Realizzazione di un acquisitore di dati atmosferici con trasmissione dati in telemetria

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 7

2 Progettazione 8

2.1 Design del database 8

2.2 Design dell’architettura di sistema 9

3 Implementazione 10

4 Test 10

4.1 Protocollo di test 10

4.2 Risultati test 10

4.3 Mancanze/limitazioni conosciute 11

5 Consuntivo 11

6 Conclusioni 11

6.1 Sviluppi futuri 11

6.2 Considerazioni personali 11

7 Bibliografia 11

7.1 Bibliografia per articoli di riviste: 11

7.2 Bibliografia per libri 11

7.3 Sitografia 11

8 Allegati 11

# Introduzione

## Informazioni sul progetto

* Allievo: Nicola Mazzoletti, allievo 4° informatica
* Docente responsabile: Adriano Barchi
* Scuola Arti e Mestieri Trevano (SAMT)
* Data inizio progetto: 29.08.2017
* Data consegna progetto: 21.12.2017

## Abstract

*In this project, the client asked to me to make a system that can capture some atmospheric data with the help of detectors.*

*The system is called DataLogger and must be StandAlone, with this feature, the client can move it where he wants and capture data from the current location of the station. At the moment that the data were captured, they will be sent throught telemetry, if is possible, to a server that store the data in a database, if instead the server is not reachable, the data will be saved into the memory of the DataLogger.*

*There is also the option to add more than just one station that capture data.*

## Scopo

Lo scopo del progetto è quello di avere un sistema “mobile”, alimentato con una batteria, che sia in grado di rilevare dei dati tramite dei sensori.

I dati che la stazione avrà rilevato saranno inviati tramite telemetria ad un server e verranno salvati in un database.

## Analisi

## Analisi del dominio

Il cliente ha bisogno di un sistema in grado di catturare i seguenti dati: temperatura, umidità e pressione atmosferica. Il prodotto dovrà essere in grado di catturare questi dati e inviarli ad un server tramite l’utilizzo della telemetria sulla frequenza 433 MHz, una volta che il server avrà ricevuto i dati dovrà salvarli in un database. Nel caso che il server non sia raggiungibile, il sistema dovrà salvare i dati localmente e in seguito, una volta ristabilita la connessione dovrà trasmettere i dati salvati.

Questo sistema, che viene comunemente chiamato DataLogger, deve essere alimentato con una batteria in modo che il cliente possa muoverlo senza dover utilizzare una presa per l’alimentazione, in questo modo si crea un sistema Standalone.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | DataLogger |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Implementare un sistema in grado di acquisire, tramite scheda Arduino, la temperatura, la pressione e l’umidità. |
| **002** | Il DataLogger deve essere in grado di inviare i dati ad un Server. |
| **003** | Il sistema deve inviare, oltre ai dati del sottorequisito 001, la tensione della batteria, la data e l’ora al momento dell’invio. |
| **004** | Il sistema deve essere alimentato da una batteria in modo che sia un sistema standalone. |
| **005** | Il DataLogger deve essere identificabile dalla parte di web analisi. |

