

INTERFAZ GRÁFICA PARA MONITOR DE SEÑALES VITALES

VITALS GRAFR UY

AUTORES: DIEGO ALONSO, GERMAN MINETTI, WALTER BARREIRO

Emails: diego.alonso.ruiz@fing.edu.uy german.minetti@fing.edu.uy walter.barreiro@fing.edu.uy

Tutores: Leonardo Steinfeld y Varinia Cabrera

Emails: leo@fing.edu.uy, vcabrera@fing.edu.uy

Instituto de Ingeniería Eléctrica - Facultad de Ingeniería - UDELAR

RESUMEN

Este proyecto consiste en el diseño e implementación de un sistema capaz de recibir y desplegar en pantalla tres señales vitales de un paciente hospitalario -ECG (Electrocardiograph), IBP (Invasive Blood Pressure) y SpO2 (Pulse Oxygen Saturation)-, así como parámetros e indicadores de las mismas. A su vez, el sistema consta de un sistema de alarmas visuales. Los rangos de las alarmas son configurables manualmente por el usuario mediante el menú de ajustes ubicado en la interfaz de la pantalla, así como a través de una interfaz en un PC comunicada mediante protocolo Wifi. La pantalla principal también cuenta con un reloj. El sistema está basado en un Launchpad MSP-EXP432P401R, un display BOOSTXL-K350QVG y un BoosterPack CC3100 que incluye una radio Wifi, todos equipos de Texas Instruments.

Índice

1. Introducción	2
1.1. Descripción del problema	2
1.2. Antecedentes	3
2. Objetivos	4
3. Alcance	5
4. Descripción del Sistema	6
5. Implementación	8
5.1. Hardware	8
5.2. Software	9
5.2.1. Módulos	9
5.2.2. Programa Principal (main)	18
5.2.3. Interfaz en ordenador	21
6. Pruebas y Mediciones	23
6.1. Pruebas de Módulos	23
6.1.1. Módulo Recepción	23
6.1.2. Módulo Display	23
6.1.3. Módulo Procesamiento	23
6.1.4. Módulo Wifi	24
6.1.5. Módulos Restantes	25
6.2. Integración de Módulos	25
6.3. Reporte de Uso de memoria	25
7. Conclusiones	26
8. Anexo	27
8.1. Conceptos del curso aplicados al proyecto	27
8.2. Planificación del proyecto	27
8.3. Especificación del proyecto	27
Bibliografía	28

1. Introducción

Ante la actual crisis sanitaria producida por la pandemia causada por COVID-19, Uruguay se enfrenta a la necesidad de ampliar la capacidad de trabajo de los Centros de Tratamiento Intensivo (CTI). Esto implica adquirir más equipamiento e insumos que actualmente son escasos a nivel local y que, debido a la demanda mundial, también podrían resultar difíciles de importar, aumentando su costo o directamente impidiendo al país contar con ellos.

Los monitores de signos vitales son equipos fundamentales, requeridos y de uso obligatorio en los CTI. A través de este proyecto asumimos el desafío de buscar soluciones alternativas al mismo. Nos enfocaremos en el monitoreo de las tres señales vitales.

El sistema implementado se encarga de desplegar en tiempo real en una pantalla LCD (Liquid Crystal Display) táctil las señales de ECG (electrocardiograma), IBP (presión sanguínea) y SpO2 (saturación de oxígeno), así como algunos valores derivados. Además, se desplazan alarmas visuales si esas señales superan un cierto valor máximo o mínimo, previamente configurado por el usuario en una pantalla de configuración del LCD o a través de una interfaz en PC (Personal Computer).

El sistema está compuesto por un LaunchPad MSP-EXP432P401R, un BoosterPack CC3100 y un BoosterPack BOOSTXL-K350QVG-S1, productos de Texas Instruments.

1.1. Descripción del problema

Ante la inminente llegada del coronavirus al Uruguay, y basados en la experiencia de los países más afectados, se toma como línea general a nivel país el fortalecimiento integral del sistema de salud. Esto implica la contratación y reasignación del personal de salud, la creación de salas de internación especiales así como de los CTI y el reacondicionamiento de los existentes. Estas reformas requieren de un presupuesto elevado, en un momento en que el futuro de la economía del país se veía incierto y donde los insumos relacionados con la salud se habían encarecido considerablemente.

En este sentido se presenta este proyecto, con el fin de aportar un sistema necesario que complemente los equipamientos de los CTI, pero a su vez sea desarrollado en el país y a un bajo costo. Con este sistema de despliegue de señales se pueden mostrar los principales indicadores vitales de un paciente, tanto gráficamente las formas de onda de las señales como los valores más significativos, además de recibir alertas de condiciones anormales, permitiendo una rápida intervención del personal de salud.

El sistema está constituido por un dispositivo, donde se agrupa el hardware y se desarrolla la mayor parte del programa, y una interfaz gráfica externa de uso opcional para configuraciones. Las funcionalidades del sistema incluyen la recepción de señales digitales, su posterior procesamiento para elaborar las gráficas y los valores numéricos, así como su correspondiente despliegue en pantalla. Para ello se contará con una interfaz en display LCD, incluido en el dispositivo, donde se podrán observar las señales en tiempo real, los valores numéricos (mínimo, máximo y medio), así como las alarmas, la hora actual y el menú de ajustes. En este menú podrán configurarse los valores límites de las alarmas y el reloj. Se agrega también la opción de contar una interfaz de ajustes desde una PC, comunicándose a través de Wifi (por protocolo TCP/IP) o UART (Universal Asynchronous Receiver-Transmitter). Con esta interfaz externa se podrán configurar los mismos parámetros que en el display.

1.2. Antecedentes

Para el desarrollo del sistema nos basamos en dos proyectos de fin de curso de esta materia y en las especificaciones y funcionalidades de un equipo de monitoreo comercial.

DPM 4 Patient Monitor

Este modelo de monitor cuenta con display, botones para comando del mismo (*e.g.* seteo de alarmas) y en algunos casos incluye pantalla táctil [1]. Comúnmente cuentan con comunicación para configuración y/o envío de datos a un equipo central.

Este monitor de señales portátil está diseñado para el seguimiento de parámetros fisiológicos, además del registro y alertas de los mismos, en pacientes adultos y pediátricos. El monitor cuenta con un display de alta resolución que permite el despliegue de gráficas, valores y alarmas visuales, además de botones para la navegación e incluso botón de emergencia para solicitar atención médica.

El equipo es capaz de monitorear siete parámetros, entre los que se encuentran las 3 señales que vamos a utilizar en nuestro sistema.

Por último, el monitor también emite alarmas sonoras y visuales ante un signo vital anormal, o ante problemas tanto mecánicos como eléctricos que puedan presentarse.

Proyecto GuríZ

Este proyecto del año 2019 [2] forma parte también de un proyecto de fin de carrera y fue consultado puesto que cuenta con un display BOOSTXL-K350QVG-S1, utilizado en nuestro proyecto para el despliegue de señales en pantalla.

El proyecto diseña e implementa un dispositivo médico destinado a medir la impedancia eléctrica transtorácica en pacientes pediátricos. El dispositivo es capaz de recibir señales analógicas de tensión y digitalizarlas, promediar las muestras y a partir de éstas realizar cálculos, desplegándolos en un monitor.

El hardware incluye el LaunchPad MSP-EXP430F5529LP y el BoosterPack BOOSTXL-K350QVG-S1. Para la interacción con el usuario se emplea dos botones para navegación y un display, para el que se utilizó la librería gráfica disponible en Texas Instruments.

El equipo además permite guardar datos para descargar en PC. La comunicación se realiza vía UART.

Proyecto DAMAVA

Este proyecto [3] sirve de base para una comunicación inalámbrica con la interfaz en PC de nuestro sistema, en base a tecnología Wifi. Además, se utilizará como referencia la implementación de la comunicación SPI entre los dos Launchpad utilizados.

Este proyecto consiste en el desarrollo de módulos de comunicación y adquisición de datos de un sistema para la detección de la mastitis vacuna, basado en el monitoreo de la conductividad y temperatura en la leche.

A través de sensores adquiere medidas enviando los datos mediante Wifi a una PC, empleando también una interfaz de usuario. El sistema está compuesto por un LaunchPad MSP-EXP432P401R, un LaunchPad SimpleLink Wifi CC3100 y un router utilizado como Access Point.

El diseño del sistema permite que pueda ser utilizado por otras aplicaciones que requieran procesos de adquisición y envío de datos, cambiando únicamente la lógica de algunos módulos.

2. Objetivos

El objetivo principal es desarrollar un sistema de bajo costo que permita la recepción de tres señales vitales digitalizadas y el posterior despliegue en pantalla en el dispositivo, tanto de sus formas de onda como de parámetros de interés, así como un sistema de alarmas configurables para detectar la recepción de valores anómalos de las señales en cuestión. Se busca implementar también una interfaz externa que permita configurar las alarmas y el reloj desde una PC.

En particular planteamos también los siguientes objetivos particulares:

- Procesamiento de datos de tensiones para su transformación en datos graficables.
- Despliegue en pantalla de las tres señales vitales y sus valores numéricos correspondientes.
- Pantalla de ajustes para la configuración de los parámetros de alarmas y reloj.
- Sistema de alarmas configurables.
- Interfaz en PC con pantalla de ajustes.
- Comunicación Wifi entre el dispositivo y la interfaz en PC.

3. Alcance

El proyecto abarca el diseño y la implementación de un sistema que despliegue señales en pantalla, además de un reloj y alarmas configurables.

El sistema a diseñar podrá recibir desde un adquisidor externo los datos digitales de las señales vitales del paciente, así como los valores *mínimo*, *medio* y *máximo* de cada una. En un principio nuestro sistema tomará y graficará datos reales pregrabados, dejando la posibilidad de configurar la recepción a tiempo real en un futuro.

El sistema de alarmas verificará que los datos ingresados no excedan los rangos definidos para cada señal. En caso de detectarse un valor fuera del rango definido, se activará una alarma visual en la pantalla y un botón para reconocer y desactivar la misma. Los rangos mencionados serán configurables manualmente por el usuario en el menú de ajustes. En dicho menú se podrá configurar también la hora actual.

Se contará con una aplicación a usar en PC a modo de interfaz de usuario, en la cual se podrán configurar los parámetros de las alarmas y el reloj, al igual que en la pantalla del dispositivo. La comunicación entre el dispositivo y la aplicación podrá ser vía Wifi o UART.

El sistema no incluye la adquisición de señales eléctricas ni su conversión analógico-digital. Tampoco incluye grabación de señales ni datos del paciente.

4. Descripción del Sistema

El sistema a desarrollar graficará en el display LCD las formas de onda de las tres señales vitales mencionadas anteriormente. También mostrará valores mínimos, medios y máximos de frecuencia cardíaca, presión de sangre y del pulso de saturación de oxígeno. La información de las tres señales se estima que será recibida a partir de un sistema externo, que se encargará de la adquisición y digitalización de las tensiones analógicas. En la figura 1 se puede ver el diagrama de bloques de alto nivel del sistema.

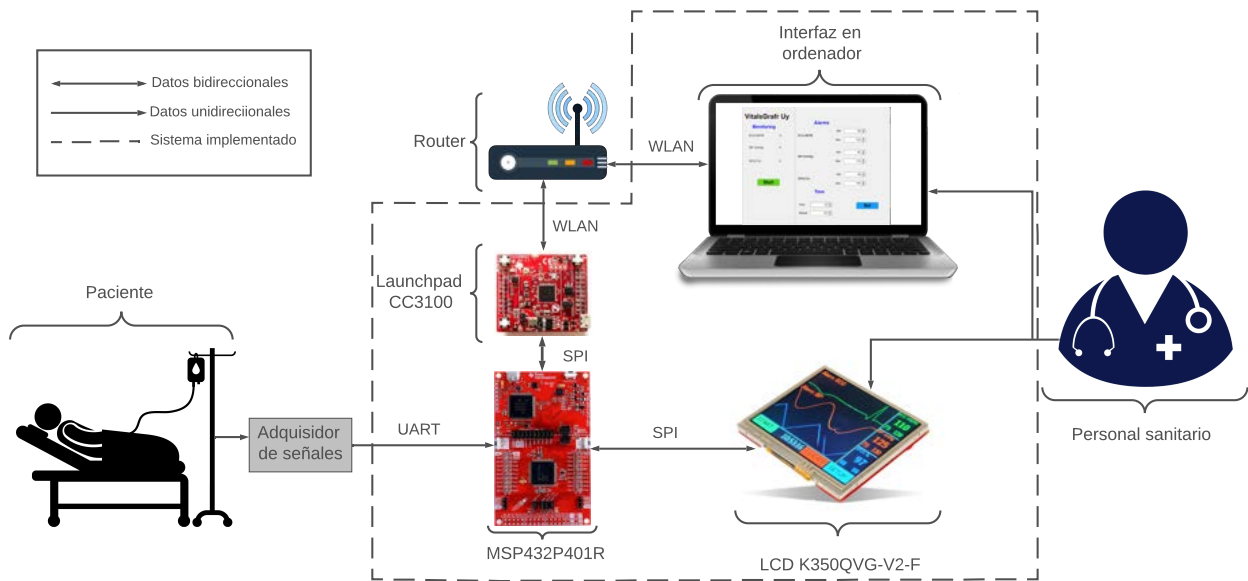


Figura 1: Diagrama de bloques de alto nivel.

