

# Smart Water Network

Claudio Marche	-	65066
Luca Puggioninu	-	65052
Roberto Piredda	-	65055
Simone Madau	-	65045

Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni  
RETI RADIOMOBILI - A.A. 2017/18

# Sommario

<b>Lista dei componenti</b>	<b>2</b>
<b>Principio di funzionamento del progetto</b>	<b>3</b>
1. Esposizione del problema	3
2. Soluzione proposta	4
3. Procedura impiegata per ottenere la soluzione richiesta	4
<b>Assemblaggio Hardware</b>	<b>5</b>
<b>Sketch completo del codice o dei codici impiegati</b>	<b>11</b>

## Lista dei componenti

Quantità	Nome + codice	Budget
4	Flussometri da 1/2 e portata max 30 l/m ( <u>ARD09180CN</u> )	56.96 €
1	ESP8266 ( <u>P17100210</u> )	8,09 €
1	Cavo microUSB ( <u>ICOC MUSB-A-006S</u> )	4.89 €
1	Arduino MEGA ( <u>ARDUINOR2560</u> )	22 €
1	Breadboard ( <u>ARD01002CN</u> )	3.16 €
1m	Cavo elettrico Ethernet ( <u>V932144</u> )	2.85 €
1m	Tubo da giardino (186874)	5 €
10	Raccordi M-F (35877)	20 €
6	Riduttori da applicare al flussometro	9 €
7	Raccordi M per rubinetto ( <u>90320</u> )	10.50 €
1	Rubinetto 1/2 pollice	3 €
1	Raccordo a T per rubinetto ( <u>8614</u> )	3.5 €
1	Teflon ( <u>SMK-Q0316</u> )	0.60 €
1	Nastro isolante ( <u>62379</u> )	2.50 €
2	Power bank ( <u>YB6016-STGOL</u> )	20 €

# Principio di funzionamento del progetto

## 1. Esposizione del problema

Il problema principale che si intende risolvere è lo spreco d'acqua potabile in ambiente urbano, infatti l'Italia è un Paese ricco d'acqua, ma ne spreca "quantità enormi" per colpa di infrastrutture carenti, obsolete e inadeguate: le perdite di rete sono pari al 31,9%. Ciò ci costringe ad aumentare il prelievo di acqua alla fonte, impoverendo la risorsa ed esponendo alcuni territori a cronici disservizi. Servirebbero 65 miliardi di euro per recuperare il terreno perduto, rimettendo a posto acquedotti colabrodo e realizzando reti fognarie e impianti di depurazione delle acque reflue adeguati.

## 2. Soluzione proposta

Per risolvere il problema citato nel paragrafo precedente abbiamo pensato ad un sistema che monitori continuamente il flusso dell'acqua in una rete idraulica e qualora venga rilevata una perdita superiore ad una certa portata nel (nostro caso il 10%) il sistema avvisi tramite un messaggio tramite un BOT di Telegram gli addetti al sistema comunicando in tempo reale sia che si sta verificando una perdita d'acqua sia la posizione della perdita. Inoltre tutte le informazioni riguardanti i vari flussi d'acqua e le perdite saranno consultabili in una piattaforma di IoT (nel nostro caso abbiamo utilizzato ThingSpeak) e tali informazioni potranno essere reperibili anche interrogando il sistema con delle stringhe di testo inserite nel BOT di Telegram che restituirà in output le informazioni desiderate.

## 3. Procedura impiegata per ottenere la soluzione richiesta

Per ottenere la soluzione proposta abbiamo realizzato un prototipo di una rete idraulica per poter monitorare il flusso dell'acqua attraverso 4 flussometri posizionati in maniera strategica all'interno della rete. La rete è costituita dai seguenti componenti: dispositivo Arduino e un modulo ESP8266, 4 flussometri, del tubo da giardino, un rubinetto e vari raccordi. Con questo materiale abbiamo realizzato una rete idraulica per poter monitorare il flusso dell'acqua.

Nel dettaglio abbiamo un raccordo che funziona da ingresso al quale è collegato il primo flussometro ( $F_{max}$ ), a questo segue uno sdoppiatore dal quale escono due tratti di tubo che terminano con due ulteriori flussometri ( $Flux1$ ) e ( $Flux2$ ). A valle di  $Flux2$ , collegati tramite dei tubi, abbiamo un rubinetto con il quale simuleremo la nostra perdita e dopo il quale abbiamo posizionato il quarto ed ultimo flussometro ( $Flux3$ ).

Prima dell'installazione abbiamo provveduto a tarare i singoli flussometri, facendo passare un quantitativo d'acqua noto attraverso la valvola e verificando il numero di impulsi che ci venivano restituiti. Abbiamo poi sviluppato un algoritmo con cui riusciamo ad avere le informazioni per quanto riguarda la portata dei 4 flussometri e inoltre ad avere una notifica quando la perdita simulata supera il 10% della portata massima. In quest'ultimo caso viene inviata una notifica attraverso un bot di Telegram ad una chat scelta in fase di progetto. Il bot ci permette anche di richiedere in tempo reale i flussi dei singoli flussometri.

## Assemblaggio Hardware

Analizzato il problema che si intende risolvere si può passare all'assemblaggio, vero e proprio, del progetto.

Il montaggio verrà spiegato passo passo con l'ausilio di immagini; prima di tutto occorre mostrare una panoramica del materiale, precedentemente analizzato, che è stato utilizzato.

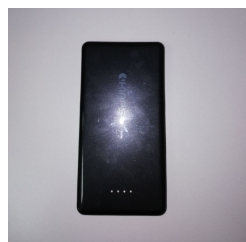


Flussometri

Tubo da giardino

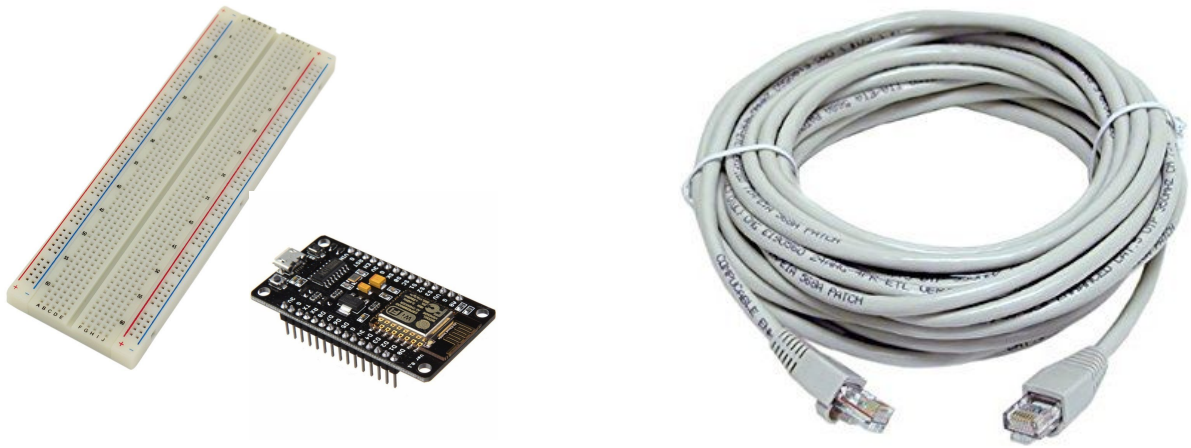


Raccordi e riduttori



Power Bank

In più, naturalmente, il nodeMCU, una breadboard e del cavo ethernet da utilizzare per i collegamenti.



