

Conference Paper

Pedagogical tool for programming and reading a PIC microcontroller by means of an embedded Raspberry pi system and an easy-to-use graphical interface

Herramienta pedagógica de programación y lectura de un microcontrolador PIC por medio de un sistema embebido Raspberry pi y una interfaz gráfica de fácil uso

Mario Ernesto Serrato Maldonado, Robin Alfonso Blanco Canon, and Fabian Barrera Prieto

universidad ECCI, Bogotá D.C., Colombia

Corresponding Author:

Mario Ernesto Serrato

Maldonado

marioe.serratom@ecc.edu.co

Received: 24 December 2019

Accepted: 2 January 2020

Published: 8 January 2020

Publishing services provided by
Knowledge E

© Mario Ernesto Serrato

Maldonado et al. This article is distributed under the terms of

the [Creative Commons](#)

[Attribution License](#), which

permits unrestricted use and redistribution provided that the original author and source are credited.

Selection and Peer-review under the responsibility of the SIIPRIN-CITEGC Conference Committee.

Abstract

In this paper is presented the development of a system for the control of a PIC18F4550 microcontroller from a user-friendly interface implemented in a Raspberry Pi 3, where the detailed description of full hardware and software is made, as goal to make a tool that reduces the problems when someone is learning microcontrollers in a classic subject of this technology in the ECCI University. A better understanding of the microcontroller registers and functions was obtained in the tests, thus this system will be an useful tool to the teaching of microcontrollers.

Resumen. En el siguiente trabajo se presenta el desarrollo de un sistema para el control de un microcontrolador PIC18F4550 a partir de una interfaz de fácil uso implementada en una Raspberry Pi 3, donde se hace la descripción a detalle de todo el hardware y software implementado, con la finalidad de hacer una herramienta que reduzca los problemas a la hora de aprender microcontroladores en una materia clásica de esta tecnología en la Universidad ECCI. Se obtuvo en las pruebas un mejor entendimiento de los registros y funciones del microcontrolador, siendo esto un complemento útil a la enseñanza de microcontroladores.

Keywords: Raspberry pi 3, Pic, ICSP, Programing, Python

1. Introducción

Los sistemas que implementan dispositivos como por ejemplo los microcontroladores requieren de la integración de hardware y software, los procesadores digitales como

 OPEN ACCESS

lo son los microcontroladores requieren habilidades de programación en lenguajes específicos, además de conceptos de principalmente la lógica electrónica digital.

En la institución universitaria ECCI se presentan problemas en el proceso de formación de un estudiante para el aprendizaje de microcontroladores, a causa de que se enseña a programar inicialmente en una unidad de computo, donde, no es necesario configurar previamente la máquina y no existe la necesidad de una conexión de periféricos demasiado compleja, sin embargo un microcontrolador requiere de una programación previa de la máquina y conexiones específicas teniendo diversas variantes algunas complejas y otras sencillas, lo cual hace que el dispositivo sea propenso a errores de programación y montaje.

Dentro de los errores más frecuentes que se presentan en la implementación de microcontroladores para el desarrollo de dispositivos, fueron causados por la confusión de la sintaxis del lenguaje en uso, junto con la equivocación de los estudiantes en las estructuras de programación y el uso de los periféricos. Por lo general las problemáticas de sintaxis se puede identificar por medio del compilador del programa, pero los desaciertos en el uso del dispositivo no se pueden identificar fácilmente.

La visualización de esta problemática hizo que se planteara la posibilidad del desarrollo de una herramienta que pudiera solventar las dificultades en el aprendizaje de microcontroladores, por medio de la identificación de errores a través de la depuración en software, esta herramienta permite solucionar los problemas presentes al usar un microcontrolador, cuando no se sabe a ciencia cierta si el montaje falla o si la verdadera problemática está presente en la estructura de programación.

Debido a que la dificultad de implementación de un sistema aumenta a medida que se complejiza la aplicación, una herramienta de depuración ayuda a que los tiempos de desarrollo y aprendizaje disminuyan mientras se desarrolla la actividad.

