

# Remo-te



Giacomo Cerino  
Studente di Laurea magistrale in Ingegneria Informatica  
Presso Alma Mater Studiorum University of Bologna

Fig. 1,2: Remo-te senza la copertura esterna  
visto da destra e da davanti

# Indice

|   |           |
|---|-----------|
| <b>Indice</b>   | <b>2</b>  |
| <b>Introduzione</b>                                   | <b>3</b>  |
| <b>Il progetto iniziale</b>                           | <b>4</b>  |
| <b>Sviluppo della soluzione</b>                       | <b>4</b>  |
| Problema dell'elettrovalvola                          | 5         |
| Problema della stampa 3d                              | 5         |
| Problema del motore                                   | 6         |
| Problema dell'attrito                                 | 8         |
| Problema della caduta                                 | 10        |
| Problema del voltaggio                                | 11        |
| Problema della comunicazione tra schede               | 12        |
| Problema della bustina di te                          | 13        |
| Interfacciamanto a ArduinoCloud                       | 13        |
| <b>Materiali e metodi utilizzati</b>                  | <b>14</b> |
| Modellazione 3d                                       | 14        |
| Stampa 3d   | 14        |
| Schede utilizzate                                     | 14        |
| Componenti elettronici                                | 15        |
| Componenti non elettronici                            | 15        |
| Disegno del circuito                                  | 16        |
| Programmazione delle schede e debug                   | 16        |
| Interfacciamento al Cloud                             | 16        |
| <b>Risultati ottenuti</b>                             | <b>17</b> |
| <b>Sviluppi futuri</b>                                | <b>17</b> |
| Telegram bot  | 17        |
| Aggiunta parametri come zucchero e tempo di infusione | 17        |
| Aggiunta di sensori                                   | 18        |
| <b>Open Source</b>                                    | <b>18</b> |
| <b>Bibliografia e Sitografia</b>                      | <b>19</b> |

# Introduzione

Il progetto è nato come una sfida per raggiungere un certo livello di comodità nella preparazione del te. A molte persone il te piace caldo e appena preparato, a me per esempio piace freddo. Il problema con il te freddo è che bisogna prima prepararlo per infusione in acqua calda e poi aspettare che si raffreddi.

Per velocizzare questo processo si potrebbe utilizzare un frigorifero o del ghiaccio. Entrambe queste due soluzioni generano problemi: mettendo il te nel frigorifero si alza di molto la temperatura del frigorifero in quanto in un ambiente a circa 4° si introduce un elemento a più di 90 °C. Inoltre viene generata della condensa che per la mia esperienza peggiora il sapore del te rendendolo più amaro. Utilizzando del ghiaccio il tè inevitabilmente si diluisce e inoltre bisogna accertarsi di disporre del ghiaccio nel freezer.

Perciò mi è venuto in mente che sarebbe comodo poter avviare con un certo anticipo rispetto al mio arrivo a casa la preparazione di una certa quantità di te. Il risultato del processo autonomo di infusione mi aspetterà a casa raffreddandosi.

Chiaramente questo progetto è adatto anche a preparare il te mentre si è sulla via del ritorno dal lavoro. In modo che arrivati a casa il te sia ancora caldo.

## Il progetto iniziale

Il progetto iniziale doveva essere un macchinario abbastanza semplice che utilizzava una pompa per riempire un bollitore, un servomotore per innescare il processo di bollitura e una ulteriore pompa per pompare l'acqua calda nella tazza o nella teiera. Inoltre era previsto un sistema di arrotolamento e srotolamento della cordicella di una bustina di te in modo da poter controllare anche il tempo di infusione.

Alla base del bollitore era prevista l'aggiunta di un "rubinetto controllato elettronicamente", come un elettrovalvola o un dispositivo da giardinaggio.

Il bozzetto del design era abbastanza compatto prevedeva di assomigliare a una macchinetta del caffè dalla cui parte bassa si poteva estrarre la tazza piena di te.

