1.1.14 MQTTClient, client for a MQTT server

Type	Library
FB	eLLabNetworkLib_A400

Questo blocco funzione permette di gestire la connessione ad un server utilizzando il protocollo MQTT (Message Queuing Telemetry Transport) di IBM che l'Organization for the Advancement of Structured Information Standards (OASIS) ha dichiarato come lo standard di riferimento per la comunicazione per l'Internet delle Cose. Il protocollo adotta un meccanismo di pubblicazione e sottoscrizione per scambiare messaggi tramite un appostivo "message broker" che fa da server. Il FB permette sia di pubblicare che di sottoscrivere topics sul broker.

Questo è un blocco funzione protetto per utilizzarlo occorre richiedere il codice di protezione, vedi ***protezione funzioni e blocchi funzione***. **E' comunque possibile utilizzarlo liberamente su tutti i broker che non richiedono autenticazione (Username e Password non definiti)**. Sui broker autenticati il FB funziona in modo test per 30 Min.

Con il comando ***Enable*** si forza la connessione al ***Server*** (Message broker) sulla porta ***Port***, utilizzando il ***Username*** e la ***Password*** definite. Se la connessione và a buon fine si attiva l'uscita ***Connected***. A connessione avvenuta ogni 1/3 del tempo definito in ***KeepAlive*** viene inviato un comando di ping per mantenere la connessione. Se il broker non riceve il ping nel tempo definito in ***KeepAlive*** ritiene la connessione terminata.

Il comando di ***Publish*** permette di pubblicare sul broker il ***Topic*** indicato con relative ***Value***, ***Flags*** e ***QoS***. Se la pubblicazione và a buon fine si attiva per un loop l'uscita ***Published***.

Il comando di ***Subscribe*** permette di sottoscrivere sul broker il ***Topic*** indicato con relative ***Flags*** e ***QoS***. Se la sottoscrizione và a buon fine si attiva per un loop l'uscita ***Subscribed***.

Ad ogni sottoscrizione del valore di un topic sul broker, il broker ne invia il valore a tutti i client che lo hanno sottoscritto. Il FB riceve sia il nome del topic ritornato in ***TBufferRxD*** che il valore ritornato in ***VbufferRxD*** ed attiva per un loop ***TopicRxD***. In ***VLengthRxD*** è ritornata la lunghezza del valore di topic ricevuto.

In ***CTime*** è ritornato il tempo di comunicazione con broker, rilevato alla connessione ed a ogni esecuzione di ping. O sulla pubblicazione di topic solo se ***QoS*** è 1.

In caso di errore viene eseguita una disconnessione e riconnessione al broker e si attiva per un loop l'uscita ***Fault***.

Enable (BOOL)	Comando abilitazione connessione al server MQTT "message broker".
SpyOn (BOOL)	Se attivo permette di spiare il funzionamento del FB.
Publish (BOOL)	Comando pubblicazione di un topic sul server.
Subscribe (BOOL)	Comando sottoscrizione di un topic sul server.
Server (@USINT)	Puntatore stringa definizione server MQTT.
Port (UINT)	Numero porta TCP a cui connettersi.
Username (@USINT)	Puntatore stringa definizione nome utente.
Password (@USINT)	Puntatore stringa definizione password accesso.
ClientID (@USINT)	Puntatore stringa definizione identificativo client (Massimo 23 caratteri).
RemoteFile (@USINT)	Puntatore stringa definizione nome del file su server FTP.
KeepAlive (UINT)	Tempo di vita connessione da parte del broker (S).
Flags (DWORD)	Flags connessione, pubblicazione, sottoscrizione.
QoS (USINT)	Quality of Service.
Topic (@USINT)	Puntatore stringa definizione nome topic (Pubblicazione/Sottoscrizione).
Value (@USINT)	Puntatore valore topic (Pubblicazione).
VLength (UDINT)	Lunghezza valore topic (Pubblicazione).

