Progetto TERMOSTATO con MQTT

Obiettivo:
Realizzare un termostato composto da tre “Unità”:
- Una unità con DUE switch “Attuatori” da posizionare in prossimità delle unità Riscaldanti e/o Refrigeranti che si attivano/disattivano.
- Una unità “Sensori” da posizionare dove è più comodo/corretto il rilevamento della temperatura interna dell'ambiente. Un secondo sensore per rilevare la temperatura esterna senza influenza (per ora) sugli switch attuatori.
- Uno SmatFone che funziona da Unità “Manager” per i settaggi e la visualizzazione dati.

Dubbi e problemi:

- Usare (ESP8266 | un Raspberrypi Zero | Arduino con ETH-Shield) come client 'Attuatori' che attiva (DUE Digital Output TTL) quindi due Relays di risposta ad allarmi di temperatura 'bassa' o 'alta'?
- Usare ((ESP8266 | un Raspberrypi Zero con dongle WiFi | Arduino con ETH-Shield) come client gateway 'Sensori' che misura due temperature 'interna', 'esterna' e la data 'data' e invia i messaggi all'attuatore?
- Provare il Broker mqtt prima su PC poi su server Remoto iot.eclipse.org (o meglio Amazon AWS?)
- Sviluppi ulteriori è utile un LCD sull'unità sensori per mostrare i dati?
- Quale App mqtt usare sullo SmartFone che dovrà fungere da 'manager'?

Passo uno: Il gateway “Attuatori”

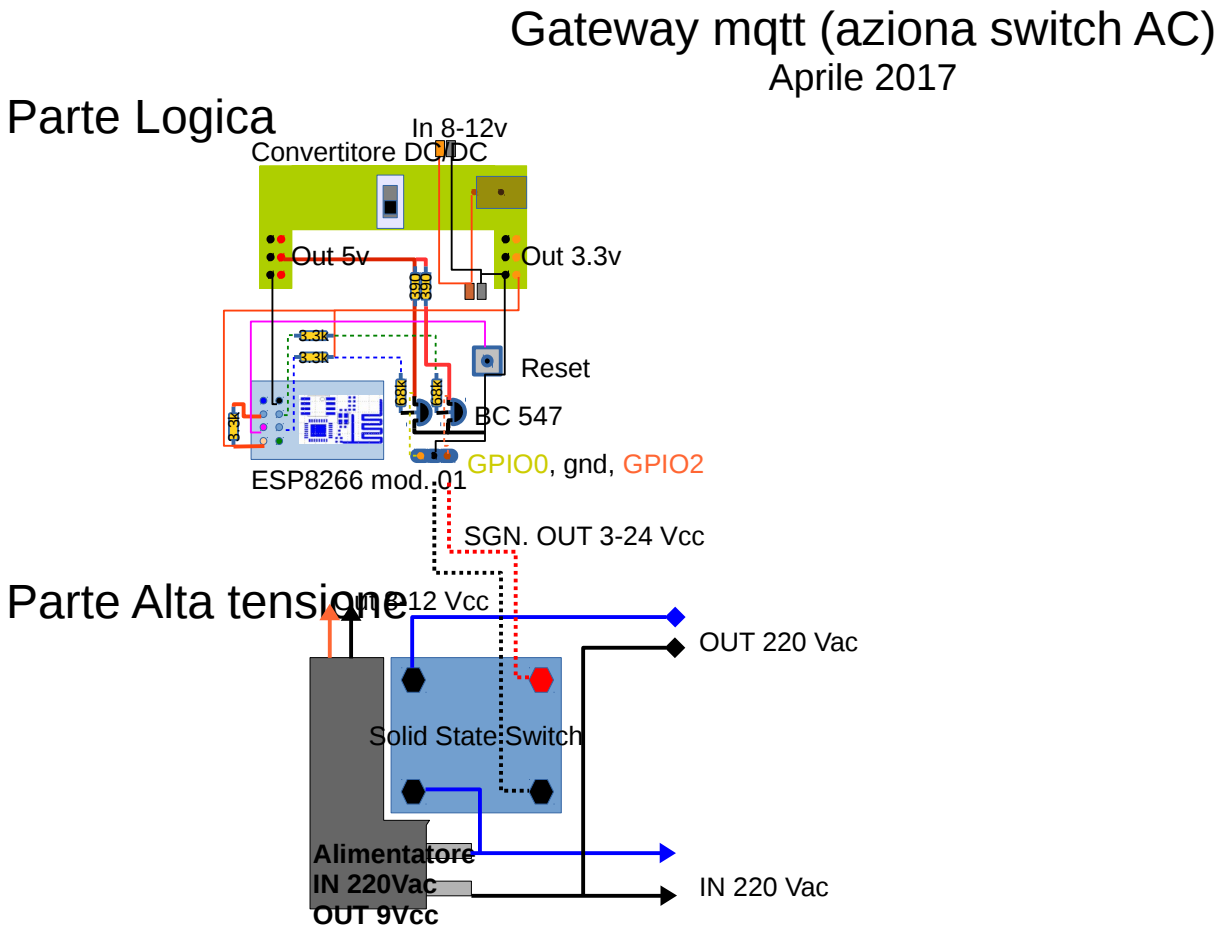
Prerequisiti di conoscenza (facili da acquisire anche per inesperti come il sottoscritto)

