



Limitador de consumo de corriente programable mediante Bluetooth

Fabián H. Alvarez ^{a*}, Juan P. Gross ^{b,c}, Guillermo A. Fernández ^{b,c}

^a *Universidad Nacional de Misiones, Facultad de Ingeniería, Oberá, Misiones, Argentina.*

^b *Universidad Nacional de Misiones, Facultad de Ingeniería, Instituto de Materiales de Misiones (IMAM), Misiones, Argentina.*

^c *ETCOLAB, GIDE, Juan Manuel de Rosas 325, Oberá, Misiones, Argentina.*

fabianhalvarez99@gmail.com, gross@fio.unam.edu.ar, fernandez@fio.unam.edu.ar

Resumen

Este trabajo presenta los resultados de la actividad integradora realizada en la asignatura Técnicas Digitales 2 de la carrera Ingeniería Electrónica, que tiene por objetivo de articular los saberes propios de la misma y de otras asignaturas previas. En esta actividad se desarrolla un limitador de corriente programable mediante Bluetooth, realizándose el diseño y construcción de un sistema embebido basado en un microcontrolador de la familia ESP-32 y desarrollándose una aplicación Android apropiada para la programación del limitador a través de Bluetooth. Seguidamente se detallan los cálculos, consideraciones de diseño y la explicación sobre el funcionamiento del sistema desarrollado. También se presentan los resultados alcanzados en los ensayos, observándose que el sistema implementado funciona correctamente. Cabe mencionar que, en los ensayos realizados, las señales de salida de los sensores de tensión y corriente que requiere el sistema propuesto, son simulados mediante tensiones de corriente continua variables obtenidas con potenciómetros. A través de la aplicación desarrollada y un celular con Bluetooth, puede establecerse el límite en el consumo de corriente de una carga, lográndose que el circuito interrumpa esta corriente cuando la misma excede el límite programado. Con esto se concluye el cumplimiento de los objetivos del trabajo integrador, cubriéndose áreas del saber propias de la asignatura indicada y además desarrollándose capacidades complementarias como la programación de aplicaciones para dispositivos móviles.

Palabras Clave – Consumo de corriente, limitador, Bluetooth, ESP32, Microcontroladores.

1. Introducción

Los circuitos limitadores de corriente son dispositivos diseñados para la desconexión/reconexión automática de la corriente consumida por cierta carga, la cual puede estar constituida por un circuito eléctrico o electrónico. La desconexión de la carga se da cuando el consumo de corriente excede un límite establecido. Este consumo excesivo puede darse debido una condición de cortocircuito o bien por una sobrecarga, actuando así el limitador como dispositivo de protección, tanto para la carga como para la fuente que le provee energía. Es por esto que los limitadores de corriente resultan de gran importancia para en diversas aplicaciones, tanto domésticas como industriales. Por otra parte, actualmente los limitadores de consumo de corriente están siendo usados para restringir el consumo energético en sistemas denominados “Microrredes Eléctricas”, los cuales constituyen pequeños sistemas de generación con fuentes de energías renovables, almacenamiento de energía y distintos tipos de cargas. Debido a la baja capacidad que poseen las fuentes de energía en las microrredes, es

* Autor en correspondencia.

necesario que exista un control adecuado en el consumo de energía para mantener en todo momento la igualdad entre generación y demanda [1]. Una forma de contribuir con el cumplimiento de esta igualdad, es evitando que las cargas superen un consumo límite preestablecido. Esto puede lograrse limitando la corriente que absorben los grupos de cargas de la microrred que son consideradas no críticas (no son esenciales que se mantengan energizadas). Estas cargas deben incorporar un limitador que las desconecte cuando exceden el límite de corriente establecido para las mismas [2].

En base a lo explicado, este trabajo expone el diseño e implementación del prototipo de un sistema embebido basado en un microcontrolador ESP-32-D0WDQ6 [3], [4], que tiene la capacidad de medir la corriente y tensión aplicada sobre una carga, abrir el circuito en caso de que la corriente exceda un límite preestablecido y transmitir los datos obtenidos a través de Bluetooth [5]. La información puede ser visualizada en un dispositivo que opere con el sistema operativo Android (como ser un celular o una tableta), utilizando una aplicación específica que también fue desarrollada en el presente proyecto. La aplicación mencionada, además de mostrar las mediciones de corriente y tensión, permite al usuario programar el límite de consumo de corriente deseado.

El trabajo se enmarca en la finalización del cursado de la asignatura Técnicas Digitales 2, correspondiente al cuarto año de la carrera de Ingeniería Electrónica. Tratándose de una actividad de carácter integrador, tiene por objetivo la articulación de saberes adquiridos tanto en la materia como en asignaturas previas, requiriendo para esto el diseño, implementación y puesta a punto de un circuito electrónico digital basado en un microcontrolador ampliamente disponible en el mercado nacional.