TBufferRxD (@USINT)	Puntatore buffer nome topic ricevuto da broker.
TBLengthRxD (UDINT)	Dimensione buffer nome topic ricevuto da broker.
VBufferRxD (@USINT)	Puntatore buffer valore topic ricevuto da broker.
VBLengthRxD (UDINT)	Dimensione buffer valore topic ricevuto da broker.
Connected (BOOL)	Si attiva se connessione al server MQTT avvenuta.
Fault (BOOL)	Attivo per un loop se errore esecuzione.
Published (BOOL)	Attivo per un loop su pubblicazione di un topic sul server.
Subscribed (BOOL)	Attivo per un loop su sottoscrizione di un topic sul server.
TopicRxD (BOOL)	Attivo per un loop su ricezione di un topic dal server.
VLengthRxD (UINT)	Lunghezza valore topic ricevuto dal broker
CTime (REAL)	Tempo di comunicazione con il broker (S).

Trigger di spy

Se ***SpyOn*** attivo viene eseguita la funzione ***SysSpyData*** che permette di spiare il funzionamento della FB. Sono previsti vari livelli di triggers.

TFlags	Descrizione
16#00000001	***Tx:*** Frame comando inviati al server.
16#00000002	***Rx:*** Frame dati ricevuti dal server.
16#00000100	***Sc:*** Stringa compilata da funzione compilazione.
16#00000200	***Rl:*** Calcolo Remaining Length.
16#00001000	***Pt:*** Topic pubblicato.
16#00002000	***St:*** Topic sottoscritto.
16#00004000	***Rt:*** Topic ricevuto.
16#10000000	***Lg:*** Messaggio di log.
16#40000000	***Er:*** Messaggio di erro

Codici di errore

In caso di errore si attiva l'uscita ***Fault***, con ***SysGetLastError*** è possibile rilevare il codice di errore.

Codice	Descrizione
10067010	FB eseguita in una task diversa dalla task di background.
10067020	FB protetta, terminato tempo funzionamento in modo demo.
10067030	***ClientID*** non definito.
10067031	***ClientID*** non corretto (Troppo corto o più lungo di 23 caratteri).
10067032	***ClientID*** non corretto (Contiene caratteri errati).
10067100	FB di connessione al server in fault
10067110	Disconnessione dal server, il server ha chiuso la connessione.
10067200	Ricezione frame incompleto da server.
10067210	Ricevuto frame CONNACK non atteso.
10067211~3	Errore nel frame CONNACK ricevuto.
10067220	Ricevuto frame PUBACK non atteso.
10067221~2	Errore nel frame PUBACK ricevuto.
10067230	Ricevuto frame SUBACK non atteso.
10067231~3	Errore nel frame SUBACK ricevuto.
10067240	Ricevuto frame PINGRESP non atteso.
10067241	Errore nel frame PINGRESP ricevuto.
10067280~9	Errore nel frame PUBLISH ricevuto.
10067300	Errore connessione al server.
10067310	Errore gestione ping verso server.
10067320~1	Errore esecuzione comando publish.
10067330	Errore esecuzione comando subscribe.
10067400	Valore stringa nullo.
10067402	Stringa troppo lunga.
10067403	Stringa contiene caratteri non ascii.