Il primo passo consiste nell'assemblaggio della struttura su cui sono posizionati i 4 flussometri e su cui, successivamente, verranno inseriti i collegamenti elettrici.

Com'è possibile notare nella figura relativa al tubo da giardino, due raccordi sono già stati inseriti; tale tubo permetterà il collegamento tra un normale rubinetto dell'acqua e la struttura che andremo a creare. Le misure del tubo sono relative allo spazio di utilizzo e alla distanza dalla fonte d'acqua; si consigliano in ogni caso una decina di metri.

Innanzitutto serviranno dei segmenti di tubo, di misura più o meno variabile (in base allo spazio in cui verrà utilizzato il progetto, in questo caso non superano i 10 cm), come rappresentati in figura.



Tali segmenti saranno fondamentali per la creazione della struttura nella quale verranno connessi i flussometri; sarà indispensabile l'utilizzo dei raccordi per il loro assemblaggio.

Iniziamo con il montaggio del primo segmento:



I raccordi vengono connessi come in figura; questo segmento sarà il primo segmento che permetterà la connessione alla rete idrica tramite, come detto in precedenza, il tubo da giardino.

A tale segmento, per il calcolo del flusso principale, viene connesso il primo flussometro.



Da un lato, grazie all'utilizzo di un riduttore, viene fissato il primo flussometro, mentre dall'altro lato un raccordo, per permettere la divisione del flusso principale, della prima linea, in due linee derivate.



Nei due nuovi rami creati possono essere inseriti due nuovi segmenti (creati con le stesse modalità del primo) e in uno dei due può essere inserito il secondo flussometro, che sarà utilizzato per il calcolo del flusso d'acqua nel secondo ramo.



I due flussometri verranno in seguito connessi al nodeMCU tramite uno specifico schema elettrico mostrato successivamente nel tutorial.

Uno dei due rami viene utilizzato esclusivamente per il calcolo di un flusso, l'altro, invece, anche per la simulazione di una perdita.

In tale ramo può essere perciò posizionato il terzo flussometro, relativo al calcolo del secondo flusso derivato dal ramo principale, e, con l'utilizzo di un altro raccordo, un ulteriore ramo responsabile della perdita. A tale ramo può essere così connesso un rubinetto (grazie al quale la condizione di perdita o meno può essere arbitraria) e un ulteriore flussometro, fondamentale per il calcolo del flusso perso. Vengono

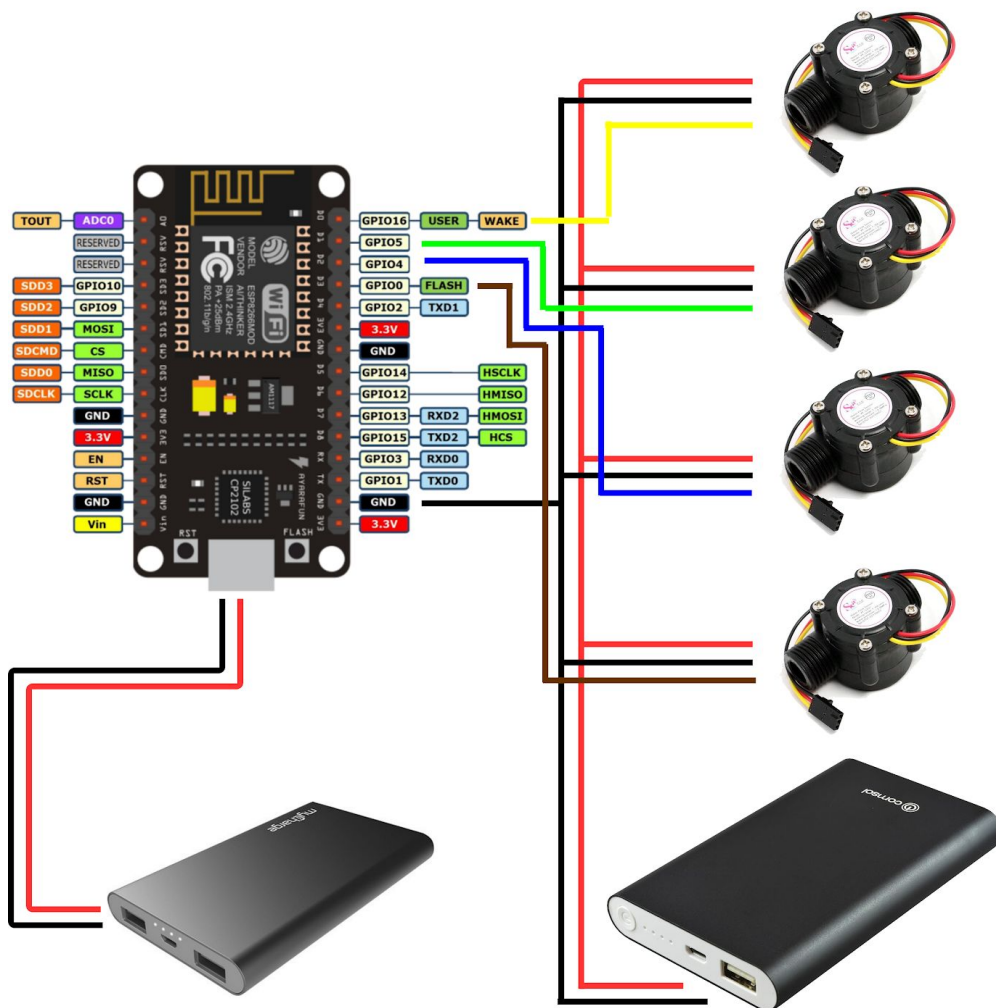
perciò mostrate le due figure relative a queste ultime connessioni.



Effettuando queste operazioni viene perciò completata la struttura comprendente i 4 flussometri. Per creare una base su cui vengono poggiati i 2 Power Bank e il nodeMCU, viene inserito un piccolo piano in polistirolo, come in figura:



Per completare l'assemblaggio, viene ora mostrato lo schema circuitale da adottare per il conclusivo collegamento dei flussometri al nodeMCU.



Com'è possibile osservare in figura, il nodeMCU viene alimentato indipendentemente da un Power Bank (comunque sostituibile da un alimentatore connesso alla rete elettrica), in maniera separata dai flussometri. Nel caso illustrato in questo progetto, si sono scelti due differenti alimentazioni per motivi logistici, in ogni caso potrebbe essere utilizzato un unico alimentatore (importante però garantire un'alimentazione con un voltaggio di 5 V in grado di erogare almeno  $15\text{ mA} \times 4$  per i flussometri, più la corrente necessaria per il nodeMCU, in questo caso sarebbe sufficiente 1 A); viene consigliato un alimentatore a 5 V in grado di erogare 2-3 A.