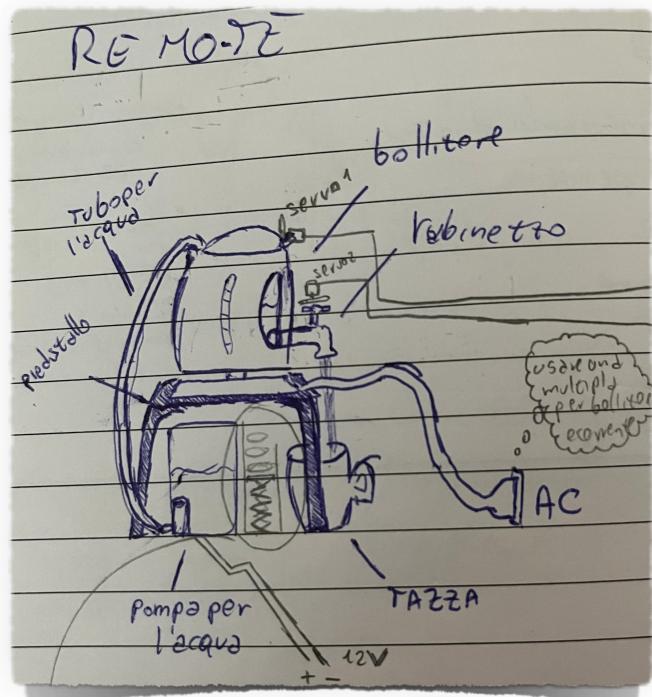


Fig. 3: Bozzetto iniziale

## Sviluppo della soluzione

In questo paragrafo descrivo la successione degli eventi che mi hanno portato poi a risolvere tutti i problemi svilcerati nella sezione problemi.e in pratica è un riassunto di tutti i passaggi e le interazioni che hanno portato al primo prototipo funzionante.

Ho cominciato comprando online una pompa provvista di serbatoio per acqua. La scelta è ricaduta su una pompa universale per tergilavatrici. Essendo pronta all'uso avrei solo dovuto provvedere all'energia elettrica. Prima di utilizzarla per preparare del te ho trattato il serbatoio con dell'aceto e del bicarbonato per rimuovere il cattivo odore che sprigionava.

Nel frattempo ho comprato un bollitore in un negozio dell'usato e mi sono recato in un negozio di componenti elettronici e idraulici Dove ho comprato due

relais e ho chiesto informazioni riguardo all'idea dell'elettrovalvola e mi hanno consigliato un posto più specializzato dove recarmi.



Fig. 4: Pompa universale per tergilicristalli con tubo e serbatoio



Fig. 5: Bollitore elettrico da 1.7l di capacità massima

## Problema dell'elettrovalvola

Una volta recatomi in loco, il commesso mi ha risposto che l'idea originaria di perforare il bollitore per aggiungere l'elettrovalvola oltre che abbastanza costosa sarebbe stata fallimentare in quanto sicuramente la tenuta stagna si sarebbe presto guastata a causa della dilatazione termica dovuta la variazione della temperatura dell'acqua che per definizione si genera dentro un bollitore.

**Soluzione:** Tornato a casa mi sono messo a pensare a una soluzione alternativa per ottenere comunque il risultato previsto dal progetto. E ho pensato a un sistema di ingranaggi in grado di inclinare il bollitore in modo da versare gradualmente l'acqua dentro un recipiente e poi ripristinare la sua posizione verticale.

## Problema della stampa 3d

Ho disegnato il sistema di ingranaggi 3D aiutandomi con un CAD e mi sono recato in un laboratorio dell'università per farmeli stampare con stampanti FDM in PLA. Purtroppo viste le imminenti vacanze estive ho potuto fare solamente una stampa e come è facile intuire il mio disegno aveva delle imperfezioni di cui mi sono reso conto solamente una volta ottenuti i pezzi.

La progettazione e la stampa di pezzi in 3d è anche detta prototipazione veloce. Per avere un prototipo funzionante di solito ci vuole qualche iterazione in modo da perfezionare il design ma in questo caso non è stato possibile.

**Soluzione:** Seppur imperfetti, i pezzi che avevo ottenuto riuscivano a soddisfare il bisogno primario per cui li avevo stampati, ovvero incastrarsi fra loro e ruotare di 90° l'uno sull'altro. La parte imperfetta era l'incastro con la parte inferiore del bollitore, ma grazie ad un adattamento con un saldatore, un Dremel, e un adattatore fatto in cartone sono riuscito a produrre un incastro che funziona. Ad ogni modo non avrei avuto la possibilità di stampare a filamento altri pezzi così voluminosi fino a settembre.