2. Marco teórico

El protocolo de comunicación "In Circuit Serial Programming" (ICSP) fue diseñado por la empresa Microchip® para comunicar sus microcontroladores (PIC) con algún dispositivo, permitiéndole acceder a la memoria interna del microcontrolador y a su vez facilitando el manejo de los periféricos que componen la herramienta.

El protocolo ICSP se compone por una trama de 20 bits, donde los cuatro primeros bits definen el tipo de operación que efectuara el microcontrolador y los 16 bits restantes manejan los datos, la comunicación utiliza el método serial donde el bit menos significativo va primero en la trama de datos, trabajando de manera independiente los 16

bits de datos y los cuatro bits de comando precedidos de una señal de reloj para cada pulso.

El ICSP es un protocolo que está ya definido en el dispositivo por esto no hay que hacer una configuración previa para usarlo el único requisito para que se pueda entrar en modo de programación es suministrar 9 voltios al pin de "master clear" (MCLR). (Technology, 2010)

La Raspberry Pi es un sistema embebido que surgió en 2012, que cuenta con diferentes características (procesador ARM, memoria RAM, memoria SD flash (disco duro), periféricos, sistema operativo, Wifi, Bluetooth, entre otras), además se define como una computadora de pequeñas dimensiones con altas frecuencias de trabajo. (System on Chip).(Maksimović, Vujović, Davidović, Milošević, & Perišić, 2014)

Esta plataforma permite implementar y correr algoritmos de alta complejidad en tiempo real, a través del lenguaje de programación Python.

3. Teoría del dominio y trabajos previos

A nivel académico se han desarrollado diferentes enfoques y métodos para la enseñanza de sistemas microcontrolados donde normalmente se compone de una parte teórica y una práctica, buscando el objetivo de solucionar problemas de ingeniería implementando los conocimientos vistos en la asignatura, pero los elementos físicos necesarios para realizar los desarrollos son la unidad de computo, el microcontrolador y los programadores que son las herramientas necesarias para el desarrollo de actividades, actualmente las herramientas disponibles para la depuración y programación de microcontroladores proporcionadas por el fabricante encontramos el PM3, ICD2, ICD3, PICkit3, PICkit4 mostradas en la figura Fig. 1 sin embargo, son herramientas costosas y que vienen coligadas a MPLAB® lo cual limita la portabilidad de los sistemas desarrollados.(Ibrahim, 2014)

Actualmente los proyectos de desarrollo enfocados al control de entornos requiere dispositivos con la posibilidad de conectarse físicamente con el medio, pero los limitantes de los sistemas que usan microcontroladores para captar medios físicos son la limitada capacidad de procesamiento que disponen para el uso avanzado; para suplir esto investigadores realizaron desarrollos basados en la tecnología "IoT", Haciendo aplicaciones de sistemas de alta complejidad como por ejemplo un medidor de energía eléctrica subida a la nube controlado con un PIC18F4550, otro caso de estudio fue la combinación de un PIC Y una Raspberry pi para la gestión de un proceso que consistía en medir y controlar diferentes parámetros como: corriente, tensión, temperatura,



Figure 1: Programadores y depuradores.

intensidad lumínica y nivel de agua.(Raguvaran & Thiyagarajan, 2015)(Raguvaran & Thiyagarajan, 2015)

4. Metodología

En este proyecto se están teniendo diferentes etapas para el desarrollo de la implementación de la programación en tiempo real para el PIC18F4550 las cuales son las enumeradas a continuación.

4.1. Creación de un programa en lenguaje ensamblador

El diseño del programa se planteó haciendo uso de una rutina que se ejecutara en un bucle infinito donde se redujeran al máximo las líneas necesarias de ejecución, usando solamente uno de los puertos del microcontrolador, implementado el programa básico de encender y apagar un led como se puede observar en la Fig. 2.

Para poder visualizar la ejecución del programa creado se usó un oscilador externo, el cual fue configurado para que trabajara a una frecuencia de 10Hz; fue implementado el integrado Im555 en configuración astable, estableciendo la frecuencia planteada y se procediendo a montar el generador de pulsos.

