



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE
Corso di Laurea in Informatica

**CONVERSIONE DI
STRUMENTI VINTAGE
PER LA DATA
PHYSICALIZATION**

Relatore: Andrea Trentini

Tesi di Laurea di: Davide Busolin
Matr. 930814

Anno Accademico 2020/2021

Indice

1	Introduzione	2
2	Funzionamento	3
2.1	Attori coinvolti	3
2.2	Modalità API	4
2.3	Modalità MQTT	5
3	Cap3	6
4	Conclusioni	7

Capitolo 1

Introduzione

TCP/IP over Avian Carriers[1]

Capitolo 2

Funzionamento

2.1 Attori coinvolti

Prima di parlare del funzionamento dell'oggetto è il caso di soffermarsi su chi interagisce con esso:

Sviluppatore

È in grado di apportare modifiche al codice sorgente. Può aggiungere, rimuovere e modificare funzionalità del programma e apportare modifiche dirette ai file di configurazione.

Utente esperto

Conosce le API ed è in grado di comprendere il formato dei dati che restituiscono. Gli è sufficiente un'interfaccia anche spartana per selezionare la sorgente dei dati e il campo specifico che vuole rappresentato.

Utente inesperto

Non conosce le API né il formato JSON: ha bisogno di un prodotto già pronto che richieda la minima configurazione possibile e questa deve essere particolarmente intuitiva, ad esempio una piccola interfaccia web per scegliere la rete WiFi e inserirne la password al primo avvio. La sorgente dei dati deve essere preconfigurata e se ne viene resa disponibile più di una la scelta deve essere molto semplice, possibilmente tramite interazione fisica con il dispositivo.

2.2 Modalità API

Questa modalità di operazione si basa sull'esecuzione ad intervalli di tempo configurabili di richieste HTTP o HTTPS ad API REST e sulla conversione del risultato in una tensione di uscita. La configurazione è salvata nella memoria flash del microcontrollore in formato JSON, esempio:

```
1 {  
2   "apiUrl": "http://api.coindesk.com/v1/bpi/  
   currentprice/USD.json",  
3   "filterJSON": "{bpi:{USD:{rate_float:true}}}",  
4   "path": "bpi/USD/rate_float",  
5   "min_value": 58700,  
6   "max_value": 58900,  
7   "min_pwm": 0,  
8   "max_pwm": 1023,  
9   "request_interval_ms": 15000  
10 }
```

Dove:

- `apiUrl` è l'URL alla risorsa (stringa)
- `post_payload` in caso di richiesta POST contiene la stringa da inviare nel body
- `filterJSON` è un documento JSON che contiene `true` come placeholder del/i campi che si vuole considerare dalla risposta (stringa/oggetto)
- `path` è una stringa che contiene il "percorso" del campo che si vuole rappresentare fisicamente (campo: float, path: stringa)
- `min_value` indica il valore minimo della scala per rappresentare il valore (float)
- `max_value` indica il valore massimo (float)
- `min_pwm` indica il valore minimo di uscita della PWM da mappare al valore restituito dalle API (int)
- `max_pwm` indica il valore massimo [su ESP il duty cycle massimo equivale a 1023] (int)
- `request_interval_ms` indica l'intervallo di tempo in millisecondi che trascorre tra una richiesta e l'altra alle API (int)

La configurazione può essere alterata tramite una pagina web accessibile all'indirizzo IP nella rete locale del microcontrollore.

[SCREENSHOT PAGINA WEB]

I form sono precompilati con la configurazione in esecuzione (*running-conf*) ed è presente una sezione che riporta la configurazione salvata nella flash (*saved-conf*).

È possibile inoltre inviare un JSON di configurazione tramite MQTT al topic `Phys/setFromJSON`. I campi possono essere modificati singolarmente ai topic:

- `Phys/setApiUrl`
- `Phys/setFilterJson`
- `Phys/setPath`
- `Phys/setMinValue`
- `Phys/setMaxValue`
- `Phys/setMinPwm`
- `Phys/setMaxPwm`
- `Phys/setRequestIntervalMs`

Le modifiche non sono automaticamente salvate nella memoria flash ma ciò deve essere richiesto esplicitamente. Nei form di configurazione è presente una checkbox da spuntare in caso si voglia che questo avvenga.

2.3 Modalità MQTT

Non ancora implementata, ma l'idea è che consenta di visualizzare un valore ricevuto direttamente tramite sottoscrizione a un topic MQTT invece che andarlo a reperire tramite richieste HTTP(S).

Capitolo 3

Cap3

Capitolo 4

Conclusioni

Bibliografia

- [1] David Waitzman. Standard for the transmission of ip datagrams on avian carriers. Technical report, 1990.