El procesamiento de los datos recibidos consistirá en elaborar valores graficables, transformando valores de tensión en píxeles y luego en coordenadas, ajustando también su escala para finalmente construir la gráfica en pantalla.

Se podrá trabajar con una interfaz de usuario externa desde una PC. En ella se podrán configurar las alarmas y la hora mediante una pantalla de ajustes, similar a la diseñada en el display táctil del dispositivo.

El sistema se inicia en modo Station, conectándose con el Access Point (router) y abriendo el puerto 60001. Previo a la inicialización del programa, se podrán ajustar las configuraciones desde la interfaz, pudiendo hacerlo también luego desde el display del dispositivo. En caso de no realizar ajustes manualmente desde la interfaz previo al comienzo del programa, el sistema inicializa las configuraciones con valores por defecto. Para la comunicación se envía desde el dispositivo el carácter 'S' solicitando la transmisión de los datos. La interfaz entonces envía el conjunto de configuraciones, terminando con un carácter 'F' indicando el fin de una transmisión exitosa.

Luego de una pantalla de presentación, el sistema muestra el menú principal del programa, donde se encuentran los botones "START" y "SETUP". Al seleccionar el botón "START" se da inicio al despliegue de las señales y los parámetros. Con el botón "SETUP" se ingresa al

menú de ajustes.

En el menú de ajustes se puede configurar el reloj y los valores límite de las alarmas. También se encuentran el botón "SET", para confirmar los cambios en las configuraciones, y el botón "BACK" para regresar a la pantalla principal.

El sistema de alarmas funciona identificando la recepción de valores fuera del rango definido por el usuario. En ese caso, se desplegará en pantalla un texto de color rojo indicando la activación de la alarma de la señal correspondiente. Esta alarma podrá ser desactivada mediante la selección del botón "STOP ALARM". Sin embargo esta acción no ignorará alertas posteriores, es decir, si los siguientes valores continúan fuera del rango, la alarma continuará apareciendo en pantalla. Esta funcionalidad evita que por descuido, el usuario desactive las alarmas indefinidamente.

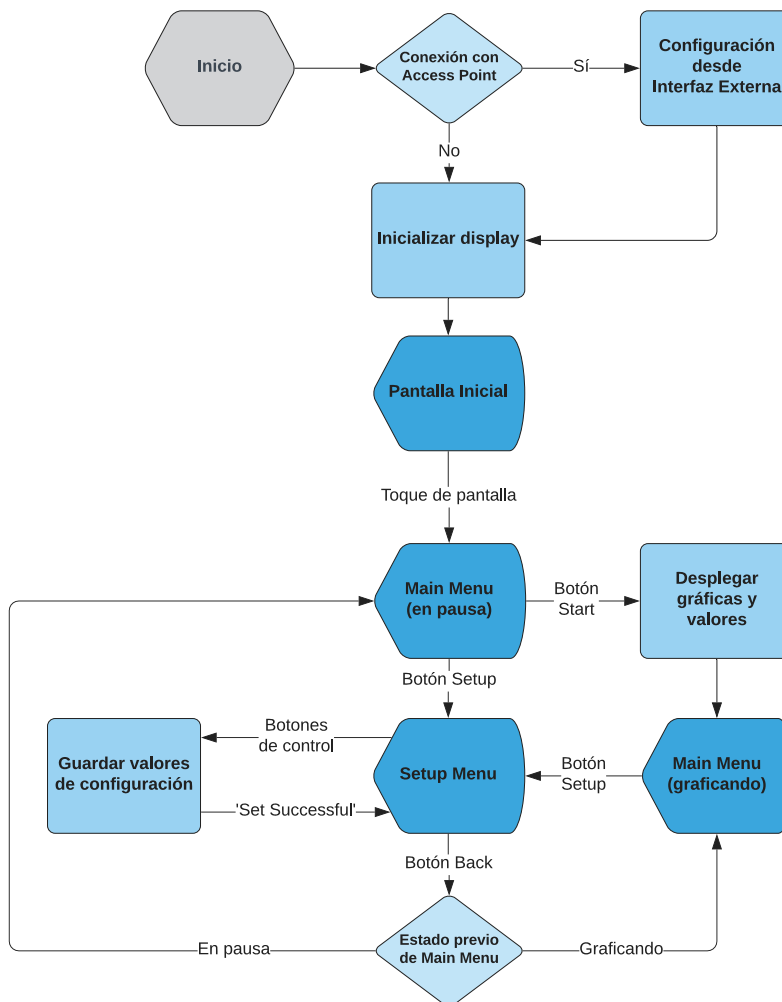


Figura 2: Diagrama de flujo de la interfaz en el display

5. Implementación

5.1. Hardware

La composición del hardware del sistema parte de un Launchpad provisto con un microcontrolador similar al utilizado en los laboratorios del curso. Se le agregan también un display para el despliegue de las señales y un módulo extra para la red Wifi.

Plataforma de hardware

- Launchpad MSP-EXP432P401R [4] con microcontrolador MSP432P401R [5] de Texas Instruments

Características de interés:

- Procesador Core-ARM® 32-bit Cortex®-M4FCPU
- Frecuencia 48 MHz
- 256kB Memoria Flash
- 64kB Memoria SRAM
- 32kB Memoria ROM
- 2 Timers de 32-bits, con generación de interrupción

- BoosterPack BOOSTXL-K350QVG-S1 [6], con display Kentec QVGA de Texas Instruments

Características:

- 3.5 pulgadas con QVGA (320x240 resolución).
- Interfaz SPI.
- Pantalla táctil resistiva.
- Luz LED blanca de fondo.

- SimpleLink Wi-Fi CC3100 wireless network processor BoosterPack [7] de Texas Instruments

En caso que la comunicación se implemente vía WiFi.

Características de interés:

- CC3100 Wi-Fi Network Procesador en QFN package
- 2 x 20-pin stackable connectors (BoosterPack headers) para conectar TI LaunchPads y otros BoosterPacks
- On-board chip antenna
- 2 botones
- 4 LEDs

- Hardware externo al sistema.

Si bien no forma parte del sistema, para la implementación de la interfaz externa se utilizará también:

- Router de características regulares.
- Computadora con programa MATLAB instalado.

5.2. Software

El sistema fue desarrollado en lenguaje C con un arquitectura de Round Robin con Interrupciones. Esta arquitectura fue elegida dado que era la de menor exigencia que permitía cumplir con los requerimientos de nuestro programa. Se aplicaron los conceptos de modularización y buenas prácticas de programación. Se utilizaron también librerías aportadas por Texas Instruments. La programación del microcontrolador se realizó con el programa Code Composer Studio [8] como IDE, basado en Eclipse. Para programar la interfaz gráfica en ordenador, utilizada para configurar parámetros del sistema, se utilizó el programa Matlab como IDE.

En la figura 3 se representan los módulos de software junto con los equipos de hardware y la capa de abstracción de hardware.

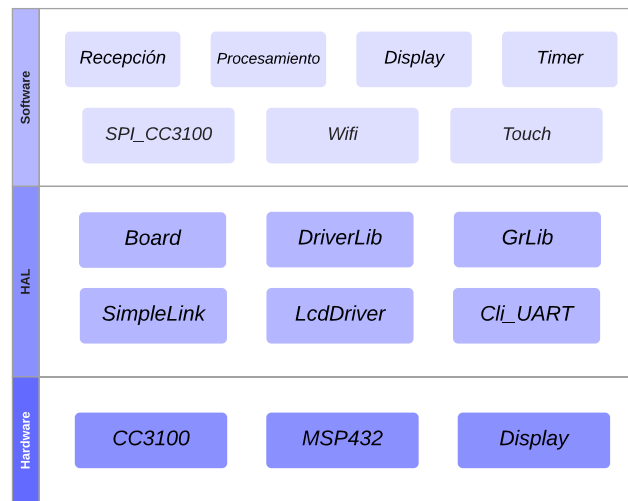


Figura 3: Diagrama de capas del sistema.

5.2.1. Módulos

El sistema fue dividido a nivel de software en módulos, diferenciados por su funcionalidad y tareas. Se desarrollaron 10 módulos en total, de los cuales 4 fueron diseñados completamente por nosotros (recepción, procesamiento, display y timer), mientras que los 6 restantes fueron modificados en base a módulos de proyectos anteriores y de Texas Instruments. Se utilizaron también 3 librerías aportadas por el fabricante.

Los módulos de software pueden agruparse según el destino de sus funcionalidades. De acuerdo a la figura 4, observamos que, además del main, podemos identificar los módulos relativos al tratamiento de datos en color violeta, módulos que refieren al control del display en color azul, y en verde módulos dedicados a la conectividad inalámbrica con la interfaz externa. Las flechas indican el módulo del que se toman funciones, por ejemplo el módulo Display utiliza funciones definidas en el módulo Procesamiento.

Los módulos de la capa de abstracción de hardware aportan funciones a los módulos de software. La comunicación entre software y HAL se observa en la figura 5, donde las flechas que señalan el bloque de módulos de la capa de abstracción están nominadas con los módulos de los que se toman funciones (*e.g.* el módulo Touch utiliza funciones de DriverLib). A su vez, dentro de la capa HAL también existen comunicaciones intermodulares, indicadas por las flechas en la misma figura.

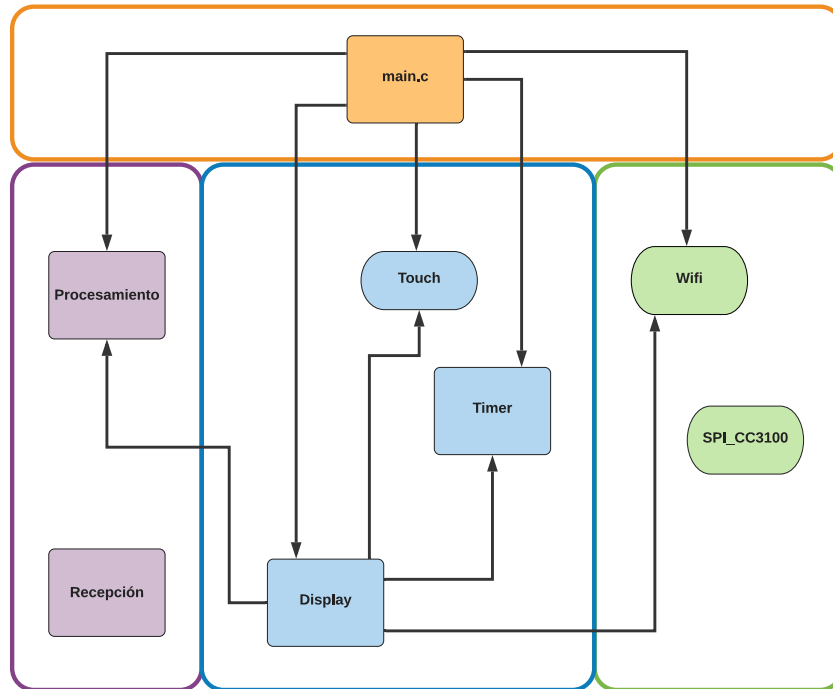


Figura 4: Diagrama de módulos de software representando su comunicación entre sí. Se toma como criterio que el módulo del que se origina la flecha toma funciones del módulo al que llega la misma. No se muestran en el diagrama los módulos correspondientes a HAL.

