sensor secundario, además, extrae identificación, modos de operación, altitud, rumbo y otros parámetros relevantes de la aeronave. Ambos sensores se encuentran montados sobre un mismo eje (Wolff, Radar Basics).

Durante la operación de un sistema de radar es deseable en todo momento que, de existir información primaria y secundaria de una aeronave, la misma pueda ser visualizada de manera asociada, es decir: un solo punto en el espacio con toda la información disponible concentrada.

El **E**xtractor **D**igital de **D**atos de **R**adar (en adelante, E.D.D.R.) es el componente del sistema de radar que posibilita la asociación, representación y transmisión digital de la información analógica capturada por los sensores primario y secundario. El mismo puede encontrarse implementado como un sistema discreto, que se añade a un sistema de radar preexistente -por lo general antiguo, como es el caso en este trabajo- o bien, incorporado dentro de las etapas de adquisición y procesamiento de señal, en sistemas modernos.

Un E.D.D.R., en general, puede considerarse compuesto de tres subsistemas perfectamente diferenciables:

- Procesador de información primaria
- Procesador de información secundario
- Procesador de asociación y seguimiento (o Combinador-Tracker[2])

[1] A veces, simplemente se denomina a estos sensores: radar primario y radar secundario.

[2] El término *tracker*, se utiliza para referirse a un seguidor o rastreador, en inglés.

La tarea fundamental de los procesadores primario y secundario consta de convertir las señales analógicas, provenientes de los detectores del radar, en señales digitales denominadas plots[3]. Estos son transmitidos, luego, hacia el procesador de asociación y seguimiento.

Una vez dentro del procesador de asociación y seguimiento, un bloque denominado Combinador interpreta la información brindada por los procesadores primario y secundario, la unifica y crea plots más completos denominados plots combinados.

Finalmente, cada plot analizado por el Combinador es transmitido hacia el bloque de seguimiento (o Rastreador)[4]. Este último, se encarga de correlacionar los sucesivos plots de cada aeronave para determinar, con cada vuelta de radar, la trayectoria (pista) que esta sigue. Para ello, hace uso de algoritmos de asociación y filtrado (o suavizado) de trayectorias.

El siguiente diagrama de bloques resume el principio de funcionamiento de un E.D.D.R.

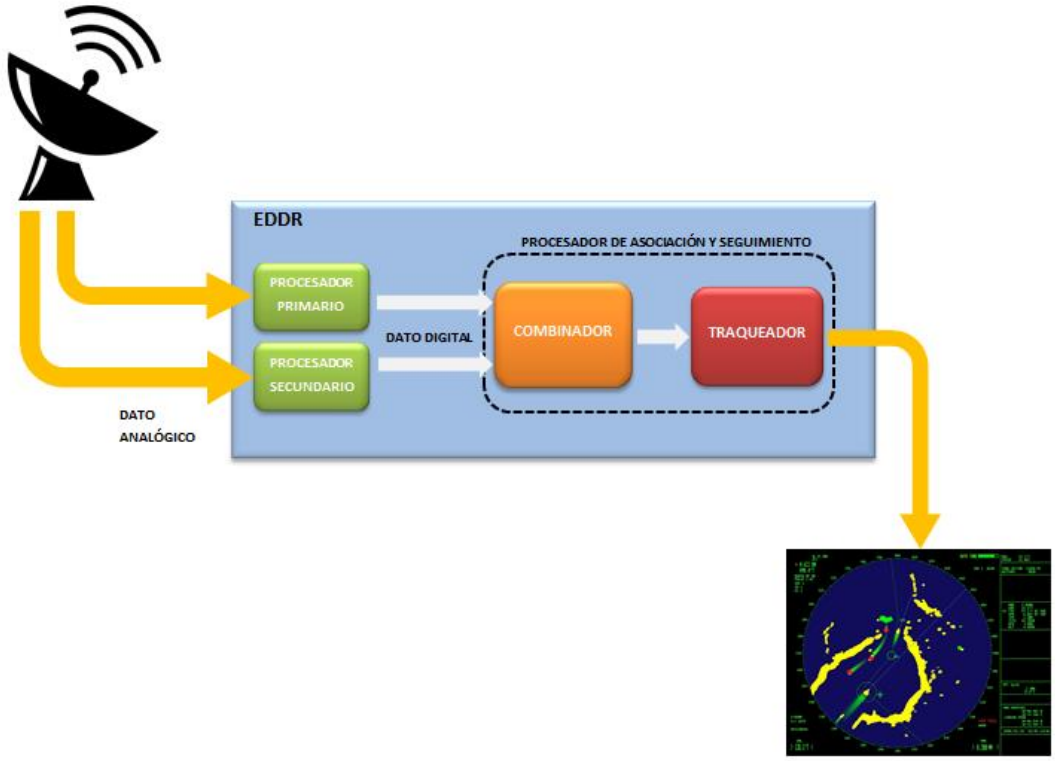


Figura 1.1: Diagrama de bloques del E.D.D.R.

[3] Representación digital de un blanco. Consultar Capítulo II, Marco Teórico para mayor información.
[4] Puede encontrarse en la misma unidad de ejecución que el Combinador o separado.

El presente trabajo se enfoca en el rediseño del bloque Combinador[5] del procesador de asociación y seguimiento de un E.D.D.R. perteneciente a la Fuerza Aérea Argentina. Este último, opera en un sistema de radar militar, con radar primario y secundario 2D, más un radar de altura asociado (separado) que aporta capacidades 3D al sistema. El conjunto se caracteriza por un tiempo de revolución de 12 segundos y un alcance típico de unas 240 MN (millas náuticas) o unos 430Km aproximadamente. La resolución angular de los sensores primario y secundario es de aproximadamente 1.5° y, en rango, menor a 0.5MN.

[5] En adelante, al hablar de cada bloque (Combinador, Tracker, etc.) se hará referencia a la aplicación de *software* del mismo, a menos que se indique lo contrario.

Problema y preguntas de investigación

El sistema de radar sobre el que aquí se trabaja, se encuentra en funcionamiento actualmente, aunque de manera defectuosa. Las mediciones que ofrece son inaceptables para el personal de FAA y por ello debió ser intervenido.

Si bien sus deficiencias son atribuibles, en parte, a su desgaste por antigüedad y su ubicación en una zona de mucho *clutter*[6], las razones principales de sus fallas son dos:

- El bloque Combinador del que dispone introduce demasiados errores: aeronaves duplicadas, omisión de ciertos plots, información desasociada, etc.

- El bloque Rastreador (o *Tracker*) no funciona.

Tanto las necesidades de mejorar la fiabilidad y robustez del sistema de radar, como la de añadirle capacidades de seguimiento de objetivos, son las que impulsan este proyecto de maestría. Particularmente, se busca aquí trabajar sobre los inconvenientes actuales del bloque Combinador; modificando o rediseñando todos aquellos elementos de *software* que sean necesarios para satisfacer los requisitos de funcionamiento exigidos por FAA. Uno de estos requisitos es que tanto el Combinador como el *Tracker* coexistan y se comuniquen dentro de la misma unidad de ejecución (corran en la misma *CPU*). Esto es un aspecto importante a tener en cuenta en el diseño de la solución buscada.

Otro requisito importante para la nueva implementación es lograr que el E.D.D.R. soporte el standard ASTERIX (Eurocontrol A.) -categorías 34 y 48- para la transmisión de la información. Esto es algo muy importante para asegurar la compatibilidad de este sistema de radar con el resto de los sistemas de radar actuales en la red nacional de vigilancia.

En vistas de lo expuesto, algunos de los interrogantes a responder para llevar a cabo este proyecto son: ¿Cómo funciona el bloque Combinador actual? ¿Qué tipo de errores introduce? ¿Cómo está implementado? ¿Qué implica el aportar compatibilidad para el standard ASTERIX? ¿Qué Sistemas Operativos de la actualidad son compatibles tanto con la aplicación a desarrollar como con el *Tracker*?

[6] Interferencia. Se expande sobre ello más adelante en el Marco Teórico.

Para responder a esta última cuestión, deben comprenderse las características fundamentales del Combinador y el *Tracker* que condicionan las plataformas elegibles en términos de seguridad, periféricos, concurrencia y drivers asociados, principalmente.

Simultáneamente, de ser factible, sería conveniente investigar acerca de qué implementaciones existen en sistemas similares a nivel nacional o internacional.

Antecedentes

El sistema de radar al que pertenece el E.D.D.R. del que aquí se habla es un sistema antiguo en tecnología (aproximadamente de la década del '70). Este fue donado por España al Ministerio de Defensa de nuestro País, en el año 2007 y se encuentra en operación desde hace poco más de 5 años.

Originalmente, este sistema de radar contaba con una consola de visualización analógica basada en fósforo, por lo que no utilizaba un E.D.D.R. La incorporación de este último al sistema tuvo lugar en el año 2010, aproximadamente. Dicha implementación se realizó íntegramente en nuestro País, por parte del personal de la FAA[7], que aprovechó el "*know-how"* con el que se contaba, por tener experiencia en el desarrollo de ciertos E.D.D.R. para sistemas de radar similares.

En materia de *hardware*, las unidades de procesamiento de los E.D.D.R. desarrollados por FAA fueron siendo actualizadas en tecnología con el correr de los años. Partiendo de un microprocesador 8086 –usado en el primer E.D.D.R. desarrollado en Argentina, en los '90- se fueron mejorando paulatinamente. En la actualidad, la implementación de E.D.D.R. que se utiliza, cuenta con un microprocesador Pentium compatible o superior.

En cuanto al *software*, tanto para el Combinador como para el Rastreador, históricamente (y hasta la actualidad) se empleó una sola aplicación global escrita en C (compilada con Borland-C) sobre el sistema operativo MS-DOS. Esta aplicación contiene toda la lógica de ambos bloques y fue creada, de manera colaborativa, por varias personas encargadas del mantenimiento del sistema a lo largo de los años.

Dado que se trata de un sistema de uso militar, no se cuenta con acceso a documentación de sistemas de radares similares a nivel nacional. Por otro lado, los nuevos radares 3D de vigilancia -fabricados por INVAP- fueron concebidos con tecnología moderna que digitaliza la señal capturada *on-site*[8], por lo que no requieren de un E.D.D.R. discreto (INVAP-Web).

[7] Comodoro Jorge Muñoz, principalmente.

[8] On-site, en inglés o "in-situ". La información analógica es digitalizada tempranamente en el proceso de adquisición de señal. Los detalles de esta implementación no son públicos sino conocimiento del autor.

Objetivos

Generales

Desarrollar una aplicación de *software* de asociación de información primaria y secundaria, compatible con el standard Asterix Eurocontrol (categorías 34 y 48), para un Extractor de Datos Digitales de Radar de FAA.

Específicos

1. Relevar el funcionamiento del actual bloque Combinador, comprendiendo su principio de funcionamiento y detectando sus deficiencias. Para ello, se hará uso de ciertas herramientas de documentación de *software* (Doxygen).

2. Investigar acerca de qué Sistemas Operativos actuales se adaptan a los requerimientos de la aplicación requerida, elegir uno e implementarlo en el E.D.D.R.

3. Investigar acerca de qué algoritmos de asociación se utilizan en radares de vigilancia, tanto nacional como internacionalmente.

4. Estudiar el standard Asterix Eurocontrol para entender qué implica el hecho de que la aplicación a desarrollar sea compatible con las categorías 34 y 48.

5. Rediseñar y desarrollar el bloque Combinador, a modo de cumplir con los requerimientos especificados por FAA y validar su funcionamiento mediante casos de uso.

6. Documentar todo el proceso de desarrollo convenientemente.

CAPÍTULO II

Marco Teórico

Como se mencionó en la introducción, un sistema de radar de vigilancia aérea es un conjunto de radares -primario y secundario- más una serie de subsistemas que articulan para posibilitar la detección y seguimiento de aeronaves. En este contexto, se explican a continuación ciertos conceptos fundamentales en la comprensión de la problemática a resolver. Los mismos explican cuáles son y cómo se relacionan los elementos intervinientes en un sistema de radar como el del presente trabajo.

Definiciones

Sistema de radar

Conjunto de antenas emisoras/receptoras de ondas de radio, fuentes de alimentación, E.D.D.R., consolas de visualización y elementos de interconexión que se utiliza para medir distancia, elevación, dirección y velocidad de objetos estáticos o móviles; ya sea se trate de aeronaves, barcos, vehículos motorizados, formaciones meteorológicas o el propio terreno. Su principio de funcionamiento se basa en la emisión de un impulso de radiofrecuencia que se refleja en un objetivo; la energía reflejada es detectada por la antena receptora y, mediante la medición del tiempo de viaje del impulso transmitido, se determina la distancia a la que se encuentra el objetivo. La posición angular del mismo está determinada por el ángulo en el que la antena emite y recibe el pulso (en referencia a un punto fijo, generalmente, el norte magnético). Esta señal detectada se denomina "eco" y, convenientemente acondicionada, puede brindar a los usuarios del sistema de radar gran cantidad de información (Skolnik, 1990).

Radar primario

Analiza señales ecos para descubrir objetivos (o blancos) determinando posición angular respecto al punto cardinal norte (o acimut), distancia de los mismos (o rango) y, en

radares 3D, elevación. Se dice que funciona de manera no cooperativa, ya que sus mediciones no dependen de ninguna interacción con el blanco detectado.

Radar secundario

Interroga a un determinado objetivo (generalmente una aeronave) mediante la transmisión de una señal de radiofrecuencia y este, a través de su transponder, responde al radar con una señal de distinta frecuencia, brindando una serie de datos propios (identificación, altura, etc.). Tanto el radar secundario como el transponder de la aeronave funcionan de manera cooperativa.

Transponder

Dispositivo utilizado en telecomunicaciones cuyo nombre viene de la fusión de las palabras inglesas Transmitter (Transmisor) y Responder (Respondedor). En la aeronave se utiliza para responder a las interrogaciones realizadas por el radar secundario, informando ciertos datos propios de la aeronave como identificación, altitud, modos de operación, entre otros (Wolff, Radar Basics).

Blanco

Un objetivo determinado. Puede ser una aeronave, un barco, un automóvil o cualquier objeto de interés cuya ubicación espacial sea relevante.

Plot

Dato digital resultante de la correlación de sucesivos ecos de radar durante una exploración de antena. Representa la presencia de un blanco en un determinado punto del espacio.

Pista (o Track)

Dato resultante de la correlación de sucesivos plots durante varias vueltas de antena. Representa la trayectoria que sigue la aeronave en su desplazamiento –posición en el plano, velocidad, altitud y rumbo. Una pista (o *track*) es un superconjunto de un plot.

Al igual que con los plots, existe una pista por aeronave y por vuelta de antena, pero difiere respecto a estos en que para su determinación se tiene en cuenta a los estados anteriores de la aeronave en cuanto a velocidad y rumbo.

Clutter

Todo eco indeseable que recibe el sistema radar. Este puede venir tanto de objetos estáticos como móviles y su efecto es disminuir la calidad de las mediciones del sistema ya que obstaculiza la detección de posibles blancos. Su efecto puede variar desde una leve atenuación de la señal recibida a un bloqueo completo de la misma, privando al sistema de radar de detectar blancos más alejados.

Sistema Operativo

Paquete de *software* que administra los recursos de *hardware* de un sistema y proporciona servicios de diversos tipos al usuario para acceder a dichos recursos.

ASTERIX

Proviene de las siglas **A**ll purpose **ST**ructured **E**urocontrol Su**R**veillance **I**nformation E**X**change. Es un standard, de origen europeo, para el formato de mensajes de comunicación entre sistemas de radar de diversos tipos.

El E.D.D.R.

Como ya se adelantó, un Extractor Digital de Datos de Radar es un componente del sistema de radar que posibilita la visualización y transmisión de la información capturada por las antenas. Sus funciones principales son:

- Tomar la señal analógica proveniente de los receptores del radar y digitalizarla.
- Procesar la señal digital para extraer datos, filtrando información espurrea e innecesaria.

- Elaborar plots –primarios, secundarios o combinados- y pistas para las aeronaves detectadas.

- Retransmitir los datos digitales hacia consolas o monitores remotos.

La importancia incorporar un E.D.D.R. a un sistema de radar antiguo radica en posibilitar la unificación del "lenguaje" de transferencia de información entre el sistema de radar y los centros de procesamiento. Además, al tratarse de información digital, se incrementa el flujo efectivo de datos en la comunicación y se reduce el ancho de banda necesario en el canal de transmisión empleado. Todo esto es posible sin intervenir significativamente el sistema de radar original –lo cual, además de costoso, muchas veces es imposible.

En la actualidad, en sistemas modernos, la conversión analógica-digital de los ecos recibidos se hace tempranamente en la etapa de adquisición de señal. Por esta razón, las capacidades del E.D.D.R. se encuentran "distribuidas" dentro de todo el sistema de radar, en lugar de verse concentradas en un sistema discreto. Sin embargo, la instancia de procesamiento de los datos digitales para combinación y seguimiento siempre es diferenciable, ya que se utilizan unidades de cómputo (CPU's) para su implementación (Wolff, Radar Signal Processor).

En las Figuras 2.1, 2.2 y 2.3, para mayor claridad, se muestran las ventajas que agrega un E.D.D.R. a un sistema de radar. También se destaca la diferencia entre la información que entrega el bloque Combinador con respecto al bloque Rastreador.

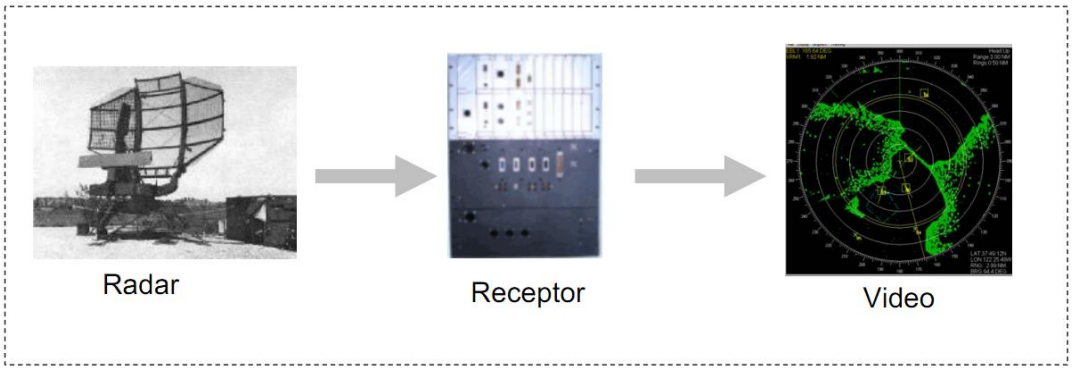


Figura 2.1: Sistema de Radar ***sin*** E.D.D.R.

