

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Estudio comparativo entre los microcontroladores PIC16F887  
y ATmega328P - pila, entradas y salidas, temporizador,  
interrupciones y memoria EEPROM**

Protocolo de trabajo de graduación presentado por Julio Javier  
Schwendener Morales, estudiante de Ingeniería Mecatrónica

Guatemala,

2022

## Resumen

El estudio a realizar consiste en la comparación de los microcontroladores PIC16F887 y ATmega328P en los módulos de pila, entradas y salidas, temporizador, interrupciones y memoria EEPROM, con el fin de determinar, por medio de analizadores lógicos, cuál es más eficiente para la enseñanza de microcontroladores en la Universidad del Valle de Guatemala.

La eficiencia se medirá, tomando como base, los estudios realizados por Limor Fried y la empresa Microchip sobre comparación de microcontroladores de 8 bits, usando como guía los laboratorios propuestos en el curso Programación de Microcontroladores IE2009, de los cuales se busca realizar la programación, tanto en ensamblador como en lenguaje C, para evaluar los módulos antes descritos de ambos microcontroladores.

## Antecedentes

El PIC16F887 es el microcontrolador empleado en la enseñanza de la programación básica de microcontroladores, basado en el lenguaje de Assembler y C, en la Universidad del Valle de Guatemala [1]. El ATmega328P es el microcontrolador empleado por la placa de desarrollo Arduino UNO, facilitando la programación del microcontrolador con el software de Arduino. En algunos modelos de Arduino UNO, este microcontrolador puede ser desmontado y programado de forma independiente, tanto en Assembler como en C, funcionando como alternativa a la programación en el software de Arduino [2].

A continuación, se describen trabajos de comparación de microcontroladores similares.

## Comparación de microcontroladores de 8 bits

Microchip es una empresa estadounidense dedicada a la manufactura de microcontroladores. Es la encargada del desarrollo y fabricación, entre otros, de los microcontroladores de la línea PIC16.

En 2002, Microchip realizó un trabajo de investigación, comparando distintos microcontroladores de 8 bits, tales como Intel 8051 y Motorola MC68HC05, con el fin de probar la superioridad de los microcontroladores de la línea PIC16C5. En la investigación realizada, se especifican las comparaciones con un enfoque técnico. Entre los aspectos comparados, Microchip realiza pruebas sobre el control de *loop*, el cual consiste en el tiempo de respuesta de los microcontroladores estudiados frente a un *loop* de un contador decreciente que, cuando llega a cero, se reinicia el contador. Para este experimento, el PIC de la línea 16C5 obtuvo el menor tiempo de todos los microcontroladores, entre 0.4 y 0.6  $\mu$ s [3].

Como parte de las pruebas realizadas, se registró el tiempo de ejecución del módulo de temporizadores con un *delay* de 10  $\mu$ s en cada uno de los microcontroladores estudiados. Para este experimento, se obtuvo que los PIC16C5 terminan la ejecución en un aproximado de 10.011 ms. Sin ser este el tiempo más bajo, Microchip indica que esto se debe a la precisión con la que los PIC16C5 pueden trabajar al momento de implementar los temporizadores en otras aplicaciones, a diferencia del resto de microcontroladores.

## Comparación entre las familias PIC y AVR

Fried [4], realizó un trabajo de investigación, publicado en 2012, comparando distintos aspectos sobre los microcontroladores de Microchip y Atmel, específicamente las familias PIC de Microchip y AVR de Atmel. Los modelos comparados en esta investigación fueron PIC12F629, PIC16F628, PIC18F452, ATtiny13, ATtiny2313 y ATmega32.