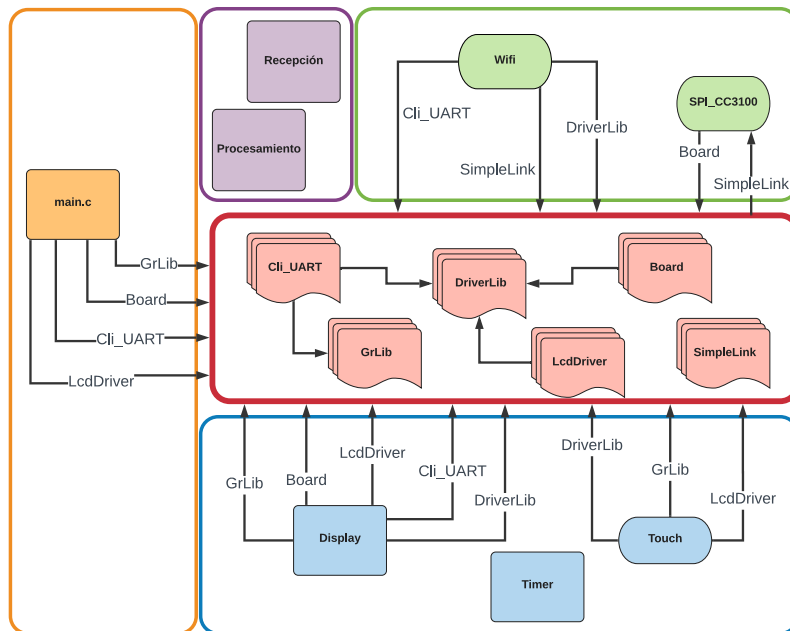


Figura 5: Diagrama representando la comunicación de los módulos de software con HAL. Al igual que en el diagrama anterior, las flechas indican de qué módulo se toman funciones. Los módulos de HAL también comparten funciones entre sí, manteniendo el criterio de flechas.

Módulo Recepción Este módulo se encarga de recibir los datos de las señales y guardarlos en un arreglo de bytes en memoria. Actualmente el sistema toma datos reales pregrabados,

ubicados en la memoria FLASH del microcontrolador. Por esta razón el módulo recepción solo modifica la escala de la gráfica, guardando el valor en variables globales (opcionalmente locales, accediendo desde el exterior del módulo mediante funciones), que serán usadas por el módulo de procesamiento. En un futuro se prevé que en este módulo se construya la recepción de los datos, pudiendo ser vía UART desde un sistema externo o a través de los periféricos del propio Launchpad del sistema.

Para el sistema actual, en el caso de la señal ECG, los datos fueron obtenidos desde el portal Physionet [9] y corresponden a un estudio de 10 segundos, con una frecuencia de 360 muestras por segundo, lo que implica un período de muestreo de 2,8 ms. Para una buena representación de las señales en tiempo real, nuestra frecuencia de despliegue será igual. En los casos de las señales restantes, no fue posible encontrar una base de datos similar, por lo que, a modo ilustrativo, se despliega una señal sinusoidal para IBP y una señal triangular para SpO2, ambas de 3 Hz de frecuencia.

Principales variables¹:

- *scaleECG*
- *scaleIBP*
- *scaleSPO2*

Módulo Procesamiento El módulo de procesamiento toma los datos recibidos y los transforma en valores graficables. Esto es, construye un arreglo de 8 datos, ajusta la ganancia, el offset (para trabajar con números positivos) y la escala, para finalmente guardarlos en memoria RAM. En este módulo también se realizan los chequeos de las alarmas, así como se modifican sus banderas de activación. Algunas de las funciones de este módulo son llamadas por otras del módulo display, a la hora de graficar nuevos datos de las señales. Una de ellas es la función *receivearrayECG*, cuyo flujo se representa en la figura 7.

Principales variables ²:

- *AlarmECG*, *AlarmIBP* y *AlarmSPO2*
- *savedata260ECG*, *savedata260IBP* y *savedata260SPO2*
- *savedataPRmin*, *savedataIBPmin* y *savedataSPO2min*

Principales funciones:

- *receivearrayECG*, *receivearrayIBP* y *receivearraySPO2*
- *checkPR_alarm*, *checkIBP_alarm* y *checkSPO2_alarm*
- *SetAlarmECG*, *SetAlarmIBP* y *SetAlarmSPO2*

¹Si bien las variables son definidas como locales, en módulo procesamiento serán utilizadas definiéndolas con la palabra reservada "*extern*".

²Todas ellas son variables locales al módulo.

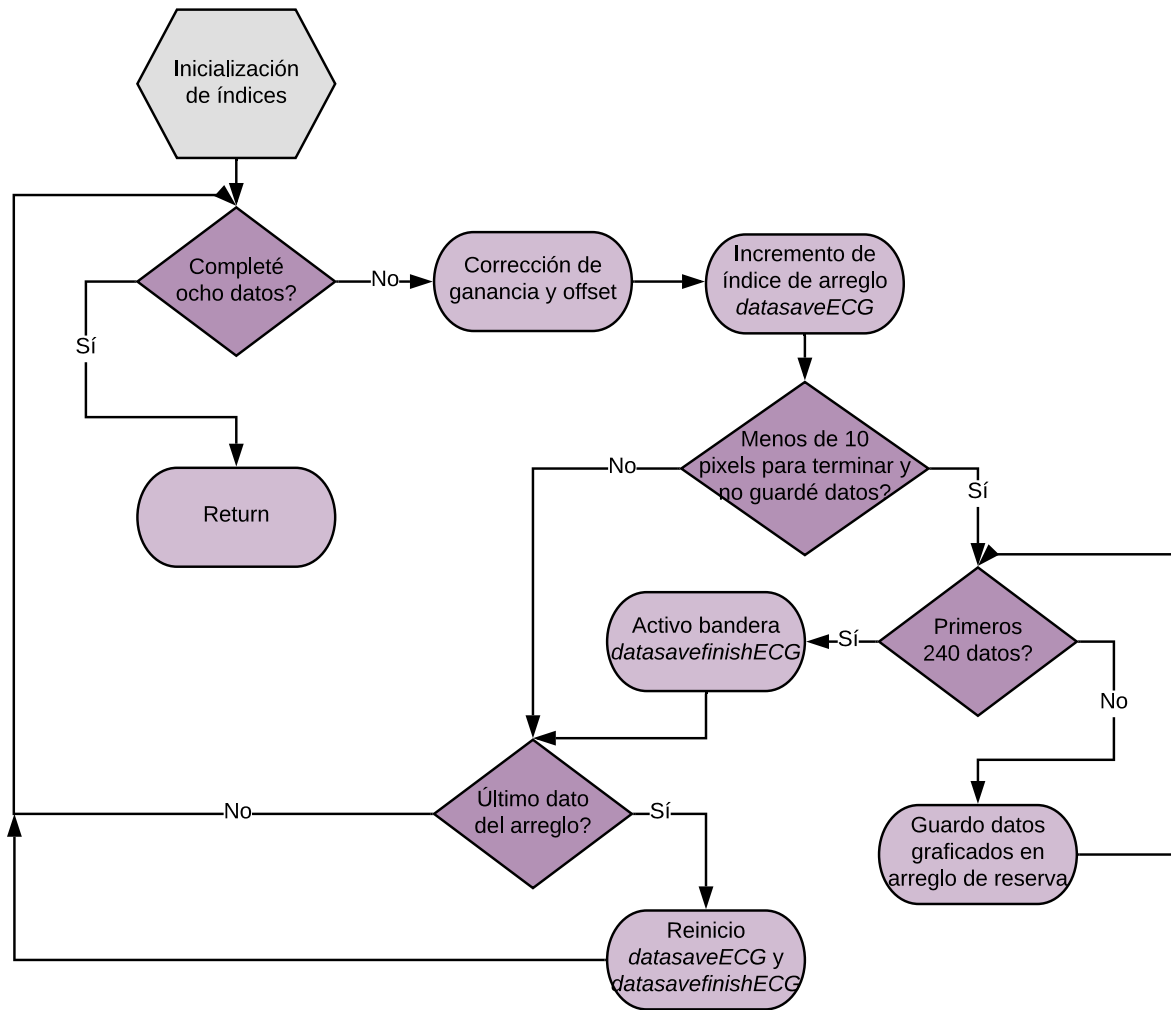


Figura 6: Diagrama de flujo de función *receivearrayECG*.

Módulo Display En este módulo se desarrollan todas las funciones dedicadas a imprimir en pantalla. Estas funciones se encargan de inicializar la pantalla, dibujar las gráficas, los valores numéricos de las señales, así como el menú de ajustes y los gráficos relativos a las alarmas. También se encarga de desplegar el reloj y de manejar los botones táctiles del display. Para implementar estas tareas, toma funciones de los módulos de procesamiento, así como de timer, touch y varias librerías.

Principales variables:

- *AlarmECG_display*, *AlarmIBP_display* y *AlarmSPO2_display*
- *BufferTx* y *BufferRx*
- Estructuras de la librería GrLib: botones y contexto
- Variables de coordenadas de pixels
- Booleanos de selección de botones

Principales funciones:

- *display_graph_ECG*, *display_graph_IBP* y *display_graph_SPO2*
- *display_value_PR*, *display_value_IBP* y *display_value_SPO2*
- *display_init*, *DrawMainMenu*, *DrawSetupMenu*
- *stopAlarm*
- *SetInterfaz*, *refreshCmds*
- Funciones de despliegue de parámetros de alarmas

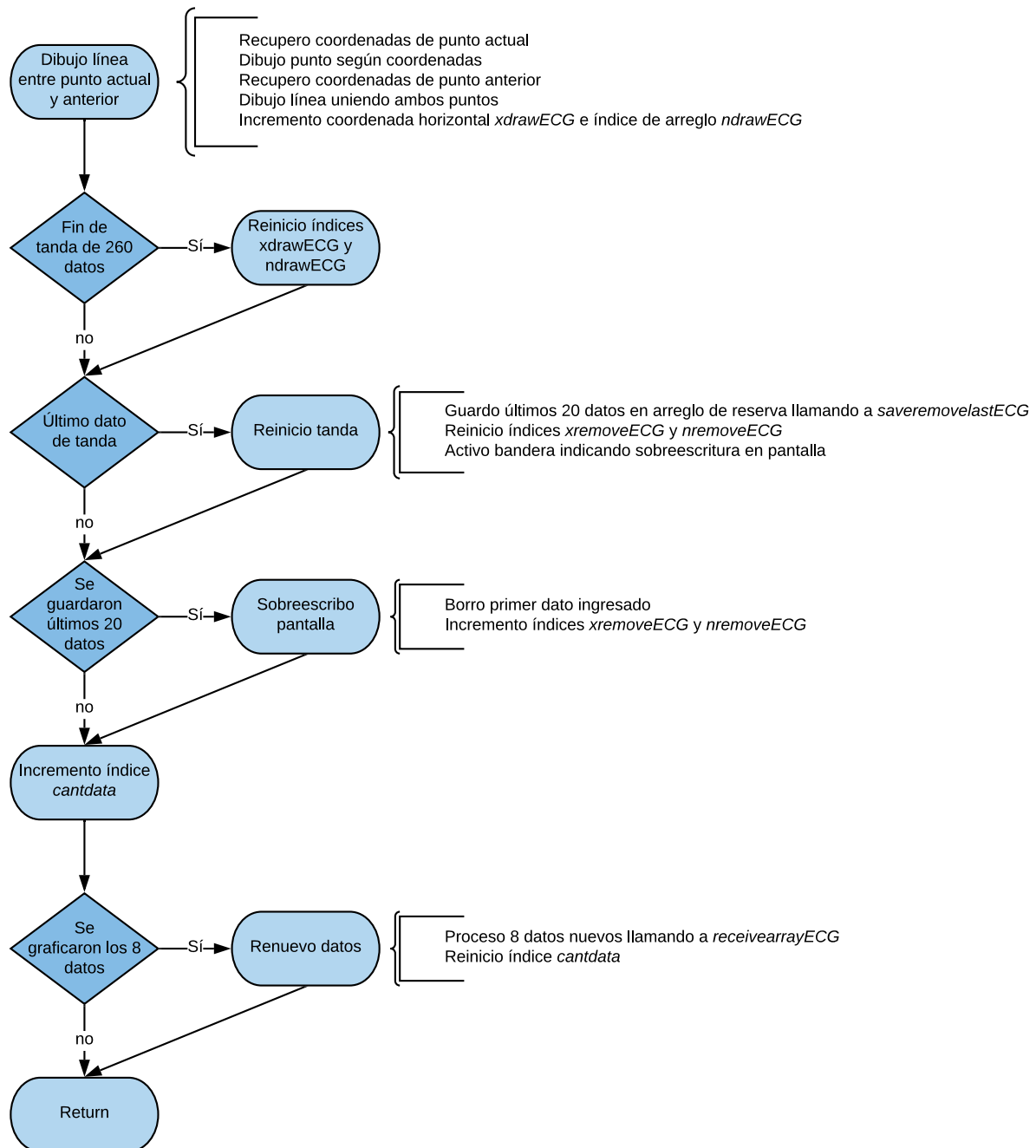


Figura 7: Diagrama de flujo de función *display_graph_ECG*.