A continuación, se presenta el sistema desarrollado y la fundamentación de su funcionamiento tanto desde el punto de vista del hardware como del software. Posteriormente se exponen cálculos y consideraciones correspondientes a cada etapa del circuito, mostrándose finalmente los resultados obtenidos junto a las conclusiones pertinentes.

2. Diseño del sistema embebido

En base a las características del problema, el mismo se ha resuelto desarrollando un sistema embebido basado en el microcontrolador ESP-32-D0WDQ6, el cual fue seleccionado principalmente por contar con conectividad Bluetooth, necesaria para el proyecto. La Fig. 1 representa un esquema en bloques simplificado del limitador de corriente programable desarrollado. El circuito es gobernado por el microcontrolador mencionado, el cual recibe en sus entradas analógicas la señal de salida de los sensores de tensión y corriente. Con cada muestra obtenida a partir de la medición de corriente, el a través de su programa, el microcontrolador compara este valor obtenido con el límite de corriente programado. En caso de que la corriente medida supere el límite, a través del actuador se desconecta la carga. El sistema embebido construido, también tiene un pulsador cuyo accionamiento permite reconectar la carga, en caso de que el sistema se encuentre operando sin un dispositivo Android conectado.

Al iniciar su operación, el sistema comienza la comunicación Bluetooth como servidor. De esta forma, constantemente se encuentra emitiendo los valores medidos de corriente, tensión y el límite de corriente programado para la carga con la que interactúa el circuito. Al conectarse un teléfono inteligente utilizando la aplicación desarrollada en este trabajo, el mismo recibe la información para poder visualizar los datos y, en caso de que la carga se encuentre desconectada, poder reconectarla.

Además, mediante la interfaz gráfica diseñada en la aplicación, el usuario puede configurar el valor límite de corriente deseado.

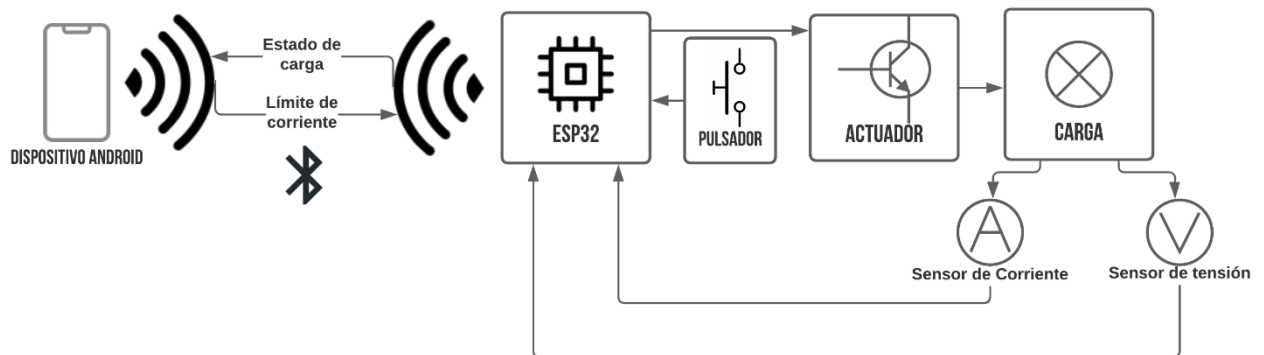


Fig. 1. Diagrama de bloques del limitador de consumo de corriente programable a través de Bluetooth.

A continuación, se describen las características del hardware (circuito) y el firmware (programa del ESP-32) del sistema implementado.

2.1. Hardware

En el diseño del circuito se utilizó la placa de desarrollo Hiletgo ESP-32S mostrada en la Fig. 2, la misma posee un microcontrolador ESP-32-D0WDQ6 de la empresa Espressif Systems [3], [4]. Este dispositivo se trata de un microcontrolador de 32 bits que integra WiFi y Bluetooth. Esto último es el motivo de la selección de dicho dispositivo para esta aplicación.



Fig. 2. Placa de desarrollo HiLetgo ESP-32S.

El sistema propuesto fue construido fuera de los laboratorios de la facultad, debido a las restricciones a la presencialidad de público conocimiento. Es por esto que no se contaron con algunos de los dispositivos requeridos en este trabajo. Tal es el caso de los sensores corriente y tensión, los cuales fueron simulados a través de dos potenciómetros lineales, como para poder generar señales analógicas variables a voluntad.

El actuador, encargado de efectuar la desconexión de la carga tras la orden del programa ejecutado en el microcontrolador, consta de un relé comandado por un par de transistores conectados en un arreglo Darlington. La base de este arreglo de transistores es excitada por la señal proveniente de un pin digital del microcontrolador. El relé se encuentra en paralelo con un led indicador de estado, como se observa en la Fig. 3. Para el dimensionamiento de los resistores de polarización del led indicador y el relé actuador, en primer lugar se calculó la resistencia limitadora R_2 para el led.

$$R_2 = \frac{V_{cc} - V_{LED} - V_{CE}}{I_{LED}} \quad (1)$$

donde V_{cc} es la tensión de alimentación, V_{LED} es la caída de tensión en directa del led, V_{CE} es la caída de tensión colector-emisor del transistor e I_{LED} la corriente a través del led. Considerando $V_{cc} = 5$ V, $V_{LED} = 2$ V, $V_{CE} = 0,7$ V e $I_{LED} = 10$ mA, resulta en $R_2 = 230 \Omega$, adoptando un valor comercial de $220 \Omega/0,25$ W.

