

## ANÁLISIS DE UN MODELO DE PROGRAMACIÓN DE ALTO NIVEL PARA LOS MICROCONTROLADORES DE BAJOS RECURSOS



AUTORES: José René Castillo Villamar<sup>1</sup>

DIRECCIÓN PARA CORRESPONDENCIA: ([castillo-jose7772@unesum.edu.ec](mailto:castillo-jose7772@unesum.edu.ec))

Fecha de recepción: 10/01/2021

Fecha de aceptación: 21/09/2021

### RESUMEN

El objetivo de la investigación fue analizar un modelo de programación de alto nivel para los microcontroladores de bajos recursos, que son aquellos lenguajes de programación donde son varias instrucciones se sustituyen por una sentencia, estos lenguajes son fundamental ya que genera código de forma sencilla y fácil de comprender a los programadores, de esta manera se escribe código que es válido para diversas máquinas y sistemas operativos. Permite utilizar paradigmas de programación, y todo esto se puede insertar a los microcontroladores. Los microcontroladores son circuitos impresos programables anidados en muchos objetos de nuestra vida diaria. Debido a sus recursos limitados, a menudo se programan en lenguajes como C o en lenguaje ensamblador. Aunque estos no ofrecen las mismas abstracciones y las mismas garantías que los lenguajes de alto nivel debido a que son microcontroladores de bajos recursos considerados como gama baja. A continuación, este artículo propone un conjunto de soluciones destinadas a enriquecer la programación de microcontroladores con paradigmas de programación. Se utilizó el método científico revisión bibliográfica, permitiendo realizar la investigación. Los resultados demostraron que estas soluciones traen un aumento gradual de la abstracción, en particular, haciendo posible producir programas independientes del equipo utilizado lo que hace posible aprovechar las muchas ventajas del lenguaje manteniendo una pequeña huella de memoria.

PALABRAS CLAVE: instrucciones; código; máquinas; paradigmas; abstracción.

## ANALYSIS OF A HIGH-LEVEL PROGRAMMING MODEL FOR LOW-RESOURCE MICROCONTROLLERS

### ABSTRACT

The objective of the research was to analyze a high-level programming model for low-resource microcontrollers, which are those programming languages where several instructions are replaced by a sentence, these languages are fundamental since it generates code in a simple and easy way

---

<sup>1</sup> Estudiante de Ingeniería en Tecnologías de la Información. Universidad Estatal del Sur de Manabí, Manabí, Ecuador. [castillo-jose7772@unesum.edu.ec](mailto:castillo-jose7772@unesum.edu.ec), Código ORCID: <https://orcid.org/0000-0001-6619-8660>

To understand programmers, in this way code is written that is valid for various machines and operating systems. It allows to use programming paradigms, and all this can be inserted into microcontrollers. Microcontrollers are programmable printed circuits nested in many objects in our daily lives. Due to their limited resources, they are often programmed in languages such as C or assembly language. Although these do not offer the same abstractions and the same guarantees as high-level languages because they are low-resource microcontrollers considered low-end. Next, this article proposes a set of solutions aimed at enriching the programming of microcontrollers with programming paradigms. The scientific method bibliographic review was used, allowing the investigation to be carried out. The results showed that these solutions bring a gradual increase in abstraction, in particular, making it possible to produce programs independent of the equipment used, which makes it possible to take advantage of the many advantages of the language while maintaining a small memory footprint.

**KEYWORDS:** traffic; platforms; infrastructure; threat; security.

## **INTRODUCCIÓN**

Durante los últimos tiempos, la tecnología ha avanzado significativamente donde en la actualidad se interactúa con una cantidad impresionante de microcontroladores en un día típico de la vida cotidiana. Estos dispositivos electrónicos, pequeños tamaños (unos pocos milímetros a unos pocos centímetros), de hecho, anidan en muchos objetos que usamos todos los días y usamos silenciosamente en muchos aspectos de nuestras vidas. Un ejemplo sería una persona tomó el transporte público para llegar a su lugar de trabajo, o más bien a su automóvil personal, Luego, cientos de microcontroladores lo acompañaron en su viaje, con el fin de realizar numerosas tareas automáticas, como comprobar el sistema de frenado del vehículo, su presencia no se limita a los medios de transporte: por ejemplo, en el trabajo, la máquina de café, el sistema de aire acondicionado o incluso los periféricos.

