



Guida Modulo

ACC

Capitolo	Pagina
Indice	2
ACC (Arduino Connection Controller)	3
ACC-Server - autoconf	3
ACC-Server - set	4
ACC-Server - set toggle	4
ACC-Server - il codice	4
ACC-Client (Arduino Connection Controller - Client)	5
ACC-Client-ID	5
ACC-Client-Key	5
ACC-Client - alive	5
ACC-Client - set	5
ACC-Client - get	6
ACC-Client - il codice	6

ACC (Arduino Connection Controller)

È un protocollo che abbiamo ideato per comunicare con facilità con l'Arduino, mentre lo stavamo progettando ci siamo accorti che questo protocollo può essere esteso per qualunque micro controllore che possa essere connesso ad una rete LAN, siccome è basato sul protocollo HTTP, per poterlo utilizzare basta un server HTTP personalizzato ed un client HTTP per eseguire le richieste al server ACC.

Il server HTTP e il client, sono entrambi sia sul server ACC che sul client ACC (Microcontrollore).

Per semplicità il server verrà chiamato **ACC-Server** mentre il lato client verrà denominato **ACC-Client**.

Questo protocollo, come un modulo a se stante, dal progetto **domotics**, quindi potrà venir utilizzato anche da altri progetti.

L'idea del funzionamento di questo modulo, è poter inviare dei valori da settare sui pin, oppure richiedere lo stato dei pin. Questo protocollo deve funzionare in maniera "sicura" se vi è presente un ACC-Server, oppure in maniera autonoma se il suo server.

La modalità **sicura** utilizza una chiave per scambiare i valori fra l'ACC-Client e l'ACC-server, la chiave è una stringa esadecimale di 12 caratteri. Quando la modalità sicura è abilitata, per richiedere i valori al microcontrollore o settare dei valori sui pin, mentre nella modalità senza l'ACC-Server, chiunque conosce l'indirizzo IP del server ed il funzionamento del protocollo può inviare comandi o richiedere valori all'ACC-Client.

ACC-Server (Arduino Connection Controller - Server)

L'ACC-Server, è composto di un server HTTP ed un elemento per creare le richieste HTTP. Il server HTTP, ha bisogno di una pagina, la quale deve essere in grado di interpretare due richieste:

- **autoconf**: Questa richiesta richiede tramite il suo ID, la quale ritorna la chiave di comunicazione.
 - **set**: Questa richiesta deve contenere, la chiave di comunicazione, il pin ed il valore, questa serve per aggiornare l'ACC-Server nel caso in cui un pin (per esempio bottone), cambia stato.
- L'ACC-Server, si basa sul un database, che viene utilizzato per risolvere le richieste, per esempio quale input deve accendere o spegnere quale luce. Per il database è stata fatta una guida apposita:

[doc/Documentazione/Guide/GuidaFunzionamentoDatabase.md](#).

ACC-Server - autoconf

La richiesta deve essere:

```
http://<serverAddress>:<serverPort>/acc?autoconf&id=<ACC-Client-ID>
```

E la risposta sarà

```
{"id":"<ACC-Client-ID>", "key":"<ACC-Client-KEY>", "server_address":"<serverAddress>:<serverPort>"}
```

Tutte le risposte saranno inviate in formato JSON, questo per facilitare il l'interpretazione da parte del client.

ACC-Client-ID e ACC-Client-KEY sono spiegati nel capitolo successivo



ACC-Server - set

Quando cambia lo stato di un pin digitale di input sull'arduino, (per esempio la pressione di un bottone) questo deve notificarlo al server, per permettere al server di eseguire le giuste operazioni, per esempio modificare lo stato di altri pin.