Módulo Timer Integrando a nuestro sistema el módulo Timer, creado para uno de los laboratorios del curso, podemos indicarle al microcontrolador con precisión cuándo graficar. Con el mismo Timer se implementarán dos banderas, una con un período de 1 segundo para imprimir los valores numéricos de las señales, y otra con un período de $2,8ms$ para dibujar los puntos de los gráficos. El flujo de la rutina de atención a interrupción se muestra en la figura 8.

Principales funciones:

- *initTimer, SetFlagTimerA0*
- *TA1_0_IRQHandler*
- *set_time*

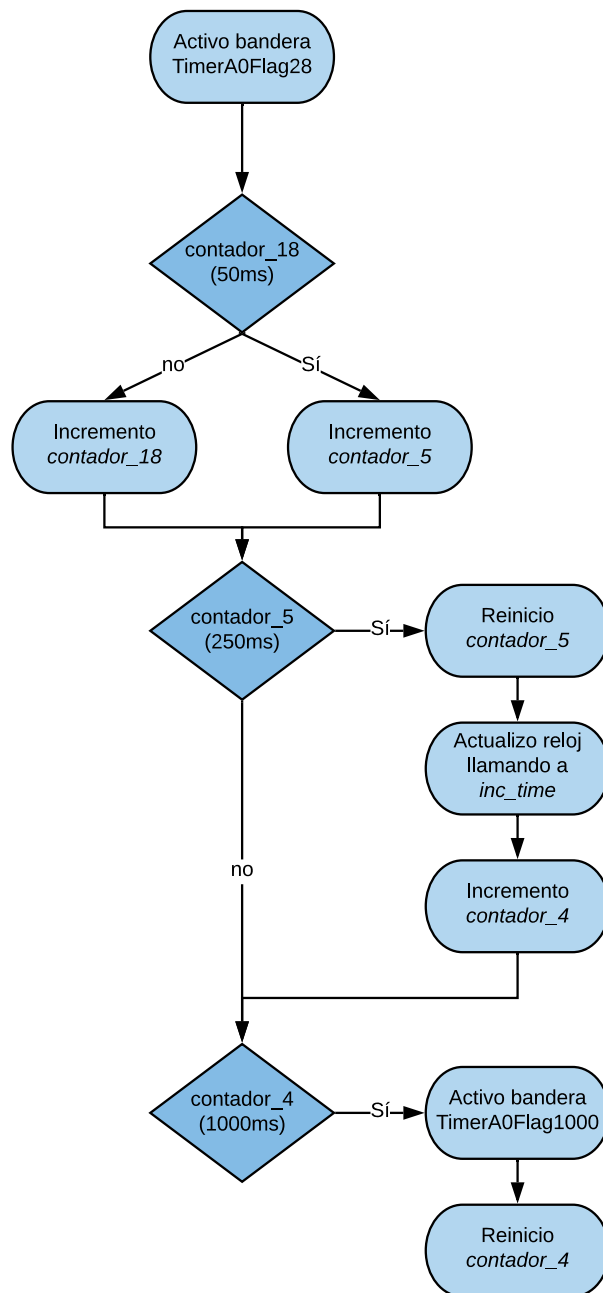


Figura 8: Diagrama de flujo de rutina de atención a interrupción del timer.

Módulo Touch Aportado por Texas Instruments, este módulo se encarga de identificar un toque en la pantalla y recuperar sus coordenadas, devolviendo los datos en una variable global, disponible para otros módulos. Este módulo se apoya fuertemente en la librería GrLib, mientras que sus funciones son utilizadas principalmente por el módulo display.

Principales variables:

- *touch_calibrationData*

Principales funciones:

- *touch_initInterface, touch_updateCurrentTouch*
- *touch_sampleX, touch_sampleY*

Módulo Wifi Una de las opciones para la conexión con la interfaz es vía Wifi, formando una red WLAN con protocolo TCP/IP vía sockets. Si bien la configuración de la red a bajo nivel se realiza a través de una librería del fabricante, en este módulo se desarrollan las funciones que inician y finalizan la conexión. En este módulo también se construyen las funciones *SendDataWifi* y *RecvDataWifi*, mediante las cuales se envían y leen datos transportados por la red.

Principales funciones:

- *InitWifi, WifiClose*
- *SendDataWifi, RecvDataWifi*

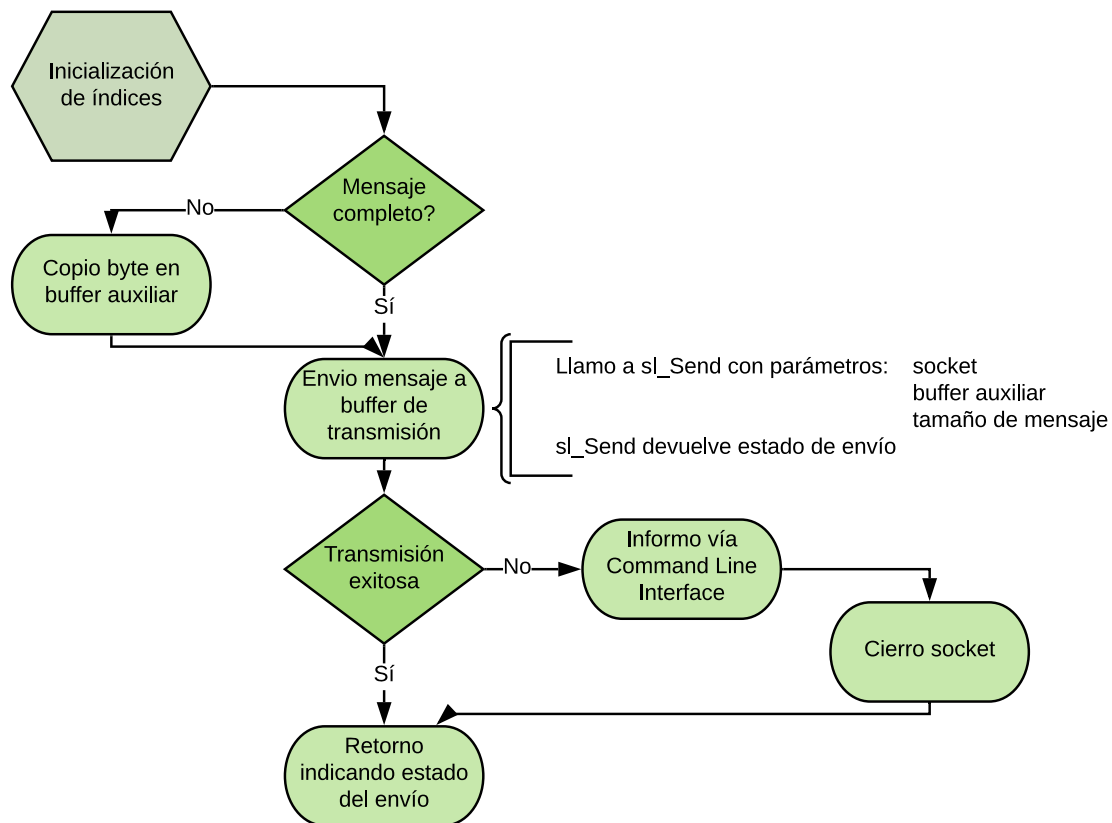


Figura 9: Diagrama de flujo de función *SendDataWifi*.

Módulo SPI Este módulo fue construido por el proyecto DAMAVA, a partir de la recomendación de la librería Simplelink de Texas Instruments. En este módulo se construyen las funciones que configuran, envían, reciben y deshabilitan la transmisión de datos del módulo SPI del Boosterpack CC3100. Dicha transmisión será realizada por el periférico eUSCI B1 del Launchpad.

Principales funciones:

- *spi_Open, spi_Close*
- *spi_Write, spi_Read*

Módulo Board Este módulo es parte de la Hardware Abstraction Layer y es aportado por Texas Instruments. Mediante este módulo se configuran diversas características del microcontrolador, como son las interrupciones, el watchdog timer, LEDs, relojes, etc. A través de las funciones *CC3100_enable* y *CC3100_disable*, se modifican los pines requeridos para habilitar y deshabilitar el Boosterpack.

Principales funciones:

- *CC3100_enable, CC3100_disable*
- *stopWDT, initClk*
- *registerInterruptHandler*

Módulo LcdDriver La configuración a bajo nivel del display se implementa con este módulo. El mismo es obtenido a partir de un demo de Texas Instruments para una aplicación del display. El módulo también forma parte del HAL del sistema, configurando la comunicación SPI mediante la cual (utilizando el periférico B0 del Launchpad) el display recibe comandos de bajo nivel, los pines a utilizar, etc. En este módulo también se implementan las funciones que controlarán los píxeles de la pantalla.

Principales funciones:

- *Kitronix320x240x16_SSD2119Init, HAL_LCD_initLCD*
- *HAL_LCD_selectLCD, HAL_LCD_writeCommand, HAL_LCD_writeData*

Módulo CLI_UART A través del Command Line Interfaz (CLI) se transmitirán mensajes de error o éxito a lo largo del programa. Esta transmisión se realizará mediante protocolo UART, cuyas funciones de control se construirán en este módulo. La misma se realizará mediante el módulo eUSCI A0 del Launchpad. Las funciones de este módulo serán utilizadas principalmente por los módulos Wifi y display.

Principales funciones:

- *CLI_Configure*
- *CLI_Write, CLI_Read*

Librería SimpleLink para CC3100 Esta librería, disponible en Texas Instruments, se encarga de configurar y controlar la conexión y transmisión de datos entre el Boosterpack CC3100 y el exterior. A su vez también se encarga de la comunicación del mismo con el microcontrolador, teniendo SPI y UART como opciones. El fabricante aclara que para esta última comunicación, se deben elaborar las funciones que envíen, reciban y configuren la transmisión de datos, que serán usadas por la librería. En nuestro sistema dicho módulo es denominado SPI_CC3100.

Principales módulos incluidos:

- Device: controla el comportamiento del dispositivo CC3100.
- Netapp: activa aplicaciones de red.
- Netcfg: controla las configuraciones de las direcciones.
- Socket: controla las opciones de programación y capacidades de los sockets.
- Wlan: controla el uso del Wifi (características de conexión, características avanzadas).

Librería Graphics Library Graphics Library es un conjunto de funciones que permiten tanto el control y la configuración del display utilizado, como el despliegue de figuras, texto, botones e imágenes en la pantalla. Esta librería utiliza funciones del módulo LcdDriver para controlar el display, y sus funciones son utilizadas en el módulo Display para implementar los despliegues en pantalla así como controlar los botones.

Principales módulos incluidos:

- Fuentes
- Context
- Button
- Line
- String

Librería DriverLib La librería DriverLib es un conjunto de API's (Application Programming Interface) que permiten configurar y controlar los periféricos del microcontrolador. De esta forma se pueden modificar los registros de bajo nivel a través de funciones que permitan un código mas legible y entendible.

Principales módulos incluidos:

- SPI
- UART
- Interrupt
- WDT
- GPIO

5.2.2. Programa Principal (main)

En el main se encuentra el lazo infinito en el que se desarrollará el programa. De acuerdo a la arquitectura elegida de Round Robin con Interrupciones, en dicho lazo se llaman los handlers de cada bandera. En el diagrama de flujo de la figura 10 se muestra el flujo de ejecución detallando las diferentes banderas que se van verificando si son verdaderas (activas) y los handlers que se ejecutan.