La prueba del programa se realizó haciendo uso del PIC18F4550 el cual fue programado con el diagrama de flujo planteado en la Fig. 2, este programa ejecuto las secuencias necesarias y funciono correctamente.

En el desarrollo del proyecto resulto necesario definir los códigos binarios que se codificaron por medio del compilador. La codificación binaria correspondiente a cada instrucción del diagrama de flujo presentado en la Fig. 2, en donde se detalla la configuración de los registros TRIS y LAT del puerto además de las instrucciones de limpieza, llenado, no operación y redireccionamiento de memoria. Esta información se puede encontrar en el "datasheet" del PIC18F4550 (Microchip, 2009) en la página 68 y 316 respectivamente. Los códigos binarios presentes en (Microchip, 2009, Pag 68, Table5-1) y (Microchip, 2009, Pag 316, Table 26-2) son guardados en la memoria de instrucciones del microcontrolador (ROM) la cual es una memoria no volátil permitiendo almacenar las instrucciones indefinidamente; a lo contrario de lo que sucede con la RAM, que guarda las variables utilizadas durante la ejecución de las instrucciones.

4.2. Uso del Protocolo de comunicación ICSP

El protocolo ICSP requiere cinco líneas de señal para su uso, la primera línea que se encarga de proporcionar 9V (voltios) para habilitar la comunicación, la segunda proporciona 5V (Voltios) de alimentación, la tercera otorga tierra (GND), la cuarta se encarga de generar los pulsos de reloj y la quinta línea se encarga de la sincronización del envío de los datos, estas conexiones se pueden encontrar en la hoja técnica del fabricante (Microchip, 2009)(Microchip Technology Inc, 2013).

Este protocolo es síncrono con características "half duplex" tan solo puede comunicar dos dispositivos donde uno de ellos es un PIC y otro suele ser un programador suministrado por el fabricante, usado solo por los microcontroladores desarrollados por Microchip® puesto que otros microcontroladores utilizan protocolos diferente dependiendo de la empresa productora.

Para poder realizar la comunicación ICSP desde algún tipo de dispositivo a un microcontrolador es necesario realizar la conmutación de 5V (Voltios) o 0V(voltios) a 9 V (Voltios), esto se requiere para permitir que el PIC se disponga a entrar en modo programación enviado los voltajes a la conexión "Master Clear" (MCLR), es importante hacer la salvedad de que no es que el dispositivo se disponga a ser programado, simplemente habilita el pin 40 (PGD) y el pin 39 (PGC) para que procesen la información enviada por algún elemento externo.

La estrategia utilizada para hacer la conmutación de voltajes requerida para él ICSP, se realizó haciendo uso de un amplificador de voltaje "booster" el cual se encarga de convertir los 5V (Voltios) de alimentación en 9V (Voltios) de salida, este dispositivo está conectado a un arreglo de transistores que por medio de señales digitales con un

voltaje de 3.3V es posible cambiar los voltajes enviados al Pin MCLR, la configuración diseñada para esta aplicación se pueden observar en la Fig. 3 y la combinación binaria de funcionamiento se plantea en la Tabla 1.

