

# Capitolo 4

## Misurazione della glicemia tramite Arduino

Lo scopo della prima parte del progetto è costruire un sistema in grado di scansionare il sensore Freestyle Libre tramite la comunicazione NFC attraverso un modulo apposito. Tale modulo comunicherà con una scheda Arduino la quale, una volta ricevuto i dati grezzi in ingresso, elaborerà i dati attraverso un algoritmo ben definito per ottenere il valore attuale di glicemia.

Il dato glicemico verrà inviato attraverso la comunicazione wireless in un foglio Google Sheets che conterrà lo storico del paziente diabetico e sarà la base di dati su cui farà riferimento l'applicazione "Diabetic" per il calcolo del dosaggio insulinico.

### 4.1 Livello Hardware

Per realizzare il prototipo in grado di comunicare con il sensore e salvare i dati sono stati usati diversi componenti elettronici:

1. Scheda Arduino
2. Modulo WiFi
3. Monitor LCD
4. Supporto con batteria
5. Modulo NFC (BM019)

**1. Scheda Arduino** Il componente principale è la scheda fornita da Elegoo, reperibile su Amazon e compatibile con Arduino IDE. Essa, tramite il codice<sup>1</sup> eseguito all'interno, sarà in grado di ricevere il dato dal modulo NFC, elaborarlo, stamparlo a video sullo schermo LCD e inviarlo tramite porta seriale al modulo WiFi ESP32.

**2. Modulo WiFi** E' stato utilizzato un modulo ESP32 che, una volta ricevuti i dati tramite porta seriale e connesso all'hotspot Wifi, invierà i dati usando il servizio IFTTT in un foglio Google Sheet.

**3. Monitor LCD** Il valore attuale di glicemia ottenuto viene mostrato a video tramite un display LCD 16X2 con retroilluminazione blu dotato di un convertitore I2C che permette il collegamento ad Arduino utilizzando solo due pin per connessione, più i due cavi di alimentazione.

**4. Supporto con Batteria** Per poter funzionare sempre, tutto il sistema viene dotato di un modulo che permette di collegare una batteria ad Arduino e, tramite un pulsante, di accendere e spegnere ad ogni uso.

**5. Modulo NFC** Questa scheda è basata sul modulo ST Micro CR95HF. Tale chip consente alla scheda di comunicare tramite NFC e inviare i dati tramite interfaccia SPI o UART. La frequenza operativa di questo RFID è 13,56 MHz. Supporta tre diversi protocolli che sono ISO/IEC 14443, 15693 (stesso protocollo del sensore Freestyle Libre) e 18092. L'antenna è progettata per trovarsi nella parte inferiore della scheda, la distanza massima di funzionamento per il rilevamento dei tag NFC è di 28 mm a seconda delle misure attuali dell'antenna.

La sua dimensione è di 27,94 mm x 38,10 mm x 1,27 mm e il suo peso è di 45,4 g.