Luego se procedió a calcular la resistencia de base R_1 del arreglo Darlington, de manera que los transistores se encuentren polarizados operando en corte-saturación. En este paso se tuvo en cuenta la corriente nominal del relé utilizado [6]. Los cálculos realizados son los que se muestran a continuación:

$$I_C = I_{LED} + I_{relay} \quad (2)$$

$$I_B \geq 5 \times \frac{I_C}{\beta} \quad (3)$$

$$R_1 \leq \frac{V_{OH(ESP32)Min} - 2 \times V_{BE}}{I_B} \quad (4)$$

Reemplazando los valores correspondientes resultó en un valor máximo para R_1 de 30266Ω . Con el fin de asegurar la polarización deseada y según la disponibilidad de componentes durante el desarrollo, se empleó un resistor de $22 \text{ k}\Omega/0,25$ W.

A continuación, se calculó el valor de la resistencia pull-up R_3 , asociada al pulsador BTN de reconexión de la carga.

$$R_3 < \frac{V_{cc(3,3)} - V_{IHMin}}{I_{IHMax}} \quad (5)$$

Dado que el valor límite superior calculado para R_3 es de $16,5 \text{ M}\Omega$, para la misma se empleó una resistencia de $10 \text{ k}\Omega/0,25$ W.

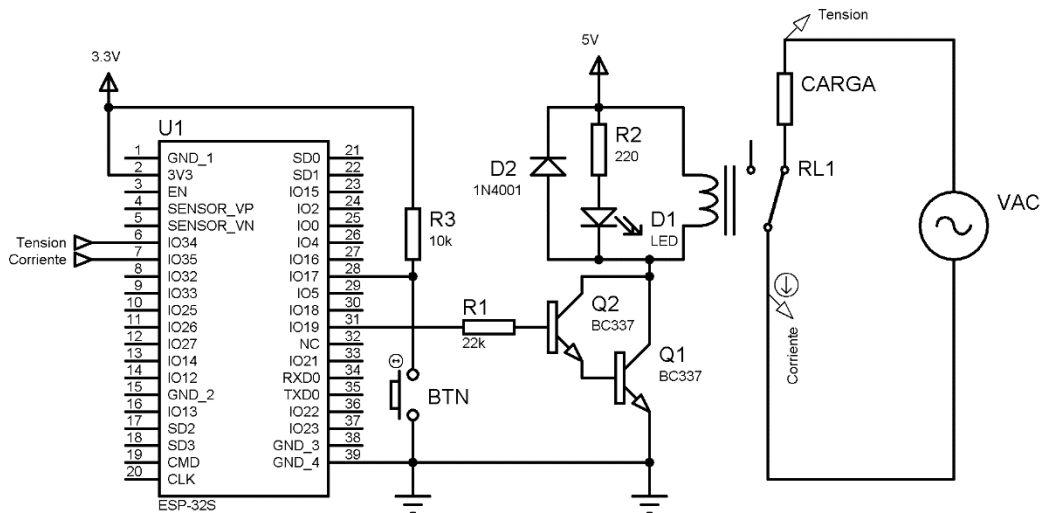


Fig. 3. Esquema del circuito correspondiente al limitador de corriente.

De esta manera, quedaron determinados todos los componentes del circuito indicado en la Fig. 3. La siguiente etapa de desarrollo se relaciona con el diseño e implementación del programa (firmware) que ejecuta el microcontrolador para comandar al sistema.

2.2. Firmware (programa)

El microcontrolador fue programado utilizando el Entorno de Desarrollo Integrado (IDE) de la plataforma Arduino [7], haciendo uso de la compatibilidad que posee el ESP-32S con este entorno de desarrollo. El motivo de emplear este IDE es la posibilidad de integrar las librerías disponibles y simplificar el desarrollo de esta etapa del trabajo.

A partir de lo mencionado, se utilizó la librería “*BluetoothSerial.h*” [8] para poder administrar de forma sencilla la comunicación Bluetooth a través de los métodos “*read()*” y “*write()*”. También se utilizó la librería “*EEPROM.h*” para la escritura y lectura de datos en la memoria no volátil, que en el caso del ESP-32 es una memoria flash y no EEPROM. Esto permitió el almacenamiento del límite de corriente configurado por el usuario.

