POLITECNICO DI TORINO

Corso di laurea Magistrale in Ingegneria Meccanica indirizzo Automazione



Tesi di laurea magistrale

Automazioni industriali mediante integrazione tra PLC e server remoti nell' industria 4.0

Relatore

Prof Luigi Mazza

Tutor Aziendale

Tonny De Abreu

Alessandro Incisa

Candidato

Giuseppe Sette

A.A. 2021/2022

Abstract

L'avvento di Internet ed il continuo progresso tecnologico porta l'uomo ad essere sempre più "connesso" gli uni con gli altri abbattendo quelle che sono le distanze in termini di tempo e spazio. Questa rivoluzione tecnologica, parte integrante e fondamentale dell'industria 4.0, viene spesso indicata con la definizione "Internet of Things". Ma in un mondo dove le macchine sono sempre più connesse, come possiamo sfruttare le nuove tecnologie per migliorare il processo industriale?

Questa tesi si pone come obiettivo l'analisi e l'implementazione di una soluzione pratica che consente di attuare una comunicazione bidirezionale e remota uomomacchina, evidenziando i rivolti pratici sui vari campi industriali. Gli attori
principali di questa comunicazione sono i PLC che gestiscono e comandano macchine
automatiche e i device come smartphone, tablet e PC i quali si interfacciano
direttamente con gli utenti e possono assumere quindi un ruolo predominante nel
controllo della produzione in un ambiente industriale.

Nella prima parte di questo documento verrà trattato lo studio del protocollo di comunicazione MQTT, al fine di comprendere le funzionalità del protocollo e l'architettura della comunicazione.

Nella parte centrale l'implementazione dell'architettura di comunicazione lato hardware e successivamente lato software. Il focus principale è stato dato al programma realizzato interamente su Tia Portal ed eseguito dal PLC. Questo consente alla macchina di connettersi e comunicare con un server remoto, ovvero inviare o ricevere dati da esso. In parallelo per testare e realizzare un esempio pratico son state implementate delle applicazioni mobile con interfacce utente. D'altro canto il software Python è stato di supporto per valutare alcune caratteristiche della comunicazione come tempistiche e affidabilità di invio del dato.

In conclusione l'elaborato pone come obiettivo quello riportare una serie di esempi di campi industriali dove l'implementazione trova spazio e apporta un miglioramento del controllo e gestione di un flusso produttivo.

Indice

Abstract	1
Indice	2
Introduzione esperienza tirocinio in azienda	4
Cap 1: Industria 4.0	5
1.1 Dalla meccanizzazione alla Smart Factory	5
1.1 Le tecnologie abilitanti	7
1.2 Cyber-Physical Systems (CPS) e Smart Factories	8
1.2.1 Cyber-Physical Systems (CPS)	8
1.2.2 Smart Factories	9
1.4 Stato attuale e previsioni future	10
1.4.1 L'inclusività come caratteristica dell'industria 4.0	10
1.4.2 Dieci anni di industria 4.0	11
1.4.3 Nuovi Megatrend per il prossimo decennio	12
Cap 2: Il PLC	15
2.1 Descrizione generale	15
2.2 Storia del PLC dalle origini ad oggi	16
2.2.1 Origini del PLC	16
2.2.2 L'evoluzione dei PLC nell'automazione industriale	17
2.3 Struttura hardware e software	18
2.4 Principio di funzionamento	20
2.5 I PLC continuano ad evolversi	21
Cap 3: Protocollo MQTT per applicazioni IoT	22
3.1 Introduzione	22
3.2 Il protocollo MQTT	24
3.2.1 Architettura del protocollo MQTT	25
3.2.2 Concetti base del MQTT	26
3.2.3 Standard	27
3.2.4 Caratteristiche	28
Cap 4: Libreria MQTT	30
4.1 Introduzione	30
4.2 Implementazione dell'architettura	32
4.2.1 I PLC S7-1200	33
4.2.2 La scheda Raspberry Pi	34
4.2.3 PC, Tablet, Smartphone	35
4.3 Descrizione della libreria	36
4.4 Function Block LMQTT Client	38
4.4.1 LMQTT Client interface description	39

4.4.2 LMQTT Client parametri di connessione	42	
4.4.3 Funzioni del "LMQTT Client"	43	
4.5 Protocollo di sicurezza SSL/TLS		
4.6 Principio di funzionamento della libreria		
Cap 5: Test e prove sulla Libreria		
5.1 Test affidabilità		
5.2 Test velocità di invio		
5.2.1 Prova velocità: dimensione messaggio 65 byte	51	
5.2.2 Prova velocità: dimensione messaggio 128 byte	52	
5.2.3 Prova velocità: dimensione messaggio 196 byte	5 3	
5.2.4 Prova velocità: dimensione messaggio 256 byte	54	
5.2.5 Prova velocità: tempo ciclo 16-18 ms	55	
5.2.6 Prova velocità: tempo ciclo 38-40 ms	56	
5.2.7 Prova velocità: tempo ciclo 95-100 ms	57	
5.3 Risultati	58	
Cap 6: Implementazione pratica		
6.1 Requisiti	59	
6.2 Integrazione e configurazione libreria	59	
6.2.1 Integrazione PLC tags	60	
6.2.2 Integrazione PLC data types	61	
6.2.3 Integrazione PLC Program blocks	62	
Cap 7: Risultati e possibili applicazioni		
7.1 Introduzione	63	
7.2 Comunicazione macchina e server remoti		
7.3 Possibili implementazioni e risvolti pratici sui vari settori	65	
7.3.1 Liberia MQTT a servizio della personalizzazione della produzione	65	
7.3.2 Libreria MQTT a servizio di software gestionali	66	
7.3.3 Liberia MQTT a servizio della manutenzione predittiva	67	
7.3.4 Libreria MQTT a servizio delle analisi e test sperimentali	68	
Cap 8: Conclusioni	69	
Bibliografia e Sitografia		

Introduzione esperienza tirocinio in azienda

Dal mese di maggio fino a novembre ho avuto modo di portare avanti un'esperienza di tirocinio in azienda finalizzata allo studio e sviluppo di un progetto di tesi magistrale.

IOMA è una giovane e innovativa realtà composta da un team di software developer che perseguono l'obiettivo abbattere le barriere tra mondo digitale e società (umani).

Sono molteplici i settori nei quali l'azienda opera, per quanto riguarda l'automazione e l'IoT sono specializzati nella progettazione PLC, robot e domotica comprendendo la programmazione di macchinari industriali linee automatiche e progetti di industria 4.0 invece in parallelo nel settore mobile e Cloud si occupano di progettazione e sviluppo app, gestione server e Cloud.

Tali compiti vengono portati avanti da un team il quale è composto da figure qualificate in campi differenti dell'ingegneria e dell'informatica, le quali osserveranno, analizzando e studiano la soluzione ottimale e performate migliore.

La principale caratteristica che porta al raggiungimento degli obiettivi e l'enorme capacità di integrare differenti discipline abbattendo distanze e realizzando connessioni sinergiche tra varie tecnologie tali da facilitarne il loro utilizzo ma sempre garantendo valore aggiunto performance e risultati ottimali.

Il moto con la quale si identificano è "we are connectors".

Cap 1: Industria 4.0

1.1 Dalla meccanizzazione alla Smart Factory

Il termine Industria 4.0 viene utilizzato per indicare una tendenza all'automazione industriale, che sfrutta le nuove tecnologie produttive per ottimizzare prodotti e condizioni produttive. In accordo si creano nuovi modelli di business finalizzati alla miglioria della qualità e quantità produttiva.

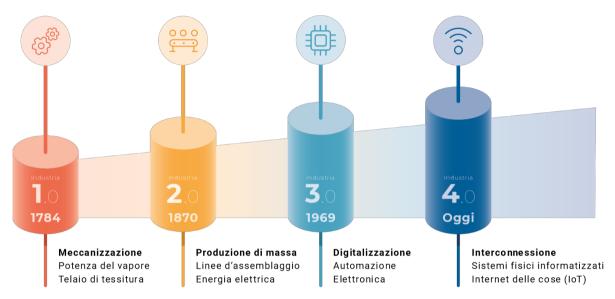


Figura 1: Rivoluzioni industriali

Il termine più specificatamente identifica quella che dovrebbe essere la quarta fase dell'industrializzazione, la quale deriva dall'evoluzione delle precedenti:

1784 Rivoluzione 1.0: La meccanizzazione

L'Industria 1.0 fa riferimento a una rivoluzione della manifattura e nello specifico coinvolge l'invenzione della macchina a vapore, la quale consentì alle fabbriche di abbandonare vecchi mulini a vento o ad acqua, questa introduzione venne accompagnata da un aumento della meccanizzazione nella produzione tessile e metallurgica, all'insegna di una maggiore velocità e potenza.

1870 Rivoluzione 2.0: La produzione di massa

L'Industria 2.0 rappresenta la seconda generazione energetica, legata all'utilizzo dell'elettricità prima e del petrolio poi, che permise l'introduzione sia di componenti elettrici che del motore a scoppio. Si ebbe così un incremento ulteriore dei livelli di meccanizzazione e di produzione. È grazie a questa rinnovata potenza che nella manifattura si afferma progressivamente quella catena di montaggio che inaugura l'era della produzione di massa.

1969 Rivoluzione 3.0: La digitalizzazione

L'industria 3.0 riassume l'ingresso in fabbrica dell'ICT (tecnologie dell'informazione e della comunicazione) di prima generazione: informatica ed elettronica incrementano ulteriormente i livelli di automazione non solo nell'ambito produttivo ma in particolare organizzativo. Si diversificano le infrastrutture e si avviano nuovi processi che, all'insegna della progressiva digitalizzazione, diversificano e agevolano il lavoro delle persone migliorando la qualità della produzione.

2011 Rivoluzione 4.0: Uno sviluppo orizzontale e interconnessione

L'industria 4.0 nasce in sede europea in particolare, il presidente della tedesca Acatech (Accademia tedesca delle Scienze e dell'Ingegneria), *Henning Kagermann*, sviluppa la prima idea e pubblica, nell'anno 2013, il manifesto del paradigma Industrie 4.0. In Europa, tali temi sono stati sviluppati all'interno della Public-Private Partnership Factories of the Future.

L'Industria 4.0, o quarta rivoluzione industriale, fa riferimento all'integrazione di sistemi cibernetici/virtuali e fisici/reali (Cibernetics Physical Systems, CPS), impiego di Internet of Things (IoT).

Mentre la terza rivoluzione industriale ha portato ad uno sviluppo verticale dei sistemi, migliorando ogni processo in modo autonomo, la quarta rivoluzione industriale prevede un completo cambio di paradigma: il focus in questo caso è orizzontale, l'obiettivo è aumentare la sinergia e l'interconnessione tra tutti i processi coinvolti.

I progressi tecnologici portati dall'Industria 4.0 saranno tali da trasformare la produzione realizzando un flusso di produzione integrato, automatizzato e ottimizzato.

1.1 Le tecnologie abilitanti

I processi produttivi che stanno alla base della quarta rivoluzione industriale si basano essenzialmente sulle seguenti tecnologie abilitanti:

- Advanced manufacturing solutions: dove trovano spazio i robot collaborativi interconnessi e agilmente programmabili. Tramite una varietà di sensori installati, acquisiscono una sensibilità tale da essere in grado di lavorare fianco a fianco con gli esseri umani in condizioni di sicurezza e precisione;
- Additive manufacturing: la personalizzazione di massa del prodotto tramite la produzione per addizione di materiale consente la realizzazione di forme complesse e la flessibilità nell'uso della stessa linea produttiva per produzioni diverse:
- *Augmented reality:* una tecnologia che fa riferimento a dispositivi in grado di incrementare le informazioni a disposizione dell'utente in ambienti reali;
- Simulation: simulazione di macchine, processi e lavorazioni. Dalla quale si ottiene un primo riscontro ottimo per l'individuazione e risoluzione di problematiche;
- *Internet of Things (IoT):* tecnologie che si connettono e scambiano dati con altri dispositivi e sistemi su Internet o altre reti di comunicazione;



Figura 2: Tecnologie abilitanti [a]

- *Cloud:* si riferisce ai server a cui si accede tramite Internet, sui quali si eseguono software e i database che si eseguono, ci consentono di archiviare, elaborare e accedere da un utente tramite una connessione internet;
- *Cybersecurity:* tecnologie per proteggere collegamenti, dispositivi e dati da accessi non autorizzati, garantendone sicurezza e privacy;
- Big Data and Analytics: raccolta e analisi di un'ampia base dati con lo scopo di estrarre informazioni utili per ottimizzazione di prodotti e processi lungo l'intera catena del valore;

1.2 Cyber-Physical Systems (CPS) e Smart Factories

Le nozioni di Industria 4.0 sono riconducibili ad un paradigma basato principalmente su due concetti *Cyber-Physical Systems* (CPS) e *Smart Factories*.

1.2.1 Cyber-Physical Systems (CPS)

I *CPS* sono definibili come un'integrazione di sistemi di diversa natura, il cui obiettivo primo è il controllo di un processo fisico e attraverso il feedback, il suo adattamento in tempo reale a nuove condizioni operative.

Il termine physical si riferisce a oggetti reali in quanto "percepibili" dai sensi umani, mentre il termine cyber si riferisce all'immagine virtuale in cui gli oggetti reali sono rappresentati e arricchiti di un livello aggiuntivo di informazioni. La relazione tra gli oggetti fisici e la loro "interpretazione" virtuale è stata da alcuni autori indicata con l'efficace definizione di "social network degli oggetti".

I CPS si fondano su oggetti correlati che, attraverso sensori, attuatori e connessioni in rete, generano e acquisiscono dati di varia natura, riducendo così le distanze e le asimmetrie informative tra tutti gli elementi del sistema.

Con l'aiuto di sensori diffusi, il CPS può determinare in maniera autonoma il suo stato operativo corrente, all'interno dell'ambiente in cui è situato e qualunque sia la distanza tra gli oggetti che lo compongono. Gli attuatori svolgono azioni pianificate ed eseguono decisioni correttive, ottimizzando un processo o risolvendo un problema. Le decisioni sono assunte da un'intelligenza che valuta le informazioni interne al CPS e in alcuni scenari, anche le informazioni provenienti da altri CPS.

La peculiarità dei CPS è quella di essere sistemi automatici dove i computer non solo raccolgono input dai sensori e inviano comandi conseguenti agli attuatori, ma hanno anche la responsabilità di "chiudere il loop" di controllo di alto livello senza l'elemento umano come intermediario. È importante sottolineare che i CPS non eliminano affatto la funzione dell'uomo ma piuttosto ne elevano i compiti a nuovi livelli di complessità. Gli esseri umani assumono il ruolo di controllo di più alto livello, supervisionando le operazioni di processi per lo più automatizzati e autoorganizzati.

