



**Tecnológico  
de Monterrey**

## **Engineering Manual – Electrónica**

### **Selección de Microcontrolador**

Diseño e implementación de sistemas mecatrónicos MR3002B

Grupo 501

Carlos Daniel López Montero - A01024473

2 de Mayo 2023

Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Monterrey

## Selección de microcontrolador a usar

### Establecimiento de objetivo

A lo largo de esta entrada al *Engineering Manual* en la sección de electrónica se establece cumplir con la siguiente meta, la cual fue establecida en el documento *Actividad 2.1. Definición de objetivos, justificación y resultados esperados*.

**Meta 1.3.** Tomar los resultados de las metas 1.1/1.2 y realizar una investigación y selección preliminar del microcontrolador/SoC a usar en el proyecto.

### Resultados esperados

- a) Selección del microcontrolador a usar para la parte de control del mecanismo.
- b) Preselección del microcontrolador a usar para la interfaz de control inalámbrica.

### Selección de componentes

Cabe decir que, aunque la fuente de poder ya fue seleccionada, la selección del microcontrolador no debería afectarnos debido a que al momento de dimensionar la capacidad de la batería se tomó como consideración un consumo máximo por parte del microcontrolador de 370mA con flexibilidad de que el consumo sea de hasta 555mA.

A continuación, se establecieron los siguientes requerimientos para seleccionar los microcontroladores.

- a) Debe tener capacidad de comunicarse inalámbricamente, ya sea por Bluetooth o WiFi, con la aplicación de control en un teléfono; en el caso del desarrollo de la interfaz de control inalámbrica se determinó que los microcontroladores sean fáciles de intercomunicar y proveer un sistema de conexión que no requiera intervención por parte del operario, así como proveer cierta protección/criptación al comunicarse entre ellos.
- b) De ser posible, y debido a que se planea diseñar una interfaz de control digital, el microcontrolador deberá poder programarse en MicroPython/Python.
- c) Por la selección de los componentes de control del motor (TB6600/A4988/DRV8825) y los sensores propuestos (Switch fin de carrera y Encoders) se requiere que los microcontroladores tengan:
  - 1. 6 pines digitales, de los cuales 2 necesitan tener funcionalidad PWM para las señales de paso del motor.
  - 2. Una a dos interfaces de comunicación I2C para realizar la comunicación con los encoders seleccionados (encoder magnético AS5600).
  - 3. 2 pines digitales para conectar los sensores de fin de carrera.
  - 4. Se estima que se requieren 8 pines digitales para poder implementar una interfaz local en el sistema.
  - 5. En caso de realizar la interfaz inalámbrica, este microcontrolador deberá contar con una interfaz SPI (en caso de emplear una pantalla táctil) o los 8+ pines estimados para la interfaz analógica/digital.

Retomando la lluvia de ideas en la entrada anterior, se consideraron los siguientes microcontroladores. A esta lista se agrega un posible microcontrolador.

- 1. Raspberry Pi Zero W.
- 2. Teensy 4.1.
- 3. Arduino Uno.
- 4. Arduino Mega.
- 5. ESP32.

Una vez conocidos los microcontroladores y los requerimientos, se procede a ver sus características en la siguiente tabla.