El trabajo presentado por Fried intenta emparejar los tres microcontroladores de la familia PIC con los de Atmel tomando en cuenta los pines que estos poseen. La investigación se enfoca en beneficio del usuario, por lo que los aspectos a comparar fueron, mayoritariamente, en torno a la facilidad de adquisición y programación de los microcontroladores

Fried comienza exponiendo los precios de compra de los distintos microcontroladores y, tomando en cuenta el emparejamiento inicial, indica que los costos son similares y no declara una ventaja en este aspecto para ninguna de las familias. El siguiente aspecto comparado son los lenguajes de programación. Fried, compara la programación tanto en lenguaje C como en *Assembler*. En el entorno de *Assembler*, Fried indica que los microcontroladores de Atmel, al no tener bancos de memoria y utilizar un formato más amigable con los usuarios, tienen una clara ventaja en este aspecto, mientras que en los entornos de lenguaje C, atribuye la ventaja a Atmel debido a las optimizaciones que permite el compilador, además de la excelente compatibilidad entre compiladores, a diferencia de los microcontroladores PIC, los cuales no son 100 % compatibles con otros compiladores.

## Microcontroladores en otras universidades

En otras universidades con cursos de electrónica y programación de microcontroladores, se pueden indentificar los distintos microcontroladores utilizados en estas tareas. En la India, en Muzaffarpur Institute of Technology [5], para el curso de microcontroladores, utilizan microcontroladores de la familia PIC y el microcontrolador Intel 8051. Este curso abarca la arquitectura de los microcontroladores, programación en entorno *Assembler*, manejo de temporizadores e iterrupciones, y comunicaciones seriales.

En Europa, universidades como la Universidad de Génova (UniGe) [6], utilizan los microcontroladores ARM Cortex-M4. En el curso de programación de microcontroladores se enfocan en el aprendizaje en el lenguaje C, manejo de procesos en paralelo, tipos de datos, arquitectura y las instrucciones del microcontrolador, comunicaciones seriales y temporizadores.

En Estados Unidos, la universidad Northwestern [7], en Illinois, Chicago utilizan microcontroladores proporcionados por Texas Instruments, específicamente el MSP430FR5994. Su aprendizaje de programación de microcontroladores comienza con el estudio de la arquitectura del microcontrolador. La programación abarca el manejo de temporizadores, comunicaciones seriales, conversiones analógico a digital e implementación de PMWs.

Finalmente, el alcance del curso de programación de microcontroladores de la Universidad del Valle de Guatemala abarca el aprendizaje en entorno ensamblador y abarca el manejo de los bancos de memoria, temporizadores, lectura de entradas digitales y analógicas, comunicaciones seriales, y la implementación de procesos en paralelo.

## Justificación

El uso de microcontroladores es una parte fundamental de la electrónica, ya que su uso está implícito en una gran variedad de dispositivos electrónicos. La programación de los microcontroladores es, entonces, un campo de gran interés para las ingenierías Electrónica, Mecatrónica y Biomédica.

Actualmente, en la Universidad del Valle de Guatemala [1], la programación de microcontroladores se enseña con el PIC16F887, de Microchip, comenzando con en el entorno de ensamblador para, posteriormente, pasar al lenguaje C en cursos más avanzados. Esta tarea abarca el aprendizaje del funcionamiento básico del microcontrolador: configuración de los pines, tanto entradas como salidas; entradas y salidas analógicas; temporizadores; interrupciones; y comunicaciones seriales.

Por este trabajo, se pretende evaluar la eficiencia del microcontrolador utilizado en la Universidad del Valle de Guatemala (PIC16F887) y el ATmega328P con el fin de determinar, bajo el contexto de Guatemala, cuál de los dos es una mejor opción para el aprendizaje de programación de microcontroladores.

Es importante mencionar que, a futuro, los resultados de este trabajo podrían influenciar en la forma de enseñanza de los cursos de programación de microcontroladores en la Universidad del Valle de Guatemala, lo cual significaría mantener una continuidad respecto a los lenguajes de programación y plataformas empleadas durante el transcurso de las carreras de Ingeniería Electrónica, Ingeniería Mecatrónica e Ingeniería en Biomédica.