La comunicazione di rete diventa la base essenziale e fondamentale nei CPS per consentire a oggetti diversi ed eterogenei di scambiare dati tra loro e con l'uomo, in qualsiasi istante e in qualsiasi condizione. Le prestazioni di comunicazione, in termini di latenza, larghezza di banda e affidabilità, incidono in larga misura sulle interazioni dinamiche tra sottosistemi nei CPS, poiché i processi fisici sono la composizione dinamica di eventi che si svolgono in parallelo. Tuttavia, ogni CPS ha un'architettura specifica che richiede una piattaforma di rete in grado di soddisfare requisiti specifici.

Ad esempio, la latenza è un elemento critico per tutti i casi in cui un controllore o un sistema avanzato di intelligenza artificiale devono prendere decisioni e azioni in tempo reale. Il supporto di una elevata densità di dispositivi connessi diviene, al tempo stesso, cruciale nei casi in cui il CPS si basi su una raccolta massiva di dati da sensori. Questa problematica viene superata con l'utilizzo di tecnologie che permettono di rispondere a requisiti diversi con una stessa rete facendo leva sulle potenzialità del cosiddetto "network slicing".

1.2.2 Smart Factories

Le *Smart Factories* ovvero fabbriche intelligenti, perseguono principalmente l'obbiettivi di controllo sull'andamento della azienda, garantendo lo sfruttamento delle risorse in maniera ottimale, questo concetto viene applicato sui vari livelli:

- Smart production: nuove tecnologie produttive che creano collaborazione tra tutti gli elementi presenti nella produzione ovvero collaborazione tra operatore, macchine e strumenti;
- *Smart services*: tutte le "infrastrutture informatiche" e tecniche che permettono di integrare i sistemi; ma anche tutte le strutture che permettono, in modo collaborativo, di integrare le aziende (fornitore cliente) tra loro e con le strutture esterne (strade, hub, gestione dei rifiuti, ecc.).
- Smart logistic: Tracciabilità/monitoraggio della filiera tramite tag RFID (Radio-Frequency Identification) e sensoristica, monitoraggio della catena del freddo, gestione della sicurezza in poli logistici complessi;
- *Smart energy*: tutto questo sempre con un occhio attento ai consumi energetici, creando sistemi più performanti e riducendo gli sprechi di energia secondo i paradigmi tipici dell'energia sostenibile.



Figura 3: Smart Factory

1.4 Stato attuale e previsioni future

1.4.1 L'inclusività come caratteristica dell'industria 4.0

In maniera più o meno marcata si è davanti a delle vere e proprie Smart Factory dotate di sensori avanzati, software integrati e componenti di robotica che acquisiscono e analizzano dati in continuo, consentendo di ottimizzare il processo decisionale e quello produttivo.

I dati provenienti dalla produzione vengono combinati con i dati operativi di ERP (Enterprise Resource Planning), supply chain, servizi di assistenza ai clienti e altri sistemi aziendali, ottenendo così livelli di know-how e competenza produttiva completamente nuovi, avvalendosi altresì di informazioni precedentemente archiviate.

L'analisi delle grandi quantità di dati raccolti dai sensori del reparto di produzione garantisce la visibilità in tempo reale degli asset di produzione e può fornire strumenti per l'esecuzione della manutenzione predittiva al fine di ridurre al minimo i tempi di inattività delle apparecchiature.

L'Internet delle cose (IoT) e i sistemi cyber-fisici sono ora una realtà fabbriche costruite di recente. Allo stesso tempo, negli stabilimenti esistenti, la connettività tra macchine, strumenti, pezzi in lavorazione e lavoratori qualificati sono stati migliorati, basandosi su varie migrazioni e tecnologie ponte per Industria 4.0.

La sostituzione dell'ispezione manuale con insight visivi con tecnologia di AI riduce gli errori di produzione e consente di risparmiare tempo. Applicando algoritmi di machine learning, i produttori possono rilevare gli errori immediatamente, piuttosto che in fasi successive, quando il lavoro di riparazione è più costoso.

L'integrazione delle tecnologie di abilitazione, tra cui IoT (Internet of Things), il Cloud computing e l'Analytics Intelligence, l'AI e il Machine Learning nei loro impianti di produzione, durante le diverse fasi. Ma la peculiarità principale dell'industria 4.0 è l'inclusività: le nuove tecnologie del 4.0 hanno carattere generale, quindi per loro intrinseca natura riguardano ogni tipo di impresa, dalle startup alle multinazionali, e ogni settore, dai servizi alla manifattura.

1.4.2 Dieci anni di industria 4.0

A dieci anni di distanza dalla coniazione del termine "industria 4.0" è possibile osservare come questa filosofia si sia diffusa e continua a diffondersi con differenti velocità sui vari livelli di industria.

Tuttavia, per il prossimo decennio emergerà tutto il lavoro fatto e da fare per elevare l'industria al 4.0. Questo in concomitanza alle sei tendenze chiave: IA industriale, edge computing fino all'Edge cloud, 5G in fabbrica, robotica di gruppo, sistemi intralogistici autonomi e infrastrutture di dati affidabili.

Prerequisito di base e fattore di successo sarà mettere l'essere umano al centro. Il networking, la connettività delle persone, degli oggetti e le macchine intelligenti, l'uso di architetture orientate ai servizi, e la composizione di servizi e dati provenienti da differenti fonti per formare nuovi processi di business sta aprendo opportunità.

L'industria 4.0 non porterà a fabbriche vuote di persone. Al contrario, i dipendenti sono supportati da sistemi di assistenza fisica che cognitiva realizzati da robot collaborativi (Cobot) e software agenti (Softbot), che supportano gli esseri umani in complesse attività di produzione.



Figura 4: Robot in collaborazione con l'uomo [b]

Industria 4.0 è la base per la creazione di valore basata sui dati, modelli di business innovativi, forme organizzative agili, ma anche per nuove soluzioni in ambiti quali energia, salute, e mobilità. Questa visione è avvincente perché mette le persone al centro, promettendo un significativo progresso per l'economia e la società in generale.

In termini economici, inizialmente ha comportato un cambiamento dall'automazione tradizionale con risultati predeterminati all'apprendimento e all'auto-adattamento macchine e ambienti che rispondono in tempo reale ai cambiamenti della domanda dei clienti, come oltre che a disservizi imprevisti. Questo è accompagnato da un passaggio dalla produzione di massa a personalizzazione di massa, ovvero la produzione a prezzi competitivi di prodotti personalizzati, su misura manufatti.

In termini sociali, l'attenzione si è concentrata sulla cooperazione uomo-macchina senza la paura di perdere il controllo, attraverso la creazione di posti di lavoro il "nearshoring" e l'inclusione di anziani e disabili, supportati da attività fisiche e sistemi di assistenza cognitiva dei lavoratori.

Ecologicamente, l'efficienza delle risorse e dell'energia è stata un obiettivo centrale sin dall'inizio: Industrie 4.0 ha il potenziale per stabilire un'economia circolare che disaccoppia economica crescita dal consumo di risorse. Sostenibilità attraverso l'upcycling e la resilienza Factory sono stati due dei casi d'uso proposti nelle nostre raccomandazioni.

1.4.3 Nuovi Megatrend per il prossimo decennio

La direzione è quella di continuare a guidare l'interoperabilità semantica e collaborazione internazionale in ecosistemi aperti. Questo verrà garantito dai sei nuovi megatrend che influenzeranno in modo decisivo lo sviluppo dei prossimi 10 anni: IA industriale, edge computing fino all'edge cloud, 5G in fabbrica, robotica di gruppo, sistemi intralogistici autonomi, e infrastrutture di dati affidabili.

Il primo livello, quello di rendere tutti i dati della produzione e della catena di fornitura disponibili digitalmente e mobile via sistemi cloud, è ampiamente raggiunto. Questi dati possono ora essere analizzati dai sistemi di intelligenza artificiale in tempo reale e interpretati nel contesto, in modo che possano essere utilizzati attivamente per nuove catene del valore e modelli di business.



Figura 5: Intelligenza artificiale [c]

I dati digitali rappresentano l'allenamento per i sistemi di apprendimento automatico. È possibile utilizzare i sistemi di intelligenza artificiale non solo per la manutenzione predittiva, già diffusa, ma sempre più per controllo di qualità incrementale, principalmente tramite sensori video. Quindi, la prossima fase di Industria 4.0 mirerà a una produzione senza difetti basata sull'IA. Capacità di autoapprendimento e l'autonomia modulare a lungo termine piuttosto che la semplice automazione caratterizzeranno il nuovo generazione di 'Smart Factory' e, oltre all'estrema flessibilità, garantiscono estrema produzione robusta, elevata sicurezza sul lavoro, efficienza energetica e un elevato livello di risorse conservazione. Un'architettura di produzione orientata alle capacità garantisce espandibilità e mutabilità al livello successivo di Industrie 4.0 per rispondere rapidamente alla volatilità dei mercati.

Nelle reti campus 5G, i dispositivi edge possono sfruttare l'elevata larghezza di banda e la bassa latenza garantito con 5G per costruire un cloud perimetrale locale in grado di soddisfare i requisiti in real time al piano della fabbrica.

Nelle "fabbriche intelligenti", la pianificazione intralogistica e la pianificazione della produzione sono coordinate in tempo reale, altamente flessibile: robot mobili, droni di fabbrica e sistemi di trasporto senza conducente assicurano che le parti e gli strumenti necessari per la successiva fase di produzione pianificata siano disponibili appena in tempo, nella giusta isola di produzione.

La pianificazione della produzione viene rivoluzionata da una nuova architettura di produzione orientata ai servizi. La nascita del *Digital Twin* ovvero la coppia del prodotto in forma digitale, trova spazio nella produzione assumendo la funzione di gestione e controllo. Quindi, "gemello digitale" prende parte in un'architettura multiagente, dove le competenze dei lavoratori e le macchine sono coordinate in real time.

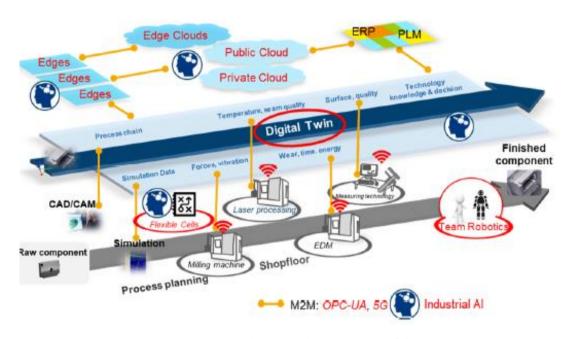


Figura 6: architettura del digital twin [b]

Team ibridi di lavoratori e robot collaborativi con diversi set di competenze ne consentono una nuova forma di robotica di squadra che si concentra sull'interazione uomo-macchina guidata da un essere umano esperto personale. Per risolvere complessi compiti di produzione, stanno lavorando fianco a fianco robot come una squadra.

In conclusione, le fabbriche registrano, trasmettono e archiviano digitalmente tutta la produzione, tramite l'acquisizione di dati provenienti da dispositivi periferici di processo. Questo è un primo passo verso maggiore produttività e maggiore trasparenza dei processi produttivi. In ogni caso, l'interpretazione di queste fonti di dati richiede ancora l'analisi manuale dei dati da parte di esperti umani utilizzando vari strumenti di visualizzazione e analisi dei dati digitali. A causa dell'enorme quantità di dati forniti in tempo reale in una fabbrica di Industria 4.0, gli analisti di dati umani raggiungeranno presto i propri limiti.

Un traguardo importante per il prossimo decennio di Industria 4.0 è quindi l'automatico interpretazione dei dati industriali basata sull'intelligenza artificiale (AI). Correlato a questo si dovrà andare oltre il cloud di oggi quindi puntare su sistemi multi-cloud, poiché i sistemi avanzati di produzione distribuita necessitano di sky computing come una nuvola di nuvole. Il mercato del cloud di oggi è frammentato, con molti proprietari servizi in esecuzione su acceleratori hardware proprietari e offerte API incompatibili. Basato sulla compatibilità e sugli strati inter Cloud, saranno necessarie piattaforme informatiche che consentono l'esecuzione delle applicazioni su più cloud in modo trasparente e interconnesso. Tali piattaforme sono urgentemente necessarie se vogliamo realizzare la visione di circuiti completi di economia circolare in soluzioni distribuite per Industria 4.0 con migliaia di fornitori di dati e consumatori di dati.

La ricerca, l'innovazione e la cooperazione internazionale saranno determinati per la prossima fase di questa quarta rivoluzione industriale. L'interoperabilità dei componenti software e hardware gioca un ruolo cruciale, in particolare per garantire l'accesso al mercato internazionale per le PMI e le startup e consentire così un'ampia e radicale diffusione.

Cap 2: Il PLC

2.1 Descrizione generale

Il PLC, acronimo di *Programmable Logic Controller*, è uno speciale elaboratore di tipo industriale, per il controllo di macchine e processi industriali. Il loro design utilizza componenti modulari in un unico dispositivo per automatizzare i processi di controllo personalizzati. Nasce come elemento sostitutivo della logica cablata e dei quadri di controllo a relè. Lo si può paragonare ad un computer dotato di circuiti, ovvero interfacce ingresso/uscita, capaci di dialogare con dispositivi che possono essere pulsanti, sensori, azionamenti e apparecchiature elettroniche di qualsiasi tipo.



Figura 7: PLCs vs microcontrollori [d], [e]

A differenza dei comuni PC (personal computer) è stato realizzato per lavorare in ambienti industriali dove ci sono temperature elevate, un alto tasso di umidità, disturbi elettrici e vibrazioni.

Rispetto al microcontrollore, la differenza fondamentale sta nel linguaggio di programmazione. I microcontrollori si programmano normalmente con i linguaggi prettamente informatici come assembly e C. I PLC, invece, si programmano con linguaggi dedicati.

In primo luogo è possibile affermare che la tipologia di programmazione si presta maggiormente, rispetto ai microcontrollori, all'integrazione nel settore dell'automazione industriale. L'approccio al microcontrollore è più da progettista elettronico, mentre quello del PLC è più da progettista in sistemi di automazione.

2.2 Storia del PLC dalle origini ad oggi

2.2.1 Origini del PLC

L'automazione industriale è iniziata molto prima dei PLC. Nei primi anni del 1900 fino alla loro invenzione, l'unico modo per controllare i macchinari era attraverso l'uso di complicati circuiti di relè elettromeccanici. L'obbiettivo era quello di comandare accensione e spegnimento in maniera individuale e automatica dei vai componenti delle macchine. Ciò ha portato le fabbriche a necessitare di enormi armadi pieni di relè di potenza.

Poiché l'automazione industriale continuava a crescere, le moderne fabbriche dell'epoca necessitavano di dozzine di motori con interruttori ON/OFF per controllare una macchina e tutti questi relè dovevano essere cablati in un modo molto specifico. I PLC sono stati sviluppati come soluzione per avere un controllo solido come sostituto elettronico per i sistemi di relè cablati.