Esempi

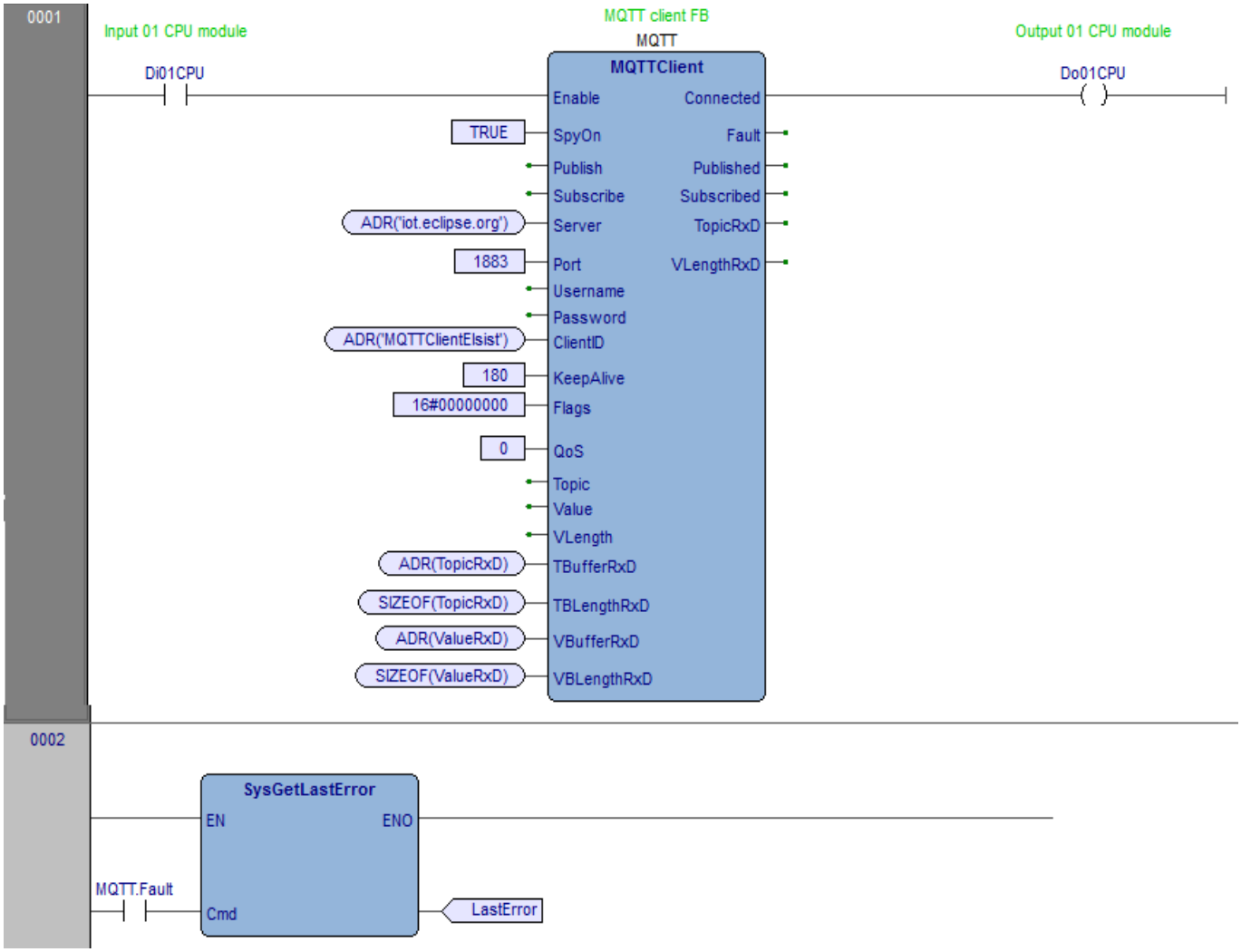
L'esempio permette la connessione ad un broker gratuito ***iot.eclipse.org*** dove viene pubblicato lo stato di un ingresso digitale del modulo CPU e viene sottoscritto il comando di una uscita digitale del modulo CPU. Nei rami seguenti viene istanziato il FB MQTTClient (Ramo 0001) e viene acquisito un eventuale errore di esecuzione che sarà trasferito nella variabile ***LastError*** (Ramo 0002).

Come si vede è stato attivato il comando di spionaggio che permette da Telnet di spiare il funzionamento del blocco funzione.

