

Sviluppato in C# WPF, con l'aggiunta della comunicazione seriale con Arduino e di scrittura e lettura su un file, ecco il seguente progetto.

L'idea che ho avuto per il suo sviluppo è semplice, ovvero fare un programma su C# che legga i dati mandati sulla porta seriale da Arduino, i quali sarebbero le temperature lette dal DHT Sensor, e mostri un'interfaccia la quale permetta di visualizzare graficamente su una lista e su un istogramma questi dati rilevati, aggiornando in automatico entrambi, lista e istogramma, man mano che i dati vengono inviati sulla seriale e letti dal programma. Sempre in automatico, esso deve salvare tutte le temperature rilevate su un file di backup, e la sua interfaccia deve inoltre permettere all'utente, selezionando una temperatura della lista, di rimuovere e di visualizzare i dettagli di quest'ultima.

Nonostante la semplicità dell'idea, lo sviluppo del progetto è risultato molto interessante e coinvolgente, in quanto ha portato a delle sfide come ad esempio una ricerca accurata delle informazioni utili e necessarie per il progetto oppure la messa in pratica delle conoscenze su C# WPF e su Arduino, in particolare sul DHT Sensor, componente del quale conoscevo ben poco fino a quel momento.

Parlando più nei dettagli del progetto e del suo sviluppo, inizialmente mi sono concentrato su Arduino e sul componente a me necessario, ovvero il DHT Sensor: prima di tutto ho trovato lo schema di collegamento del componente, poi ho cercato una libreria per far sì che esso potesse effettivamente rilevare le temperature, e, una volta trovata, ho scritto il programma in C++ su Arduino IDE per rilevare ogni 20 secondi una temperatura, controllare che fosse valida, e inviarla sulla porta seriale.

Successivamente quindi mi sono spostato alla parte di C# WPF, dove all'inizio ho fatto diversi test per assicurarmi che la comunicazione seriale funzionasse, e che quindi riuscissi a leggere su C# le temperature rilevate da Arduino. Sistemati piccoli problemi e con la comunicazione adesso completamente funzionante, il passo successivo è stato creare l'interfaccia grafica del programma, costituita da bottoni per eseguire determinate azioni sulle temperature, una lista che permette di visualizzarle, fornita anche di una barra di ricerca per trovarne facilmente una o un insieme di esse, e l'istogramma per vederne immediatamente l'andamento. Contemporaneamente, ho sviluppato anche il codice per far sì che al momento della lettura di una temperatura, la lista e il grafico si aggiornassero in automatico, e per far funzionare correttamente le azioni che ciascun bottone deve svolgere.

Archiviata questa fase, ho completato il progetto con l'aggiunta di scrittura e lettura su un file di testo in formato csv: quest'ultima implementazione della gestione del file è necessaria per poter possedere un backup di tutte le temperature rilevate così da non perderle e poterci accedere in futuro. Questo porta quindi alla conseguente possibilità di poter modificare e visualizzare le temperature rilevate in qualsiasi momento si voglia, e risulta fondamentale sia all'avvio del programma per caricare la lista e il grafico delle temperature rilevate fino a quel momento, sia per fare in modo che ad ogni temperatura inviata sulla porta seriale, il dato venga salvato sul file automaticamente.