Las computadoras (pantallas, mouse, teclado, etcétera.) contienen potencialmente docenas de ellas, cargadas con los funcionamientos de estos diferentes sistemas, contribuyen así diariamente a las diversas tareas de nuestra vida moderna. Visto desde el interior, un microcontrolador es un circuito impreso programable los diversos elementos de los cuales son equivalentes a los componentes de una computadora muy simplificada, pero sin embargo lleno.

De hecho, un microcontrolador tiene una unidad de cálculo aritmético y lógico, un conjunto de memorias, generalmente compuestas por una memoria de acceso aleatorio que puede procesar datos dinámicos un programa y memorias no volátiles destinadas a almacenar el programa o ciertos datos, así como varios dispositivos de entrada/salida, llamados pines o pines.

De hecho, no es excepcional que los programas destinados a ser ejecutados en microcontroladores continúan escribiéndose directamente en lenguaje ensamblador. Por tanto, la programación del microcontrolador es una actividad difícil, ya que obliga al desarrollador a familiarizarse con el juego de instrucciones, a menudo poco expresivas, del lenguaje ensamblador del microcontrolador de destino. El usuario también debe conocer con precisión el hardware que ejecutará su programa, ya que el nivel la abstracción material que ofrecen estos lenguajes es muy limitada, si no inexistente.

Esta especialización de programas extremos destinados a microcontroladores, además de ser de difícil abordaje para un programador novato más familiarizado con lenguajes de nivel superior

como Python o Java, restringido la capacidad de un programa para ser portado fácilmente a una arquitectura diferente.

## **DESARROLLO**

### **Microcontroladores económicos y de bajo consumo**

(Ronceros, 2020) afirma que el microcontrolador es un circuito integrado que posee un su interior una arquitectura similar a la de un computador, como memorias RAM, EEPROM, CPU y periféricos de entrada y salida o también llamados I/O.

Los microcontroladores que son económicos considerados como bajo consumo y estos tienen incorporados en sus PIC Flash que son de dos particiones que es capaz de actualizar de manera rápida e instantánea, así mismo manejan software que son independientes lo cual permite que la programación insertada sea de forma simultánea en una sola partición donde se está ejecutando el código de la programación en otra aplicación.

Los microcontroladores PIC24F-GB6 considerados de gama baja o de bajo recursos su consumo de corriente está considerada en solo 190  $\mu\text{A}/\text{MHz}$  y 3,2  $\mu\text{A}$ -standby. Todo esto es gracias a la capacidad del firmware en cuanto actualización conocidas como update de manera inalámbrica, las empresas que desarrollan estos microcontroladores de bajo recursos ofrecen estos PIC económicos que son fiable y seguros a la hora de actualizar y codificar la programación en estos mismos.

(Nuñez, 2018) Afirma: Siendo en general una buena CPU, ésta tenía malas prestaciones de entrada y salida, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de entrada/salida a la CPU.

Generalmente en la implementación de los controladores se omiten características importantes de la electrónica y de la dinámica de los procesos.

### **Usos para los microcontroladores económicos de gama baja**

Esta combinación de microcontroladores que utilizan funciones básicas hace los microcontroladores PIC24F-GB6 sean seguros y fácil de adquirir por lo que es ideal utilizarlas para todo tipo de aplicaciones, para el sector industrial, el área de informática, en el área médica, equipos de ejercicios físico y portátiles donde se exijan que estos microcontroladores o dispositivos que lo usen tengan una larga autonomía y duración de la batería ya que su uso va hacer primordial y dependerá de muchos factores o si no se tendría consecuencias, y así mismo tener una buena transferencia, y almacenar datos sin tener que recurrir a memorias extraíbles o memorias externas. También podemos contar que entre estas aplicaciones que utilizan PIC y que siempre se encuentran en contadores como de electricidad, así mismo como el control del clima, dispositivos de reconocimiento y entre otros dispositivos que utilizan Microcontroladores y alguna programación de alto-bajo o medio nivel.