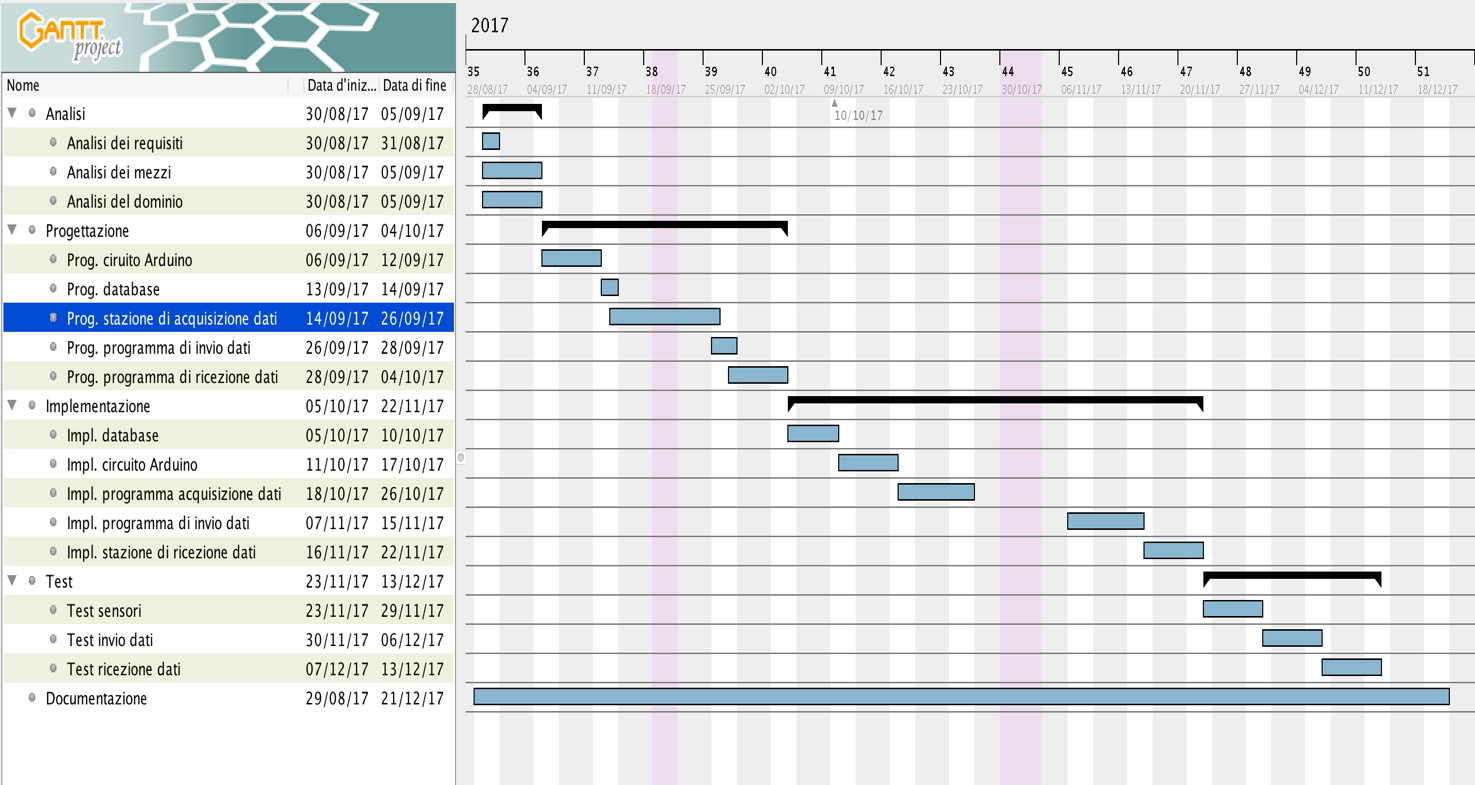
|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Invio dei dati acquisiti |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Il sistema deve inviare i dati tramite telemetria |
| **002** | Per inviare i dati tramite telemetria bisogna utilizzare le frequenze 433 MHz e 900 MHz. |
| **003** | Nel caso non si riesca a trasmettere i dati al server tramite telemetria, bisogna salvare quest’ultimi localmente e inviarli una volta che la connessione sarà ristabilita. |
| **004** | Il sistema di invio dati deve essere attivato solamente quando necessario, quando i dati non vengono inviati il trasmettitore dovrà essere disattivato. |

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Ricezione dei dati |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Il sistema di ricevimento deve essere sempre in ascolto. |

## Use case

## Pianificazione

In questo capitolo è mostrato il Gantt preventivo per l’organizzazione del lavoro nel progetto.