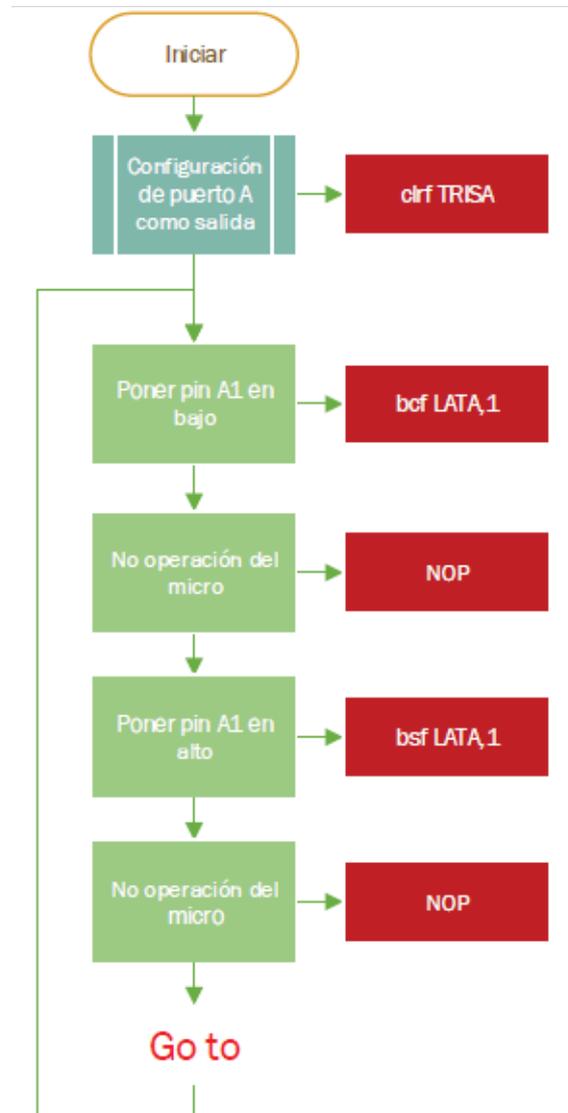


Figure 2: Programa base Encender y apagar un led, en el esquema se trabajan líneas en lenguaje ensamblador donde su significado es: clrf es poner en cero todo el registro, bcf es poner un bit específico en cero, bsf es poner un bit específico en uno, TRIS es un registro de configuración, LAT es un registro de salida, NOP es la no operación por un ciclo de bus (cuatro pulsos de reloj).

La trama de comunicación se compone por 20 bits por cada acción realizada que esta precedida por 4 bits que definen el tipo de operación que efectuara el microcontrolador, los 16 bits restantes dependiendo del tipo de operación definida tendrán características diferentes esto se puede ver en (Technology, 2010, Pag 33 fig4-1), las operaciones como tiene un orden específico de bits se pueden apreciar en (Technology, 2010, Pag 19 fig2-18). La comunicación ICSP en su trama de bits se envía primero los bits menos

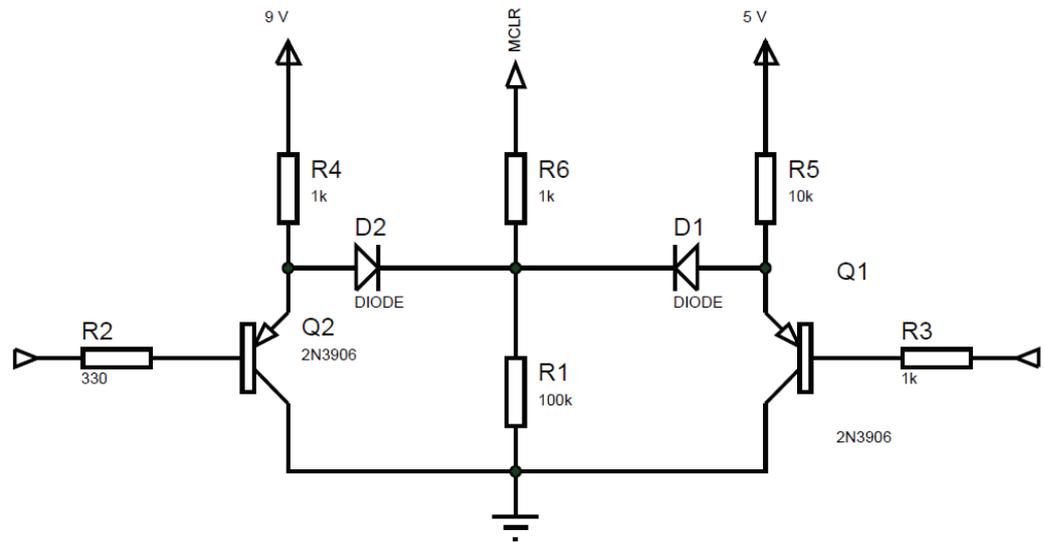


Figure 3: Sistema de conmutación de voltaje diseñado por medio de un arreglo de transistores.

significativos y por último los más significativos, esto se efectúa en los primeros 4 bits de comando y luego con los 16 bits restantes (dependiendo de la operación).

TABLE 1: "Look-Up tables" voltaje MCLR.