### **Lenguajes de programación**

Existen diferentes lenguajes de programación que se puede utilizar en los microcontroladores entre ellos están los lenguajes de bajo nivel considerados los primeros lenguajes como el lenguaje máquina, el ensamblador es el primer lenguaje que utilizaron los pic y que cualquier microcontrolador lo ejecuta, de ahí tenemos el Lenguaje C que es también un lenguaje que utilizan muchos PIC de bajo recursos o gama baja donde el microcontrolador ejecuta el programa que se carga en la memoria Flash, y a este código que se ha insertado también se la llama código ejecutable y este código se compone por número binarios como 0 y 1.

También esto depende de la arquitectura que tiene el microcontrolador, este código de ceros y uno también llamado código binario se conforma por 12, 14 o 16 bits en anchura. Donde cada palabra que conforma la programación la traduce y la entiende un dispositivo o computador en forma de instrucción que serán ejecutada a medida que el programa esté en funcionamiento en el microcontrolador, entonces todas las instrucciones que se reconoce se ejecutarán colectivamente en un conjunto denominado instrucciones.

(Puente y Perdomo, 2013) aseguran que las implementaciones software tienen la gran ventaja del bajo costo, mayor versatilidad y la posibilidad de ser llevadas a sistemas embebidos como un microcontrolador (MCU), obteniéndose así, en adición, las ventajas de este tipo de sistemas

### ¿Cómo programar un microcontrolador con PIC C?

Para programar un microcontrolador, una de las herramientas que se pueden utilizar es el programa PIC C ya que permite programar todo tipo de microcontrolador en el lenguaje de programación C y la diferencia que tiene entre otros programas que tienen lenguaje máquina o ensamblador en sus librerías, este PIC C es fácil de manejar, codificar, analizar y entender la sintaxis para los PIC. PIC C fue desarrollado por la compañía del mismo nombre PIC CMU, donde se dispone de cientos de miles de librerías o controladores para poder realizar un programa optimizado y poder utilizar dispositivos externos para la programación, tales como pantallas LCD, memorias RAM, ROM, etc. Observaremos ahora en la figura 1 donde encontraremos que está estructurado de la siguiente forma.

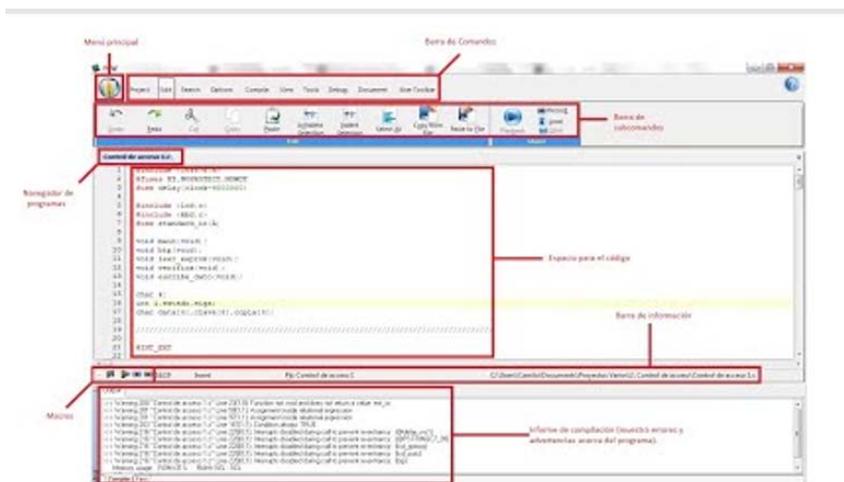


Figura 1. PIC C

El lenguaje C es un lenguaje de programación utilizado en microcontroladores también considerado como uno de los lenguajes más básicos en esta área de microcontroladores, sin embargo, este lenguaje es una gran herramienta para realizar una codificación para microcontroladores ya sean de bajo recursos, gama baja, gama media, etcétera. Este lenguaje consiste en una alternativa para poder realizar un programa en C que ayuda a hacerlo de forma correcta es hacer un diagrama de flujo para entender los pasos que se van a realizar y para tomar decisiones.