Fig. 6: Boiler, ovvero il pezzo da attaccare al bollitore



Fig. 7: Dock, ovvero il pezzo su cui si appoggerà il bollitore

## Problema del motore

Per azionare il sistema di ingranaggi in modo automatico sarebbe stato necessario l'utilizzo di un motore a cui avrei attaccato una puleggia per allungare o diminuire la lunghezza di un tirante che manteneva in equilibrio il bollitore. A casa avevo a disposizione un motore passo passo di cui non ricordo neanche la provenienza. La sfida è diventata quella di farlo funzionare in modo soddisfacente.



Fig. 8: Motore passo passo "17OM-J349-G2VS Minabea Motor" con puleggia di plastica incollata



Fig. 9: Cuscinetto a sfera e spago che inizialmente era usato come tirante.

In quel momento io avevo a disposizione solamente un Arduino UNO e perciò cominciai a sperimentare utilizzando soltanto i segnali dell'Arduino UNO. Dopo poco tempo mi sono reso conto che avrei avuto bisogno di un driver per motori passo passo. Avendo come ostacolo sempre le vacanze estive ho cercato di trovare una soluzione che non prevedesse l'acquisto di uno di questi driver.

E frugando tra i componenti elettronici che avevo a disposizione ho trovato un driver L9110 adatto per la guida di due motori a corrente continua oppure un motore passo passo e ho cominciato a scrivere il codice utilizzando quello. Dopo qualche esperimento mi sono reso conto che il problema era che il mio motore passo passo richiedeva una quantità di corrente e tensione tale che non poteva essere gestita da questo chip.

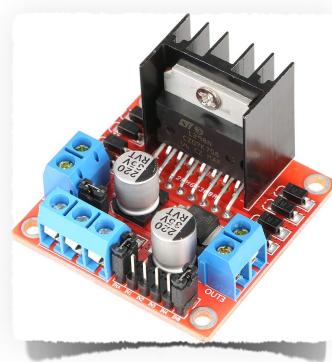


Fig. 10: Driver L298N

**Soluzione:** Mi sono ricordato a questo punto che un mio amico mi aveva portato un driver L298N. Un driver vecchio per due motori a corrente continua o un motore passo passo. Un motore che ha dei noti problemi in quanto si surriscalda fin troppo facilmente spegnendosi ma tutto sommato motore che poteva gestire (fino a un certo punto) la corrente che gli avrei fornito. Mi sono messo perciò a scrivere il codice su Arduino per provare a guidare il

motore e mi è stato molto utile guardare alcuni video su YouTube [2],[3], e in particolare video di Science Fun [1].

Come ho detto il driver del motore è soggetto a surriscaldamento e spegnimento improvviso, perciò è fondamentale fornire energia al motore soltanto per il tempo strettamente necessario, per fare ciò ho usato i due relais per creare un sistema di gestione della corrente in grado di spegnersi appena il motore non sia più necessario.



Fig. 11: Fotogramma di uno dei primi video in cui stavo provando per la prima volta il sistema del motore con la puleggia.

Si può vedere come i due pezzi stampati in 3D si incastrano tra loro, il bollitore possa essere acceso con un servo motore applicato sul retro, la puleggia srotoli e arrotoli un laccio da scarpe che avevo usato come tirante, l'acqua fuoriesca dal bollitore per finire in un recipiente.

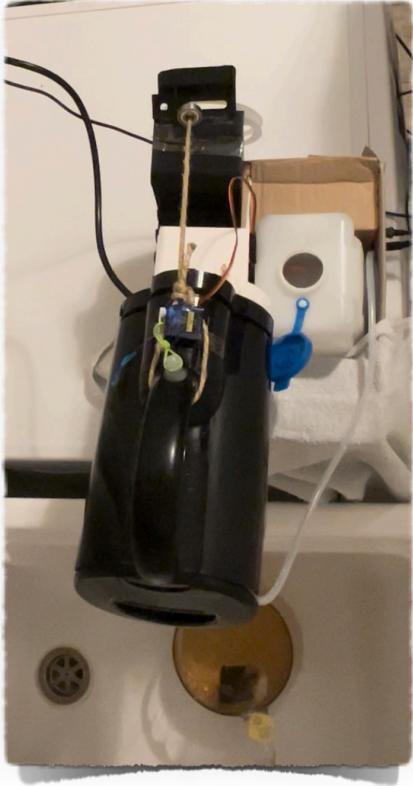
## Problema dell'attrito

Inizialmente il tirante che avevo pensato di usare era fatto di spago (ancora prima da un laccio di una scarpa). Uno dei difetti del design dei pezzi stampati in 3D era che avevano alcuni spigoli non smussati. Perciò lo spago e sfregando sullo spigolo si sarebbe presto deteriorato, questo mi ha fatto pensare al problema dell'attrito.

