






Design and Development of a Wireless Programmer for the Arduino Ecosystem and its Implementation in LAB-VEE Educational TechMakers Training Boards

Reymi Then, MS¹ , Laura Espinal, MS² , Edwin Marte, PhD³ , Starlyn Suárez⁴ , Danishell Tiburcio⁵ 
^{1,4,5}*Universidad Tecnológica de Santiago (UTESA), República Dominicana, reymithen@docente.utesa.edu, suarezreyesstarlyn@gmail.com, danishelltiburcio@gmail.com*

²*Universidad Internacional Iberoamericana (UNiB), Puerto Rico, espinalyunen@gmail.com*

³*University of Florida (UF), United States, emartezorrilla@ufl.edu*






Abstract– This article presents the design and development of a wireless programmer for Arduino-compatible devices, aimed at eliminating the need for physical connections for programming and communication, allowing users to upload code remotely from computers, providing new connection possibilities for interaction with embedded electronic devices. The motivation arose from the interest in develop a card for training students in programming languages and programmable devices, where it was desired that it could be programmed wirelessly, in solution to certain limitations of physical connections such as mobility restriction, inability to connect to encapsulated devices without opening them and the risks associated with damage due to electrical coupling between the device and the computer.

Therefore, in response to the limited existence of a programmer that addresses these needs, proceeded to develop a programmer that not only facilitates wireless programming, but also integrates seamlessly into the Arduino ecosystem safely, with high performance and also allowed mass production while remaining a low-cost solution. The article also shows its implementation in the LAB-VEE ecosystem training card, which was conceived for teaching technology and programming to K-12 students. This card, thanks to its programming and wireless communication capacity, expands educational possibilities in STEM areas, offering a practical and advanced resource for learning technological skills.

The success of the project was proven through exhaustive tests, which validated the reliability and effectiveness of both the wireless programmer and its integration with the card for the use of its peripherals. So together these contributions open new possibilities for technological education and the development of embedded electronic projects.

Keywords— *Wireless programmer, Arduino, AVR, LAB-VEE, Microcontrollers.*

Diseño y Desarrollo de un Programador Inalámbrico para el Ecosistema Arduino y su Implementación en Tarjetas de Entrenamiento de LAB-VEE Educational TechMakers

Reymi Then, MS¹ , Laura Espinal, MS² , Edwin Marte, PhD³ , Starlyn Suárez⁴ , Danishell Tiburcio⁵ 
^{1,4,5}Universidad Tecnológica de Santiago (UTESA), República Dominicana, reymithen@docente.utesa.edu, suarezreyesstarlyn@gmail.com, danishelltiburcio@gmail.com
²Universidad Internacional Iberoamericana (UNiB), Puerto Rico, espinalyunen@gmail.com
³University of Florida (UF), United States, emartezorrilla@ufl.edu

Resumen– Este artículo presenta el diseño y desarrollo de un programador inalámbrico para dispositivos compatibles con Arduino, orientado a eliminar la necesidad de conexiones físicas para programación y comunicación, permitiendo a los usuarios cargar código de forma remota desde computadoras, brindando nuevas posibilidades de conexión para la interacción con los dispositivos electrónicos embebidos. La motivación surgió del interés en desarrollar una tarjeta para la formación de estudiantes en lenguajes de programación y dispositivos programables, donde se deseaba que pudiera programarse de forma inalámbrica, como solución a ciertas limitaciones de las conexiones físicas como restricción de movilidad, imposibilidad de conectarse a dispositivos encapsulados, sin abrirlos y los riesgos asociados a daños por acoplamiento eléctrico entre el dispositivo y el ordenador.

Por lo tanto, ante la limitada existencia de un programador que atienda estas necesidades, se procedió a desarrollar un programador que no solo facilite la programación y comunicación inalámbrica, sino que también se integre perfectamente al ecosistema Arduino de forma segura, con alto rendimiento y que además permita la producción en masa sin dejar de ser una solución de bajo costo. El artículo también muestra su implementación en la tarjeta de formación del ecosistema LAB-VEE, que fue concebida para la enseñanza de tecnología y programación a estudiantes de K-12. Esta tarjeta, gracias a su capacidad de programación y comunicación inalámbrica, amplía las posibilidades educativas en áreas STEM, ofreciendo un recurso práctico y avanzado para el aprendizaje de habilidades tecnológicas.

El éxito del proyecto quedó comprobado mediante pruebas exhaustivas, que validaron la fiabilidad y eficacia tanto del programador inalámbrico como de su integración con la tarjeta para el uso de sus periféricos. De manera que, en conjunto, estas contribuciones abren nuevas posibilidades para la educación tecnológica y el desarrollo de proyectos electrónicos embebidos.

Palabras clave— Programador inalámbrico, Arduino, AVR, LAB-VEE, Microcontroladores.

I. INTRODUCTION

En las últimas décadas, la evolución de los sistemas embebidos ha jugado un papel de gran importancia en la transformación digital que permea todos los aspectos de la vida moderna [1]. En este escenario, las tarjetas Arduino han emergido como herramientas esenciales, facilitando el desarrollo de una amplia variedad de aplicaciones en industrias tan diversas como la automotriz, biomédica, domótica, robótica, Internet de las Cosas (IoT) [2], [3], entre otras. Las tarjetas Arduino han ganado mucha popularidad entre profesionales, estudiantes y aficionados, esto se ha debido en parte a su uso de microcontroladores Advanced Virtual RISC (AVR) de Atmel, ahora parte de Microchip Technology, conocidos por su arquitectura de Conjunto de Instrucciones Reducido (RISC), amplias capacidades de procesamiento y facilidad de programación [4]. No obstante, el espectro de opciones disponibles no se limita a estos; tarjetas basadas en microcontroladores PIC de Microchip, como los STM32 de STMicroelectronics [5], los MSP430 de Texas Instruments [6], así como los ESP8266 y ESP32 de Espressif Systems, entre otros, también ofrecen características únicas y ecosistemas robustos para satisfacer diversas necesidades del mercado.