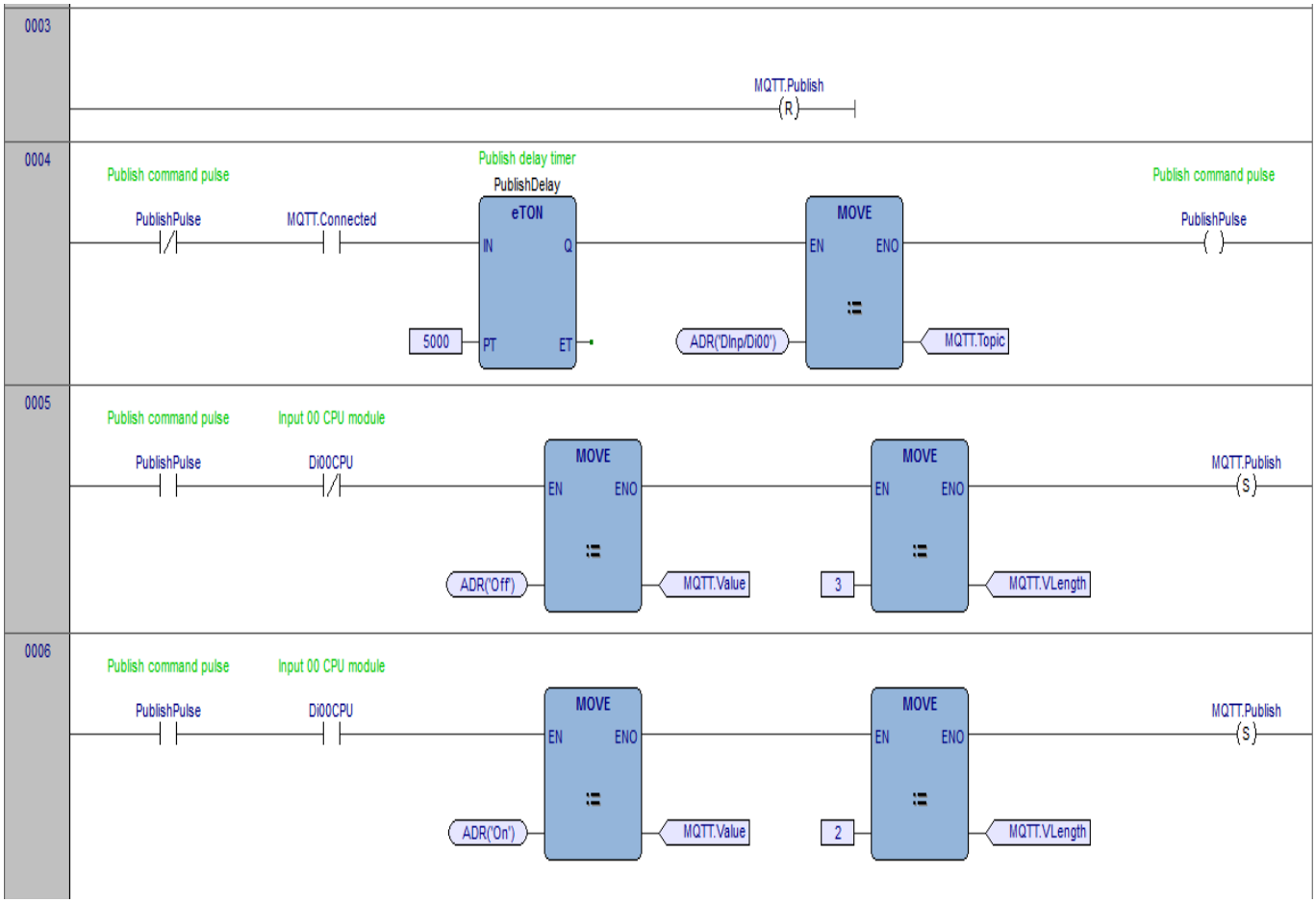
Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	MQTT	MQTTClient	Auto	No		..	MQTT client FB
2	PublishPulse	BOOL	Auto	No		..	Publish command pulse
3	PublishDelay	eTON	Auto	No		..	Publish delay timer
4	TopicRxD	STRING	Auto	[64]		..	Topic received buffer
5	ValueRxD	STRING	Auto	[64]		..	Value received buffer
6	LastError	UDINT	Auto	No		..	Last error