L'attrito rovina il tirante e stressa di più il motore richiedendo più dispendio di energia.

**Soluzione:** Ho disegnato in 3D un alloggio per un cuscinetto a sfera basandomi su un design di [Hondjog1](#) [4],[5] attraverso il quale far passare il tirante, non è il metodo più efficiente per eliminare l'attrito ma il metallo genera molto meno attrito e la componente mobile del cuscinetto agevola eventuali rotazioni del tirante. Ho detto prima non ho avuto la possibilità di stampare questo pezzo con una stampante a filamento, ma ho usato la mia stampante a resina. Stampare prototipi in resina non è particolarmente indicato perché:

- È sconsigliabile stampare grandi volumi: la resina non può avere spazi vuoti all'interno dei modelli in quanto essendo liquida rimarrebbe all'interno e non verrebbe solidificata con il processo di curing.
- Nel caso in cui si stampino pezzi che devono combaciare al millimetro con altri, è consigliabile utilizzare un orientamento verticale dell'asse di incastro in modo tale da non avere problemi di formazione durante la stampa.



(Sopra) Fig. 12: Foto dell'alloggio per cuscinetto a sfera, come si può vedere è stampato in resina e ha molti fori esagonali per diminuire molto il volume della resina utilizzata ma mantenere la stabilità e la rigidezza, inoltre i fori sono stati molto utili per incollarlo al pezzo sottostante. (Notare lo spigolo nero).

(Sinistra) Fig. 13: Fotogramma del primo video in cui ho preparato un vero e proprio te. È ripreso dall'alto. La scatola è ancora di cartone e l'alloggio per il cuscinetto a sfera è ancora fatto con materiali di fortuna per evitare l'eccessivo sfregamento

Per questo motivo il disegno che ho fatto occupa poco volume effettivo perché ha numerosi buchi esagonali. Inoltre l'alloggio per il cuscinetto a sfera è stato stampato con l'asse di incastro allineata con quella verticale e perciò è stato un successo alla prima stampa.

In aggiunta ho sostituito il tirante di spago con un tirante fatto di lenza da pesca intrecciata in modo da aumentare la resistenza.

## Problema della caduta

A questo punto avevo ottenuto un sistema funzionante di gestione dell'inclinazione del bollitore. L'unico guaio era che il bollitore in posizione di equilibrio non era soggetto all'influenza del motore, in pratica doveva essere sbilanciato in modo da poter poi essere inclinato a piacere grazie al motore.

**Soluzione:** grazie a un servo motore collocato sotto la base del bollitore si riesce a sbilanciare quest'ultimo in avanti grazie a una rotazione del braccio di 90°.



Fig. 14: Servo motore applicato alla base del bollitore



Fig. 15: Servo motore che si occupa dell'accensione del bollitore ruotando in senso orario

Ne approfitto visto che stiamo parlando di servomotori per spiegare che il sistema di accensione del bollitore è anch'esso operato da un servo motore che tira una cordicella di lenza da pesca corta in modo da azionare l'interruttore che poi ritornerà alla posizione iniziale quando il bollitore si spegnerà da solo.

**Problemi noti:** Il servo motore non è particolarmente performante e perciò non riesce a sollevare il peso del bollitore se esso è completamente aderente alla superficie del Dock e contiene più di un litro d'acqua. Per questo il motore viene sempre tenuto leggermente inclinato in modo che il servo motore possa eseguire con successo la sua funzione.

## Problema del voltaggio

Dopo aver ottenuto una versione funzionante del primo prototipo che utilizzava Arduino come micro controllore, mi sono messo a traslare il codice sulla scheda ESP8266 Node MCU 12.1. La quale essendo provvista di un modulo Wi-Fi ha la capacità di connettersi a Arduino cloud.

Sebbene inizialmente mi era sembrato un compito facile in quanto le due schede condividono il supporto per il linguaggio di programmazione di Arduino, mi sono presto accorto che il voltaggio operativo delle due schede era differente, mentre Arduino opera a 5V ed è quindi in grado di fornire segnali digitali e analogici fino a 5V, la scheda ESP8266 invece opera a 3.3V ed è perciò in grado di inviare segnali digitali analogici soltanto fino a 3.3V. Per i segnali digitali non sussiste nessun problema in quanto 3.3V è interpretato come logic high anche da componenti che lavorano 5V. Ma il codice scritto per guidare il motore passo passo utilizza segnali analogici nel range [0,5V]. Dove a 5V si raggiunge la coppia massima per il motore. Questo è importante perché durante il processo di raddrizzamento del bollitore serve più energia di quanta ne serva per farlo inclinare.