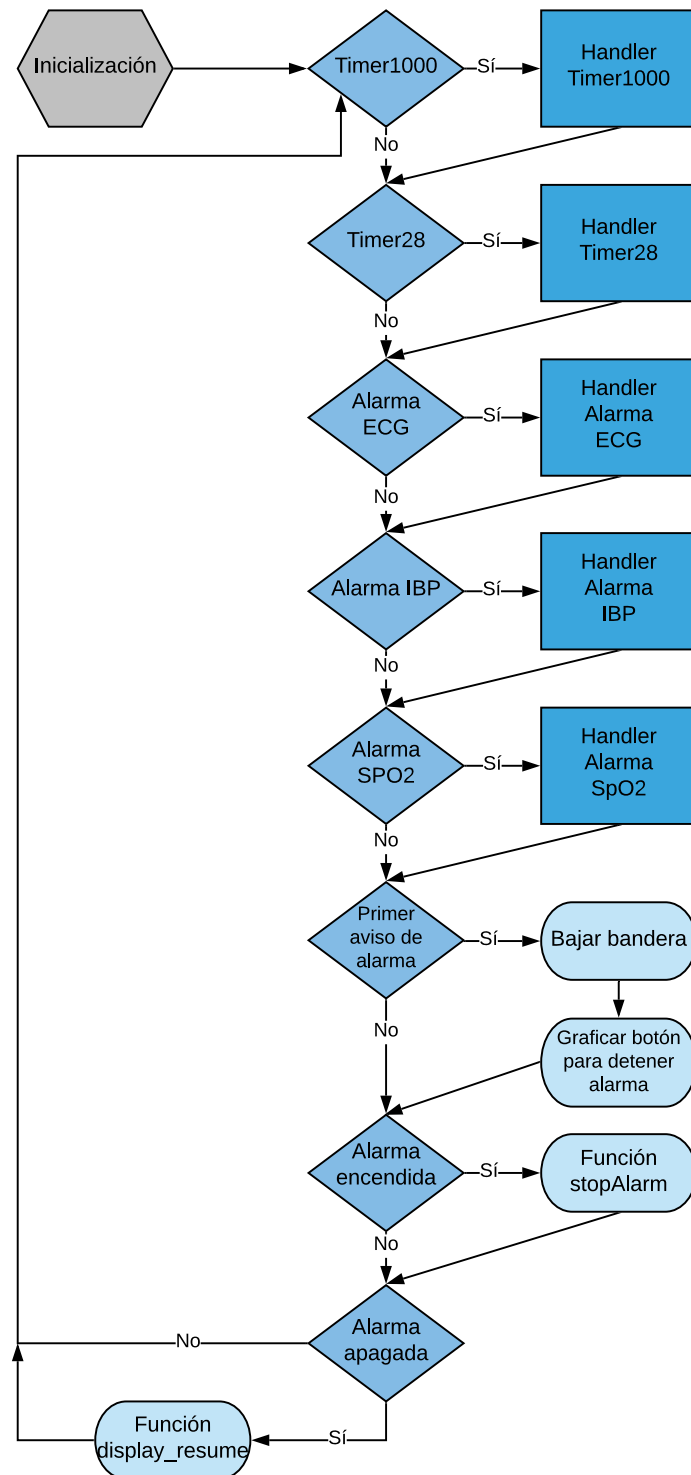


Figura 10: Diagrama de flujo del lazo infinito ubicado en main

El main se compone de handlers a los que se atiende por activación de banderas. A continuación se detallan las banderas y se plantean los principales handlers del sistema en las figuras 11, 12 y 13.

Bandera Timer1000 El timer implementado tiene dos períodos de interrupción, cada 1 segundo y cada 2,8ms. En la interrupción generada a cada segundo, el handler se encarga de imprimir el reloj (si se está en la pantalla principal), imprimir los valores de los parámetros de las señales y resetear el conteo del timer. En el caso de usar conexión vía Wifi con la interfaz externa, también se reciben las configuraciones de la misma. Se puede observar el diagrama de flujo de su handler en la figura 11.

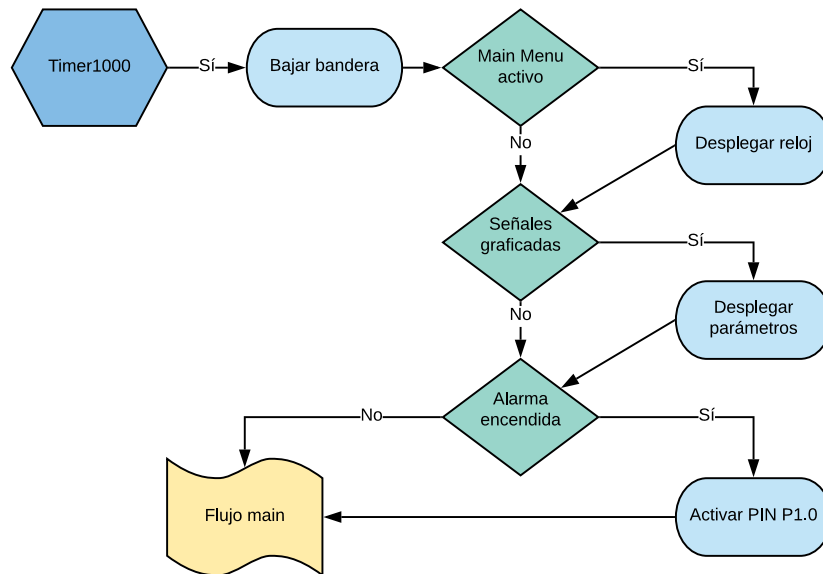


Figura 11: Diagrama de flujo del handler del Timer1000, activado cada 1 segundo.

Bandera Timer28 La segunda bandera del timer interrumpe cada 2,8ms. En este caso su handler, luego de verificar que el boton "START" haya sido presionado, se encarga de graficar las señales vitales. También verifica si se han graficado todos los puntos disponibles de la memoria FLASH, inicializando los índices en caso que esto ocurra. En el futuro se desarrollará la recepción en tiempo real vía UART. En la figura 12 se observa el diagrama de flujo del handler.

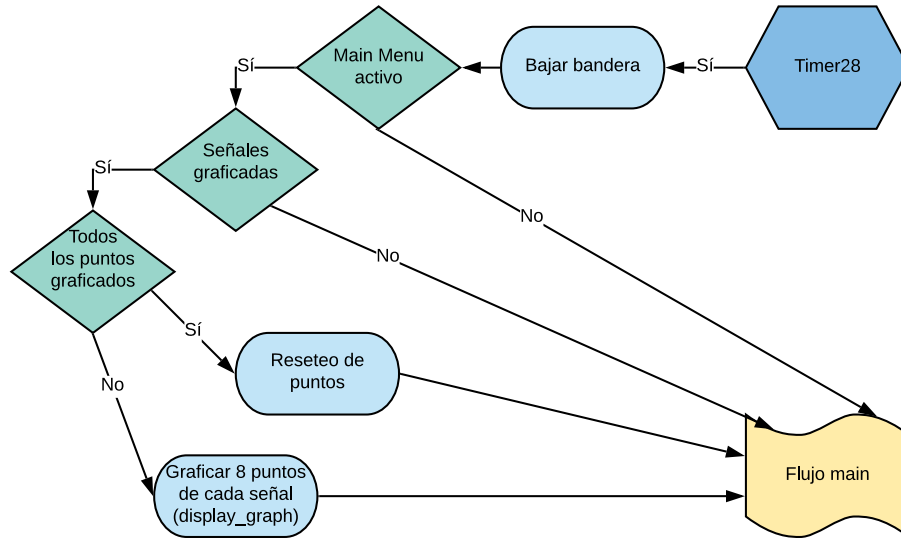


Figura 12: Diagrama de flujo del handler del Timer28, activado cada 2,8ms.

Banderas de Alarmas El módulo Procesamiento se encarga de chequear que los valores a graficar se encuentren dentro del rango establecido de las alarmas. En caso de activarse para alguna de las señales, se levanta la bandera correspondiente y se la atiende en el flujo del main, de acuerdo a la arquitectura del sistema. El handler de las alarmas levanta una segunda bandera que será utilizada para chequear el boton "STOP ALARM" y deshabilitar los botones "START" y "SETUP", además de imprimir en pantalla el texto en color rojo indicando a qué señal corresponde la alarma. Se representan los handlers mediante el diagrama de flujo de la figura 13.

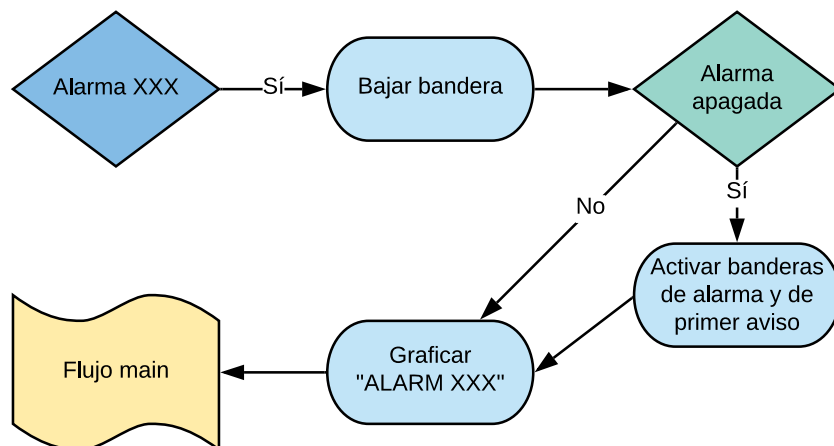


Figura 13: Diagrama de flujo del handler de activación de alarmas.

Bandera de Primer Aviso de Alarma Mediante esta bandera el sistema identifica que aún no ha sido desplegado el botón "STOP ALARM" y lo imprime en pantalla.

Bandera de Alarma Encendida Esta bandera se activa luego de activarse una alarma. Al atender esta bandera, el sistema chequea si el botón "STOP ALARM" fue presionado y reacciona borrándolo o manteniéndolo, según corresponda. Al activarse esta bandera también quedan deshabilitados los botones "START" y "SETUP".

Bandera de Alarma Apagada Luego de desactivar la bandera anterior, la pantalla vuelve a su estado original, quedando atento a los botones "START" y "SETUP" del menú principal.

5.2.3. Interfaz en ordenador

La interfaz en ordenador fue implementada utilizando el IDE Matlab, programando sus componentes con ayuda de la funcionalidad APPS que incluye este IDE. La interfaz, que se puede ver en la figura 14, dispone de varios spinners en los cuales se pueden ingresar los valores de mínimos y máximos de las alarmas de cada una de las señales, además de configurar la hora. Luego de ingresadas las alarmas y la hora, se selecciona el botón "START" y la interfaz abre el server, el cliente se conecta y envía el carácter 'S' a la interfaz, que al recibirlo, envía los datos en un arreglo de 8 bytes, finalmente el cliente envía un carácter 'F' y la interfaz cierra el server al recibirlo. La sección "Monitoring" de la figura 14 tiene como fin mostrar los valores medios de las señales, enviados desde el microcontrolador. Esta funcionalidad está por fuera del alcance de nuestro sistema.

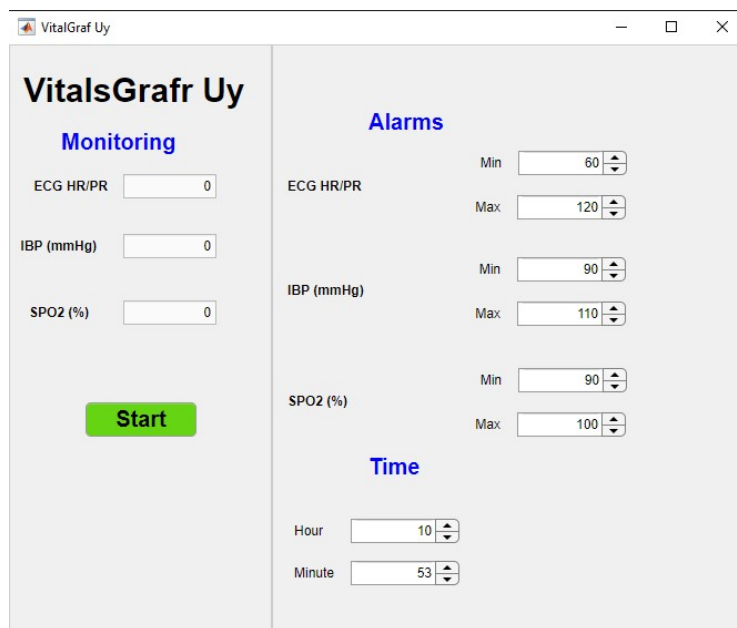


Figura 14: Interfaz en ordenador.

Al presionar el botón START, se ejecuta un archivo de nombre ServerVitals.m, cuyo funcionamiento se describe en el diagrama de flujo de la figura 15.