En la figura 2.1, la información de cada plot se transmite a través de una señal analógica denominada video[9]. Esta señal, convenientemente acondicionada por el receptor, se imprime sobre una pantalla analógica fosforescente.

En la figura 2.2 se muestra como, a partir de la incorporación del E.D.D.R., la misma información de video más otra información adicional puede verse en monitores actuales, de manera digital.

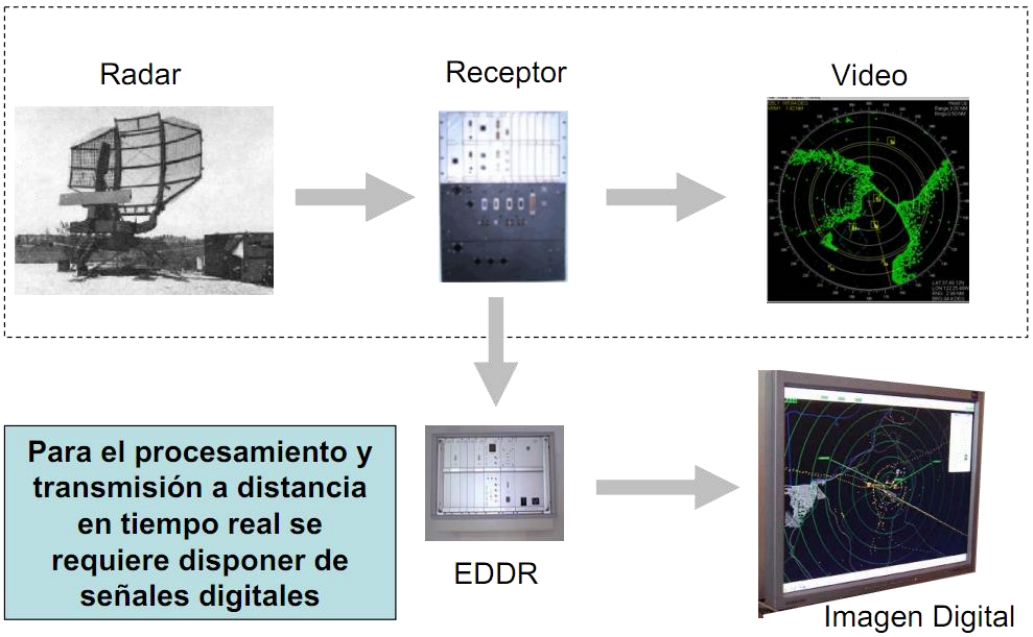


Figura 2.2: Sistema de Radar ***con*** E.D.D.R ***sin*** *Tracker*.

[9] Se denomina "**video**" a la señal analógica que sale de los detectores del radar. En otras palabras, es la señal de eco, bajada a banda base.

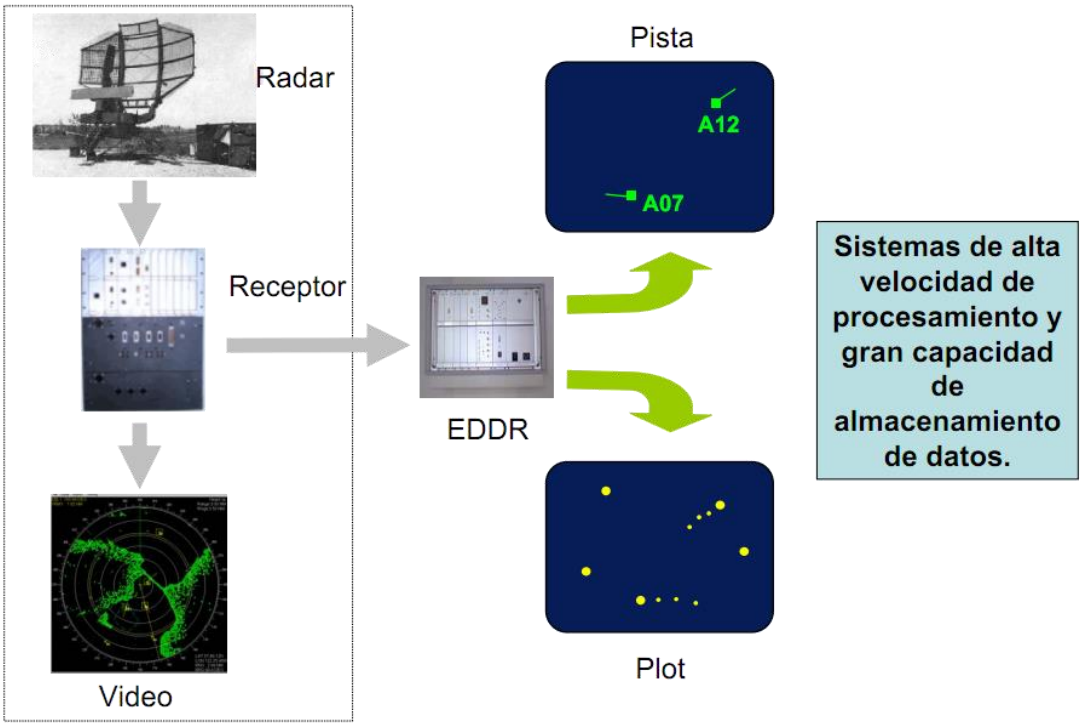


Figura 2.3: Sistema de Radar ***con*** E.D.D.R ***con*** *Tracker*[10].

En pocas palabras, el Combinador permite una asociación de la información primaria y secundaria para entregar plots más descriptivos. El bloque Rastreador, en cambio, recoge información sobre los distintos blancos detectados y la correlaciona en cada vuelta de radar para, a través de un proceso estadístico, elaborar información sobre la trayectoria –o pista– que podría estar siguiendo cada objetivo.

[10] Las imágenes de este capítulo y del siguiente fueron obtenidas de una presentación hecha por el Centro de Investigaciones Aplicadas (CIA) en un seminario para el Instituto Universitario Aeronáutico, Córdoba, Argentina.

CAPÍTULO III

Relevamiento

Para la primera parte de este trabajo se realizó un relevamiento, tanto en *hardware* como *software*, del E.D.D.R. de FAA. Los datos recopilados en esta etapa se utilizan como puntapié inicial para el posterior desarrollo del Combinador, ya que permiten comprender el principio de funcionamiento del sistema de manera global.

Hardware

El E.D.D.R. en cuestión, se compone de un conjunto de módulos de *hardware* estructurados a partir de un rack estándar de 19". Cada módulo presenta una interfaz bien definida de comunicación con el resto. Dicha comunicación se realiza a través de un *bus* común, en donde los módulos se encuentran insertos como si fueran tarjetas o placas de expansión en una PC.

Los módulos (o placas) principales en los que se divide el E.D.D.R. son siete; sin embargo, se detallan aquí las cuatro más importantes: el detector y adaptador de señales, más las tres unidades de procesamiento, denominadas: Procesador Primario, Procesador Secundario y Combinador-Traqueador. En la siguiente página se muestra una imagen real de este sistema y se indica cada una de sus partes. Más adelante se detalla la función de cada una de estas cuatro etapas principales.

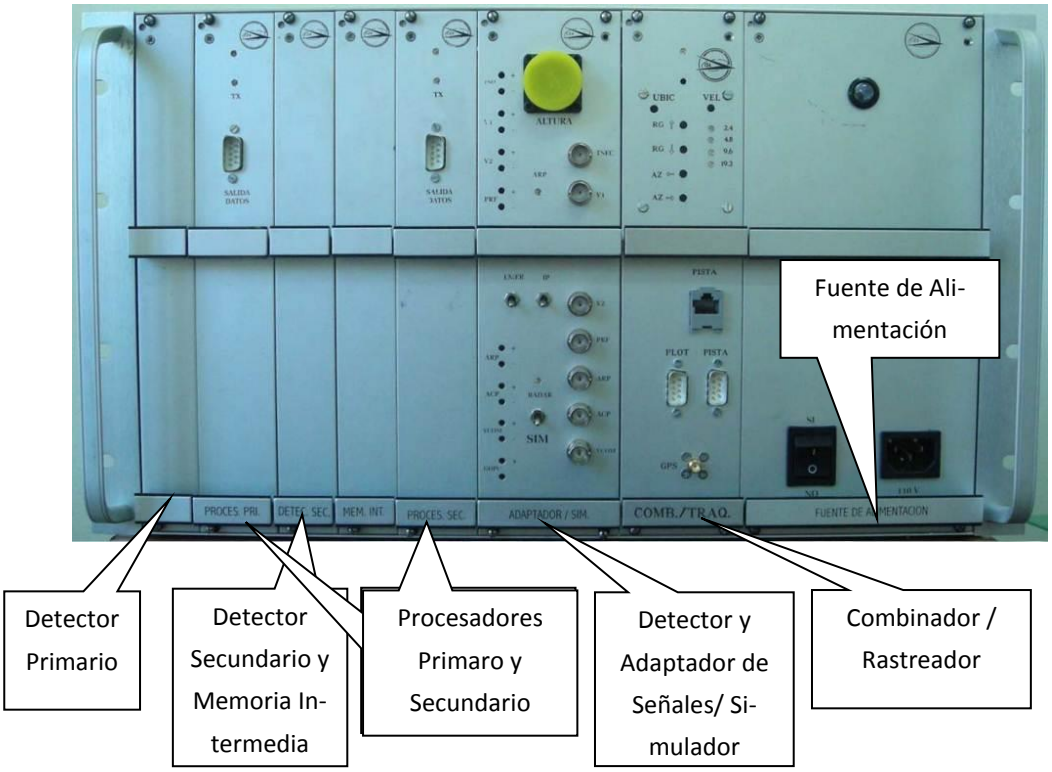


Figura 3.1: Extractor Digital de Datos de Radar de FAA

Detector y Adaptador de Señales

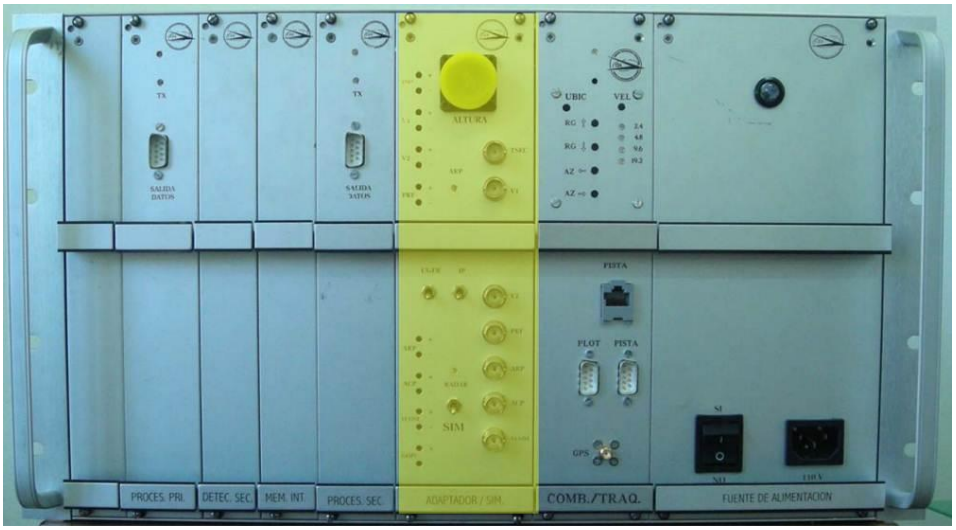


Figura 3.2: Detector y Adaptador de señales

Siete conectores coaxiales y un conector circular multi-contacto, son los puertos de ingreso de señales desde el Sistema Radar hacia el bloque Detector y Adaptador del E.D.D.R. En esta etapa, se filtran las señales analógicas provenientes de los receptores de radar con el fin de eliminar su componente de ruido en la medida que sea posible. Estas señales analógicas pueden ser de dos tipos: Sincronismo o Video.

- **Sincronismo:** sirven como puntos de referencia para el E.D.D.R., indicando por ejemplo, el momento en el que se da una emisión de radar, comienzo de cálculo de rango, cruce con el punto cardinal norte (Top Norte), etc.
- **Video:** Las señales de video acusan la presencia de un blanco y, posiblemente también, algún código de respuesta del transponder interrogado, en caso de tratarse del radar secundario. Este último tipo de señal se denomina "video compuesto".

Las señales de video son utilizadas en conjunto con las señales de sincronismo para detectar rango. Por otro lado, tanto el acimut como la altura de los blancos se determinan en base a la lectura de los *encoders* respectivos al instante de la detección.

Mediante una llave situada en el panel frontal de este Detector y Adaptador, puede conmutarse su modo de funcionamiento para alternar entre Adaptador y Simulador. En el primer modo, el E.D.D.R. se configura para recibir las señales de video y sincronismo, es decir, cuando realmente está conectado al sistema radar. En el segundo modo de configuración, el equipo simula todas las señales correspondientes a los radares Primario y Secundario, incluidos los ecos de estos sensores. Por defecto, en este modo se simulan dieciséis plots primarios y dieciséis plots secundarios por giro de antena. Este modo posibilita efectuar ensayos de mantenimiento y calibración sobre las distintas partes del E.D.D.R. cuando no se dispone físicamente del sistema de radar.

Unidad de Ejecución

Tanto los procesadores Primario y Secundario, como el Combinador y el Rastreador, se encuentran implementados en arquitecturas del tipo SBC (*Single Board Computer*[11], en inglés). Cada uno de ellos es básicamente una aplicación C que se ejecuta sobre una

[11] En compatibilidad con el standard PC/104: http://www.pc104.org/ (20-05-2014)

arquitectura x86. Específicamente, el modelo de las SBC que se utilizan en estos procesadores es el PPM-GX500 de la firma WinSystems, compatible con el estándar de apilado PC-104[12] como se muestra en la Figura 3.3, a continuación.

Figura 3.3: SBC PPM-GX500

Sus características son las siguientes:

- Procesador AMD Geode™ GX500@1.0W (367MHz)
- Memoria de 256 MB SODIMM PC2700 DDR
- Interfaz Ethernet INTEL 82551ER 10/100 Mbps
- Unidad de almacenamiento tipo *Compact Flash* de 2GiB
- 4 Puertos seriales RS-232 tipo 16550A
- 2 puertos USB 1.1
- 1 puerto IDE
- 1 puerto CompactFlash
- Factor de forma PC/104 (90mm x 96mm)

Cada una de estas SBC se encuentra montada sobre una placa base como la que se muestra en la Figura 3.4, a continuación.

[12] Información disponible en http://www.pc104.org/ (20-05-2014)

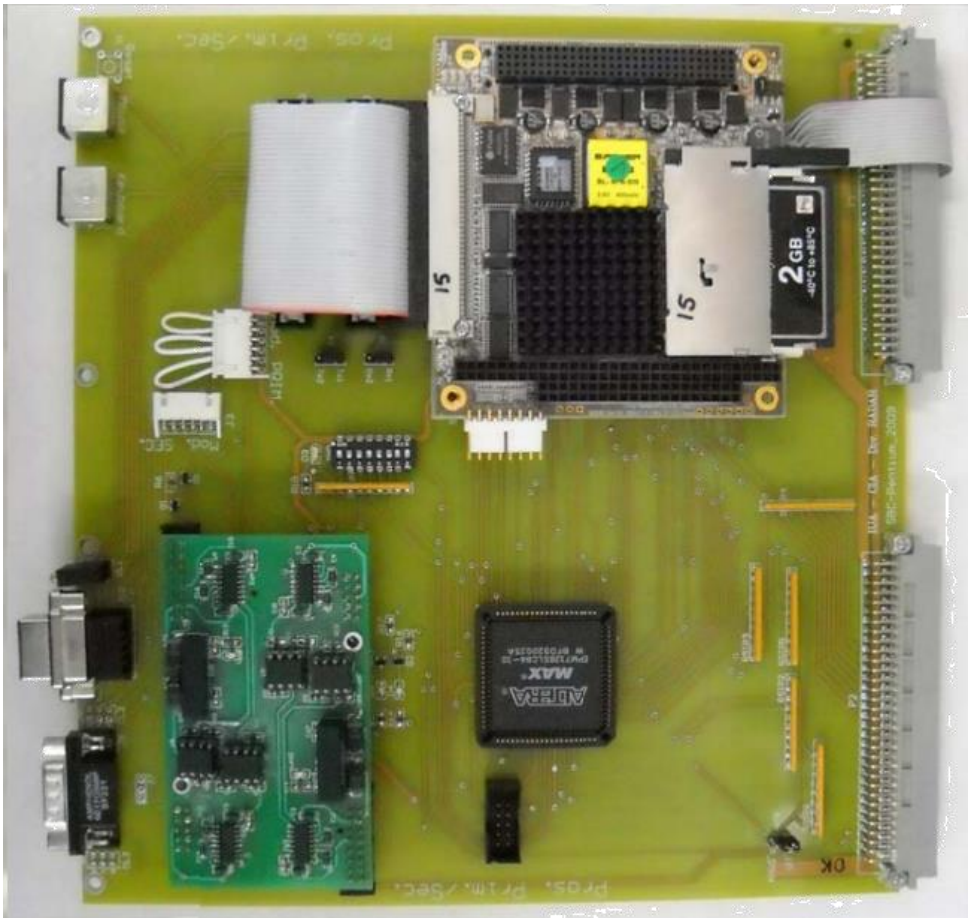


Figura 3.4: Placa base de los procesadores

Estas últimas son las que se conectan al *bus* común a través de los conectores color gris que se aprecian en la figura. Es a través de estas placas que cada SBC se comunica con los demás dispositivos que utiliza el procesador, como ser: memorias FIFO[13], interfaz *Ethernet*, puertos seriales, interfaz GPS, teclado de calibración del panel frontal, fuente de alimentación, etc.