- Sapere come funziona il protocollo Mosquitto (Mqtt) con Publish|Subscribe
- Sapere come si programma ESP8266 mod_01 con IDE di Arduino
- minima conoscenza pratica per la realizzazione circuitale

Scelte:

- Ho optato per ESP_01 per realizzare il client “Attuatori” software ESP_01_mqtt_xxx.ino lo puoi scaricare dal https://github.com/gcupini/TINY .
- Per la programmazione di ESP01 (si usa un PC con IDE Arduino) ma è necessario realizzare un piccolo circuio ad hoc con un convertitore USD Seriale (o in alternativa una breadboard).
- I due pin digitali di cui dispone sono sufficienti per (accendere i due Led di prova e quindi gli SWITCH Bassa – SWITCH Alta)
- Per ora ho realizzato solo Switch.bassa con un relè allo stato solido pilotato con la “logica di ESP_01”
- Il Broker mqtt o lo si installa provvisoriamente sul PC in rete locale con ESP oppure si utilizza il broker remoto gratuito e aperto per le prove “iot.eclipse.org”
- Si può provare subito il Client “Attuatori” (senza realizzare il client “sensori”) con una App mqtt su smartfone. Che pubblica provvisoriamente solo i due Topic Led/bassa Led/alta.

Schema di realizzazione del Gateway “Attuatori”:



Materiali:

Convertitore DC/DC	Input 8-12vdc	Output 5 e 3.3vdc	costo: 2.0
ESP8266 mod 01			costo: 2.5
Componenti elettronici 2xBC547, 6 resistori, un pulsante, un piccola millefori su cui saldare il tutto, Spina, Presa			costo: 4.0
Un alimentatore switching 220vac to 9 vdc			costo: 3.0
Un Solid state Switch 220vac 16A (tensione di comando 3-24vdc)			costo: 3.5
App per SmatFone Android MyMQTT (grezza e semplice)			free:

Il Software per ESP_01 “attuatori” lo trovi link https://github.com/gcupini/TINY.

Le foto mostrano il “Pacco attuatori” inserito in una Scatola stampata con 3DRAG e lo schermo di un vecchio SmartFone LG senza Sim che lavora in WIFI per inviare i comandi Accendi|Spegni.
Si vedono i due Topic necessari per verificarne il funzionamento. Topic “Led/bassa” Payload 1|0 che corrispondo ad Accedi|Spegni.

Passo due: Il gateway “Sensori”

Prerequisiti di conoscenza

- Conoscere RaspberryPi e il sistema operativo Raspbian
- Programmare “da neofita” con python

Scelte:

- Dopo varie prove con RaspberryPi. RaspberryPi ZERO, CHIP computer, ho optato per CHIP per realizzare il client “Sensori” software → Chip_mqtt_xx.py lo puoi scaricare qui https://github.com/gcupini/TINY