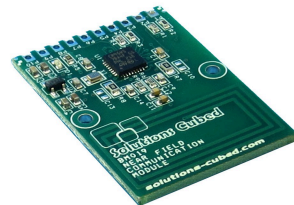


Figura 4.1: Modulo BM019

---

<sup>1</sup>Progetto di riferimento <https://github.com/JoernL/LimitTter>

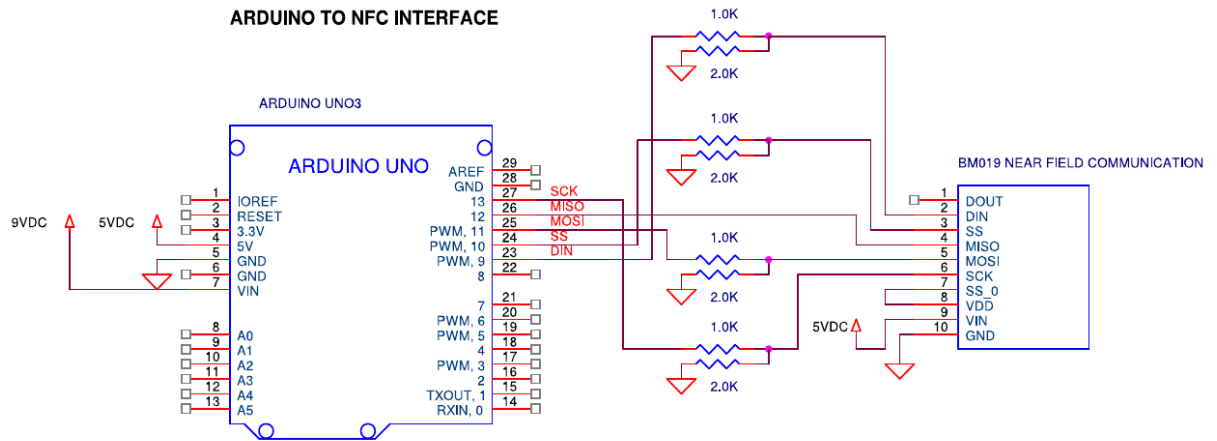


Figura 4.2: Schema elettrico modulo BM019

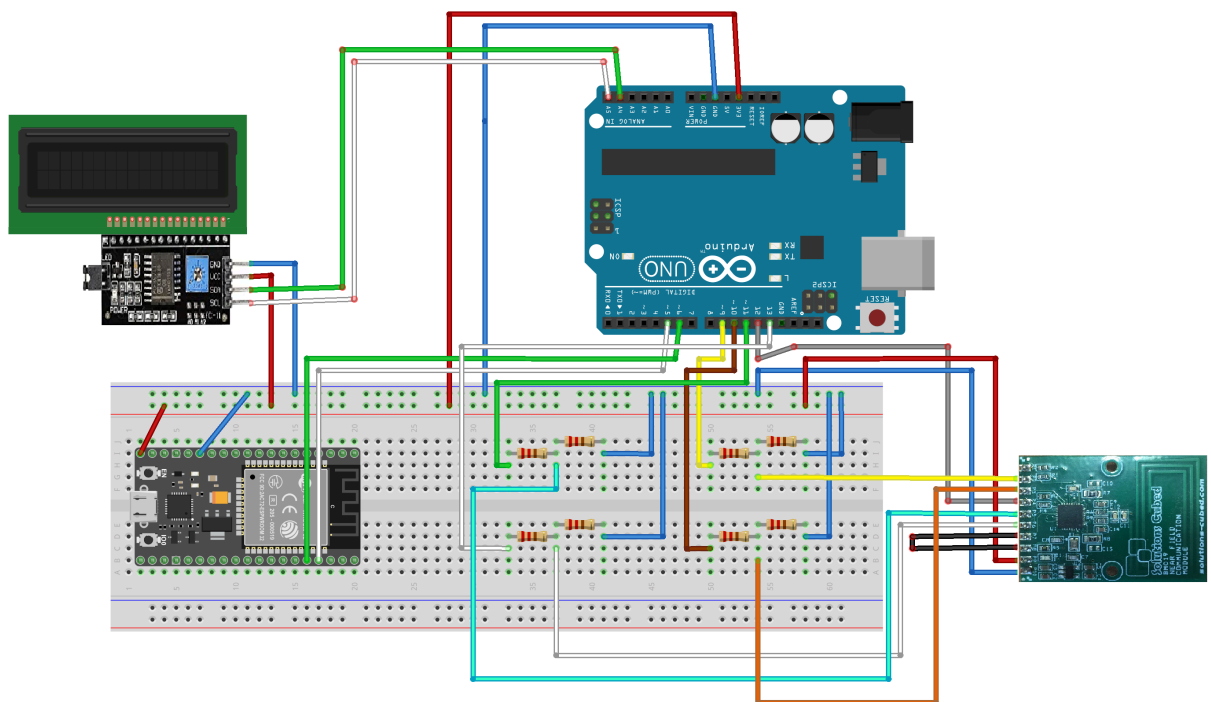


Figura 4.3: Breadboard finale

## 4.2 Livello software

Per scelte progettuali, per non sovraccaricare troppo la memoria e per la poca compatibilità del modulo BM019 con ESP32 è stato deciso di tenere separata la presa dei dati grezzi con la conversione in dati effettivi dall'invio dei dati stessi.

Il codice eseguito nell'Arduino si occupa di scannerizzare tramite NFC il sensore Freestyle Libre, elaborare i dati grezzi e inoltrarli tramite la porta seriale all'ESP32, stampandoli inoltre a video nello schermo LCD.

L'ESP32, una volta ricevuti i dati elaborati (la glicemia attuale), invierà i dati in un foglio Google Sheets che costituirà il diario vero e proprio del paziente diabetico con le relative glicemie misurate durante il giorno.

### 4.2.1 Estrapolazione dei dati dal sensore

L'estrapolazione di dati dal sensore con la relativa elaborazione avviene per mezzo di Arduino. Per la comunicazione con il sensore Freestyle Libre è stata utilizzata l'interfaccia SPI di Arduino.