Las tarjetas Arduino, introducidas en 2005 por los investigadores Massimo Banzi, David Cuartielles, Gianluca Martino, David Mellis y Tom Igoe [7], [8], ofrecen ahora una gama extensa de versiones, permitiendo a los diseñadores elegir entre diferentes capacidades de memoria, velocidad de procesamiento, posibilidad de uso de periféricos incorporados como Wi-Fi, Bluetooth, sensores, actuadores, pantallas y GPS [9], además de ofrecer la posibilidad de conectar una amplia variedad de periféricos externos, lo cual permite realizar una gran cantidad de experimentos y proyectos, tanto a nivel de estudiantes, aficionados como profesional [10].

Sin embargo, dependiendo del uso que se les quiera dar, el desarrollo de aplicaciones basadas en tarjetas Arduino

presenta desafíos, especialmente en términos de comunicación, programación y depuración [11], [12], cuando se requiere ponerlas a trabajar en lugares poco accesibles o sencillamente cuando no se desea, o posibilita, una conexión de manera física entre la computadora y la placa. Tradicionalmente, estos procesos se han realizado mediante conexiones físicas entre la tarjeta y una computadora [13], limitando la movilidad y flexibilidad al utilizarla. Además, las conexiones físicas frecuentes exponen tanto la tarjeta como al equipo de programación a riesgos de daño [14]. Estas limitantes se convirtieron en desafíos para el equipo investigador al verse en la necesidad de desarrollar una tarjeta programable basada en microcontroladores AVR que fuera compatible con el Entorno de Desarrollo Integrado (IDE) de Arduino [15] y su protocolo de programación. Estos requerimientos debían ser superados para lograr la creación de recursos, tanto físicos (hardware) como digitales (software) y contenidos educativos para la enseñanza de competencias tecnológicas, los cuales serían utilizados en la continuación de un proyecto de investigación. Dicho proyecto se desarrolló por parte de la empresa LAB-VEE Educational TechMakers [39], [16], la cual se enfoca en propiciar y fomentar la formación de los estudiantes desde el Nivel Primario hasta el Nivel Secundario (K-12) en las áreas de Ciencias, Tecnologías, Ingenierías y Matemáticas (STEM) [13], [17], [18]. De manera que el propósito de LAB-VEE con el desarrollo de esta tarjeta, consistió en permitir que los estudiantes pudieran realizar prácticas para adentrarse al mundo de las tecnologías como la electrónica, la programación y la creación de dispositivos embebidos, similar a como se utiliza una placa de Arduino con diversos periféricos, pero eliminando la limitante de la conexión física entre la tarjeta desarrollada y la computadora [19]. Así que el objetivo principal estuvo centrado en que esta tarjeta permitiera comunicación y programación de manera inalámbrica desde la computadora. [20].

Dado este interés, surgió la necesidad de que el equipo investigador explorara alternativas existentes para dar solución al requerimiento planteado; sin embargo, dado el alto costo de equipos similares y la poca diversidad ofertada de dispositivos que permitan tal funcionalidad [21], se vio en la necesidad de incurrir en el diseño y desarrollo de un programador inalámbrico para implementarlo en la tarjeta de LAB-VEE orientada al entrenamiento de estudiantes de K-12 en lenguaje de programación y dispositivos programables, que fuera compatible con Arduino. Este desarrollo requirió superar varios desafíos técnicos, tanto para la transmisión como para la recepción de datos. Dentro de estos desafíos se encuentra la identificación de los dispositivos electrónicos a utilizar en el transmisor para posibilitar la comunicación USB con la computadora [22], [23], con el fin de decodificar el protocolo de comunicación del programador del IDE de Arduino. De igual manera, se requirió el desarrollo de un firmware (ver figura 1) que pudiera gestionar dicho protocolo, así como la

identificación del microcontrolador a utilizar con el firmware desarrollado para convertir el protocolo de programación del IDE de Arduino a un protocolo propio que posibilite ser gestionado. Por último, se procedió a realizar la selección de un módulo de comunicación inalámbrica que permitiera transmitir los datos recibidos del programador del IDE de Arduino haciendo uso de su conversión al nuevo protocolo establecido. Sin embargo, para lograr la programación del microcontrolador AVR de la tarjeta LAB-VEE, no basta con lo realizado hasta este punto, ya que, para realizar la programación, además de la transmisión de los datos, también se debió reinvertir el proceso en el módulo hardware receptor. A todo este proceso se le sumó el requerimiento de que la conexión entre el transmisor y el receptor debía ser estable y fiable, para que la programación se realizara de manera segura y sin errores.

```

ESP32_Programador_Masterino
62 //
63 pinMode(InterruptPin, INPUT_PULLUP); // Configura el pin como entrada con resistencia pull-up
64 attachInterrupt(digitalPinToInterrupt(InterruptPin), InterruptHandler, FALLING); // Configura la interrupción en Flanco de bajada
65 }
66
67 void loop(){
68 // Serial.println(SerialBT.connected());
69
70
71
72 while(!SerialBT.connected()) SerialBT.connect("BT-PROUEBAS", delay(100));
73
74 if (Serial2.available()) {
75   char ch = Serial2.read();
76   Serial.write(ch);
77   SerialBT.write(Serial2.read());
78 }
79
80 if (SerialBT.available()) {
81   Serial2.write(SerialBT.read());
82 }
83
84 }

```

Figura 1. Fragmento de firmware para la gestión de la programación y la comunicación.

Fuente: Elaboración propia.

De manera que este artículo se centra en presentar el proceso llevado a cabo para el desarrollo de dicho programador inalámbrico en las tarjetas de LAB-VEE compatibles con Arduino, pasando por la concepción de la idea, el proceso de desarrollo, su implementación y los desafíos superados.