Dick Morely conosciuto come "Il padre del PLC", ha avuto per la prima volta la visione di un controller programmabile in grado di funzionare per ogni lavoro. Ha messo insieme la proposta nel gennaio 1968. Insieme al team della sua azienda (Bedford and Associate) hanno creato un progetto per un'unità che sarebbe stata modulare e robusta senza utilizzare interruzioni. Lo chiamarono lo 084, che prese il nome dal loro 84° progetto.



Figura 8: Dick Morely [f]

Nel 1968, l'invenzione del primo PLC ha rivoluzionato l'industria dell'automazione.

Contemporaneamente alla 084, Bill Stone con GM Hydramatic (divisione cambio automatico della General Motors) aveva lo stesso problema: problemi di affidabilità e documentazione per le macchine nel suo stabilimento. La sua soluzione proponeva un controller a stato solido come sostituto elettronico dei sistemi di relè cablati.

Per questo Morely insiste di non essere l'inventore del PLC. Morley ha dichiarato: "Il tempo del controller programmabile era quello giusto. Si è inventato da solo perché ce n'era bisogno, e altre persone avevano lo stesso bisogno".

2.2.2 L'evoluzione dei PLC nell'automazione industriale

I PLC sono stati progettati in modo che potessero essere facilmente compresi e utilizzati da ingegneri di impianti ed elettricisti di manutenzione, utilizzando un software chiamato Ladder Logic. Ampiamente utilizzato oggi nei PLC, Ladder Logic è un linguaggio di programmazione che utilizza diagrammi ladder che assomigliano ai binari e ai pioli di un circuito logico a relè tradizionale.

Dopo il successo iniziale con lo 084, Bedford and Associates ha cambiato il proprio nome in Modicon PLC, che sta per Modular Digital Controller. Modicon 084 divenne il nome associato al primo PLC.



Figura 9: Modicon 084 [g]

Nei decenni successivi, il PLC si è evoluto in numerosi modi per adattarsi a vari ambienti e integrare le ultime tecnologie moderne.

L'emergere di concorrenti che hanno sviluppato sistemi simili che rivaleggiavano con Modicon ha suscitato la necessità di nuove innovazioni. Di conseguenza, lo sviluppo di "Data Highway" di Allen-Bradley e "Modus" di Modicon ha consentito ai PLC di scambiare informazioni tra loro.

Con l'adozione più ampia dei PLC, la necessità di un linguaggio di programmazione standardizzato indipendente dal fornitore per l'automazione industriale ha portato all'introduzione dello standard IEC 61131-3, lo standard internazionale per il software PLC realizzato dalla commissione elettrotecnica internazionale.

All'inizio degli anni '90, gli utenti finali iniziarono a fare richieste speciali. I responsabili dell'impianto desideravano che i nuovi macchinari avessero terminali industriali con software di monitoraggio PLC. Volevano macchine che potessero dire ai tecnici cosa non andava, piuttosto che passare ore a risolvere i problemi; ciò ha portato allo sviluppo dell'interfaccia uomo-macchina programmabile (HMI). Da qui in poi si iniziarono a sviluppare dei PLC sempre più complessi e sempre più performanti capaci di integrarsi con una varia gamma di devices odierni.

2.3 Struttura hardware e software

La struttura del PLC è costituita dall'insieme dalla integrazione di due parti:

- Hardware, ovvero la parte fisica, che comprende il processore, memorie, circuiti elettronici e ausiliari.
- Software, che è composto dalle istruzioni che formano il programma utente in funzione delle richieste specifiche.

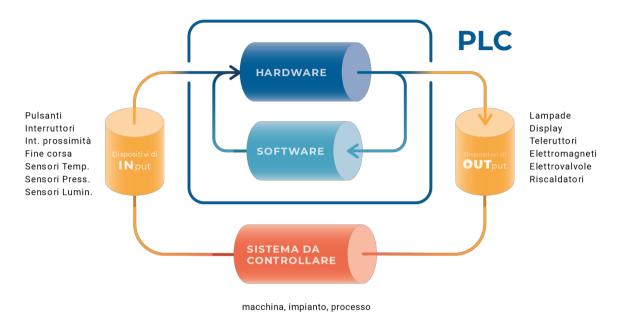


Figura 10: Schema hardware e software PLC

Gli elementi principali che compongono un PLC sono:

CPU (Central Processing Unit) È il cuore del PLC, ovvero il dispositivo che determina l'esecuzione del programma, dei calcoli logici ed aritmetici. Esso interagisce con le memorie e con le periferiche I/O.

Alimentazione la quale provvede a fornire i livelli di tensione tipici valori sono: 230 Vac, 24 Vdc.

Unità INPUT le quali possono essere di tipo analogico e digitale in funzione della tipologia del segnale. Gli ingressi digitali sono segnali provenienti da contatti, pulsanti, termostati, ecc. e che tipicamente hanno tensione 0 allo stato basso e tensione +24V allo stato alto. Gli ingressi analogici sono segnali provenienti da trasduttori di pressione, temperatura, umidità, portata, analizzatori chimici, e altri dispositivi che trasducono la grandezza fisica analizzata in un segnale proporzionale in corrente (tipicamente 4-20 mA) o in tensione (0-10V).

Unità di OUTPUT anch'esse possono essere di tipo analogico e digitale. Le uscite digitali sono i segnali con i quali il PLC comanda (tramite relè ausiliari e/o contattori) gli attuatori, quali motori, elettrovalvole, segnalazioni, ed altri circuiti. Le uscite analogiche sono segnali atti a pilotare valvole proporzionali, strumenti indicatori, registratori, regolatori di velocità per motori e altre apparecchiature regolatrici.

Le unità di comunicazione, consentono al PLC lo scambio dati con computer, altri PLC e una vasta tipologia di dispositivi. Il collegamento avviene tramite tipi di connessione standard come: RS232, RS422/RS485 e porta ethernet con connettori R545 o M12.

La comunicazione avviene tramite protocolli standard, ad esempio: PROFIBUS, TCP/IP, PROFINET.

PROFINET è un protocollo di comunicazione per lo scambio di dati tra controllori e dispositivi, può operare in ambienti industriali esigenti e può fornire la velocità e la precisione richieste dagli impianti di produzione. Per garantire prestazioni adeguate PROFINET utilizza comunicazioni TCP/IP (o UDP/IP) per compiti specifici non critici dal punto di vista temporale, come configurazione, parametrizzazione e diagnostica.

Le performance della connessione sono in grado di garantire comunicazioni veloci e deterministiche per applicazioni esigenti e là dove è richiesto il Real Time poiché è possibile ottenere tempi di ciclo da 512 ms fino a 250 µs.

Un frame Ethernet PROFINET RT in arrivo ha il PROFINET EtherType: 0×8892. All'arrivo al nodo di destinazione, il frame viene indirizzato direttamente da Ethernet all'applicazione PROFINET. Il frame salta i livelli TCP/IP ed evita il tempo variabile necessario per essere elaborato. Pertanto, la velocità di comunicazione e il determinismo migliorano in modo significativo.

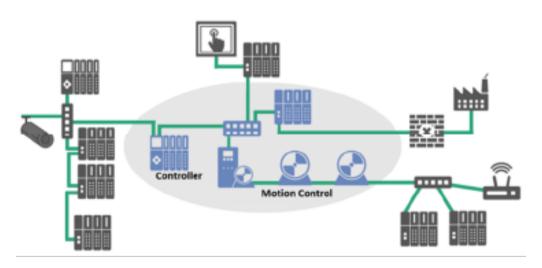


Figura 11: Esempio di collegamenti PROFINET [h]

Topologie: quando si utilizza PROFINET non ci sono limitazioni di topologia. L'utente può implementare configurazioni di linea, stella, albero o anello. L'unico requisito è che tutti i nodi che utilizzano devono essere interconnessi sullo stesso dominio o sezione di rete (come nell'esempio in figura soprastante). È possibile collegare altri nodi PROFINET o Ethernet.

2.4 Principio di funzionamento

Il principio di funzionamento del PLC può essere complessivamente riassunto nello schema sottostante. Il controllore programmabile legge i valori dei segnali di ingresso derivanti dai sensori sul campo, li elabora e sulla base del programma utente contenuto nella sua memoria decide come comandare le uscite.

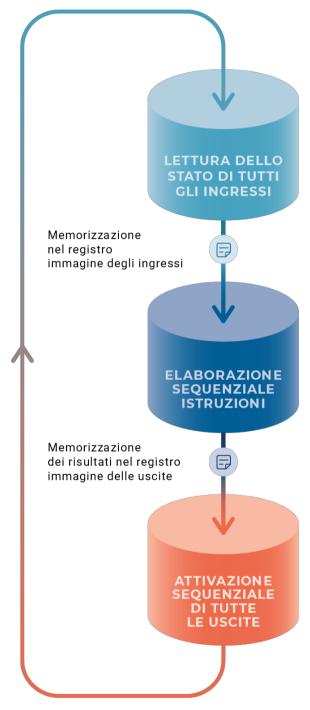


Figura 12: Diagramma funzionamento PLC

Il tempo di scansione o ciclo di scansione per programmi di media complessità si raggira sull'ordine delle decine di millisecondi.

2.5 I PLC continuano ad evolversi

I PLC restano una delle innovazioni più importanti nella storia dell'automazione industriale. Questi dispositivi, che hanno sostituito complicati relè elettromagnetici con un unico controller, sono ancora ampiamente utilizzati oggi. Nei moderni stabilimenti industriali e negli impianti di produzione, il PLC è ancora il sistema preferito rispetto ai PC. Il design si è evoluto per migliorare la tecnologia dimensioni e performance, pur mantenendo le sue peculiarità di resistere alle temperature estreme, agli agenti chimici, alle vibrazioni e ad altre condizioni tipiche di un ambiente industriale.

In parallelo all'evoluzione e diffusione si trova una richiesta sempre maggiore di esperti tecnici PLC che sono qualificati per l'installazione, la riparazione e la manutenzione dei sistemi.

Il grandissimo successo dei PLC risiede principalmente nella velocita con la quale riesce a scansionare il programma e di conseguenza nelle tempistiche di risposta alla variazione di uno stato. Questo è un aspetto molto importante poiché nel controllo di un processo riuscire ad operare con una frequenza di controllo maggiore rispetto a quella del fenomeno fisico che si sta controllando si avrà una gestione robusta sicura e ottimale del processo. Anche nel momento decisionale i ritardi nella risposta non vengono inputati alla unità PLC dal fatto che la CPU riesce a scansionare ciclicamente sul programma con una velocità che si raggira sulla decina di millisecondi.

Particolare attenzione deve essere rivolta alla parte della sensoristica relativa alla sicurezza, la quale grazie a queste tempistiche di risposta può essere integrata al PLC con la completa garanzia di funzionamento. Da considerare che attualmente non vi sono controller più performanti a parità di accessibilità.

D'altro canto oggigiorno le case costruttrici forniscono software sempre più aggiornati e vicini all'utente, questo per semplificare e facilitare la programmazione ma in particole per riuscire a sfruttare al meglio le potenzialità del PLC.

In ultimo la peculiarità che rende insuperabile il PLC oltre ad integrare una numerosissima quantità e tipologia di sensori e attuatori, è che riesce ad inserire e implementare librerie che garantiscono la creazione di interfacce per la realizzazione della comunicazione tramite protocolli standard con dispositivi della stessa o differente natura. Questo è un potenziale non trascurabile nel momento in cui ci si ritrova a dover rispondere a più esigenze le quali al di fuori della portata del PLC, per esempio il campionamento e l'elaborazione di dati provenienti da un sensore, in questo caso il PLC fungerà da campionatore e trasmettitore al database, di conseguenza un calcolatore provvederà all'elaborazione.

Cap 3: Protocollo MQTT per applicazioni IoT

3.1 Introduzione

Il primo dispositivo IoT fu idealizzato e realizzato da David Nichols, uno studente laureato del dipartimento di Informatica della Carnegie Mellon University di Pittsburgh, in Pennsylvania.

Il dispositivo nasceva dall'esigenza di essere a conoscenza da remoto, della presenza o meno di lattine di coca-cola all'interno dei distributori i quali distanti e sparsi i tutto il dipartimento.

Il sistema consisteva nel collegamento del distributore alla rete internet e attraverso un piccolo pannello situato nel suo ufficio, visualizzava le informazioni tramite dei led lampeggianti. Ogni qualvolta che qualcuno acquistava una bibita e quindi a mano a mano che la fila veniva svuotata esso segnalava attraverso il lampeggio di lucine su pannello la presenza residua o meno delle lattine. Integrò anche un ulteriore funzione ovvero quella che riportava a pannello se il distributore era stato ricaricato da poco e quindi la presenza di bibite ancora calde.



Figura 13: Distributore coca cola [i]

Sebbene fosse primitivo per gli standard odierni, detiene una distinzione unica: è stato, per quanto si sa, il primo dispositivo IoT al mondo.

Tuttavia la parola "Internet of Things" è stata coniata dall'imprenditore britannico Kevin Ashton, il dirigente direttore del centro Auto-ID nell'anno 1999. Oggigiorno la comunicazione tra oggetti simili o l'invio di istruzioni non mira esclusivamente ai meccanismi di trasmissione wireless su piccola scala e apparecchi di bassa potenza come elettrodomestici, ma l'Internet of Things è diventata effettivamente la comunicazione da macchina a macchina o da uomo a macchina.

Affinché la comunicazione tra dispositivi e entità sia efficace ed efficiente vi è la necessità di definire una serie di regole condivise e note. L'insieme di esse vanno a comporre quelli che comunemente si chiamano protocolli di comunicazione, i quali fanno sì che i dispositivi riescano a dialogare capendosi.

I protocolli di comunicazione, in telecomunicazioni e informatica, sono definiti come un sistema di convenzioni attraverso il quale sono consentite due o più entità che fanno parte di un sistema di comunicazione, comunicare a vicenda.

Pertanto, il protocollo di comunicazione è costituito dalle regole e quindi da uno standard con cui sono dichiarati la sintassi, la semantica e la sincronizzazione della comunicazione, nonché i possibili metodi per il recupero degli errori. Un protocollo può essere implementato da hardware o software.

Sebbene si possono aver differenti protocolli di comunicazione nella maggior parte dei casi saranno caratterizzati dalle seguenti proprietà: Rilevano la connessione fisica sottostante o se esiste un altro endpoint o un nodo; Il handshaking o a istituzione di comunicazione, che allude a una negoziazione dinamica tra peer in modo dinamico secondo i parametri di un canale di comunicazione, prima che inizi la normale comunicazione nel suo canale; La negoziazione di diverse caratteristiche di connessione; Come inizia e finisce il messaggio; Le procedure per la formattazione del messaggio; Cosa viene fatto con messaggi corrotti o formattati in modo errato; Come rilevare una perdita imprevista per la connessione e cosa viene fatto in seguito; La fine della sessione e o della connessione; Strategie per migliorare la sicurezza; Come verrà costruita una rete fisica.