[13] **FIFO**: del inglés **F**irst **I**n, **F**irst **O**ut. Indica el comportamiento de una cola.

Procesadores Primario y Secundario

Figura 3.5: Procesadores Primario y Secundario

Los procesadores Primario y Secundario son unidades de procesamiento que se encargan de digitalizar los datos entregados por el Bloque de Adquisición (Detector).

El Procesador Primario está a cargo de calcular las coordenadas de los centros de los blancos (acimut y rango), asociarlos a un valor de altura y transmitirlos, mediante un puerto serial RS-232, al Combinador. El Procesador Secundario, además de realizar una tarea análoga a la del Procesador Primario, determina los modos de interrogación y los códigos de las aeronaves que detecta para luego, al igual que el Primario, transmitir estos datos hacia el Combinador por puerto serial[14].

Combinador-Tracker

Este subsistema constituye la más compleja de las unidades de procesamiento. El mismo se encarga de efectuar la asociación de plots primarios y secundarios –lo cual implica determinar cuáles corresponden a un mismo blanco–, ubicarlos correctamente en el espacio, correlacionar sus posiciones vuelta a vuelta de radar, empaquetar tanto

[14] Dicha comunicación serial se realiza a través del BUS posterior del EDDR, el cual es compartido por todas las placas que lo conforman.

plots como pistas y transmitir los mismos hacia el exterior. Para esta tarea, en su placa base se incluyen los siguientes componentes:

- 1 SBC
- 1 CPLD
- 1 Interfaz con GPS
- 2 Optoacopladores (RS-232)
- 8 Jumpers para configuración

Así mismo, para tareas de mantenimiento, verificación, ajuste, etc., se incluyen interfaces para salida de video VGA y entradas de teclado y mouse.

A los efectos de ajustar las posiciones de plots primarios y/o secundarios que ingresan a la placa (calibración de posición) tanto en distancia como en acimut, se dispone de un teclado especial en el panel frontal de este módulo; este teclado se interrelaciona con el programa al estilo *online*. A continuación se muestra esta etapa en el E.D.D.R. y un diagrama de bloques de su arquitectura.

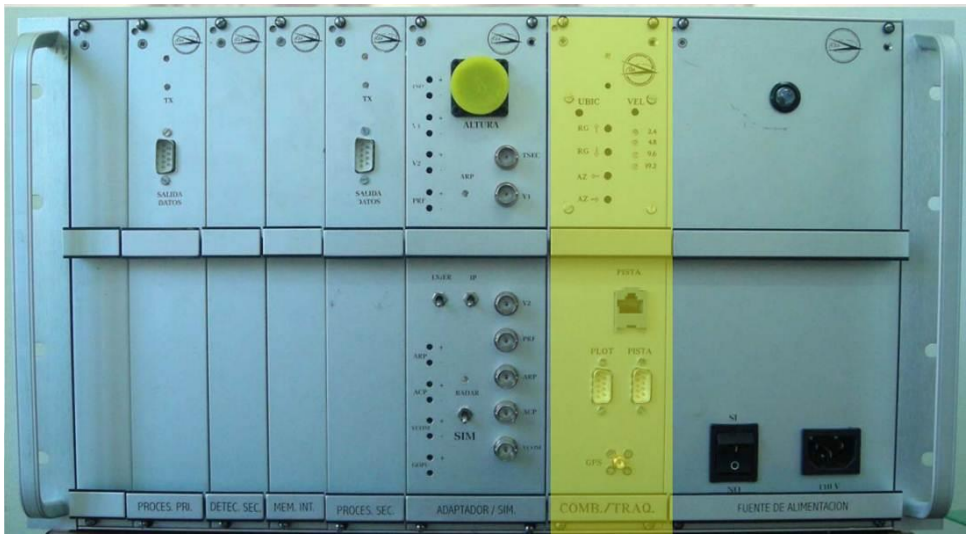


Figura 3.6: Combinador-Tracker

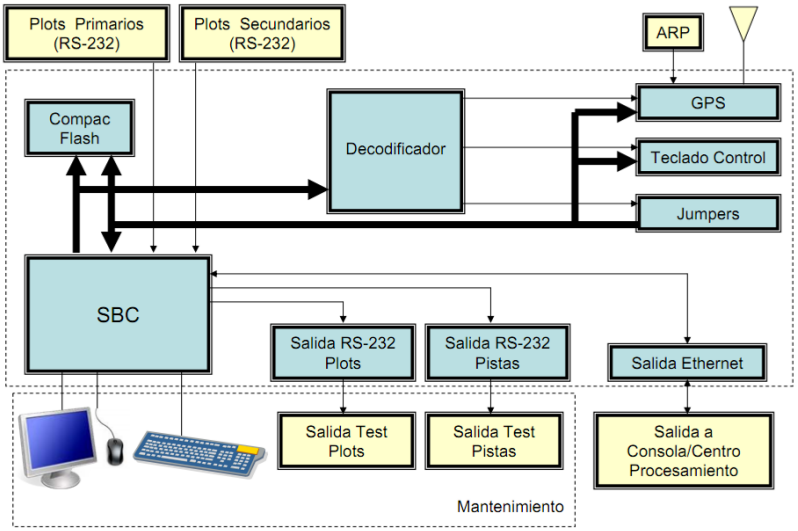


Figura 3.7: Esquema funcional del Combinador-Tracker

Software

La aplicación Combinador se encuentra programada en C, con unas 6000 líneas de código. Tanto la documentación como la estructuración de dicho código es escasa. Se observó una gran cantidad de variables globales, en su mayoría sin nombres descriptivos que permitan vincularlas con algún bloque funcional. También se advirtió bastante acoplamiento con la FIFO de entrada en las llamadas de adquisición de datos primarios y secundarios por puerto serial. El programa hace *polling*[15] sobre esta FIFO para determinar su estado instantáneo.

Adicionalmente, el Combinador hace uso de un archivo de configuración externo para la calibración por teclado del sistema. En este archivo se almacenan los últimos datos que se hayan ajustado mediante el teclado del panel frontal (estos datos se indican como corrimientos -*offsets*- de la posición por defecto del rango y acimut de los plots). En el arranque del sistema, el programa principal lee este archivo y en función de los *offsets* que este indique, ajusta la configuración posicional de todos los plots primarios y secundarios que procese.

[15] Polling hace referencia a una operación de consulta constante, hacia un dispositivo de hardware o software, para crear una actividad sincrónica sin el uso de interrupciones.

Básicamente, el programa lee los plots provenientes de los procesadores –primario y secundario– mediante interrupciones. Una vez finalizada la recepción de datos de plots de un sector, el programa busca las posibles asociaciones en base a la distancia euclideana que existe entre ellos; es decir, la asociación es por proximidad entre los plots. Una vez completado el análisis de todos los plots del sector, los envía a un buffer para su transmisión. Asociadas a estas rutinas básicas, existen otras como ser las de chequeo del teclado, determinación de errores, etc. El siguiente diagrama de flujo resume su funcionamiento.

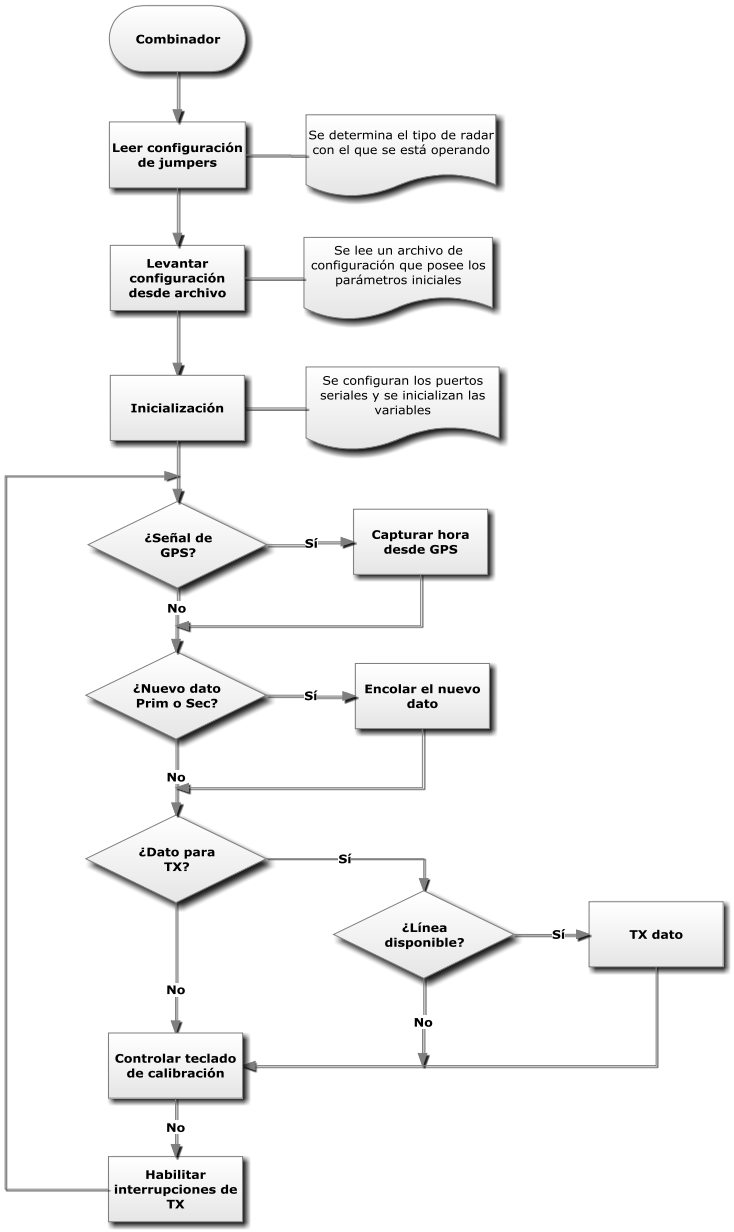


Figura 3.8: Diagrama de flujo del Combinador

Como conclusión de este relevamiento de *software*, puede decirse que la característica más relevante en el Combinador original es el elevado acoplamiento de sus algoritmos. Programas con este nivel de acoplamiento no resultan muy flexibles ante determinados acontecimientos. En otras palabras, la posibilidad del sistema de entregar datos coherentes ante una eventual variación del flujo normal o esperado del programa, está sujeta a la ventana de tiempo disponible para procesar esos datos desde que son adquiridos. Por ejemplo, si se aumentara considerablemente la tasa de adquisición o envío de datos del E.D.D.R., podría darse el caso en que las FIFO de adquisición se llenaran y las unidades de procesamiento, al detectar esta situación, debieran vaciarlas constantemente, derivando esto en un rendimiento no aceptable del equipo.

Por otro lado, los algoritmos basados en *polling* son muy ineficientes en cuanto al uso de la CPU. Esta no es una característica deseada en sistemas multitarea como los que deberán implementarse si lo que se pretende es integrar Combinador y Traqueador en una misma unidad de ejecución.

Adicionalmente, se observó que la documentación del *software* resulta muy escasa y su estructuración, también. Esto puede influir en el funcionar defectuoso del sistema actual. Las fallas que se aprecian por el personal de FAA respecto a esta implementación original van desde transmisión de plots falsos y omisión de plots verdaderos, hasta paquetes con información corrompida o aeronaves duplicadas.

CAPÍTULO IV

Sistema Operativo

En vistas de los requerimientos impuestos por FAA, sólo se consideró implementar en el E.D.D.R., sistemas basados en UNIX –más precisamente, GNU/Linux. Las principales razones para tal requisito pueden resumirse en:

- **Costo de adquisición:** se ofrecen muchas versiones gratuitas.

- **Licencia GPL:** posibilidad de modificación/configuración del Sistema Operativo bajo demanda.

- **Multitarea y multihilo:** aporta flexibilidad al sistema, permitiendo la ejecución de múltiples procesos concurrentemente.

- **Bajos requerimientos de hardware:** puede funcionar en la arquitectura de las SBC sin necesidad de actualizar ningún componente.

- **Lenguaje C nativo:** al igual que las aplicaciones del E.D.D.R., GNU/Linux se encuentra programado mayoritariamente en C e incluye un compilador (GCC) para desarrollo en cada distribución.

- **Compatibilidad:** al tratarse, la SBC, de una placa PC compatible, la mayoría de los controladores de cada dispositivo son ampliamente soportados.

- **Estabilidad:** esta plataforma ha demostrado ser sumamente confiable en cuanto a tiempo de funcionamiento (*uptime*); característica necesaria para equipos como el E.D.D.R., donde el mantenimiento del sistema es esporádico y el tiempo de ejecución continua predomina.

Luego de una investigación sobre distintas distribuciones posibles de implementar, se optó por implementar una versión de Debian GNU/Linux[16]. La justificación de esta elección tiene origen en varios aspectos:

1- Se trata de una distribución sumamente reconocida y utilizada a nivel mundial, en gran medida, debido a su robustez.

2- Es fácilmente configurable para ser adaptada a la SBC del E.D.D.R. y su limitada memoria flash.

3- Permite el desarrollo en un entorno de escritorio standard y la migración de los archivos ejecutables sin necesidad de realizar compilaciones cruzadas.

4- Al contarse con experiencia de trabajo sobre esta plataforma, se ahorra mucho tiempo de configuración de la misma y se aborda más rápidamente la verdadera problemática a resolver.

Si bien para la elección del Sistema Operativo se investigó sobre alternativas de tiempo real (RTEMS) (RTAI), no se consideró imprescindible una implementación de este tipo ya que, para los recursos de hardware con los que cuenta la SBC del E.D.D.R. y la tasa de adquisición de datos -115200 bps como máximo-, **no hay restricciones de tiempo tales que justifiquen el uso de estos sistemas**.

[16] http://www.debian.org/ (20-08-2014)

Instalación y configuración

Para la implementación del Sistema Operativo se escogió GNU/Linux Debian Lenny (kernel 2.6.26). Esta distribución es muy apropiada para los recursos de hardware de la SBC a utilizar, ya que una instalación recortada de la misma es lo suficientemente liviana para ejecutarse sin problemas en el E.D.D.R. y permite, al mismo tiempo, fácil configuración y adaptación a las restricciones de funcionamiento que posee – principalmente boot-time[17], non-journaled FS[18].

El proceso de implementación del Sistema Operativo consta de dos etapas, fundamentalmente: la instalación del sistema base y su apropiada configuración. Dicho proceso se realiza inicialmente sobre un kit de desarrollo, provisto por FAA, como el de la figura 4.1[19]. Luego, se copia una imagen del disco del kit de desarrollo a la compact-flash de la SBC, evitando así gran cantidad de escrituras en la tarjeta de memoria innecesariamente.

Figura 4.1: Kit de desarrollo utilizado

[17] Tiempo de inicio o Boot-time, en inglés, es el tiempo necesario para que un sistema arranque y quede completamente operativo. En el caso del E.D.D.R., el mismo no debía superar los cuarenta (40) segundos.

[18] Non-journaled File Systems, son los sistemas de archivos utilizados en conjunto con el almacenamiento de tipo flash, ya que minimiza las escrituras en memoria, dando así mejor rendimiento.

[19] Kit compuesto de un disco rígido, una fuente, una lecto-grabadora de DVD y un sócalo en donde se conecta la SBC sobre la que se pretende desarrollar.

La lista de tareas realizadas para la correcta instalación y configuración del Sistema Operativo se encuentra adjunta, al final del trabajo, como ANEXO II.

Con estos pasos se logró dejar el Sistema Operativo instalado en la compact-flash y a la espera de la aplicación de software del Combinador.

CAPÍTULO V

El standard ASTERIX

ASTERIX[20] (**A**ll-purpose **ST**ructured **E**urocontrol Su**R**veillance **I**nformation E**X**-change) es un protocolo para la conformación de mensajes de comunicación entre sistemas de control de tráfico aéreo –sistemas de radar y aeronaves, principalmente. Fue creado a fines de la década del '80 por la Organización Europea para Seguridad en Navegación Aérea o EUROCONTROL y adoptado mundialmente para la estandarización de los mensajes de comunicación en aviación, tanto civil como militar.

ASTERIX ofrece 256 categorías que se utilizan, según la aplicación, para definir la estructura de los mensajes que serán transmitidos. Por ejemplo, la información que transmite un sistema mono-radar primario hacia los centros de operación no es la misma que la que transmite un radar de barrido de superficie o uno de multilateración. Por ello, los mensajes emitidos/recibidos por estos sistemas son de distinta categoría.

En vistas de lo anterior, cabe destacar que son de interés en el presente trabajo las categorías 34 y 48. Estas son la evolución de las categorías 1 y 2 para transmisión de mensajes de servicio y reportes de blancos, respectivamente, desde un Sistema de Radar de Vigilancia convencional hacia uno o más Sistemas de Procesamiento de Datos de Vigilancia –en nuestro caso, los segundos son el E.D.D.R. y los centros de vigilancia de FAA[21]. En otras palabras, la categoría 34 sirve para informar acerca del estado actual del sistema de radar; sus mensajes pueden indicar cuatro tipo de eventos o situaciones:

- **Top Norte:** cruce de la antena de radar por el norte geográfico en acimut.
- **Cambio de sector:** cada vez que la antena de radar cruza uno de los 32 ángulos límites en los que se divide la vuelta completa de escaneo.

[20] https://www.eurocontrol.int/asterix (15-06-2014)

[21] Para mayor información, consultar https://www.eurocontrol.int/asterix, pestaña "*How do I choose?*".

- **Filtrado geográfico:** para indicar que determinados plots deben ser filtrados por su ubicación geográfica (cuando se usa mapa de *clutter*, por ejemplo)
- **Presencia de interferencia:** también denominado *jamming strobe.* Mediante estos mensajes se indica que se ha detectado que el Sistema de Radar está siendo interferido electrónicamente.

La categoría 48, en cambio, comprende los mensajes de detección de blancos; esto es, la información medida por el radar. Sus mensajes pueden corresponder a plots o tracks según se requiera.