Dopo vari esperimenti ho constatato che 3.3V non erano abbastanza per comunicare al motore l'informazione riguardante alla potenza più elevata necessaria. O almeno non con quello specifico Driver.

Insieme alla scheda ESP8266 avevo ordinato anche la sua maschera (ESP8266 motor shield) capace di aumentare i pin disponibili e provvedere anche grazie a un'alimentazione esterna alla gestione di un motore passo passo. Nonostante io ci abbia speso un paio di giorni sopra, ho trovato poca documentazione a riguardo e

quella che ho trovato non mi ha permesso di ottenere un funzionamento del motore accettabile.

**Soluzione:** Dopo aver passato giorni a cercare di tradurre i segnali analogici [6] nel range [0,3.3V] al range [0,5V] ho deciso che la soluzione più pratica sarebbe stata quella di utilizzare due schede. ESP8266 per interfacciarsi al cloud e Arduino nano per controllare tutti i componenti hardware che funzionano a 5V.

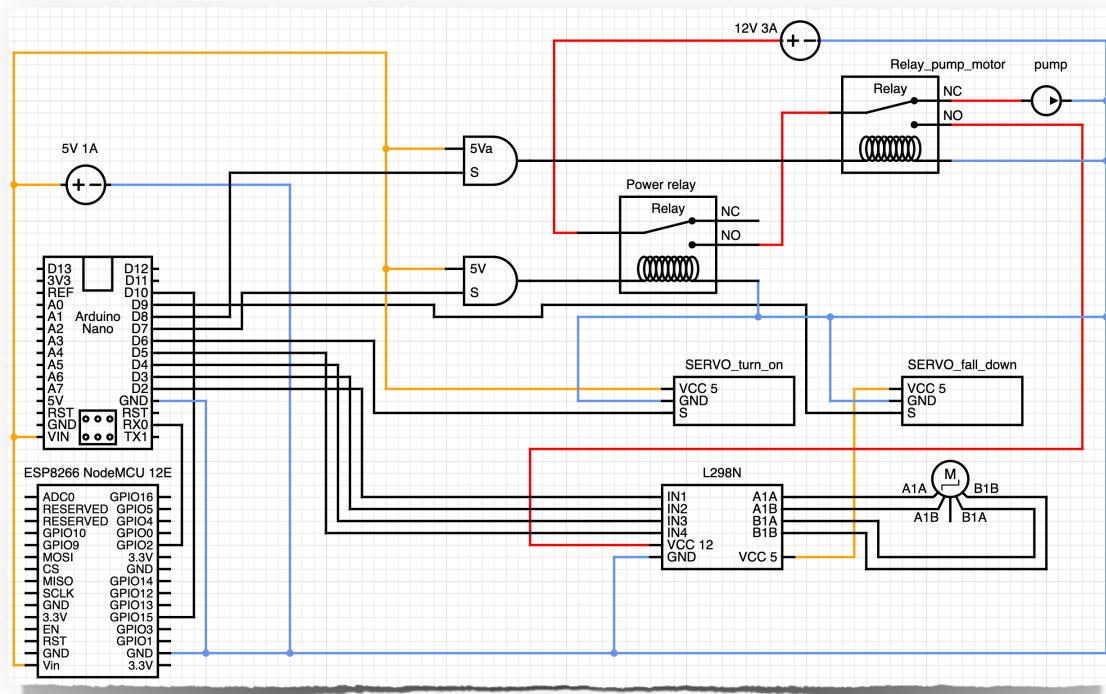
## Problema della comunicazione tra schede

Una volta deciso che avrei usato due schede si è posto il problema di come sarebbe avvenuta la comunicazione tra le due.

Ho pensato di usare dei segnali digitali basilari per inviare comandi molto semplici ma avendo studiato durante il corso i protocolli di bus e in particolare il protocollo UART ho deciso di provare con quello. Ho presto scoperto però che mentre ESP8266 ha due porte seriali UART, Arduino nano ne ha solo una.