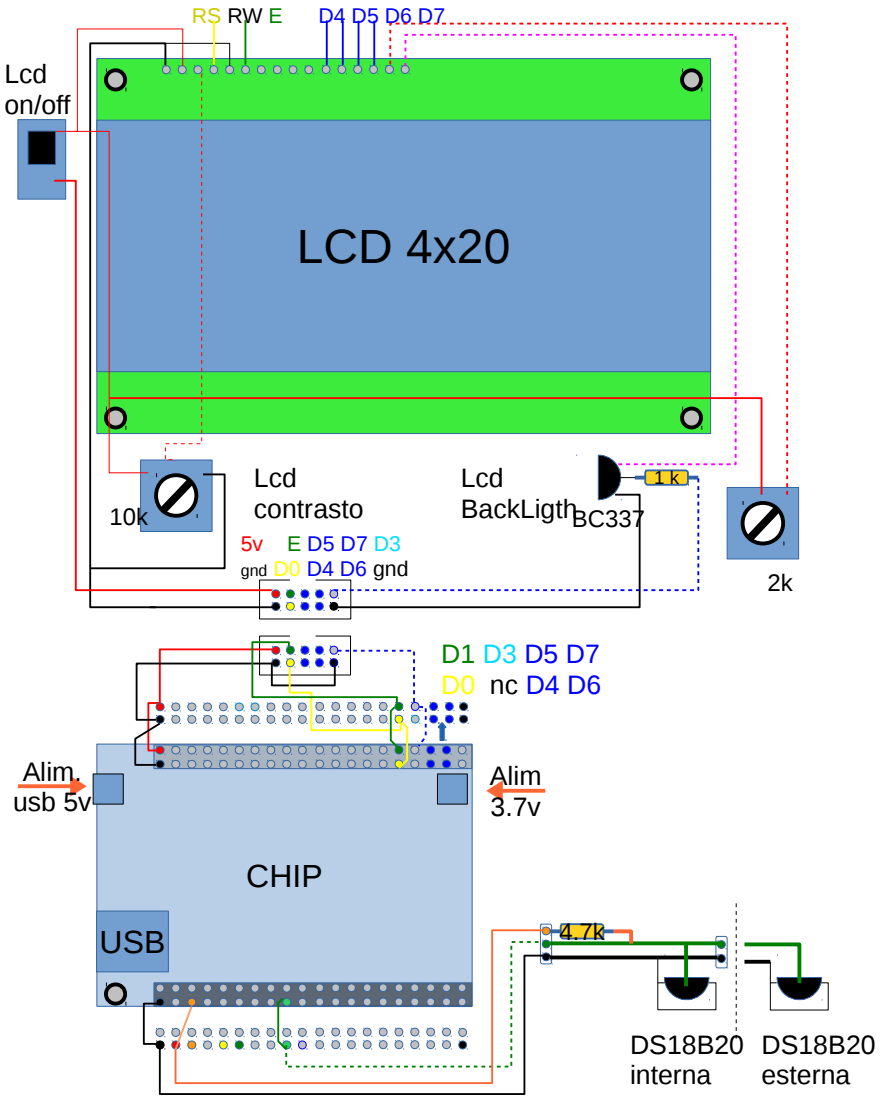
- Ragioni della scelta :
  - CHIP costa poco
  - dispone in modo nativo della connessione WIFI (Raspberry ZERO NO)
  - Il GPIO di cui dispone è compatibile con Raspberry è ha pin sufficienti per connettere sia i sensori di temperatura che un eventuale LCD
  - Si poteva usare anche Arduino Uno con uno Shield WiFi ma era più ingombrate e costoso.

- La programmazione del client “sensori” è fatta in python per via della maggiore rapidità di realizzazione (viste anche le mie conoscenze)
- La lentezza di python rispetto a C non ha molta importanza per un Termostato.
- come Broker mqtt ho usato (o PC in rete) oppure il broker remoto gratuito e aperto per le prove “iot.eclipse.org”
- Anche in questo caso si può interagire subito “in modo grezzo” con i due Client “attuatori e sensori” con la App mqtt su smartfone.

Scelte dei Topic MQTT necessari (vedi programma python):

- il client “Sensori” deve inviare (pubblish) al client “Attuatori” i due Topic Led/bassa, Led/Alta con Payload 0|1 che indicano se il termostatato è in Allarme per Bassa temperatura e quindi lo Switch deve “accendere la Stufa”. Il Topic Led/alta per ora non ha uno Switch corrispondente ma il segnale può essere testato ugualmente con un Led.
- Per decidere la temperatura desiderata e quindi le condizioni di allarme, gestisco (pubblico da SmartFone) due Topic corrispondenti Temp/min Temp/max (le temperature minima e massima che attivano gli allarmi).
- Il client “Sensori” deve inviare le due temperature (al Display LCD e allo SmartFone). I tre Topic Temp/interna, Temp/esterna, Temp/data sono intuitivamente la temperatura ambiente interna che aziona il termostato, quella esterna semplicemente informativa e la data e l'ora dell'ultima rilevazione.
- Tre topic di gestione Cmd/read (che ordina la rilettura dei dati) Cmd/alt (che interrompe il programma in python da SmartFone) e Temp/run (che testa se il programma sta correndo).
- Il software realizzato utilizza diversi moduli i due principali sono relativi ad MQTT <MQTTClient.h> usato su ESP (per IDE Arduino) e paho.mqtt.client usato in python. Sono pubblici e scaricabili dalla rete. Faccio uso anche due moduli python di mia realizzazione ancora in corso di test e sono: TINY.py (usata solo per accedere via 1-wire ai sensori) e Lcd.py (per pilotare LCD 4x20) per ora lavorano nella directory del programma Chip_mqtt_xxx.py in quanto la fase di prova è ancora in corso. Il link che consente di scaricare i modiuli è https://github.com/gcupini/TINY.