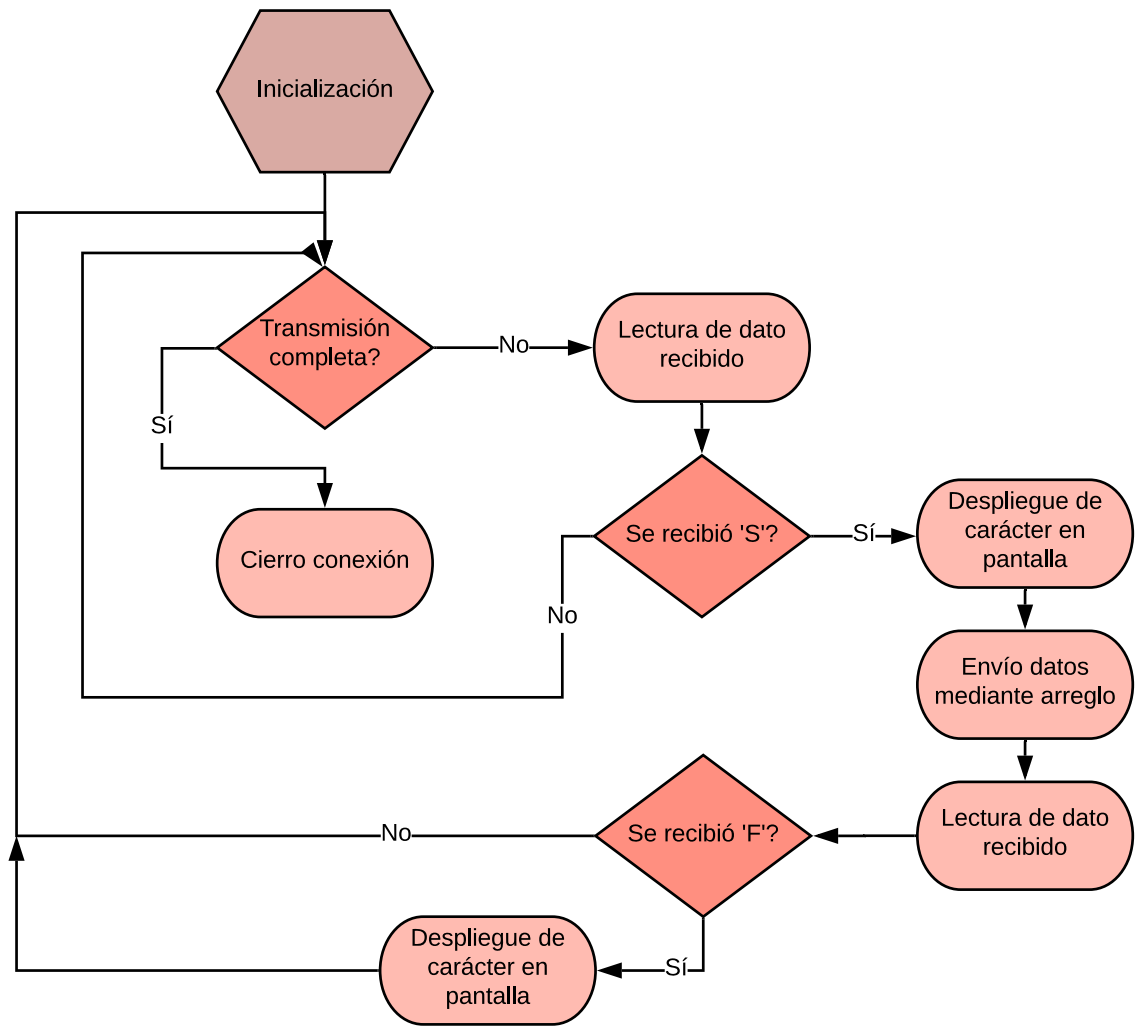


Figura 15: Diagrama de flujo de ServerVitals.m

6. Pruebas y Mediciones

6.1. Pruebas de Módulos

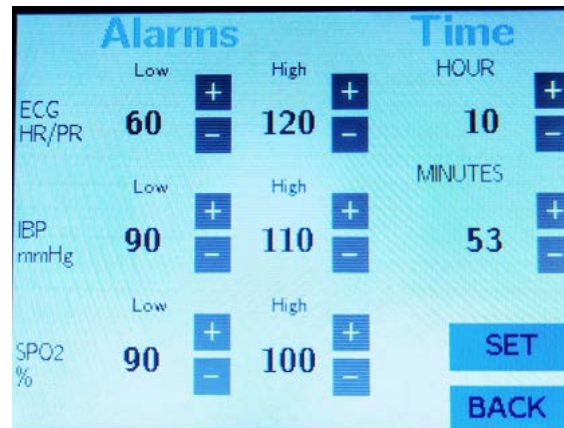
A nivel de software nuestro sistema fue dividido en módulos, por lo que las pruebas de los mismos se dieron de forma acumulativa, agregando los módulos nuevos a los ya verificados.

6.1.1. Módulo Recepción

Originalmente este módulo se encargaba de leer un archivo en formato *CSV* para recuperar los datos de las señales. Testeado usando la herramienta Debug del IDE, encontramos que los datos eran recuperados con éxito. Sin embargo, encontramos un impedimento al no incluir el archivo en la memoria del microcontrolador. Se cambió entonces la estrategia y los datos fueron incluidos de forma directa en la memoria FLASH mediante un archivo header perteneciente a este módulo. Utilizando la herramienta Assembler del IDE, se verifica que los datos son correctamente cargados en el microcontrolador.

6.1.2. Módulo Display

Este módulo toma datos con valores en formato acorde a coordenadas para nuestro display, e imprime en pantalla lo que corresponda. En ese sentido, se tomó como prueba un conjunto de datos ya preparados para la funcionalidad del módulo. Podemos verificar gráficamente que en la pantalla se imprime lo esperado, tomando como ejemplos las figuras 16a y 16b, por lo que el módulo funciona correctamente.



(a) Imagen de inicialización. Se proyecta por 3 segundos antes de desvanecerse y mostrar el menú principal. (b) Menú de ajustes. Se pueden configurar los valores límite de las alarmas así como el reloj.

Figura 16: Ejemplos de impresiones en pantalla del programa.

6.1.3. Módulo Procesamiento

Una vez verificado el módulo display, contamos con la pantalla como herramienta para realizar pruebas. Esta herramienta resulta particularmente útil para las tareas de este módulo.

En ese sentido, tomando los datos incluidos en el módulo recepción, verificamos mediante la herramienta Assembler que los datos eran procesados con éxito y guardados correctamente en la memoria. También pudimos observar visualmente que eran graficados sin inconvenientes. Por último, se modificaron algunos datos para incumplir los límites de las alarmas y forzar la

activación de éstas. Se observa en pantalla entonces la alarma visual correspondiente y el botón para detenerla. En las figuras 17a y 17b podemos ver el resultado gráficamente.



(a) Menú principal visualizando gráficas, parámetros y el reloj. (b) Menú principal con alarmas activadas y botón de desactivación.

Figura 17: Resultado en display de datos procesados y graficados.

6.1.4. Módulo Wifi

Si bien este módulo fue importado de otro proyecto, se hicieron ciertas modificaciones, por lo que también debió ser puesto a prueba. Para ello se dispuso establecer una conexión usando un router como access point, el network processor como cliente y la interfaz en PC como servidor. La conexión fue correctamente verificada.

Una vez confirmada la correcta conexión, se prueba la transmisión de datos. Para ello se estableció un protocolo de envío y recepción de información entre el Launchpad y la interfaz, explicado en la sección 4. Se puede observar en la figura 18 que la transmisión de datos se da de forma exitosa.

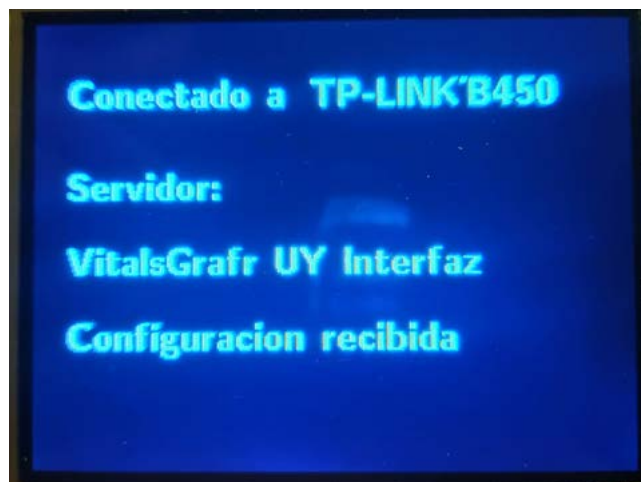


Figura 18: Inicialización de configuraciones de alarmas desde la interfaz externa, mediante conexión Wifi.

6.1.5. Módulos Restantes

El resto de los módulos fueron importados de otros proyectos y del fabricante, por lo que no requirieron pruebas particulares para su uso. De todas formas, se cumplió el procedimiento de verificar su correcto funcionamiento a medida iban siendo integrados a los módulos verificados. El módulo Timer resulta un caso particular ya que fue desarrollado y probado durante los laboratorios del curso.

6.2. Integración de Módulos

Dadas las características de las pruebas realizadas, la integración de los módulos fue realizada a medida que se probaba y verificaba cada módulo. En la mayor parte de los casos no surgieron inconvenientes, adaptándose sin problemas, mientras que en algunos se tuvieron que realizar pequeños ajustes para una correcta incorporación.

De todas formas, si bien los módulos Wifi y display funcionaron de forma correcta y cumplieron sus respectivas pruebas, encontramos que al implementarlos en conjunto no se obtenían los resultados esperados. En particular, una vez iniciada la conexión con el punto de acceso, observamos que al llamar ciertas funciones intermedias se invalidaba toda tarea relativa a la transmisión. El sistema pudo ser adaptado a la situación modificando el uso de la interfaz y su comunicación con el dispositivo. En ese sentido se determinó que la comunicación con la interfaz se realice antes de la inicialización del programa, permitiendo así configurar los parámetros correspondientes desde la interfaz externa.

6.3. Reporte de Uso de memoria

Como se indicara en las secciones previas del documento, el sistema utiliza recursos tanto de la memoria FLASH como la memoria RAM. En la figura 21 se representa el uso de la misma.

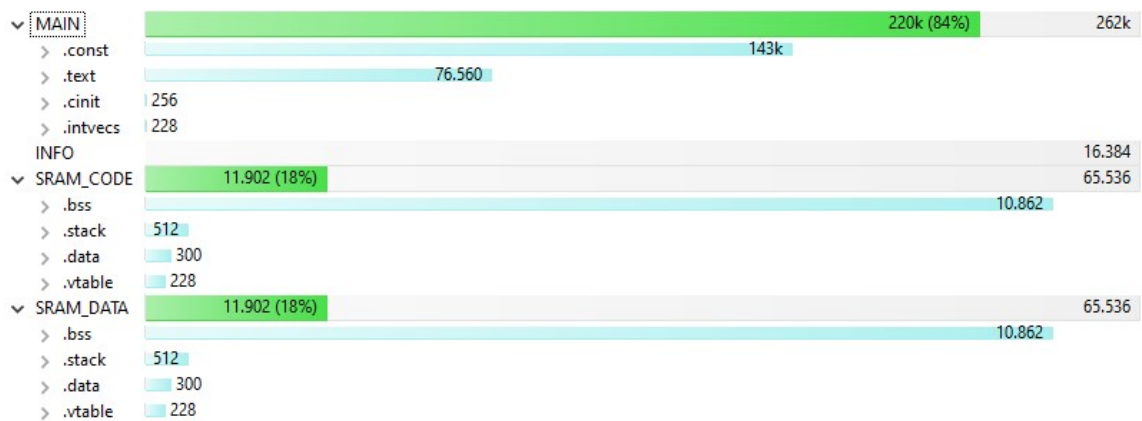


Figura 19: Uso de memoria.

De los 220 kB usados en la FLASH, 78.6 kB corresponden a la imagen de inicio de la interfaz y 43.2 kB a las señales pregrabadas.

7. Conclusiones

Hemos presentado un sistema que toma datos de señales vitales y los despliega en pantalla, tanto en formato gráfico como numérico, incluyendo también un sistema de alarmas visuales y una interfaz externa para la configuración de ciertos parámetros. Entendemos que el proyecto está completo y se satisficieron los objetivos generales planteados.

A lo largo del desarrollo del proyecto se plantearon situaciones problemáticas que debieron ser resueltas para la realización del sistema. Algunas de ellas fueron solucionadas, otras fueron evitadas tomando opciones alternativas y algunas desembocaron en la desestimación de algunas funcionalidades.

Un problema que logramos solucionar por completo surgió al integrar el sistema de alarmas con el resto del programa. En particular, al agregar el botón "STOP" para detener las alertas debimos deshabilitar los botones "START" y "SETUP". El inconveniente surge al seleccionar el botón "STOP", dado que el programa no volvía a quedar atento al toque de los botones restantes. Esta situación fue solucionada agregando la bandera de alarma apagada (detallada en la sección 5.2) y la función *display_Resume*.