Independientemente de la categoría, cada mensaje ASTERIX intercambiado constituye un bloque de datos. El mismo se compone de tres partes, fundamentalmente: un indicador de la categoría utilizada (CAT), un indicador del largo total, en bytes, del mensaje (LEN) y un registro que contiene la información propiamente dicha de la categoría empleada. Un registro, a su vez, se compone de campos (coordenadas espaciales, hora del mensaje, código de aeronave, entre otros) y estos campos, a su vez, de ítems (los valores numéricos que conforman a los campos). La unidad más pequeña de información es el Byte (comúnmente expresada como octeto) y es por esto que la longitud de cada campo incluido en un bloque de datos es múltiplo entero de un Byte.

La estructuración de cada bloque de datos queda definida, entonces, como se muestra en la figura 5.1.

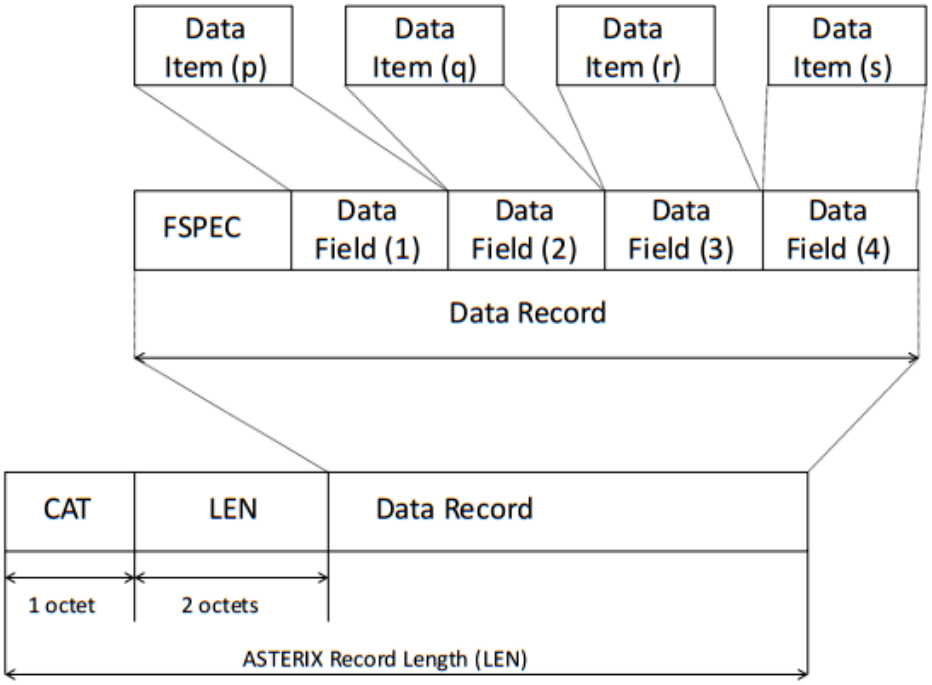


Figura 5.1: Bloque de datos en ASTERIX

El campo FSPEC (Field Specification) indica qué campos de la categoría correspondiente se incluyen en un registro determinado. Dicha indicación se hace mediante 1's o 0's, denotando presencia o no, repectivamente de dichos campos. Por ejemplo, en un mensaje de servicio típico (CAT 34), los campos posibles son:

1 I034/010 Data Source Identifier
2 I034/000 Message Type
3 I034/030 Time-of-Day
4 I034/020 Sector Number
5 I034/041 Antenna Rotation Period
6 I034/050 System Configuration and Status
7 I034/060 System Processing Mode
FX N/A Field Extension Indicator

Si uno quisiera indicar que el mensaje a enviar incluye los campos 1, 2, 3 y 5 únicamente, FSPEC sería de la forma:

	CAMPO1	CAMPO2	CAMPO3	CAMPO4	CAMPO5	CAMPO6	CAMPO7	CAMPO8
FSPEC	1	1	1	0	1	0	0	0
FSPEC	**0xE8**							

Figura 5.2: Estructura del campo FSPEC

Y el bloque de datos completo, para un mensaje de este tipo tendría la siguiente forma:

CAT	LEN		FSPEC	SAC	SIC	MT	TIME			REV	
0x22	0x00	0x0C	0xE8	?	?	0x01	?	?	?	?	?

Figura 5.3: Un bloque de datos completo

En la Figura 5.3 puede observarse que, la categoría del bloque es la 34 (0x22 en hexadecimal), su longitud total es de 12 octetos (0x0C en hexadecimal), FSPEC es similar a la del ejemplo anterior, se proveen identificadores del sistema generador del mensaje (SAC y SIC), el tipo de mensaje (0x01 indica que se trata de un mensaje de cruce de antena por el norte magnético), la marca de tiempo y el período de revolución de la antena de radar (expresado en milisegundos) (Eurocontrol A. , CAT034 - ASTERIX - Monoradar Service Messages (Part 2b - next version of Cat 002)).

Similarmente se conforman los mensajes de la categoría 48 (Eurocontrol A. , CAT048 - ASTERIX - Monoradar Target Reports (Part 4 - next version of Cat 001)), sólo que los mismos exhiben la información relativa a los blancos detectados por el sistema de radar y no al estado actual del mismo, como en la categoría mencionada anteriormente.

Teniendo esto en cuenta y habiendo estudiado, tanto las especificaciones del protocolo como las especificaciones de FAA para la conformación de mensajes requerida, se continuó con el siguiente paso del proyecto: el desarrollo del Combinador.

CAPÍTULO VI

El Combinador

En este capítulo se muestra el proceso de rediseño de la aplicación Combinador original. En la primera sección se parte de un diagrama funcional del mismo para comprender cuáles son las tareas específicas que se le exigen. En la segunda sección se detalla cómo se llevan a cabo las mismas en la nueva implementación. Finalmente, en la tercera sección se dan algunas consideraciones a tener en cuenta sobre el ciclo de funcionamiento del Combinador y algunos escenarios posibles.

A partir de este punto del trabajo, se hace necesario el uso de terminología más técnica y específica de las tecnologías de *software*; sin embargo, cada concepto será apropiadamente presentado a medida que se lo vaya requiriendo.

Las modificaciones realizadas sobre el Combinador fueron pensadas en conjunto con personal de FAA, en base a sus requerimientos. Dichos requerimientos de funcionamiento se expresan en la siguiente sección.

Su función

Idealmente, la función específica del Combinador dentro del E.D.D.R., es la de recibir la información proveniente de los Procesadores Primario y Secundario, sincronizarla, procesarla y retransmitirla junto a otros datos de control generados por el propio Combinador: los cruces de norte geográfico, o Top Norte.

La información de entrada llega al Combinador a través de dos puertos seriales (uno por cada Procesador) y en un protocolo propio de FAA. Así, la primera tarea importante del Combinador es decodificar esta información y extraer la ubicación espacial de cada plot que se va recibiendo. A continuación, se evalúan todos los plots primarios y secundarios de un mismo sector y se los asocia –si se encuentran lo suficientemente

próximos, en base a su distancia euclidiana- para indicar que corresponden a una misma aeronave, dando lugar así a la generación de plots **combinados**. En el mismo proceso se eliminan aquellos plots que se encuentren repetidos, que presenten valores atípicos o que, por defectos del sistema de radar, hayan sido interpretados como correspondientes a sectores equivocados. Este filtrado de información mejora considerablemente la calidad de las mediciones finales del Sistema de Radar, ya que se eliminan importantes fuentes de error.

Una vez discriminados todos los plots de un sector, se realiza una adaptación de escala de las coordenadas espaciales originales para compatibilizar con la norma del protocolo ASTERIX. Esto último es necesario porque el sistema de radar en cuestión, en cuanto a tecnología, es bastante antiguo y sus resoluciones en rango, acimut y altura son menores a las que manejan sistemas semejantes de la actualidad y sugiere el standard.

A continuación, se completan los paquetes de datos ASTERIX para cada plot, con los campos que FAA requiere y se los transmite: por puerto serial –para monitoreo local- y por Ethernet, hacia la Intranet de FAA. Simultáneamente, estos paquetes son copiados en un *buffer* en memoria RAM que es accedido por el Rastreador para obtener su información de entrada.

Adicionalmente, el Combinador debe ser capaz de detectar e indicar cada momento en el que la antena de radar cruza el norte geográfico. Esta información es brindada por el Sistema de Radar, a través de una señal analógica denominada Pulso de Referencia en Acimut o ARP (**A**zimuth **R**eference **P**ulse, en inglés) que se envía hacia el Combinador cuando este cruce ocurre. Dicha señal es capturada por el PIC presente en la placa base del Combinador y comunicada al sistema de GPS asociado para que el mismo brinde la hora exacta del evento. Tras generarse el pedido de hora, el Combinador crea un mensaje de Top Norte, con esa marca de tiempo y en protocolo ASTERIX, que se transmite inmediatamente.

En paralelo a todo este proceso, el Combinador debe ser capaz de atender comandos ingresados por el operario mediante el panel frontal de ajuste de video o bien, nuevos parámetros de funcionamiento a través de una conexión de red[22]. Entre estos últimos,

[22] La misma se realiza utilizando SSH (*Secure Shell*) y modificando un archivo de configuración dedicado a tal fin.

pueden mencionarse características como: dirección IP del Combinador, velocidad de transmisión de los puertos de comunicación serial o rangos máximos y mínimos de escaneo.

El siguiente diagrama resume los requerimientos de funcionamiento planteados por FAA, en base a toda la lógica anteriormente descrita.

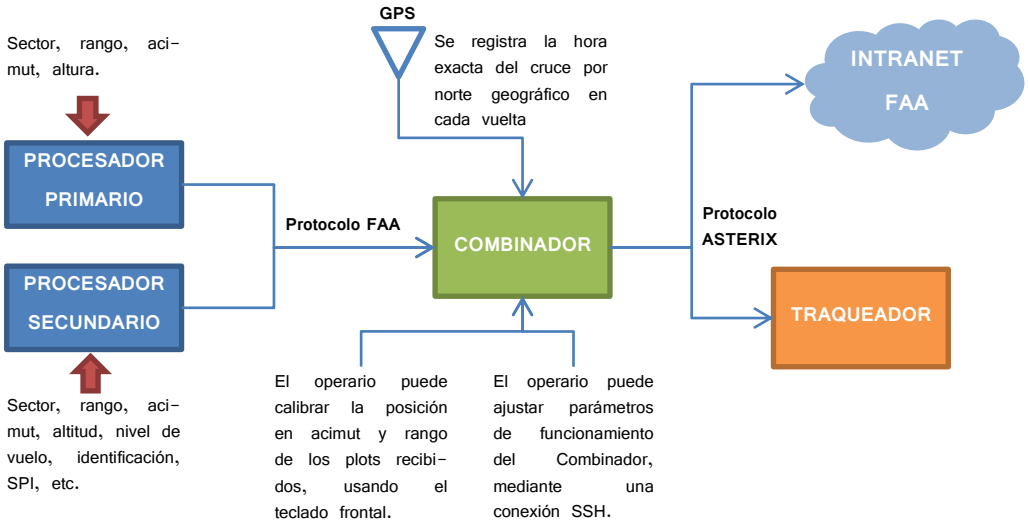


Figura 6.1: Funcionamiento del Combinador

Implementación

Como ya se destacó en el Capítulo III, la implementación original del Combinador consta de un único programa en donde se incluye toda la lógica del mismo. Dicha estructuración resulta muy inconveniente al momento de depurar, mejorar o mantener el código. Por esta razón, en la nueva implementación se optó por dividir la funcionalidad del mismo en distintas entidades; cada una de ellas con una responsabilidad específica asociada.

Esta metodología, comúnmente utilizada en informática, se denomina Programación Orientada a Objetos[23] (P.O.O) y, bajo este paradigma, las entidades mencionadas anteriormente se denominan **Objetos**.

Los objetos son instancias de ciertas plantillas (o modelos) denominadas **Clases**. A su vez, una Clase es una definición formal de características (atributos) y funciones (métodos) que toda instancia de esa clase posee. Salvando las distancias, puede verse a una Clase como una "receta" sobre cómo crear un Objeto.

Para reconocer qué objetos intervienen en el algoritmo del Combinador se estudió tanto el diagrama de flujos en la Figura 3.8 como el diagrama de requerimientos de la Figura 6.1. El primero, por conveniencia, se presenta nuevamente a continuación.

[23] La Programación Orientada a Objetos es un paradigma de programación que apunta a desarrollar software enfocado en estructuras de datos llamadas Objetos, considerando a estos, representaciones abstractas pero intuitivas de las entidades intervinientes en los problemas cotidianos a resolver en la programación de software.

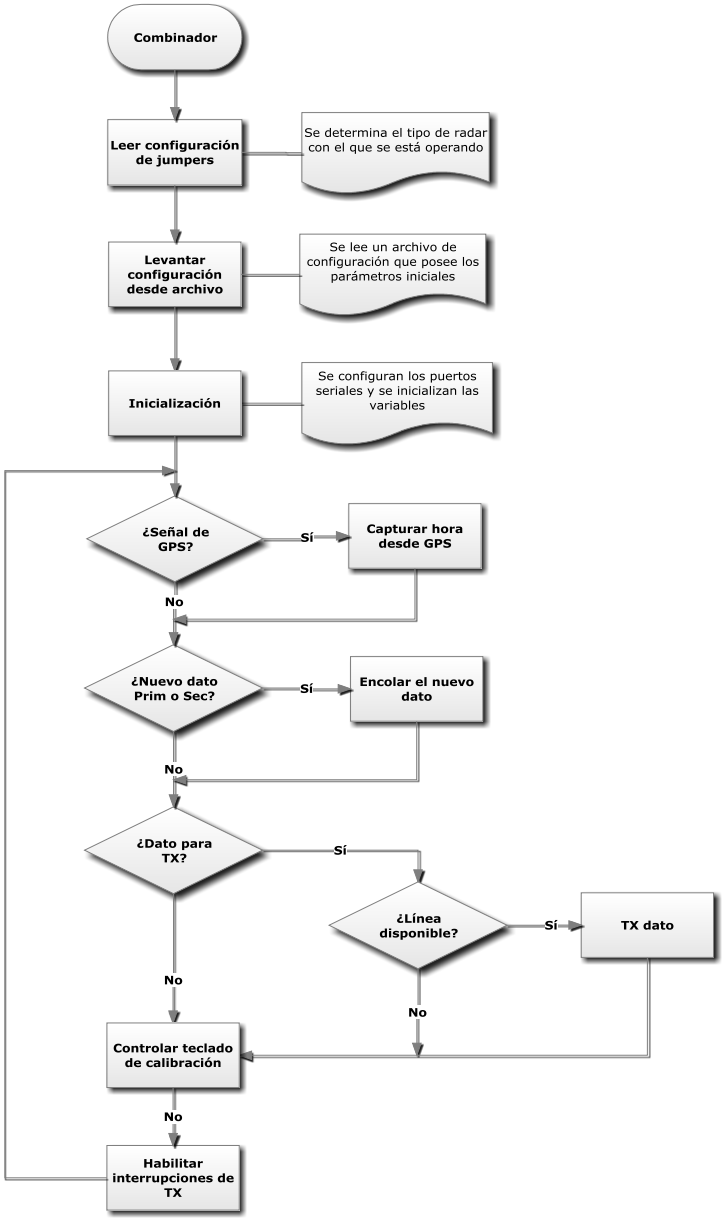


Figura 6.2: Diagrama de flujo del Combinador

La Figura 6.1 determina las interfaces y responsabilidades con las que debe cumplir cada objeto del Combinador, en otras palabras, habla de "¿Qué se debe hacer?". La Figura 6.2, en cambio, da una idea del "¿Cómo hacerlo?", ya que pone en evidencia el conjunto de tareas específicas a realizar durante el ciclo de trabajo del Combinador, constituyendo un puntapié inicial para el algoritmo de la nueva implementación.

Para el presente trabajo se detectaron seis objetos, de seis clases diferentes. Los mismos se muestran a continuación en la Figura 6.3 y se describen más abajo, junto a su respectivo diagrama de clases.

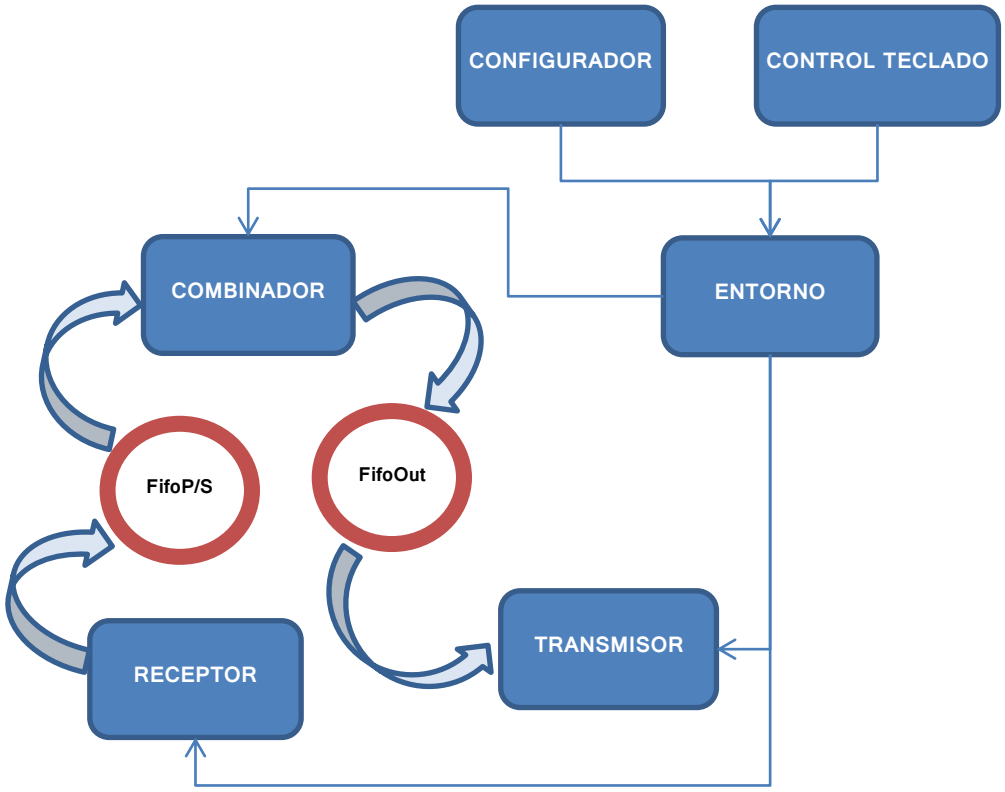


Figura 6.3: Diagrama de Objetos del Combinador

El objeto **ControlTeclado** es un hilo que se encarga de atender las peticiones de ajuste de posición, ingresadas por el operario del radar, mediante el teclado frontal del E.D.D.R. Este tipo de ajuste se requiere, como solución provisoria, cuando existe una calibración inapropiada del sistema de radar. Bajo este escenario, las mediciones de coordenadas de los blancos detectados por los radares Primario y Secundario, difieren notablemente y se hace necesario introducir ciertos *offsets*[24] de posicionamiento de los plots para acondicionar las mediciones.