La Fig. 4 presenta el diagrama de flujo que explica la secuencia del programa desarrollado para el microcontrolador. Al iniciar el programa, se comprueba la existencia de datos correspondientes a la configuración del límite de corriente en la memoria no volátil del microcontrolador. En caso de que no existiera esta información (primera vez que se usa el sistema), por defecto se configura en 5A el límite. En caso contrario, el programa recupera el límite guardado. Luego de configurar los pines de entrada y salida, inicializa el servidor Bluetooth e inicia el bucle de muestreo y transmisión de datos. Tras obtener las mediciones de corriente y tensión, compara el primer dato con el límite preestablecido. En caso de que la corriente sea excesiva, desconecta la carga poniendo en bajo la salida del pin digital correspondiente, y transmite por Bluetooth el mensaje “disc” haciendo referencia a “*disconnected*”. Luego genera un string para cada variable medida, con el valor de la muestra precedida por un caracter de identificación: ‘c’ para la corriente y ‘v’ para la tensión. Posteriormente se verifica si hay datos no leídos provenientes de la comunicación Bluetooth; en caso afirmativo, el

programa identifica si el mensaje recibido es el número “8017”, siendo este el código utilizado para la orden de reconexión de la carga (si la misma se encuentra desconectada). De tratarse de otro caso, se guarda el valor numérico del mensaje como límite de corriente. Finalmente se transmiten los datos de la medición de corriente y tensión, y del límite configurado. Esta transmisión es efectuada cada 150 ms para evitar superposición de datos en esta operación.

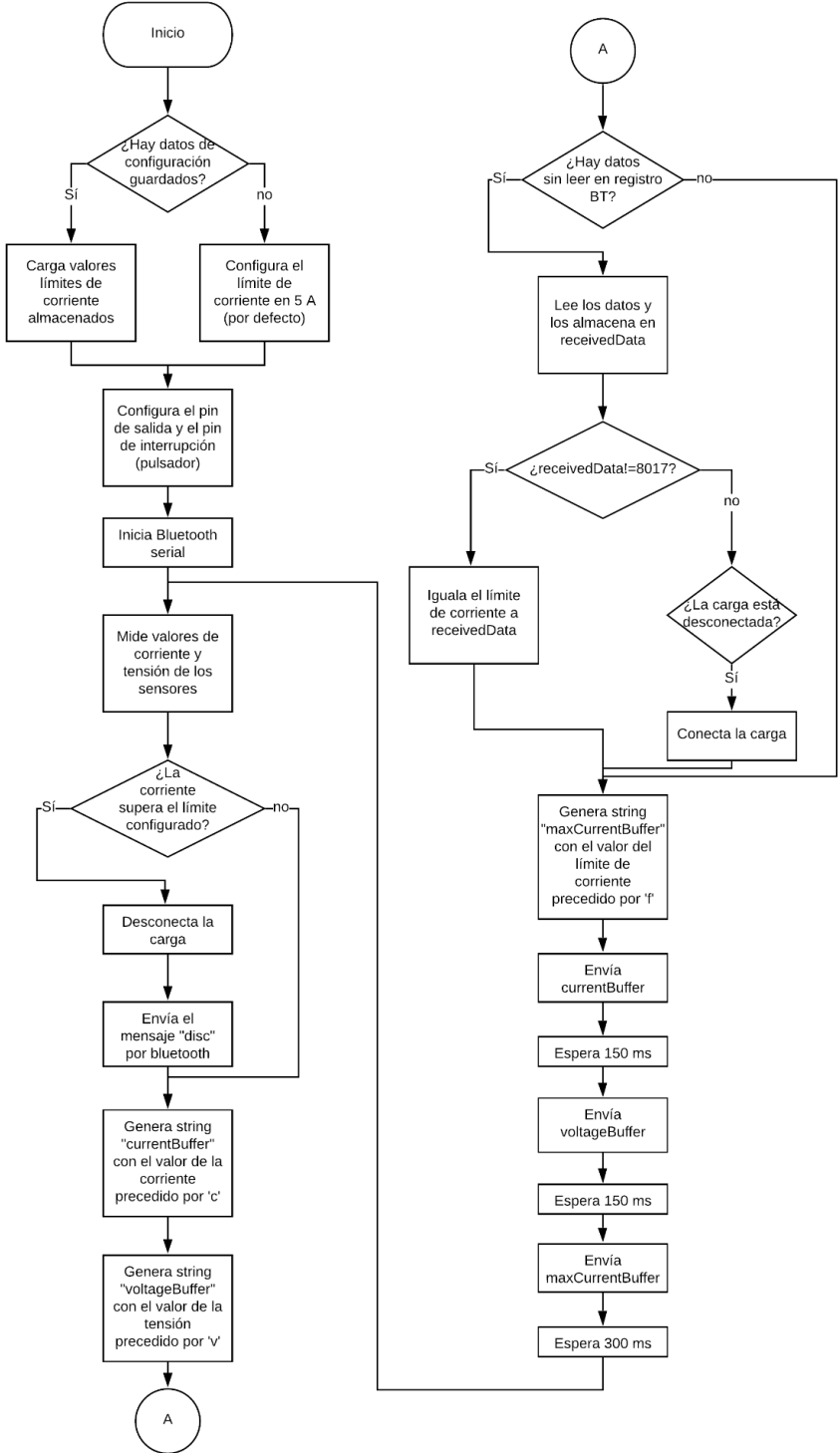


Fig. 4. Diagrama de flujo del programa ejecutado en el ESP-32-D0WDQ6.