A seconda dello scopo o dell'applicazione presentano delle peculiarità che possano variare notevolmente. Protocollo di base impostato per la comunicazione di networking è il TCP / IP. Garantisce la trasmissione di informazioni tra dispositivi appartenenti alla rete Internet.

Il TCP o "Transmission Control Protocol", protocollo di controllo della trasmissione, è progettato appunto per consentire affidabilità ovvero trasferimento ordinato di uno stream di dati), controllo della congestione quindi diminuzione del tasso di trasmissione del mittente quando la rete è congestionata e controllo di flusso, il mittente non deve sovraccaricare il ricevente.

Il livello IP permette l'indirizzamento univoco degli host, definisce l'unità di trasferimento dati di internet, ha funzioni di routing (scelta del percorso nella rete) e ha una consegna non affidabile dei pacchetti (consegna non garantita però con impegno, e privo di connessione).

In IoT, si introducono un'altra serie di caratteristiche come la necessità di leggerezza nella comunicazione ma sempre garantendo standard elevati di affidabilità e robustezza.

Il protocollo MQTT è quello più adatto alla comunicazione dei messaggi per dispositivi IoT, poiché pensato e costruito su una rete con una scarsa larghezza di banda e per funzionare con macchine a bassa potenza.

Il proseguo del capitolo pone come obbiettivo quello di illustrare il protocollo MQTT, iniziando da una descrizione in generale per poi descriverne l'architettura ed infine arrivare a quelle che sono le peculiarità che può presentare nell'integrazione di esso nelle macchine automatiche.

3.2 Il protocollo MQTT

Il protocollo "Message Queue Telemetry Transport" o comunemente conosciuto con l'acronimo MQTT, viene utilizzato nell'"Internet of Things" come protocollo di comunicazione. Il suo approccio leggero e flessibile apre nuove possibilità per l'automazione.

Snello e veloce MQTT è un semplice protocollo di pubblicazione e sottoscrizione binario integrato a livello TCP/IP. È adatto per la messaggistica tra dispositivi a bassa funzionalità e alla trasmissione su reti inaffidabili, a bassa larghezza di banda e ad alta latenza. Con queste caratteristiche, MQTT gioca un ruolo importante per l'IoT e nella comunicazione machine to machine.

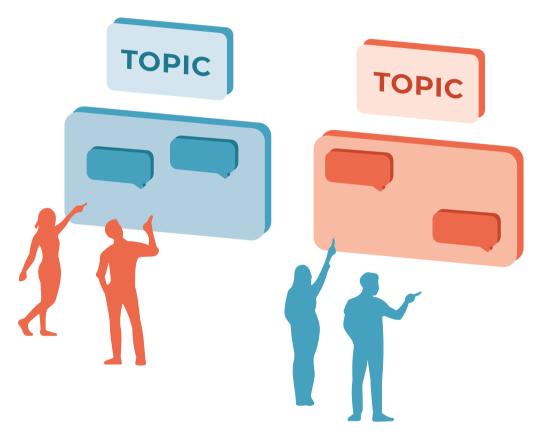


Figura 14: Esempio di comunicazione MQTT

Per introdurre l'idea del protocollo MQTT si può pensare a uno spazio virtuale nel quale vi è una molteplicità di contenitori, con un nome ben definito. Le varie persone vanno a mettere o prendere l'oggetti di proprio interesse. Oppure è possibile pensare differenti tabelloni nominati sui quali le persone pubblicano o leggono ciò a cui sono interessati.

3.2.1 Architettura del protocollo MQTT

L'architettura del protocollo di comunicazione MQTT è basata su un modello di pubblicazione e sottoscrizione da parte di clients e opera attraverso un apposito broker. I client inviano messaggi relativi ad un argomento specifico e i destinatari si iscrivono ovvero si mettono in ascolto sugli argomenti di interesse. Questa comunicazione avviene attraverso il broker. Un client MQTT può essere device o un'applicazione che si appoggia a una libreria MQTT a sua volta connessa in rete internet a un broker.

Quindi all'interno della rete il protocollo MQTT definisce due tipologie di entità: un broker e un certo numero di client.

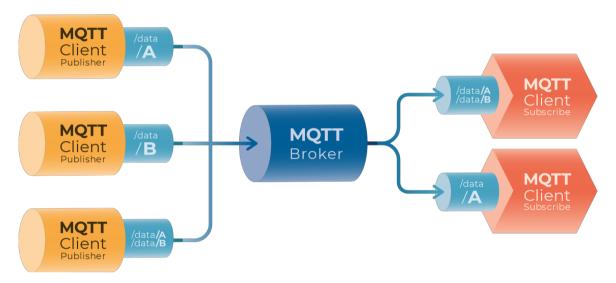


Figura 15: Broker e clients

Il client MQTT pubblica messaggi relativi a un argomento inviandoli al broker. I messaggi MQTT sono organizzati per argomento e i client possono richiedere di leggere e scrivere su qualsiasi argomento. Il messaggio consta semplicemente di un'intestazione che definisce la tipologia, un argomento e infine un payload.

I broker MQTT gestiscono la ricezione dei messaggi e il successivo invio ai sottoscrittori e in alcuni casi gestiscono anche le autorizzazioni. Ciò significa che i mittenti e i destinatari possono accreditarsi presso il broker così che questi li riconosca nel momento in cui inviano un messaggio o si iscrivono a uno o più argomenti. In questo modo il broker comprende elementi importanti, per esempio sa che un determinato client può ascoltare un argomento, ma non può scrivere nulla in merito allo stesso argomento. Addirittura, il broker potrebbe gestire in autonomia tutti gli argomenti possibili, bloccando la creazione degli stessi ai client, ma si tratta di una configurazione particolare che non rappresenta lo standard.

3.2.2 Concetti base del MQTT

MQTT è un protocollo di pubblicazione e sottoscrizione il quale consente di disaccoppiare un client che invia messaggi (Publishers) da uno o più client che ricevono i messaggi (Subscriber). Ciò significa che i "Publisher" non sanno nulla dell'esistenza dei "Subscriber" e viceversa. C'è un terzo componente nell'architettura MQTT, il broker MQTT. Il Broker MQTT si trova tra " Publishers " e "Subscriber" ha il compito di controllare la comunicazione.

Il client si connette al broker e può effettuare la sottoscrizione e la pubblicazione a ogni argomento di interesse. Questa connessione può essere una semplice TCP/IP o una TLS crittografata per i messaggi sensibili.

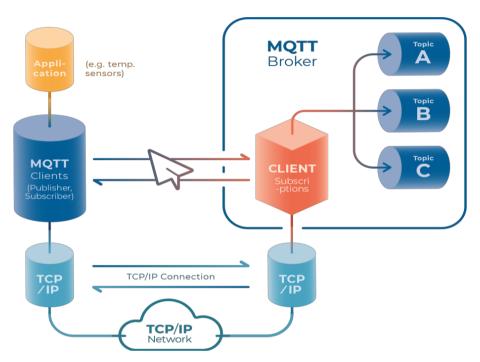


Figura 16: Comunicazione MQTT Client e Broker

Gli elementi fondamentali della comunicazione MQTT sono il client, il broker e il messaggio scambiato.

Un client MQTT è un programma o un dispositivo che utilizza MQTT. Un client stabilisce sempre attivamente la connessione al broker. Un client può eseguire le seguenti funzioni:

- Inviare messaggi con un soggetto definito "Topic", a cui altri clienti potrebbero essere interessati, al Broker MQTT.
- Sottoscrivere messaggi che seguono un determinato argomento presso il broker MQTT.
- Annulla l'iscrizione ai messaggi sottoscritti.
- Disconnettersi dal broker.

Un broker MQTT è il componente centrale di MQTT e può essere un programma o un dispositivo. Il Broker MQTT funge da intermediario tra il Cliente MQTT mittente e il Cliente MQTT abbonato. Il Broker MQTT gestisce gli argomenti inclusi i messaggi in essi contenuti e regola l'accesso agli argomenti. Il Broker MQTT ha le seguenti funzioni:

- Accetta le connessioni di rete dai client MQTT.
- Ricevere messaggi da un client MQTT.
- Modifica le richieste di abbonamento dai client MQTT.
- Inoltra messaggi ai client MQTT che corrispondono al tuo abbonamento.

Un messaggio con MQTT è composto da diverse parti:

- Un argomento definito "Topic".
- Un criterio assegnato per la "Qualità del servizio".
- Il testo del messaggio.

I messaggi MQTT sono organizzati in argomenti detti anche "Topics". Un argomento descrive un'area tematica. Gli argomenti possono essere sottoscritti dai Client MQTT. Il mittente di un messaggio è responsabile della definizione del contenuto e dell'argomento durante l'invio del messaggio. Il broker si preoccupa quindi che gli abbonati ricevano le notizie dagli argomenti sottoscritti. Gli argomenti seguono uno schema definito. Sono simili a un percorso di directory e rappresentano una gerarchia.

3.2.3 Standard

MQTT definisce uno standard ISO. A seconda dei protocolli di sicurezza utilizzati, funziona su diverse porte di accesso. Le porte considerate sono:

- 1883: MQTT, non crittografato
- 8883: MQTT, crittografato
- 8884: MQTT, crittografato, certificato client richiesto
- 8080: MQTT tramite WebSocket, non crittografato
- 8081: MQTT tramite WebSocket, crittografato

3.2.4 Caratteristiche

L'MQTT mette a disposizione una serie di caratteristiche tra le quali:

Quality of Service

La specifica MQTT fornisce tre qualità di servizio per la garanzia della qualità della trasmissione dei messaggi:

- QoS "0": il livello più basso 0 è un metodo "fire'n'forget". Ciò significa che non vi è alcuna garanzia che il messaggio arrivi.
- QoS "1": il livello QoS 1 assicura che il messaggio finisca nella coda degli argomenti almeno una volta. Il broker MQTT conferma la ricezione del messaggio.
- QoS "2": Nel livello 2 più alto, il Broker MQTT garantisce tramite handshake multiplo con il Client MQTT che il messaggio viene archiviato esattamente una volta.

Last will

MQTT supporta la funzione "Last Will and Testament". Questa funzione viene utilizzata per notificare ad altri client MQTT se la connessione a un client MQTT è stata disconnessa accidentalmente.

Ogni Cliente MQTT può specificare la sua ultima volontà durante la connessione al Broker MQTT e notificare il Broker MQTT. Quest'ultima volontà è costruita come un normale messaggio MQTT, inclusi argomento, QoS e payload. Il broker MQTT salva l'ultimo testo e non appena il Broker MQTT si accorge che la connessione con il Client MQTT in questione è stata interrotta bruscamente, il Broker MQTT invia l'ultimo testo come messaggio MQTT a tutti gli abbonati che si sono registrati per l'argomento. In questo modo, gli abbonati apprendono anche che il client MQTT è stato disconnesso.

Keep-Alive

MQTT supporta la funzione "Keep-Alive". Ciò garantisce che la connessione sia ancora aperta e che il client MQTT e il broker MQTT siano connessi.

Per il Keep-Alive, i client MQTT definiscono un intervallo di tempo e lo comunicano al broker MQTT durante la configurazione della connessione. Questo intervallo è il periodo di tempo più ampio possibile tollerato in cui il Cliente MQTT e il Broker MQTT possono rimanere senza contatto. Se il tempo viene superato, il broker MQTT deve disconnettersi.

Fintanto che il client MQTT invia periodicamente messaggi al broker entro l'intervallo Keep-Alive, il client MQTT non deve intraprendere alcuna azione speciale per mantenere la connessione. Tuttavia, se il client MQTT non invia alcun messaggio entro l'intervallo Keep-Alive, deve eseguire il ping del broker MQTT prima della scadenza del termine. Con questo ping, il client MQTT segnala al broker MQTT che è ancora disponibile.

Quando un messaggio o un pacchetto ping è stato inviato al broker MQTT, il tempo per l'intervallo Keep-Alive ricomincia.

Message persistence

Se la connessione a un client MQTT viene interrotta, il broker può memorizzare nella cache i nuovi messaggi per questo client per la consegna successiva.

Retained messages

La prima volta che un client MQTT si iscrive a un argomento, di solito riceve un messaggio solo quando un altro client MQTT invia un messaggio con l'argomento sottoscritto. La funzione "Retained messages" consente di far ricevere l'ultimo messaggio inviato prima che lui si iscrivesse a quel argomento. Quindi il broker archivia l'ultimo messaggio conservato e la QoS corrispondente per quell'argomento. Ogni client che si iscrive a un modello di argomento che corrisponde all'argomento del messaggio conservato riceve il messaggio conservato subito dopo la sottoscrizione. Il broker archivia solo un messaggio conservato per argomento.

Cap 4: Libreria MQTT

4.1 Introduzione

La comunicazione tra due dispositivi, PLC e in generale devices avviene tramite librerie di comunicazione definite ed elaborate in maniera tale da poter sfruttare al pieno le potenzialità dei protocolli ma allo stesso tempo garantire una serie di requisiti come affidabilità, bassa latenza e sicurezza.

Il compito della libreria di comunicazione è quello fare da interprete tra PLC e utenti da remoto. Permettendo l'accesso sia in lettura che in scrittura delle variabili PLC. Questo avviene realizzando dei canali o strade di comunicazione che permettono il transito continuo o su richiesta di pacchetti dati.

La tipologia di variabili PLC di interesse, possono essere stati logici, valori numerici oppure parole meglio definite come stringhe, sono informazioni che il PLC contiene all'interno della sua infrastruttura logica, sulla quale opera e gestisce oppure possono essere informazione appartenenti direttamente a sensori e attuatori fisici sul campo.

Il principio di funzionamento base della libreria è quello di stabilire la connessione tra dispositivi ovvero la strada sulla quale inviare, esporre dati. Nel momento in cui a connessione tra server remoto e PLC in meccanismo che si esegue e quello di prelevare tramite modelli, codificare (ovvero impacchettare il dato all'interno strutture ben definite dette JSON) e inviare. Stesso procedimento nella recezione, quindi viene ricevuto il pacchetto contenente il dato, decodificato e reso a disposizione del PLC.

La richiesta o trasmissione dei dati può essere gestita secondo diverse logiche, per esempio invio dato su richiesta oppure ciclicamente con una determinata frequenza.

Per garantire che le fasi di l'invio e di ricezione siano avvenute in maniera corretta son stati implementati degli algoritmi che si occupano della gestione dato in caso di errori o mal funzionamenti del sistema.

Altrettanto importante per quanto riguarda la libreria, dovrà essere la possibilità di integrarla a programmi software presenti nei PLC senza dover andare ad intaccare o variare il programma, quindi deve presentare una interfaccia espandibile, configurabile e modellabile in base alla tipologia di dati.

Il lavoro è focalizzato principalmente sullo sviluppo della libreria su PLC ma allo stesso tempo per poter elaborare, testare e verificare il corretto funzionamento è stato necessario studiare l'infrastruttura di rete network e implementare il protocollo MQTT su vari devices i quali ricopriranno diverse funzioni.