V_R2	V_R3	V_MCLR
0	0	9V
0	1	9V
1	0	5V
1	1	0V

4.3. implementación del lenguaje de programación Python en el sistema Raspberry Pi 3

La Raspberry Pi 3 es un sistema embebido mostrado en la Fig. 4 en la parte superior izquierda que permite hacer una gran cantidad de desarrollos de alto nivel, ya que las cualidades físicas que posee le permiten soportar aplicaciones que requieran una capacidad tanto de procesamiento como de memoria, para el desarrollo del proyecto se decidió hacer uso de esta herramienta no solo por las cualidades anteriormente mencionadas, fue también seleccionada por la cualidad de no necesitar de una unidad de cómputo y tampoco requerir un módulo de interpretación para poder manipular la memoria de instrucciones del PIC.

En Python se desarrolló todo el programa encargado de la lectura del PIC haciendo uso de los puertos GPIO y trabajando diferentes estrategias de programación para

solventar las necesidades del proyecto, se ha estado manejando el IDLE de este programa para el desarrollo de todo el programa de lectura.

Una de las cualidades de este sistema embebido es que todos sus pines solo trabajan con un voltaje de 3.3V sin embargo el PIC en los pines digitales trabaja con 5V, si este voltaje lo llegara a recibir la Raspberry esta se quemaría una solución fue el uso de un convertor bidireccional de 3.3V a 5V que se puede ver en la Fig. 4 donde se especifican las conexiones entre los dispositivos y el desarrollo del programa en Python en la Fig. 5.

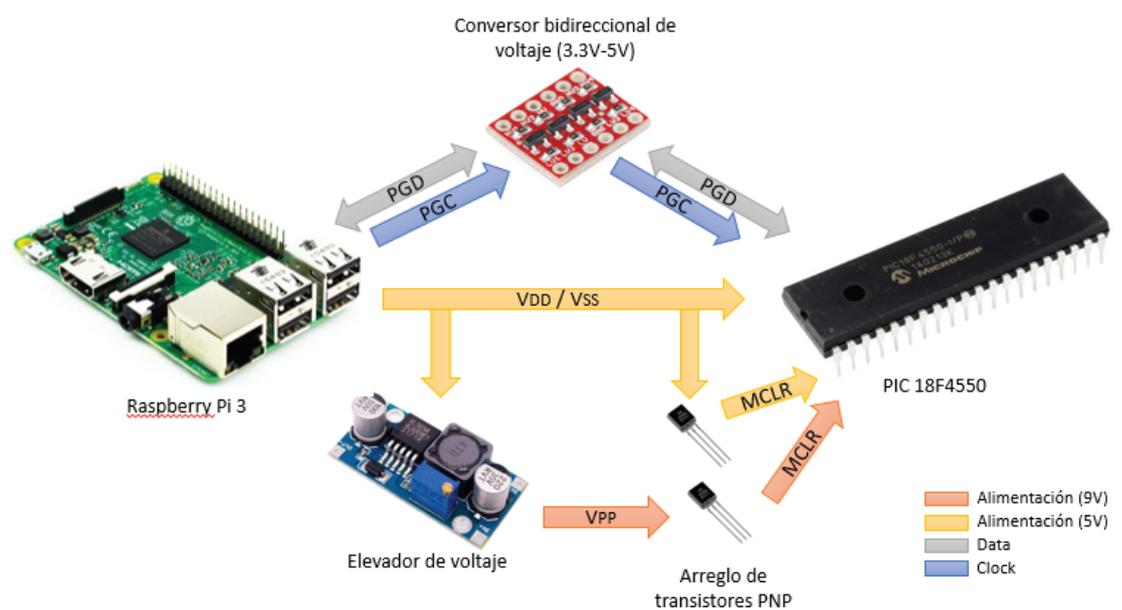


Figure 4: Conexión de dispositivos.