II. CONCEPCIÓN DE LA IDEA

La iniciativa de desarrollar este dispositivo surgió del interés de facilitar la programación de manera inalámbrica para una de las tarjetas de entrenamiento de LAB-VEE, la cual se estaba diseñando para el entrenamiento de estudiantes en la creación y uso de dispositivos embebidos. Como microcontrolador de esta tarjeta se tenía establecido el uso de dispositivos Atmel, de la compañía Microchip Technology, de manera específica aquellos compatibles con el IDE de Arduino. Con el interés de que esta pueda pasar a ser parte de la comunidad de Arduino, convirtiéndose en un recurso más del ecosistema que pueda interactuar con software, librerías, hardware, periféricos, materiales educativos y ejemplos de este, enriqueciendo así las posibilidades de su uso. Además de que este programador fuera compatible con el IDE de Arduino, se debía cumplir con otras especificaciones mecánicas y técnicas para que pudiera ser utilizado como parte de la tarjeta de entrenamiento en el diseño y uso de

dispositivos embebidos. Dentro de estas especificaciones se encontraba en primer lugar el requerimiento de sustituir el cable USB de programación que se utiliza entre la computadora y las placas de desarrollo de Arduino por una alternativa que permitiera reemplazar esta vía por otra que fuera inalámbrica y que posibilitara la misma funcionalidad. En segundo lugar, el tamaño del hardware debía cumplir con dimensiones específicas, ya que el espacio designado en la tarjeta de entrenamiento para su implementación era de 30 mm de ancho, 40 mm de largo y 4 mm de profundidad. En tercer lugar, se requería que el dispositivo inalámbrico permitiera la comunicación entre la computadora y la tarjeta a desarrollar para uso de la comunicación serial con diversos recursos tanto con el IDE de Arduino como con otros programas. En cuarto lugar, se perseguía minimizar el riesgo de errores de conexión y la estabilidad de la misma para que quien lo use no experimente estos inconvenientes, para evitar que se vea en la necesidad de retornar al uso de la conexión USB. En quinto lugar, se perseguía que el dispositivo desarrollado fuera de bajo costo y que permitiera su producción en masa. En sexto lugar, era de interés que este dispositivo no pudiera conectarse con cualquier otro dispositivo cercano no deseado, sino que tuviera una conexión de punto a punto segura.

III. DESARROLLO DEL PROGRAMADOR INALÁMBRICO

Dado que el desarrollo de este dispositivo estaba enfocado a ser compatible con todo el ecosistema de Arduino, fue necesario que el equipo investigador realizara una adecuada revisión del estado del arte de todo lo concierne a estas placas, su uso y programación. Por lo cual, se inició por documentar y comprender los microcontroladores de Atmel, quienes fueron los desarrolladores de la arquitectura AVR en 1996 [24].

Estos microcontroladores se distinguieron por su arquitectura RISC, su eficiencia en términos de consumo de energía y su potencia de procesamiento. Esta familia de microcontroladores ha experimentado un crecimiento y una evolución significativa desde su introducción, mejorando cada vez más la línea de los AVR, lanzando nuevos modelos con mayores capacidades de procesamiento, integración de periféricos y mayores velocidades de reloj. Uno de los microcontroladores AVR más populares es el ATmega328 [25], lanzado en 2008, el cual se ha convertido en un estándar de referencia en el mundo de la electrónica tanto para aficionados como para la educación por su facilidad de uso, amplia disponibilidad y versatilidad [26]. La popularidad de este microcontrolador en gran medida ha sido debido al desarrollo de placas de Arduino, como la Arduino Uno y la Arduino Nano. Arduino nació con el objetivo de proporcionar una plataforma económica y de fácil uso para enseñar electrónica a los alumnos del Instituto de Diseño Interactivo de Ivrea (Italia) [7]. Actualmente, es un ecosistema de hardware y software libre que ha venido creciendo cada vez más hasta hoy día, además de estar siendo utilizadas a nivel de aficionados, también están siendo empleadas por muchas universidades y profesionales del área, sin mencionar la gran comunidad de desarrolladores que se

encuentran creando librerías y mejorando continuamente su IDE. Además, numerosas empresas están adaptando sus periféricos y microcontroladores para que sean compatibles con el ecosistema de Arduino, lo cual permite que, cada vez más, Arduino se convierta en un entorno de mayor versatilidad y de amplio uso.

Parte del éxito de las placas de Arduino también se debe a que estas pueden realizar la función de programar el microcontrolador sin la necesidad de dispositivos externos. Este es el método de programación simple y fácil de usar que solo requiere que se conecte la placa Arduino a la computadora mediante un cable USB y utilizando el IDE de Arduino es posible compilar y cargar el código en la placa, evitando tener que desmontar el microcontrolador o conectar dispositivos adicionales, siendo este método compatible con la mayoría de las placas Arduino modernas, como Arduino Uno, Nano, Mega, etc.[27].

Dicho método de programación corresponde al Bootloader vía USB, y consiste en la programación de una placa Arduino mediante la Clase de Dispositivo de Comunicación (CDC) haciendo uso del protocolo Modelo de Control Abstracto (ACM) que por medio de una conexión USB permite que la placa de Arduino sea reconocida como un dispositivo de comunicación serial por el sistema operativo de la computadora [28]. En el contexto de Arduino y otros dispositivos similares, el protocolo USB CDC ACM se utiliza para emular un puerto serie virtual tipo Transmisión y Recepción Asíncrona Universal (UART) cuando el dispositivo está conectado a una computadora a través de USB [29].

Para que la programación de una placa Arduino se pueda producir bajo este protocolo, se requiere que previamente se le haya cargado al microcontrolador el firmware de bootloader. Este firmware es el que realiza la función de un programador, recibiendo el nuevo programa y guardándolo en la memoria flash. Para cargar el firmware del bootloader a una placa Arduino se debe utilizar un programador como el USBasp, AVR ISP mkII o la opción de utilizar otro Arduino como programador mediante un sketch llamado ArduinoISP el cual convierte un Arduino en un programador de Programación Serial en el Circuito (ICSP) [30].

Los pines utilizados para la programación ICSP son:

- MISO (Master In Slave Out): El cual se utiliza para enviar datos desde el microcontrolador al programador.
- MOSI (Master Out Slave In): Este pin se utiliza para enviar datos desde el programador al microcontrolador.
- SCK (Serial Clock): Este proporciona el reloj serial generado por el programador para sincronizar la transferencia de datos.

- RESET: Se utiliza para reiniciar el microcontrolador y ponerlo en modo de programación.
- VCC: Con este se suministra la alimentación eléctrica al microcontrolador durante la programación.
- GND: Este pin es la conexión a tierra común entre el programador y el microcontrolador.