## Analisi dei mezzi

### Software

* Microsoft Word 2016, utilizzato per la stesura della documentazione,
* GanttProject 2.8.5, utilizzato per la creazione del diagramma Gantt,
* Fritzing 0.9.2, utilizzato per progettare lo schema del circuito dell’Arduino, sia su breadboard che su veroboard,
* Creately, web app per creare diagrammi er, usata per la creazione dello schema er del database,

### Hardware

|  |  |  |
| --- | --- | --- |
| **Quantità e tipo di componente** | **Utilizzo nel progetto** | **Altro** |
| 1x Arduino Micro | Gestisce il rilevamento dei dati atmosferici e l’invio di quest’ultimi al server. |  |
| 1x Sensore umidità e temperatura DHT11 | Si occupa di catturare l’umidità e la temperatura. |  |
| 1x Kit Radio Telemetria FPV 433MHz | Componente per l’invio dei dati da Arduino a Server. |  |
| 1x Sensore di pressione BMP180 | Si occupa di catturare la pressione atmosferica. |  |
|  |  |  |
|  |  |  |

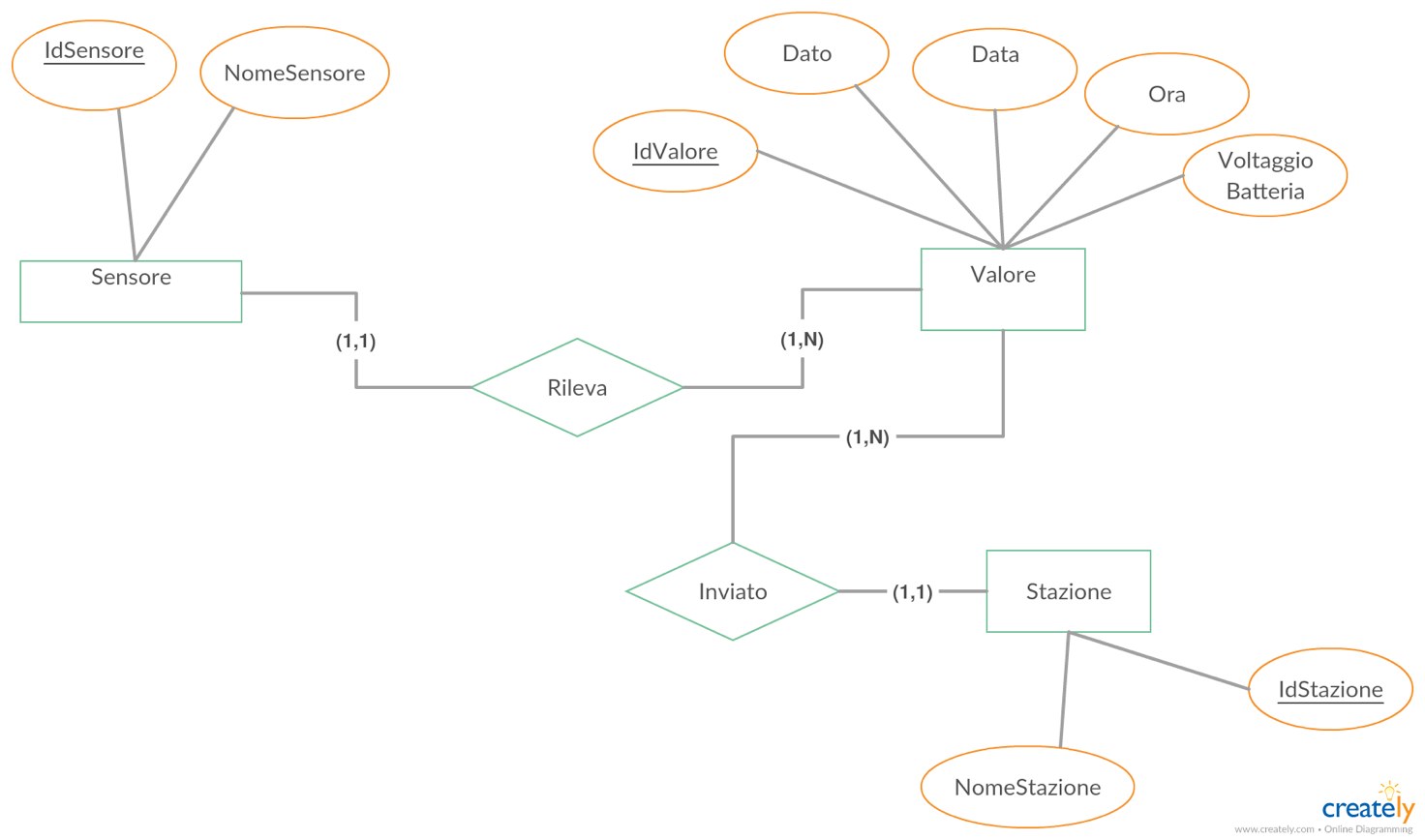
# Progettazione

## Design del database

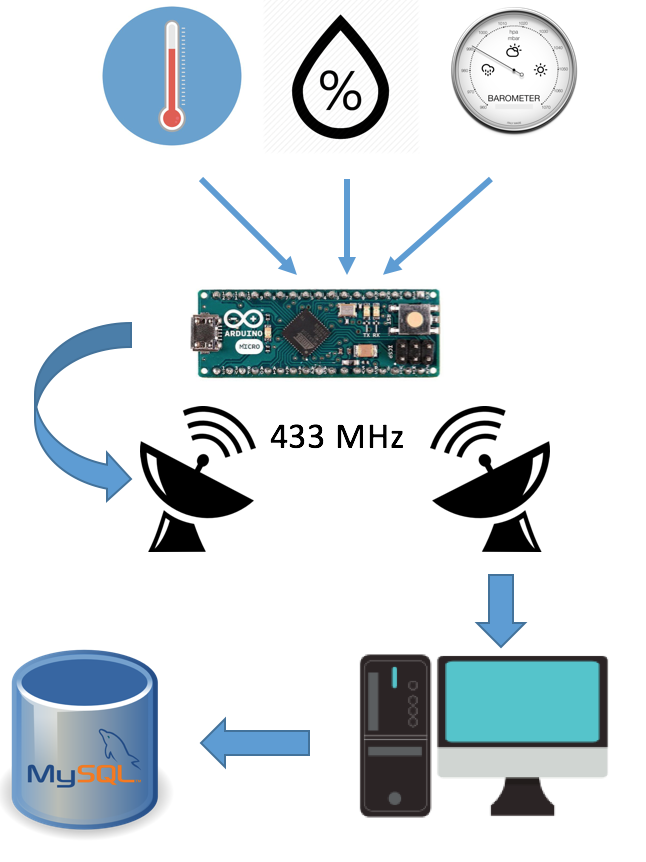
Il seguente schema ER rappresenta la struttura del database dove vengono salvati i dati rilevati.

La tabella ‘Sensore’ serve a salvare i tipi di sensore che sono presenti sulla stazione mentre la tabella ‘Stazione’ è utilizzata per sapere da quale stazione proviene il dato rilevato.

Nella tabella ‘Valore’, vengono salvati i dati rilevati dai sensori, la data e l’ora del rilevamento e il voltaggio della batteria al momento del rilevamento.



