

MONITORAGGIO LUCE AMBIENTALE

Progettare un sistema che misuri la quantità
di luce ambientale in una stanza per
ottimizzarne l'illuminazione artificiale

I2A Gruppo 7

Julian C.
Michea C.
Ambra G.
Giada G.
Roeld H.

Il programma

Il progetto consiste nella creazione di un sistema integrato che misura la luce ambientale e ottimizza l'illuminazione artificiale in base ai livelli rilevati. L'obiettivo principale è regolare l'accensione dei LED in funzione della luminosità. I componenti hardware utilizzati sono:

- **Scheda BasysMX3**: utilizzata come unità centrale di controllo.
- **Sensore di luce Grove (TAOS-TSL2561)**: responsabile della rilevazione dell'intensità della luce ambientale.
- **LED onboard**: simulazione dell'accensione delle lampade LED in base alla luce rilevata.
- **Speaker onboard**: generazione di segnali acustici (beep).
- **Pulsante BTNC**: gestione di interruzioni.

Funzionalità

Menù di selezione

All'accensione del sistema, il LED RGB si illumina di verde e il terminale mostra un menù con le seguenti opzioni:

1. Avvio monitoraggio luce ambientale
2. Visualizza ultima detezione luminosa (in LUX)
3. Reset ultima detezione registrata

Funzione 1: Monitoraggio luce ambientale

- Beep di avvio: generato con un segnale PWM (duty cycle 50%, frequenza 10 kHz).
- Indicatore di stato: il LED RGB si illumina di blu.
- Misurazione della luce: la scheda rileva la luminosità circostante attraverso il sensore Grove per poi convertire il dato in bit
 - I dati vengono visualizzati sul display LCD:

```
Light: 300 LUX  
LED accesi: 3
```
 - Il sistema gestisce l'accensione progressiva degli 8 LED: al buio tutti i LED sono accesi, alla luce tutti i LED sono spenti.
- Interruzione con BTNC: la pressione del pulsante arresta il monitoraggio e salva l'ultima misurazione in memoria flash, salvando il MSB e l'LSB su due registri diversi.

Funzione 2: visualizza ultima detezione luminosa

Sul terminale appare l'ultimo valore salvato in memoria flash. Il sistema legge prima il LSB e poi il MSB, combinandoli per ricostruire il valore completo.

Funzione 3: reset misurazione LUX

Cancella la memoria flash e resetta il valore registrato.

Svolgimento del programma

Da momento in cui il sensore inizia a rilevare la luminosità il processo si divide in tre fasi:

1. Il sistema **legge i livelli di luce** ambiente tramite il sensore TAOS-TSL2561.
2. I **dati vengono analizzati** per determinare la necessità di illuminazione.
3. In base ai risultati, il sistema **accende** il numero di LED necessari.

Lettura dei livelli di luce

Il sensore Grove sfrutta un fotodiodo per rilevare la luce visibile e infrarossa. I valori ottenuti vengono convertiti in dati digitali grazie a un convertitore analogico-digitale (ADC) interno, garantendo una misura precisa dell'intensità luminosa. Questa informazione viene poi inviata alla scheda BasysMX3 tramite un'interfaccia I2C, consentendo un'integrazione rapida e affidabile con il sistema di controllo. Inoltre, questa operazione può essere interrotta in qualsiasi momento attraverso il pulsante BTNC.

Analisi dell'illuminazione

I dati vengono analizzati e vengono calcolate le soglie dividendo la massima luminosità rilevata dal sensore (1800 LUX) per il numero di LED (8), determinando incrementi di 225 LUX per ciascun LED.

