

# Plant Guardian

Silvia del Piano

Giacomo Acitelli

Alessio Devoto

18 agosto 2019

## 1 WHAT

Ci siamo proposti di realizzare un piccolo software che possa permettere di ricavare informazioni su una pianta, in particolare, attraverso appositi sensori, di ricavare dati sulla temperatura ed esposizione alla luce nella zona dove si trova la pianta, e sull'umidità del terreno.

## 2 HOW

Il software si compone di due parti principali: un client che viene eseguito sul portatile, il cui codice è contenuto nella cartella PC, e un “server” eseguito su un microcontrollore avr ATmega2560, contenuto nella cartella Arduino.

### 2.1 Client e comunicazione attraverso le seriale

Il client prende in input i comandi dell'utente, il quale interagisce con esso tramite la shell, li invia al microcontrollore. Esso li processa e fornisce in output i dati richiesti dal client, che si occuperà di mostrarli all'utente. La comunicazione avviene attraverso la seriale, e la correttezza della trasmissione è controllata tramite un meccanismo di checksum. Sono quindi presenti funzioni di serializzazione e deserializzazione dei comandi e dei dati inviati: il messaggio sarà separato dal checksum calcolato dal carattere \$ , e il tutto verrà inviato come un unico pacchetto. Inoltre per assicurarsi che il microcontrollore abbia ricevuto il comando, esso invia un ACK di conferma ogni volta che ciò accade. Il client è realizzato tramite due grossi loop, dopo le operazioni di inizializzazione, il ciclo più esterno legge il comando dato dall'utente dalla shell, lo manda al microcontrollore, e verifica la ricezione dell'ACK. Il loop più interno riceve tutti i dati inviati dall'avr come risposta al comando ricevuto, e termina dopo che riceve istruzioni per l'invio di un nuovo comando. Il microcontrollore, dopo aver fatto le dovute inizializzazioni, si metterà a leggere periodicamente i sensori e a memorizzare i dati sulla memoria EEPROM. Quando però arriva un comando, scatta l'Interrupt Service Routine collegata alla seriale, che si occuperà di processarlo agendo opportunamente. Se il comando non è riconosciuto ne è richiesto un altro. I possibili comandi da inviare sono:

- `Read humidity sensor` : per leggere il sensore di umidità.
- `Read temperature sensor`: per leggere il sensore di temperatura.

- Read photosensor : per leggere il fotosensore.
- Log : per vedere tutti i dati nella memoria EEPROM.
- quit : per arrestare l'esecuzione del programma.

Per rendere il programma di più facile fruizione le stampe sono di colore diverso a seconda del contenuto:

- bianco: client
- verde: log relativi al client
- ciano: avr
- blu: log relativi all'avr

## 2.2 Sensori

Si è scelto di utilizzare tre tipi di sensori:

- Sensore di temperatura
- Sensore di umidità del suolo
- Sensore di luminosità dell'ambiente

Questi effettuano letture analogiche i cui risultati sono comunicati al micro-controllore attraverso i pin come mostrato in Figura 1.

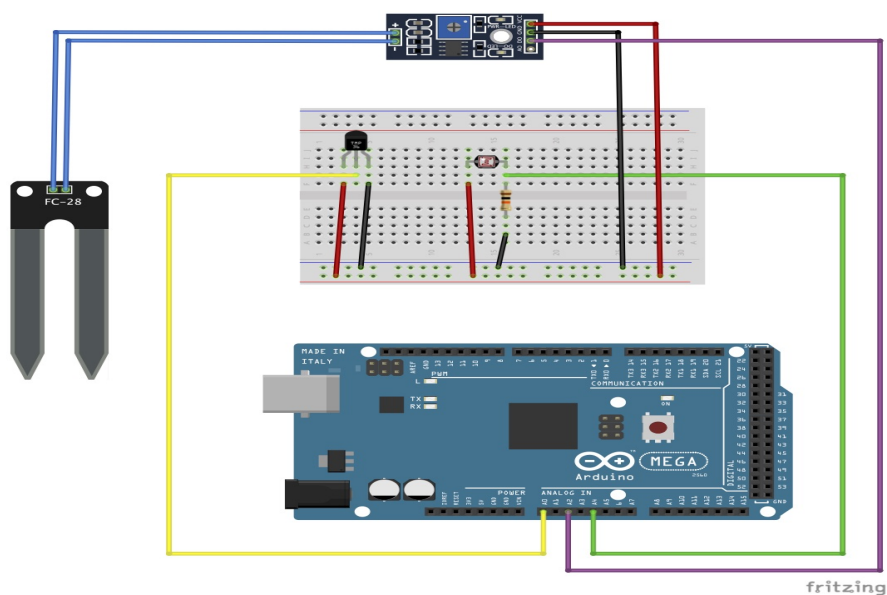


Figura 1: schema

Il codice che si occupa della gestione dei sensori è contenuto nel file “sensor.h”. È presente una funzione che viene invocata una sola volta per settare la conversione ADC dell’AVR, e tre funzioni per ogni sensore. Queste si occupano della preparazione dei registri e dei pin da utilizzare per ogni sensore specifico, della lettura dei dati, e della loro conversione in unità di misura opportune per ogni tipo di dato.

### 2.3 EEPROM

Le funzioni che provvedono alla gestione della EEPROM sono contenute all’inizio del file “Arduino.c”. Si è scelto di salvare i dati letti dai sensori ad ogni iterazione del main loop in una struct che contiene tre interi senza segno `uint16_t`. Ad ogni lettura una struct viene popolata con i dati letti e salvata nella EEPROM in un buffer circolare. Quando il PC richiede una lettura, si utilizza un indice globale per leggere all’indirizzo corrente. Le operazioni di lettura e scrittura sopracitate sono atomiche.

## 3 HOW TO RUN

Dopo aver scaricato il codice dalla cartella di GitHub, la si apra dentro una shell, e si colleghi il microcontrollore al computer. Nella cartella Arduino si eseguano “make” e “make Arduino.hex”. Nella cartella PC si esegua “gcc -o PClient PClient.c Pclient.h”. Per eseguire il programma “./PClient”.