## Design dell’architettura di sistema

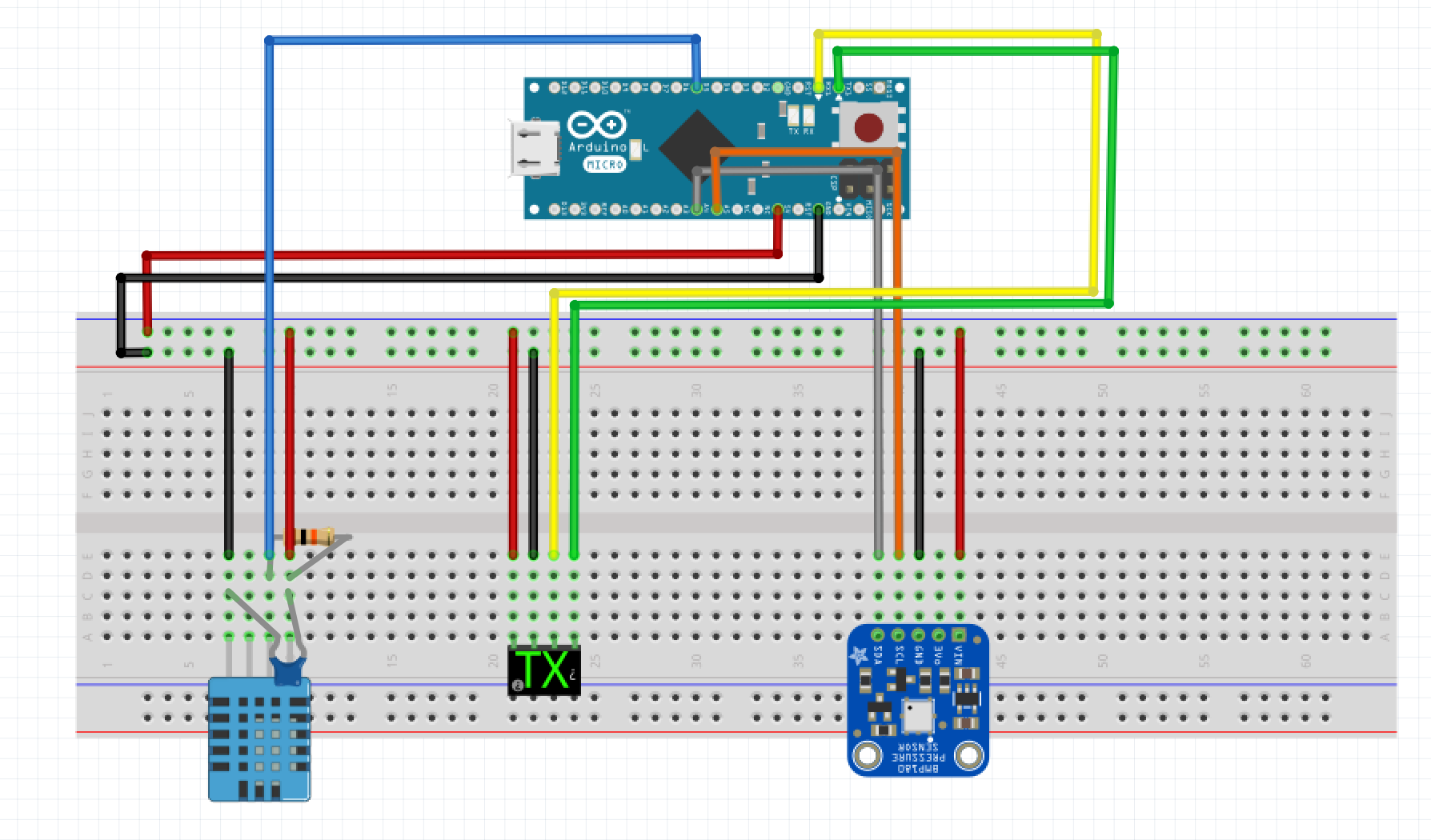


Il sistema è stato strutturato i 3 “sezioni”, la ricezione dei dati, l’invio dei dati e il salvataggio di quest’ultimi.

In primo piano c’è un Arduino Micro che si occupa di raccogliere i dati, la temperatura in gradi Celsius, la pressione in mBar e la percentuale di umidità.

Ogni volta che verranno catturati dei dati, l’Arduino si occuperà tramite l’utilizzo della telemetria di inviare i dati ad un computer che si occuperà di caricarli in un server mysql.