Esempio LD *(PTP119B100, ST_FTPClient)*



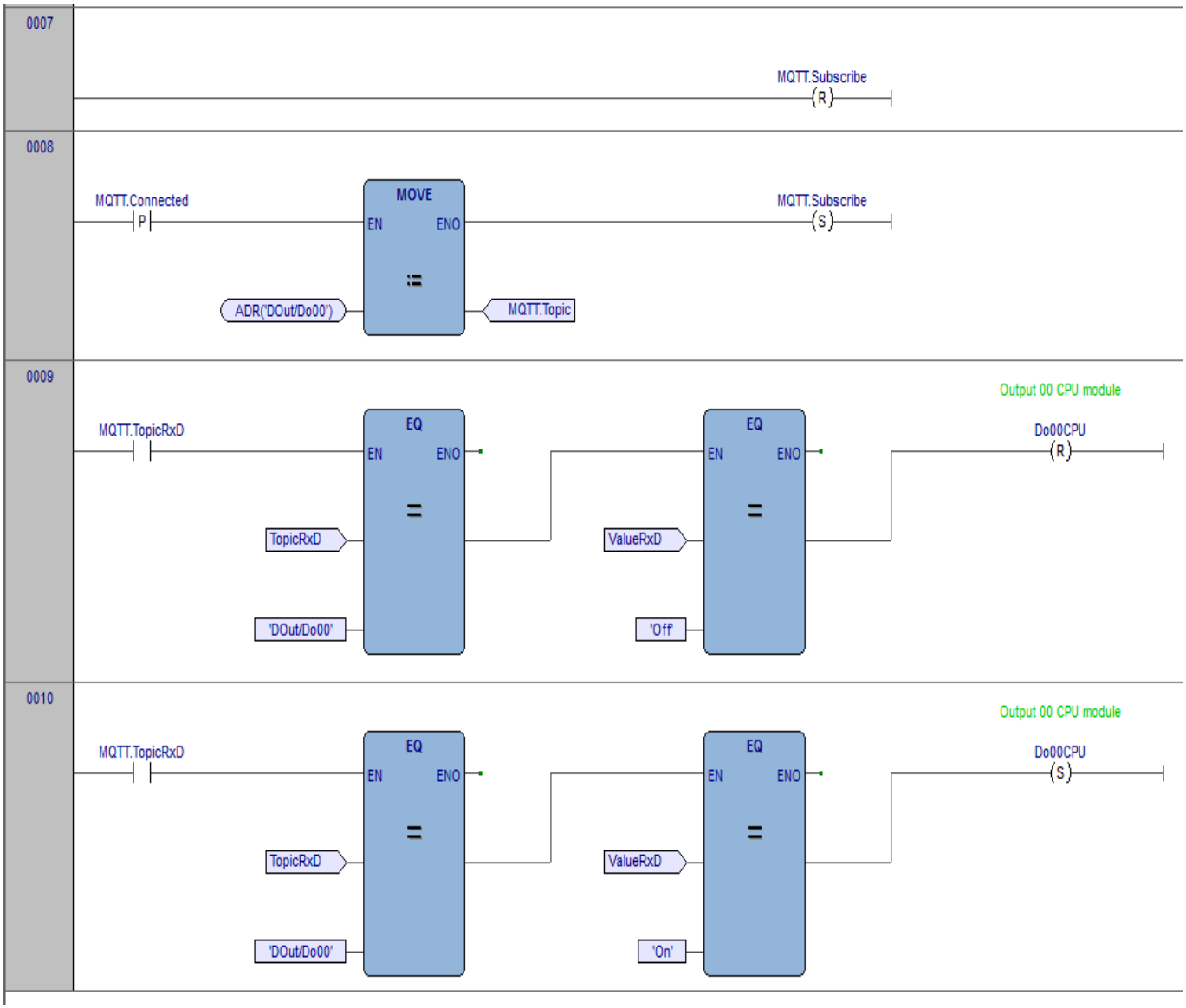
Nei rami seguenti viene gestita la pubblicazione dello stato dell'ingresso 00 del modulo CPU. Siccome il comando di pubblicazione deve durare un solo loop di programma, il comando è sempre resettato (Ramo 0003) e viene settato ogni 5 secondi. Se l'input non è attivo viene inviata la stringa ***Off*** (Ramo 0005), se è attivo viene inviata la stringa ***On*** (Ramo 0006).



Nei rami seguenti viene gestita la sottoscrizione del comando dell'uscita 00 del modulo CPU. Siccome il comando di sottoscrizione deve durare un solo loop di programma, il comando è sempre resettato (Ramo 0007) e viene settato sulla connessione del FB al broker (Ramo 0008).