Di fondamentale importanza è il collegamento comune della massa (GND) tra l'alimentazione dei flussometri e il nodeMCU. Le misure effettuate sui pin di data del node, provenienti dai flussometri vengono processate come "confronto" con un segnale di riferimento, risulta perciò intuibile rendere comune tale riferimento.

Si consiglia l'utilizzo di un piccolo saldatore di precisione per i vari collegamenti tra i flussometri, utilizzando cavi Ethernet, grazie ai quali viene permesso un ottimo trasporto del segnale, una bassa attenuazione e una buona reiezione ai disturbi).

In figura viene mostrata la finale configurazione del progetto:



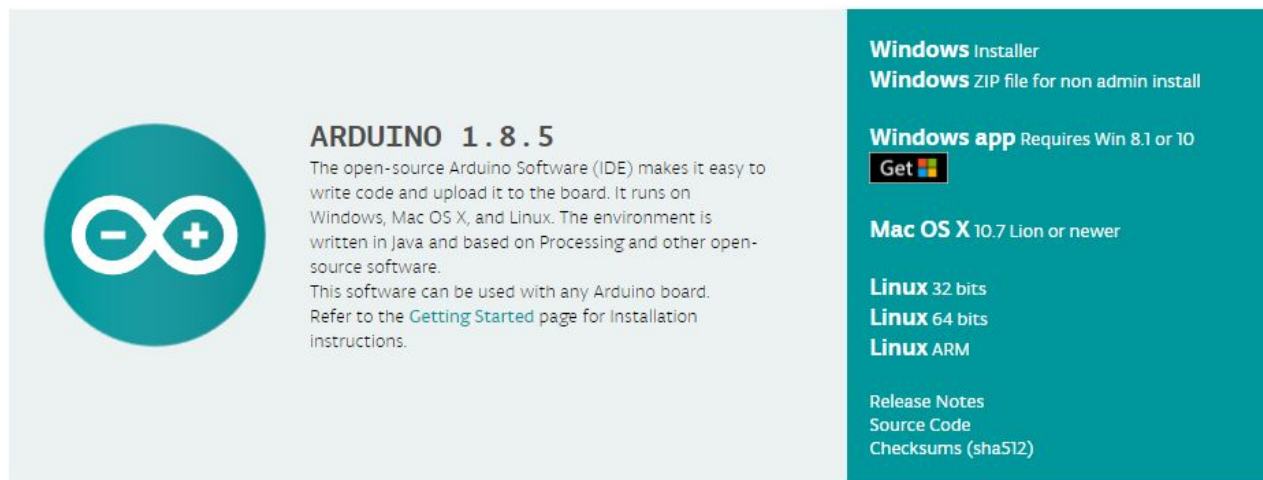
Il cavo in uscita dal node può essere connesso al secondo Power Bank o ad un alimentatore precedentemente discusso.

# Sketch completo del codice o dei codici impiegati

## 1. Arduino

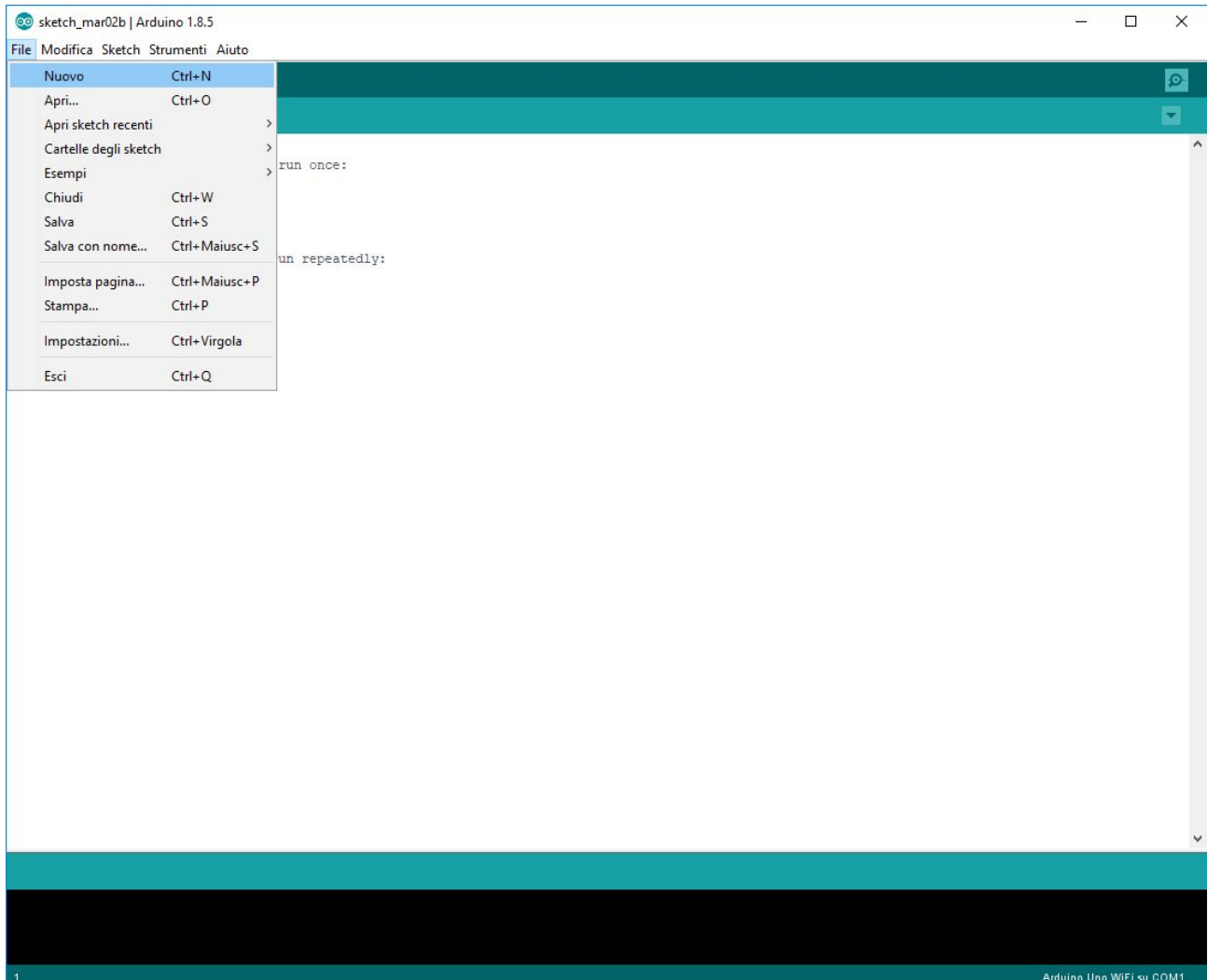
Andare sul sito <https://www.arduino.cc/en/Main/Software> e scaricare l'ultima versione del software per poter eseguire e caricare il codice all'interno del modulo NodeMCU.

### Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of this icon, the text reads: **ARDUINO 1.8.5**, followed by a paragraph describing the IDE as open-source software that runs on Windows, Mac OS X, and Linux. Below this, it states that the software can be used with any Arduino board and refers to the 'Getting Started' page for installation instructions. On the right side of the page, there is a teal sidebar with links for different operating systems: **Windows** (Installer, ZIP file for non admin install), **Windows app** (Requires Win 8.1 or 10, with a 'Get' button), **Mac OS X** (10.7 Lion or newer), **Linux** (32 bits, 64 bits, ARM), **Release Notes**, **Source Code**, and **Checksums (sha512)**.

Una volta scaricato il software, installarlo ed avviarlo. Una volta avviato creare un nuovo sketch come mostrato nell'immagine.



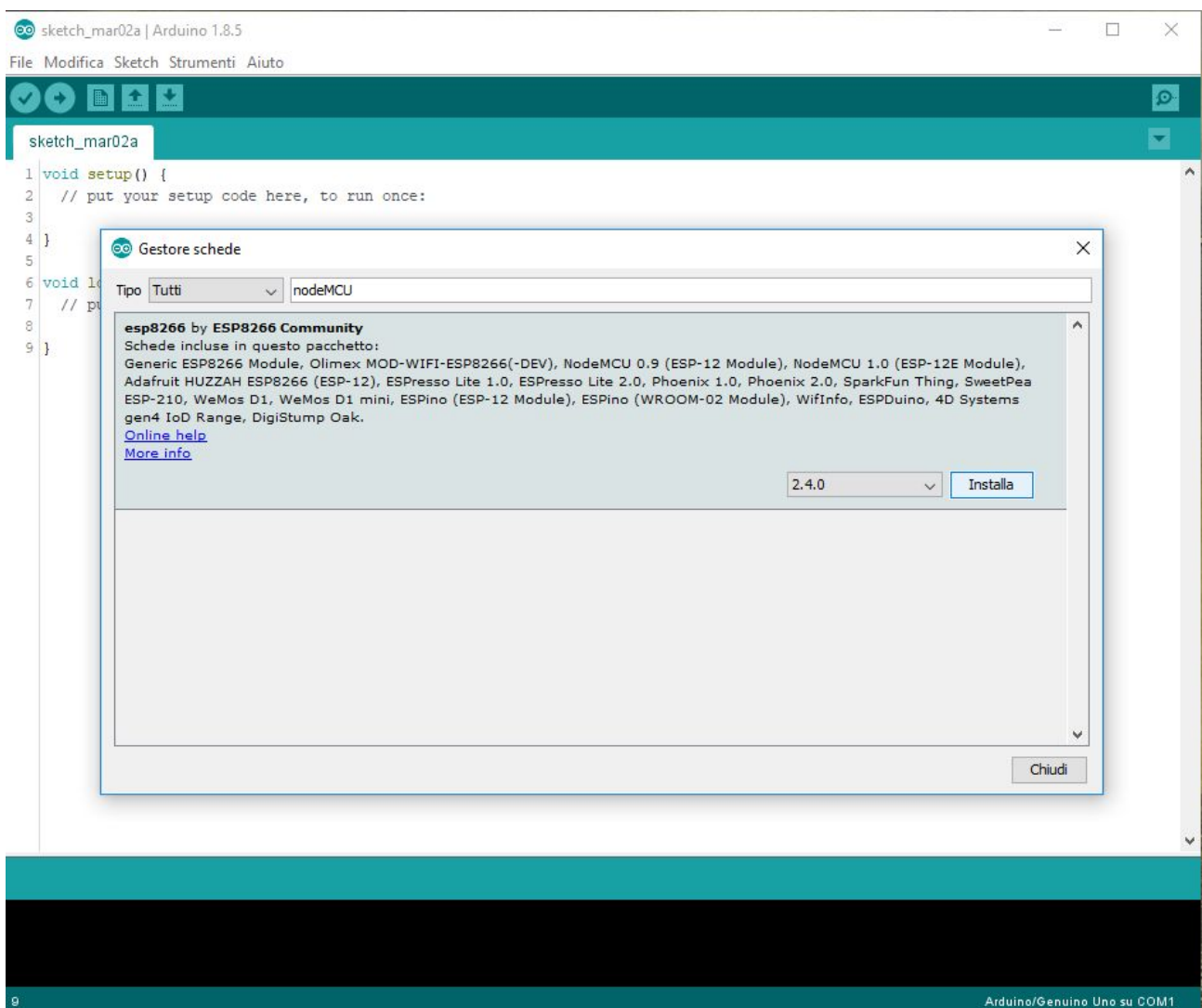
Ora abbiamo creato il nostro primo sketch. Dal menù Strumenti selezionare la board di interesse e la porta di comunicazione impiegata (Porta COM). Non appena selezionate correttamente queste due impostazioni, nella parte in basso a destra della finestra della IDE di Arduino potrete trovare qualcosa simile all'immagine precedente.



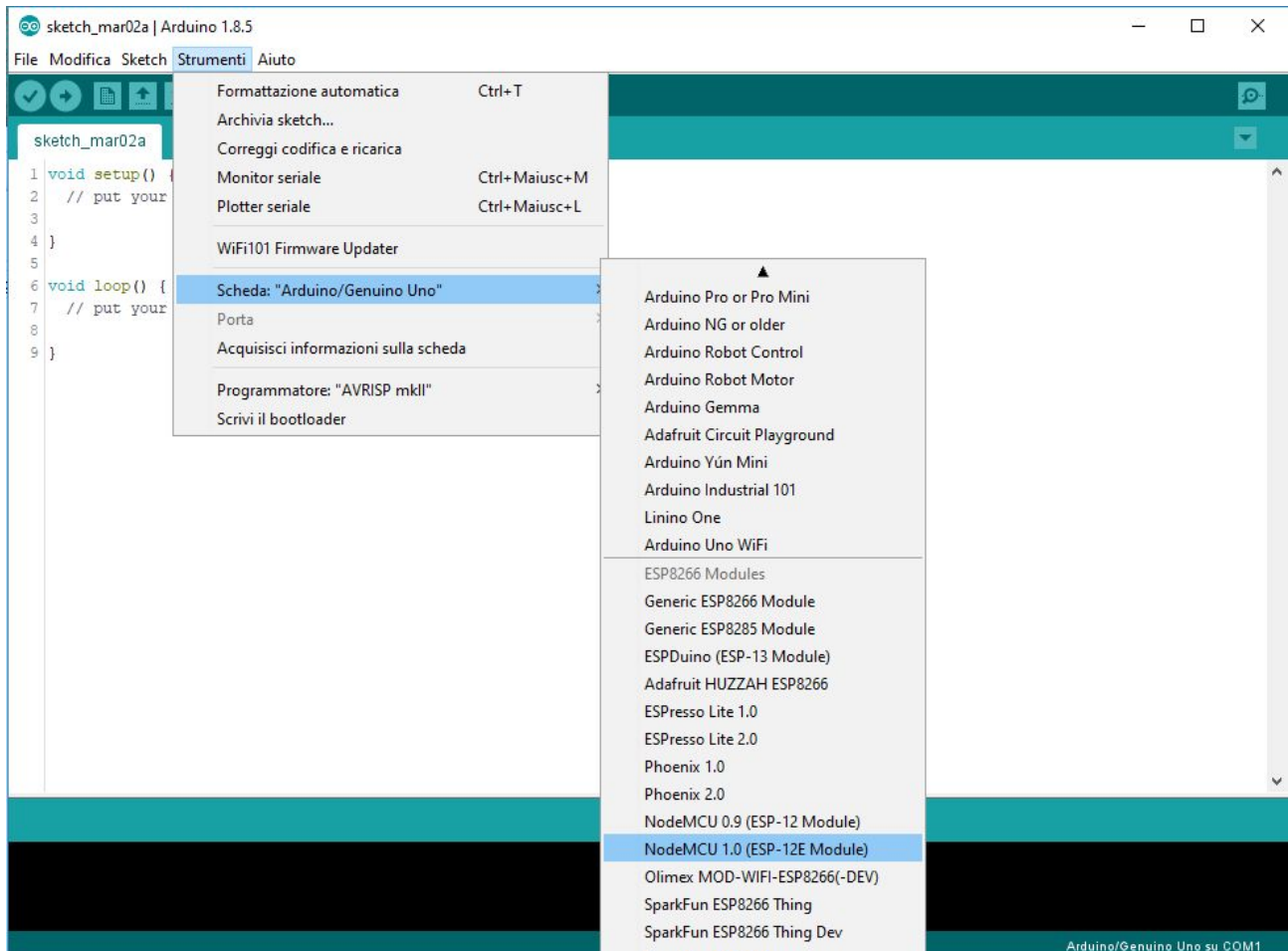
## 2. Installazione ed inclusione delle librerie