Cabe destacar que el microcontrolador del sistema constantemente realiza mediciones y envía información a través de Bluetooth, de manera indistinta de si un dispositivo Android se encuentra conectado o no al mismo. Este comportamiento simplifica la comunicación entre ambos dispositivos y permite que el limitador trabaje en un único modo de funcionamiento continuo.

3. Desarrollo de la Aplicación Android

La aplicación Android realizada gestiona la selección del dispositivo de medición en la conexión Bluetooth, visualizar las mediciones de tensión y corriente realizadas por el microcontrolador, permite configurar el límite de corriente deseado para el control de la carga y también comandar la reconexión en caso de que esta última se halle desconectada. La aplicación obtenida es compatible con cualquier dispositivo que funcione con sistema operativo Android 6.0 [9] o superior y fue desarrollada en lenguaje Java en la plataforma Android Studio 7.0 [10]. Esta última permitió diseñar la interfaz gráfica de usuario en forma completa y también escribir el programa correspondiente a la aplicación para cumplir con las funcionalidades solicitadas para el sistema.

3.1. Interfaz de usuario

Como puede observarse en la Fig. 5, la interfaz de usuario (IU) consta en la parte superior de una burbuja donde se indica el nombre del dispositivo conectado, su dirección MAC y un botón para realizar la desconexión de la comunicación. Más abajo pueden visualizarse las mediciones de tensión y corriente realizadas por el microcontrolador. En su parte inferior se encuentra un deslizador con el cual el usuario puede establecer la configuración del límite de corriente al cual se desconectará la carga.



Fig. 5. Aplicación Android: Interfaz de usuario de la pantalla principal.

En el menú superior, el usuario puede escoger entre obtener la lista de dispositivos previamente vinculados, o bien realizar una búsqueda de dispositivos cercanos para así realizar una nueva conexión

3.2. *Conexión y administración de la comunicación Bluetooth*

Para efectuar las tareas relacionadas a la comunicación Bluetooth, se siguieron las indicaciones de la guía para desarrolladores de Android [11]. Cabe destacar que el bloque que corresponde a la etapa de comunicación mencionada se ejecuta en un hilo de programa diferente al de la interfaz de usuario, esto es para evitar “congelamientos” de la misma durante tiempos muertos o de espera. Como puede observarse en la Fig. 6, la etapa de comunicación comienza por la obtención de un objeto de clase *BluetoothAdapter* (adaptador Bluetooth), el cual constituye la interfaz necesaria para utilizar el protocolo Bluetooth en el dispositivo móvil. A continuación, se comprueba si el módulo Bluetooth se encuentra habilitado en el dispositivo; en caso de que no, se solicita al usuario habilitarlo en la ejecución de la aplicación. Si el usuario decide no habilitarlo, la aplicación se cierra debido a que no puede operar de esa manera.

Una vez habilitado el módulo Bluetooth del dispositivo móvil, el usuario puede escoger entre dos opciones para seleccionar el dispositivo al cual conectarse: seleccionar entre los dispositivos previamente vinculados, o bien realizar la búsqueda de dispositivos cercanos. En el segundo caso, tras la selección se solicita al usuario su aprobación para realizar la vinculación.

Conocida la dirección MAC del dispositivo seleccionado, el teléfono intenta conectarse como cliente. Si la conexión se efectúa con éxito, el programa comienza con la espera constante de nuevos datos enviados por el microcontrolador, los cuales son visualizados si se reciben correctamente. Si no se logra establecer la conexión a través de Bluetooth, eventualmente se pierde la misma, el usuario debe volver a conectarse con el dispositivo de forma manual.

3.3. *Obtención de datos*

Debido a que el dispositivo ESP-32 está transmitiendo en forma constante los datos de corriente, tensión y límite de corriente configurado, la aplicación necesita un método para reconocer a qué variable corresponde cada dato recibido. Para ello, interpreta el primer carácter de cada mensaje, el cual actúa como identificador. El carácter inicial ‘c’ indica que el dato corresponde a la medición de corriente, el carácter ‘v’ corresponde a la medición de tensión, y el carácter ‘f’ representa a la configuración del límite de corriente. Posteriormente se obtienen los valores numéricos de los datos recibidos y se actualizan las respectivas variables medidas. La lectura del valor límite de corriente es utilizada una sola vez para conocer el estado de configuración previo a la conexión.