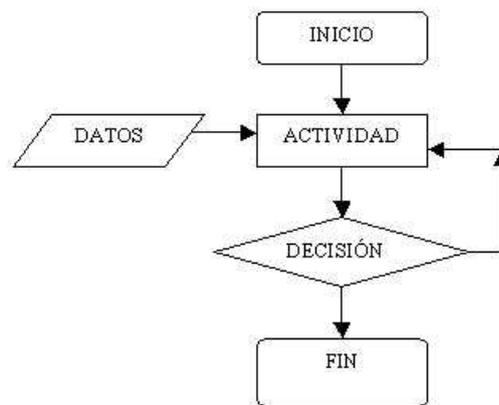


Figura 2. Ejemplo de un diagrama de flujo

MikroBasic PRO para PIC es un compilador BASIC con todas las características para microcontroladores PIC de Microchip. Afirma (Ganchozo, 2018) que está diseñado para desarrollar, construir y depurar aplicaciones embebidas basadas en PIC, el entorno de desarrollo cuenta con una amplia variedad de características tales como: una sintaxis BASIC fácil de aprender, IDE fácil de usar, un código muy compacto y eficiente, muchos equipos y bibliotecas de software, la documentación completa, el simulador de software, un depurador de hardware, la generación de archivos COFF, etc.

### Estructura de un programa en C

Un programa en PIC C se lo puede hacer de distintas maneras en este caso se puede elegir de dos maneras entre ellas tenemos crear un proyecto y la otra es escribiendo el código en extensión .C y esto lo podemos hacer en cualquier editor de código, cualquier forma es válida y va a funcionar, aquí adjunto la estructura de lenguaje c en la figura 3.

```

#include <16f876.h>
#fuses XT,NOVDDT,PUT,NOVVRT
#use delay(clock=4000000)
#use fast_io(B)
int1 cambio=0;
#INT_EXT

ext_isr()
{
    output_toggle(PIN_B7);
}

void main()
{
    set_tris_b(0x01);    //Configura el Port B
    output_low(PIN_B7);
    port_b_pullups(TRUE);
    enable_interrupts(int_ext);
    ext_int_edge(L_TO_H);
    enable_interrupts(GLOBAL);
    while(1)
    {
    }
}

```

Diagrama de estructura de código con líneas de conexión:

- Las líneas de código `#include`, `#fuses`, `#use delay`, `#use fast_io`, `int1 cambio` y `#INT_EXT` están conectadas por una línea horizontal a la etiqueta **Directivas**.
- Las líneas de código `ext_isr()` y su cuerpo están conectadas por una línea horizontal a la etiqueta **Función o funciones**.
- Las líneas de código `void main()` y su cuerpo están conectadas por una línea horizontal a la etiqueta **Función principal**.
- Las líneas de código `set_tris_b(0x01);` y `//Configura el Port B` están conectadas por una línea horizontal a la etiqueta **Instrucciones** y **Comentarios** respectivamente.

Figura 3. Estructura de un PIC en C

## METODOLOGÍA

El presente trabajo tiene un diseño tipo documental, donde se hace uso de métodos teóricos y empíricos, es de tipo documental porque se ha recurrido a lo que son fuentes de manera directa e indirectas (informes técnicos, periódicos, artículos científicos, folletos, ponencias entre otros) donde se recopiló información más relevante y opiniones por diversos investigadores para fundamentar la investigación.

Se utilizaron métodos de investigación científica:

De nivel teórico:

**Análisis – síntesis:** Se utilizó para analizar la programación de alto nivel en C para determinar el funcionamiento en los microcontroladores de bajo recursos.

**Histórico – lógico:** Se usó para la construcción en busca de información técnicas para la elaboración del artículo sobre la programación de alto nivel y los PIC de bajos recursos.

**Inducción–deducción:** Se utilizó aquel método científico que alcanza conclusiones generales partiendo de antecedentes en particular.

**Revisión bibliográfica:** Se utilizó en la recopilación de la información para la elaboración de la investigación sobre la programación en C como está estructurado y el funcionamiento para los microcontroladores de bajo recursos.

## RESULTADOS Y DISCUSIÓN

Como resultados tenemos que se puede programar en microcontroladores de bajos recursos, PIC de gama baja, media y como resultados tenemos el siguiente circuito usando un microcontrolador donde se insertó la programación de C y consiste en encender y apagar un led con PIC.