4.4. Ejecución de la lectura del programa

La lectura de la memoria de instrucciones del PIC se puede lograr haciendo uso de los comandos de operación de la 18 table 2-8 (Technology, 2010, Pag 18, Table2-8) manipulando solamente los que tienen la función de lectura y respetando las condiciones para él envío de datos. Para la lectura de la memoria ROM hay que tener en cuenta el orden en el que se programó específicamente el microcontrolador en este caso como lo muestra la Fig. 2, en este diagrama se puede evidenciar el orden que tienen las operaciones programadas que son las que definen la posición de memoria en la cual se almacenaran los dígitos binarios correspondientes al "opcode" (Microchip, 2009) Pag317, el almacenamiento de los datos es atípico a la forma en que se digitan las instrucciones en el compilador.



Figure 5: Programa base desarrollado en Python.

El guardado de las instrucciones tiene una lógica donde primero se almacena el registro que se va a usar y a consiguiente se ejecuta en la memoria la acción que se va a realizar sobre el registro.

En el ICSP la lectura tiene una estructura específica en la cual se contempla que los primeros 4 bits para comandos de lectura se mantienen siguiendo las condiciones para él envío de datos, aunque los próximos 16 bits se trabajan de una forma distinta, donde los 8 primeros bits se envían datos despreciados por el microcontrolador precedidos por su respectiva señal de reloj, y a partir del noveno bit es cuando el PIC genera los datos que tiene almacenados en la memoria esto se puede ver en la hoja técnica (Technology, 2010) pag33, es importante especificar que en el momento que se entra en modo de programación el microcontrolador se dirige a la posición 0x00 de su memoria de instrucción, para obtener una instrucción en lenguaje ensamblador es necesario efectuar dos veces una operación de lectura que cambie por si sola el apuntador de tabla.

4.5. Caso de validación de la herramienta programadora en desarrollo

La validación de la herramienta se efectuará por medio de un estudio con dos personas, donde se planteará una prueba en la que los participantes del estudio estarán en igualdad de condiciones: no tienen experiencia en programación de microcontroladores y actualmente están cursando la asignatura de microcontroladores, a ellos se les asignará realizar la misma tarea de programación.

El estudio se desarrollará evaluando que tan eficiente es la herramienta en proceso de desarrollo frente a las herramientas clásicas de programación en cuestiones de tiempo, comparando el comportamiento de dos sujetos solucionando un problema de una aplicación que requiera el uso de un microcontrolador, donde uno de ellos estará utilizando la herramienta en desarrollo junto las herramientas clásicas como el "datasheet" y páginas Web, mientras por otro lado el segundo colaborador estará usando tan solo las herramientas clásicas para programar microcontroladores.

Las personas que colaboraran en el estudio llenarán un consentimiento informado donde se les informará todo lo que compete al desarrollo de la evaluación del proyecto, de igual manera, será comunicado que tipo de beneficios obtendrán donde no existirá una retribución monetaria, sin embargo, obtendrán la posibilidad de ayudar en el desarrollo de nuevas herramientas educativas y un nuevo dispositivo tecnológico, será aclarado que el proceso a realizar no generará ningún daño físico o mental expresando que el apoyo a la evaluación del proyecto es totalmente libre y voluntario.

La forma de obtener una conclusión de esta validación será por medio de un registro fílmico y la aplicación de una encuesta al final, con el propósito de analizar las complicaciones y posibilidades de mejora de cada estrategia, en pro de medir la efectividad y factibilidad del uso de la herramienta en desarrollo en la enseñanza del curso de microcontroladores. En el caso del video se clasificarán las actividades más relevantes y se determinará el tiempo gastado total y en cada una de ellas, para observar si hay algún cambio en el uso de la herramienta en diseño en el proceso de ejecución de la tarea asignada.

5. Experimentos y Resultados

Ya planteada la teoría se procede con la implementación en el sistema Raspberry pi la cual se puede ver en la Fig. 5 donde por código en Python se controlan los pines GPIO para generar las señales digitales que conforman el protocolo de comunicación ICSP.

Para poder efectuar la comunicación se integró en una "protoboard" todos los requerimientos de hardware presentados en el punto 4.2 con las conexiones del oscilador externo al PIC y con todos los requerimientos del protocolo ICSP, el montaje se puede encontrar en la Fig. 6.

