

Documento di specifiche

Klaudio Merja

Mat. 2075538

<https://github.com/klamerja/SensorFlowUNIPD>

Indice

| | | |
|----------|----------------------------------|----------|
| 1 | Introduzione | 1 |
| 2 | Descrizione del modello | 1 |
| 3 | Polimorfismo | 2 |
| 4 | Persistenza dei dati | 2 |
| 5 | Funzionalità implementate | 2 |
| 6 | Rendicontazione ore | 2 |

SensorFlow

Progetto in itinere di Programmazione ad Oggetti
LT in Informatica
Università degli Studi di Padova

1 Introduzione

SensorFlow è un software di gestione per sensori in ambito domotico. Ogni sensore è identificato tramite un UUID ed è caratterizzato da un nome, dalla tipologia e dalla distribuzione dei dati generati.

Le tipologie di sensori per cui l'applicazione fornisce supporto sono:

- **Temperatura e umidità:** permette di analizzare la temperatura (in °C) e l'umidità (in percentuale)
- **Pressione atmosferica** (in hPa - ettopascal)
- **Elettricità:** permette di analizzare il consumo istantaneo (in W - watt) e la tensione elettrica (in V - volt)
- **Qualità dell'aria:** permette di analizzare i livelli di CO2 (in ppm - parti per milione), il PM2.5 ed il PM10 (in µg/m³)

Le operazioni principali che l'applicazione permette di svolgere sono:

- aggiunta/rimozione dei sensori
- modifica delle informazioni relative ai singoli sensori
- visualizzazione dei dati generati

Una delle caratteristiche fondamentali del software è quella di poter visualizzare i dati generati dal sensore in tempo reale.

I dati, per fornire una simulazione del sensore, sono generati secondo una tipologia di distribuzione tra le seguenti:

- Casuale
- Uniforme
- Gaussiana

L'utente ha la possibilità di decidere quale distribuzione adottare per ogni singolo sensore e di modificarla in un secondo momento.

2 Descrizione del modello

Il modello logico è strutturato in due parti: la prima parte comprende le classi che descrivono i vari sensori utilizzabili all'interno dell'applicativo, mentre la seconda parte si occupa di creazione, lettura e aggiornamento del file JSON che si occupa del salvataggio dei sensori.

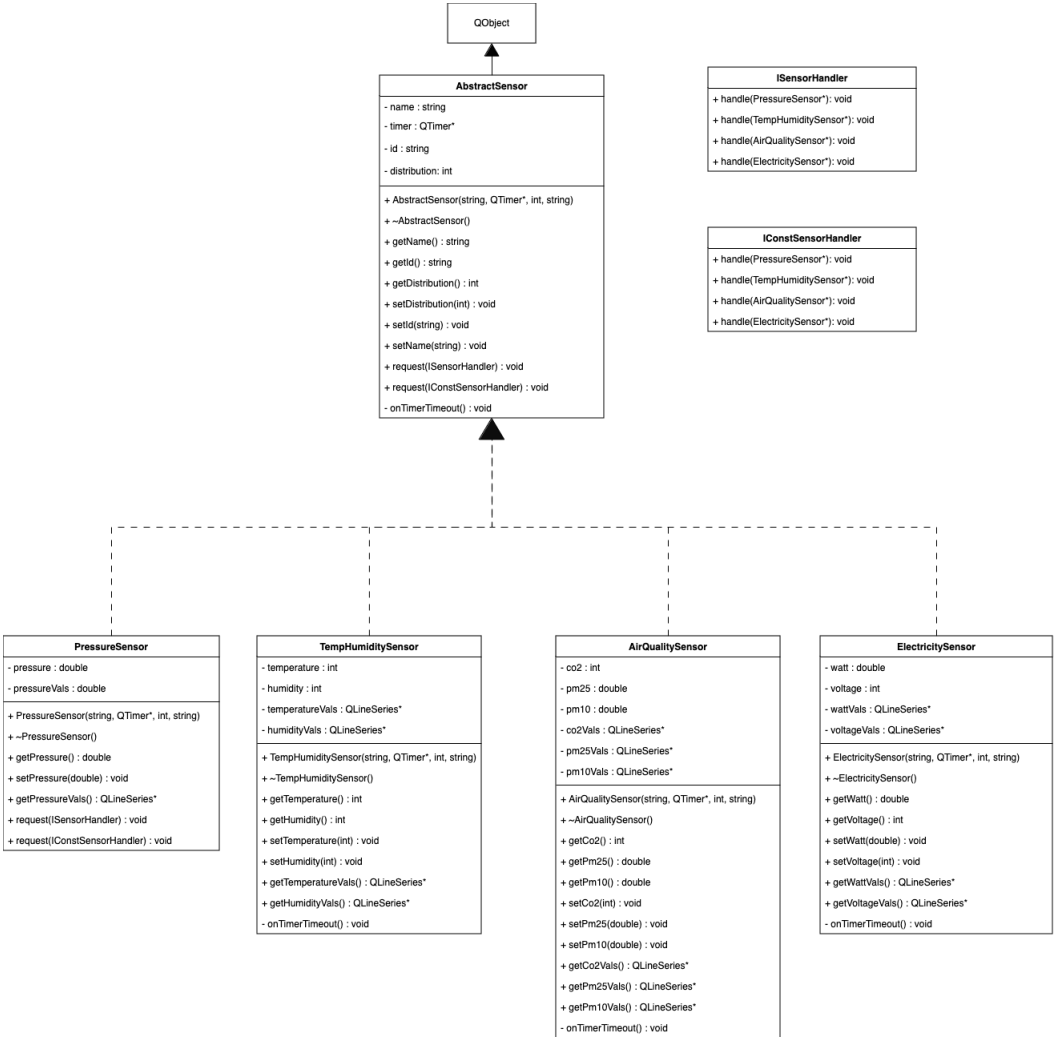


Figura 1: Diagramma delle classi dei sensori

`AbstractSensor` è la classe base astratta che rappresenta le informazioni comuni a tutti i sensori che possono essere creati, ovvero il nome, il timer, l'identificatore univoco `UUID` e la tipologia di distribuzione. Oltre ai relativi metodi *getter* e *setter* per le varie variabili d'istanza, è presente il metodo `onTimerTimeout`, che si occupa di effettuare delle azioni ad ogni *timeout* emesso dal timer: solitamente, le azioni che vengono performate sono quelle di aggiornamento dei valori dei dati che vengono generati dal sensore, oltre a cancellare valori dalle eventuali serie che risultano non necessarie ai fini del funzionamento del software e, in particolare, per la generazione del grafico. Le classi figlie di `AbstractSensor` sono:

- `PressureSensor`
- `TempHumidity`
- `AirQualitySensor`
- `ElectricitySensor`

3 Polimorfismo

4 Persistenza dei dati

5 Funzionalità implementate

6 Rendicontazione ore

| Attività | Ore Previste | Ore effettive |
|---------------------------------|--------------|---------------|
| Studio e progettazione | 10 | 10 |
| Sviluppo del codice del modello | 10 | 10 |
| Studio del framework Qt | 10 | 10 |
| Sviluppo del codice della GUI | 10 | 10 |
| Test e debug | 10 | 10 |
| Stesura della relazione | 10 | 10 |
| Totale | 10 | 10 |