

July 2021

Stazione Meteo Taikomo

Real-Time Embedded Systems
Project

Prepared by:

Giacomo Guiduzzi
Taisia Marconi

Descrizione

Il nostro progetto consiste in una stazione meteo ad utilizzo domestico a cui l'utente può accedere attraverso un'interfaccia web; su questa interfaccia si trovano grafici per mostrare l'andamento dei dati nel tempo.

È possibile interagire con l'interfaccia selezionando da 2 appositi fieldset il periodo di aggiornamento dei dati e le unità di misura di alcuni di questi.

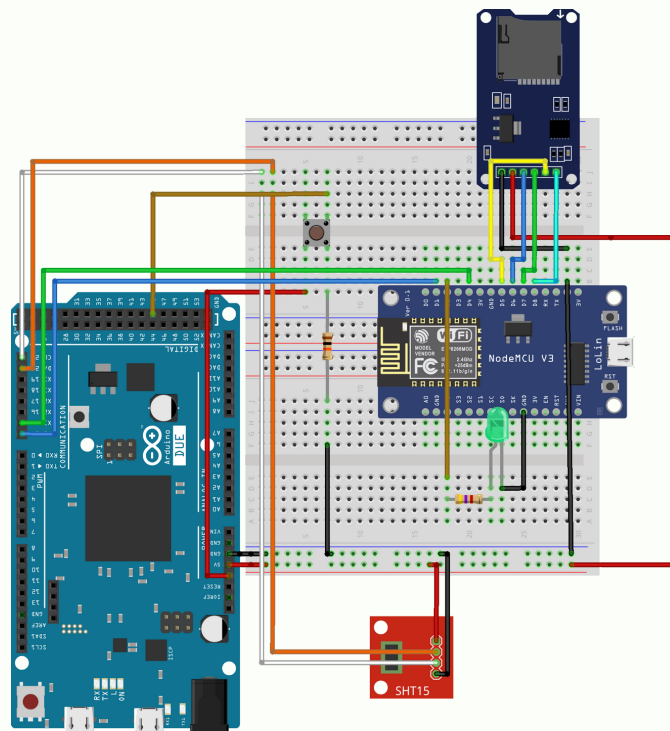
I dati raccolti sono:

- Temperatura;
- Umidità;
- Pressione;
- Altitudine;
- Luminosità ambientale.

Sulla breadboard è presente anche un LED verde che mostra l'effettiva comunicazione tra le schede.

Architettura

Scheda	CPU	FLASH	RAM
Arduino Due	Atmel SAM3X8E ARM Cortex-M3@84MHz	512 KB	96 KB
Lolin ESP-8266 NodeMCU 1.0 (ESP-12E)	ESP-8266@80 MHz	4 MB	64 KB



Struttura

Il progetto è suddiviso su un totale di **4** file:

1. **FreeRTOS_ArduinoDue.ino** - implementazione del SO FreeRTOS;
2. **ESP8266_Webserver.ino** - webserver su ESP-8266;
3. **index.html** - pagina principale dell'interfaccia web;
4. **functions.js** - file di script Javascript per le librerie grafiche e le richieste asincrone AJAX.

Entrambe le schede sono state programmate e testate attraverso **ArduinoIDE**.

Tasks

FreeRTOS, sull'Arduino Due, inizializza 7 task:

- **TaskGetTemp** - recupera la temperatura dal sensore;
- **TaskGetHum** - recupera l'umidità;
- **TaskGetPress** - recupera la pressione;
- **TaskGetAlt** - recupera l'altitudine;
- **TaskGetBright** - recupera la luminosità;
- **TaskSendData** - comunica alla scheda ESP-8266 i dati raccolti quando pronti;
- **TaskGetDelay** - controlla potenziali messaggi dall'ESP-8266.

I primi 5 tasks hanno priorità più bassa, *TaskSendData* ha priorità media e *TaskGetDelay* priorità massima (e periodo più breve di tutti).

TaskSendData non ha periodo in quanto è svegliato tramite un meccanismo di notifica (*xTaskNotifyGive()* e *xTaskNotifyTake()*) messo a disposizione da FreeRTOS.

I task che comunicano con il sensore scrivono all'interno di un array il proprio completamento. Quando questo array è "pieno" significa che tutti i task hanno raccolto il proprio dato; solo a questo punto viene svegliato *TaskSendData* che tramite UART manda la nuova struttura dati al webserver.

TaskGetDelay può ricevere un nuovo delay da parte del webserver, oppure può ricevere una risposta di conferma del nuovo delay impostato quando questo viene modificato dal bottone fisico, il quale gestisce la modifica attraverso un interrupt.

Il delay, che può essere influenzato dal bottone o dal client web, determina il periodo dei task che interrogano il sensore. *TaskGetDelay* ha un periodo fisso.

Requisiti soddisfatti

Obbligatori

- Uso di 1 task per canale input/output: ✓
 - *TaskGetData* e *TaskGetDelay* permettono l'interazione con il web server tramite UART.
- Utilizzo di 3 o più canali di input: ✓
 - Bottone fisico sulla breadboard;
 - Bottone per cambiare delay;
 - Bottone per le unità di misura sulla pagina web.
- Utilizzo di 2 o più canali di output: ✓
 - Pagina web;
 - LED verde che mostra l'effettiva trasmissione di dati tra le schede.
- Utilizzo di 2 o più sensori o 1 sensore ed una scheda di rete: ✓
 - Sensore GY-39 che comprende i sensori di temperatura, umidità, pressione, altitudine e luminosità;
 - Scheda ESP-8266 con funzionalità di rete wireless.
- Utilizzo di 1 o più task di comunicazione many-to-one: ✓
 - I task che raccolgono i dati dal sensore (many) si occupano di notificare *TaskSendData* (one) che invierà i dati al web server.

Requisiti soddisfatti

Addizionali

- Easy:
 - Git versioning: ✓
 - <https://github.com/ServantGrunt/rtes> (repo privata)
 - Periodic execution: ✓
 - Tutti i 7 tasks dell'Arduino Due (8 compreso il Default Task) considerano il tempo di esecuzione nel calcolare il tempo rimanente al prossimo periodo tramite la funzione `xTaskGetTickCount()`;
 - Tuning: ✓
 - Task's constraint optimisation: Il periodo dei task è regolabile e un task, che non ha senso abbia un periodo, viene svegliato solo se soddisfatte certe condizioni;
- Challenging:
 - MISRA C compliance: ✓
 - abbiamo appurato tramite un software chiamato Cppcheck che il nostro software è predicibile in tutte le sue parti e rispetta lo standard.