La richiesta deve essere:

```
http://<serverAddress>:<serverPort>/acc?key=<ACC-Client-KEY>&pin=<changedPin>&set=<pinStatus>
```

La risposta sarà:

```
{"status":"OK","message":"<valoreSettato>"}
```

ACC-Server - set toggle

In alcuni casi potrebbe essere comodo avere una funzione toggle, per esempio per i bottoni, eseguire la richiesta **set**, la quale semplicemente indica che il pin ha cambiato stato. Per eseguire questa richiesta:

```
http://<serverAddress>:<serverPort>/acc?key=<ACC-Client-KEY>&pin=<chang>&set=toggle
```

La risposta sarà:

```
{"status":"OK","message":"<valoreSettato>"}
```

ACC-Server - il codice

Nel caso di domotics l'ACC-Server, è stato implementato in java, per poterlo integrare direttamente con il modulo del web e per riutilizzare in questi due ambienti gli stessi modelli dei dati.

Quindi è stata creata una pagina del web server, con le funzionalità dell'ACC-Server ([src/acc/AccServlet.java](#)), la quale sarà resa disponibile dal webserver all'indirizzo: [http://<serverAddress>:<serverPort>/acc](#) .

ACC-Client (Arduino Connection Controller - Client)

Anche L'ACC-Client, è composto di un server HTTP ed un client, il server rimane in ascolto sulla porta **8080**, per la ricezione delle richieste dell'ACC-Server, mentre il client esegue le richieste all'ACC-Server, quando cambia lo stato di un bottone (o di un pin digitale di input).

ACC-Client-ID

Codice identificativo di un ACC-Client, formato da 12 numeri esadecimali:

```
1234567890ABCD
```

ACC-Client-KEY

È un codice di comunicazione fra il ACC-Client e ACC-Server, viene utilizzato per riconoscere che le informazioni sono autentiche. Questa viene generata dal ACC-Server ed inviata al client al momento della configurazione. Anch'essa è formata da 12 numeri esadecimali:

```
1234567890ABCD
```

ACC-Client - alive

Per controllare che l'arduino sia attivo e funzioni correttamente, per il quale eseguire la seguente richiesta:

```
http://<ACC-ClientIP>:<ACC-ClientPort>/alive
```

La relativa risposta sarà:

```
{"status":"OK"}
```

ACC-Client - set

Per settare il dei valori sui pin di output del arduino, eseguire la seguente richiesta all'ACC-Client:

```
http://<ACC-ClientIP>:<ACC-ClientPort>/acc?key=<ACC-Client-KEY>&pin=<pinToSet>&set=<valueToSet>
```

La relativa risposta sarà:

```
{"status":"OK","message":"<settledValue>"}
```

ACC-Client - get

Per richiedere all'arduino dei valori di dei pin di input o output, bisogna eseguire la seguente richiesta all'ACC-Client:

```
http://<ACC-ClientIP>:<ACC-ClientPort>/acc?key=<ACC-Client-KEY>&pin=<requiredPin>
```

La relativa risposta sarà:

```
{"status":"OK","message":"<value>"}
```

ACC-Client - il codice

Nel caso dell'ACC-Client il codice è stato scritto in python, siccome l'Arduino YÚN (Lilino) permette di scrivere il codice in python ed eseguirlo a stretto contatto con il codice sul lato Arduino. Per una spiegazione più accurata del funzionamento dell'Arduino, usare la guida [doc/Documentazione/Guide/arduinoYun.md](#).

Principalmente il codice è strutturato in due parti, il codice eseguito sull'Arduino, che semplicemente aggiorna lo stato dei pin di output (prendendo i valori dal Bridge condiviso con la parte Lilino) e aggiorna lo stato del bridge con i pin di input. Codice: [ino/ACC-Client.ino/ACC-Client.ino.ino](#).

Mentre il codice relativo a Lilino, si occupa di eseguire il web server, esaudire le richieste dell'ACC-Server e di inviare al server, i cambiamenti dei pin di input, come quelli dei bottoni. Il codice: [py/acc-client.py/](#). Il tutto è strutturato come mostrato nell'immagine sottostante, che rappresenta il diagramma delle classi.