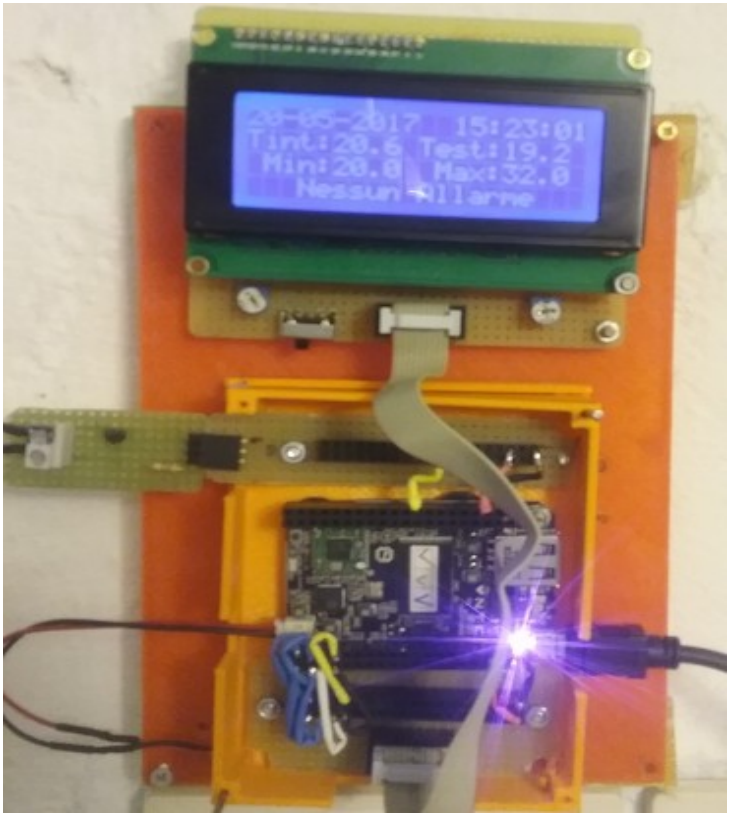
Schema di realizzazione del Gateway “Sensori”:



Materiali:

- CHIP Computer costo: 8.0
- Alimentatore USB per SmartFone standard da 2 Ampere costo: 2.0
- Hd44780 2004 LCD 4x20 (opzionale si usa solo lo smartfone) costo: (9.0)
- due sensori DS18B20 costo: 2.0
- Connettori per cavo flat, 2 Trimmer, BC337, 2 Resitori, ecc costo: 2.0
- Lo SmartFone (o il Tablet) con WiFi costo: ---

Dalle foto della pagina seguente si vedono i diversi dispositivi assemblati in una scatola in PLA stampata con 3DRAG. Si può notare a sinistra una piccolo schedina mille-fori estraibile che contiene il sensore DS18B20 (interno) e le due connessioni verso in DS18B20 (esterno).
LCD mostra Data e Ora, Temperature, Min e Max e stato di allarme.
Sotto lo schermata del Tablet (o smartfone) che gestisce le informazioni e i settaggi con la App (MQTT Dash) più elegante della precedente (MyMQTT).


15:23:01
Tint:20.6 Test:19.2
Min:20.0 Max:32.0
Nessun Allarme


Mia
T interna
20.4
2 ore fa
T esterna
23.9
2 ore fa
Rileggi
2 ore fa
STOP
21 ore fa
Minima
22.0
2 ore fa
Massima
32.0
2 ore fa
Allarme BASSA
1
2 ore fa
Allarme ALTA
0
2 ore fa
Data e Ora
22-05-2017
08:57:16
2 ore fa
Stato programma
ON
2 ore fa