In questo modo è stato chiuso l'anello di comunicazione che comprende un insieme di tecnologie hardware e software che opportunamente collegate consentono la realizzazione della comunicazione.

La seguente presentazione seguirà l'illustrazione della soluzione intrapresa per implementare la libreria di comunicazione PLC basata sul protocollo MQTT.

Si inizierà con l'analisi dell'architettura fisica, verranno definiti e studiati i componenti hardware, andando ad evidenziare la loro funzione nonché i software essenziali istallati per garantire il corretto funzionamento.

Per poi spiegare i vari livelli della libreria sviluppata su PLC, in primo luogo saranno argomentate le due parti di programma di cui è composta, in secondo luogo analizzeremmo in maniera dettagliata il Function Block MQTT Client messo a disposizione dalla Siemens fino all'integrazione di esso in una logica più complessa.

Una parte verrà dedicata a informazioni generali sul protocollo SSL/TLS fondamentale per la messa in sicurezza dei messaggi.

Infine descriveremmo il principio di funzionamento specificando i vari passaggi che si verificano da quando parte la richiesta di connessione al broker fino all'invio e ricezione dei messaggi.

Successivamente con l'implementazione vedremmo quali saranno le potenzialità e limiti della soluzione implementata.

4.2 Implementazione dell'architettura

Alla base dell'architettura di comunicazione vi sono i client con capacita di pubblicare e sottoscriversi ad un argomento su quale inviano o ricevono un messaggio, d'altro canto si trova il broker che si occupa della gestione dei messaggi e autorizzazioni dei clients.

Nel proseguo di questo capitolo affronteremmo l'architettura studiata e implementata per la comunicazione MQTT tra un broker che si trova su Cloud e una serie di clients connessi alla rete internet con cavo ethernet o wi-fi.

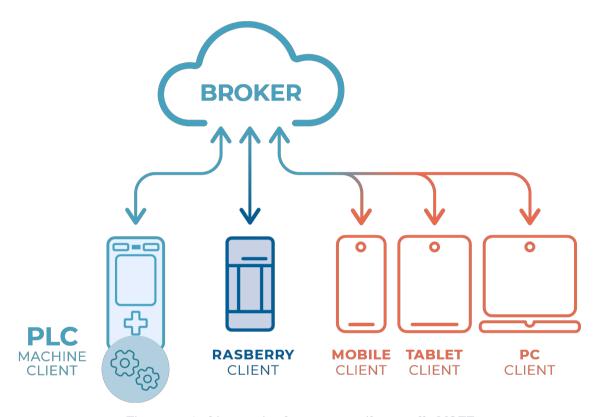


Figura 17: Architettura implementata per il protocollo MQTT

Per quanto riguarda il broker o gestore connessione e dati, è stato utilizzato un servizio online che offre AWS ovvero l'MQTT broker. Questa è una delle tante soluzioni che si potevano intraprendere, per esempio il broker poteva essere inizializzato su Raspberry o su machina virtuale sempre in Cloud. In generale il broker può trovarsi su un qualsiasi server a patto che esso sia connesso e accessibile tramite la rete internet.

Dal punto di vista di broker vi sono due strade, la prima è quella di utilizzare un servizio già esistente online oppure installarlo e configurarlo su server locali con vari middleware.

In base alla loro funzione possiamo suddividere i clients in tre differenti categorie:

- I PLC S7-1200, macchina automatica
- La scheda Raspberry pi, server identificativo
- PC, Tablet, Smartphone, utenti

4.2.1 I PLC S7-1200

Il PLC all'interno della comunicazione rappresenta il controllore che gestisce e comanda la macchina automatica, la quale attua per esempio un processo di produzione ed e ha conoscenza dei dati di nostro interesse.

I PLC persi in considerazione sono della Siemens serie S7-1200 un controllore largamente utilizzato poiché sistema flessibile e potente in grado di controllare un'ampia varietà di dispositivi e di rispondere alle più diverse esigenze del settore dell'automazione. La CPU riunisce i seguenti elementi in un involucro compatto per creare un controllore di grande potenza composto da un microprocessore, un'alimentazione elettrica integrata, circuiti di ingresso e di uscita, PROFINET integrato e I/O veloci per il controllo del movimento.

Una volta caricato il programma la CPU contiene la logica necessaria per il controllo e il comando dei dispositivi utilizzati nell'applicazione. Quindi controlla gli ingressi e modifica le uscite in base alla logica del programma utente, il quale può comprendere operazioni booleane, di conteggio e di temporizzazione, operazioni matematiche complesse, funzioni per il controllo del movimento e la comunicazione con altri dispositivi intelligenti.

In parallelo al programma si trova la libreria di comunicazione MQTT la quale si integra ad esso senza modificare o alterare le logiche già esistenti. Essa si occuperà esclusivamente di inviare o ricevere dati trasmessi tramite il protocollo MQTT. Tutto ciò avviene tramite dei passaggi ben definiti e gestiti da una logica intelligente che in grado di garantire correttezza, robustezza e sicurezza di ogni singolo dato. Da ricordare d'altronde che il traffico di dati transita su rete internet, motivo per la quale la sicurezza risulta essere un aspetto non trascurabile.

I PLC S7 1200 non sono gli unici che possono supportare questa libreria ma vi è anche la serie S7 1500.

Nel caso specifico la casa costruttrice Siemens, offre la possibilità di utilizzare un software, Tia portal, per sviluppare progetti di automazione, basati dei prodotti della serie SIMATIC.

TIA Portal (Totally Integrated Automation Portal) è il software dove l'utente ha la possibilità di configurare, programmare, e progettare il sistema automatico in modo semplice, intuitivo ed efficiente, con l'ulteriore possibilità di navigare all'interno del progetto in modalità online per eseguire la diagnostica del programma realizzato.

4.2.2 La scheda Raspberry Pi

Un altro componete hardware che fa parte del sistema è la scheda Raspberry Pi, esso ricopre la funzione di server identificativo dei vari sia per gli utenti che per i PLC. La seconda funzione è quella di mettere a conoscenza i vari utenti su quale topic avverrà la comunicazione di loro interesse.

Il Raspberry è un mini computer completo e racchiuso in un singolo board abbinato a una piattaforma di programmazione. Il Raspberry Pi è una piccola scheda madre, con una configurazione principale che prevede una porta LAN Wi-Fi, ingressi USB e HDMI, uno slot per schede SD e un'uscita audio jack stereo, con una porta micro USB per l'alimentazione. A seconda del modello sono presenti molti altri connettori, per collegare device esterni come display LCD e perfino webcam. La scheda supporta il sistema operativo Linux, Ubuntu, oltre ovviamente alla piattaforma nativa Raspberry Pi OS.

Con Raspberry Pi si possono realizzare tantissimi tipi di progetti, infatti con questo mini PC è possibile creare videogiochi, sistemi audio, computer e stampanti 3D, ma anche media center e smartphone. Inoltre è possibile usare il board per la smart home, la gestione di robot.

La funzione del Raspberry è quella di server, ovvero gestisce le connessioni e le sottoscrizioni ai topic da parte dei device.



Figura 18: Raspberry Pi

Più nel dettaglio sulla sceda è stato installato il sistema operativo Ubuntu sul quale sono stati installati un'altra serie di programmi ausiliari per poter inserire e configurare il firmware mosquito MQTT e il software Python. La gestione dei clients viene fatta completamente da una logica implementata su uno script Python il quale tramite un ulteriore software PM2 è in grado auto avviarsi subito dopo la accensione della scheda Raspberry. Questo è un aspetto importante, in particolare nel caso in cui viene a mancare per qualche istante l'alimentazione, il dispositivo e in grado di riavviarsi senza un ausilio esterno e senza compromettere la trasmissione degli altri clients.

Il programma Python contiene all'interno un documento scritto in JSON nel quale sono memorizzate tutte le relazioni tra clients. Questo è utile per la gestione della comunicazione. Più precisamente un client nel momento che si connette al broker invia una richiesta al server chiedendo a quali topic deve sottoscriversi e a su quali deve pubblicare, il server provvede prima di tutto ad identificare il dispositivo (client) e se esso è in memoria procede inviando le informazioni richieste dal client.

4.2.3 PC, Tablet, Smartphone

Infine vi sono gli utenti di questo servizio ovvero coloro i quali sono sia interessati a alle informazioni provenienti dalla macchina e sia ad inviare delle informazioni che potrebbero essere utili alla macchina per il proseguo del processo.

Sono rappresentati da una serie di dispositivi come tablet smartphone e pc. Sui quali deve essere installata un'applicazione e essi devono avere la possibilità di connettersi ad internet ed un display per visualizzare le informazioni.

L'applicazione sviluppata in Flutter integra la libreria di comunicazione MQTT con una logica che decodifica i messaggi in arrivo e codifica quelli in uscita per poi rendigli disponibili e aggiornabili ad un'interfaccia grafica. Allo stesso tempo vi è una piccola logica che consente in prima accensione di inviare un messaggio al server identificativo per avere informazioni sui topic di pubblicazione e di ascolto.

4.3 Descrizione della libreria

Affinché i dati provenienti da un PLC arrivino in maniera robusta, affidabile e sicura ad un broker è necessaria la presenza di un'interfaccia adattabile e flessibile che abbia accesso alle variabili del programma già esistente all'interno del controllore e di una logica che consenta l'invio e la ricezione ordinata in maniera tale da non perdere nessuna tipologia di dati.

Quest'interfaccia è rappresentata dalla libreria che sviluppata interamente su TIA Portal e scaricata su PLC, consente di elaborare e realizzare un programma definito "libreria MQTT". Nel quale vede presente l'integrazione del blocco principale fornito dalla libreria Siemens con una serie di logiche e modelli sviluppati con linguaggi e strumenti di programmazione offerti dal software. In questo modo si è testata una delle tante soluzioni, risultato della collaborazione delle tecnologie attualmente esistenti: software, linguaggi di programmazione e protocolli di comunicazione.

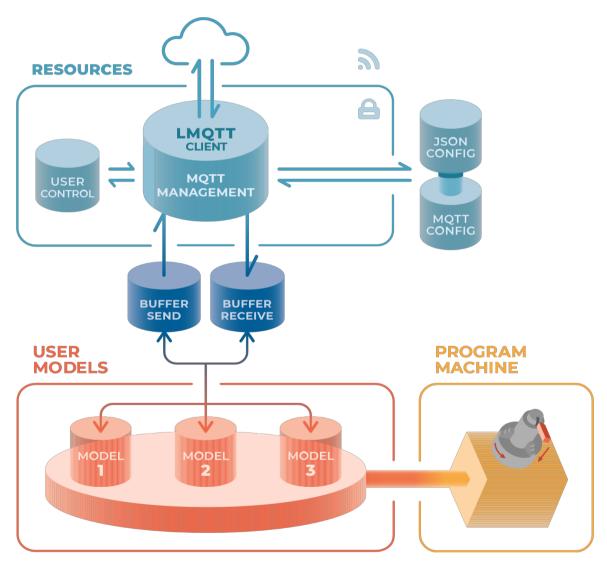


Figura 19: Libreria MQTT

La libreria MQTT è quindi l'interfaccia interposta tra broker e il programma della macchina, per andare a analizzare tutti le varie parti logiche di programma risulta opportuno considerare l'interfaccia complessiva come l'unione di due parti di programma, così definite: Resources e User models.

Il programma della libreria è costituito da due parti resources e user models.

Nella parte di programma definita *resources* ritroviamo il Function block MQTT MANAGEMENT che integra un ulteriore Function block LMQTT Client con la logica di gestione.

Il Function block LMQTT Client consente l'apertura della comunicazione con il broker e il transito dei messaggi sui vari topic.

La logica di gestione ruota attorno a quest'ultimo blocco è ha il compito di asservire il blocco LMQTT client. Le principali funzioni sono: passare i parametri di configurazione JSON e di connessione, gestisce i messaggi con i topic di ricezione e invio.

La gestione dei messaggi in invio si articola in questo modo: a fronte del segnale LMQTT ready e se all'interno del buffer vi è la presenza di messaggi, la logica preleva a mano i messaggi dal buffer, li codifica in JSON, li posiziona nella relativa allocazione di invio e invia.

La gestione in ricezione prevede la decodifica del JSON ricevuto per poi posizionarlo nel buffer di ricezione.

Un ulteriore funzione è quella di offrire un user control nel quale è possibile visionare lo stato della connessione e del funzionamento.

In generale la parte di programma *resources* non necessita modifiche nel momento in cui viene implementata, è necessario esclusivamente definire i parametri di configurazione.

User models è la parte di programma che si interfaccia direttamente con il software del machina e provvede il collegamento tra modelli e singole variabili che devono essere lette o scritte.

I modelli sono delle strutture dati definite con nome e proprietà, le quali sono estremamente flessibili e vengono usate per organizzare i dati che a loro volta devono essere prelevati o inseriti.

Dal punto di vista logico l'user program in condizioni di invio preleva i dati dai modelli e li incolonna nel buffer di invio. Nel caso della ricezione se vi sono dei messaggi nel buffer di ricezione, li preleva e li va ad allocare nei modelli, ovvero li passa al programma della macchia. I modelli vanno definiti e realizzati in fase preliminare, in base alle esigenze.

4.4 Function Block LMQTT Client

Per implementare il protocollo di comunicazione MQTT in un controllore SIMATIC S7, la libreria Siemens offre una soluzione adeguata ovvero il blocco funzionale "LMQTT Client" che integra la funzione MQTT Client. La quale consente di inviare messaggi MQTT a un broker, e di creare sottoscrizioni, ovvero mettersi in ascolto o in attesa di ricezione messaggi. Tale blocco è stato sviluppato da Siemens per due tipologie di PLC SIMATIC S7-1500 e SIMATIC S7-1200.



Figura 20: Function block Tia portal

La comunicazione avviene una volta che il blocco stabilisce la connessione con un broker, sottoscrivendosi e pubblicando su un Topic. Lo scambio dati può essere protetto tramite una connessione TLS. Il Transport Layer Security è un protocollo Internet standard che cripta i messaggi per assicurarne la privacy e il recapito protetto impedendo l'accesso non autorizzato, durante il transito attraverso le connessioni a Internet.

4.4.1 LMQTT Client interface description

Il blocco funzione "LMQTT Client" presenta un'interfaccia per realizzare la comunicazione, nella quale si trovano diverse variabili input e output. Le variabili devo essere passate al blocco funzionale secondo un data type ben definito ed esse rappresentano informazioni di configurazione base, messaggi in input o output, topic sul quale si pubblica e o sottoscrive, ma allo stesso tempo delle variabili che ripotano informazioni sullo stato della connessione e eventuali stati di errori.