La forma en la que se opera cuando se requiere un ajuste de posición es la siguiente:

[24] Desplazamiento u *offset*, en inglés: corrimientos a partir de un punto de referencia.

- Si el operario presiona durante 5 segundos la tecla UBIC del panel frontal, el sistema entra en modo de configuración de posición de plots primarios. Se encienden todos los LEDs del panel para indicar este modo y se queda a la espera de los ajustes en rango y/o acimut deseados. Luego de 10 segundos de no registrarse ajustes la rutina sale del modo de configuración y se graban en un archivo los parámetros modificados.

- Si el operario presiona durante 5 segundos las teclas UBIC y RANGO ↑ del panel frontal simultáneamente, el sistema entra en modo de configuración de posición de plots secundarios. Se encienden todos los LEDs del panel para indicar esta situación y se queda a la espera de los ajustes en rango y/o acimut deseados. Al igual que en el caso anterior, luego de 10 segundos de no registrarse ajustes la rutina sale del modo de configuración y se graban en un archivo los parámetros modificados.

Cabe destacar que, el objeto **ControlTeclado** sólo trabaja durante la rutina de configuración de posición y colabora con el objeto Entorno al registrar los cambios realizados. Esto significa que, durante la mayor parte del ciclo de vida de la aplicación del Combinador, este hilo duerme[25].

El objeto **Configurador** es otro hilo encargado de levantar, desde un archivo de texto, los parámetros iniciales del sistema radar en cada arranque del E.D.D.R. Entre estos parámetros se incluye a los ajustes de posicionamiento mencionados anteriormente, la velocidad de los puertos seriales, el IP del sistema de radar, entre otros. Es responsabilidad de este objeto la inicialización de los parámetros esenciales del objeto Entorno.

El ciclo de vida de este objeto está limitado al arranque del E.D.D.R., ya que es a partir de allí que se mantiene la configuración del Sistema de Radar hasta un eventual reinicio –algo que no debería ocurrir casi nunca, idealmente.
Las modificaciones al archivo de texto de configuración se realizan a través de una conexión de red segura y siempre a cargo de personal autorizado de FAA. Cada vez que

[25] En informática, un objeto o proceso que duerme es aquel que no ocupa tiempo de CPU para ninguna tarea. Debería ser el estado por defecto de cualquier proceso u objeto cuando se encuentra ocioso.

se realizan cambios en este archivo de configuración debe reiniciarse el E.D.D.R. a modo de que los mismos surtan efecto.

El objeto **Entorno** es un *singleton*[26] que representa al sistema de radar en general. Este concentra todos sus parámetros de funcionamiento, como ser: códigos de identificación del sistema -SAC y SIC[27]-, su dirección IP dentro de la Intranet de FAA, los rangos máximos y mínimos de vigilancia, la ventana de asociación[28], la velocidad de los puertos serie, los *buffers* de entrada y salida y algunas constantes o coeficientes de conversión importantes, entre otros. Este objeto colabora con el objeto Configurador en la inicialización del E.D.D.R. y es a quien acude el resto de los objetos cuando necesitan acceder a algún dato global del sistema durante el funcionamiento en régimen.

Los *buffers* del sistema son tres y se denominan: **fifoP**, **fifoS** y **fifoOut**. Los primeros dos son utilizados por los objetos Receptor y Combinador, en el proceso de adquisición de datos Primarios y Secundarios respectivamente; el tercero es compartido entre el objeto Combinador y Transmisor, en la instancia de salida de datos ASTERIX hacia el exterior.

Estos *buffers* se han implementado como vectores FIFO, concurrentes[29], que se acceden con exclusión mutua. Los mismos funcionan en espacio de usuario.

En su desempeño, estas FIFO son similares a las FIFO de GNU/Linux, sólo que agregan algunas funcionalidades necesarias para el manejo de ciertas situaciones atípicas que pueden darse en el Combinador. Entre ellas, nos referimos a eventos en donde la información puede quedar almacenada de manera inconsistente –ante cortes abruptos en la transmisión de datos desde los Procesadores, por ejemplo.
El objeto **Receptor** atiende los puertos seriales de donde se obtiene la información de entrada para el Combinador –plots primarios y secundarios. Su única función es ir en-

[26] Es un patrón de diseño de software utilizado cuando se requiere una única instancia de una clase, de acceso global.

[27] System Area Code (SAC) y System Identification Code (SIC) son, respectivamente, los códigos de area o país donde se encuentra el Sistema Radar y su identificación puntual dentro de esa área o país.

[28] Indica cuán próximo debe estar un blanco primario de un secundario para ser considerados una misma aeronave y así, combinados.

[29] Esto significa que pueden ser accedidos por más de un proceso simultáneamente.

colando los plots recibidos en los dos *buffers* fifoP y fifoS, para los plots primarios y secundarios respectivamente.

La importancia del objeto Receptor radica en el hecho de que su presencia desacopla el proceso de adquisición de datos del proceso de tratamiento de los mismos, lo cual hace a la aplicación más modular y, por ende, versátil. En otras palabras, el hecho de tener un objeto Receptor dedicado facilita la adaptación posterior de toda la aplicación a nuevos tipos de datos de entrada, si eventualmente así se requiriera.

El objeto **Transmisor** accede al *buffer* de salida de datos, fifoOut, cada vez que hay un plot ASTERIX listo para ser enviado. Cada plot se transmite, tanto por puerto serial –para el monitoreo en la consola local- como a través de un socket UDP, hacia la Intranet de FAA.

Los objetos Receptor y Transmisor, al igual que ControlTeclado y Configurador, corren en hilos separados al flujo de programa principal.

Finalmente, el objeto **Combinador** se ocupa de, en base a los parámetros de entorno fijados, leer los plots primarios y secundarios que hubiera en los *buffers* de adquisición, ordenar aquellos que pertenecen a un mismo sector y extraer sus coordenadas espaciales. Luego, en base a la distancia euclidiana que exista entre primarios y secundarios, busca asociaciones para elaborar plots combinados. Finalmente, elimina aquellos plots inconsistentes y empaqueta los plots válidos en protocolo ASTERIX para almacenarlos en el *buffer* de salida, fifoOut.

El lenguaje elegido para esta implementación es C++ debido, en gran medida, a su mejor *performance* respecto de otros lenguajes de más alto nivel, pero también a que se cuenta con experiencia de trabajo sobre este. A continuación se incluyen los diagramas de clases y de secuencia asociados al Combinador.

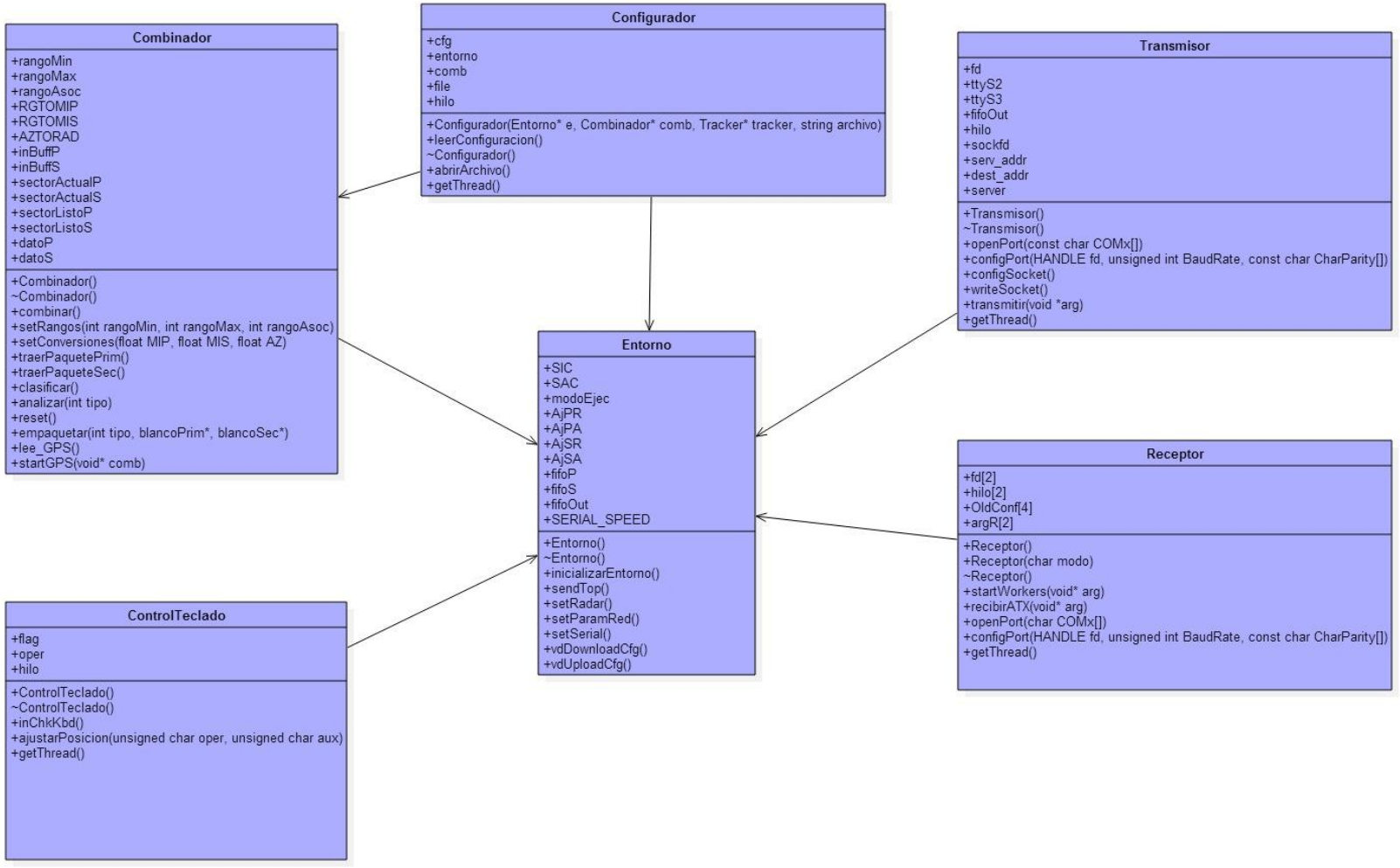


Figura 6.4: Diagrama de Clases de la aplicación Combinador

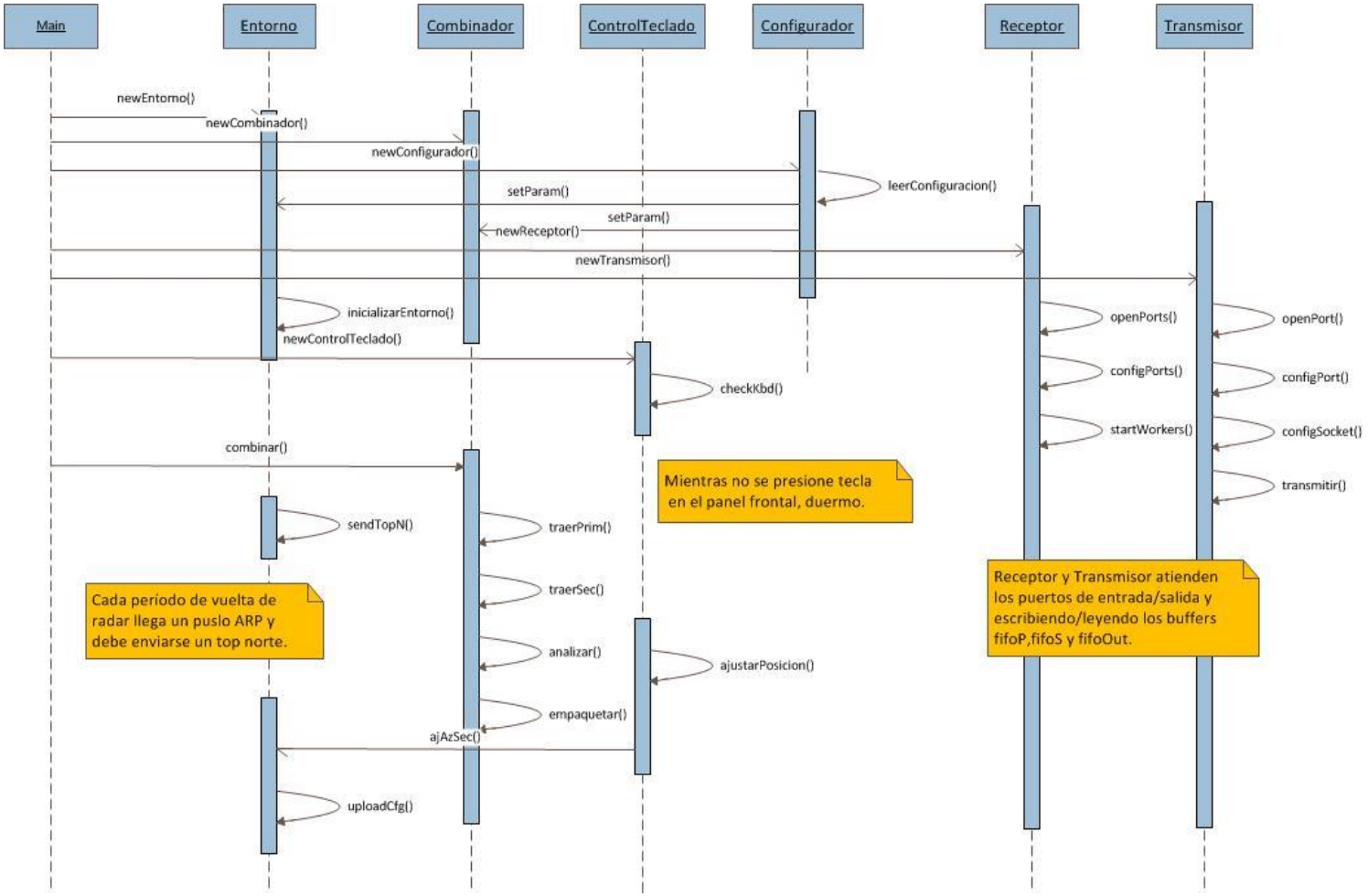


Figura 6.5: Diagrama de Secuencia de la aplicación Combinador

Consideraciones

En el diagrama anterior se muestra el ciclo de vida típico del Combinador durante funcionamiento normal, el cual puede validarse fácilmente cuando el E.D.D.R. se encuentra en modo de simulador. Sin embargo, al operar sobre el sistema de radar real, puede ocurrir una serie de situaciones excepcionales que alteran el flujo normal del programa y deben ser tenidas en cuenta para lograr que el sistema sea tolerante a fallas y continúe operando y entregando información válida.

Por ejemplo: se ha observado que, eventualmente, alguno de los procesadores deja de funcionar momentáneamente o por un tiempo considerable (se apaga). Cuando esto ocurre, el Combinador indica esta anomalía por pantalla y continua trabajando con la información proveniente del procesador que se encuentra en operación. Es evidente que, ante este escenario, los plots de salida del Combinador no son plots combinados sino primarios o secundarios según qué procesador continúe en funcionamiento. En el caso en que ambos procesadores se encuentren apagados, pero la antena de radar continúe rotando, el envío de señales de top norte se mantiene intacto.

Todos estos escenarios se encuentran detallados en la sección a continuación.

CAPÍTULO VII

Validación y Documentación

Este capítulo se encuentra dividido en dos secciones. La primera muestra cuáles fueron los casos de uso que se utilizaron para validar el funcionamiento del Combinador desarrollado. Se incluye aquí tanto a los ensayos en condiciones normales como aquellos realizados para poner a prueba la tolerancia a fallos del sistema ante una anomalía.

En la segunda sección se detalla el proceso de documentación de todo el proyecto y se muestra, a modo de ejemplo, parte de la documentación terminada.

Casos de Uso

Para constatar el correcto funcionamiento del Combinador desarrollado, se diseñó una serie de casos de uso. Los mismos se categorizan en tarjetas de dos tipos: normal y alternativo. Las primeras se utilizan para validar el régimen normal de funcionamiento del Combinador; en pocas palabras, se constata que el sistema enciende correctamente y, de recibir datos coherentes de entrada, entrega datos acordes de salida. Las tarjetas de casos de uso del segundo tipo simulan situaciones anómalas, por lo general asociadas a fallas en el sistema de radar. Como ejemplo de estas puede mencionarse a un corte en la transmisión de plots primarios o secundarios por parte de los procesadores o un corte de energía general.

Las aplicaciones de *software* que se utilizaron para realizar estas pruebas son, básicamente, dos: Wireshark[30] y una consola virtual propia de FAA que permite visualizar plots primarios, secundarios y combinados que se encuentren en formato ASTERIX. Mediante el uso de Wireshark se puede comprobar si el Combinador envía efectivamente los paquetes ASTERIX y, al mismo tiempo, verificar los valores de los campos de estos paquetes individualmente. La consola virtual de FAA permite tener una repre-

[30] https://www.wireshark.org/

sentación visual de estos plots para saber si los mismos son direccionados correctamente hacia las distintas posiciones en el plano.