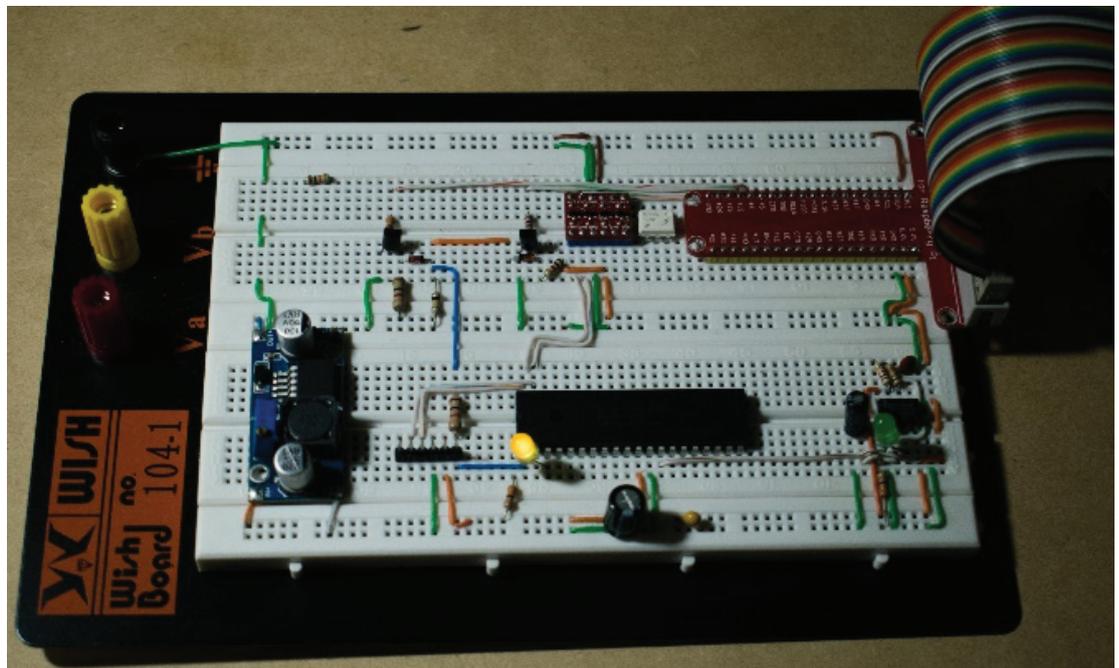


Figure 6: Montaje para comunicación ICSP PIC- Raspberry pi.

El montaje de la Fig. 6 fue ensamblado en una "protoboard" con una placa metálica para evitar problemas con señales electromagnéticas externas trabajando como una jaula de Faraday para aislar corrientes, entre la línea de alimentación de 5V (voltios) se dispuso un condensador para evitar algún tipo de rizado en la alimentación del circuito.

Por medio de código en la Raspberry se produjeron señales de pulsos que coincidieran con el protocolo ICSP obteniendo la señal presentada en la Fig. 7. Los resultados de la prueba de comunicación concuerdan con el análisis de los resultados esperados por el protocolo ICSP confirmando que todo lo obtenido concuerda con lo que se debería de obtener en la posición 00 00 00h.

6. Conclusiones y trabajos futuros

El uso de sistemas embebidos de gran capacidad y bajo costo como la Raspberry Pi como herramienta complementaria a la enseñanza de microcontroladores, permite hacer un proceso más didáctico y más idóneo como complemento a la teoría y práctica que se imparten normalmente en esta materia, además de la portabilidad e integración