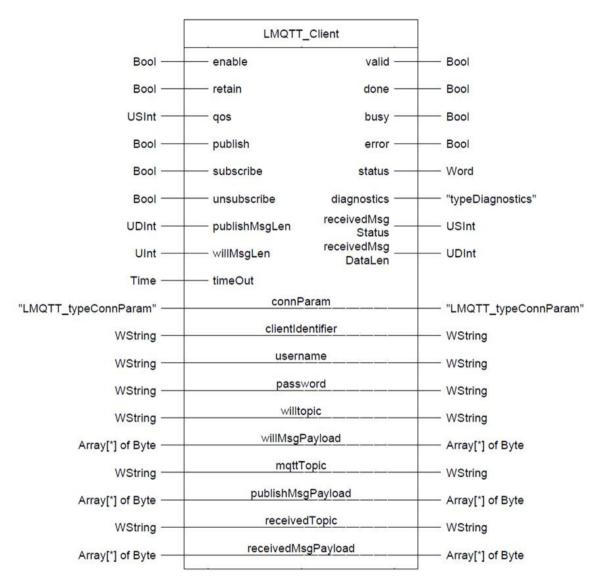


Figura 21: Function block con data type di input e output

Le tabelle sottostanti riportano per ogni variabile la dichiarazione la tipologia di dato (data type) ed effettivamente a cosa corrisponde.

Nome	Dichiarazione	Data type	Descrizione
enable	Input	Bool	TRUE: La connessione al broker MQTT viene stabilita e mantenuta. FALSE: La connessione è interrotta.
retain	Input	Bool	TRUE: I dati vengono inviati con il flag "retain".
			FALSE: I dati vengono inviati senza il flag "retain".
qos	Input	USInt	Quality of Service
			0: I messaggi vengono inviati o sottoscritti con QoS 0.
			1: I messaggi vengono inviati o sottoscritti con QoS 1.
			2: I messaggi vengono inviati con QoS 2 (l'FB non supporta QoS 2 per la sottoscrizione).
publish	Input	Bool	Con connessione esistente a MQTT Broker: Rising edge: la pubblicazione è stata eseguita.
subscribe	Input	Bool	Con connessione esistente a MQTT Broker: Rising edge: la sottoscrizione è stata eseguita.
unsubscribe	Input	Bool	Con connessione esistente a MQTT Broker: Rising edge: L'annullamento dell'sottoscrizione è stata eseguita.
publishMsgLen	Input	UDInt	Lunghezza corrente dei dati validi nel parametro array "publishMsgPayload".
willMsgLen	Input	UInt	Lunghezza corrente dei dati validi nel parametro array "willMsgPayload".
timeOut	Input	Time	Parametro opzionale per configurare il monitoraggio del tempo. Trascorso il tempo, il lavoro corrente sarà considerato come non riuscito.
valid	Output	Bool	TRUE: L'FB esegue la sua funzione senza errori.
done	Output	Bool	TRUE: Il lavoro corrente (pubblica/sottoscrivi/annulla iscrizione) è stato eseguito correttamente.
busy	Output	Bool	TRUE: L'FB è occupato.

Nome	Dichiarazione	Data type	Descrizione
error	Output	Bool	TRUE: C'è stato un errore. Se
			"occupato" è contemporaneamente
			TRUE, il blocco tenta di correggere
			l'errore stesso.
status	Output	Word	Stato e codici di errore
diagnostics	Output	"typeDiag	Informazioni diagnostiche avanzate
		nostics"	
receivedMsgSta tus	Output	USInt	Indica per un ciclo alla volta la ricezione di un nuovo messaggio (sottoscrizione)
			0: Nessun nuovo messaggio ricevuto.
			1: Nuovo messaggio valido ricevuto.
			2: Nuovo messaggio ricevuto, ma il
			messaggio non è valido o i dati ricevuti
			sono più grandi dell'area di memoria
			dell'argomento di ricezione o del
			messaggio di ricezione
ReceivedMsgDa	Output	UDInt	Numero di dati validi nel parametro
ta Len	Gutput		array "receivedMsgPayload".
connParam	InOut	"LMQTT	Parametri per stabilire una connessione
		_typeCon	a MQTT Broker.
		nParam"	
clientIdentifier	InOut	WString	Identificatore client utilizzato per
			stabilire una connessione.
username	InOut	WString	Facoltativo: nome utente per la
username	mout	Wouling	creazione della connessione
			creazione dena connessione
password	InOut	WString	Opzionale: password per la creazione
			della connessione
willtopic	InOut	WString	Facoltativo: topic a cui viene inviato il
			messaggio "Last Will".
	7.0		
willMsgPayload	InOut	Array [*]	Facoltativo: messaggio inviato come
		of bytes	"Last Will".
mqttTopic	InOut	WString	Topic MQTT utilizzato per
			pubblicare/sottoscrivere/annullare
			l'iscrizione.
publishMsgPay	InOut	Array [*]	Messaggio MQTT che viene trasmesso
load		of bytes	come dati utente durante la
			pubblicazione.
receivedTopic	InOut	WString	Topic MQTT su cui è stato ricevuto un
•			messaggio sull'abbonamento.
receivedMsgPa	InOut	Array [*]	Dati utente ricevuti nel messaggio
yload		of bytes	relativo a un abbonamento.
		_	

4.4.2 LMQTT Client parametri di connessione

I *TypeConnParam* non sono altro che i vari parametri necessari affinché venga stabilita la connessione MQTT, tali parametri vengono impostati in base alle proprie specifiche.

Tra i dati di connessione più importanti ritroviamo le informazioni riguardanti:

- Indirizzo IP o nome DNS per stabilire la connessione
- Informazioni di accesso al broker
- Certificati per connessione TLS crittografata

La tabella seguente mostra la descrizione completa di ogni parametro del "LMQTT_typeConnParam".

Nome	Data type	Descrizione
hwId	HW_ANY	Identificatore hardware dell'interfaccia PN/IE per la realizzazione del collegamento. Se si seleziona "0", ne viene selezionato automaticamente uno adatto.
connId	CONN_OUC	ID connessione per stabilire la connessione.
mqttBrokerAddress	Struct	Struttura per le informazioni di connessione del broker MQTT.
mqttBrokerAddress qdnAddress	String	Nome (es. nome DNS) del broker MQTT. Se si utilizza questo parametro, è possibile omettere l'immissione dell'indirizzo IP.
mqttBrokerAddress. ipAddress	IP_V4	Indirizzo IP del broker MQTT
mqttBrokerAddress.	UInt	MQTT-Porta
tls	Struct	Struttura per le informazioni di connessione protetta
tls.enableTls	Bool	TRUE: la connessione è protetta con TLS. FALSE: Connessione non protetta.
tls.validateServerId entity	Bool	TRUE: Il certificato del broker MQTT viene convalidato quando viene stabilita la connessione. FALSE: Nessuna convalida.
tls.brokerCert	UDInt	ID certificato del certificato del broker MQTT.
tls.clientCert	UDInt	ID certificato del certificato del cliente MQTT.
keepAlive	UInt	Meccanismo Keep-Alive di MQTT in pochi secondi. 0: Nessun Keep-Alive.

4.4.3 Funzioni del "LMQTT Client"

Le principali funzione del LMQTT Client sono:

- Logging al MQTT Broker
- Push mechanism
- Subscribe and unsubscribe mechanisms
- Ping mechanism
- Unsubscribe dal MQTT Broker.

MQTT Connection

La connessione viene stabilita tra un client MQTT e il broker MQTT di conseguenza non è possibile una connessione diretta client-client.

Un client MQTT avvia la connessione non appena esso invia un pacchetto "CONNECT" al broker MQTT. All'interno del "Payload" inserisce nei campi esistenti le seguenti informazioni le quali vengono visualizzate nel seguente ordine:

- ID cliente: l'ID cliente viene utilizzato per identificare il cliente presso il broker MQTT. L'ID cliente deve apparire come primo campo nel "payload".
- Argomento di volontà: il campo viene visualizzato facoltativamente se il flag "Will" è impostato su "TRUE".
- Messaggio di volontà: il campo viene visualizzato opzionalmente se il flag "Will" è impostato su "TRUE".
- Nome utente: il campo viene visualizzato opzionalmente se il flag " username" è impostato su "TRUE".
- Password: il campo viene visualizzato opzionalmente se il flag "Password" è impostato su "TRUE". Se positivo, il Broker MQTT risponde con un pacchetto "CONNACK" e un codice di stato.

Il Broker MQTT chiude immediatamente la connessione nei seguenti casi:

- Se il pacchetto "CONNECT" è difettoso.
- Se la struttura del pacchetto "CONNECT" non soddisfa la specifica.
- Se la connessione richiede troppo tempo.

MQTT Push Mechanism

Una volta che un client MQTT si connette al broker MQTT, può inviare messaggi al broker MQTT. Per fare ciò, il client utilizza il pacchetto "PUBLISH". Poiché i messaggi MQTT vengono filtrati e gestiti in base agli argomenti, ogni messaggio MQTT deve contenere un argomento. Il topic fa parte del "Variable Header". Il testo effettivo del messaggio è contenuto nel "payload".

MQTT Sub-Unsub Mechanism

Una volta che un client MQTT si è connesso al broker MQTT, può creare o annullare l'iscrizione ai topics.

MQTT Ping Mechanism

Se l'intervallo Keep-Alive è maggiore di "0", la funzione Keep-Alive è attiva. Se la funzione Keep-Alive è attiva, il Client MQTT deve inviare almeno un messaggio al Broker MQTT entro l'intervallo Keep-Alive. In caso contrario, il Broker MQTT deve terminare la connessione al Client MQTT. Per prevenire questo tipo di interruzione forzata, il client MQTT deve eseguire il ping del broker MQTT prima che scada il tempo Keep-Alive. A tale scopo viene utilizzato il pacchetto di controllo "PINGREQ".

MQTT Disconnection

Un client MQTT può chiudere la connessione a un broker MQTT inviando un pacchetto "DISCONNECT" al broker MQTT. Dopo che il client MQTT ha inviato il pacchetto "DISCONNECT" e chiuso la connessione, non è necessario inviare altri pacchetti di controllo MQTT. Quando il broker MQTT riceve un pacchetto "DISCONNECT", elimina tutte le informazioni di "ultima volontà e testamento". Poiché il Cliente MQTT è connesso attivamente e volontariamente, il Broker MQTT non invia i suoi ultimi desideri agli abbonati registrati.

4.5 Protocollo di sicurezza SSL/TLS

SSL sta per "Secure Socket Layer", TLS per "Transport Layer Security". Entrambi sono protocolli di crittografia per il livello di trasporto di Internet. Il loro compito è criptare i flussi di dati tra client e server.

Il protocollo TLS applica il seguente processo per risolvere il problema:

Quando il client, si mette in contatto con il server, quest'ultimo gli invia prima di tutto il suo certificato. Questo Certificato SSL dimostra che si tratta di un server autentico e non di una falsa identità.

Il client verifica la validità del certificato e invia al server una cifra casuale, crittografata con la chiave pubblica (Public Key) del server.

Da questa cifra casuale il server crea una chiave di sessione (Session Key) che servirà a crittografare la comunicazione. Poiché la cifra casuale è generata dal client, si può essere certi che la chiave di sessione sia realmente generata dal server destinatario.

Il server invia la chiave di sessione al client in forma crittografata. Questa operazione di crittografia avviene con l'ausilio dello scambio di chiavi.

Adesso entrambe le parti possono inviare i loro dati in sicurezza grazie alla chiave di sessione.

Requisiti per la crittografia TLS/SSL.

Per configurare una comunicazione MQTT sicura tra la CPU SIMATIC S7 e un broker MQTT nella propria rete, devono essere soddisfatti i seguenti punti:

- Il Broker MQTT deve essere configurato per la procedura TLS.
- Il certificato CA necessario del broker MQTT è a disposizione.
- L'ora della CPU è impostata sull'ora corrente.

Un certificato contiene sempre un periodo di tempo in cui è valido. Per poter crittografare con il certificato, anche l'ora della CPU S7 deve rientrare in questo periodo. Con una CPU S7 nuova di zecca o dopo un reset totale della CPU S7, l'orologio interno viene impostato su un valore predefinito che si trova al di fuori del Runtime del certificato. Il certificato viene quindi contrassegnato come non valido.

Per quanto riguarda il PLC, TIA Portal offre uno strumento "Certificate Manager" per la gestione e l'utilizzo dei certificati. Questo ci consente di caricare i certificati e passarli al blocco MQTT. Nell proseguo vedremmo nel dettaglio i passaggi e la configurazione.

4.6 Principio di funzionamento della libreria

Una volta che la libreria viene interita in parallelo al programma software principale della macchina è opportuno definire i modelli di prelievo e ricezione delle variabili e successivamente impostare i parametri di configurazione, i quali saranno statici ovvero una volta impostati rimarranno sempre quelli. Fatti tutti i collegamenti con il software, la libreria potrà entrare in funzione.

La comunicazione MQTT può essere suddivisa in tre fasi principali una preliminare che si realizza ogni qual volta il PLC si connette al broker, una successiva riguardante l'identificazione del client da parte del server e un'ultima dove avviene lo scambio dei messaggi.

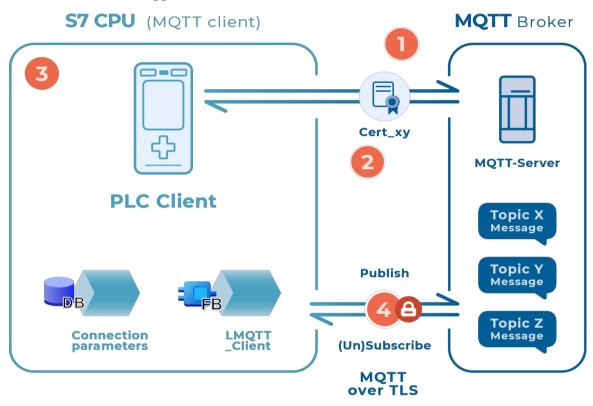


Figura 22: Funzionamento libreria: fase preliminare di connessione

La fase preliminare può essere descritta con la successione di quattro passaggi:

- 1. Richiesta al broker MQTT del certificato CA.
- 2. Ricezione da parte del PLC del certificato CA.
- 3. Autentificazione e validazione del certificato.
- 4. Connessione stabilita, il blocco funzione "LMQTT Client" può assume i seguenti ruoli:
 - Publisher, inviare messaggi MQTT al broker MQTT
 - Subscriber, mettersi in ascolto e ricevere messaggi

Il messaggio MQTT viene crittografato tramite una connessione protetta MQTT su TLS. La fase preliminare quindi consente al PLC di connettersi al broker garantendo un canale di comunicazione aperto e sicuro.

Prendiamo il caso di avere una app di messagistica molto semplice, questo passaggio può essere visto come l'iscrizione all'app la quale prevede l'inserimento e la validazione dei dati per verificare la veridicità non nostra identità. Alla fine di tale processo noi otteniamo l'abilitazione all'invio e ricezione dei messaggi su un canale sicuro.

La fase successiva è anch'essa suddivisa in due, dove nella prima fase vi è l'identificazione del server e nella seconda invece la comunicazione o scambio di dati tra i vari client.