De manera que la programación a través de ICSP es fundamental cuando se desea cargar el bootloader a un microcontrolador por primera vez o cuando se necesita restaurarlo o recargarlo en caso de que haya sido borrado o corrompido [31]. De no estar este bootloader previamente cargado en el microcontrolador, no será posible su programación mediante el protocolo USB CDC ACM utilizado por el IDE de Arduino.

En síntesis, el bootloader es un código que está almacenado en una sección reservada de la memoria flash y es el primero en ejecutarse cuando el microcontrolador es energizado o reiniciado. Después de la inicialización del microcontrolador, el Controlador de Programa (PC) se establece para apuntar a la dirección de inicio del bootloader, ejecutándolo para que este verifique si se deben tomar acciones especiales, como escuchar comandos a través del puerto UART que indiquen si se desea cargar un programa mediante el protocolo de programación del IDE de Arduino. Esto usualmente involucra un breve período de tiempo, donde si se detecta una señal específica, el bootloader entra en un modo de carga de programa. Del bootloader no se detecta ninguna acción de carga dentro de un tiempo específico, finaliza su ejecución y pasa el control al programa de usuario cambiando el PC a la dirección de inicio del programa de usuario almacenado en la memoria flash. [32].

Para lograr la comunicación y programación entre la tarjeta de LAB-VEE, compatible con Arduino, y la computadora fue necesario realizar múltiples pruebas para la validación y categorización de distintos dispositivos electrónicos, protocolos y tecnologías que permitieran alcanzar el propósito principal del desarrollo, el cual consistía en la programación inalámbrica de la tarjeta en cuestión. Las validaciones realizadas fueron necesarias debido a que el método de programación utilizado por Arduino está basado en el protocolo de comunicación USB CMC ACM, utilizando el Atmega16u2 [33] para emular un puerto UART y que la computadora le reconozca como un dispositivo de comunicación serial que pueda ser seleccionado por programas informáticos como el IDE de Arduino, por lo cual los dispositivos electrónicos a utilizar para el desarrollo debían permitir una operación que reemplazara o complementara la funcionalidad del emulador de comunicación serial de la placa de Arduino para lograr la programación inalámbrica de la tarjeta a desarrollar. Dentro de las validaciones realizadas para la comunicación inalámbrica, se exploraron varios dispositivos en busca de identificar aquel que tuviera la mayor

compatibilidad con el método de programación de Arduino, realizándose pruebas con: 433mhz RF Transmitter and Receiver Module [34], NRF24L01 Wireless Transceiver Module [35], CC1101 Wireless Transceiver Module [36], HC-05 Wireless Bluetooth RD Transceiver [37] y ESP32-WROOM-32D [38].

A partir de las validaciones realizadas fue posible identificar el dispositivo que dispone de las mayores prestaciones para el cumplimiento de los requerimientos de comunicación establecidos para lograr disponer del dispositivo de programación inalámbrica a ser desarrollado. Siendo seleccionado el ESP32-WROOM-32D, dadas sus prestaciones tanto para el procesamiento de datos como para la comunicación inalámbrica. Lográndose, por medio de este, el desarrollo de dos hardware que cumplieran con todos los requerimientos demandados por la arquitectura concebida para la gestión del protocolo de comunicación USB CMC ACM que permitiera la programación de la tarjeta LAB-VEE. En la figura 2 se puede apreciar un diagrama en bloques de dichos hardware, los cuales han consistido en dos tarjetas, una para fungir como transmisor de la programación/comunicación, la cual se conecta a uno de los puertos USB de la computadora y otro que ha sido integrado en la tarjeta desarrollada para LAB-VEE. En conjunto, estos hardware permiten que el IDE de Arduino o cualquier otro programa informático pueda comunicarse con una placa de Arduino o microcontrolador AVR, permitiendo que se le pueda cargar un Sketch (Firmware), así como también posibilita la comunicación UART con este en todo momento.

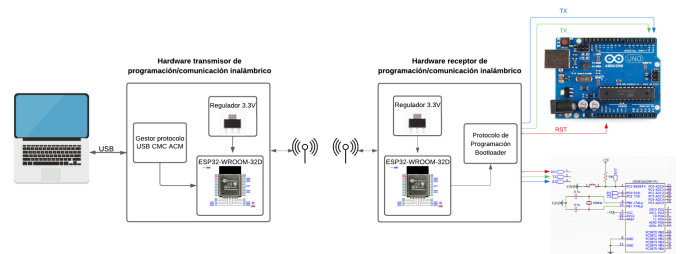


Figura 2. Diagrama en bloques del hardware de transmisión y del hardware de recepción para el programador inalámbrico.
Fuente: Elaboración propia

El hardware del transmisor se ha podido desarrollar para cumplir la función de interpretar el protocolo de comunicación USB CMC ACM, realizar el procesamiento de este por medio de un firmware creado para el ESP32-WROOM-32D y posteriormente enviar los datos recibidos desde la computadora, bajo el nuevo protocolo, de manera inalámbrica. Mientras que el hardware del receptor se ha desarrollado para recibir de manera inalámbrica los datos enviados por el transmisor, mediante otro ESP32-WROOM-32D que haciendo uso de un firmware cargado procesa estos datos llevándolos finalmente al protocolo de programación Bootloader vía USB, utilizado por Arduino, lográndose así la programación del microcontrolador.

IV. RESULTADOS OBTENIDOS

Durante el desarrollo de la tarjeta de LAB-VEE para el entrenamiento en lenguaje de programación y dispositivos programables, se logró crear e implementar el programador inalámbrico como parte de la misma. Esta solución ha permitido prescindir de la tradicional conexión física por medio de un cable USB entre la computadora y el dispositivo programable de la familia AVR a ser programado. Para su validación se llevaron a cabo pruebas exhaustivas para asegurar la estabilidad y fiabilidad de la comunicación, garantizando que pueda ser realizada de manera segura y sin errores.

Durante las pruebas realizadas al dispositivo, se utilizó un analizador lógico para contrastar la gestión de los datos de este en relación con la comunicación USB. En la figura 3 se puede apreciar una trama de datos enviada por el IDE de Arduino a través de comunicación USB, para cargar a la tarjeta de LAB-VEE un sketch que ocupaba un 95% de la memoria flash. De igual manera, en la figura 4 se muestra la misma trama de datos; sin embargo, esta vez para cargarlo por medio del programador inalámbrico desarrollado. Al comparar estas dos tramas, es posible identificar que el dispositivo tiene una fiabilidad del 100%, debido a que no se presentan pérdidas, por errores, al recibir los datos contenidos en la trama.

