Trabajo Practico 3
Temporización - Memoria EEPROM

Martín Cogo Belver

19 de marzo de 2024

1. Detalles de implementación

1. **Comunicación Serial**

   a) *USBport*: Esta clase es una interfaz que provee los métodos para enviar y leer los datos del bus serie.

   b) *Protocolo de comunicación*:

   - El servidor envía tramas con la forma "*sym;data*".
     - **sym**: Puede ser alguno de los siguientes caracteres:
       - **"?"** Indica al Arduino que envie todos los eventos alamcenados-
       - **"*"** Indica al Arduino data configurará el tiempo del reloj.
       - **"~"** Indica que se debe eliminar todos los evento en la EEPROM.
     - **data**: Si se envía el símbolo "*" contiene la cantidad de segúndos con los que se configurara el reloj.
   - El Arduino responde con tramas con la forma "*type;time;pin*".
     - **type**: Indica si el dato enviado por el Arduino es tiempo (t) o un evento (e).
     - **time**: Indica en segundos el tiempo enviado por el Arduino.
     - **pin**: si el type es (e) indica el numero de pin que causo la interrupción en el Arduino.

2. **Comunicación Cliente-Servidor**

   a) El socket se crea al conectarse a /tp3/dashboard

   b) El script *dashboard.js* en el **cliente** se encarga de detectar eventos y enviar el tiempo cuando el usuario quiere actualizarlo desde la vista hasta el servidor utilizando la librería socketIO con javascript.

3. **Threads**

   a) *SendDataThread*:Espera a que la cola tenga algún elemento para enviárselo al cliente.

   b) *SendTimeThread*: Envía periódicamente el tiempo al cliente si cambia de el ultimo valor enviado.

   c) *ReadThread*: Este hilo se encarga de leer el contenido del bus utilizando la clase **USBport** y almacena eventos en una cola, y el tiempo en una variable.

4. **Tareas de FreeRTOS**

   - *AddOneSecondTask*: Tarea que se encarga de añadir 1 segundo al tiempo almacenado en la EEPROM utilizando **vTaskDelayUntil**.
   - *SendTimeTask*: Tarea que se encarga de enviar el tiempo en segundos por el bus serial.
   - *ReadSerialTask*: Tarea que se encarga de la lectura del bus serie.

5. **Interrupciones**

- *pin2Interruption*: Rutina que interrumpe cuando el botón en el pin 2 cuando es soltado y almacena el evento en la EEPROM.
- *pin3Interruption*: Rutina que interrumpe cuando el botón en el pin 3 cuando es soltado y alamcena el evento en la EEPROM.

2. Librerias utilzadas

1. Flask framework: Para la creación de la página web.
2. pySerial: Para la comunicación con puerto Serial.
3. socketio: Para la comuniación con websockets.

3. Diagrama

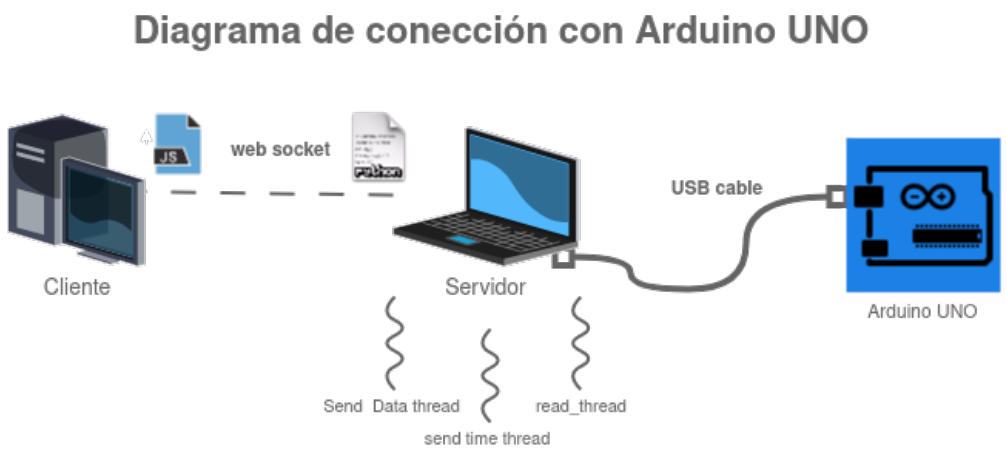

Figura 1: Diagrama representativo y los hilos del programa

4. Página web

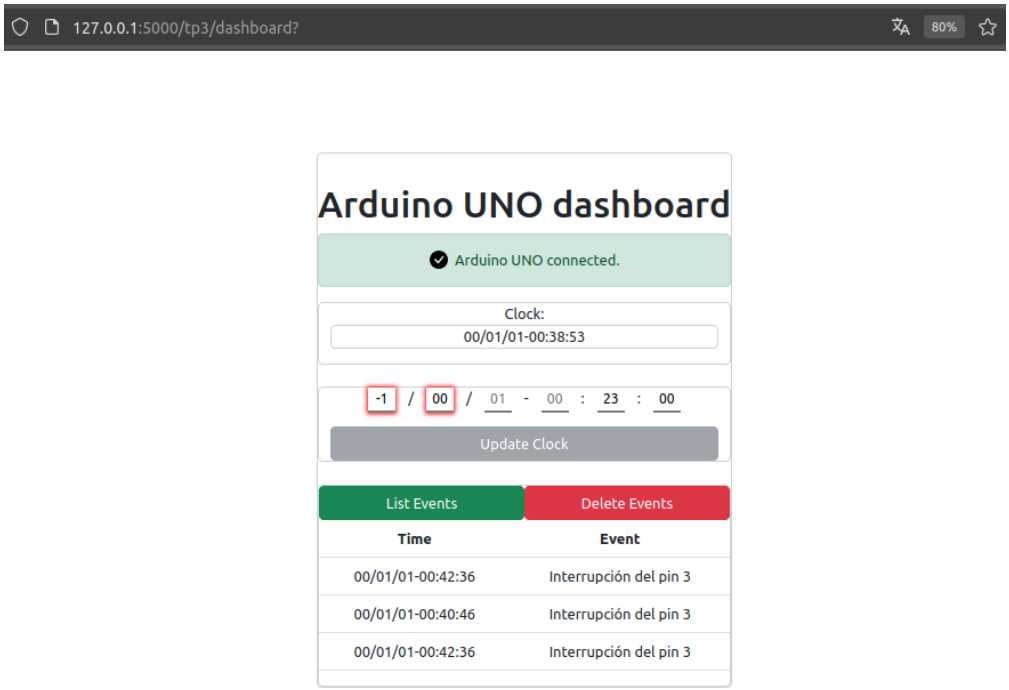

Figura 2: Página web diseñada para controlar el Arduino UNO.