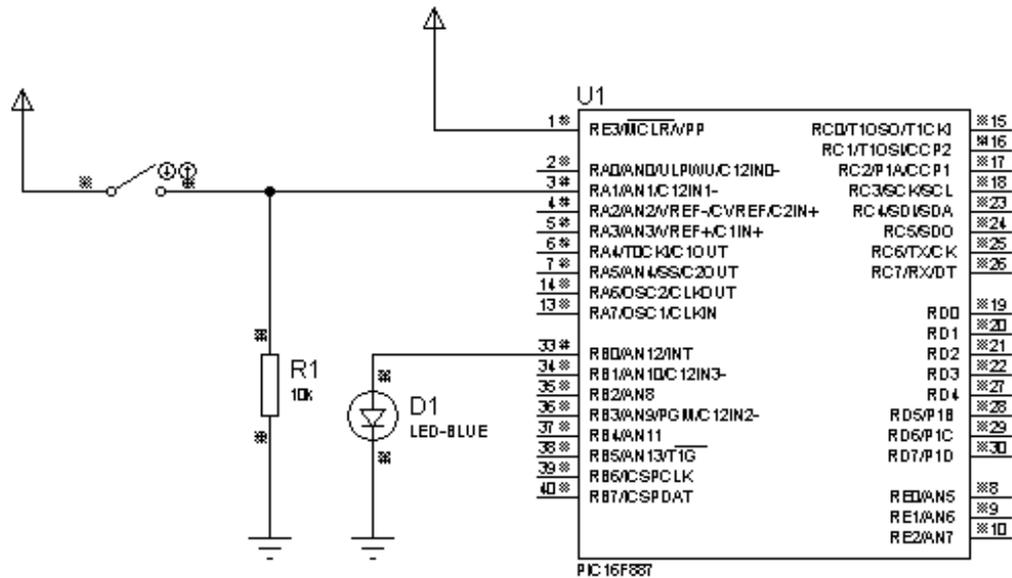


Figura 4. Circuito electrónico con PIC

### Programación del PIC

Encender el LED conectado a RB0 si el switch conectado a RB1 está Activo.

```
#INCLUDE<16f887.h>
#USE DELAY(CLOCK=4000000)
#FUSES XT,PROTECT,NOWDT,NOBROWNOUT,PUT,NOLVP
#DEFINE SW PORTA,1
#DEFINE LED PORTB,0
#BYTE PORTA= 5
#BYTE PORTB= 6
MAIN()
{
    SET_TRIS_A(0B10);
    //Configura el puerto A
    SET_TRIS_B(0B11111110);
    //Configura el puerto B
    WHILE(TRUE)
    // Haga por siempre
```

```

{
  IF(BIT_TEST(SW))
// Si SW esta activado
  {
    BIT_SET(LED);
// Active el led
  }
  ELSE // Sino, es decir si SW esta desactivado
  {
    BIT_CLEAR(LED); // Apagar led
  }
}
}

```

Es un código sencillo, donde se determina la pregunta en instante si SW(PortA,1) se encuentra activo. De caso que sea cierta se activa la salida LED(PortB,0).