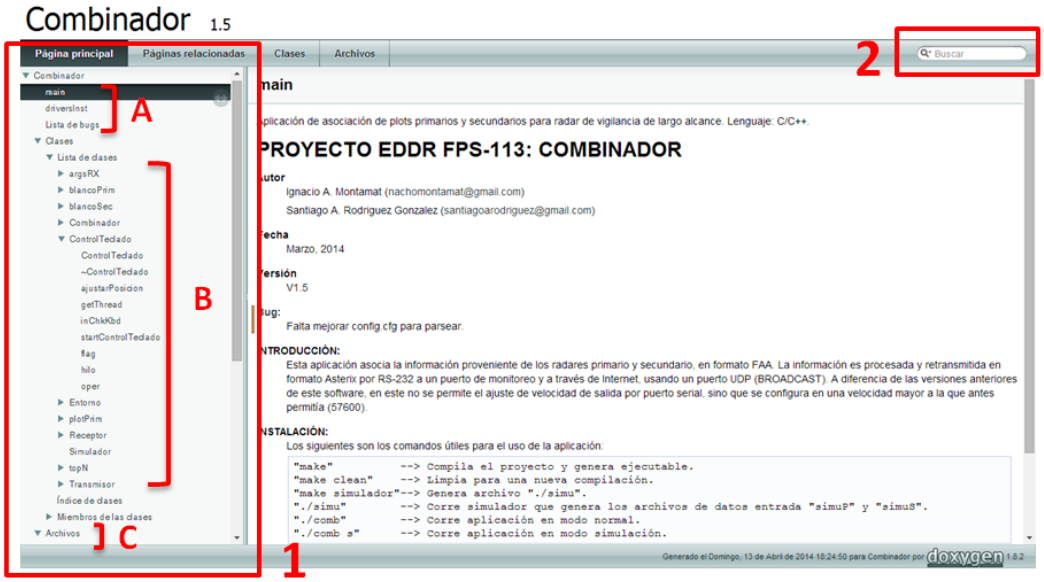
Para consultar estas tarjetas de casos de uso, referirse al ANEXO I de este documento, donde se encuentran adjuntas.

Documentación

La documentación[31] de este proyecto se encuentra en formato de página web para ser vista en cualquier explorador de Internet (Chrome, Firefox, IE, etc.). La misma se realizó de la mano de la herramienta *doxygen* y se organiza como se muestra a continuación.

Cabe destacar que, además de la documentación del Combinador propiamente dicha, se confeccionaron ciertos manuales de operación que, si bien no se incluyen en este trabajo, pueden ser consultados previa autorización.

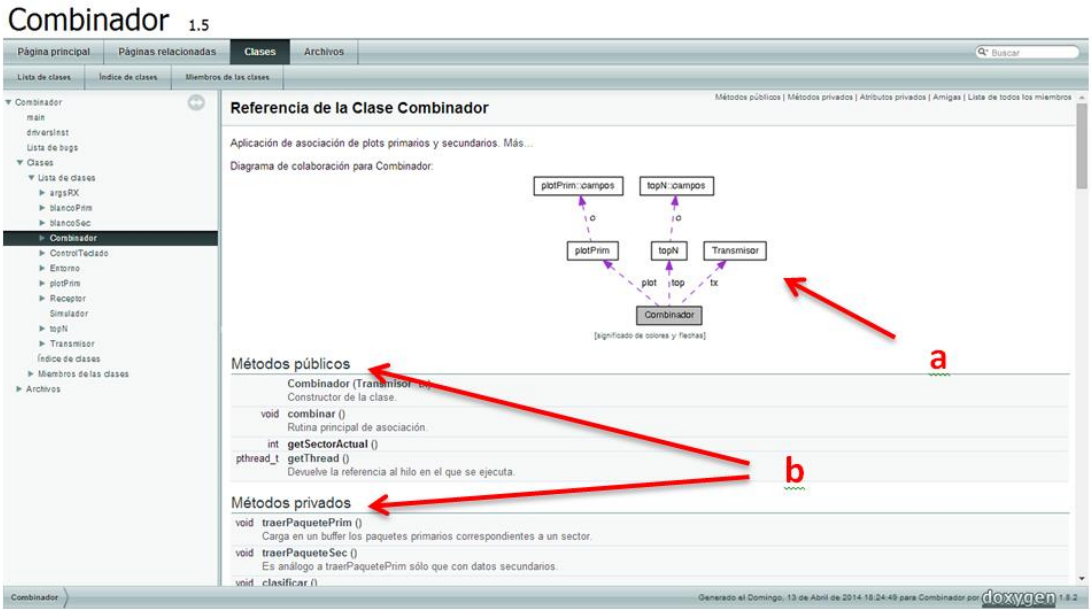
La página web de documentación está constituida de la siguiente forma:



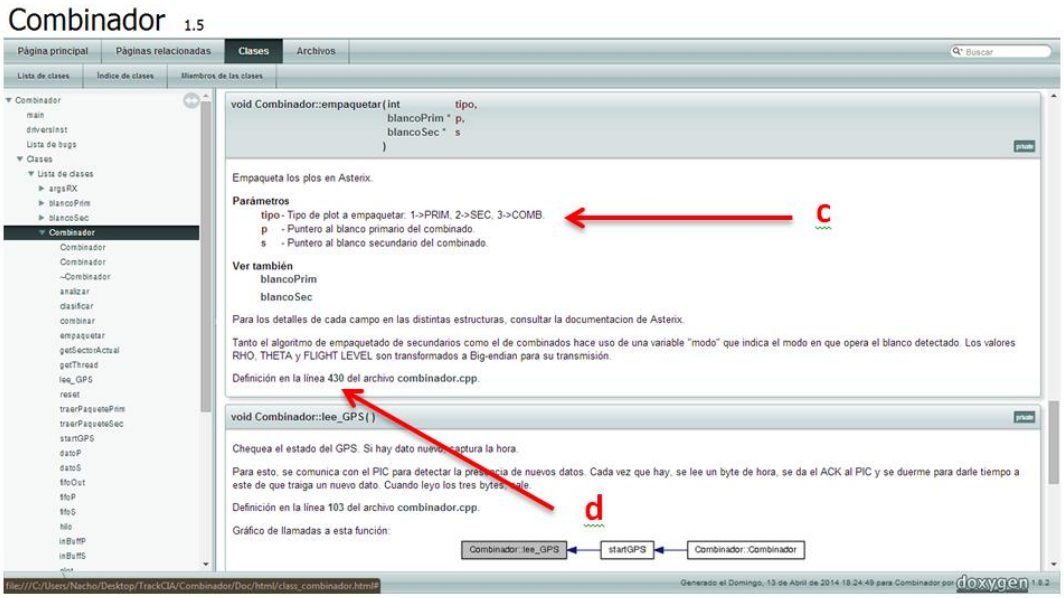
[31] Si bien la documentación de este proyecto es confidencial por tratarse de un sistema de uso militar, la misma puede ser accedida previa consulta con el autor.

1. BARRA DE NAVEGACIÓN

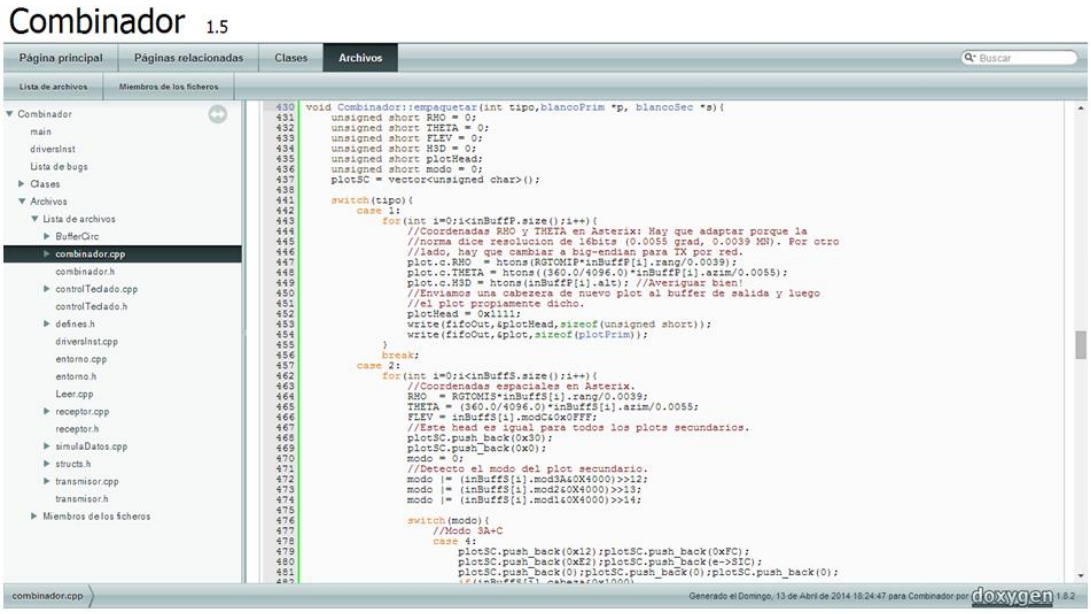
Este menú muestra la distribución general del proyecto y permite acceder a sus elementos ya sea por página donde se ubiquen (A), por clase a la que pertenezcan (B) o por archivo en el que se encuentren implementados (C). Al elegir un componente de la aplicación, como por ejemplo, la clase Combinador, se muestra lo siguiente:



Puede verse que la documentación muestra un diagrama de colaboración (con hipervínculos) de la clase (a), como así también sus métodos públicos y privados (b). Más abajo, en la misma clase, se especifica cada método particularmente con sus respectivos parámetros de entrada y salida, como se muestra a continuación (c).



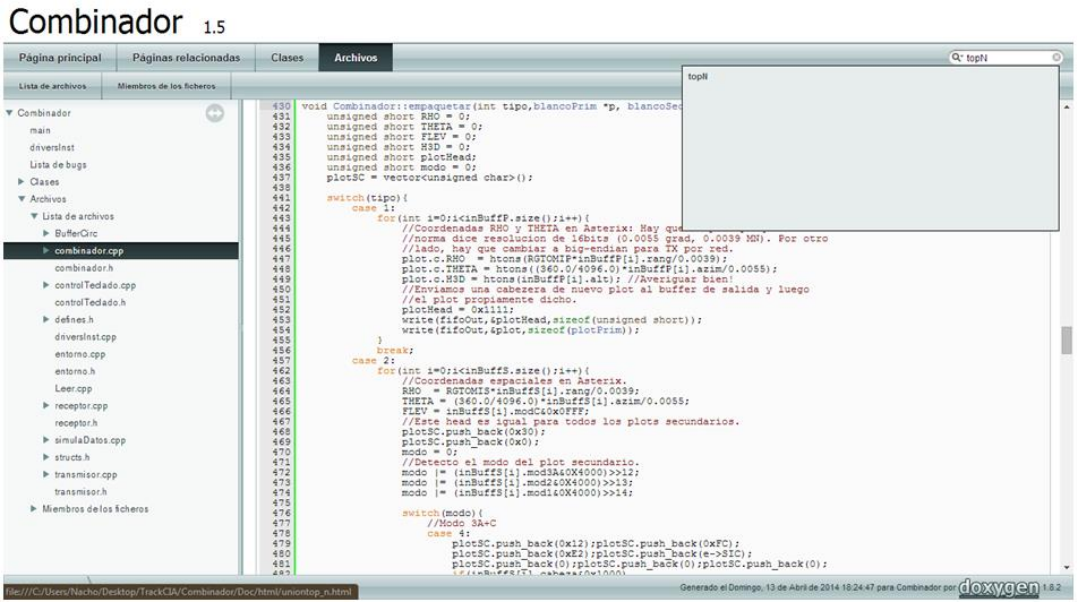
Si se desea ver el código en donde se implementa un determinado elemento de código (en nuestro caso, el método Combinador::empaquetar) basta con dar click al número de línea donde se definió (d). La página con el código fuente de la llamada seleccionada se mostrará en pantalla, como en la siguiente figura:

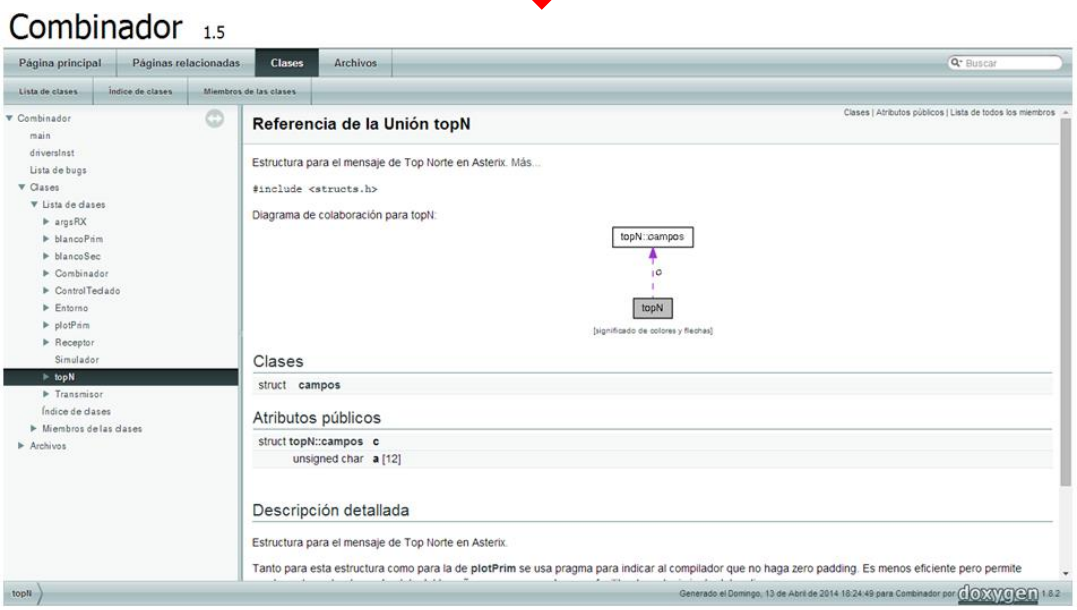


Como puede verse arriba, en rojo se encuentran comentarios específicos de la implementación que no se incluyen en la documentación por una cuestión de claridad. Estos están disponibles al inspeccionar el código fuente, sin embargo, sólo los comentarios más generales e importantes son los que se documentan.

2. BARRA DE BÚSQUEDA

Este es un combo de búsqueda común y corriente. En él puede buscarse cualquier elemento de software de la aplicación. Reconoce tanto clases como métodos, atributos, estructuras, uniones, etc. En la siguiente secuencia de figuras se ve un ejemplo para la unión “topN”:





CAPÍTULO VIII

Conclusiones

Como conclusión de este trabajo de maestría, podemos decir que se ha podido cumplir casi la totalidad de los objetivos planteados. El relevamiento del E.D.D.R. en general y de la aplicación de asociación original, más el estudio del standard ASTERIX fueron tareas clave en la comprensión de la problemática a resolver. Su realización permitió obtener una visión sistémica de un dispositivo que, a primera instancia, parecía una caja negra.

Mediante la investigación sobre los Sistemas Operativos disponibles se encontró una plataforma acorde a los requerimientos, conocida, robusta y sencilla de configurar. Esta elección permitió ahorrar tiempo en la adecuación de la plataforma y, en consecuencia, acelerar el proceso de desarrollo a nivel global respecto de otras alternativas que, si bien podían ser incluso más simples en algunos aspectos, requerían cierta curva de aprendizaje sin aportar verdadera ganancia en eficiencia o versatilidad a la solución.

El rediseño del Combinador para adaptarlo a un paradigma de P.O.O. permitió modularizar el trabajo, desglosando el problema total en una serie de partes más simples y fáciles de atacar. Al mismo tiempo, se logró un código más legible, más simple de mantener y, sobre todo, más fácil de modificar.

La validación mediante casos de uso y las asociadas iteraciones de desarrollo para cumplir con los mismos, derivaron en un diseño de Combinador tolerante a fallos, lo cual, sumado a la robustez hereda del Sistema Operativo que lo aloja, vuelve al sistema en conjunto más confiable.

La documentación total del proyecto permite realizar ajustes sobre el mismo según se requiera e independizarse, en gran medida, de quien lo diseñó –en este caso, el autor. Además de lo expuesto en el capítulo anterior, se confeccionaron manuales para procedimientos simples que permiten al operador realizar tareas de mantenimiento básí-

cas de manera autónoma; algo muy conveniente cuando se trabaja en lugares tan remotos como en los que se encuentra este tipo de sistemas de radar militares.

Sin embargo, es importante destacar que ciertos aspectos de este proyecto no fueron tan positivos, ya que durante el desarrollo del mismo hubo que sortear ciertas dificultades no previstas que impactaron fuertemente en los tiempos de entrega pactados. Entre ellas, podemos nombrar a la volatilidad de los requerimientos planteados por FAA: no hubo un detalle exhaustivo sobre qué se esperaba de la solución pedida y a menudo debió hacerse correcciones sustanciales de diseño por no comulgar la implementación con nuevas especificaciones que iban surgiendo sobre la marcha. Por otro lado, no hubo tantas pruebas de campo con el sistema de radar real como se hubiera querido. La mayoría de los ensayos se realizaron con el simulador disponible, cuyos escenarios distan considerablemente de los que se presentan en el sistema de radar real. Como dicho sistema se encuentra muy alejado del lugar de trabajo y el acceso al mismo debe hacerse respetando cierto protocolo, el proceso de prueba de campo para el Combinador se tornó indeseablemente viscoso.

Finalmente, un objetivo en el cual en cierta medida se fracasó, fue en la obtención de información acerca de sistemas militares semejantes nacionales o internacionales. Esto era algo previsible aún al comienzo del proyecto, ya que este tipo de tecnologías son confidenciales por razones obvias de seguridad nacional.

Trabajo a futuro

Para el presente trabajo, si bien se logró una implementación acorde a los requisitos exigidos por FAA, se reconoce que ciertos aspectos del mismo son perfectibles. Los mismos pueden ser mejorados en gran medida siguiendo las tareas que se recomiendan a continuación:

- Realizar más pruebas de campo con el sistema de radar real.
- Establecer un modelo de protección contra corrupción de *filesystem* por actualización, dentro la misma compact-flash, aprovechando la memoria ociosa disponible.
- Re-formatear la tarjeta de memoria utilizando un sistema de archivos más robusto (como ext3 o JFFS2).
- Añadir un proceso "*keep-alive*" para que, ante una eventual caída del proceso Combinador, se reinicie el mismo sin necesidad de reiniciar el E.D.D.R. completo.