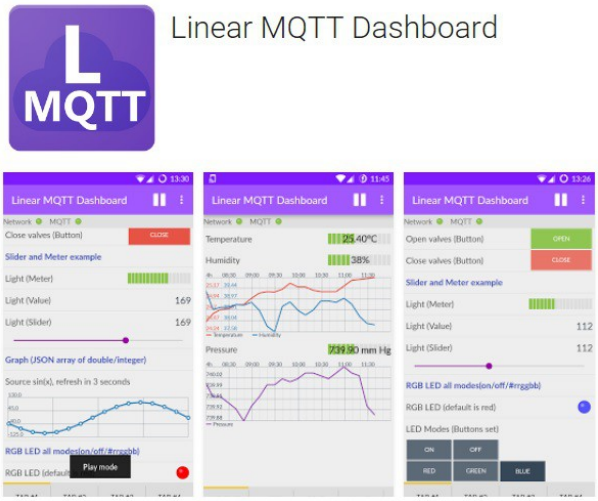
Nel ramo 0009 viene controllata la ricezione di un pubblicazione dal broker, e se il topic è ***DOut/Do00*** ed il valore è la stringa ***Off*** viene resettata l'uscita ***Do00***.

Nel ramo 0010 viene controllata la ricezione di un pubblicazione dal broker, e se il topic è ***DOut/Do00*** ed il valore è la stringa ***On*** viene settata l'uscita ***Do00***.



Client per smartphone

E' possibile testare l'esempio utilizzando un client gratuito per smartphone, scaricando l'app Android ***Linear MQTT Dashboard***. Questa App permette di creare delle schede nelle quali inserire oggetti per la visualizzazione ed il comando sul sistema SlimLine. Naturalmente il dialogo tra l'App e lo SlimLine passa attraverso il broker.



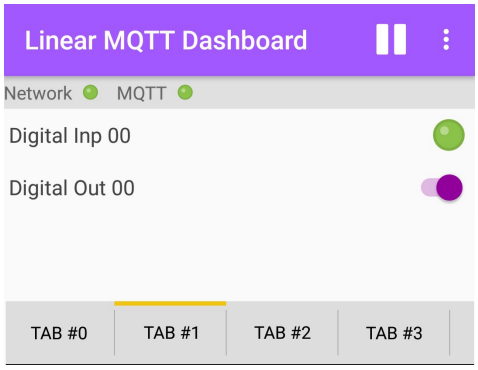
Visualizzazioni: L'FB ***MQTTClient*** sullo SlimLine pubblica degli stati sul brocker e l'App sullo smartphone li sottoscrive visualizzandoli.

Comandi: L'App sullo smartphone pubblica i comandi sul broker, comandi che l'FB ***MQTTClient*** sullo SlimLine ha sottoscritto e che quindi riceve per la gestione.

Vediamo di seguito le operazioni da eseguire per configurare l'App in modo da farla funzionare c on il nostro programma di esempio sullo SlimLine.

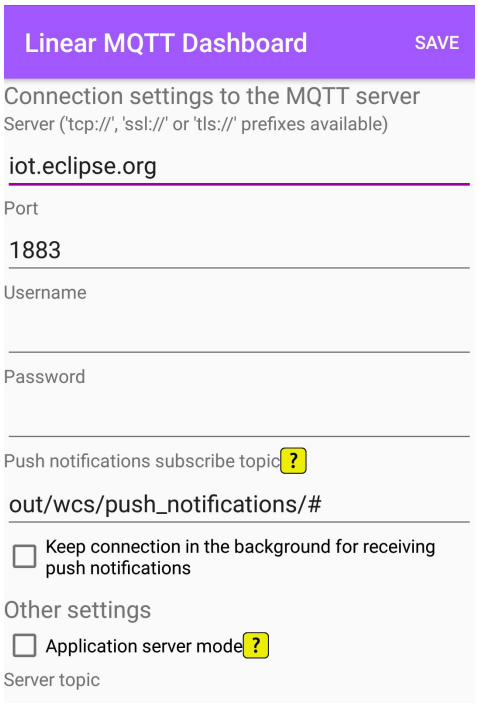
Ecco la scheda che visualizza lo stato dell'ingresso digitale ***Di00CPU*** e permette il comando dell'uscita digitale ***Do000CPU*** del nostro sistem,a SlimLine

In pratica un LED riporta lo stato dell'ingresso mentre un pulsante permette di impostare l'uscita.