### Diseño del circuito electrónico en Proteus Profesional 8 usando el PIC 16F887

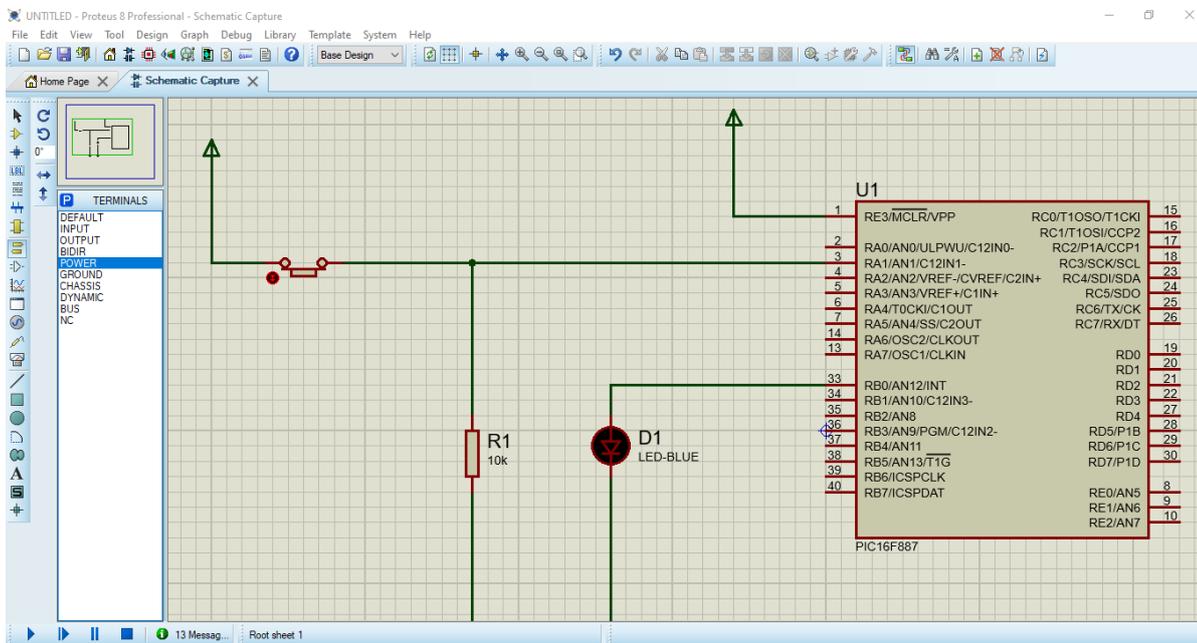


Figura 5. Diseño del circuito en Proteus

En la figura 5 se observa los materiales que se uso para el diseño del circuito electrónico en el Software profesional Proteus donde se usaron resistencias, led color azul, pulsador y el microcontrolador 16f887.

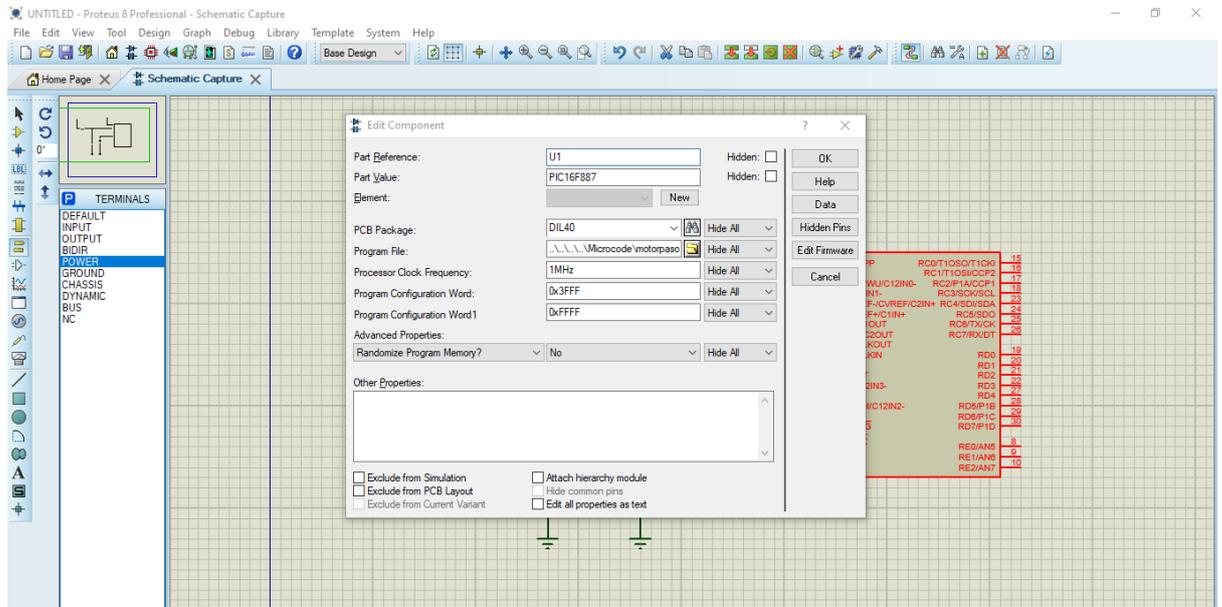


Figura 6. Inserción de la programación al PIC

En la figura 6 se observa las propiedades del PIC para insertar la programación compilada en hex, que fue compilada con el programa de Mikro PIC de Lenguaje c a hexadecimal.