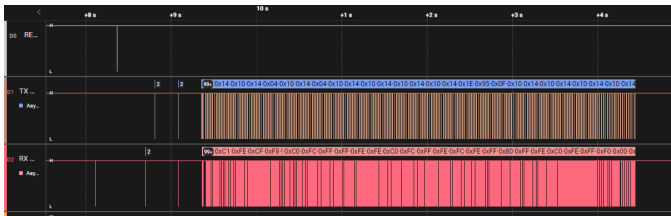


Figura 3. Trama del firmware de prueba con cable USB, para sketch que ocupa 95% de la memoria flash.

Fuente: Elaboración propia

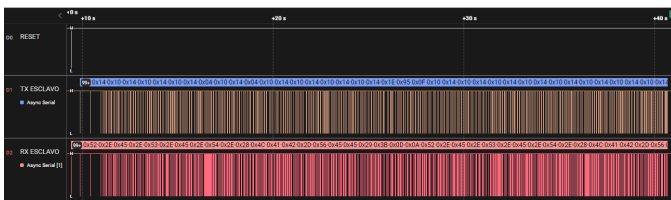


Figura 4. Trama del firmware de prueba con programador inalámbrico, para sketch que ocupa 95% de la memoria flash.

Fuente: Elaboración propia

Al analizar las gráficas, también se pueden identificar los tiempos requeridos para la programación, donde la comunicación USB realiza la carga del Sketch en un tiempo de unos 6.3 segundos, mientras que la inalámbrica requiere unos 33 segundos, lo cual significa que el programador inalámbrico necesita 5.23 veces más tiempo que el USB para cargar un sketch que ocupe el 95% de la memoria flash.

Además de realizar estas comparativas, el programador inalámbrico se sometió a un uso continuo durante un periodo de tiempo de más de 3 horas, en el cual se utilizó para programar la tarjeta de LAB-VEE 35 veces, así como también, por medio de los sketches cargados se hizo uso de comunicación serial con el IDE de Arduino para mantener una comunicación UART continua. De manera que, a partir de las pruebas de validación, a las que fue sometido el programador inalámbrico desarrollado, se puede concluir, que además de poseer una fiabilidad del 100% en la gestión de los datos, también posee una estabilidad del 100% para un uso continuo convencional.

Luego de haberse desarrollado los hardware y realizado las validaciones, se ha podido lograr el objetivo principal de desarrollar el programador inalámbrico que permitiera que la tarjeta de LAB-VEE, para el entrenamiento en lenguajes de programación y dispositivos programables, pudiera tener comunicación y ser programada de manera inalámbrica desde la computadora. De manera específica con el desarrollo de este dispositivo se logró que este permitiera la sustitución del cable USB usado tradicionalmente con las placas de Arduino para su programación y uso, un diseño que pudiera ser utilizado dentro de las dimensiones designadas, que el dispositivo no solo realizara la programación del microcontrolador, sino que también permitiera comunicación serial tipo UART, que este disponga de baja posibilidad de errores de comunicación y alta estabilidad de la misma, que el dispositivo desarrollado fuera de bajo costo y que permitiera una conectividad segura punto a punto.

En la figura 5, se puede visualizar el hardware desarrollado para el transmisor del programador inalámbrico, el cual es un dispositivo compacto que se conecta a la computadora vía USB, realizando la interpretación del protocolo de comunicación USB CMC ACM, para luego, mediante un ESP32-WROOM-32D convertir este a un protocolo propio que finalmente permite enviar los datos recibidos desde la computadora al hardware del receptor, ya sean para programación o comunicación.



Figura 5. Transmisor desarrollado para el programador inalámbrico.

Fuente: Elaboración propia

La figura 6 presenta la tarjeta que se le ha incorporado el programador inalámbrico, siendo ambos desarrollados para ser compatibles con Arduino, a partir de la cual se puede identificar que posee varios módulos de entrada, salida y HUBS, así como puertos para la comunicación con los periféricos incorporados en el microcontrolador, el cual es un AVR ATmega328P. Estos módulos incorporan varios

periféricos que pueden ser gestionados por el microcontrolador para crear dispositivos electrónicos embebidos. Además, cada módulo posee uno o varios terminales RJ12 mediante los cuales es posible conectarlos a uno de los terminales RJ12 de los puertos del microcontrolador, ya sea de manera directa o a través del uso de los HUBS.



Figura 6. Tarjeta de LAB-VEE para el entrenamiento en lenguajes de programación y dispositivos programables.
Fuente: Elaboración propia

Los módulos de entrada que incorpora la tarjeta incluyen:

- Potenciómetro Analógico (AP) de 10k ohmios, que permite ajustar un voltaje analógico entre 0 y 5 voltios de acuerdo al ángulo de rotación de la perilla.
- Resistencia Dependiente de Luz (LDR), diseñada para variar el voltaje analógico entre 0 y 5 voltios en función de la luz que recibe.
- Sensor Táctil (TOUCH), el cual permite la posibilidad de cambiar su salida digital a 0 en ausencia de contacto y a 1 cuando es tocado.
- Receptor Infrarrojo (IR), este capta comandos de un control remoto infrarrojo a través del protocolo de comunicación serial de un solo hilo, permitiendo identificar el botón del control que ha sido presionado.
- Identificación por Radio Frecuencia (RFID), este módulo decodifica el código de identificación de tarjetas o llaveros RFID mediante el protocolo de comunicación SPI.
- Balanza (BAL), realiza la función de llevar el peso de un objeto a un valor digital equivalente, permitiendo su conversión a una unidad de medida de masa, mediante el protocolo de comunicación SPI.
- Teclado Matricial (KP) 4x4, permite que el usuario presione 1 de sus 16 botones y envía este por medio del protocolo de comunicación I2C.

Los módulos de salida disponibles en la tarjeta se encuentran:

- Bocina (SPEAKER), mediante la comunicación UART este permite seleccionar y reproducir archivos MP3, además de otras funciones, para que el usuario los pueda reproducir y gestionar.