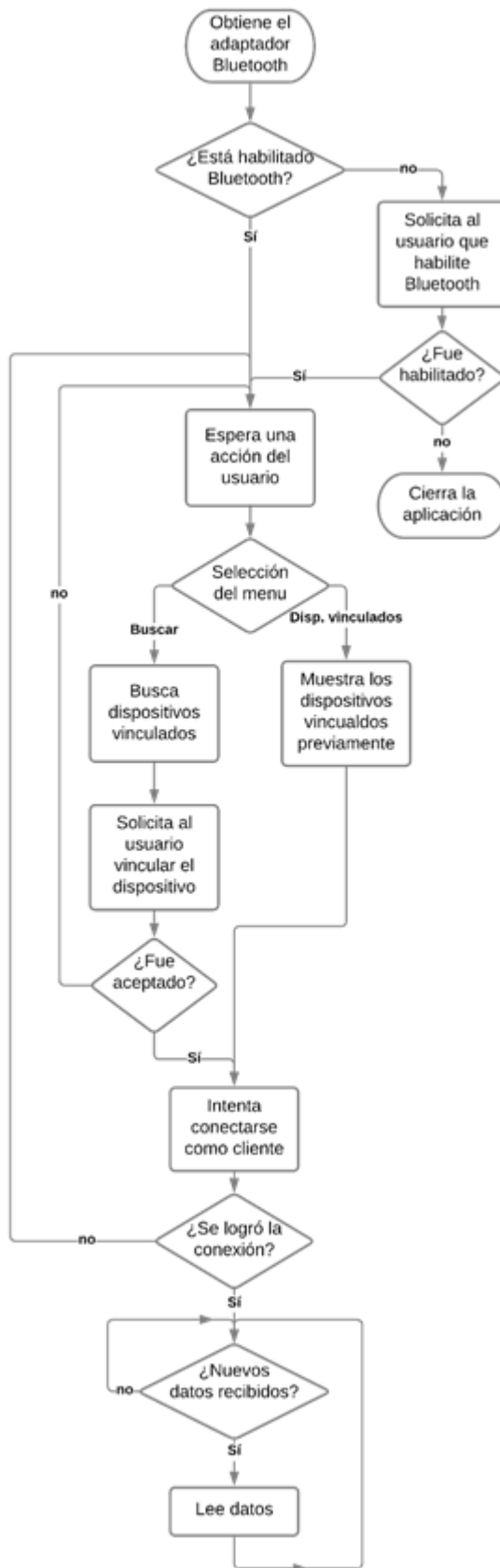


Fig. 6. Diagrama de flujo simplificado del hilo de programa de la etapa de comunicación Bluetooth en la aplicación.

4. Resultados

Dado que todo el desarrollo de este trabajo se ha realizado fuera de los laboratorios de la facultad, debido a las restricciones a la presencialidad que hubo durante el cursado del año 2020, el circuito del sistema fue desarrollado a modo de prototipo experimental, montándolo en un protoboard como se aprecia en la Fig. 7 y la aplicación fue instalada en un teléfono celular con Android. Para comprobar el funcionamiento del prototipo se realizaron diversas pruebas considerando varios escenarios posibles de operación, tales como: variaciones del sensor de tensión, del sensor de corriente (ambos casos simulados mediante los potenciómetros), conexión y desconexión del teléfono móvil, conexión de la carga mediante la aplicación o con el pulsador y la desconexión de la carga ante una corriente excesiva. En la Fig. 8 se observan las distintas pantallas de la aplicación durante las pruebas.