**Serial Peripheral Interface:** Lo standard Serial Peripheral Interface (SPI) è un bus di comunicazione sincrono, con architettura master/slave. Il suo funzionamento si basa su piccoli registri (con dimensioni generalmente da 8 bit) posizionati uno sul dispositivo Master e uno per tutti i dispositivi slave utilizzati. Tramite lo "shift" dei bit di tali registri e un collegamento che permette di considerare i due registri come se fossero contigui l'uno con l'altro, si instaura la comunicazione. Si considera quindi un unico registro da 16 bit, dove l'ultimo bit è collegato con il primo. Lo shift tra i registri avviene grazie alla sincronizzazione della comunicazione, ottenuta tramite una linea riservata al segnale di clock. L'interfaccia hardware di SPI è composta da 4 pin principali:

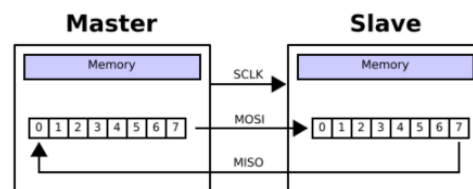


Figura 4.4: Master/Slave

- **SCLK** (Serial Clock): la scheda genera il segnale clock che passa per questo canale e serve per la sincronizzazione dei canali usati per lo scambio dei messaggi

- **MISO** (Master Input Slave Output) : serve per rendere il primo bit del registro Master contiguo all'ultimo bit del registro slave
- **MOSI** (Master Output Slave Input): serve per rendere l'ultimo bit del registro Master contiguo al primo bit del registro slave
- **SS** (Slave Select): serve per la selezione dello slave desiderato

Nella funzione `Setup()` vengono definiti i pin dedicati alla comunicazione SPI tramite le funzioni della libreria dedicata "SPI.h" e viene inviato un impulso al modulo BM019 per inizializzarlo nella modalità SPI. Viene inoltre inizializzata la comunicazione seriale per poter inviare i dati all'ESP32 dopo averli elaborati e viene inizializzato infine lo schermo LCD per poter stampare a video il dato glicemico attuale.

La funzione `Loop()` si basa sul controllare ogni volta lo stato del sensore (salvato in una variabile inizializzata a 0) che può trovarsi in tre configurazioni diverse:

- Stato=0: il sensore non è pronto e verrà richiamata una funzione per impostare il protocollo di comunicazione (ISO/IEC 15693); se il sensore risponde positivamente entrerà nello stato=1, altrimenti si attenderà qualche secondo.
- Stato=1: il sensore è pronto e verrà controllato se esso si trova nel range e se è possibile quindi iniziare il trasferimento di dati. In caso affermativo si passerà allo stato=2, altrimenti si rimarrà allo stato=1.
- Stato=2: avviene la lettura della memoria salvata all'interno del sensore

Una volta verificato lo Stato=2 del sensore avverrà la lettura dei dati e la successiva conversione in dati utili tramite un algoritmo reperito su internet e già utilizzato da privati per raggiungere lo stesso scopo.

Il dato attuale di glicemia appena elaborato verrà stampato a video nel monitor LCD e inviato tramite comunicazione seriale all'ESP32. Infine verrà impostato lo stato=0 per poter fare successive letture del sensore.

#### 4.2.2 Invio dei dati tramite ESP32

Le funzioni `Setup()` e `Loop()` rappresentano il core principale del codice implementato nell'ESP32.

Nella funzione `setup()` è definita la configurazione iniziale della scheda: qui, tramite la libreria "Wifi.h" e le relative funzioni che la sfruttano viene instaurata la connessione all'hotspot WIFI.

All'interno del `Loop()`, dove vengono definite le operazioni che verranno eseguite in modo ciclico, dopo aver ricevuto i dati tramite la porta seriale, vengono caricati nel file Google Sheets.

Per caricare i dati è stato utilizzato il servizio IFTTT acronimo di "If this than that". È proprio su questo principio di concatenazione che si basa IFTTT: connettendo differenti servizi e dispositivi, al verificarsi di predeterminate condizioni, IFTTT consentirà l'elaborazione automatica di azioni che, altrimenti, si dovrebbero svolgere manualmente. Ad ogni specifico evento corrisponderà un altro evento precedentemente impostato.

Dopo aver creato un account nel sito ufficiale [ifttt.com](https://ifttt.com) bisognerà impostare il trigger (THIS) che causerà l'esecuzione automatica di un'azione (THAT) anch'essa da impostare.

- **"Receive a web request"**: questo trigger agisce ogni volta che viene ricevuta una richiesta WEB.
- **"Add row to spreadsheet"**: azione che aggiungerà una nuova riga al foglio spreadsheet collegato.

La richiesta WEB verrà sviluppata nel codice dell'ESP32 tramite concatenazione di stringhe. Una volta ricevuti i dati da Arduino tramite la porta seriale, verrà richiamata una procedura. Utilizzando tale procedura che sfrutta la libreria "HTTPClient.h", viene prima creato l'URL e poi verrà eseguita la richiesta WEB tramite il metodo GET. La richiesta porterà all'aggiunta automatica di una riga nel foglio Google Sheets collegato contenente l'orario e il valore attuale di glicemia passati per parametro nella richiesta WEB.