CAPÍTULO IX

Bibliografía

Bernaerts, N. (s.f.). *Debian - Install a server on a compact-flash memory*. Recuperado el 10 de Junio de 2014, de GENEALOGIE: http://bernaerts.dyndns.org/linux/75-debian/53-debian-server-compact-flash

Doxygen. (s.f.). *Doxygen*. Recuperado el 3 de Diciembre de 2014, de http://www.stack.nl/~dimitri/doxygen/

Eurocontrol, A. . (s.f.). *Asterix - Eurocontrol - Cap. V*. Recuperado el 25 de Julio de 2014, de Eurocontrol Web Site: http://www.eurocontrol.int/asterix

Eurocontrol, A. (s.f.). *CAT034 - ASTERIX - Monoradar Service Messages (Part 2b - next version of Cat 002)*. Recuperado el 25 de Julio de 2014, de https://www.eurocontrol.int/sites/default/files/content/documents/nm/asterix/cat034-asterix-monoradar-service-messages-part-2b-next-version-of-cat-002.pdf

Eurocontrol, A. (s.f.). *CAT048 - ASTERIX - Monoradar Target Reports (Part 4 - next version of Cat 001)*. Recuperado el 25 de Julio de 2014, de http://www.eurocontrol.int/sites/default/files/service/content/documents/nm/asterix/cat048-asterix-tmtr-part4-v1.21-20120701.pdf

INVAP-Web. (s.f.). *Radar Primario Argentino 3D (RPA)*. Recuperado el 3 de Diciembre de 2014, de http://www.invap.com.ar/es/espacial-y-gobierno/proyectos-de-gobierno/radar-primario-argentino-3d-rpa.html

RTAI. (s.f.). *RTAI Web*. Recuperado el 20 de Agosto de 2014, de http://www.rtai.org

RTEMS. (s.f.). Recuperado el 20 de Agosto de 2014, de http://www.rtems.org

Skolnik. (1990). *Radar Handbook 2nd Ed.* McGraw-Hill.

Wolff, C. (s.f.). *Radar Basics*. Recuperado el 2014 de Junio de 10, de radartutorial.eu: http://www.radartutorial.eu/02.basics/PSR%20vs.%20SSR.en.html

Wolff, C. (s.f.). *Radar Signal Processor*. Recuperado el 24 de Junio de 2014, de radartutorial.eu: http://www.radartutorial.eu/10.processing/sp05.en.html

ANEXO I

Casos de Uso

ID	0.1
TIPO	**NORMAL**
PROPÓSITO	COMPROBAR QUE EL SISTEMA OPERATIVO ARRANCA NORMALMENTE Y PERMITE APAGADO REPENTINO SIN ROMPER EL SISTEMA DE ARCHIVOS.
PRECONDICIONES	- MÓDULO ADAPTADOR/SIMULADOR EN MODO SIMULACIÓN - TARJETA COMPACTFLASH CON ETIQUETA "COMBINADOR" INSTALADA EN EL MÓDULO COMBINADOR - MÓDULOS PRIMARIO Y SECUNDARIO DESCONECTADOS - MÓDULO COMBINADOR CONECTADO
PROCEDIMIENTO	1) CONECTAR ALIMENTACIÓN A LA FUENTE DEL EDDR 2) CONECTAR MONITOR A LA SALIDA VGA DEL MÓDULO COMBINADOR 3) ENCENDER EL EDDR 4) OBSERVAR LA PANTALLA 5) LUEGO DEL ARRANQUE DEL SISTEMA OPERATIVO, APAGAR EL EDDR DESDE EL BOTÓN DE ENCENDIDO/APAGADO Y REPETIR EL PUNTO 3
POSTCONDICIONES	1) LA PANTALLA DEBERÁ MOSTRAR QUE EL SISTEMA OPERATIVO ARRANCA Y LA APLICACIÓN COMBINADOR SE EJECUTA, INICIALIZANDO TODOS LOS SUBSISTEMAS Y QUEDANDO A LA ESPERA DE DATOS PRIMARIOS/SECUNDARIOS 2) INDEPENDIENTEMENTE DE CUÁNTAS VECES SE REALICE EL ENSAYO, EL SISTEMA OPERATIVO DEBE ARRANCAR SIEMPRE
OBSERVACIONES	EL TIEMPO DE ARRANQUE DEL SISTEMA OPERATIVO DEBE SER MENOR A 40 SEGUNDOS.

El resultado de la aplicación del caso de uso 0.1 puede apreciarse en la figura A, a continuación. Se observa cómo el EDDR, luego del arranque del Sistema Operativo, ejecuta la aplicación Combinador y queda a la espera de datos primarios y secundarios (procesadores desconectados) y enviando los mensajes de Top Norte en cada ciclo de antena. El tiempo de inicio del sistema medido es de 32 segundos.

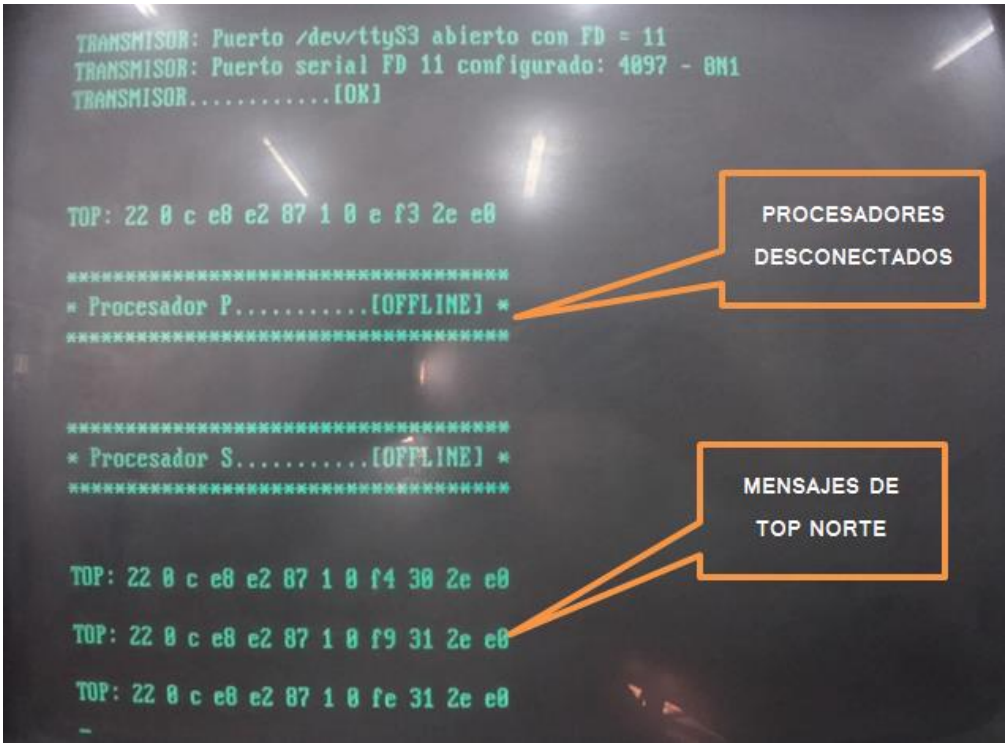


Figura A: Post-condiciones del caso de uso 0.1.

ID	**0.2**
TIPO	**NORMAL**
PROPÓSITO	COMPROBAR QUE, ANTE EL ENVÍO DE PLOTS PRIMARIOS Y SECUNDARIOS, SE OBTIENEN PLOTS COMBINADOS A LA SALIDA DEL COMBINADOR.
PRECONDICIONES	- MÓDULO ADAPTADOR/SIMULADOR EN MODO SIMULACIÓN - TARJETA COMPACTFLASH CON ETIQUETA "COMBINADOR" INSTALADA EN EL MÓDULO COMBINADOR - MÓDULOS PROCESADOR PRIMARIO, PROCESADOR SECUNDARIO Y COMBINADOR, CONECTADOS - CABLE DE RED CONECTADO DESDE LA SALIDA RJ-45 DEL MÓDULO COMBINADOR HACIA UNA PC QUE CUENTE CON WIRESHARK v1.12.2 (O SUPERIOR)
PROCEDIMIENTO	1) CONECTAR ALIMENTACIÓN A LA FUENTE DEL EDDR 2) ENCENDER EL EDDR 3) EN LA PC, EJECUTAR WIRESHARK Y CONFIGURARLO PARA CAPTURAR DESDE LA INTERFAZ DE RED CABLEADA DONDE SE CONECTÓ EL CABLE DESDE EL E.D.D.R. Y SELECCIONAR EL FILTRO "udp port XXXX[32]" 4) OBSERVAR LA PANTALLA DE WIRESHARK
POSTCONDICIONES	LUEGO DEL ARRANQUE DEL SISTEMA OPERATIVO DEL EDDR, SE DEBERÁN OBSERVAR PAQUETES UDP QUE LLEGAN A LA PC.
OBSERVACIONES	HACIENDO CLICK EN CADA UNO DE LOS PAQUETES SE CORROBORA SU INTEGRIDAD.

[32] El número de puerto a colocar dependerá de la configuración de red con la que se haya inicializado el Combinador. Ver archivo configParam.conf.

Luego de aplicar el caso de uso 0.2, los resultados obtenidos son los que se muestran en las figuras B y C, a continuación:

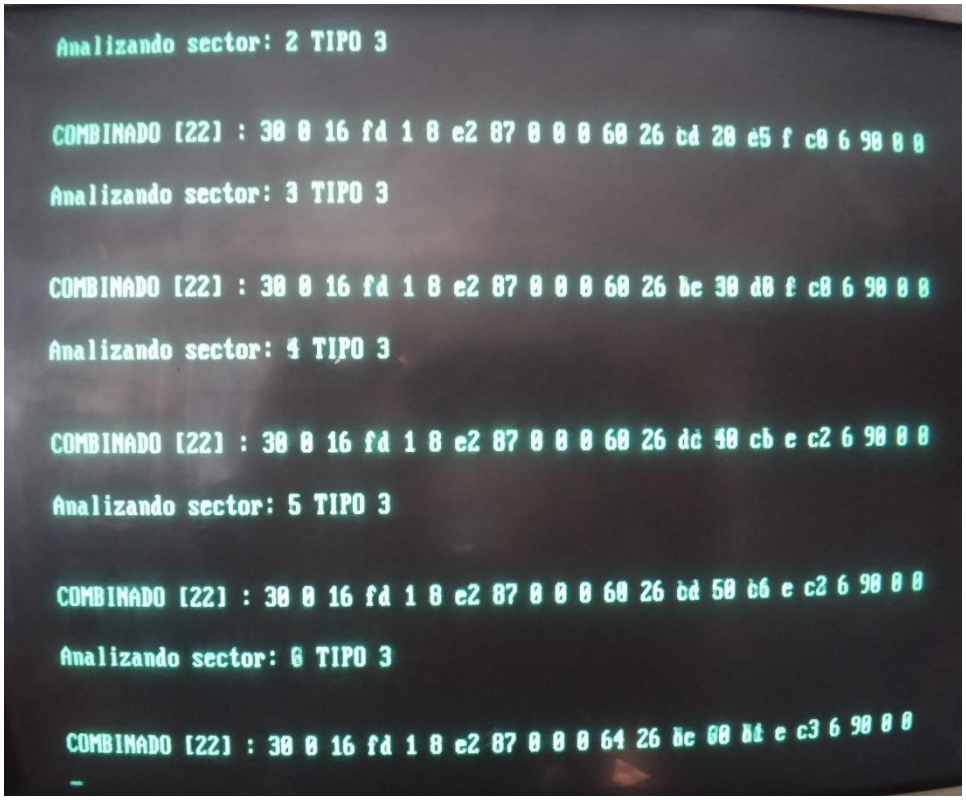


Figura B: Post-condiciones del caso de uso 0.2. Combinador funcionando.

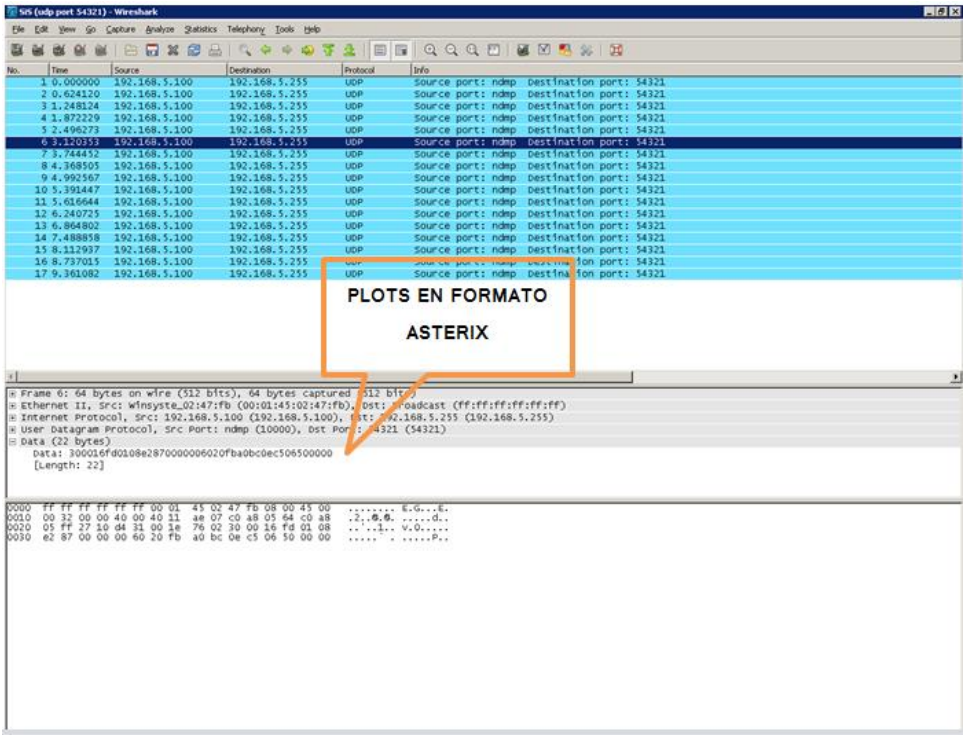


Figura C: Post-condiciones del caso de uso 0.2. Plots combinados, enviados a través de interfaz Ethernet.

ID	0.3
TIPO	NORMAL
PROPÓSITO	COMPROBAR QUE LOS PLOTS ANALIZADOS SE MAPEAN CORRECTAMENTE EN EL ESPACIO.
PRECONDICIONES	- MÓDULO ADAPTADOR/SIMULADOR EN MODO SIMULACIÓN - TARJETA COMPACTFLASH CON ETIQUETA “COMBINADOR” INSTALADA EN EL MÓDULO COMBINADOR - MÓDULOS PROCESADOR PRIMARIO, PROCESADOR SECUNDARIO Y COMBINADOR, CONECTADOS - CABLE DE RED CONECTADO DESDE LA SALIDA RJ-45 DEL MÓDULO COMBINADOR HACIA UNA PC QUE CUENTE CON LA CONSOLA DE VISUALIZACIÓN DE FAA
PROCEDIMIENTO	1) CONECTAR ALIMENTACIÓN A LA FUENTE DEL EDDR 2) ENCENDER EL EDDR 3) EN LA PC, EJECUTAR LA CONSOLA DE VISUALIZACIÓN (ARCHIVO .JAR) 4) CLICKEAR EL BOTÓN INCIAR CONEXIÓN EN LA CONSOLA 5) ELEGIR LA OPCIÓN DE VISUALIZACIÓN DE PLOTS COMBINADOS
POSTCONDICIONES	LUEGO DEL ARRANQUE DEL SISTEMA OPERATIVO DEL EDDR, LOS PLOTS ENVIADOS POR EL COMBINADOR DEBERÁN SER REPRESENTADOS EN LA PANTALLA DE RADAR VIRTUAL. LOS MISMOS DEBEN APARECER DE A UNO O MÁS POR SECTOR, SEGÚN SE HAYA CONFIGURADO EL SIMULADOR. LOS PLOTS PRIMARIOS SE GRAFICAN EN AMARILLO, LOS SECUNDARIOS EN CELESTE Y LOS COMBINADOS EN NARANJA.
OBSERVACIONES	CADA 12 SEGUNDOS SE DEBERÁ VISUALIZAR QUE EL CUADRADO DE LA ESQUINA SUPERIOR DERECHO DE LA CONSOLA VIRTUAL PARPADEA EN AZUL, INDICANDO LA LLEGADA DE UN MENSAJE DE TOP NORTE.

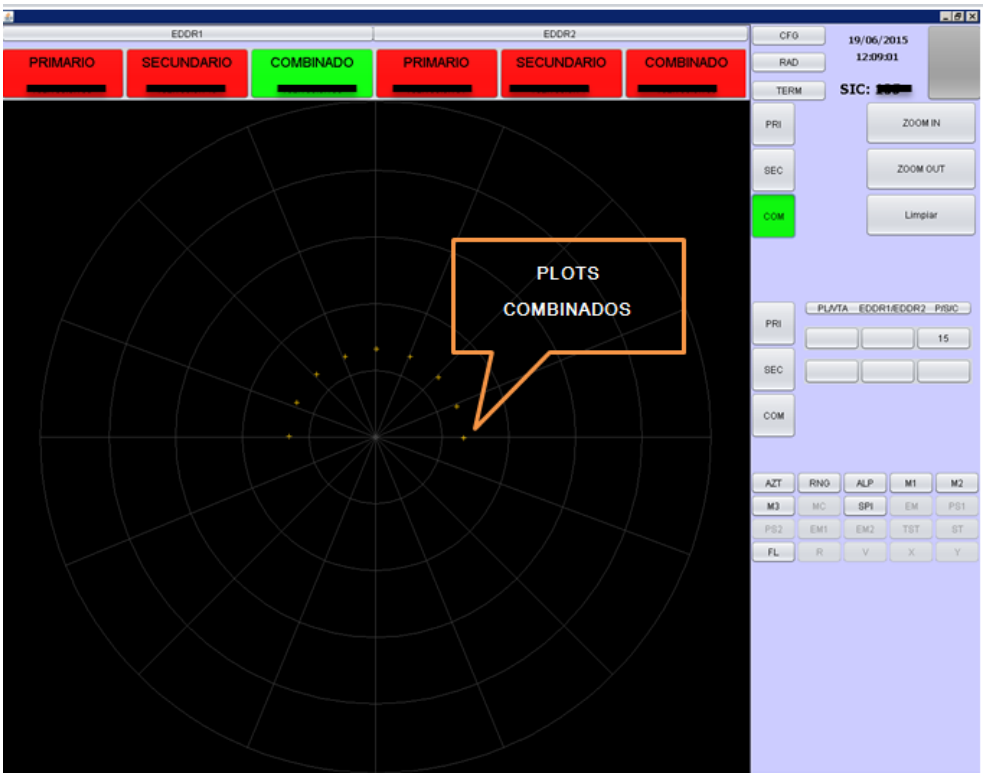


Figura D.1: Post-condiciones del caso de uso 0.3. Plots mapeados correctamente.