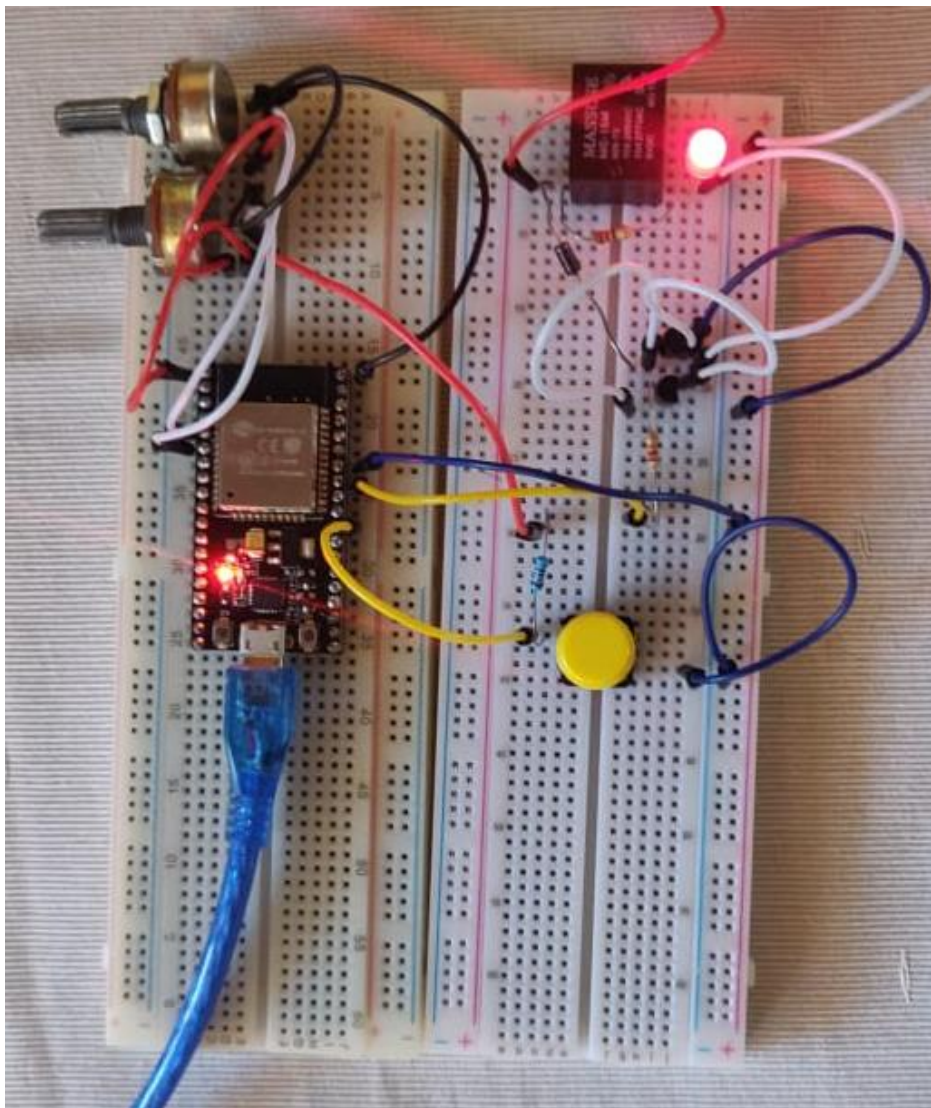


Fig. 7. Prototipo experimental del limitador de corriente.

Los resultados fueron favorables, mostrando tanto el software como el hardware un funcionamiento correcto. La aplicación posee una interfaz sencilla y no presentó fallas, ni “congelamientos” de pantalla durante los procesos de conexión entre los dispositivos. El sistema mostró una rápida respuesta para la desconexión de la carga cuando se excedió el límite de corriente programado.