Accensione LED

In base ai risultati, i LED vengono accesi o spenti direttamente impostando il valore dei bit nella porta LATA.

```
// Funzione per aggiornare i LED in base al valore di Lux
void update_leds(int lux) {
    // Calcola il numero di LED da accendere in base al valore di lux
    int num_leds = NUM_LEDS - (lux * NUM_LEDS) / MAX_LUX;

    // Aggiorna i LED
    unsigned char pattern = 0x00;
    for (int i = 0; i < num_leds; i++) {
        pattern |= (1 << i); // Accende il LED corrispondente
    }

    // Imposta il pattern sui LED (porta A)
    LATA = (LATA & 0xFF00) | pattern;
}
```

Per ulteriori dettagli riguardo all'ordine di esecuzione del programma, è presente la sezione di intitolata "[FlowChart](#)".

TSL2561.c

Gran parte della logica del funzionamento del sensore si trova in questo file, in particolare nella funzione “read_lux,”. Questa funzione legge i dati dai canali dedicati del sensore, assegnandoli alle variabili CH0 e CH1. CH0 corrisponde al canale della luce a banda larga, mentre CH1 misura la luce infrarossa.

Una volta acquisiti i valori dai canali, i dati vengono verificati per escludere eventuali letture errate. Successivamente, viene calcolato il rapporto tra i due canali per determinare con precisione l'intensità luminosa dell'ambiente.

```
// Funzione per leggere i dati di luce (lux) dal sensore
unsigned int TSL2561_read_lux(void) {
    uint16_t CH0 = TSL2561_read_channel(TSL2561_REG_DATA0LOW, TSL2561_REG_DATA0HIGH); // Canale 0
    uint16_t CH1 = TSL2561_read_channel(TSL2561_REG_DATA1LOW, TSL2561_REG_DATA1HIGH); // Canale 1

    if (CH0 == 0xFFFF || CH1 == 0xFFFF || CH0 > 65535 || CH1 > 65535) {
        UART4_WriteString("Sensore saturato o errore nella lettura.\r\n");
        return 0;
    }

    // Calcola il rapporto tra i canali e determina il valore di lux
    float ratio = (float)CH1 / (float)CH0;

    float lux = 0.0f;
    ...
}
```

Problematiche

Poco dopo la consegna del sensore, è stato rilevato un malfunzionamento. Durante il primo montaggio, a causa della particolare architettura del dispositivo, è stato commesso l'errore di inserirlo al contrario. Il sensore fornito dalla SUPSI risultava modificato rispetto a un normale sensore Grove: i pin erano stati tagliati e fissati sul lato opposto. Sebbene i sensori fossero identici per il resto, questo errore ha reso necessario l'ordine di un nuovo dispositivo, causando un leggero ritardo rispetto alle previsioni iniziali.