## Objetivos

### Objetivo General

Realizar un estudio comparativo entre los microcontroladores PIC16F887 y ATmega328P, de los módulos de pila, entradas y salidas, temporizador, interrupciones y memoria EEPROM, y evaluar los parámetros necesarios para determinar la mejor opción para la enseñanza de microcontroladores en la Universidad del Valle de Guatemala.

### Objetivos Específicos

- Comparar el desempeño de cada microcontrolador con el uso de los analizadores lógicos disponibles en la Universidad del Valle de Guatemala, evaluando los módulos de pila, entradas y salidas, temporizador, interrupciones y memoria EEPROM.
- Realizar los laboratorios y proyectos propuestos en el curso de Programación de Microcontroladores en ambas plataformas relacionados a los módulos a evaluar.

## Marco teórico

### Generalidades de un microcontrolador

Según Bettina [8], se puede resumir un microcontrolador como un procesador reducido, que posee memoria, temporizadores, pines de entrada y salida (*I/O pins*), y otros periféricos; todas estas características integradas en un espacio pequeño, optimizando costos de manufactura y tiempos de desarrollo.

Los microcontroladores son empleados en diversas aplicaciones de control, monitoreo y procesamiento de sistemas. También son empleados en dispositivos de uso común, como teléfonos celulares, máquinas de lavado, microondas y vehículos [9].

Generalmente, los microcontroladores incluyen un procesador central, puertos de entrada y salida, memoria para programar y almacenamiento de datos, un reloj interno (en inglés *internal clock*) y una gran variedad de periféricos [9]. Entre los periféricos más comunes se encuentran los temporizadores (*timers*), contadores, conversiones analógico a digital (por sus siglas en inglés *ADC*) y comunicaciones seriales [8].

### Pila

La pila (más conocida como *stack*), es un área de la memoria RAM de los microcontroladores que permite el almacenamiento de los parámetros de variables, también es la responsable de recordar el orden en que las funciones son llamadas para poder regresar de forma correcta [9].

### Entradas y salidas

Estas son todos los pines con capacidad de recibir o emitir un dato digital, es decir un voltaje *HIGH* o *LOW*, entre el rango definido. Estos voltajes son traducidos por el microcontrolador como un valor binario independientemente del valor del voltaje que se esté recibiendo, tomando en cuenta las especificaciones del fabricante de mínimos y máximos admitidos y las zonas en las que se interpretan como 1 o 0 lógicos [9].

### Interrupciones

Una interrupción es una señal asíncrona, que funciona como mecanismo para evitar pérdidas del tiempo de procesador de los microcontroladores [9]. Las interrupciones pueden ser activadas por tanto por banderas internas del microcontrolador o por señales externas de algún origen previamente especificado. Se pueden resumir como instrucciones que serán realizadas en el momento exacto en que alguna señal es recibida por el microcontrolador.

## Temporizadores

Los temporizadores (conocidos en inglés como *timers*), son contadores, los cuales, al alcanzar un número previamente determinado, activan interrupciones en el programa. Pueden tener funciones extras como la captura y comparación de datos en tiempo, así como pre-escalados [9].

## Memoria EEPROM

Por sus siglas en inglés *Electrically-Erasable Programmable Read-Only Memory*, es utilizada como un almacenamiento no volátil, diferenciándose de la memoria RAM y permitiendo el almacenamiento de datos aún cuando la energía es removida [9].

## PIC16F887

El PIC16F887 [10], es un microcontrolador de 8 bits y 40 pines desarrollado por Microchip. Posee 14 canales de conversiones analógico a digital de 10 bits, tiene una memoria EEPROM programable de 256 bytes, 2 salidas PWM, comunicaciones SPI, UART e I2C, 3 temporizadores, comparadores, 35 pines de entrada y salida de propósito general y opera a una frecuencia máxima de 20MHz. Los microcontroladores PIC son generalmente utilizados en una gran variedad de sistemas embebidos.

