
Laboratorio # 4

Arduino: GPIO, Timers, ADC, comunicaciones, PWM, PM, EEPROM, Iot

Instrucciones Generales:

Este laboratorio se puede realizar de manera individual o en pareja. Se debe utilizar git para registrar el avance y aporte de cada estudiante(aquel estudiante que no registre aportes/contribuciones tendrá una nota de cero), en caso de hacer el laboratorio en pareja se debe utilizar un mismo repositorio.

El laboratorio debe de entregarse antes del 1 de noviembre a las 23:59.

Utilice capturas de pantalla para demostrar la funcionalidad, estas capturas de pantalla deben mostrar sólo la información pertinente al paso correspondiente.

Entregue un archivo comprimido que incluya un directorio llamado **informe** con los archivos necesarios para generar el PDF del informe (.tex, imágenes, código, entre otros) y un directorio llamado **src** con los archivos de código fuente que lleven a la solución. Cualquier otro formato o entrega tardía no se revisará y el laboratorio tendrá una nota de cero.

Estación meteorológica



Figura 1: Estación meteorológica

Se deberá desarrollar una estación meteorológica para registrar y estudiar la condición del tiempo midiendo variables físicas de temperatura, humedad, velocidad del viento e intensidad luminosa. Esta estación meteorológica se ubicará dentro de parques nacionales, la solución deberá considerar las condiciones de difícil acceso, alimentación por baterías (cargadas por paneles solares) y poco ancho de banda para el envío de datos. Por lo tanto debe cumplir con los siguientes requisitos:

1. Debe utilizar una placa Arduino Mega.
2. Leer la temperatura con un sensor RTD o un termistor.
3. Leer un sensor de humedad (DHT22). Puede simular esta salida con una fuente de voltaje variable que cumpla con las características eléctricas de operación del sensor.

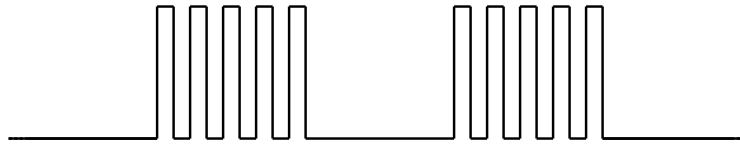


Figura 2: Diagrama de temporización para led intermitente

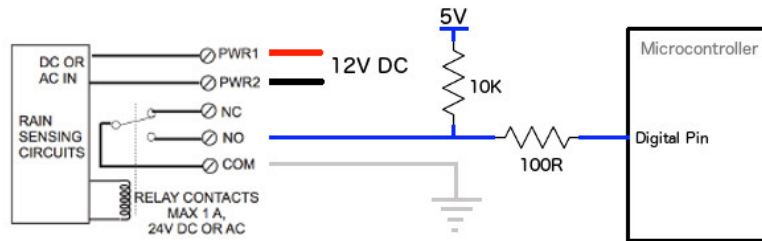


Figura 3: Circuito para el sensor de lluvia.

4. Leer la intensidad luminosa con un sensor LDR.
5. Leer un sensor de velocidad de viento (JL-FSX2 4-20MA). Puede simular esta salida con una fuente de voltaje variable que cumpla con las características electricas de operación del sensor.
6. Leer un sensor de lluvia (Hydreon RG-11). Cada vez que una cantidad predeterminada de lluvia golpea el sensor cambiará un relé, este interruptor es detectado por el Arduino. Ver figura 3 para el circuito con que se conecta el sensor al Arduino. Simule la salida del sensor con un switch que maneja el relé.
7. Utilizar una pantalla LCD PCD8544 donde se visualicen los valores de los sensores de temperatura, humedad, intensidad luminosa, velocidad del viento y si se detecta lluvia.
8. Incluir un switch para habilitar/deshabilitar el despliegue de información en la pantalla LCD PCD8544
9. Cada 5 minutos debe guardar en la memoria EEPROM los valores de los sensores de temperatura, humedad, intensidad luminosa, velocidad de viento y detección de lluvia. En caso de llenar la memoria EEPROM debe borrar todo el contenido e iniciar nuevamente con la escritura de datos.
10. Incluir un switch para habilitar/deshabilitar comunicaciones por una USART
11. Un led debe parpadear indicando la habilitación de la transmisión/recepción de datos por el periférico USART. Con el fin de prolongar la vida útil del led, este deberá parpadear de forma intermitente mientras se encuentre activo (ver figura 2). Las frecuencias y ciclos de trabajo quedan a criterio propio.
12. Cada 10 minutos debe enviar por la USART (si esta habilitado la comunicación de datos) los valores de los sensores de temperatura, humedad, intensidad luminosa, velocidad de viento y detección de lluvia.
13. Crear un script de python que lea el puerto serial y que envíe la información de los sensores de temperatura, humedad, intensidad luminosa, velocidad de viento y detección de lluvia para ser desplegados en un dashboard de una plataforma Iot thingsboard (Queda a criterio propio la utilización de los widgets para visualizar esta información)
14. Debe leer el nivel de la batería cuyo rango es de $[0,12]V$, en caso de estar cerca del límite mínimo de operación del Arduino Mega debería encender un led de alarma parpadeante y enviar la notificación de batería baja al dashboard de thingsboard, posteriormente debe entrar a un modo de ahorro de energía y mantenerse en el mismo hasta tener un voltaje de operación adecuado. Puede simular esta salida con una fuente de voltaje variable.
15. La dirección del panel solar debe ser manejado utilizando servo motores. Un servo motor para la orientación horizontal y otro servo motor para la orientación vertical. (Ambos operando con un rango de 180 grados donde 90 indica el centro)
16. Utilizar potenciómetros para ajustar la dirección del panel solar.

Puntos opcionales con puntaje extra(todo o nada):

17. Habilitar el control remoto del servo motor utilizando widgets de control de la plataforma Iot thingsboard
18. En caso de cumplir con el punto anterior. Incluir un switch que conmute el control de los servo motores entre el control presencial por medio de potenciómetros y el control remoto por medio de la plataforma Iot.

19. En caso de cumplir con el punto anterior. Incluir un led indicando que el control remoto esta habilitado. Con el fin de prolongar la vida útil del led, este deberá parpadear de forma intermitente mientras se encuentre activo (ver figura 2). Las frecuencias y ciclos de trabajo quedan a criterio propio.