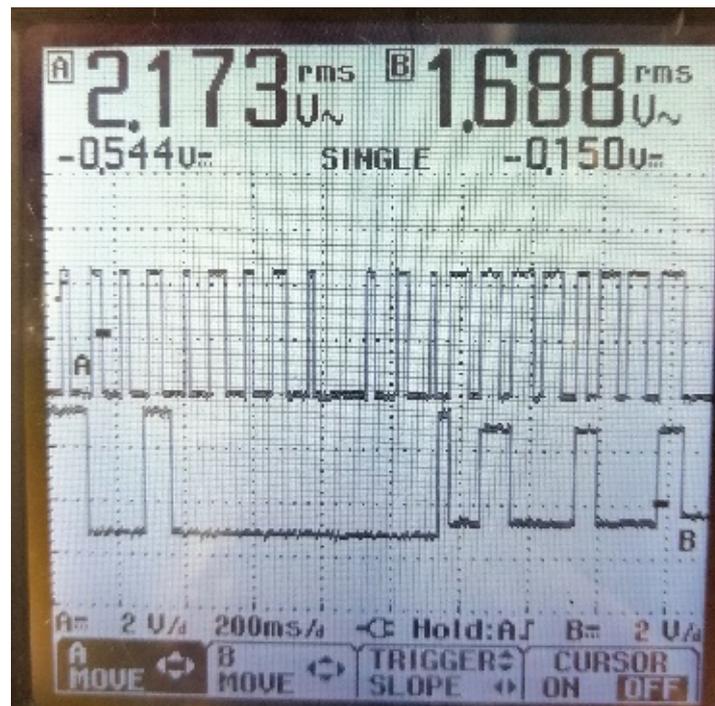


Figure 7: Trama de lectura dirección 00 00 00h.

de otras tecnologías logrando un mayor alcance en la persona que se encuentra en este proceso de aprendizaje.

References

- [1] Ibrahim, D. (2014). A New Approach for Teaching Microcontroller Courses to Undergraduate Students. *Procedia - Social and Behavioral Sciences*, 131, 411–414. <https://doi.org/10.1016/j.sbspro.2014.04.139>
- [2] Maksimović, M., Vujović, V., Davidović, N., Milošević, V., & Perišić, B. (2014). Raspberry Pi as Internet of Things hardware: Performances and Constraints. *Proceedings of 1st International Conference on Electrical, Electronic and Computing Engineering IcETRAN 2014, Vrnjačka Banja, Serbia, June 2 – 5, 2014*, 3(JUNE), 8. Retrieved from <https://www.researchgate.net/publication/280344140>
- [3] Microchip. (2009). Pic18F4550. Pic, 438. Retrieved from <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- [4] Microchip Technology Inc. (2013). PICkit™ 3 In-Circuit Debugger/Programmer User's Guide For MPLAB® X IDE.
- [5] Raguvaran, K., & Thiyagarajan, J. (2015). Raspberry PI based global industrial process monitoring through wireless communication. *Proceedings of 2015 International*

- Conference on Robotics, Automation, Control and Embedded Systems, RACE 2015, (February). <https://doi.org/10.1109/RACE.2015.7097298>
- [6] Technology, M. (2010). PIC18F2XXX / 4XXX FAMILY Flash Microcontroller Programming Specification. Technology, 1–46.
- [7] Ibrahim, D. (2014). A New Approach for Teaching Microcontroller Courses to Undergraduate Students. *Procedia - Social and Behavioral Sciences*, 131, 411–414. <https://doi.org/10.1016/j.sbspro.2014.04.139>
- [8] Maksimović, M., Vujović, V., Davidović, N., Milošević, V., & Perišić, B. (2014). Raspberry Pi as Internet of Things hardware: Performances and Constraints. *Proceedings of 1st International Conference on Electrical, Electronic and Computing Engineering IcETRAN 2014, Vrnjačka Banja, Serbia, June 2 – 5, 2014*, 3(JUNE), 8. Retrieved from <https://www.researchgate.net/publication/280344140>
- [9] Microchip. (2009). Pic18F4550. Pic, 438. Retrieved from <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- [10] Microchip Technology Inc. (2013). PICkit™ 3 In-Circuit Debugger/Programmer User's Guide For MPLAB® X IDE.
- [11] Raguvaran, K., & Thiyagarajan, J. (2015). Raspberry PI based global industrial process monitoring through wireless communication. *Proceedings of 2015 International Conference on Robotics, Automation, Control and Embedded Systems, RACE 2015, (February)*. <https://doi.org/10.1109/RACE.2015.7097298>
- [12] Technology, M. (2010). PIC18F2XXX / 4XXX FAMILY Flash Microcontroller Programming Specification. Technology, 1–46.