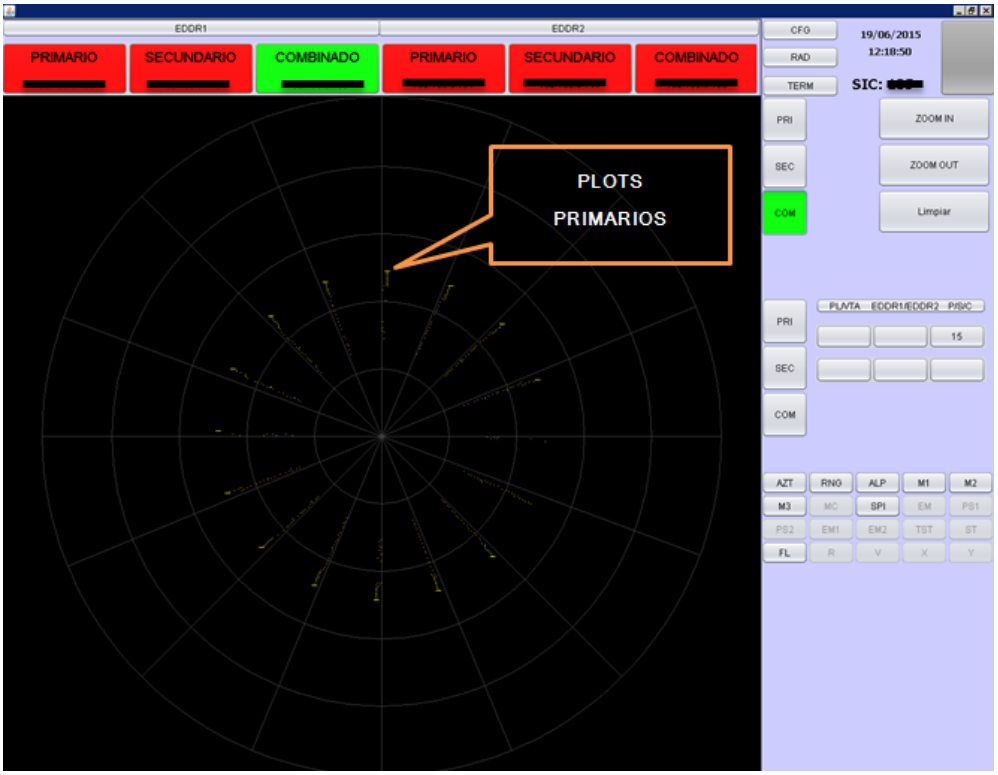


Figura D.2: Post-condiciones del caso de uso 0.3. Plots mapeados correctamente.

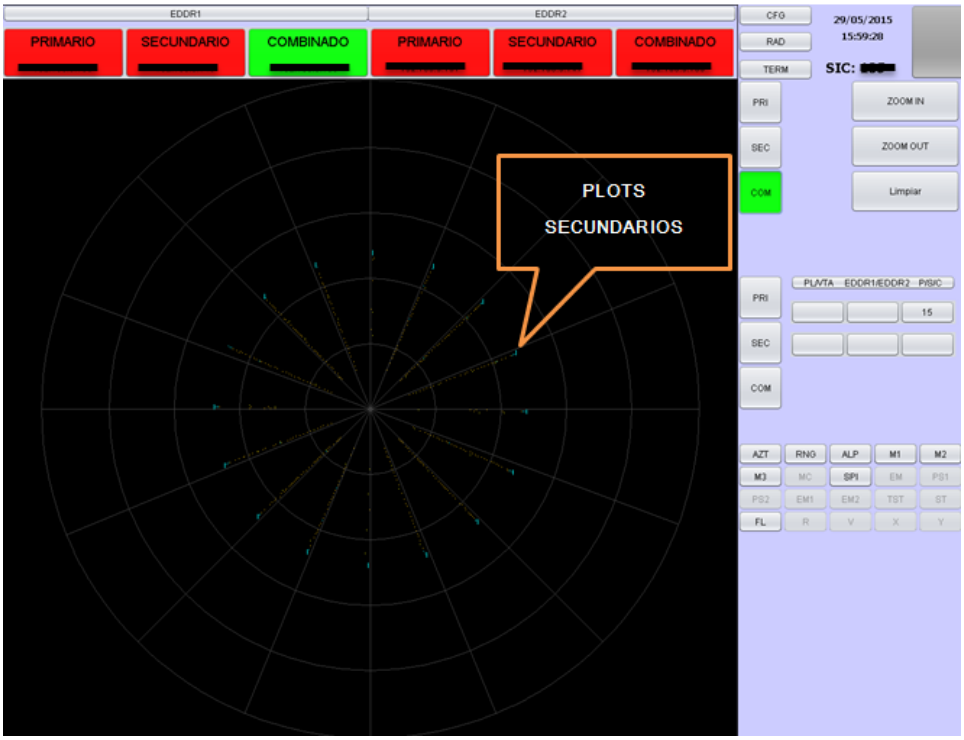


Figura D.3: Post-condiciones del caso de uso 0.3. Plots mapeados correctamente.

ID	**1.1**
TIPO	**ALTERNATIVO**
PROPÓSITO	COMPROBAR QUE ANTE LA AUSENCIA DE UNO DE LOS PROCESADORES, EL COMBINADOR CONTINUA ENVIANDO LOS PLOTS DEL PROCESADOR QUE SE ENCUENTRA EN FUNCIONAMIENTO.
PRECONDICIONES	- MÓDULO ADAPTADOR/SIMULADOR EN MODO SIMULACIÓN - TARJETA COMPACTFLASH CON ETIQUETA “COMBINADOR” INSTALADA EN EL MÓDULO COMBINADOR - MÓDULO PROCESADOR PRIMARIO O SECUNDARIO DESCONECTADO - MÓDULO COMBINADOR CONECTADO - CABLE DE RED CONECTADO DESDE LA SALIDA RJ-45 DEL MÓDULO COMBINADOR HACIA UNA PC QUE CUENTE CON LA CONSOLA DE VISUALIZACIÓN DE FAA
PROCEDIMIENTO	1) CONECTAR ALIMENTACIÓN A LA FUENTE DEL EDDR 2) CONECTAR MONITOR A LA SALIDA VGA DEL MÓDULO COMBINADOR 3) ENCENDER EL EDDR 4) CLICKEAR EL BOTÓN INCIAR CONEXIÓN EN LA CONSOLA 5) ELEGIR LA OPCIÓN DE VISUALIZACIÓN DE PLOTS SECUNDARIOS O PRIMARIOS SEGÚN QUÉ PROCESADOR SE ENCUENTRE CONECTADO 6) OBSERVAR EN EL MONITOR Y EN LA CONSOLA DE VISUALIZACIÓN LA SALIDA DE DATOS
POSTCONDICIONES	SI EL PROCESADOR DESCONECTADO ES EL PRIMARIO, DEBERÁ OBSERVARSE UNA SERIE DE PLOTS SECUNDARIOS EN LA PANTALLA VIRTUAL DE RADAR. LA SITUACIÓN INVERSA DEBERÁ OBSERVARSE SI EL PROCESADOR DESCONECTADO ES EL SECUNDARIO.

OBSERVACIONES	CADA 12 SEGUNDOS SE DEBERÁ VISUALIZAR QUE EL CUADRADO DE LA ESQUINA SUPERIOR DERECHO DE LA CONSOLA VIRTUAL PARPADEA EN AZUL, INDICANDO LA LLEGADA DE UN MENSAJE DE TOP NORTE. EL MISMO PROCEDIMIENTO PUEDE ENSAYARSE "EN CALIENTE", ES DECIR, DESCONECTAR ALGUNO DE LOS PROCESADORES MIENTRAS ESTÉ FUNCIONANDO. EN EL MONITOR CONECTADO AL COMBINADOR SE OBSERVARÁ UN MENSAJE QUE INDICA QUE EL PROCESADOR RECIENTEMENTE DESCONECTADO SE ENCUENTRA "OFFLINE".

Los resultados de la aplicación del caso de uso 1.1, en la pantalla virtual de radar, son análogos a las figuras D.2 y D.3. Sin embargo, se muestra a continuación la salida en pantalla que ofrece el Combinador al operador del Sistema Radar, ante la desconexión repentina del Procesador Secundario.

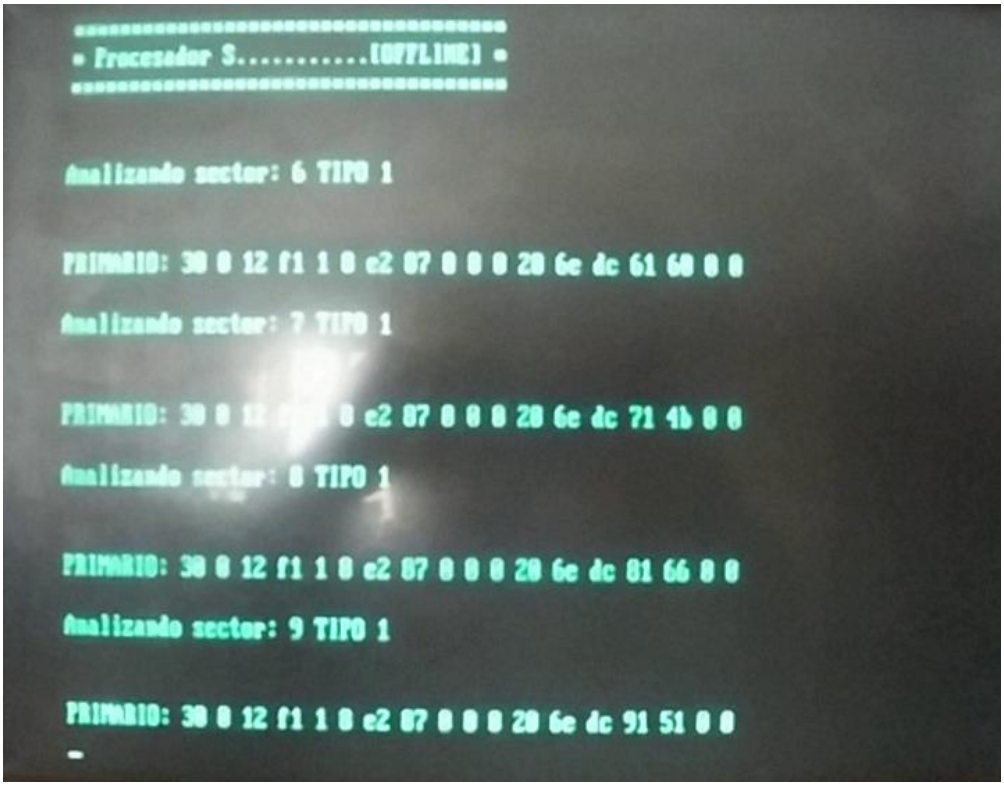


Figura E: Desconexión del Procesador Secundario.

ID	**1.2**
TIPO	**ALTERNATIVO**
PROPÓSITO	COMPROBAR QUE, AL REANUDARSE EL FUNCIONAMIENTO DE ALGUNO DE LOS PROCESADORES (PRIMARIO O SECUNDARIO), EL COMBINADOR COMIENZA A ENTREGAR PLOTS COMBINADOS.
PRECONDICIONES	- MÓDULO ADAPTADOR/SIMULADOR EN MODO SIMULACIÓN - TARJETA COMPACTFLASH CON ETIQUETA "COMBINADOR" INSTALA

	DA EN EL MÓDULO COMBINADOR - MÓDULO PROCESADOR PRIMARIO O SECUNDARIO DESCONECTADO - MÓDULO COMBINADOR CONECTADO - CABLE DE RED CONECTADO DESDE LA SALIDA RJ-45 DEL MÓDULO COMBINADOR HACIA UNA PC QUE CUENTE CON LA CONSOLA DE VISUALIZACIÓN DE FAA - EDDR FUNCIONANDO Y ENTREGANDO PLOTS SECUNDARIOS O PRIMARIOS RESPECTIVAMENTE
PROCEDIMIENTO	1) MIENTRAS EL EDDR SE ENCUENTRA EN FUNCIONAMIENTO, CONECTAR EL PROCESADOR QUE SE ENCUENTRE DESCONECTADO 2) ESPERAR UNOS 20 SEGUNDOS HASTA QUE SE INICIALICE DICHO PROCESADOR 3) OBSERVAR EN EL MONITOR Y EN LA CONSOLA DE VISUALIZACIÓN LA SALIDA DE DATOS
POSTCONDICIONES	A PARTIR DEL ARRANQUE DEL PROCESADOR RECIENTEMENTE CONECTADO, DEBERÁ OBSERVARSE, TANTO EN EL MONITOR CONECTADO AL COMBINADOR, COMO EN LA PANTALLA DE RADAR VIRTUAL, UNA SERIE DE PLOTS COMBINADOS.
OBSERVACIONES	CADA 12 SEGUNDOS SE DEBERÁ VISUALIZAR QUE EL CUADRADO DE LA ESQUINA SUPERIOR DERECHO DE LA CONSOLA VIRTUAL PARPADEA EN AZUL, INDICANDO LA LLEGADA DE UN MENSAJE DE TOP NORTE.

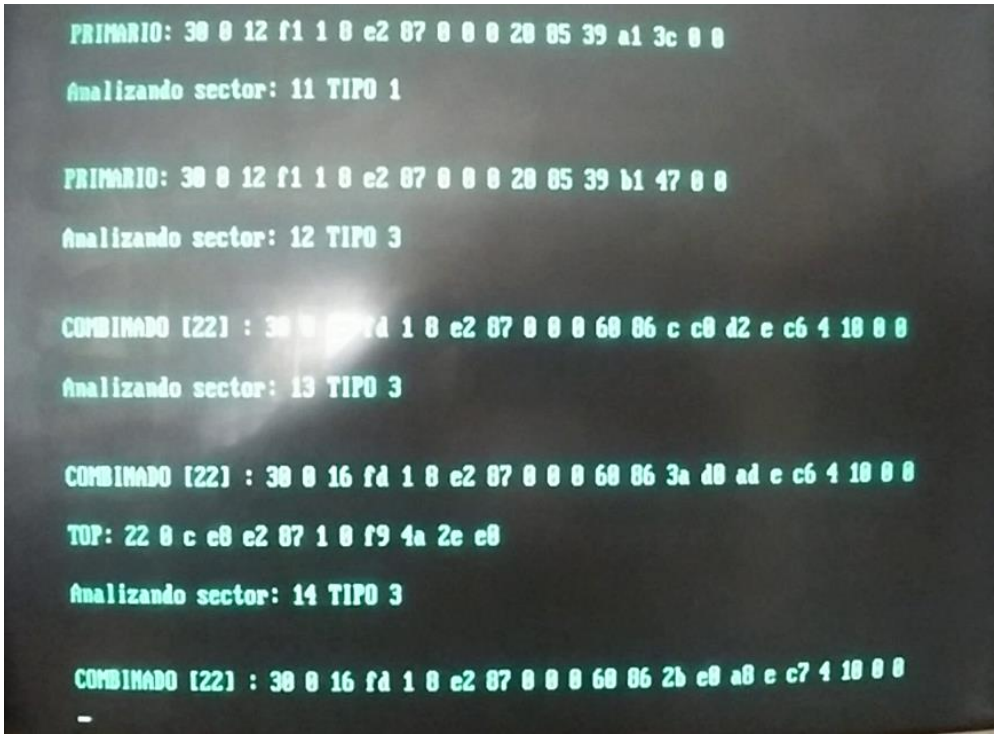


Figura F: Combinador reanuda la operación al reconectar el Procesador Secundario.

ANEXO II

Instalación y Configuración del S.O.

Se enumeran, a continuación, las tareas realizadas para la instalación del Sistema Operativo seleccionado en la tarjeta de memoria de la SBC del E.D.D.R.[33]. Se consultó un artículo a modo de guía para la configuración (Bernaerts).

1- Se instaló una versión básica de Debian GNU/Linux Lenny utilizando el CD-ROM de dicha distribución en el disco rígido del kit de desarrollo. Se particionó el disco acorde a la memoria flash disponible en la SBC; esto es, 2GiB de memoria total, sin unidad de SWAP[34].

2- Se agregaron los paquetes necesarios para el desarrollo del software Combinador: compiladores, make, nano, librerías fuente (*headers*) necesarias para el desarrollo de drivers, entre otros.

3- Se eliminaron todos aquellos paquetes no utilizados por el Combinador (controladores de sonido, periféricos y servicios innecesarios).

4- Se configuró el archivo "/etc/fstab"[35] para ejecución completa desde memoria RAM y se configuró la unidad de almacenamiento para dicho fin, usando ***tune2fs***.

5- Se configuró la interfaz de red para ser compatible con la intranet de FAA.

6- Se creó una imagen de la partición completa, ya configurada y se replicó la misma en varias tarjetas compact-flash. Una se instaló en la SBC del Combinador y las demás

[33] Para mayor información al respecto, contactar al autor.

[34] Unidad de SWAP (o paginación) es una porción de disco que el Sistema Operativo reserva para alojar, temporalmente, contenido de la memoria RAM, liberando así espacio en memoria para alojar a otros datos importantes.

[35] **fstab**, la tabla de sistemas (*filesystem table*, en inglés) de archivos de un sistema basado en UNIX. Lista las particiones disponibles en el sistema, su punto de montaje y sus opciones de configuración

se guardaron como respaldo, en caso de fallas o de necesidad de implementar más Combinadores en otros E.D.D.R. La versión terminada del sistema de archivos tiene un tamaño de 500MB, aproximadamente.