### NodeMCU

Innanzitutto bisogna scaricare la libreria del modulo ESP8266 nodeMCU. Quindi andare sul menù “Sketch->#include libreria->Gestione librerie andiamo a cercare le librerie nodeMCU sulla barra di ricerca. Dovrebbe apparire la libreria “esp8266 by ESP8266 Community” come nell’immagine seguente.



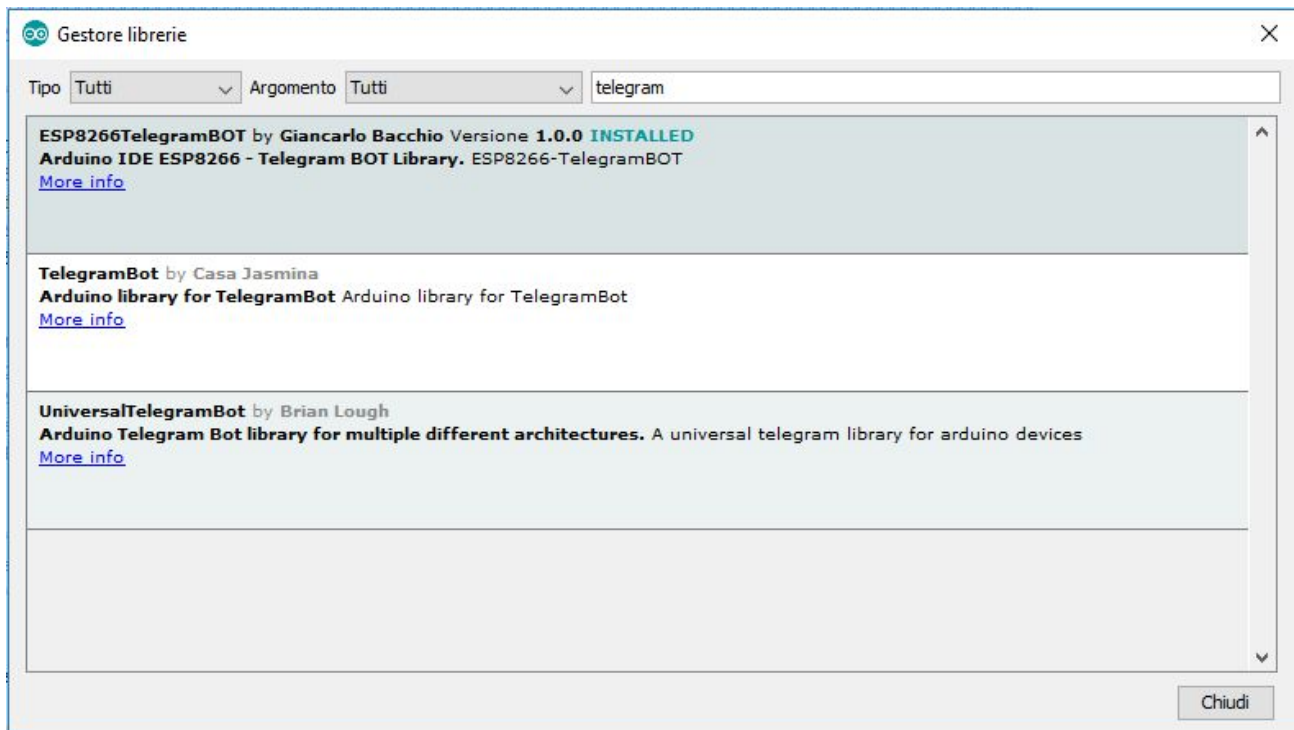
Una volta trovata la libreria cliccare sul pulsante “Installa” e il gioco è fatto. Una volta installata la libreria andremo a selezionare dal menù: Strumenti -> Scheda -> NodeMCU 1.0 (ESP-12E Module) come raffigurato in seguito.



Una volta installata la libreria del NodeMCU bisogna installare quelle relative a Telegram e ThingSpeak; i due portali principali di cui ci avvarremo per comunicare col nostro sistema e ricevere informazioni importanti.

## Telegram

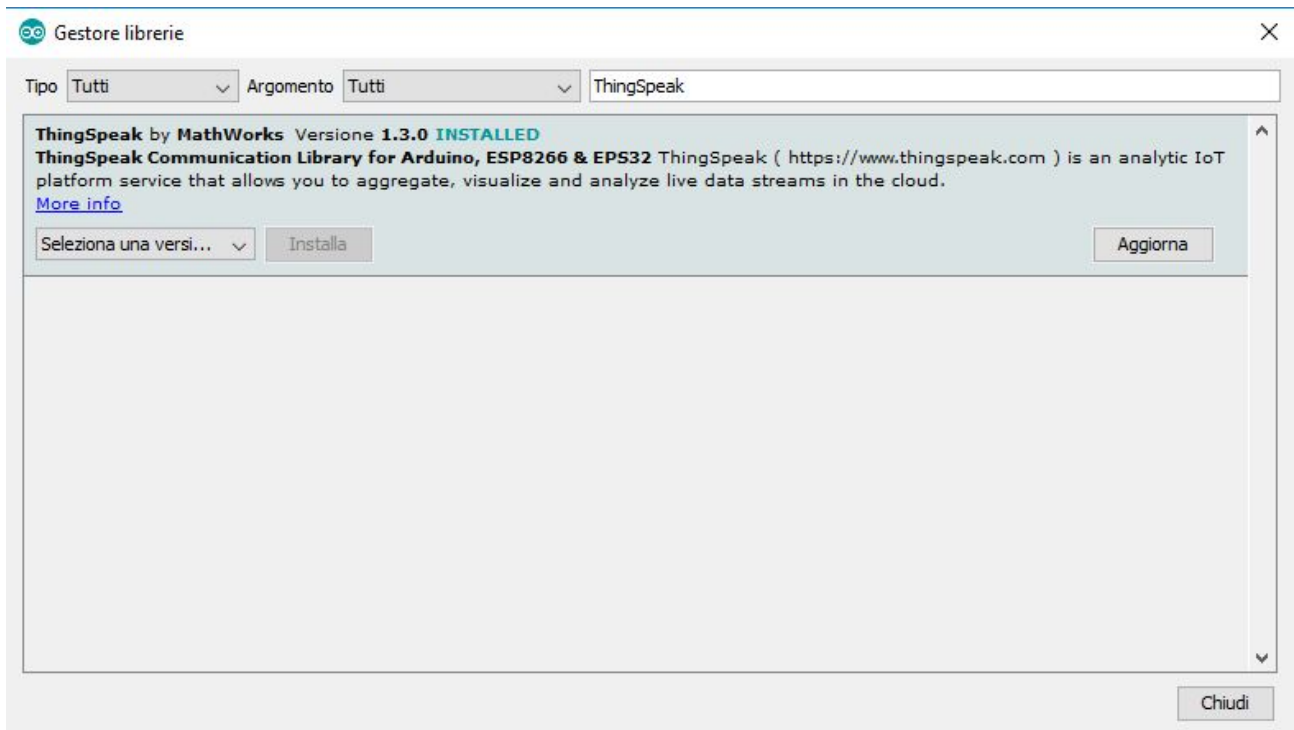
Come fatto precedentemente andare sulla gestione delle librerie e scrivere sulla barra di ricerca “telegram”. Una volta individuata la libreria “ESP8266TelegramBOT” installarla qualora non fosse già installata.





## ThingSpeak

Scrivere nella barra di ricerca: “thingspeak”, individuare la libreria ed installarla



## Includere le librerie

Le librerie di cui ha bisogno tutto il sistema per funzionare sono quindi le seguenti:

```
#include <ESP8266WiFi.h>
```

```
#include <WiFiClientSecure.h>
```

```
#include <ThingSpeak.h>
```

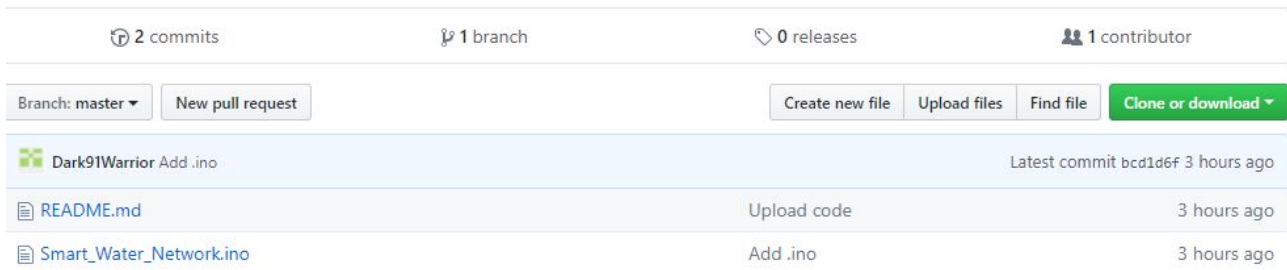
```
#include <ESP8266TelegramBOT.h>
```

Notiamo le librerie di Telegram e ThingSpeak che abbiamo scaricato precedentemente.

Adesso possiamo andare ad aggiungere le informazioni mancanti all'interno del codice quali: la connessione Wi-Fi che si intende utilizzare, i canali di ThingSpeak, il Bot Telegram ecc.

### 3. Download del codice da GitHub

Andare al seguente sito <https://github.com/Dark91Warrior/Smart-Water-Network> e scaricare tramite il pulsante Clone or Download il codice che utilizzeremo. Una volta scaricato estrarlo dall'archivio e inserire il file Smart\_Water\_Network in una cartella avente lo stesso nome all'interno dei progetti di Arduino che si trovano solitamente nella cartella "Documenti/Arduino/" (per chi usa Windows).

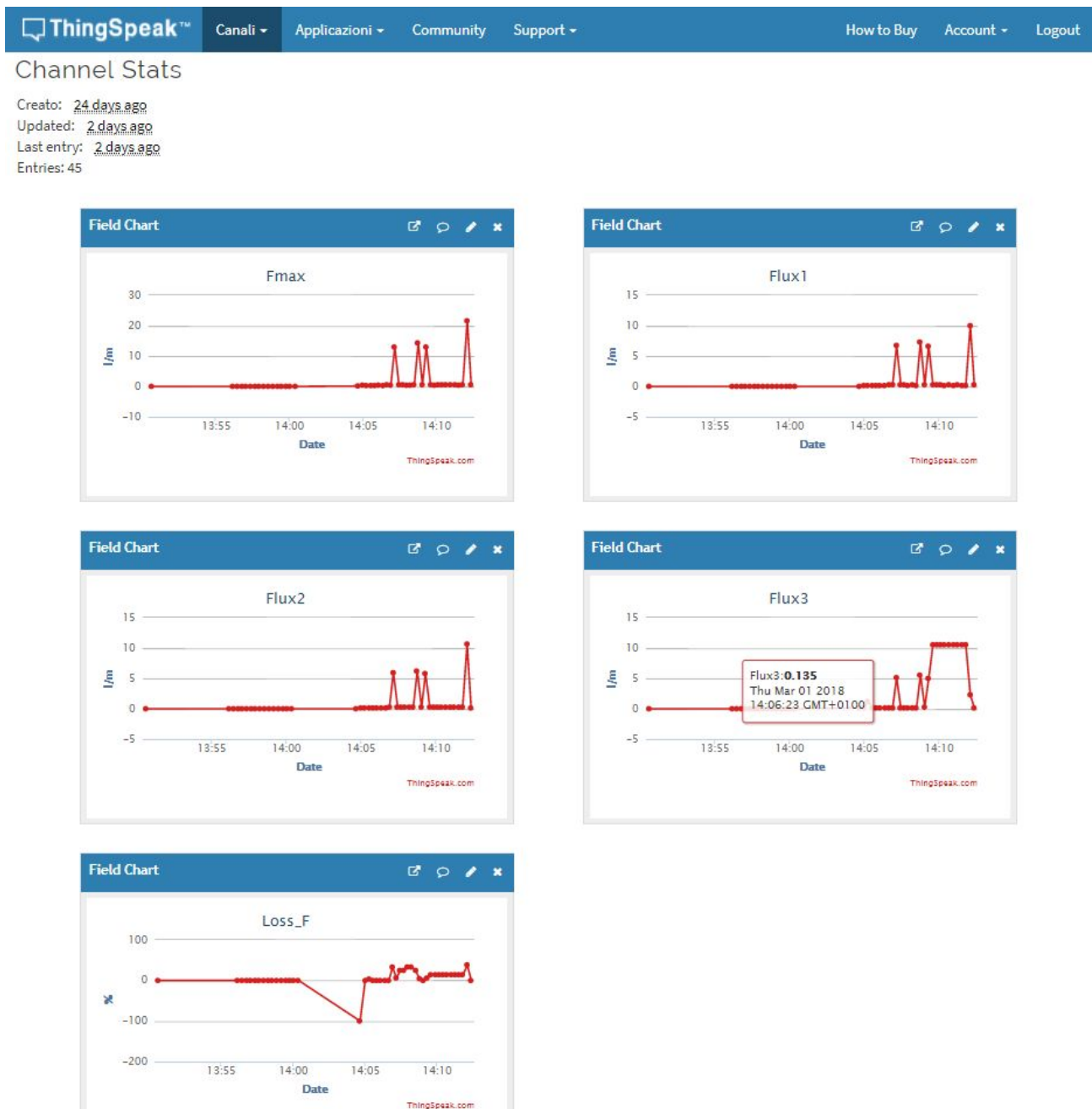


Una volta effettuati questi passaggi possiamo andare ad analizzare il codice.

## 4. Creazione dei canali ThingSpeak

Innanzitutto andate sul sito ThingSpeak <https://thingspeak.com/> e, se non lo avete già fatto, create un nuovo account cliccando il pulsante “Registrati” in alto a destra. Una volta creato l’account andate nella sezione “I miei canali” e cliccate su “New Channel”. A questo punto possiamo dare un nome e una descrizione al nostro canale oltre a selezionare i campi necessari alla visualizzazione dei dati rilevati e trasmessi alla piattaforma ThingSpeak. Nel nostro caso useremo cinque campi: Campo 1 dove inseriremo “Fmax”, Campo 2 con “Flux1”, Campo 3 con “Flux2”, Campo 4 con “Flux3” e Campo 5 con “Loss\_F”. Nella figura seguente è mostrato un esempio di come appariranno i campi. Infine clicchiamo su “Save Channel” lasciando invariate le altre opzioni. A questo punto abbiamo creato il nostro canale. Affinché possa essere impiegato correttamente, occorre che il nostro modulo NodeMCU conosca il “token” associato a tale canale. Cliccate su “Chiavi API” e memorizzate l’ID Canale e la Chiave API di scrittura. Questi due valori serviranno nel momento in cui scriverete il vostro codice e cercherete di immettere nella vostra sezione personale (identificabile attraverso ID Canale e Chiave di scrittura) i valori letti dai flussometri.

The screenshot shows the ThingSpeak channel creation and API key generation interface. At the top, the channel name "SWN" is displayed. Below it, the channel details are shown: "ID Canale:" followed by a blue box, "Author: lpuggioninu", and "Access: Private". To the right, the channel description "Smart Water Network" is visible. Below these details, there are five tabs: "Private View", "Public View", "Channel Settings", "Sharing", and "Chiavi API". The "Chiavi API" tab is selected. Under this tab, the heading "Chiave API di Scrittura" is displayed. Below the heading, there is a label "Chiave" next to a blue box. At the bottom, there is an orange button labeled "Genera Nuova Chiave di Scrittura".



Una volta creati e configurati i canali nel portale, bisogna aggiungere le informazioni del canale precedentemente citate all'interno del codice. Le variabili del codice che andremo a modificare sono quelle raffigurate nell'immagine successiva. Andremo quindi a sostituire nella variabile "myChannelNumber", l'asterisco con l'id del canale creato su ThingSpeak. Aggiungeremo poi tra i doppi apici della variabile "myWriteAPIKey" la API in scrittura del nostro canale. Queste due informazioni consentiranno al nostro modulo NodeMCU di comunicare con il canale creato sul portale web.

```
27 /*****
28  * ThingSpeak configuration *
29  *****/
30 unsigned long myChannelNumber = *; // Inserisci il canale di ThingSpeak che hai configurato al posto dell'asterisco
31 const char* myWriteAPIKey = ""; // Inserisci le API che consentiranno la comunicazione col nostro canale ThingSpeak
```

## 5. Creazione del Bot Telegram

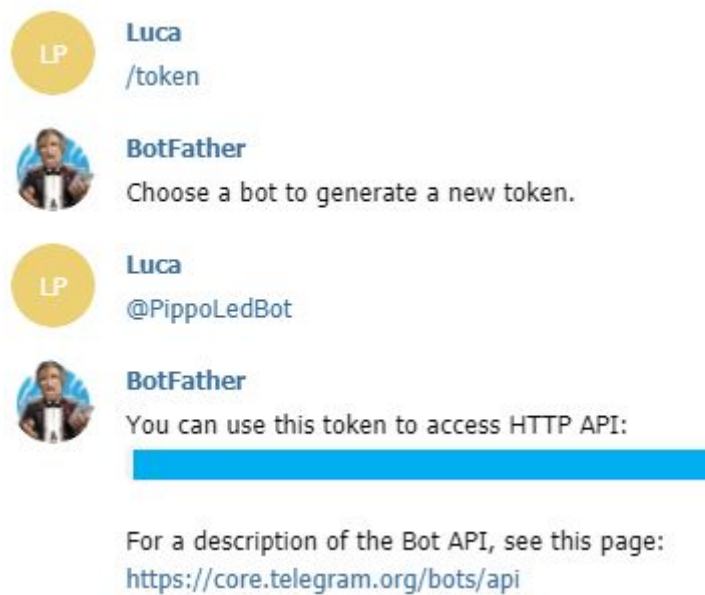
Il primo passo consiste nella creazione di un BOT e per fare ciò si utilizza un particolare BOT di Telegram. Aprire Telegram e cercare "BotFather". La sequenza di creazione del BOT è la seguente:

1. /newbot per iniziare la creazione di un nuovo BOT
2. Digitate il nome del vostro BOT
3. Digitare lo username del vostro bot (deve finire con \_bot o bot)
4. Se la procedura è andata a buon fine, il BOT è stato creato correttamente.

Le informazioni date dal BotFather devono essere memorizzate per essere poi inserite nel codice: nome del BOT, username del BOT, token comunicato dal BotFather. Il punto del codice in cui sostituire le informazioni è raffigurato nell'immagine seguente.

```
36 /*****
37  * Configurazione del BOT di Telegram *
38  *****/
39 #define BOTtoken      "" // Inserisci il token del BOT da te creato
40 #define BOTname       "" // Inserisci il nome del BOT
41 #define BOTusername   "" // Inserisci l'username del BOT
```

Una volta creato il Bot ci serviranno anche le informazioni della chat in cui vogliamo inviare. Possiamo inviare messaggi ad un utente in particolare o a un gruppo. In primo luogo avviamo la chat con il comando "/start". Quindi apriamo un qualsiasi browser all'indirizzo <https://api.telegram.org/bot<token>/getUpdates> in cui sostituiremo a <token> il codice generato in fase di creazione del bot. Per ricavare il token del Bot basta richiederlo al Bot BotFather come raffigurato nell'immagine seguente.



Una volta scritto un messaggio al Bot quindi otterremo delle informazioni scritte nel formato:

"chat":

```
{  
  "username": "yourusername",  
  "first_name": "yourfirstname",  
  "id": XXXX  
}
```

per ogni messaggio che abbiamo inviato al Bot nel sito precedentemente indicato. Per quanto riguarda l'utente, estrapoleremo l'id dell'utente o degli utenti a cui desideriamo fornire l'accesso e lo inseriremo nel codice, precisamente nella parte sottostante. Per più utenti basta dichiarare un'altra variabile (come utente2, utente3 ecc.) ed assegnargli l'id corrispondente.

```
48 String utentel = ""; // Tra i doppi apici inserire l'id dell'utente a cui si vogliono dare i permessi per comunicare col Bot  
49 // Copiando e incollando la stringa precedente e cambiando nome della variabile e id si possono gestire permessi per più utenti
```

Una volta scelto il numero degli utenti a cui fornire i permessi, andremo ad aggiungerli nella relativa sezione di gestione dei messaggi di Telegram. Qualora abbiate fornito i permessi a un solo utente potete lasciare il codice invariato. Se aggiungete più utenti basta aggiungere nella condizione if l'id degli altri utenti con degli OR in questa semplice maniera:

if (0 == id.compareTo(utente1) || 0 == id.compareTo(utente2) || 0 == id.compareTo(utente3)) e così via fino a coprire il numero di utenti desiderato.

```
103 /*****
104  * Condizione per gestire i permessi degli utenti. Per gestire più utenti, dichiarare le variabili *
105  * precedentemente e aggiungerli semplicemente con degli or. Per esempio: *
106  * if (0 == id.compareTo(utente1) || 0 == id.compareTo(utente2) || 0 == id.compareTo(utente3)) *
107  * e così via per tutti gli utenti che si vogliono aggiungere *
108  *****/
109 if (0 == id.compareTo(utente1))
---
```

Per quanto riguarda invece la/e chat in cui inviare il messaggio di avviso bisogna modificare il seguente pezzo di codice aggiungendo l'id della/e chat a cui volete inviare il messaggio. L'id della chat può essere estratto come abbiamo fatto con quello dell'utente. Una volta individuato dovete inserirlo all'interno della funzione bot.sendMessage("", "C'è una perdita!", ""); e più specificatamente all'interno delle prime virgolette dentro le parentesi.

```
80 /*****
81  * Gestione del Bot Telegram *
82  *****/
83
84 // Funzione per inviare l'avviso qualora si verificasse una perdita
85 void invia_avviso()
86 {
87     // Invio un messaggio di avviso qualora il sistema percepisca una perdita
88     bot.sendMessage("", "C'è una perdita!", ""); // Inserisci al posto delle prime virgolette l'id della chat in cui desideri che sia inviato il messaggio
89     // Qualora volessi aggiungere altre chat basta copiare e incollare la riga precedente e semplicemente cambiare l'id della chat
90 }
---
```

## 6. Configurazione della connessione al Wi-Fi

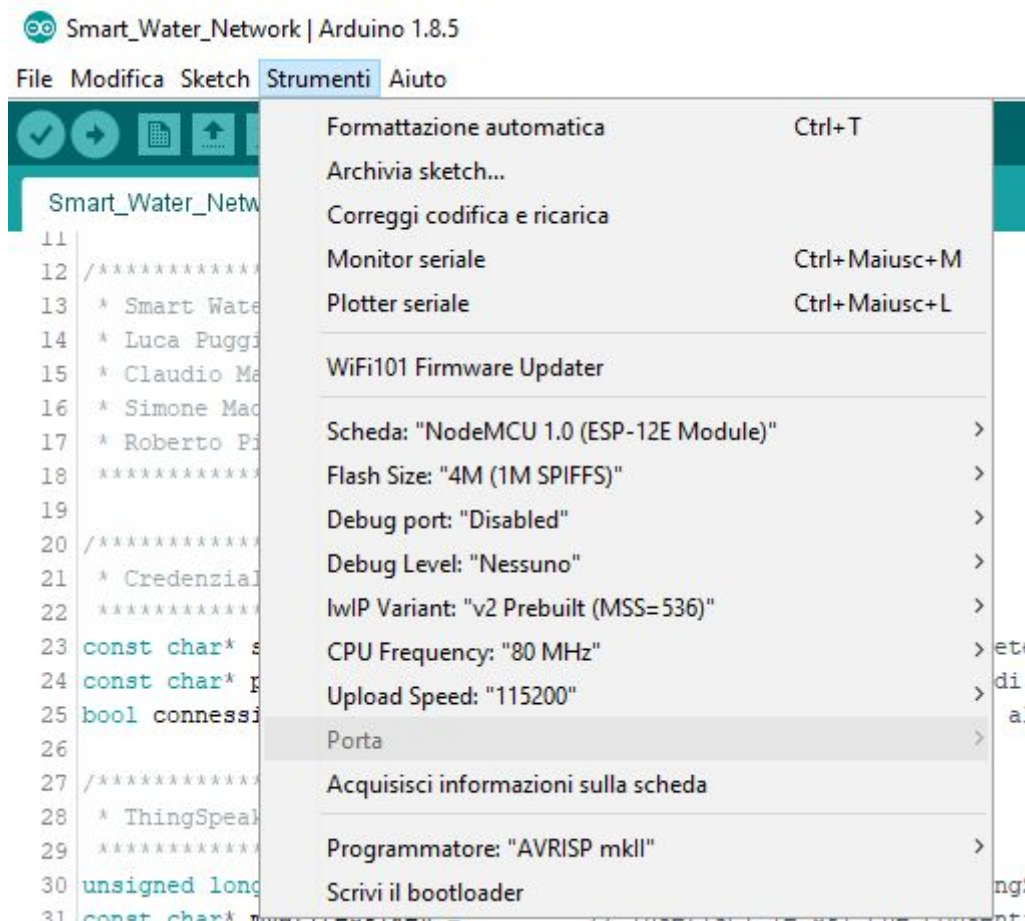
Infine bisogna configurare la connessione al Wi-Fi per permettere la comunicazione del NodeMCU con le applicazioni Telegram e ThingSpeak. Nel campo ssid tra le virgolette inserite il nome della rete Wi-Fi a cui volete connettervi e nel campo password la password relativa ad essa.

```
20 /*****
21  * Credenziali Wi-Fi *
22  *****/
23 const char* ssid = ""; // Inserisci il tuo SSID di rete tra le virgolette
24 const char* password = ""; // Inserisci la tua password di rete tra le virgolette
25 bool connessione = 0; // Inizializzo la connessione alla rete
```

## 7. Caricamento del codice sul modulo

Una volta effettuate tutte queste operazioni, collegate il vostro NodeMCU con un cavetto USB al vostro PC. Andate su “Strumenti” e dopo aver collegato il modulo dovrebbe apparirvi la porta a cui è connesso. Qualora non fosse già selezionata, selezionatela.





Tramite il pulsante “Carica” a forma di freccia in alto a sinistra caricate il codice nel modulo.





Ora dovrebbe essere tutto a posto e potrete godervi il vostro lavoro. Se avete già esperienza con questo tipo di moduli e software potete anche smanettare nel codice per modificare alcuni parametri (quali i tempi di attesa o delay ecc.). Buon divertimento!