Ho fatto svariate prove per trovare una soluzione che ponesse il sistema in grado di comunicare in entrambe le direzioni con il protocollo UART.

**Soluzione:** Dopo questi esperimenti ho capito che la cosa migliore data la mia necessità era quella di utilizzare la comunicazione UART soltanto nella direzione da ESP a Arduino nano. E visto che Arduino nano aveva tra i suoi compiti di comunicazione soltanto quello di notificare il momento in cui il te sarebbe stato pronto, utilizzare nella direzione da Arduino nano a ESP soltanto un segnale digitale.



## Problema della bustina di te

Come era stato formulato inizialmente il progetto sarebbe stato necessario fornire la possibilità di alzare e abbassare la bustina di te in modo da garantire un determinato tempo di infusione. tuttavia avendo riscontrato problemi con l'elettrovalvola inizialmente, ho dovuto concentrarmi sulla soluzione di quel problema. Una volta risolto quel problema ho pensato che avrei potuto "riciclare" il movimento del bollitore per muovere anche la bustina di te.

**Soluzione (anche se classificabile come una beta):** Creare la possibilità di fissare al bollitore una bustina di te grazie a un magnete. E aspettare il tempo di infusione prima di risollevarne il bollitore in posizione verticale. Inoltre ho aggiunto un piccolo piatto per raccogliere le gocce di te che cadranno dalla bustina.



Fig. 17: Una delle ultime iterazioni, si può notare un piattino di plastica che serve a raccogliere le gocce e una molletta nella parte superiore per fissare la bustina di te

## Interfacciamento a ArduinoCloud

Durante la risoluzione dei problemi di cui sopra, ho anche portato avanti degli esperimenti per far funzionare l'interfacciamento con l' Arduino Cloud. Ciò che ho constatato è che l'ESP8266 non è sempre collaborativo in quanto a volte rifiuta di connettersi alla rete, e a volte anche se si connette alla rete non si connette con il cloud. Nella mia esperienza la frequenza di questi avvenimenti è randomica e imprevedibile. La soluzione più comoda sarebbe quella di usare un Arduino nano 33 IoT, provvisto della funzionalità wi-fi e perfettamente integrato con Arduino Cloud.

# Materiali e metodi utilizzati

## Modellazione 3d

La modellazione 3D di questo progetto è stata fatta interamente con Autodesk Fusion 360 in quanto è un software che negli anni ho imparato autonomamente a utilizzare perciò mi sono trovato in un ambiente familiare, in cui sapevo come muovermi.

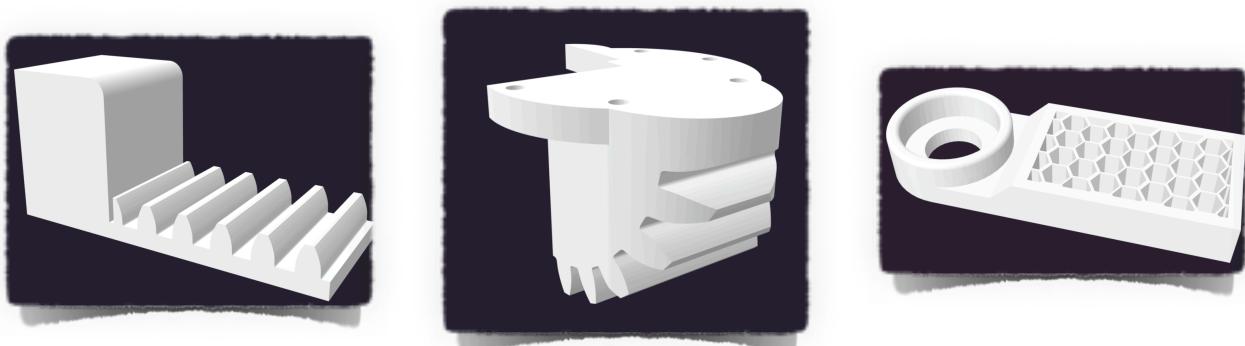


Fig. 17,18,19: Immagini dei file .stl dei tre pezzi stampati

## Stampa 3d

I primi due pezzi che ho stampato (Dock e Boiler) sono stati stampati con una stampante a filamento in materiale PLA, senza avere particolari caratteristiche di durezza, morbidezza o elasticità. Il terzo pezzo ovvero l'alloggio per cuscinetto a sfera (Bearing\_housing) invece stato stampato con una stampante a resina e le gru Saturn in resina lavabile in acqua verde chiaro e applicato successivamente al pezzo chiamato Dock con della colla a caldo.