- Pantalla de 7 segmentos (DISPLAY), por medio de la comunicación I2C es capaz de mostrar números Binario Codificado en Decimal (BCD) desde el 0 hasta el 99.
- Diodo Emisor de Luz (LED) Rojo Verde Azul (RGB), puede ser controlado mediante estados lógicos 1 y 0 o mediante Modulación por Ancho de Pulso (PWM) para controlar el color de la luz que emite y su intensidad.
- Pantalla (SCREEN) Emisora de Luz Orgánica (OLED) 128x64, mediante la comunicación I2C esta puede ser usada para presentar caracteres y gráficas.
- Proyecto Intercambiable, adaptable según el tipo de práctica a realizar, el cual por defecto corresponde a un proyecto de recreación de vuelo de una nave, incorporando dos terminales RJ12 para la interacción. El terminal SERVOS es el que permite recrear los movimientos de vuelo de una nave posicionada en el módulo, mediante el control de dos servomotores. El terminal GYROS permite medir los movimientos de la navegación de la nave mediante un giroscopio analógico de tres ejes. Este módulo también puede ser utilizado para que la nave de un videojuego replique los movimientos realizados por la nave del proyecto.

En cuanto a los módulos HUBS, la tarjeta posee 4 de estos, donde cada uno posee 1 terminal RJ12 de entrada y 4 terminales RJ12 de salida. El terminal de entrada es el del centro, mientras que los terminales de salida están a ambos lados del terminal de entrada y numerados de acuerdo a su uso.

Finalmente, para la comunicación con el microcontrolador, la tarjeta posee 7 puertos:

- Puerto de Entrada y Salida Digital (DIGITAL IN/OUT), con 4 terminales RJ12 que ofrecen acceso a 16 pines digitales, 4 pines por cada terminal. Estos pines pueden ser configurados y utilizados por separado como entradas o salidas.
- Pines de Entrada y Salida de Propósito General (GPIO), incluye un terminal RJ12 con 4 pines digitales, los cuales son controlados de manera individual y directamente por el microcontrolador.
- Convertidor Análogo Digital (ADC), dotado de un terminal RJ12 que conecta 4 pines analógicos de 10 bits, los cuales son gestionados directamente por el microcontrolador.
- Dos puertos de Modulación por Ancho de Pulso (PWM), cada uno con un terminal RJ12 que proporciona 4 señales de PWM de 12 bits, donde cada uno de estos pueden ser configurados de manera independiente.
- Interfaz Periférica Serial (SPI), posee un terminal RJ12 que es manejado de manera directa por el microcontrolador.
- Puerto UART, con un terminal RJ12 que es gestionado de manera directa por el microcontrolador.

- Circuito Inter-Integrado (I2C), dispone de 4 terminales RJ12 que son manejados de manera directa por el microcontrolador.

La inclusión de esta tarjeta de entrenamiento en el ecosistema LAB-VEE con todos los módulos que posee, ha enriquecido significativamente la oferta de herramientas disponibles para el entrenamiento en lenguajes de programación y dispositivos programables. Su facilidad de uso, tanto para principiantes como para profesionales, y la posibilidad de conexión inalámbrica para la programación y comunicación, han ampliado las posibilidades de enseñanza y aprendizaje, permitiendo complementar la secuenciación de tarjetas para la capacitación de estudiantes K-12 en la adquisición de competencias tecnológicas de áreas STEM. Estas tarjetas se pueden integrar con una estación de trabajo que ha desarrollado LAB-VEE, la cual incluye una amplia gama de equipos necesarios en un entorno de laboratorio de electricidad/electrónica, la cual se conecta directamente a la computadora para la interacción con dichos equipos. Además, las tarjetas pueden ser insertadas en esta estación para aprovechar todos los recursos disponibles en el ecosistema LAB-VEE.

En la figura 7, se presenta un beneficio extra que ha resultado de este desarrollo, el cual corresponde a un shield compatible con Arduino UNO, que puede ser utilizado con dichas placas para establecer comunicación inalámbrica entre la computadora y el Arduino para permitir su programación y comunicación serial de igual manera que con el cable USB.

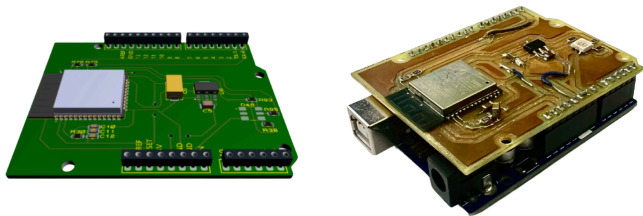


Figura. 7. Shield receptor del programador inalámbrico, compatible con Arduino UNO.
Fuente: Elaboración propia

IV. DESAFIOS SUPERADOS

Durante el desarrollo de esta tarjeta para el entrenamiento en lenguajes de programación y dispositivos programables, se enfrentaron y superaron varios desafíos técnicos que marcaron el camino hacia la innovación y la eficiencia de esta tarjeta para que, además de todos los recursos técnico-educativos que posee, también pudiera contar con programación y comunicación inalámbrica. Por lo cual, uno de los logros más significativos fue superar los desafíos técnicos de la comunicación inalámbrica, que abarcó desde la elección de dispositivos y tecnologías, hasta protocolos de procesamiento de datos y la implementación de técnicas avanzadas para

asegurar una transmisión de datos fiable y eficiente. La optimización de la conexión inalámbrica y la corrección de errores se volvieron fundamentales para mantener la fiabilidad de la información y la estabilidad de la conexión, al mismo tiempo que se implementaron medidas de seguridad como el cifrado y la autenticación para fortalecer la protección de la conexión.

Otro gran desafío consistió en alcanzar la mayor velocidad de comunicación inalámbrica, para lograr la mejor adaptación al sustituir el método cableado por el inalámbrico. Con lo cual se logró minimizar los tiempos de carga y mejorar el desfase que podría percibir el usuario sin que esto requiriera sacrificar la calidad de la transmisión. A esto se sumó el reto de realizar ajustes en la configuración y protocolo de comunicación para permitir una transferencia de datos efectiva y eficiente mediante la conexión inalámbrica. De igual manera, se implementaron algoritmos de corrección de errores para combatir las interferencias y pérdidas de datos inherentes a las comunicaciones inalámbricas, asegurando así la fiabilidad en la transmisión. Este enfoque estuvo regido por el interés de alcanzar una optimización general del rendimiento del programador inalámbrico, así como reducir los tiempos de programación inalámbrica y ofrecer una experiencia de usuario de alta valoración hacia los beneficios del uso del dispositivo desarrollado.

