

TP DE LABORATORIO 1: MICROCONTROLADORES Y NETWORKING

El trabajo práctico constará de dos partes:

- 1) El grupo deberá diseñar un circuito original junto con el programa necesario para manejar dicho circuito. Para esto, el grupo contará con al menos un sensor y un actuador, provistos por la cátedra, más el microcontrolador necesario. Cada grupo podrá constar de hasta 5 alumnos, dependiendo de la disponibilidad de microcontroladores en la cátedra.
- 2) El curso entero deberá diseñar e implementar un mecanismo de comunicación entre controladores para centralizar las lecturas de los sensores y acciones en actuadores tomados por los mismos.

Los lineamientos para realizar el trabajo son los siguientes:

- 1) Parte 1:
 - a. El grupo deberá crear un escenario ficticio en el que exista una necesidad resoluble mediante mecanismos de control automático. En este escenario, los sensores y actuadores que el grupo tenga a disposición representarán el equipamiento necesario para solucionar el problema propuesto. Al final de este documento encontrarán un ejemplo de escenario posible. El escenario propuesto no debe presentar grandes similitudes con el ejemplo presentado aquí.
 - b. La solución debe utilizar el microcontrolador que el grupo tenga disponible, y todos los sensores y actuadores disponibles.
 - c. La solución puede estar desarrollada en el lenguaje de programación que se utilizó en el laboratorio de set up del entorno de desarrollo (CircuitPython para Raspberry Pi Pico W) o bien en otro lenguaje que el grupo crea conveniente y conozca (siempre que sea compatible con el microcontrolador). Es necesario que el lenguaje utilizado sea estándar, y sea código textual (es decir, no están permitidos lenguajes visuales o de bloques como Scratch).
 - d. En la solución, cada sensor y actuador debe formar parte de, al menos, un lazo de control. Por ejemplo, de tener dos sensores y un actuador, habrá dos lazos de control (uno por cada sensor, controlando el mismo actuador); de tener dos sensores y dos actuadores, deberán existir al menos dos lazos de control (uno por cada par); etc.
En caso de contar con más sensores que actuadores, el programa deberá implementar alguna manera de activar y desactivar lazos de control para alternar qué sensor se utiliza en un momento arbitrario.
 - e. La solución debe contar con al menos dos de
 - i. Monitoreo por puerto serial de valores sensados y/o decisiones tomadas (es decir, acciones que el microcontrolador tome con respecto a los actuadores).
 - ii. Monitoreo por puerto serial de valores numéricos visibles mediante la función `plotter` de la IDE de Arduino (para esto no es necesario desarrollar el programa en la IDE de Arduino, sino simplemente utilizarla para interactuar con el microcontrolador mientras está en funcionamiento) o algún otro mecanismo que el grupo decida factible de implementar.

- iii. Control de funciones del microcontrolador mediante mensajes por puerto serial. Esto podría ser utilizado por ejemplo para cambiar de lazo de control activo.
 - iv. Control o monitoreo mediante el protocolo http.
- 2) Se formará un grupo extra que se encargará de implementar una red entre los microcontroladores de los grupos del curso, siguiendo los lineamientos que se describen a continuación:
- a. El grupo extra deberá estar conformado por un delegado de cada grupo.
 - b. El grupo extra deberá diseñar un protocolo de comunicación mediante redes de área local Wi-Fi (puede ser http, sockets, etc) que sirva para comunicar los controladores de cada grupo con un controlador maestro.
 - c. El controlador maestro deberá reunir periódicamente los datos de todos los controladores de la red.
 - d. Mediante algún mecanismo definido por el grupo extra, se deberá poder monitorear en tiempo real los valores reunidos por el controlador maestro mediante un teléfono celular o PC (por ejemplo, usando NodeRed o un servidor web accesible en la red local).
 - e. El método de monitoreo deberá mostrar los valores de alguna manera que tenga sentido para los mismos. Por ejemplo, si es un valor entero del que nos interese conocer su historial, se lo mostrará en un plot valor/tiempo. Si es un valor binario del cual no queremos saber el historial, se lo puede mostrar como una palabra o un indicador instantáneo, sin necesidad de mostrarlo como un gráfico.
 - f. Se deberá implementar un mecanismo automático de descubrimiento de controladores de manera que el controlador maestro no deba saber a priori direcciones IP de los controladores de cada grupo.
 - g. Los integrantes del grupo extra no están eximidos de trabajar con sus respectivos grupos en el desarrollo del punto 1.

La fecha de entrega del trabajo es el lunes 06/10/2025 a las 23:59.

La entrega constará, por grupo, de:

1. Una descripción detallada del escenario y solución planteadas en pdf, con carátula detallando
 - a. Materia
 - b. Profesores de la cátedra
 - c. Integrantes del grupo
2. Un diagrama detallado del circuito que implementa la solución planteada, como imagen exportada (es decir, no es válido presentar, por ejemplo, un diagrama DIA o VISIO)
3. Un diagrama detallado de la red de microcontroladores y servidores web que formen parte de la solución integral, como imagen exportada (es decir, no es válido presentar, por ejemplo, un diagrama DIA o VISIO)
4. El código fuente del programa que corre en el microcontrolador de cada grupo y el código fuente del programa que corre en el microcontrolador maestro en un repositorio de GitHub público por proyecto.