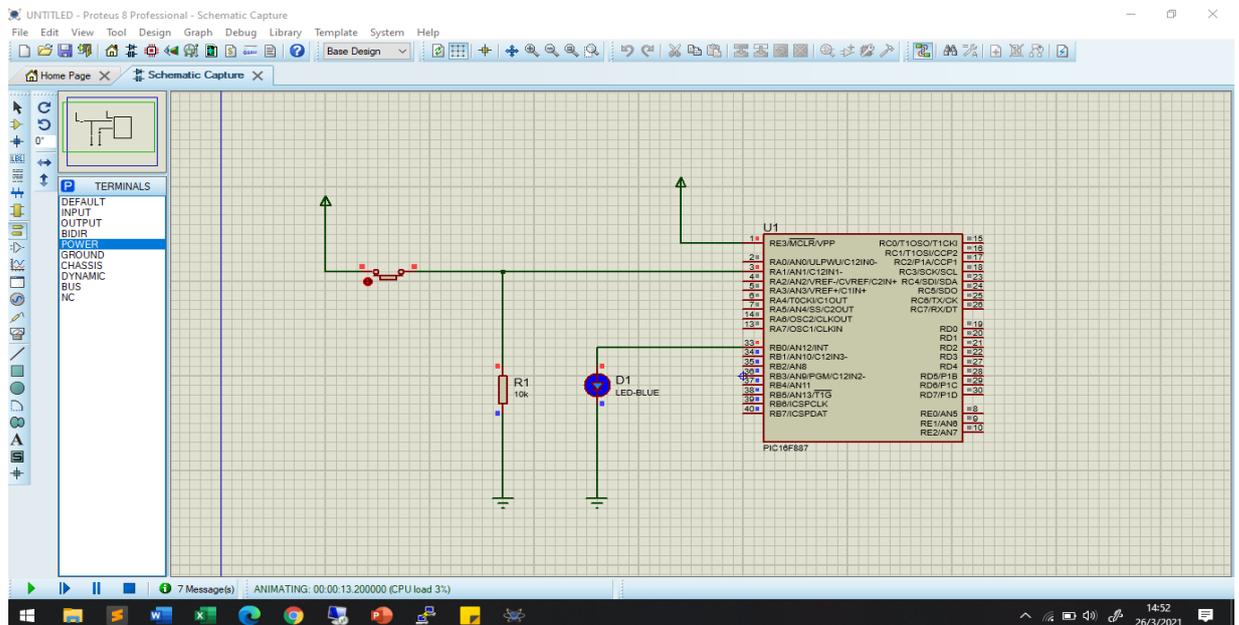


Figura 7. Funcionamiento del circuito

En la figura 7 se observa el respectivo funcionamiento del circuito, ya se diseñó y se unió las conexiones y así mismo se insertó la programación de C a hexadecimal para que el PIC lo pueda interpretar y así mismo se encendió el LED de color Azul.

## CONCLUSIONES

El objetivo de este artículo fue proponer un conjunto de soluciones que permitan la programación de microcontroladores, para demostrar que es bastante posible beneficiarse de las ventajas de modelos de programación más ricos respetando las limitaciones de dicho material.

Se pueden ejecutar programas simples, completo, en dispositivos con recursos muy limitados, así mismo destacar los beneficios de utilizar paradigmas de programación. Por tanto, un microcontrolador cuya RAM es inferior a 2,5 kb es capaz de ejecutar con éxito programas fundamentales para los PIC.

Como conclusión recordamos las diversas aportaciones de este artículo, a la luz del aumento de abstracción que ilustran y las nuevas garantías que brindan los microcontroladores. Discutimos después varias perspectivas y trabajos futuros que serían apropiados para la continuación de nuestro enfoque de mejorar la seguridad y expresividad de la programación.

## REFERENCIAS BIBLIOGRÁFICAS

- Ganchozo, E. Z. (2018). Prototipo de dispositivo tflotecnológico para la identificación de objetos en personas no videntes en PIC. *Revista Científica de Informática ENCRIPAR*. 1(2), 2737-6389.
- Núñez, E. J. (2018). *Diseño de un sistema de seguridad basado en microcontrolador PIC para la central hidroeléctrica en el centro poblado Rapaz*. Universidad Nacional José Faustino Sánchez, Huacho.
- Puente, S. F., & Perdomo, C. A. (2013). Generación automática de Sistemas Lógicos Difusos tipo Mamdani sobre microcontrolador de 8 bits. *Tecnura*, 17(1), 93-108. Obtenido de <https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/7240/8899>
- Roncero, E. D. (2020). Relevancia de la ejecución experimental de proyectos con microcontroladores en el aprendizaje de la ingeniería electrónica. *Educación [online]*, 29(56), 48-72. Obtenido de <http://dx.doi.org/10.18800/educacion.202001.003>