## Schede utilizzate

Per questo progetto sono effettivamente necessarie solamente due schede:

- ESP8266 node MCU 12.1 (o scheda analoga capace di connettersi a Arduino Cloud)

- Arduino nano (o scheda analoga capace di fornire segnali a 5V),

Tuttavia per fare vari esperimenti anche con i servomotori io mi sono servito di un ulteriore

- Arduino UNO che potevo maneggiare con più facilità in quanto non era già inserito nel circuito del dispositivo completo.

## Componenti elettronici

- 2 Relays 5Vdc 10A.
- 2 Micro Servo 9g MS18.
- L298N driver per motore passo passo.
- Power supply 12v 3A.
- Power supply 5V 1A per fornire energia alle schede .
- Una generica pompa a 12V e un piccolo tubo.
- Un motore passo passo, io ho usato un “17OM-J349-G2VS Minabea Motor”.
- Una Breadboard per sorreggere tutti i componenti.
- Una presa multipla per unire tutti i dispositivi connessi alla corrente.

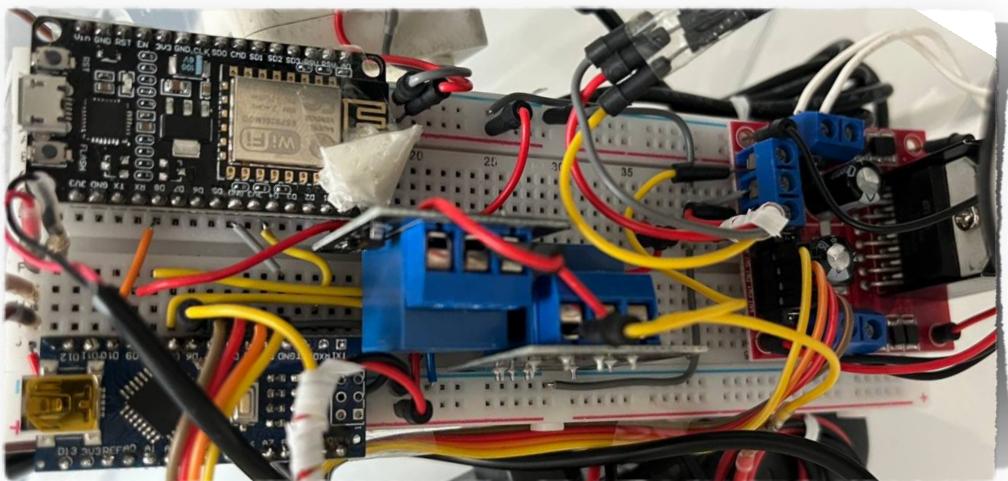


Fig. 20: Immagine del circuito visto dall'alto

## Componenti non elettronici

- Un cuscinetto a sfera
- Una scatola per contenere il tutto
- Della lenza da pesca
- Una piccola puleggia
- Nastro adesivo
- Un piccolo raccoglitore per l'acqua che sgocciola dalla bustina di te
- Una molletta
- Un magnete
- Fascette da elettricista per fissare il Dock alla scatola

## Disegno del circuito

Per disegnare il circuito mi sono servito di un sito facile e intuitivo da usare si chiama circuit diagram [7]. Qui ho trovato la maggior parte dei componenti già pronti, anche se alcuni li ho dovuti creare personalmente.

Tutto sommato non è stato difficile, però un po' dispendioso in termini di tempo

## Programmazione delle schede e debug

Per la programmazione dei microcontrollori ho utilizzato in minima parte **Arduino IDE** per poi passare a **Visual Studio Code** integrato con l'estensione per Arduino. Visual studio code ha moltissime scorciatoie da tastiera per modificare il testo in maniera veloce ed efficace. Inoltre con l'estensione di Arduino mi ha permesso di monitorare più porte seriali simultaneamente e quindi poter riscontrare in parallelo quello che succedeva in una scheda e nell'altra. Inutile dire che ha velocizzato estremamente lo sviluppo del codice. Per l'upload del codice su ESP8266, arrivato il momento di programmare l'interfacciamento con il cloud, ho dovuto usare la versione **online dell'Arduino IDE**, il quale è appositamente pensato per comunicare alla scheda in fase di upload tutte le informazioni necessarie per connettersi al cloud nonché gestire le dipendenze dalle librerie del Cloud. Previa installazione di un agent apposito per interfacciarsi alle schede.