Microcontrolador	Voltaje	Lenguaje de programación	Pines digitales/analógicos	Interfaces de comunicación	Link
 Raspberry Pi Zero W	3.3V a 5V GPIO @ 3.3V	C/C++, Python 2/3, Scratch, entre otros.	4 pines dedicados a PWM (de los 28 totales) 28 pines GPIO	2 interfaces SPI, 2 interfaces I2C y 1 interfaz UART 802.11 b/g/n wireless LAN Bluetooth 4.1 Bluetooth Low Energy (BLE)	<a href="https://www.etec hnophiles.com/r aspberry-pi-zero-gpio-pinout-specifications-programming-language/">https://www.etec hnophiles.com/r aspberry-pi-zero-gpio-pinout-specifications-programming-language/</a>
 Teensy 4.1	3.6V a 5.5V GPIO @ 3.3V	C/C++, Rust, LabView, MicroPython, entre otros.	35 pines PWM (de los 55 totales) 55 pines GPIO 18 entradas analógicas	8 interfaces UART, 3 SPI y 3 I2C. 3 CAN BUS, Ethernet y FlexIO	<a href="https://www.pjrc.com/store/teensy41.html">https://www.pjrc.com/store/teensy41.html</a>
 Arduino Uno	5V a 20V GPIO @ 5V	C++	14 pines digitales 6 pines PWM (de los 14 totales) 6 entradas analógicas	1 UART (opción a más por software) 1 SPI (opción a más por software) 1 I2C (opción a más por software)	<a href="https://store.arduino.cc/products/arduino-uno-rev3">https://store.arduino.cc/products/arduino-uno-rev3</a>
 Arduino Mega 2560	5V a 20V GPIO @ 5V	C++	54 pines digitales 15 pines PWM (de los 54 totales) 16 entradas analógicas	4 UART (opción a más por software) 1 SPI (opción a más por software) 1 I2C (opción a más por software)	<a href="https://store.arduino.cc/products/arduino-mega-2560-rev3?queryID=undefined">https://store.arduino.cc/products/arduino-mega-2560-rev3?queryID=undefined</a>
 ESP32 devkit v1	7 a 12V GPIO @ 3.3V	C/C++, Python, MicroPython, Lua, JavaScript, entre otros.	25 pines GPIO de los cuales 15 se pueden usar como ADC, 25 como PWM, 2 canales DAC y 9 inputs de tacto; 4 son solo del tipo input	3 UART, 1 I2C y 3 SPI. WiFi 802.11 b/g/n/e/i Bluetooth 4.2 BR/EDR y BLE	<a href="https://u-electronics.com/producto/devkitv1-esp32-modulo-wifi-bluetooth-esp32-arduino/">https://u-electronics.com/producto/devkitv1-esp32-modulo-wifi-bluetooth-esp32-arduino/</a>

Tabla 7. Especificaciones técnicas de los microcontroladores propuestos.

Comparando la tabla con los requerimientos técnicos que se establecieron anteriormente se podría usar cualquier microcontrolador; sin embargo, en el caso del 2do, 3er y 4to microcontrolador se tendría que emplear módulos diferentes para añadir comunicación WiFi y Bluetooth. Esto no solo aumenta el costo en la fase de selección de componentes, sino que también agrega más puntos de falla en el sistema eléctrico y dichos módulos de comunicación normalmente terminan utilizando alguna interfaz de comunicación (como SPI o I2C), dejando sin posibilidad de usar de forma confiable estas para realizar la conexión de los encoders.

Adicionalmente, en el caso del ESP32 devkit v1, se cuenta con un protocolo de comunicación inalámbrica basada en WiFi y Bluetooth BLE llamado ESP-NOW. Según Espressif (s.f.), su protocolo de comunicación propietaria permite interconectar su familia de microcontroladores de una manera sencilla (no es necesario tener un router, solo registrar un dispositivo en la red), a prueba de errores (si un nodo pierde conexión, este mismo tratará de volver a conectarse a la red por su cuenta), flexible (no importa el tipo de microcontroladores o su interfaz de comunicación inalámbrica; adicionalmente permite configurar comunicaciones one-to-many, many-to-one y de dos vías) de bajo consumo energético y computacional, así como proveer soporte para encriptar los mensajes en caso de ser necesario.

### Presentación de resultados

#### a) Selección del microcontrolador a usar para la parte de control del mecanismo.

Se llegó a la conclusión de que el microcontrolador a usar será el ESP32 devkit V1 debido a su versatilidad a la hora de configurar el GPIO, logrando así configurar sus pines para actuar como I/O o como interfaz de comunicación. Adicionalmente, se eligió este módulo debido a que ya tiene integrada la conectividad inalámbrica que necesitamos, así como nos presenta una manera sencilla y confiable de intercomunicar varios ESP32 sin mayor interacción por parte del usuario. Por último, se decidió usar un microcontrolador en lugar de un SoC (como el Raspberry Pi Zero W) debido a que no se requiere gran capacidad de procesamiento o tener un SO; esto también nos permite reducir los costos y proveer la capacidad de sustituir el microcontrolador con mayor facilidad (tanto en instalación como en la adquisición del repuesto) sin la

intervención del equipo, es decir, se plantea que Multitaskr pueda realizar el remplazo en caso de ser necesario.

b) Preselección del microcontrolador a usar para la interfaz de control inalámbrica.

Como se mencionó en el apartado anterior, el ESP32 devkit v1 provee un protocolo de comunicación propietario llamado ESP-NOW. Este protocolo facilita la comunicación entre varios microcontroladores permitiendo a los mismos la reconexión en caso de ser necesario y también permite la encriptación de la comunicación. Se plantea que se usará este microcontrolador en caso de hacer una interfaz de control inalámbrica.