Todo lo anterior deberá presentarse como un archivo comprimido en el campus virtual mediante una actividad que se creará para tal fin.

Habrán dos instancias de consulta presencial los días 08/09/2025 y 29/09/2025.

Durante todo el período de desarrollo del trabajo, los alumnos pueden optar por realizar consultas mediante el campus virtual, o bien a carlos.g.perez.1992@outlook.com o al grupo de Whatsapp de la materia.

Ejemplo de escenario

Una empresa necesita controlar equipamiento nuevo mediante un microcontrolador, y no consigue ningún ingeniero electrónico que haga el trabajo. Usted es el siguiente en la lista.

Debe preparar un programa que haga lo siguiente:

1. El equipamiento nuevo contiene una mezcladora cuya velocidad es controlada por un multiplicador de voltaje que opera con una tensión de entre 0 y 5 volts. La empresa necesita que el microcontrolador genere formas de onda para controlar dicha mezcladora mediante uno de los pines PWM (a elección del ingeniero). Las distintas frecuencias y formas de onda representan distintos "programas de mezclado".
 - a. El programa tiene que aceptar los siguientes comandos mediante comunicación serial para generar formas de onda
 - i. SIN para onda senoidal
 - ii. SQUARE para onda cuadrada
 - iii. RAMP para onda diente de sierra creciente
 - b. A continuación del comando, se debe especificar la frecuencia en Hz.
 - i. SIN 100 para una onda senoidal de 100Hz
 - ii. RAMP 50 para una onda diente de sierra de 50 Hz
 - c. Deberá ocupar todo el rango posible de salida del Arduino.
Por ejemplo, para la onda cuadrada, el valor mínimo será 0V y el máximo 5V.
(Prestar atención a los valores negativos de la función seno).
 - d. Puede optar por utilizar una tabla de muestras (wavetable) para generar una de las formas de onda a elección. Al menos dos de las formas de onda generadas deben utilizar funciones matemáticas u otros mecanismos.
 - e. Para controlar que las formas de onda se estén generando correctamente, el microcontrolador debe transmitir los valores generados por comunicación serial, para graficarlos mediante el monitor serial, y se debe poder apreciar el cambio de la función mediante algún mecanismo que el grupo prefiera con componentes electrónicos (un voltímetro que muestre los valores, un led que cambie de brillo a medida que cambie el valor, etc.).
2. A su vez, el equipamiento cuenta con un motor que funciona a régimen constante, pero que cuenta con un sistema de refrigeración insuficiente. La empresa está en tratativas de mejorar el sistema de refrigeración, pero las partes necesarias tardarán meses en llegar. Para no reducir la productividad, se llegó a la conclusión de que poniendo un ventilador industrial apuntando directamente al sistema de refrigeración será suficiente para el corto plazo.

La empresa necesita que el microcontrolador utilice un sensor para medir la temperatura y, en base a ciertos lineamientos, variar la velocidad a la que gira el motor de corriente directa del ventilador.

- a. El sensor de temperatura es un sensor TMP36. El sensor estará conectado a uno de los pines analógicos (A0 – A5) a elección. Usted deberá buscar la documentación del componente en internet, y convertir las lecturas de voltaje del sensor a un valor en grados Celsius.
- b. El motor del ventilador estará conectado a un pin PWM a elección, distinto al pin que se utilizó en el punto 1.
El motor tiene su régimen máximo a 5V, exactamente el máximo voltaje que puede entregar el Arduino, y responde linealmente al cambio de voltaje. Es decir que cuando el voltaje sea 0V el motor estará apagado, cuando sea 5V el motor estará prendido y a máximo régimen, y cuando esté a 2.5V el motor estará funcionando a la mitad de la velocidad máxima.
- c. La temperatura operativa del motor a enfriar es entre 40 y 85 grados Celsius.
 - i. Cuando el motor esté operando a menos de 40 grados, el ventilador deberá estar apagado.
 - ii. Cuando el motor esté operando a más de 85 grados, el ventilador deberá estar funcionando a régimen máximo.
 - iii. Entre 40 y 85 grados, habrá una relación lineal entre la temperatura y la velocidad del motor, donde 40 grados = 0 RPM; 85 grados = MAX RPM.
- d. El programa deberá controlar la velocidad del motor DC en respuesta al cambio de temperatura detectado por el sensor inmediatamente.
- e. A su vez, el programa deberá aceptar el comando VENT. Cuando este comando sea recibido por comunicación serial, el programa dejará de graficar la forma de onda del punto 1 y graficará en cambio la velocidad del ventilador (expresada en volts, milivolts, o en valor interno del Arduino PWM).