## Interfacciamento al Cloud

Interfacciarsi al cloud sebbene sembrasse estremamente facile visti i tutorial, è stato brigoso poiché era la prima volta che utilizzavo una tecnologia simile.

Dopo averci speso qualche giorno tra tutorial, testing e debug, sono riuscito a

- rendere visibile la scheda al cloud
- fare l'upload codice
- orchestrare tutte le funzioni e le variabili in modo che funzionassero correttamente tra loro.

## Risultati ottenuti

I risultati ottenuti sono soddisfacenti da un punto di vista funzionale, in quanto il dispositivo riesce con successo a preparare dalle due alle quattro tazze di te, purtroppo il funzionamento non è garantito al 100% in quanto ci sono ancora alcuni aspetti da perfezionare come il sistema di caduta che non è affidabile al 100%.

Ciònonostante i video del funzionamento sono reperibili al link [9] insieme alle foto dei componenti utilizzati.

La replicabilità è legata a molte variabili, ma le principali sono il tipo di bollitore e il tipo di motore. Per il resto non è un processo a cui ci si può attenere passo passo, è più una guida su come io ho risolto i vari problemi che mi si sono presentati davanti.

## Sviluppi futuri

Il dispositivo attualmente dispone delle funzionalità essenziali ma sicuramente c'è un ampio margine di miglioramento delle prestazioni.

A partire da un sistema di caduta più affidabile, in quanto quello attuale non riesce a garantire funzionamento ogni volta, sebbene la problematica non si presenti così di frequente. Tra gli altri sviluppi futuri posso pensare a:

### Telegram bot

Sebbene l'applicazione IoT Remote di Arduino sia già abbastanza comoda, si può pensare all'implementazione di un BOT di Telegram che si occupa autonomamente di gestire la comunicazione con il cloud.

### Aggiunta parametri come zucchero e tempo di infusione

Si può aggiungere un parametro che gestisca il tempo di infusione con facilità, in quanto è una variabile che modifica la durata di una determinata porzione della preparazione. Cosa che avviene già a seguito della scelta del numero di tazze, perciò basterebbe replicare quel codice.

Per lo zucchero invece occorre progettare e mettere in funzione una appendice del dispositivo in grado di gestire la quantità di zucchero erogata.

## **Aggiunta di sensori**

### **Fotoresistore**

Sebbene sia una problematica che si presenta molto poco di frequente, a volte accade che il dispositivo di accensione del bollitore fallisca ad accenderlo. Perciò si può pensare all'aggiunta di un sensore come un foto resistore capace di verificare che la luce del bollitore rimanga accesa e che quindi possa essere in grado di comunicare al dispositivo di effettuare un nuovo tentativo di accensione.

### **Temperatura e ventola**

Un'altra problematica che in teoria è gestita ma in pratica non è garantito che lo sia, è il surriscaldamento del driver del motore. Esso nonostante abbia un dissipatore integrato non è garantito che riesca a mantenere la temperatura del chip sotto la soglia limite. Si potrebbe perciò aggiungere un sensore di temperatura e una piccola ventola che possa essere accesa in caso nel caso in cui essa raggiunga una determinata soglia.

## **Open Source**

Il progetto è libero e documentato interamente con tanto di aggiornamenti giornalieri su GitHub al link [8]

# Bibliografia e Sitografia

## Riguardo ai Motori passo passo

- [1] <https://youtu.be/aKxi7B4-d2U> Science Fun
- [2] <https://www.youtube.com/watch?v=aKxi7B4-d2U>
- [3] <https://www.youtube.com/watch?v=O2LWTBfQ26s>

## Riguardo all'alloggio 3D per il cuscinetto a sfera

- [4] <https://www.thingiverse.com/hondjog1/designs>
- [5] <https://www.thingiverse.com/thing:4111647>

## Riguardo ai Transistor per gestire voltaggi differenti

- [6] <https://electronics.stackexchange.com/questions/296330/how-to-drive-a-relay-from-esp8266-gpio>

## Riguardo al Design del circuito

- [7] <https://www.circuit-diagram.org/editor/>

## Riguardo alla documentazione GitHub

- [8] <https://github.com/LobinArcolungo/Remo-te>
- [9] <https://github.com/LobinArcolungo/Remo-te/tree/main/photos>