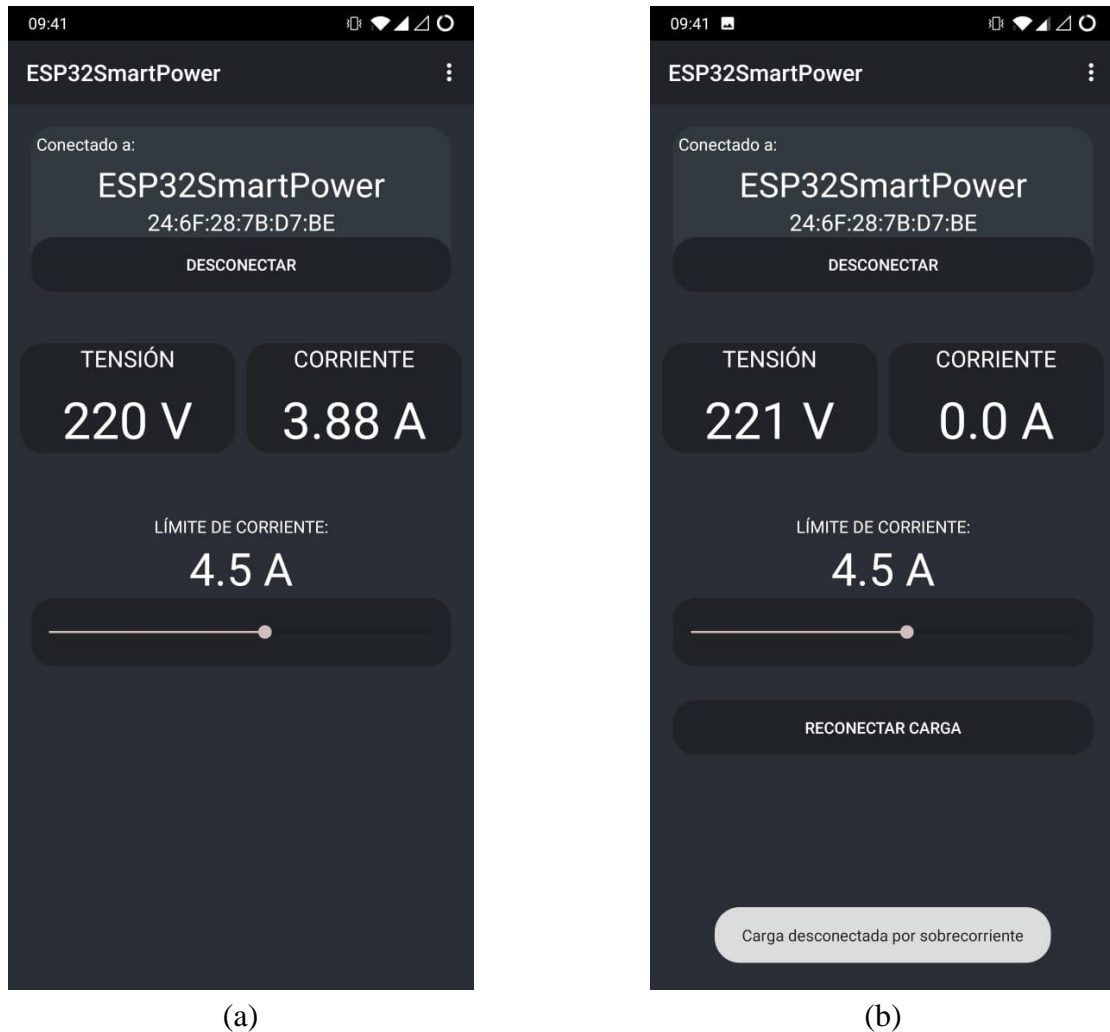


Fig. 8. (a) Pantalla principal mostrando mediciones realizadas. (b) Notificación de carga desconectada por sobre corriente y habilitación del botón para reconexión.

La comunicación entre ambos dispositivos demostró ser dinámica y sin tiempos de espera excesivos. La frecuencia de muestreo empleada en la medición de la tensión y corriente fue apropiada para la aplicación, permitiendo observar correctamente los valores de dichas variables a través de la misma.

5. Conclusiones

Se diseñó e implementó un circuito digital basado en un microcontrolador ESP-32-D0WDQ6 como solución para un limitador de corriente programable a través de Bluetooth. Para este sistema

embebido se ha desarrollado tanto el software como el hardware, aplicando conocimientos y capacidades adquiridas en la asignatura Técnicas Digitales 2.

Los resultados obtenidos en la experimentación sobre el prototipo construido confirman la satisfacción de la totalidad de los objetivos del trabajo integrador propuesto en la mencionada asignatura.

Durante el desarrollo del trabajo se articularon distintos saberes adquiridos, no sólo en Técnicas Digitales 2, sino también en otras asignaturas tales como Computación, Dispositivos Electrónicos y Electrónica Analógica, donde se estudiaron los principios de programación orientada a objetos, componentes y de circuitos electrónicos en general.

Si bien la programación de aplicaciones Android escapa de cierta manera a los objetivos de la asignatura Técnicas Digitales 2, los conocimientos de programación adquiridos en la misma facilitaron el entendimiento general necesario para la implementación de la aplicación requerida en el limitador de corriente, implicando esto el desarrollo de una capacidad complementaria de gran utilidad e importancia dentro del campo de diseño de los sistemas embebidos.

Adicionalmente, se considera que el sistema obtenido supone una plataforma de desarrollo escalable, con distintas posibilidades como ser: ampliar la cantidad de cargas a controlar o incorporar un registro histórico de los estados de carga. De esta manera es posible plantear potenciales aplicaciones tanto domésticas como industriales.

Referencias

- [1] García Vera, Yimy E.; Dufo-López, Rodolfo; Bernal-Agustín, José L. "Energy Management in Microgrids with Renewable Energy Sources: A Literature Review". *Applied Sciences*, vol. 9, n°18, year 2019.
- [2] «NT752 – Limitador De Consumo Eléctrico». [En línea]. Disponible: <http://www.rasesa.com/nt752-limitador-de-consumo-electrico/> [Último acceso: diciembre 2021].
- [3] Espressif Systems, “ESP32 Series Datasheet”, version 3.4, 2020.
- [4] Espressif Systems, “ESP32 Technical Reference Manual”, version 4.3, 2020.
- [5] «Bluetooth Technology Overview | Bluetooth® Technology Website», Bluetooth® Technology Website, 2021. [En línea]. Disponible: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/> [Último acceso: diciembre 2020].
- [6] Massuse Electric Ltd, “ME-15M relay”, 5V VDC Power relay datasheet.
- [7] «What is Arduino?» (2018). [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: diciembre 2020].
- [8] «Arduino-esp32/BluetoothSerial.h». Repositorio en GitHub. [En línea]. Disponible: <https://github.com/espressif/arduino-esp32/blob/master/libraries/BluetoothSerial/src/BluetoothSerial.h> [Último acceso: diciembre 2020].
- [9] «Android 6.0 API | Desarrolladores de Android». [En línea]. <https://developer.android.com/about/versions/marshmallow/android-6.0?hl=es-419> [Último acceso: diciembre 2020].
- [10] «Introducción a Android Studio | Desarrolladores de Android». [En línea]. Disponible: <https://developer.android.com/studio/intro> [Último acceso: diciembre 2020].
- [11] «Introducción general a Bluetooth | Desarrolladores de Android». [En línea]. Disponible: <https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es> [Último acceso: diciembre 2020].