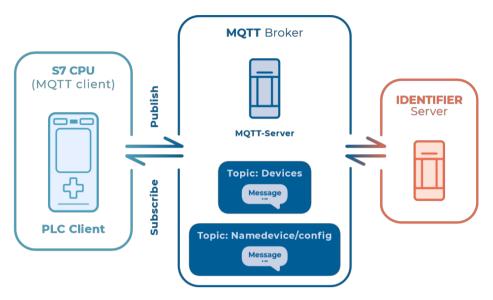


Figura 23: Fase secondaria: identificazione dei client dal server

Identificazione client dal server:

- Il PLC si sottoscrive al topic "namedevice/config" e pubblica, nel topic "devices", un JSON contenente il nome del device "{device: namedevice}".
- Il server in ascolto sul topic "devices", quando riceve il messaggio confronta il "namedevice" con i vari nomi di devices allocati in memoria.
- Se il "namedevice" è presente in memoria allora il server invierà di conseguenza un messaggio sul topic "namedevice/config", contenete i topic sui quali il device dovrà sottoscriversi e pubblicare, il JSON sarà "{sub:topicsub, pub:topicpub}".
- Il PLC leggera il messaggio decodificherà il JSON ed estrarrà i topic ai quali si sottoscriverà e pubblicherà.

Sempre riprendendo l'esempio dell'app questo passaggio può essere visto come se noi mandassimo un primo messaggio a un bot il quale ci spedisce i links di invito ai gruppi sui quali possiamo inscriverci e leggere e i links dei gruppi sui quali possiamo pubblicare.

Nella comunicazione aperta e sicura, siamo a conoscenza dei topi sui quali pubblicare e sui quale iscriversi e metterci in ascolto. È possibile distinguere due funzioni invio e ricezione dei messaggi:

- Il meccanismo di invio può essere inizializzato con logiche differenti tra le quali: invio su richiesta, invio ad evento e invio periodico.
 - Il primo individua un meccanismo nel quale il dato viene inviato ogni qualvolta viene richiesto per esplicito, il secondo invece si riferisce al fatto che il dato viene inviato ogni qual volta esso subisce una variazione rispetto a quello precedente e il terzo invece invia i dati in maniera periodica dato un periodo di tempo tra un invio e quello successivo.
 - Qualsiasi sia il meccanismo di richiesta di invio i passaggi che segue la variabile sono i seguenti: la variabile composta da nome e valore, viene posizionata nel buffer di invio, successivamente quando il blocco LMQTT presenta dichiara di essere pronto per l'invio, il nome e valore vengono codificati e inseriti all'interno di una stringa JSON, la quale viene posizionata nella relativa allocazione di invio e invia. Il buffer a mano a mano che il blocco funzionale invia i messaggi, si occuperà di far scorrere e svuotare il buffer dai messaggi arrivati dal campo. Vi à la possibilità di attivare la modalità echo ovvero il buffer non scorre fino a
 - Vi è la possibilità di attivare la modalità *echo* ovvero il buffer non scorre fino a quando il messaggio non viene inviato e ricevuto nuovamente senza errori.
- Il meccanismo di ricezione: il blocco MQTT è sottoscritto sul topic di ricezione, precedentemente passato dal server. Nel momento in cui vi è la presenza di un messaggio in arrivo provvederà a riceverlo decodificarlo e inserirlo all'interno del buffer di ricezione. Di conseguenza il buffer farà in modo di passare i valori alle relative variabili tramite i modelli.

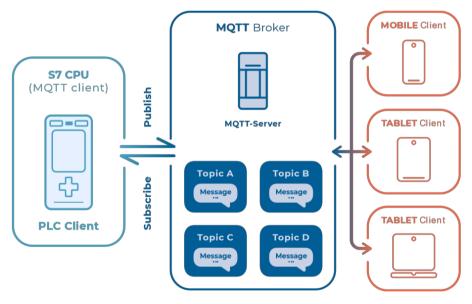


Figura 24: Comunicazione aperta PLC e Utenti tramite il broker

In conclusione il passaggio di identificazione del server dovrà essere effettuato da tutti i dispositivi o clients che vorranno mettersi in comunicazione con il PLC in maniera tale che quest'ultimo possa dire loro in che topics dovranno pubblicare o mettersi in ascolto. Per quanto riguarda il messaggio è composto da una struttura JSON la quale in questo caso specifico è composto da un elenco sequenziale di chiave e valore. La chiave identifica la variabile invece il valore lo stato della variabile che può essere uno stato un booleano o un intero o una striga.

Un alto aspetto da non trascurare è il fatto che i vari clients devono essere a conoscenza delle chiavi, sia del loro nome identificativo che della loro corrispondenza lato software PLC.

Cap 5: Test e prove sulla Libreria

5.1 Test affidabilità

Il primo test è stato effettuato verificare l'affidabilità di consegna del messaggio. Quindi è stata valutata la percentuale di successo o fallimento di ricezione del messaggio.

Per effettuare questo test è stato necessario imporre le seguenti condizioni. Il PLC dovrà spedire al broker un numero elevato di messaggi alla massima velocità, per verificare la ricezione è stato realizzato un cliente con uno script Python che realizza un conteggio dei messaggi pubblicati dal PLC. Questo test verrà ripetuto per differenti dimensioni di messaggio.

Quello che ci si aspetta è che il plc riesca ad inviare con elevato successo tutti i messaggi. D'altro canto avendo velocità di esecuzione del codice lato PLC molto elevate, valuteremmo il comportamento sia del broker che del client in ascolto.

n. Prova	n. Messaggi	Dimensione messaggio	Messaggi persi	Messaggi ricevuti	Percentuale di successo
1	10.000	65 byte	0	10.000	100 %
2	10.000	128 byte	0	10.000	100 %
3	10.000	196 byte	0	10.000	100 %
4	10.000	Max 256 byte	0	10.000	100 %

Le prove sono state quattro con una quantità di messaggi per ogni prova pari a 10.000. Possiamo affermare che in tutte le prove, pur variando la dimensione del messaggio, la percentuale di successo o di ricezione del messaggio da parte del client in ascolto è stata del 100%.

5.2 Test velocità di invio

Il test sulla velocità di invio è un test atto a misurare il tempo impiegato dal PLC a prelevare le variabili ed inviarle al broker.

Un primo test verrà ripetuto per differenti dimensioni di messaggio con obbiettivo di valutare una velocità media e se la dimensione del messaggio influisce sulle tempistiche di invio.

Un secondo test verrà effettuato con differenti carichi di lavoro da parte del PLC. Verrà inizializzato un processo che apporterà un rallentamento del tempo ciclo della CPU, questo per simulare la presenza di un progetto software su PLC più o meno con un elevato numero di operazioni. L'obbiettivo in questo caso è quello di valutare l'influenza che subisce il tempo di invio al variare del tempo ciclo.

Per effettuare questo test è stato implementato una funzione che inserisce all'interno del JSON la data e il tempo esatto nel momento della sua creazione.

A fronte del segnale messaggio inviato, proveniente dal blocco MQTT, la logica provvede a inviare immediatamente un successivo anch'esso contenente la data precisa.

Verranno inviati un numero elevato di messaggi e successivamente i valori di tempo verranno processati per valutare quello che è l'intervallo di tempo di invio tra un messaggio e quello successivo.

5.2.1 Prova velocità: dimensione messaggio 65 byte

La prima prova comporta dell'invio di 10.000 messaggi consecutivi aventi tutti la dimensione di 65 byte.

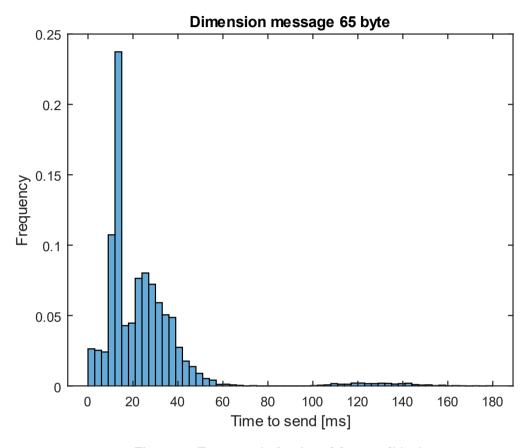


Figura 25: Frequenza in funzione del tempo di invio

Possiamo nottare come la maggiore frequenza di invio la ritroviamo per tempistiche intorno ai 12 millisecondi. Con una media di invio di 24.27 millisecondi. Allo stesso tempo meno frequenti ma da non trascurare son le tempistiche più alte nell'ordine dei 100-160 millisecondi con un valore massimo di 177 millisecondi. In particolare le tempistiche sono:

Tempo massimo	177.01 millisecondi
Tempo medio	24.27 millisecondi

5.2.2 Prova velocità: dimensione messaggio 128 byte

La prima prova comporta l'invio di 10.000 messaggi consecutivi aventi tutti la dimensione di 128 byte.

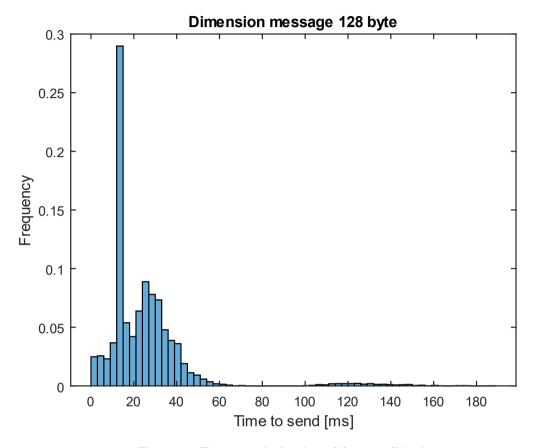


Figura 26: Frequenza in funzione del tempo di invio

In questo caso possiamo nottare come la maggiore frequenza di invio la ritroviamo per tempistiche sempre intorno ai 12 millisecondi. Con una media di invio di 24.71 millisecondi. Allo stesso tempo meno frequenti ma da non trascurare son le tempistiche più alte nell'ordine dei 100-160 millisecondi con un valore massimo di 187 millisecondi. In particolare le tempistiche sono:

Tempo massimo	187.90 millisecondi
Tempo medio	24.71 millisecondi

5.2.3 Prova velocità: dimensione messaggio 196 byte

La prima prova comporta l'invio di 10.000 messaggi consecutivi aventi tutti la dimensione di 196 byte.

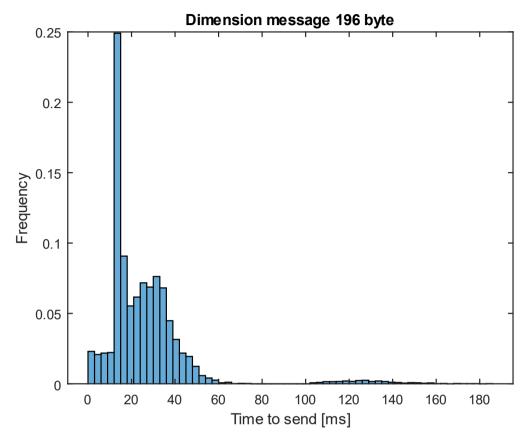


Figura 27: Frequenza in funzione del tempo di invio

In questo caso possiamo nottare come la maggiore frequenza di invio la ritroviamo per tempistiche sempre intorno ai 14 millisecondi. Con una media di invio di 25.92 millisecondi. Allo stesso tempo meno frequenti ma da non trascurare son le tempistiche più alte nell'ordine dei 100-180 millisecondi con un valore massimo di 183 millisecondi. In particolare le tempistiche sono:

Tempo massimo	183.07 millisecondi
Tempo medio	25.92 millisecondi

5.2.4 Prova velocità: dimensione messaggio 256 byte

La prima prova comporta e l'invio di 10.000 messaggi consecutivi aventi tutti la dimensione di 256 byte.

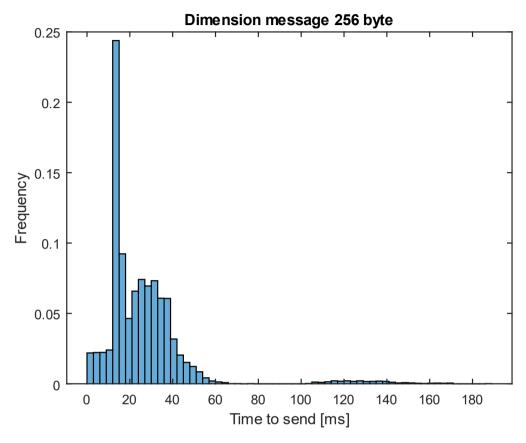


Figura 28: Frequenza in funzione del tempo di invio

In questo caso possiamo nottare come la maggiore frequenza di invio la ritroviamo per tempistiche sempre intorno ai 14 millisecondi. Con una media di invio di 26.15 millisecondi. Allo stesso tempo meno frequenti ma da non trascurare son le tempistiche più alte nell'ordine dei 100-180 millisecondi con un valore massimo di 186 millisecondi. In particolare le tempistiche sono:

Tempo massimo	186.34 millisecondi	
Tempo medio	26.15 millisecondi	

5.2.5 Prova velocità: tempo ciclo 16-18 ms

Il test prevede la simulazione di un programma PLC tramite l'inizializzazione di un processo che apporterà un rallentamento del tempo ciclo della CPU dai 7-8 ms fino a 16-18 ms. Le tempistiche di scansione della CPU possono considerarsi realistiche rispetto ad un programma di piccolo/medio carico.

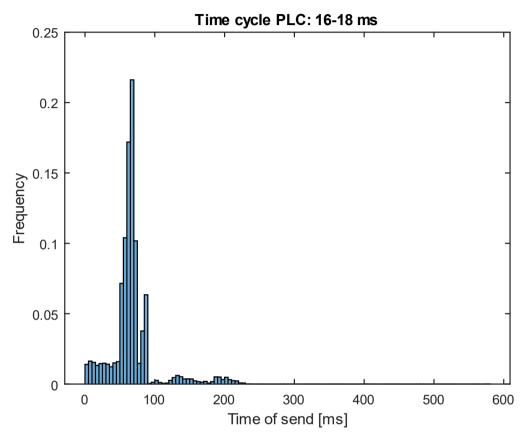


Figura 29: Frequenza in funzione del tempo di invio

Dal grafico è possibile visualizzare come l'avere un programma introduce un ritardo nell'invio dei messaggi rispetto ad un PLC vuoto arrivando alle seguenti tempistiche:

Tempo massimo	576.6 millisecondi
Tempo medio	67.8 millisecondi

5.2.6 Prova velocità: tempo ciclo 38-40 ms

Il test prevede la simulazione di un programma PLC tramite l'inizializzazione di un processo che apporterà un rallentamento del tempo ciclo della CPU dai 7-8 ms fino a 38-40 ms. Le tempistiche di scansione della CPU possono considerarsi realistiche rispetto ad un programma di carico medio.