## ATmega328P

El ATmega328P [11], es un microcontrolador de 8 bits y 28 pines, diseñado originalmente por la extinta Atmel y fabricado actualmente por Microchip. Posee de 6 a 8 canales de conversiones analógico a digital de 10 bits, pines de entrada y salida de propósito general, comunicaciones I2C, SPI y UART, y 3 temporizadores con modos de comparación. Los ATmega328P son conocidos por ser los microcontroladores empleados en la plataforma Arduino debido a su alto rendimiento y bajo consumo [12].

## Lenguajes de programación

La programación es el método utilizado para hacer llegar las instrucciones deseadas al microcontrolador. Generalmente, los microcontroladores son programados en lenguajes de alto nivel, como C++ o Java [13]. Sin embargo, los microcontroladores entienden únicamente el código de máquinas, por lo que es necesario traducir el código creado en un lenguaje de alto nivel, al lenguaje de máquinas que el microcontrolador puede entender y ejecutar [8].

## Compiladores

Una vez el código está completo, este debe ser cargado en el microcontrolador. La función de un compilador es lograr esta tarea, traduciendo el lenguaje de programación de alto nivel a lenguaje de máquinas para poder cargarlo. Un compilador es, pues, una herramienta de software que toma un código de un nivel más alto y lo optimiza para ensamblador (conocido también como *Assembler*) [13].

## Ensamblador

Escribir un código directamente en lenguaje binario puede ser complicado e ineficiente. Para solucionar este problema, se utiliza el lenguaje ensamblador, el cual asigna nombres a cada uno de los comandos que posee el microcontrolador, permitiendo a los usuarios memorizar palabras en lugar de secuencias binarias [8]. Sin embargo, estos comandos no son comprendidos por los microcontroladores, por lo que es necesario traducir los comandos a binario por medio de un compilador.

## Metodología

La experimentación consistirá en realizar las prácticas de laboratorio del curso de programación de microcontroladores de la Universidad del Valle de Guatemala [1] y comparar el funcionamiento de los módulos de pila, entradas y salidas, temporizador, interrupciones y memoria EEPROM de los microcontroladores por medio de los analizadores lógicos. A continuación se muestra la lista de actividades a realizar.

- Lectura de hoja de datos de PIC16F887.
- Lectura de hoja de datos de ATmega328P.
- Lectura de hoja de datos de analizador lógico.
- Instalación de compiladores para ambos microcontroladores.
- Elaboración de esquemáticos de los circuitos a trabajar de los módulos de pila, entradas y salidas, temporizador, interrupciones y memoria EEPROM, según se indique en las prácticas de laboratorios propuestas en el curso de programación de microcontroladores de la Universidad del Valle de Guatemala.
- Programación en entorno ensamblador para ambos microcontroladores.
- Programación en C para ambos microcontroladores.
- Comparación de módulos de ambos microcontroladores programados en C y en ensamblador por medio de analizadores lógicos.

Con el fin de comparar ambos microcontroladores, se estudiará su desempeño mediante el uso de analizadores lógicos, los cuales permitirán obtener una medición exacta sobre el funcionamiento de cada uno.

Durante el transcurso de las tareas, se manejará un repositorio de Git con el fin de mantener un orden claro y registro de versiones de los códigos que se realizarán. Los resultados obtenidos serán incorporados en el trabajo final en paralelo a la elaboración de los códigos de programación.