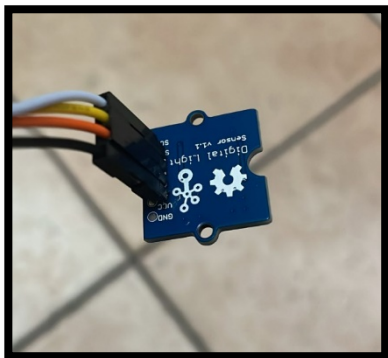


Figura 1: Sensore fornito dalla scuola

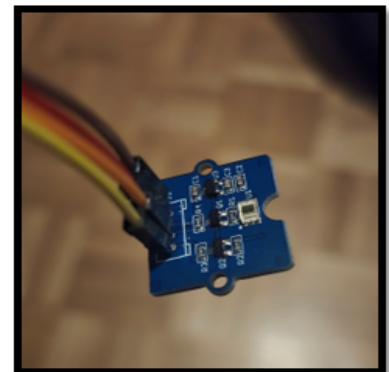
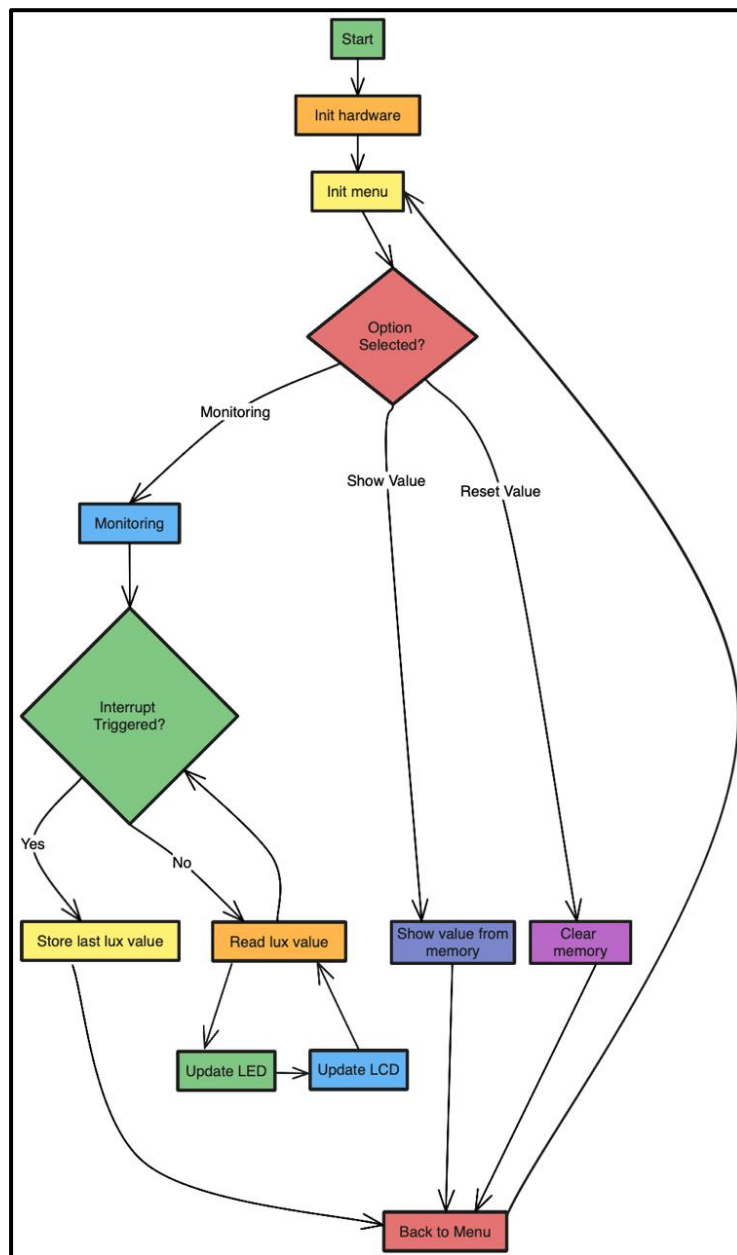


Figura 2: Sensore ufficiale

Flowchart



Questo *flowchart* illustra il semplice funzionamento del nostro programma. Una volta inizializzate tutte le componenti hardware viene mostrato il menù con le tre scelte. Per la prima opzione il programma continua a leggere la luminosità tramite il sensore, mostrandola su LED e LCD. Se dovesse venir premuto il pulsante di *interrupt* il programma salva il valore corrente e ritorna al menu. Per la seconda e la terza viene rispettivamente o mostrato l'ultimo valore salvato nella memoria oppure viene resettata la memoria stessa. Poi il programma ricomincia.

Conclusioni

Nel corso di questo progetto, abbiamo non solo potuto mettere veramente in pratica tutte le nozioni apprese durante il semestre ma anche, per la prima volta, confrontarci con una nuova sfida: una periferica completamente esterna alla nostra board. È stato quindi stimolante lavorare per raggiungere l'obiettivo finale, che è forse più vicino ad una vera e propria possibile situazione reale. Un'altra cosa che abbiamo apprezzato è stato il poter sfruttare quasi tutte le librerie che abbiamo studiato e unirle tutte in un unico programma. Il lavoro all'interno del gruppo è andato bene, non ci sono state discussioni e abbiamo sfruttato bene il tempo; questo ci ha permesso di non solo terminare il progetto ma anche di rifinire alcuni dettagli e migliorare il codice anche da un punto di vista stilistico.

Tutto sommato è stato un bel progetto e una bella sfida da affrontare.