## Design del circuito del sistema di acquisizione



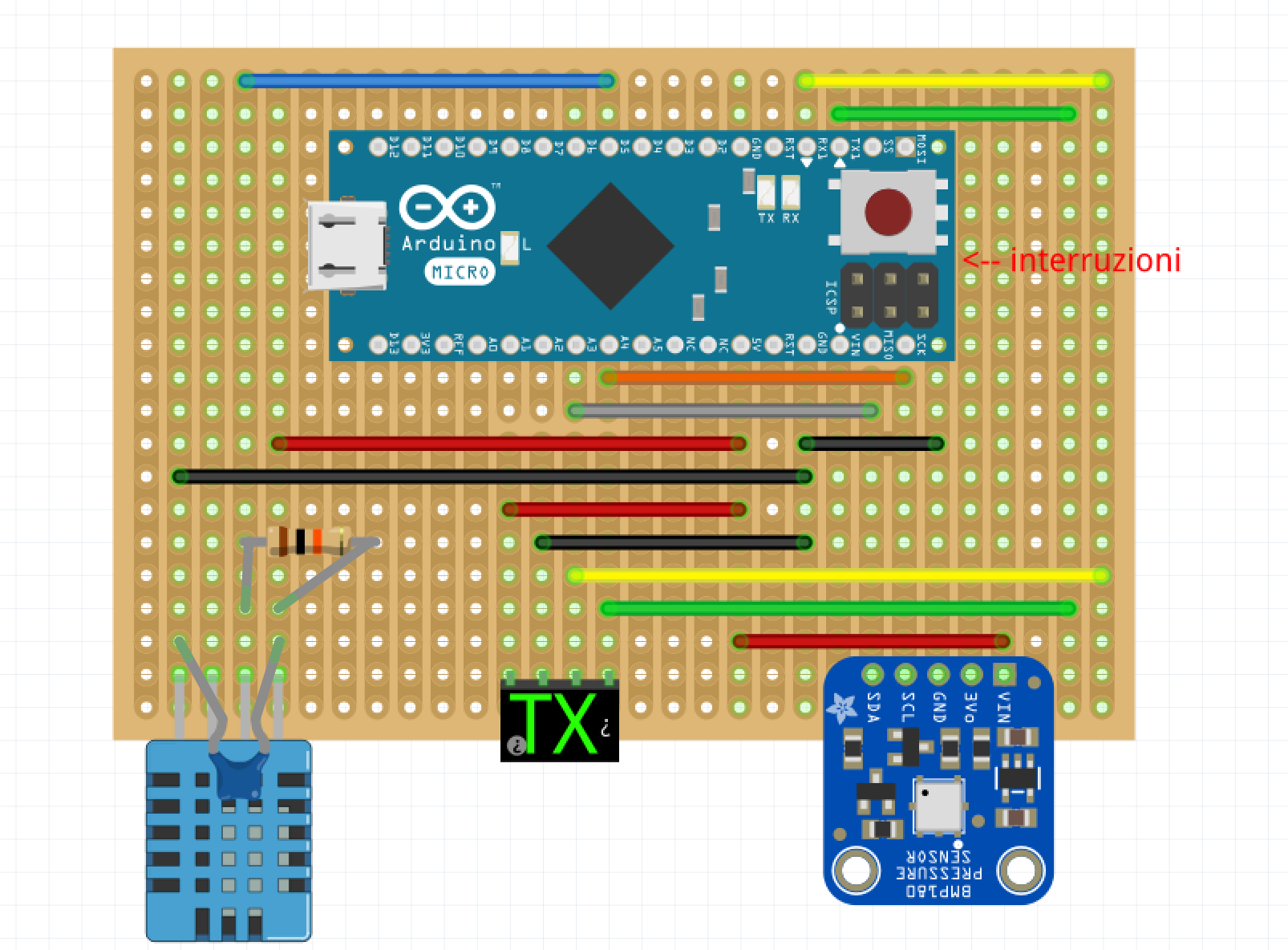
Il seguente circuito si divide in 3 parti, quella del sensore di umidità e temperatura(DHT11), quella del sensore di pressione atmosferica(BMP180) e in fine la parte di invio dei dati acquisiti.

Al sensore DHT11 viene collegato un condensatore da 100nF per filtrare il voltaggio, in questo modo non si ha l’instabilità dei dati. Viene anche collegata una resistenza da 10kΩ (nel modello utilizzato nel progetto, la resistenza è integrata nel sensore).

Il sensore di umidità e temperatura deve essere collegato al pin dei +5V e al GND in modo da alimentarlo, questo viene fatto anche per gli altri due componenti. L’ultimo pin che viene utilizzato su sensore è quello per passare i dati captati, questo viene collegato al pin D5 dell’Arduino.

Per il sensore di pressione vengono utilizzati due pin analogici dell’Arduino, A4 per il pin SDA e A5 per il pin del sensore SCL, mentre per il sistema di invio dati vengono utilizzati i pin RX e TX.

In seguito sarà mostrata la progettazione dello stesso circuito su VeroBoard.



Nel caso del circuito su VeroBoard, bisogna inserire delle interruzioni.

Anche tra le due file di pin dell’Arduino bisogna inserire delle interruzioni, altrimenti i pin faranno passare la corrente tra di loro.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gannt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo,

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o Qdc
* Prodotto
* …