V. CONCLUSIONES

En conclusión, el diseño y desarrollo del programador inalámbrico creado para tarjetas Arduino, microcontroladores compatibles con este ecosistema y más aún la tarjeta orientada a facilitar el entrenamiento de estudiantes K-12 en competencias tecnológicas de las áreas STEM, subraya una innovación y un aporte significativo para los profesionales, estudiantes y aficionados que hacen uso de dicho ecosistema. El programador no solo cumple con requerimientos técnicos y mecánicos específicos, sino que también supera las expectativas en términos de funcionalidad, al ofrecer una alternativa segura, de bajo costo y de alto rendimiento para la programación inalámbrica de dispositivos compatibles con Arduino. De manera que este logro no solo enriquece el ecosistema LAB-VEE y sus usuarios, sino que también proporciona un recurso valioso para la comunidad Arduino. Por lo tanto, este proyecto en general no solo demuestra la viabilidad técnica de la programación y comunicación inalámbrica para dispositivos compatibles con Arduino, sino que también establece un precedente en la innovación educativa y tecnológica, ofreciendo a través de la tarjeta de entrenamiento en lenguajes de programación y dispositivos programables de LAB-VEE nuevas posibilidades para el aprendizaje de competencias tecnológicas en áreas STEM, lo cual en conjunto fomenta e incentiva el desarrollo de proyectos de dispositivos electrónicos embebidos. La iniciativa de dar solución a la problemática planteada en este

artículo y la superación de los desafíos encontrados para lograrlo refleja el compromiso continuo de LAB-VEE TechMakers con la creación y mejora de recursos educativos que enriquecen la metodología STEM, para que se beneficien educadores, estudiantes y aficionados, ampliando las fronteras de lo que es posible en el ámbito de la tecnología educativa y el desarrollo de hardware.