En otro caso, encontramos una alternativa a la transmisión de datos entre la interfaz y el Launchpad. El sistema final plantea la conexión Wifi previo a la inicialización del programa, permitiendo así la comunicación pretendida. Esta situación es consecuencia del inconveniente surgido al momento de integrar los módulos Wifi y display, como fue explicado en la sección 6. De esta forma, sorteamos el problema evitando el uso de funciones relativas al display durante la conexión. Queda planteada la posibilidad a futuro de perfeccionar esta funcionalidad, manteniendo las funciones y estructuras que permiten la comunicación.

Por último, fue descartado un procedimiento que consideramos interesante y provechoso para la importación de los datos pregrabados de la señal ECG. En un principio teníamos planteado trabajar a partir de un archivo de formato CSV. Este método daba una cantidad de beneficios dado que aportaba, entre otros, una gran flexibilidad en el manejo de los datos, editándolos dentro del archivo o disponiendo de varios archivos. Sin embargo, al no poder cargarlo al microcontrolador, nos vimos forzados a cambiar el procedimiento. Finalmente resolvimos importar los datos directamente a la memoria FLASH mediante un archivo header encontrando buenos resultados.

En cuanto a la escalabilidad, consideramos que el dispositivo permite cambios en su hardware. Si bien la programación se apega a la familia de microcontroladores MSP430 y MSP432 (dado el fuerte uso de sus librerías), entendemos que es posible cambiar el display al modificar el módulo Touch. Así mismo, vemos que el Boosterpack CC3100 puede ser intercambiado al modificar el módulo SPI. Sin embargo, el uso de dicho hardware está estrechamente relacionado con la librería SimpleLink, por lo que se debe estar atento a su compatibilidad.

Como trabajos a futuro quedan planteadas algunas opciones. En primer lugar, entendemos que es posible mejorar la funcionalidad de la comunicación Wifi entre la interfaz externa y el dispositivo. Por otro lado, queda a futuro la implementación de una recepción de datos de señales externa, con comunicación UART con un sistema adquisidor, o bien mediante una adquisición interna con periféricos del Launchpad utilizado.

8. Anexo

8.1. Conceptos del curso aplicados al proyecto

Para realizar este proyecto se aplicaron varios conceptos dados a lo largo del curso. En primer lugar se utilizó GIT como sistema de control de versiones, para escribir el código de manera organizada, y gitlab para mantener el código sincronizado entre los diferentes miembros del equipo.

También se aplicó el concepto de interrupciones a lo largo del código escrito en lenguaje C. En ese sentido, manejamos las rutinas de atención a interrupción y consideramos que no tenemos problemas de datos compartidos.

Pudimos aplicar también los conocimientos sobre arquitecturas de software, siendo que se trabajó con Round Robin con interrupciones. Consideramos que esta arquitectura era la mas simple que se adecuaba a las necesidades de nuestro sistema. En este caso pudimos diseñar un flujo en el archivo main que permitió un eficaz desarrollo del programa.

Por último, consideramos que el concepto de modularización fue aplicado enfáticamente en este proyecto. El desarrollo se benefició de este método al obtener un código entendible, con una clara división de tareas y funciones, y permitiendo una buena escalabilidad. A su vez, la modularización aportó una simplificación en las pruebas realizadas, lo que significó una integración general del sistema más rápida, sencilla y con mejores resultados.

8.2. Planificación del proyecto

	Hasta hito intermedio	Hasta entrega final
Planificado	Estudio de Launchpad y Boosterpack. Módulo SPI Estudio de display (demo). Pruebas Launchpad+WiFi+Display Módulo UART Módulo de display Módulo de procesamiento de datos. Manejo de botones de display.	Interfaz usuario-ordenador Módulo de ingreso de datos. Integración de módulos. Pruebas Finales Documentación
Ejecutado	Estudio de Launchpad y Boosterpack Estudio de display (demo) Pruebas Launchpad+WiFi+Display Módulo de display Módulo UART Módulo de procesamiento de datos Manejo de botones de display	Interfaz usuario-ordenador Módulo de ingreso de datos Integración de módulos Pruebas finales Documentación

Las tareas fueron ejecutadas de acuerdo a lo planificado.

8.3. Especificación del proyecto

Bibliografía

- [1] *Operator's Manual: DPM 4 Patient Monitor*. [Online]. Available: https://www.mindraynorthamerica.com/cmsAdmin/uploads/h-046-000180-00-11-0_dpm4-ops-manual.pdf
- [2] V. Cabrera, M. Lopassio, and M. Peña, "Gurí-Z," Proyecto de fin de Curso SiSem, Universidad de la República (Uruguay). Facultad de Ingeniería., 2019. [Online]. Available: https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/G7GuriZ.pdf
- [3] M. Mendívil, H. Marichal, and F. Vique, "DAMAVA," Proyecto de fin de Curso SiSem, Universidad de la República (Uruguay). Facultad de Ingeniería., 2017. [Online]. Available: https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/memoriadanava.pdf
- [4] *MSP-EXP432P401R User's Guide (Rev. F)*. [Online]. Available: <https://www.ti.com/lit/ug/slau597f/slau597f.pdf>
- [5] *MSP-EXP432P401R Datasheet (Rev. H)*. [Online]. Available: <https://www.ti.com/lit/ds/slas826h/slas826h.pdf>
- [6] *BoosterPack BOOSTXL-K350QVG-S1*. [Online]. Available: <https://www.ti.com/tool/BOOSTXL-K350QVG-S1>
- [7] *CC3100 SimpleLink User's Guide*. [Online]. Available: <https://www.ti.com/lit/ug/swru375d/swru375d.pdf>
- [8] *Code Composer Studio (CCS) Integrated Development Environment (IDE)*. [Online]. Available: <http://www.ti.com/tool/CCSTUDIO>
- [9] PhysioBank ATM. Base de datos de ecg. [Online]. Available: <https://physionet.org/>

Nombre del proyecto

VitalsGrafr Uy

Nombres de Integrantes y Tutor

Integrantes: Diego Alonso
Walter Barreiro
Germán Minetti
Tutor: Leonardo Steinfeld

Descripción del Problema a Resolver

Ante la actual crisis sanitaria producida por la pandemia causada por COVID-19, Uruguay se enfrenta a la necesidad de ampliar los Centros de Tratamiento Intensivo. Esto implica adquirir más equipamiento e insumos que actualmente son escasos a nivel local y que, debido a la demanda mundial, también resultan difíciles de importar, aumentando su costo.

Los monitores de signos vitales son equipos fundamentales, requeridos y de uso obligatorio en los Centros de Tratamiento Intensivo (CTI). A través de este proyecto asumimos el desafío de buscar soluciones alternativas al mismo. Nos enfocaremos en el monitoreo de tres señales vitales: ECG (Electrocardiograph), IBP (Invasive Blood Pressure) y SpO2 (Pulse Oxygen Saturation).

Antecedentes

DPM 4 Patient Monitor

Hemos decidido seleccionar este modelo de monitor como referencia para nuestro diseño. El mismo cuenta con display, botones para comando del mismo (*e.g.* seteo de alarmas) y en algunos casos incluye pantalla táctil. Comúnmente cuentan con comunicación para configuración y/o envío de datos a un equipo central.

Este monitor de señales portátil está diseñado para el seguimiento de parámetros fisiológicos, además del registro y alertas de los mismos, en pacientes adultos y pediátricos. El monitor cuenta con un display de alta resolución que permite el despliegue de gráficas, valores y alarmas visuales, además de botones para la navegación e incluso botón de emergencia para solicitar atención médica.

El equipo es capaz de monitorear los siguientes parámetros:

- ECG: Heart rate (HR) - ECG waveform(s) - ST segment analysis - Arrhythmia analysis
- RESP: Respiration rate (RR) - Respiration waveform
- SpO2: Pulse oxygen saturation (SpO2) - Pulse rate (PR) - SpO2 plethy
- NIBP: Systolic pressure (NS), diastolic pressure (ND), mean pressure (NM) - Pulse rate (PR)smogram
- TEMP: Temperature of channel 1 (T1), temperature of channel 2 (T2) - Temperature differential between two channels (TD)

- IBP: 2 channels of IBP waveforms - Systolic (SYS), diastolic (DIA), and mean (MEAN) pressure - Pulse rate (PR)
- CO2: End-tidal carbon dioxide (EtCO2) - Fractional inspiratory carbon dioxide (FiCO2) - Air-way Respiration Rate (AwRR) Functions

Por último, el monitor también emite alarmas audibles o visuales ante un signo vital anormal, o ante problemas tanto mecánicos como eléctricos que puedan presentarse.

A continuación presentamos dos proyectos previos que nos será de gran utilidad para nuestro proyecto.

Proyecto GuríZ

Incluimos este proyecto de fin de carrera puesto que cuenta con un microcontrolador MSP430F5529LP y el display BOOSTXLK350QVG-S1, siendo éste último necesario en nuestro proyecto para el despliegue de señales en pantalla.

El proyecto diseña e implementa un dispositivo médico destinado a medir la impedancia eléctrica transtorácica en pacientes pediátricos. El dispositivo es capaz de recibir señales analógicas de tensión y digitalizarlas, promediar las muestras y a partir de éstas realizar cálculos, desplegándolos en un monitor.

El hardware se basa en el LaunchPad MSP-EXP430F5529LP y el BoosterPack BOOSTXLK350QVG-S1. Para la interacción con el usuario se emplea dos botones para navegación y un display, para el cual se utilizó la librería gráfica disponible en Texas Instruments.

El equipo además permite guardar datos para descargar en PC. La comunicación se realiza vía UART.

Se plantea a futuro el desarrollo de una versión con touchscreen con el fin de facilitar la navegación.

Proyecto DAMAVA

Muchos sistemas embebidos cuentan con comunicación vía Wifi para el envío y recepción de datos, así como lo hace este proyecto. Además, el mismo implementa comunicación SPI entre los dos Launchpad utilizados.

Este proyecto consiste en el desarrollo de módulos de comunicación y adquisición de datos de un sistema para la detección de la mastitis vacuna, basado en el monitoreo de la conductividad y temperatura en la leche.

A través de sensores adquiere medidas enviando los datos mediante WIFI a una PC, empleando también una interfaz de usuario. El sistema está compuesto por un LaunchPad MSP-EXP432P401R, un LaunchPad SimpleLink WIFI CC3100 y un router.

El diseño del sistema permite que pueda ser utilizado por otras aplicaciones que requieran procesos de adquisición y envío de datos, cambiando únicamente la lógica de algunos módulos.

A la hora de aplicar este proyecto en nuestro trabajo, debemos tener en cuenta algunos de los aspectos a mejorar que mencionan los autores, entre los cuales están:

- Mejorar la comunicación haciendo que el CC3100 interrumpa de forma asíncrona
- Se podrá evitar la utilización de un router con motivo de evitar costos, utilizando comunicación Bluetooth en vez de WIFI

Objetivos

El objetivo principal del proyecto es dar una alternativa a la falta de equipamiento en los Centros de Tratamiento Intensivo, dada por el costo y dificultad logística de obtener los equipos en el exterior.

En ese sentido se pretende desarrollar un monitor que reciba, procese y despliegue en un display ciertos datos digitales relativos a un paciente de CTI.

Se busca diseñar un conjunto de alarmas que den aviso a distintas situaciones, ajustables por el usuario.

Tenemos también como fin, implementar una comunicación no cableada vía WiFi con el usuario para el seteo de alarmas y hora y para recepción de datos. Como alternativa, se estudiará la posibilidad de implementar un protocolo UART para ésta comunicación.

Alcance

El sistema a diseñar recibirá datos procesados (digitales) a partir de las señales del paciente.

Los valores *mínimo*, *medio* y *máximo* también serán recibidos y se desplegarán en el display.

También se construirán las gráficas de estado de cada señal, agregando los datos nuevos a los anteriores, y se desplegarán en el monitor. Estas gráficas serán construidas con una cantidad fija de datos, eliminando los datos más antiguos al ingresar datos nuevos, dando así un efecto visual de movimiento a la gráfica.

Por último se verificará si los nuevos datos corresponden a la activación de alarmas. Estas alarmas serán de tipo sonoro y visual.

El sistema será desarrollado en base al Launchpad MSP-EXP432P401R y el BoosterPack CC3100. Para el manejo del monitor se utilizará el Kentec QVGA BOOSTXL-K350QVG-S1 Display BoosterPack.