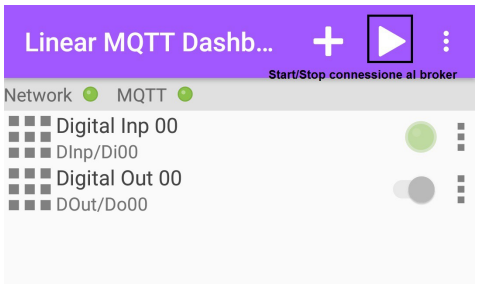


La prima operazione da eseguire è nel menù ***App settings...*** impostare i riferimenti al broker utilizzato, indirizzo URL e porta. Nel nostro esempio ***iot.eclipse.org*** e porta ***1883***.

Il server utilizzato è un server dimostrativo gratuito pertanto non è richiesta l'autenticazione, quindi si lasceranno i campi ***Username*** e ***Password*** vuoti.

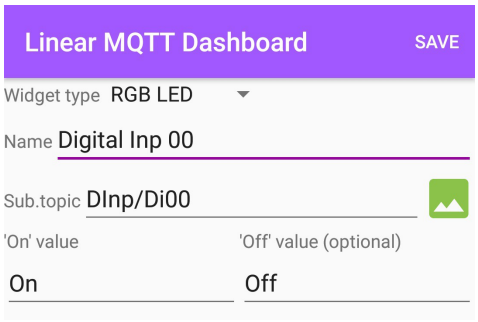


Impostato il broker è possibile andare su uno dei tabs di visualizzazione e definire gli oggetti che si desiderano. Con il broker disconnesso è possibile su ogni oggetto impostare i relativi parametri



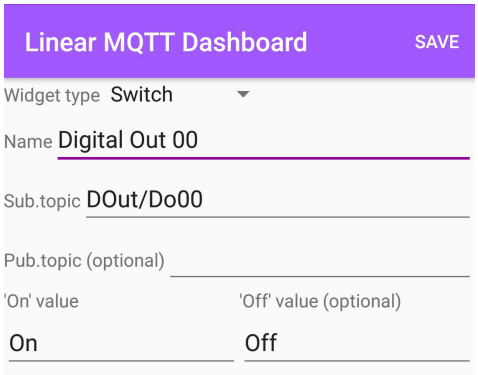
Ecco le impostazioni per il LED di visualizzazione stato ***Di00CPU***, come si vede viene assegnato un nome all'oggetto e viene definito il topic di riferimento ***DInp/Di00***. Questo è lo stesso topic che il FB sullo SlimLine pubblica alla variazione dell'ingresso.

Come si vede dal progetto LogicLab se l'ingresso non è attivo viene inviata la stringa ***Off*** e se è attivo viene inviata la stringa ***On***. Alla ricezione di queste stringhe corrispondono i due relativi stati del LED sulla App.



Ecco le impostazioni per il pulsante di impostazione comando ***Do00CPU***, come si vede viene assegnato un nome all'oggetto e viene definito il topic di riferimento ***DOut/Do00***. Questo è lo stesso topic che il FB sullo SlimLine sottoscrive e controlla sulla ricezione di un valore.

Come si vede dal progetto LogicLab alla ricezione della stringa ***Off*** viene disattivata l'uscita, ed alla ricezione della stringa ***On*** viene settata l'uscita.



Console di spionaggio

Ecco come si presenta la console di spionaggio su Toolly.

Come si vede il comando ***SpyData*** permette di spiare tutto il funzionamento del FB.

Mentre impostando il trigger (Comando ***SpyData -t 00007000***) si vedrano solo i topic pubblicati, e la ricezione di quelli sottoscritti.