REFERENCIAS

- [1] M. A. Tosini, L. Leiva, M. O. Vázquez, O. E. Goñi, and J. M. Toloza, "Técnicas de optimización de soluciones en sistemas embebidos," presented at the XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja), 2021. Accessed: Mar. 23, 2024. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/121675>.
- [2] Ó. Sanclemente Carretero, "Casa domótica con arduino," Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València, 2016. Accessed: Mar. 23, 2024. [Online]. Available: <https://riunet.upv.es/handle/10251/75797>.
- [3] J. Novillo Vicuña, D. L. Hernández Rojas, B. Mazón Olivo, J. R. Molina Ríos, and O. Cárdenas Villavicencio, "Arduino y el internet de las cosas," 2018. Accessed: Mar. 23, 2024. [Online]. Available: <https://dialnet.unirioja.es/servlet/libro?codigo=727827>.
- [4] A. A. Galadima, "Arduino as a learning tool," in 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Sep. 2014, pp. 1–4. doi: 10.1109/ICECCO.2014.6997577.
- [5] D. Arranz Ortega, "Marco para el desarrollo de aplicaciones ADA sobre microcontroladores STM32," Framework for the development of ADA applications on microcontrollers STM32, Sep. 2019, Accessed: Mar. 23, 2024. [Online]. Available: <https://repositorio.unican.es/xmlui/handle/10902/17843>
- [6] E. Zafra Ratia, "Diseño de controlador MIDI con comunicación Bluetooth en microcontrolador MSP430," 2016, Accessed: Mar. 23, 2024. [Online]. Available: <https://idus.us.es/handle/11441/48830>.
- [7] J. C. Herrero Herranz and J. Sánchez Allende, "Una mirada al mundo Arduino," *Tecnol. Desarro.*, no. 13, p. 11, 2015.
- [8] M. El-Abd, "A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions," *Int. J. Eng. Pedagogy IJEP*, vol. 7, no. 2, Art. no. 2, May 2017, doi: 10.3991/ijep.v7i2.6845.
- [9] A. Nayyar and V. Puri, "A review of Arduino board's, Lilypad's & Arduino shields," in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), Mar. 2016, pp. 1485–1492. Accessed: Mar. 23, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/7724514?denied=>.
- [10] C. Peña, *Arduino IDE: Domina la programación y controla la placa. RedUsers*, 2020.
- [11] H. Li, Y. Xu, F. Wu, and C. Yin, "Research of 'Stub' remote debugging technique," in 2009 4th International Conference on Computer Science & Education, Jul. 2009, pp. 990–994. doi: 10.1109/ICCSE.2009.5228140.
- [12] J. Dolinay, P. Dostálek, and V. Vašek, "Advanced Debugger for Arduino," *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 12, no. 2, Art. no. 2, 58/01 2021, doi: 10.14569/IJACSA.2021.0120204.
- [13] N. GOILAV and G. LOI, *Arduino: Aprender a desarrollar para crear objetos inteligentes. Ediciones ENI*, 2016.
- [14] D. G. F. Tomalá and D. S. Espinoza, "Sistema domótico por comando de voz basado en Arduino para personas con dificultades motrices," *Rev. Científica Tecnológica UPSE*, vol. 9, no. 1, Art. no. 1, Jun. 2022, doi: 10.26423/rctu.v9i1.665.
- [15] J. Noble and N. Joshua, *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*, 1st ed. O'Reilly Media, Inc., 2009.
- [16] R. Then and M. M. Larrondo-Petrie, "Portable Laboratory for Electrical Engineering Education: The LAB-VEE Ecosystem Developed in Latin America and the Caribbean," presented at the 2023 ASEE Annual Conference & Exposition, Jun. 2023. Accessed: Mar. 23, 2024. [Online]. Available: <https://peer.asee.org/portable-laboratory-for-electrical-engineering-education-the-lab-vee-ecosystem-developed-in-latin-america-and-the-caribbean>.
- [17] U. Sari, H. Çelik, H. M. Pektaş, and S. Yalçın, "Effects of STEM-focused Arduino practical activities on problem-solving and entrepreneurship skills," *Australas. J. Educ. Technol.*, vol. 38, no. 3, Art. no. 3, May 2022, doi: 10.14742/ajet.7293.
- [18] G. Organtini, "Arduino as a tool for physics experiments," vol. 1076, p. 012026, Sep. 2018, doi: 10.1088/1742-6596/1076/1/012026.
- [19] S. Muzaffar and I. M. Elfadel, "Beyond Arduino: A Guide for the Perplexed," in 2021 IEEE International Symposium on Circuits and Systems (ISCAS), May 2021, pp. 1–5. doi: 10.1109/ISCAS51556.2021.9401492.
- [20] J. M. C. García, J. B. C. Sánchez, and E. G. Parada, "Low-cost JTAG debugger with Wi-Fi interface," in 2022 Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (XV Technologies Applied to Electronics Teaching Conference), Jun. 2022, pp. 1–10. doi: 10.1109/TAE54169.2022.9840601.
- [21] "J-Link WiFi," Embedded Development Tools by SEGGER Microcontroller. Accessed: Mar. 23, 2024. [Online]. Available: <https://shop.segger.com/debug-trace-probes/debug-probes/j-link/j-link-wifi>.
- [22] A. Ismailov and Z. Jo'rayev, "Study of arduino microcontroller board," Mar. 2022.
- [23] A. Gibb, "NEW MEDIA ART, DESIGN, AND THE ARDUINO MICROCONTROLLER: A MALLEABLE TOOL," 2010. Accessed: Mar. 23, 2024. [Online]. Available: <https://www.semanticscholar.org/paper/NEW-MEDIA-ART%2C-DESIGN%2C-AND-THE-ARDUINO-A-MALLEABLE-Gibb/b1dde1e6569ccc5567d62c88d8ea947629baf5d>.
- [24] M. A. Chouksey, D. S. Changlani, and S. T. Ali, "Zigbee based Wireless Body Area Sensor Network for patient's physiological parameter monitoring," *Int. J. Eng.*, vol. 2, no. 2456, 2017.
- [25] J. A. Romero Gamboa, "Modelamiento e implementación de una arquitectura de control y monitoreo de bajo costo basado en microcontroladores para un entorno IIOT," Repos. Tesis - UNMSM, 2023, Accessed: Mar. 23, 2024. [Online]. Available: <https://cybertesis.unmsm.edu.pe/handle/20.500.12672/21143>.
- [26] M. A. Toledo, "Seguimiento y control de ganado vacuno mediante geoposicionamiento," *Proj. Des. Manag.*, vol. 4, no. 1, Art. no. 1, Mar. 2022, doi: 10.35992/pdm.v4i1.828.
- [27] L. Louis, "Working Principle of Arduino and Using it as a Tool for Study and Research," *Int. J. Control Autom. Commun. Syst.*, vol. 1, no. 2, pp. 21–29, Apr. 2016, doi: 10.5121/ijcacs.2016.1203.
- [28] P. Izquierdo González, "Aparato detector de dispositivos USB maliciosos," Detector device for malicious USB devices, Jun. 2022, Accessed: Mar. 23, 2024. [Online]. Available: <https://repositorio.unican.es/xmlui/handle/10902/25843>.
- [29] M. C. Ramon, "Arduino IDE and Wiring Language," in *Intel® Galileo and Intel® Galileo Gen 2: API Features and Arduino Projects for Linux Programmers*, M. C. Ramon, Ed., Berkeley, CA: Apress, 2014, pp. 93–143. doi: 10.1007/978-1-4302-6838-3_3.
- [30] "Bootloader | Arduino Documentation." Accessed: Mar. 23, 2024. [Online]. Available: <https://docs.arduino.cc/retired/hacking/software/Bootloader/>
- [31] N. Cvetković, P. Milenković, V. Lapčević, V. Rajović, M. Karličić, and G. Rapić, "Wireless Programming of the AVR Microcontroller Family," in 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Jul. 2022, pp. 1–6. doi: 10.1109/ICECET55527.2022.9872902.
- [32] A. Singh, "Applications of IoT in Agricultural System." Rochester, NY, May 31, 2019. doi: 10.2139/ssrn.3397022.
- [33] L. J. Bradley and N. G. Wright, "Optimising SD Saving Events to Maximise Battery Lifetime for ArduinoTM/Atmega328P Data Loggers," *IEEE Access*, vol. 8, pp. 214832–214841, 2020, doi: 10.1109/ACCESS.2020.3041373.
- [34] F. Ahmed, S. Alim, M. Islam, K. R. Kawshik, and S. Islam, "433 MHz (Wireless RF) Communication between Two Arduino UNO," 2016.

- Accessed: Mar. 23, 2024. [Online]. Available: <https://www.semanticscholar.org/paper/433-MHz-%28Wireless-RF%29-Communication-between-Two-UNO-Ahmed-Alim/cf18fb9e63b832a535f28db724239251616a6c14>.
- [35] M. Mahbub, "Automated control signal reception acknowledgement system using nRF24L01P wireless transceiver module and Arduino," EAI Endorsed Trans. Mob. Commun. Appl., vol. 5, no. 16, pp. e3–e3, Jan. 2019, doi: 10.4108/eai.14-12-2018.159337.
- [36] J. Gao and Y. Zhu, "Design and Implementation of Wireless Communication System Based on CC1100," IOP Conf. Ser. Mater. Sci. Eng., vol. 782, p. 052019, Apr. 2020, doi: 10.1088/1757-899X/782/5/052019.
- [37] B. Mathew, N. D'souza, S. Gawad, and T. Mathai, "Design of Wireless Notice Boards for Long Distance Data Transmission by using Bluetooth and nRF2410," vol. 04, no. 05.
- [38] A. Ghasemi, K. Yousefi, B. Yazdankhoo, and B. Beigzadeh, "Cost-effective Haptic Teleoperation Framework: Design and Implementation," in 2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM), Dec. 2023, pp. 253–258. doi: 10.1109/ICRoM60803.2023.10412539.
- [39] "LAB-VEE Educational TechMakers." [Online]. Available: <https://www.labvee.com/> [Accessed: 9-Feb-2024].