Se contará con comunicación entre el sistema y una aplicación a usar en PC o celular. Esta comunicación se hará vía WiFi en primera instancia, manteniendo como opción alternativa una comunicación UART.

El usuario tendrá la posibilidad de setear manualmente las alarmas, así como la hora actual (horas y minutos). Para ello se dispondrá de la aplicación y de los botones touchscreen del display.

El sistema no incluye la adquisición de señales eléctricas ni su conversión analógico-digital. Tampoco incluye grabación de señales ni datos del paciente.

Hardware adicional: Se agregará un buzzer para las alarmas, el display Kentec TFT LCD K350QVG-V2-F y el Launchpad CC3100 para el control de WiFi.

Descripción del Sistema

El sistema a desarrollar, deberá graficar en display las siguientes tres señales: ECG (Electrocardiograph), IBP (Invasive Blood Pressure) y SpO2 (Pulse Oxygen Saturation). Mostrará en display valores mínimos, medios y máximos de frecuencia cardiaca, presión de sangre y del pulso

de saturación de oxígeno. Tanto los valores de las gráficas y los valores calculados, mencionados anteriormente, son datos conocidos, provenientes de un sistema exterior que se encarga de la digitalización y cálculos.

El sistema ajustará los valores a graficar a la escala del display, ya definida, para una correcta visualización de las gráficas. En ese sentido se plantea construir una gráfica que sea capaz de mostrar un período fijo de tiempo, definiendo un valor de pixels por unidad de tiempo. De esta manera, visualmente la gráfica permanece independiente de posibles variaciones en la frecuencia de muestreo. Dicha frecuencia de muestreo con la que se obtuvieron los datos, será otro parámetro a recibir por nuestro sistema.

En el display también se mostrarán los valores mínimo, medio y máximo de cada señal, así como el tiempo (hora, minutos y segundos).

El sistema contará con botones en display (touchscreen) para configuración de alarmas, selección de la frecuencia de muestreo y ajustar el tiempo actual.

El sistema enviará a través de WiFi al ordenador los datos de los valores medios, máximos y mínimos de las tres señales. Desde el ordenador también se podrá setear las alarmas y la hora mediante una interfaz a diseñar.

Cuando es detectado un valor fuera del rango definido por el usuario, activará alarma sonora (buzzer) y lumínica (led). Estos valores serán configurables a través del touchscreen o de la PC.

Se investigará si es posible la comunicación via WiFi al ordenador. En caso de encontrar dificultades técnicas fuera de nuestro alcance y del alcance del curso, optaremos por implementar una comunicación vía UART.

Diagrama de Bloques

En la imagen observamos una representación gráfica del hardware y su conectividad. Vale aclarar que la comunicación entre el microcontrolador y el ordenador será vía WiFi (protocolo SPI) como primera opción. Dejamos como opción alternativa la comunicación UART, en caso de encontrar dificultades insalvables en el uso del protocolo SPI.

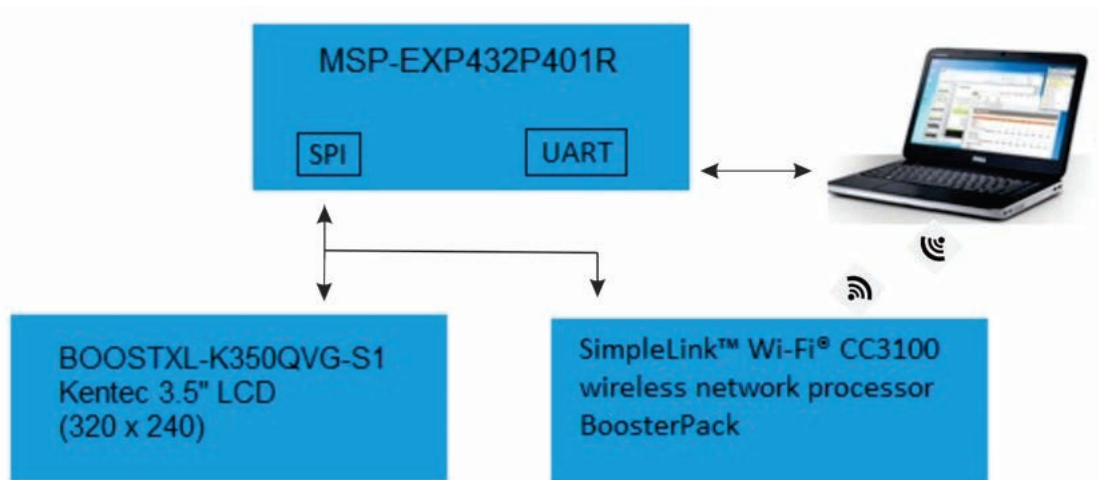


Figura 20: Diagrama de bloques del sistema a implementar.

Requerimientos y Restricciones del Sistema

La utilización del display, implica la elección de un microcontrolador que cumpla con los requerimientos necesarios para que el display pueda usarse con todas sus funcionalidades. La librería de Texas Instruments ofrece un código demo muy completo para el display, asegurando su correcto funcionamiento con los microcontroladores publicados. De las opciones publicadas hemos elegido el microcontrolador con mayor capacidad y velocidad, el MSP432P401R.

Procesamiento y memoria

Estimamos que la velocidad de procesamiento será suficiente para recibir los datos del sistema exterior, mostrar en display la información necesaria (incluidas las tres gráficas en el tiempo, a la frecuencia máxima de muestreo) y enviar los datos al ordenador.

El microcontrolador tiene un procesador ARM 32-bit Cortex®-M4 y frecuencia hasta 48 MHz.

Respecto a la memoria, este microcontrolador dispone de:

- 256KB memoria flash
- 64KB memoria SRAM
- 32KB memoria ROM

Estimamos que estas características serán suficiente para los buffers de recepción, de envío y auxiliares, para los datos a graficar y a transmitir así como para el manejo del display con todas sus funcionalidades.

Tiempos de respuesta

Nuestro sistema presenta dos restricciones de tiempo.

En primer lugar tenemos la construcción de los gráficos en display de las tres señales, con elección de la frecuencia de muestreo y sus valores particulares (medio, máximo y mínimo).

Por otro lado también debemos cumplir con el envío de los valores particulares de las tres señales al ordenador.

Todos los valores se visualizarán en tiempo real, por lo que el sistema debe ser capaz de procesar los datos y enviar o desplegar la información con mayor rapidez que la recepción de datos.

Diseño Preliminar

Plataforma de hardware

- Launchpad MSP432P401R con microcontrolador MSP432P401R de Texas Instruments

Características de interés:

- Procesador Core-Arm® 32-bit Cortex®-M4F CPU
- Frecuencia 48 MHz
- 256KB Memoria Flash
- 64KB Memoria SRAM
- 32KB Memoria ROM

- 2 Timers de 32-bits, con generación de interrupcion
- BoosterPack BOOSTXL-K350QVG-S1, con display Kentec QVGA de Texas Instruments
Características:
 - 3.5 pulgadas con QVGA (320x240 resolución).
 - Interfaz SPI.
 - Pantalla táctil resistiva.
 - Luz LED blanca de fondo.
- SimpleLink Wi-Fi CC3100 wireless network processor BoosterPack de texas Instruments
En caso que la comunicación se implemente vía WiFi.
Características de interés:
 - CC3100 Wi-Fi Network Procesador en QFN package
 - 2 20-pin stackable connectors (BoosterPack headers) para conectar TI LaunchPads y otros BoosterPacks
 - On-board chip antenna con opción para U.FL-based testing
 - Power de on-board LDO usando USB o 3.3V del MCU LaunchPad
 - 2 botones
 - 4 LEDs
 - 0.8 megabit serial flash
 - 40 MHz crystal, 32 KHz crystal y oscillator
 - U.FL y chip antenna
 - USB
- Buzzer conectado a salida del Launchpad MSP432P401R
- LED de Launchpad MSP432P401R
Se utilizará para señalización de alarma.

Arquitectura de Software

El proyecto será implementado utilizando el IDE Code Composer Studio versión 8.3.0.00009, para el desarrollo de aplicaciones de Texas Instruments. Para el display se utilizarán las librerías disponibles por Texas Instruments.

La arquitectura de software que utilizaremos es Round-Robin con interrupciones. Esta arquitectura es bien conocida y estudiada por nuestro grupo en los laboratorios del curso. Esta arquitectura se caracteriza por tener un mayor control sobre los niveles de prioridad, ser flexible y tener una rápida respuesta a las entradas de señales. Utilizando Round-Robin con interrupciones, las rutinas de atención a interrupción (ISR) tendrán prioridad frente al código de la tarea (handler). Por otro lado, esta arquitectura presenta problema con el uso de datos compartidos entre las interrupciones, por lo que se deberán aplicar técnicas vistas en el curso para evitar fallas en el código. Se esperan interrupciones de los Timers (para gráfico de los pixels y despliegue del tiempo), los botones de entrada del display y de envío y recepción de datos por Wifi o UART, según se disponga.

No utilizaremos modo bajo consumo en ningún momento puesto que el estar inactivo puede ser identificado como estar conectado a un paciente sin signos vitales. El display estará activo en todo momento.

Planificación

Actividades

- Estudio del funcionamiento del Launchpad MSP432P41R con Boosterpack CC3100 (sin display).
- Módulo de comunicación SPI.
- Estudio del funcionamiento del display.
- Pruebas de funcionamiento del Launchpad con WiFi y display.
- En caso de no poder implementar la funcionalidad de WiFi con el uso del display: Módulo de comunicación mediante UART.
- Módulo para manejo del display.
- Módulo de procesamiento de datos a desplegar.
- Manejo de los botones a utilizar por el usuario en display.
- Interfaz con el usuario con el ordenador en plataforma.
- Módulo para ingreso de datos (seteo de alarma y hora) y visualización de valores.
- Integración de los módulos.
- Pruebas de funcionamiento.
- Documentación.

Pruebas a realizar

Se hará funcionar el Launchpad MSP432P401R con Boosterpack CC3100. Se visualizará en un terminal la comunicación Wifi. La comunicacion entre Launchpad y Boosterpack es con protocolo SPI.

Se hará funcionar el código demo del display que está publicado en la librería en el Launchpad, sin el Boosterpack CC3100. La comunicación también es con SPI.

Se realizarán pruebas de funcionamiento de Launchpad con Boosterpack y display. Se estudiará y analizará la comunicación SPI de los tres dispositivos, usando las funcionalidades que presentamos anteriormente.

En caso de presentar problemas técnicos por la incorporación de la funcionalidad de WiFi usaremos comunicación UART.

Para verificar el correcto despliegue de datos, se desarrollará un módulo de prueba que simule los mismos y se verificará visualmente en display.

Realizaremos pruebas de botones en pantalla, como el seteo de las alarmas y el tiempo.

Para cada módulo se tendrá un módulo de prueba que garantice su funcionamiento para verificar el correcto funcionamiento de cada uno y minimizar los errores al usarlos conjuntamente.

En cuanto al hardware se realizará prueba del buzzer.

Hito intermedio

Se estima hacer entrega en el hito intermedio de :

- Módulo SPI o UART
- Módulo de control de display, con funcionalidades de gráficas y botones.
- Módulo de procesamiento de datos.
- Manejo de botones de display.

Cronograma

Semana 1	-Estudio de Launchpad y Boosterpack. -Módulo SPI.
Semana 2	-Estudio de display (demo). -Pruebas Launchpad+WiFi+Display. -Módulo UART.
Semana 3	-Módulo de display.
Semana 4 (Hito)	-Módulo de procesamiento de datos. -Manejo de botones de display.
Semana 5	-Interfaz usuario-ordenador. -Módulo de ingreso de datos.
Semana 6	-Integración de módulos. -Pruebas Finales
Semana 7	-Documentación

Módulo	Archivo fuente	Archivo encabezado
Board	board.c	board.h
Cli_Uart	cli_uart.c	cli_uart.h
Display	display_interfaz.c	display_interfaz.h
	display_vitals.c	display_vitals.h
Driverlib	Archivos MSP432	
LcdDriver	Drivers del Display	
GrLib	Archivos del Display	
Touch	touchP401R.c	touchP401R.h
Pocesamiento	procesamiento.c	Procesamiento.h
Recepción	recepcion.c	recepcion.h
Simplelink	Archivos comunicación Wifi	
Spi_cc3100	Archivos CC3100	
Timer	timer.c	timer.h
	timer_hw.c	timer_hw.h
Utils	utils.c	utils.h
Wifi	wifi.c	wifi.h
Aplicación	main.c	

Figura 21: Archivos del proyecto.