Cronograma de actividades

Número de actividad	Actividades a realizar	Semana														CHECK LIST
		4-Jul	11-Jul	18-Jul	25-Jul	1-Aug	8-Aug	15-Aug	22-Aug	29-Aug	5-Sep	12-Sep	19-Sep	25-Sep		
1	Lectura del funcionamiento de los logic analyzer															
2	Pruebas del funcionamiento de logic analyzer															
3	Instalación de los compiladores															
4	Lectura de los datasheet de los microcontroladores															
5	Repaso de laboratorios y proyectos del curso de programación de microcontroladores de la UVG															
6	Adquisición de materiales de trabajo															
7	Elaboración de esquemáticos de las pruebas planteadas															
8	Programación de prueba para el PIC16F887 (assembly)															
9	Armado de esquemáticos en protoboard															
10	Programación de entradas y salidas de PIC16F887															
11	Programación de entradas y salidas de Atmega328P															
12	Programación de fila de PIC16F887															
13	Programación de fila de Atmega328P															
14	Programación de interrupciones PIC16F887															
15	Programación de interrupciones Atmega328P															
16	Programación de temporizadores PIC16F887															
17	Programación de temporizadores Atmega328P															
18	Programación de la memoria EEPROM PIC16F887															
19	Programación de la memoria EEPROM Atmega328P															
20	Programación del proyecto del reloj PIC16F887															
21	Programación del proyecto del reloj Atmega328P															
22	Medición de respuestas para tareas básicas con logic analyzer															
23	Escritura del trabajo final															
24	Manejo de git para los programas realizados															

Figura 1: Cronograma



# Índice preliminar

Prefacio

Resumen

Abstract

1. Introducción

2. Antecedentes

3. Justificación

4. Objetivos

*a)* Objetivo general

*b)* Objetivos específicos

5. Alcance

6. Marco teórico

7. PicKit4 y MPLABX

8. Módulo de entradas y salidas

*a)* PIC16F887

*b)* ATmega328P

9. Módulo pila

*a)* PIC16F887

*b)* ATmega328P

10. Módulo temporizador

*a)* PIC16F887

*b)* ATmega328P

11. Módulo de interrupciones

*a)* PIC16F887

*b)* ATmega328P

12. Programación de la memoria EEPROM

*a)* PIC16F887

*b)* ATmega328P

13. Referencias

14. Anexos

## Referencias

- [1] U. del Valle de Guatemala, *Programación de microcontroladores*, 2019.
- [2] Microchip, “mega AVR Data Sheet,” 2020.
- [3] —, “A comparison of 8-Bit Microcontrollers,” 1997.
- [4] L. Fried. “PIC vs. AVR: Ultimate fight!” (2012), dirección: <http://www.ladyada.net/library/picvsavr.html>.
- [5] S. Kumar, *Course File Microcontrollers*, 2016. dirección: <https://www.mitmuzaffarpur.org/wp-content/uploads/2018/08/MICROCONTROLLER.pdf>.
- [6] F. Ansovini, *Microcontroller Programming - Syllabus*. dirección: <http://www.phdstiet.diten.unige.it/images/Documents/MicrocontrollerProgramming.pdf>.
- [7] N. University. “Microprocessor System Design.” (2019), dirección: <https://canvas.northwestern.edu/courses/92393>.
- [8] G. Bettina. “Introduction to Microcontrollers.” (2007), dirección: <https://ti.tuwien.ac.at/ecs/teaching/courses/mclu/theory-material/Microcontroller.pdf>.
- [9] J. Sanchez y M. Canton, *Microcontroller Programming - The Microchip PIC*. CRC Press, 2007.
- [10] Microchip. “PIC16F887.” (2022), dirección: <https://www.microchip.com/en-us/product/PIC16F887>.
- [11] —, “ATmega328P.” (2022), dirección: <https://www.microchip.com/en-us/product/ATmega328P>.
- [12] Arduino, *UNO R3*. dirección: <https://docs.arduino.cc/hardware/uno-rev3> (visitado 28-04-2022).
- [13] J. Foxworth, *How to program a microcontroller*. dirección: [https://www.egr.msu.edu/classes/ece480/capstone/spring15/group13/assets/app\\_note\\_john\\_foxworth.docx.pdf](https://www.egr.msu.edu/classes/ece480/capstone/spring15/group13/assets/app_note_john_foxworth.docx.pdf) (visitado 05-05-2022).