Trabajo Práctico N°3 - Sistemas Embebidos

Licenciatura en Ciencias de la Computación, Facultad de Ingeniería, UNCuyo

Integrantes:

- Masuelli, Luciano
- Silva, Yeumen
- Yornet de Rosas, Agustín

Actividad 1

> **Enunciado 1.** Implemente en el microcontrolador un reloj en tiempo real en formato Unix.

> **a.** Al iniciar la aplicación, el backend en Python deberá obtener la hora mediante el protocolo NTP (Network Time Protocol) y transmitir al Arduino la cantidad de segundos desde el origen de tiempos Unix.

El siguiente código de Python se encarga de obtener la hora NTP y enviarla al Arduino.

```
def get_ntp_time():
    try:
        client = ntplib.NTPClient()
        response = client.request('pool.ntp.org')
        unix_time = int(response.tx_time)
        ser.write(f"{unix_time}\n".encode())
        return unix_time
    except Exception as e:
        print(f"Error al obtener la hora NTP: {e}")
        return None
```

> **b.** El microcontrolador solo deberá incrementar una unidad cada segundo.

El siguiente método de C++ incremanta una unidad cada segundo, utilizando un mutex para administrar el uso del recurso compartido.

```
void incrementSecond(void *pvParameters) {
  TickType_t xLastWakeTime = xTaskGetTickCount();
  while (1) {
    if (xSemaphoreTake(xMutex, portMAX_DELAY)) {
      unixTime++;
      hours = ((unixTime % 86400L) / 3600) - 3;
      mins = (unixTime % 3600) / 60;
      secs = unixTime % 60;
      xSemaphoreGive(xMutex);
    }
    Serial.print("Hora: ");
    Serial.print(hours);
```

```
      Serial.print(":");
      Serial.print(mins);
      Serial.print(":");
      Serial.println(secs);

      vTaskDelayUntil(&xLastWakeTime, pdMS_TO_TICKS(1000));
    }
  }
```

> **c.** La página web debe incluir un botón “Actualizar hora desde servidor NTP” que actualice la hora usando el protocolo NTP.

Este código de Javascript se encarga de darle la funcionalidad al botón que actualiza la hora. Cuando se hace click en el botón realiza un fetch a el endpoint `/update_time` de la api desarrollada en Flask con un método de POST para actualizar la hora.

```
document.getElementById('updateTime').addEventListener('click', () => {
    fetch('/update_time', { method: 'POST' })
        .then(response => response.json())
        .then(data => {
            document.getElementById('message').textContent = data.message
|| data.error;
        });
});
```

Este endpoint de Flask al momento de recibir la petición utiliza el método desarrollado en el inciso `a` para obtener la hora NTP y enviarla al Arduino.

```
@app.route('/update_time', methods=['POST'])
def update_time():
    unix_time = get_ntp_time()
    if unix_time is not None:
        return jsonify({'message': f'Hora actualizada a
{ctime(unix_time)}'})
    else:
        return jsonify({'error': 'No se pudo actualizar la hora'}), 500
```

> **Enunciado 2.** Implementar interrupciones en los pines 2 y 3 del Arduino Uno, de modo que cuando se presionen los pulsadores conectados a dichos pines, se almacene en la memoria EEPROM la hora UTC completa a la cual ocurrió el evento y el evento ocurrido. La función debe ser tal que, si el Arduino se desconecta de la fuente de alimentación y se vuelve a conectar, nuevas escrituras en la EEPROM no deben sobreescribir escrituras anteriores.

Las interrupciones se setearon de la siguiente manera en los pines 2 y 3.

```
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(2), []() { storeEvent(2); },
FALLING);
  attachInterrupt(digitalPinToInterrupt(3), []() { storeEvent(3); },
FALLING);
```

El método `storeEvent` es el encargado de almacenar en la memoria EEPROM la hora UTC a la cual se presionaron los pines.

```
void storeEvent(int pin) {
  if (xSemaphoreTake(xMutex, portMAX_DELAY)) {
    EEPROM.put(eepromAddress, unixTime);
    EEPROM.put(eepromAddress + sizeof(long int), pin);
    eepromAddress += sizeof(long int) + sizeof(int);
    xSemaphoreGive(xMutex);

    Serial.print("Evento guardado en EEPROM: Pin ");
    Serial.print(pin);
    Serial.print(" a tiempo ");
    Serial.println(unixTime);
  }
}
```

> **Enunciado 3.** Implementar funciones que permitan recuperar la información almacenada en la EEPROM para mostrarse en la página web cuando se presione un botón desde la misma. En la página web se deberán mostrar los eventos ocurridos y la hora UTC en formato legible.

Se implementó esta funcionalidad de la siguiente manera, donde el método `retriveEEPROM` recorre las direcciones de memoria y retorna la información.

```
void retrieveEEPROM() {
  Serial.println("Eventos guardados:");
  int address = 0;
  while (address < EEPROM.length()) {
    long int storedTime;
    int storedPin;
    EEPROM.get(address, storedTime);
    EEPROM.get(address + sizeof(long int), storedPin);
    if (storedTime == 0xFFFFFFFF) break;

    Serial.print("Pin: ");
    Serial.print(storedPin);
    Serial.print(", Tiempo: ");
    Serial.println(storedTime);

    address += sizeof(long int) + sizeof(int);
```

```
    }
}
```

> **Enunciado 4.** La memoria EEPROM debe poder borrarse completamente cuando se presione un botón implementado en la página web.

Este código de Javascript al momento de presionar el botón se encarga de hacer un fetch al endpoint `/clear_eeprom`.

```
document.getElementById('clearEEPROM').addEventListener('click', () => {
    fetch('/clear_eeprom', { method: 'POST' })
        .then(response => response.json())
        .then(data => {
            document.getElementById('message').textContent = data.message;
        });
});
```

El backend recibe la petición y ejecuta la función `clear_eeprom_on_arduino()`.

```
@app.route('/clear_eeprom', methods=['POST'])
def clear_eeprom():
    message = clear_eeprom_on_arduino()
    return jsonify({'message': message})
```

En ese momento se inicia el método `clearEEPROM()` en Arduino, donde se limpia la memoria.

```
void clearEEPROM() {
  for (int i = 0; i < EEPROM.length(); i++) {
    EEPROM.write(i, 0xFF);
  }
  eepromAddress = 0;
  Serial.println("EEPROM borrada.");
}
```

> **Enunciado 5.** Toda la interacción del usuario con la aplicación debe realizarse a través de una página web.

Vista previa: