

Rubinetto IoT v1.0

Luigi Grossi

- Introduzione

In questo progetto ho deciso di realizzare un sistema per la gestione remota di un rubinetto. In particolare, grazie a questo sistema è possibile controllare l'apertura e la chiusura di un rubinetto tramite un bot Telegram.

Il bot funziona quindi da interfaccia del sistema, e grazie ai comandi messi a disposizione permette di aprire/chiudere, monitorare il rubinetto, e gestire un programma di apertura/chiusura del rubinetto che verrà eseguito ogni giorno all'orario e per la durata impostati dall'utente.

Una potenziale applicazione di questo sistema è il controllo di un impianto di irrigazione domestico. Infatti, collegando il rubinetto ad una rete di tubi che portano a degli irrigatori posizionati per esempio in alcuni vasi o in giardino, grazie alle funzionalità messe a disposizione dal bot Telegram, è possibile impostare un programma di irrigazione quotidiano, che risulta essere particolarmente utile durante il periodo estivo quando magari ci si allontana da casa per una breve vacanza.

In questa versione iniziale del sistema è possibile impostare un solo programma quotidiano, che però può essere modificato a piacere in ogni momento.

Questa scelta può sembrare un po' limitante, ma in realtà confrontandomi con alcuni commessi di un negozio di bricolage sono giunto alla conclusione che l'utente base di un sistema di irrigazione ha prima di tutto bisogno di un sistema semplice ed immediato da utilizzare e i prodotti che in genere acquistano questi utenti permettono di impostare un solo programma di irrigazione quotidiano. In ogni caso le funzionalità del sistema che ho implementato possono essere estese a piacere, quindi magari in futuro si potrebbe pensare ad un'ulteriore fase di sviluppo che permetta di impostare un programma di irrigazione quotidiano più articolato, o addirittura permettere una personalizzazione del programma su base settimanale o addirittura mensile.

Il sistema che ho progettato è composto da una eletrovalvola collegata direttamente ad un rubinetto, la cui apertura/chiusura viene controllata da un M5STACK ATOM lite (basato su architettura esp32-pico) tramite un modulo relè.

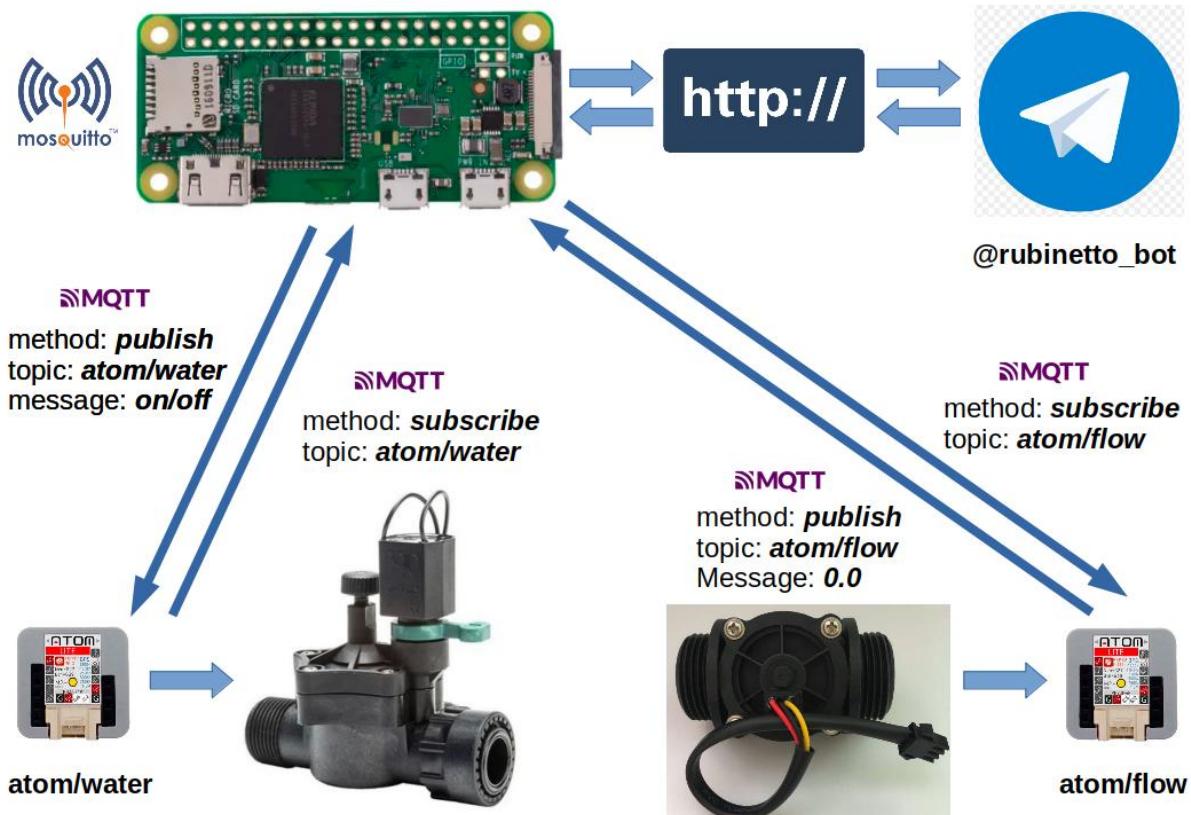
Un altro componente fondamentale del sistema è il sensore di flusso posto a valle dell'eletrovalvola che collegato ad un altro M5STACK ATOM lite invia periodicamente il valore della portata dell'acqua che attraversa il rubinetto.

Questo permette di attuare un monitoraggio costante del rubinetto, infatti è possibile controllare che le aperture e chiusure dell'eletrovalvola avvengano correttamente andando a controllare l'effettivo passaggio dell'acqua, e inoltre permette di rilevare eventuali anomalie, andando sempre a verificare la coerenza tra lo stato dell'eletrovalvola e il passaggio dell'acqua. Ovviamente eventuali anomalie vengono prontamente notificate a tutti gli utenti tramite il bot Telegram.

Le comunicazioni tra il bot e gli M5STACK ATOM lite si basano su uno dei principali protocolli in ambito Internet of Things, ovvero MQTT. Si tratta di un protocollo leggero di

tipo publish/subscribe che è particolarmente adatto per acquisire dati provenienti da sensori, oppure per comandare semplici attuatori.

Una Raspberry Pi Zero W esegue il broker (server) MQTT e il programma che gestisce l'intero sistema. L'immagine seguente mostra il sistema completo:



- Componenti

In questa sezione vengono descritti tutti i componenti e il materiale di supporto necessari per la realizzazione del progetto. Chiunque abbia intenzione di riprodurlo può utilizzare questa sezione come una vera e propria “lista della spesa”.

- Elettrovalvola bistabile 9VDC



Si tratta del componente più importante dell'intero sistema, infatti va a sostituire il ruolo del rubinetto a cui viene collegato, che dovrà restare sempre aperto, e sarà quindi compito dell'elettrovalvola regolare il passaggio dell'acqua.

E' costituita da un solenoide che quando viene alimentato dalla batteria da 9V permette di muovere un cilindretto di metallo che ha il compito di aprire o chiudere l'elettrovalvola. L'apertura e quindi il passaggio di acqua attraverso l'elettrovalvola si ottiene dando corrente con

polarità standard (cioè collegando il polo positivo della batteria a quello positivo del solenoide, e il polo negativo della batteria con quello negativo del solenoide), in questo modo il solenoide quando percorso dalla corrente si comporta come una calamita che attrae verso di sé il cilindretto di metallo.

La chiusura invece si ottiene dando corrente con polarità invertita (cioè collegando il polo positivo della batteria a quello negativo del solenoide, e il polo negativo della batteria a quello positivo del solenoide), in questo modo il solenoide si comporta come una calamita con le polarità invertite e quindi respinge il cilindretto di metallo.

Ho preferito utilizzare una elettrovalvola bistabile, perché garantisce una maggiore durata della batteria da 9V che viene usata per alimentarla, dato che basta un breve impulso elettrico (circa 200 ms) per cambiarne lo stato. Inoltre questo modello di elettrovalvola è semplice da reperire nei principali negozi di bricolage ad un costo accessibile (circa €15/20).

NOTA: per poter funzionare correttamente bisogna verificare che l'acqua che fuoriesce dal rubinetto a cui si vuole collegare l'elettrovalvola abbia una pressione statica di almeno 1 Bar.

- Sensore di flusso



Internamente è costituito da una piccola turbina sul cui asse verticale è posto un piccolo magnete. In prossimità del magnete vi è un sensore ad effetto Hall. Quando l'acqua (o un qualsiasi fluido) scorre, la turbina si muove, in questo modo il sensore ad effetto Hall rileva una variazione del campo magnetico generato dal piccolo magnete posto sulla turbina. In particolare, ad ogni giro completo della turbina il sensore ad effetto Hall invia un segnale in uscita che dovrà essere opportunamente catturato e gestito per poter

successivamente calcolare la portata del fluido che sta scorrendo.

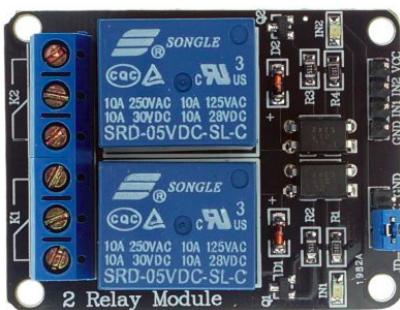
Il calcolo della portata si basa sulla relazione che la lega alla frequenza (cioè numero di giri della turbina in un determinato intervallo di tempo). Infatti, la portata è direttamente proporzionale alla frequenza:

$\text{freq} = \text{K_FACTOR} * \text{portata}$, si ottiene così che $\text{portata} = \text{freq} / \text{K_FACTOR}$ (L/min).

Il K_FACTOR è la costante di proporzionalità e dipende dal sensore utilizzato, e in genere viene indicato nel manuale o nelle specifiche tecniche del sensore. Nel mio caso il valore è compreso tra 4.5 e 4.8. Trattandosi di un sensore di flusso economico, non è possibile stabilire con certezza un valore per il K_FACTOR, ma a partire da un intervallo di possibili valori è necessario tarare il sensore facendo diverse prove e scegliendo così il valore più adatto. In questo progetto, però non sono interessato all'effettivo valore della portata dell'acqua, ma sfrutto il sensore per verificare l'effettivo scorrimento dell'acqua, quindi la precisione del valore della portata non è così fondamentale.

I sensori di flusso in genere riescono a sopportare pressioni molto alte, sicuramente maggiori di quelle che si trovano negli impianti domestici, ma nonostante questo per ridurre al minimo l'usura della turbina consiglio di posizionare il sensore di flusso a valle dell'elettrovalvola, infatti la zona a monte dell'elettrovalvola, cioè quella compresa tra il rubinetto e l'imbocco dell'elettrovalvola è una zona ad alta pressione (quando l'elettrovalvola è chiusa).

- Modulo relè a due canali



Per poter controllare l'elettrovalvola è necessario utilizzare un modulo relè.

Ho deciso di utilizzarne uno a due canali, perché mi ha permesso di gestire facilmente l'inversione della polarità della corrente di alimentazione dell'elettrovalvola.

Il modulo relè che ho utilizzato si attiva con un segnale LOW e si disattiva con un segnale HIGH.

- M5STACK ATOM lite (<https://m5stack.com/products/atom-lite-esp32-development-kit>)



Una compatta (2cm x 2cm) board di sviluppo, basata su architettura esp32-pico, quindi dotata di connettività WiFi e Bluetooth.

Inoltre è presente un pulsante, un led RGB, e diversi GPIO.

Si programma facilmente tramite l'Arduino IDE.

L'ho utilizzata per controllare l'elettrovalvola e il sensore di flusso, e grazie alla connettività WiFi è stato possibile utilizzare il protocollo MQTT.

- Raspberry Pi Zero W



Ho scelto di utilizzare questo modello di Raspberry dotato di connettività WiFi, perché ad un prezzo accessibile (circa €10) offre una sufficiente potenza di calcolo per eseguire il broker MQTT (Eclipse Mosquitto) ed il programma che gestisce il bot Telegram. Per quanto riguarda il sistema operativo, per questo progetto è sufficiente utilizzare

Raspbian lite, dato che verrà utilizzata sempre in modalità headless, e tutte le operazioni di configurazione e amministrazione possono essere eseguite, in alcuni casi, direttamente sulla scheda MicroSD (es. configurazione WiFi), oppure tramite SSH.

- Connettore JST SM 3 pin da 1 metro



Il sensore di flusso dispone di un connettore di tipo JST SM a 3 pin, ma è molto corto, per questo è necessaria una prolunga.

- Morsettiera mammut



Utilizzata per deviare i cavi provenienti dalla batteria da collegare alle uscite NC e NO del modulo relè.

- Connettori maschio e femmina



Utilizzati per collegare l'elettrovalvola al modulo relè.

- Scatole di derivazione



Usate per proteggere i circuiti.

- Batteria da 9V per alimentare l'elettrovalvola.

- Cavi USB Type-C a USB Type-A e caricatori per alimentare gli ATOM.

- Connettori jumper e filo elettrico per i collegamenti.

- Nipplo, manicotto ridotto e manicotto



Usati per collegare l'elettrovalvola al rubinetto, e per collegare l'elettrovalvola al sensore di flusso. Esistono di varie misure, e bisogna scegliere quelli più adatti al rubinetto, l'elettrovalvola e sensore di flusso che si vogliono utilizzare.

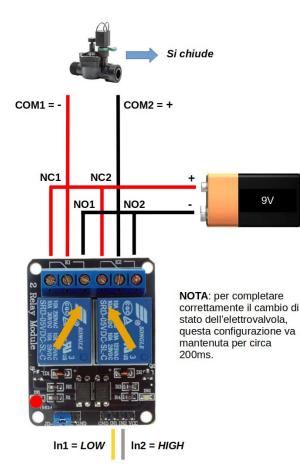
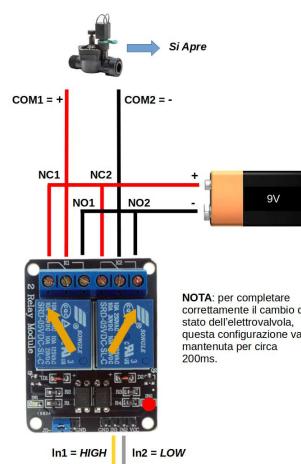
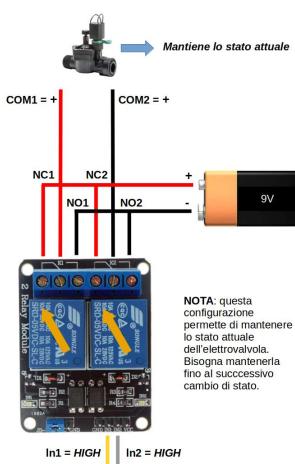
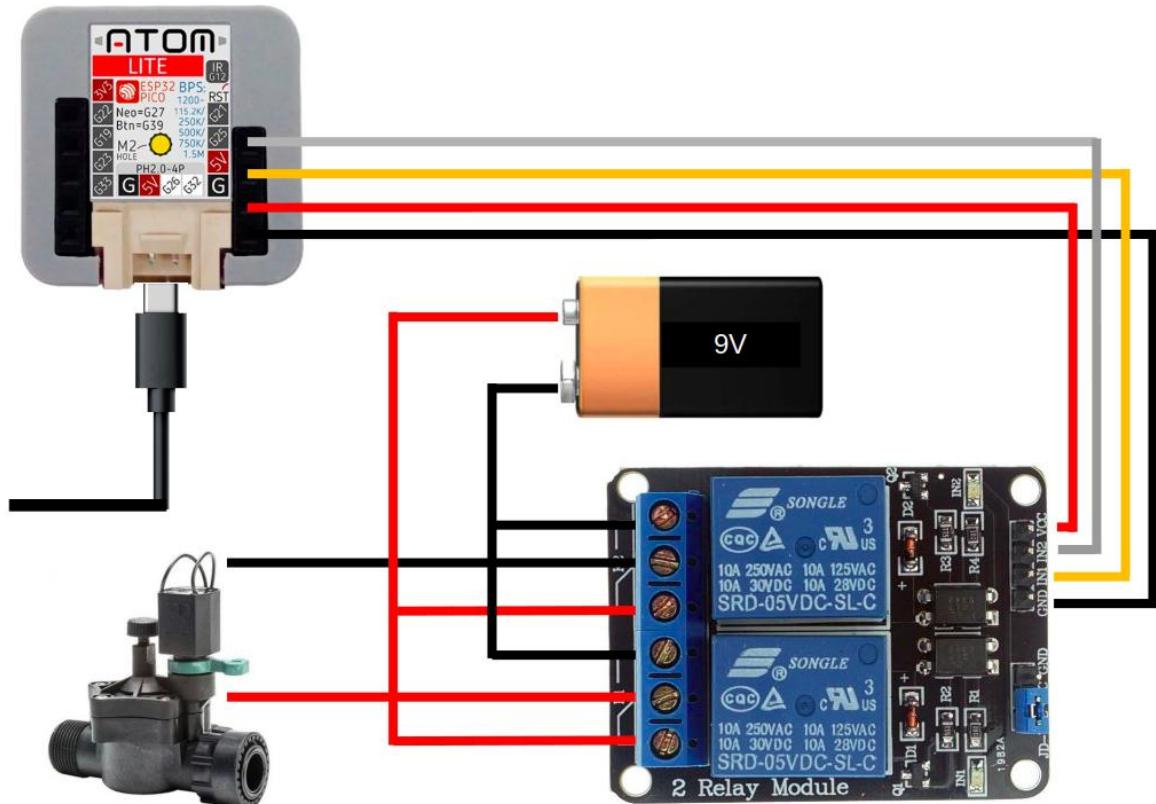
- Adattatore e raccordo rapido

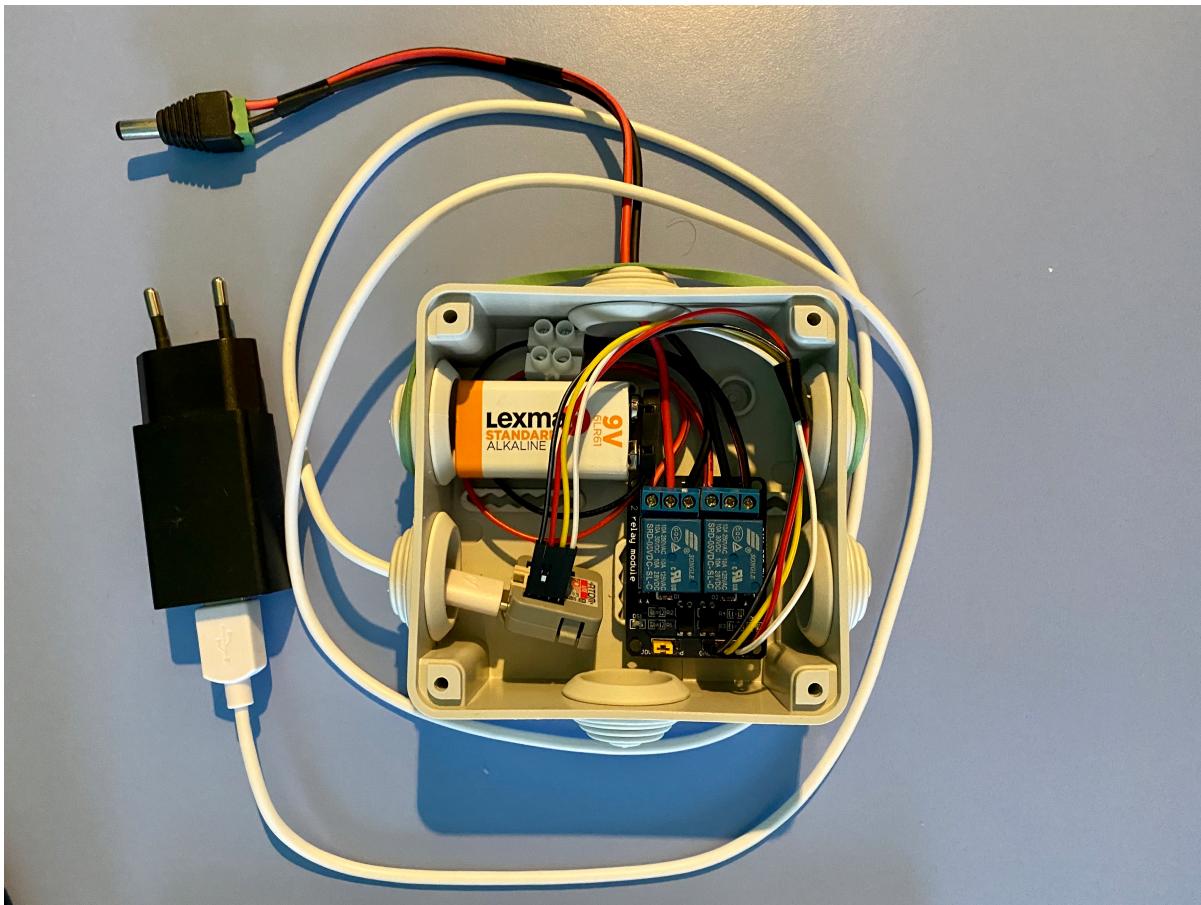


Si collegano alla parte terminale del sensore di flusso. Il raccordo rapido serve per collegare un tubo al resto dell'impianto.

- Assemblaggio

1. Circuito elettrovalvola





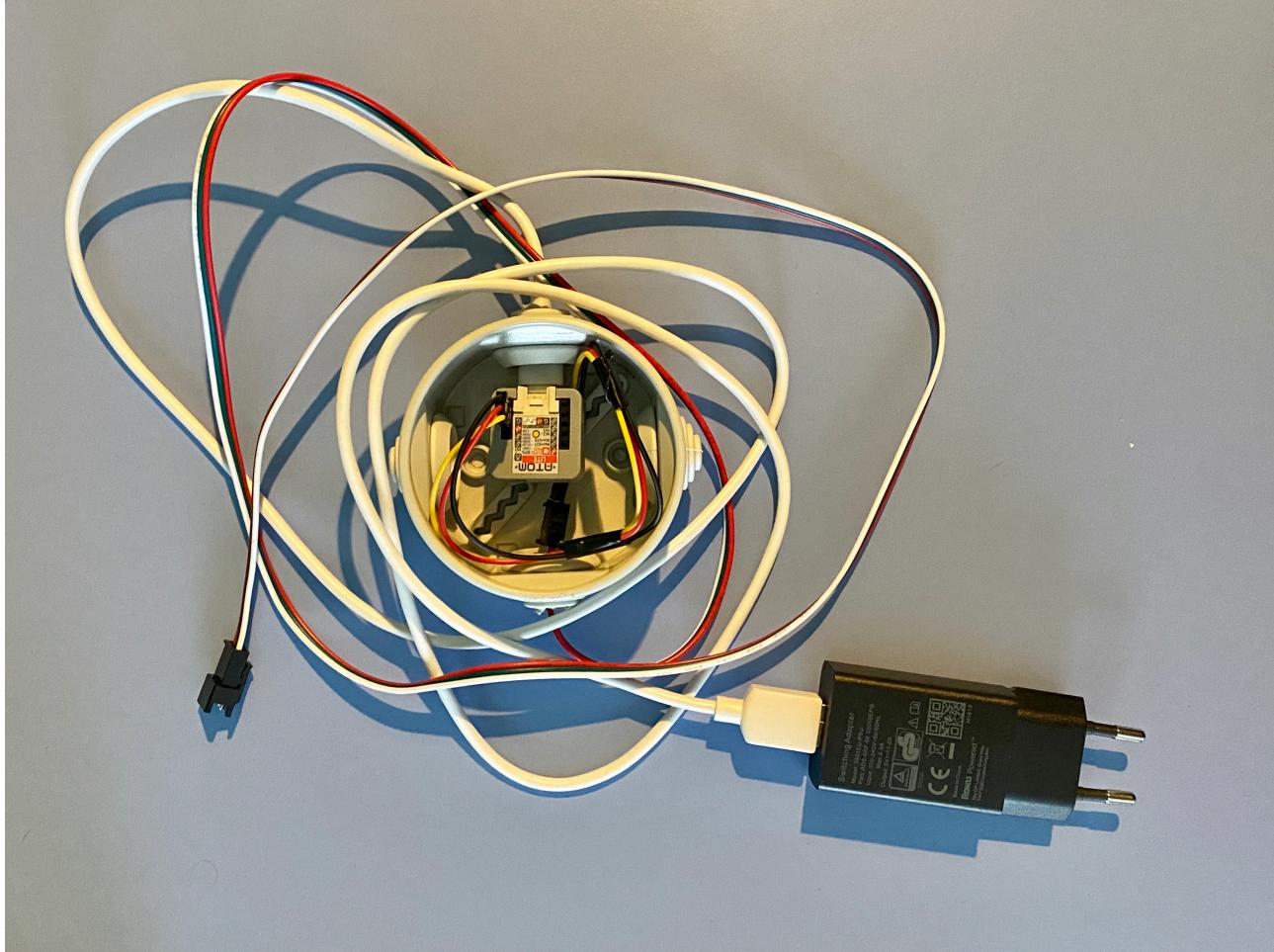
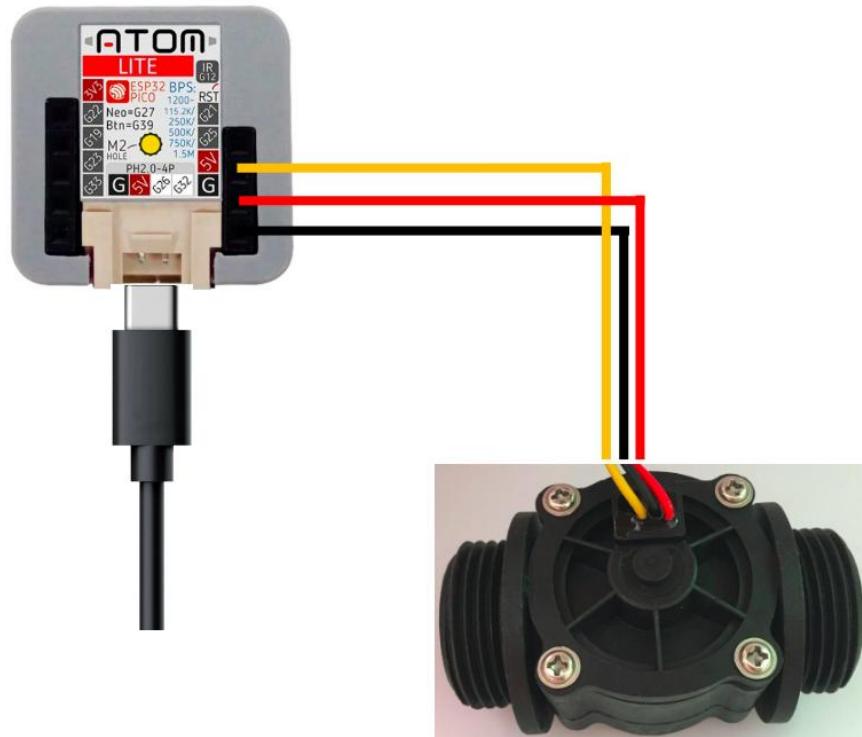
L'assemblaggio dei componenti che gestiscono l'elettrovalvola non presenta particolari difficoltà, dovrebbe essere immediato riuscire a replicare il circuito riportato nelle immagini precedenti. Per il collegamento dell'ATOM al modulo relè occorrono quattro jumper di tipo maschio-femmina.

Un passaggio importante della fase di cablaggio è il collegamento della batteria alle uscite NC e NO del modulo relè. Per poter implementare il meccanismo di inversione della polarità è necessario che i cavi provenienti della batteria da 9V vengano sdoppiati in due cavi che andranno collegati alle uscite NC (in caso si tratti del filo che parte dal polo positivo della batteria) e NO (in caso si tratti del filo che parte dal polo negativo della batteria). Per sdoppiare i cavi ho utilizzato due morsetti mammut: da una estremità si inserisce il filo proveniente da un polo della batteria che si vuole sdoppiare, e dall'altra estremità si inseriscono due fili opportunamente uniti tra di loro. A questo punto basta stringere le due viti presenti sul morsetto per ottenere un collegamento stabile.

Invece, per collegare le uscite COM all'elettrovalvola ho utilizzato una coppia di connettori maschio-femmina dotati di viti per il fissaggio dei fili. Rappresentano una scelta comoda in fase di prototipazione, perché non richiedono alcuna saldatura e sono quindi adatti anche a chi non ha particolare dimestichezza con la saldatura.

Nelle tre immagini in fondo alla pagina precedente viene mostrato come è stato utilizzato il modulo relè a due canali per invertire la polarità della corrente proveniente dalla batteria, un semplice meccanismo che permette di controllare l'apertura e chiusura dell'elettrovalvola. In pratica ognuno dei due relè viene usato come un semplice deviatore che ogni volta che viene attivato collega il proprio COM (che normalmente è collegato all'uscita NC) all'uscita NO per l'intervallo di tempo in cui resta attivo (circa 200 ms).

2. Circuito sensore di flusso



Anche questo circuito è semplice da realizzare. Per collegare l'ATOM al sensore di flusso occorre utilizzare una prolunga (che ho precedentemente indicato nella lista dei componenti necessari alla realizzazione del progetto), perché il cavo del sensore è troppo corto. Il collegamento tra la prolunga e l'ATOM si ottiene utilizzando tre jumper di tipo maschio-maschio. Una estremità di ogni cavetto jumper va inserita all'interno del connettore di tipo JST SM posto all'estremità della prolunga. Ovviamente, anche in questo caso questa è la soluzione più semplice ed immediata (trattandosi per lo più di un prototipo), ma volendo si può pensare ad un collegamento più stabile e sicuro saldando il connettore. In ogni caso la soluzione che ho adottato si è rivelata affidabile e non ho mai notato alcun segno di cedimento del collegamento.

In entrambi i circuiti che ho illustrato precedentemente gli ATOM vengono alimentati tramite cavo USB Type-C collegato ad un comune caricatore.

3. Assemblaggio rubinetto



Rubinetto 3/4"



Manicotto ridotto
3/4" → 1"



Elettrovalvola 9VDC
M1" → F1"



Nipplo 1"



Manicotto 1"



Sensore flusso
M1" → M1"



Adattatore 1"



Raccordo rapido



Per quanto riguarda l'assemblaggio del rubinetto occorre prestare particolare attenzione al collegamento dell'elettrovalvola al rubinetto. Si tratta di un vero e proprio punto critico, infatti per il corretto funzionamento dell'intero sistema il rubinetto dovrà restare sempre aperto e di conseguenza quando l'elettrovalvola sarà chiusa (cioè per la maggior parte del tempo) il tratto che la collega al rubinetto tramite il raccordo sarà sottoposto ad un'alta pressione che potrebbe causare perdite (sprecando così acqua) e di conseguenza perdere la pressione necessaria (serve almeno 1 Bar) per aprire correttamente l'elettrovalvola quando richiesto. Per questo è fondamentale stringere adeguatamente i vari collegamenti ed utilizzare il teflon per sigillare la filettatura del rubinetto e dell'ingresso dell'elettrovalvola, oppure (come nel mio caso) è possibile utilizzare dei manicotti già dotati di guarnizioni interne.

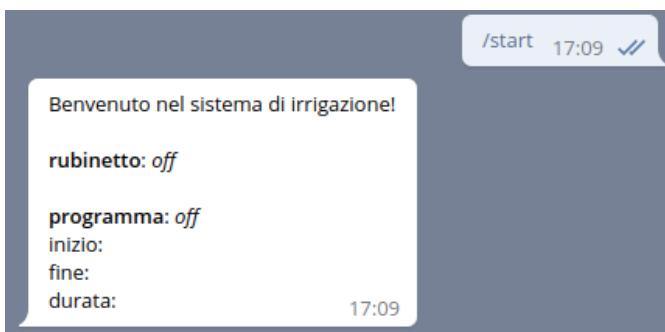
Consiglio di iniziare la procedura di assemblaggio collegando il manico ridotto al rubinetto, così da poter esercitare una certa forza senza danneggiare l'elettrovalvola. Sperimentalmente ho notato che è proprio questo il punto più vulnerabile, infatti quando sotto pressione l'acqua tende proprio ad uscire dall'alto tramite eventuali spazi tra la filettatura del rubinetto e quella del manico.

E' fondamentale dotarsi di una coppia di pinze a pappagallo per poter esercitare la giusta torsione e fissare così adeguatamente i vari componenti.

- Bot Telegram (@rubinetto_bot)

Il bot rappresenta l'interfaccia del sistema, infatti è l'unico modo che l'utente ha per interagire con il sistema. Il bot che ho creato mette a disposizione sei comandi (oltre al comando /start comune a tutti i bot) raggruppati in due categorie:

/start: avvia la comunicazione con il bot e mostra lo stato attuale del sistema.



/rubinetto: visualizza lo stato attuale dell'elettrovalvola.

/apri_rubinetto: apre l'elettrovalvola.

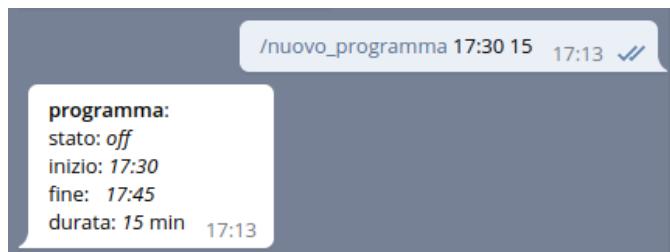


/chiudi_rubinetto: chiude l'elettrovalvola.

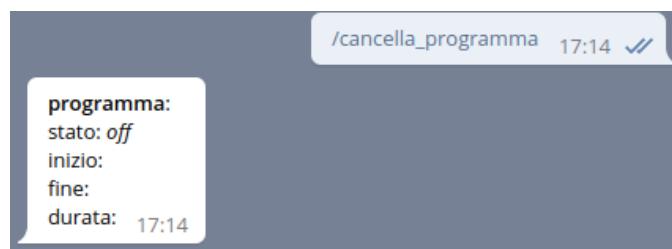


/programma: visualizza il programma impostato.

/nuovo_programma hh:mm durata: imposta/modifica un nuovo programma che verrà eseguito ogni giorno all'orario e per la durata impostati (in questa versione è possibile impostare un unico programma quotidiano, che potrà però essere modificato al bisogno).

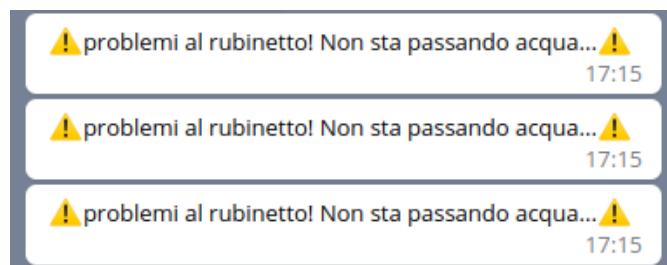


/cancella_programma: cancella il programma impostato.



Ovviamente per motivi di sicurezza il bot può essere utilizzato solamente da persone autorizzate, cioè persone che hanno fornito il proprio ID Telegram (un identificatore univoco che rappresenta un utente Telegram).

Il bot è anche in grado di notificare a tutti gli utenti autorizzati l'inizio e la fine di un programma, oppure eventuali problemi, come per esempio quando l'elettrovalvola è aperta, ma non sta passando acqua:



NOTA: Per creare un bot bisogna iniziare una conversazione con @BotFather, e inviare il comando /newbot che avvierà la procedura guidata di creazione del bot.

Per impostare altri parametri del bot (es. comandi) bisogna utilizzare i comandi che /set*.

- Software

Tutto il software (ampiamente commentato) utilizzato per questo progetto può essere consultato e scaricato al seguente link: <https://github.com/luigiq93/rubinetto>

- Demo (HOW TO)

In questa sezione sono riportate tutte le istruzioni necessarie per creare una demo funzionante del sistema.

NOTA: per il corretto funzionamento del sistema è fondamentale che i due ATOM e la Raspberry Pi siano connessi alla stessa rete WiFi.

1. Configurazione Arduino IDE per M5STACK ATOM e caricamento sketch

- Aprire Arduino IDE e *File -> Preferenze -> Impostazioni*
- Copiare https://dl.espressif.com/dl/package_esp32_index.json in *URL aggiuntive per il gestore di schede*.
- *Strumenti -> Scheda: -> Gestore Schede...*
- Nella finestra che compare cercare *ESP32* e cliccare su *Install*
- *Strumenti -> Scheda: -> M5Stick-C*
- Sketch -> #include Libreria -> Gestione Librerie...
- Cercare *FastLED* e cliccare su *Install*
- Cercare *M5Atom* e cliccare su *Install*
- Cercare *PubSubClient* e cliccare su *Install*
- Scaricare o clonare <https://github.com/luigig93/rubinetto>
- Tramite l'Arduino IDE aprire lo sketch *./rubinetto/atom/atom_flow.ino*
- Modificare nello sketch i parametri della connessione WiFi (SSID e password) e l'indirizzo IP del Raspberry Pi che esegue il broker MQTT
- Collegare uno dei due ATOM e caricare lo sketch
- Tramite l'Arduino IDE aprire lo sketch *./rubinetto/atom/atom_water.ino*
- Ripetere la procedura utilizzata per l'altro ATOM

2. Configurazione preliminare Raspberry Pi

Prima di poter iniziare questo passaggio è necessario avere già provveduto all'installazione del sistema operativo Raspbian lite, alla configurazione del WiFi, alla modifica della password di default, e bisogna aver abilitato SSH dato che per questo progetto si utilizzerà la Raspberry in modalità headless e quindi sarà fondamentale l'accesso tramite SSH.

Per chi non avesse dimestichezza con l'installazione del sistema operativo ed altre operazioni di amministrazione del sistema consiglio la lettura della documentazione ufficiale: <https://www.raspberrypi.org/documentation/>.

Dopo aver soddisfatto i precedenti requisiti, ed aver eseguito l'accesso tramite SSH si può procedere all'installazione del broker MQTT Eclipse Mosquitto:

- *sudo apt update*
- *sudo apt install mosquitto mosquitto-clients*

Per permettere al broker di partire automaticamente ad ogni boot:

- *sudo systemctl enable mosquitto.service*

Visto che uno dei compiti principali del Raspberry (oltre ad eseguire il broker MQTT), sarà quello di eseguire lo script Python che avrà il compito di gestire l'intero sistema, è

necessario installare un pacchetto che ci permetterà di creare un ambiente virtuale isolato dal resto del sistema in cui eseguire il nostro script con i rispettivi moduli:

- `sudo apt install python3-venv`

E' necessario anche installare git, visto che servirà per clonare il repository:

- `sudo apt install git`

(durante il primo utilizzo verrà chiesto di impostare nome utente e email utente)

3. Installazione script Python su Raspberry Pi

A questo punto si può procedere con l'installazione dello script Python che gestirà l'intero sistema. Quindi dopo essersi spostati nella home della Raspberry Pi si può procedere con i seguenti passaggi (assumiamo di essere loggati con l'utente "pi", in caso contrario sostituire tutti i riferimenti a "pi" con l'utente che si sta utilizzando):

- `git clone https://github.com/luigig93/rubinetto.git`
- Creare una nuova cartella che verrà utilizzata come ambiente di esecuzione:
`mkdir rubinetto_exec`
- Copiare i file necessari all'esecuzione dello script nella nuova cartella appena creata:
`cp ./rubinetto/raspberry/bot.py ./rubinetto_exec/`
`cp ./rubinetto/raspberry/config.json ./rubinetto_exec/`
`cp ./rubinetto/raspberry/requirements.txt ./rubinetto_exec/`
`cp ./rubinetto/raspberry/bot.sh ./rubinetto_exec/`
- Entrare nella cartella rubinetto_exec: `cd rubinetto_exec`
- Creare la cartella per i log dello script: `mkdir log`
- Creare il virtualenv per lo script Python: `python3 -m venv --system-site-packages ./venv`
- Attivare il virtualenv: `source ./venv/bin/activate`
- Installare moduli Python: `pip install requirements.txt`
- Editare il file config.json con i propri parametri
- Testare lo script per verificare che tutti i requisiti siano soddisfatti: `python bot.py`
- Controllare i file di log presenti nella cartella log per verificare l'esecuzione dello script
- Disattivare il virtualenv: `deactivate`

Questi passaggi bastano per poter eseguire lo script, ma è desiderabile automatizzare l'avvio dello script, in particolare fare in modo che ad ogni avvio/riavvio lo script parta automaticamente, così da poter avere il sistema sempre in esecuzione anche in caso di riavvi non desiderati (es. momentanea interruzione della corrente in casa).

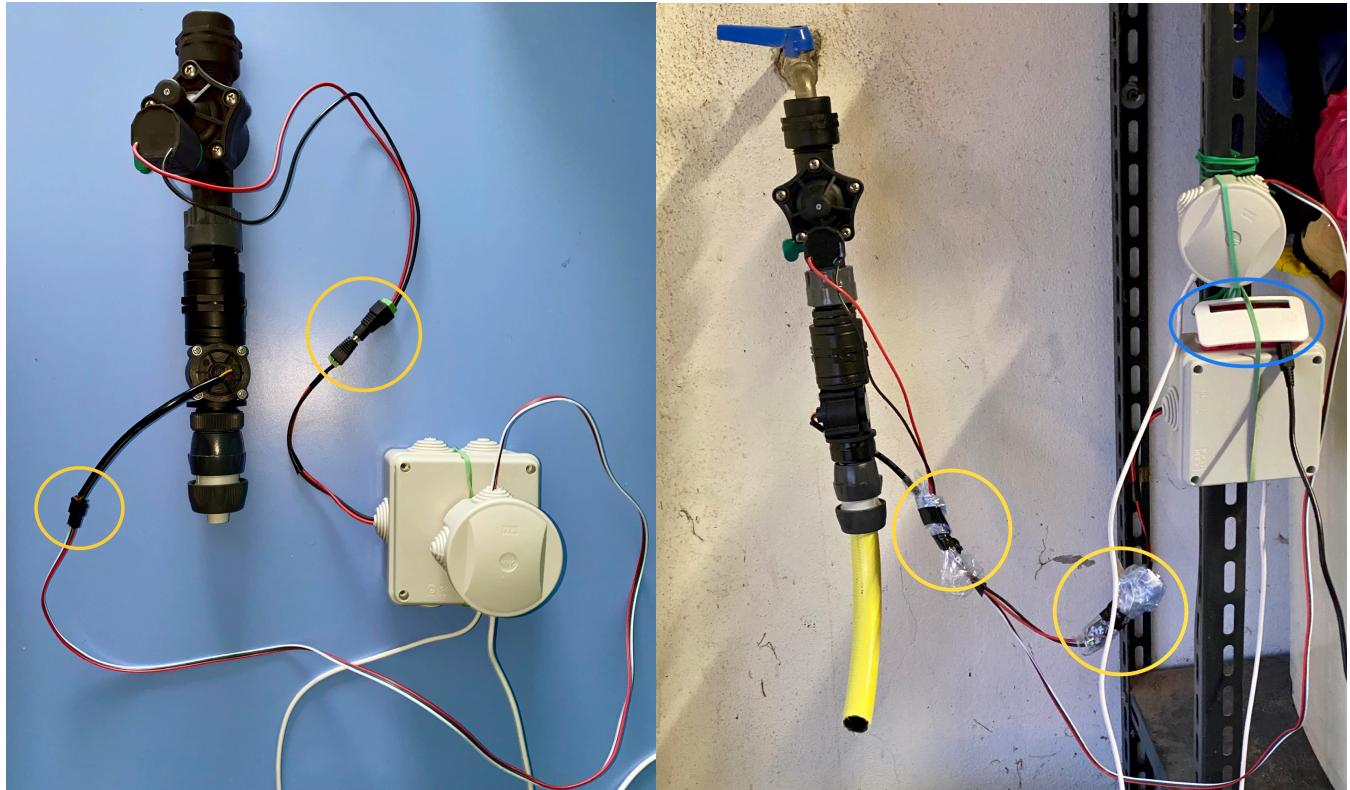
Per questo si rende necessario questo ulteriore passaggio:

- Rendere eseguibile lo script bot.sh: `chmod u+x bot.sh`
- Aprire l'editor per schedulare un nuovo task per Cron: `crontab -e`
- Inserire:
`@reboot /home/pi/rubinetto_exec/bot.sh > /home/pi/rubinetto_exec/log/cron.log 2>&1`
- Salvare e chiudere l'editor
- Controllare che il task sia stato installato correttamente: `crontab -l`

A questo punto la procedura di installazione è stata completata ed è possibile spegnere la Raspberry e proseguire con i passaggi successivi:

- `sudo halt`

4. Assemblaggio



Dopo aver assemblato il rubinetto e i circuiti come descritto in precedenza, bisogna collegare la scatola quadrata all'elettrovalvola e la scatola tonda al sensore di flusso.

L'installazione delle scatole è a discrezione e dipende molto dal contesto in cui ci si trova. Si possono per esempio fissare al muro o come nel mio caso appenderle ad uno scaffale che si trova nelle vicinanze del rubinetto. Per proteggere i collegamenti da eventuali schizzi d'acqua ho usato parte di una busta di plastica con un giro di nastro isolante. Per una soluzione più professionale è possibile proteggere i cavi con una guaina oppure un tubo di plastica per impianti elettrici.

Anche la collocazione della Raspberry è a discrezione, le uniche indicazioni sono di proteggerla all'interno di un case (nel cerchio blu è possibile vedere la soluzione che ho adottato con il case originale per Raspberry Pi Zero) e di collocarla in un punto con una sufficiente copertura della rete WiFi.

5. Avvio ATOM

A questo punto è possibile alimentare i due ATOM.

6. Avvio Raspberry Pi

E' ora possibile accendere la Raspberry.

Consiglio di controllare lo stato della Raspberry tramite: *ping indirizzo_ip_raspberry*.

In questo modo appena ci risponde possiamo accedere tramite SSH ed iniziare a controllare i log per verificare il corretto funzionamento del sistema.

7. Avvio bot Telegram

Ora è finalmente possibile controllare il sistema tramite il bot.

- Conclusioni

Dopo aver progettato, implementato e testato questa prima versione del sistema è arrivato il momento di fare alcune considerazioni.

Dal punto di vista dell'hardware mi ritengo ampiamente soddisfatto e vedo davvero poco margine di miglioramento per eventuali future versioni.

Tutti i componenti scelti si sono dimostrati all'altezza del proprio compito, in particolare sono molto soddisfatto degli M5Stack ATOM lite, che si sono dimostrati affidabili garantendomi sempre un'adeguata connettività e questo era un requisito fondamentale visto il ruolo critico ricoperto nel sistema dai due ATOM.

Un'aspetto sicuramente migliorabile in futuro è il cablaggio e l'alloggiamento dei vari circuiti, che potrebbe essere fatto in modo più professionale rispetto ad ora che è ancora in una fase prototipale.

Per quanto riguarda il software, questa prima versione rappresenta solamente il punto di partenza per uno sviluppo a lungo termine possibilmente portato avanti da più persone. Infatti, quello che in questa prima versione è mancato è una fase di beta testing.

Trattandosi di un sistema complesso le possibili combinazioni di eventi e condizioni che possono portare a degli errori sono troppe per poter essere scoperte solamente da uno sviluppatore. Ma nonostante tutto credo che questa prima versione rappresenti comunque un prototipo in uno stato avanzato che quindi può essere effettivamente usato, ma in modo supervisionato, cioè vigilando sull'effettivo funzionamento del sistema.

Per questo sconsiglio di lasciarlo in esecuzione quando ci si allontana troppo da casa, perché un malfunzionamento potrebbe causare un grave spreco di una risorsa preziosa come l'acqua. Ma in ogni caso il sistema di monitoraggio che ho implementato notificherà prontamente ogni possibile malfunzionamento.

In questa prima versione ho trascurato gli aspetti legati alla sicurezza dal punto di vista del protocollo MQTT e dell'infrastruttura di rete utilizzata.

Ho utilizzato il sistema in un ambiente fidato e gli ATOM e la Raspberry Pi non sono stati esposti all'esterno della rete domestica, per questo almeno in questa fase ho considerato accettabile questo livello di sicurezza.

Infine vorrei far notare che con circa €50 euro di materiale è stato possibile realizzare un sistema in prospettiva potenzialmente superiore alle centraline per l'irrigazione che si trovano nei negozi di bricolage e che hanno costi nettamente superiori.