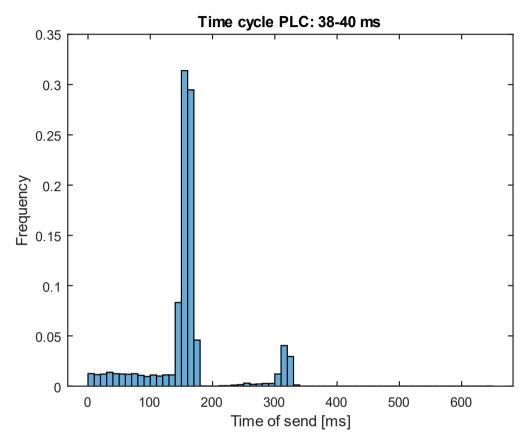


Figura 30: Frequenza in funzione del tempo di invio

Dal grafico è possibile visualizzare come l'avere un programma medio introduce un ulteriore ritardo nell'invio dei messaggi, arrivando alle seguenti tempistiche:

Tempo massimo	649.4 millisecondi
Tempo medio	159.1 millisecondi

5.2.7 Prova velocità: tempo ciclo 95-100 ms

Il test prevede la simulazione di un programma PLC tramite l'inizializzazione di un processo che apporterà un rallentamento del tempo ciclo della CPU dai 7-8 ms fino a 95-100 ms. Le tempistiche di scansione della CPU possono considerarsi realistiche rispetto ad un programma di carico elevato quasi da considerarsi in condizioni critiche, ovvero oltre queste tempiste rimane un piccolo range dopo il quale la CPU, riconosce di non essere più performante e per motivi di sicurezza si stoppa.

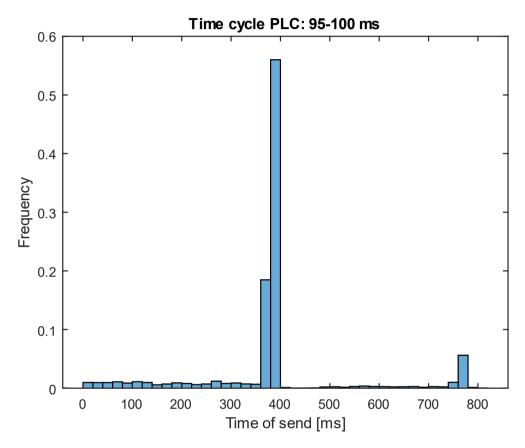


Figura 31: Frequenza in funzione del tempo di invio

Dal grafico è possibile visualizzare come l'avere un programma che introduce un numero elevato di operazioni porta con sé un ritardo nell'invio dei messaggi arrivando alle seguenti tempistiche:

Tempo massimo	800.6 millisecondi
Tempo medio	384.0 millisecondi

5.3 Risultati

Affidabilità

In generale vediamo che la dimensione del messaggio non va ad influenzare quelle che sono le tempistiche di invio, quindi a fonte di tale osservazione è preferibile inviare messaggi con una dimensione massima di 256 byte al posto di spezzare in due il messaggio poiché si va a risparmiare tempo.

Tempo ciclo

Dai test emerge che il tempo medio di invio viene influenzato dal carico presente nel PLC. Solitamente i PLC lavorano con tempi di scansione che non superano gli 80-90 millisecondi, questo per garantire che i segnali input e output vengano aggiornati con una frequenza maggiore.

Tempo ciclo PLC [ms]	Tempo medio invio [ms]	Tempo massimo di invio [ms]
16-18	67.8	576.6
38-40	159.1	649.4
96-100	384.05	800.6

Il tempo medio di invio scala di un fattore 3-4 rispetto a tempo ciclo questo significa che sono avverranno tre scansioni prima che il messaggio verrà inviato.

Cap 6: Implementazione pratica

Questa capitolo ha come obbiettivo quello di racchiudere quelli che sono le impostazioni, configurazioni o in generale i passaggi che bisogna attuare per importare la libreria nel progetto e integrarla.

6.1 Requisiti

Per quanto riguarda il PLC Siemens è opportuno verificare se il modello e la versione del firmware supporta il Function Block LMQTT_Client. Allo stesso tempo il plc deve avere la porta ethernet in modo da poterlo collegare alla rete internet.

Il Broker essere inizializzato su differenti server accessibili tramite la rette internet. Possono far parte si servizi online di piattaforme Cloud. Questi necessitano una loro configurazione e talvolta possono seguire anche un piano tariffario.

6.2 Integrazione e configurazione libreria

Una volta che si ha l'accesso al programma tramite il software Tia portal, la fase primaria è quella di importare dalla libreria la libreria MQTT.

Questo comporta l'importazione di tre parti da inserire all'interno del Program blocks, PLC tags e PLC data types.

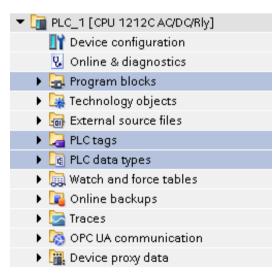


Figura 32: Integrazione tramite Tia Portal

6.2.1 Integrazione PLC tags

La prima parte da configurare si trova all'interno si trova all'interno di PLC tags

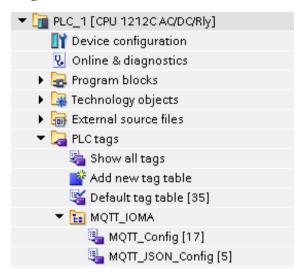


Figura 33: Integrazione PLC tags

I parametri seguenti riguardalo le caratteristiche del JSON e vanno impostati in base alle caratteristiche delle stringhe delle chiavi e valori che lo compongono.

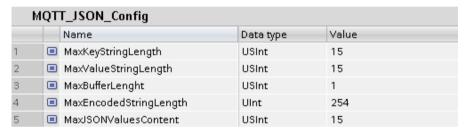
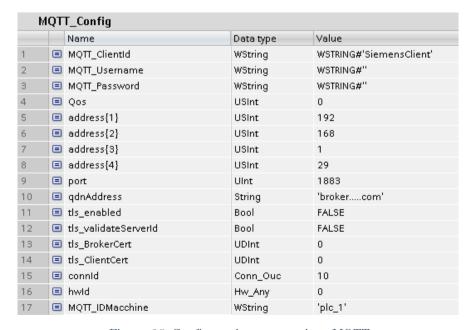


Figura 34: Configurazione JSON MQTT

Le configurazioni sottostanti riguardano i parametri relativi alla connessione



 $Figura\ 35:\ Configurazione\ connessione\ MQTT$

6.2.2 Integrazione PLC data types

I data types da creare sono i modelli ovvero strutture dati che servono ad allocare il dato prima del prelievo e invio al buffer.

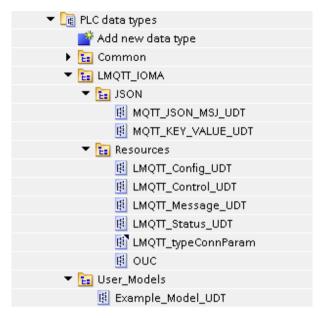


Figura 36: Integrazione Data Type

Nell' immagine sottostante vi è un esempio di modello il quale potrebbe riguardare le informazioni di un asse della macchina automatica.

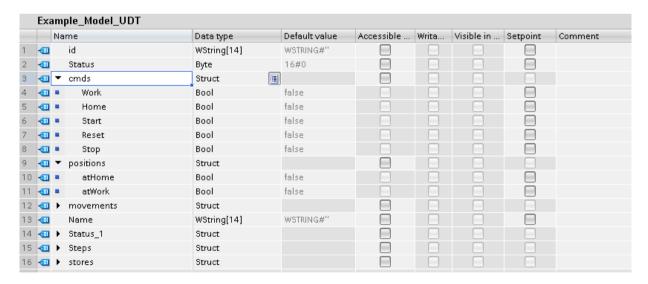


Figura 37: Esempio modello

6.2.3 Integrazione PLC Program blocks

La parte principale del programma si trova nel blocco funzionale MQTT comunication management. La quale non richiede nessuna modifica.

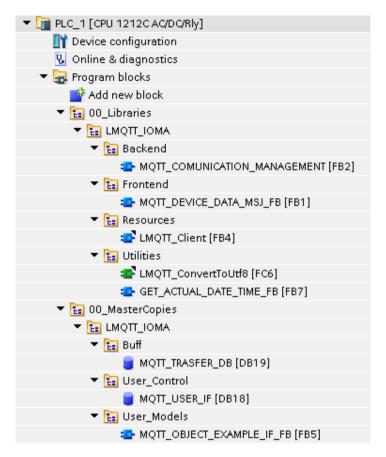


Figura 38: Integrazione Program Block

Per quanto riguarda il collegamento al programma invece sarà necessario completare la function block MQTT object example.

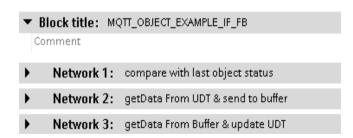


Figura 39: MQTT object example

Cap 7: Risultati e possibili applicazioni

7.1 Introduzione

L'obbiettivo che si pone questo capitolo è quello di far emergere nello specifico le potenzialità e i vantaggi che si delineano integrando la libreria di comunicazione nei vari scenari industriali.

Per fare ciò partiremo dal concetto di comunicazione machina server remoti per poi riassumere una serie di esempi pratici dove l'implementazione del protocollo torva riscontro positivo e apporta valore aggiunto ai fini aziendali.

7.2 Comunicazione macchina e server remoti

La comunicazione è lo scambio di messaggi tra due entità. Lo scopo della comunicazione è quello trasmettere gli altri una di una serie di informazioni, le quali acquisite da una seconda entità possono essere avere valore decisionale oppure valore esclusivamente informativo.

Come nel genere umano le macchine automatiche nutrono la necessità di comunicare ovvero di scambiare continuamente informazioni riguardanti il processo in evoluzione. Questo scambio avviene su più livelli di cui distinguiamo un livello locale e un livello remoto.

Il livello locale è caratterizzato da messaggi provenienti da dispositivi relativamente vicini all'unita centrale di elaborazione, per esempio una macchina, riceve il segnare di presenza pezzo da un sensore presente nel campo.

Il livello remoto invece è costituito da ricezione di informazioni da dispositivi che si possono trovare in fisicamente lontani rispetto alla macchina, per esempio tramite uno smartphone o un tablet è possibile visionare o inviare comandi a una macchina ubicata su uno stabilimento su un alto continente.

In generale il livello di comunicazione da remoto non va a sostituire il livello locale ma va a creare una tipologia differente di connessione alla macchina, tramite server remoti. I server remoti non sono altro che spazi virtuali nei quali è possibile raccogliere, elaborare dati e renderli disponibili da qualsiasi parte del modo noi ci troviamo.

Calandoci in una realtà aziendale questi spazi rappresentano luoghi virtuali nei quali è possibile creare delle connessioni e scambi dati abbattendo definitivamente la problematica della distanza. Poniamo il caso di una multinazionale la quale vuole conoscere in real time l'avanzamento della produzione o magari la presenza di guasti di un macchinario presente in un sito produttivo ubicato in un'altra nazione, la soluzione potrebbe essere chiamare il personale del sito produttivo e farsi spedire i report di produzione, oppure più semplicemente se la macchina è collegata ai server remoti è possibile accedere alla visualizzazione dei dati singoli e incrociati senza dover chiamare o richiedere per mail.

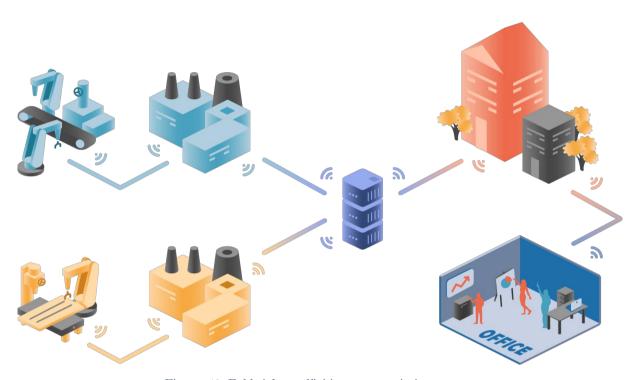


Figura 40: Fabbriche e uffici interconnessi ai server

Quello che si vuole introdurre è un modo alternativo di comunicare con le macchine più smart semplice ed efficace, una sorta di collegamento o affiancamento ad esse che introduce sinergie migliorando la collaborazione tra uomo e macchina, dimezzando tempistiche e abbattendo distanze.

Richiamando l'aspetto pratico dello scambio di dati tra macchine e server remoti esso è possibile grazie all'implementazione di protocolli di comunicazione. Nel particolare l'MQTT è uno strumento quindi per far comunicare macchine con altre macchine o dispositivi vari come i server remoti, affinché questo avvenga tutti devono connettersi alla rete che può essere internet o locale.

Nel proseguo riporteremo una varietà di casistiche nella quale il protocollo può essere integrato e gli aspetti positivi che introduce.

7.3 Possibili implementazioni e risvolti pratici sui vari settori

7.3.1 Liberia MQTT a servizio della personalizzazione della produzione

Il protocollo MQTT entra in supporto a quello che è il concetto di produzione agile e flessibile ovvero a una produzione automatizzata la quale può gestire senza alcuna problematica un mix differente di prodotti. Qui ricordiamo il concetto di ricetta ovvero una sequenza di movimenti che la macchina esegue per completare una lavorazione su un prodotto. Questo concetto è riferito a ogni singola macchina all'interno a una linea di lavorazioni.

Una volta che la linea è programmata a basso livello quindi il PLC riesce a cestire una serie di movimenti in autonomia e sicurezza, la connessione MQTT può andare a mandare la ricetta di lavorazione. Una volta che il pezzo entra in macchina si procede con la scansione del tag e il PLC manda al server remoto l'informazione identificativa alla quale corrisponde una ricetta che a quel punto viene trasmessa ed effettuata.

La grande differenza e la potenzialità della macchina interconnessa al server stanno nel fatto che se prima l'operatore doveva selezionare la ricetta a mano o per di più nel caso dell'entrata in linea di un nuovo prodotto nasceva la necessità di scaricare nuovamente la ricetta su memoria locale, ora la ricetta viene inserita tranquillamente con estrema semplicità e velocità.

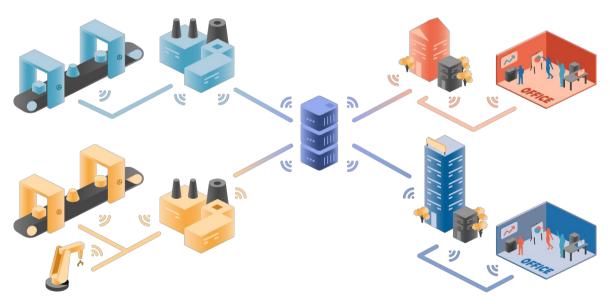


Figura 41: Esempio di personalizzazione della produzione da remoto

Quindi un responsabile può andare ad apportare una modifica software senza recarsi in loco facendo così evitare lato aziendale una serie di costi aggiuntivi derivate da eventuali trasferte o da fermi produzione fino a quando la ricetta non viene aggiornata.

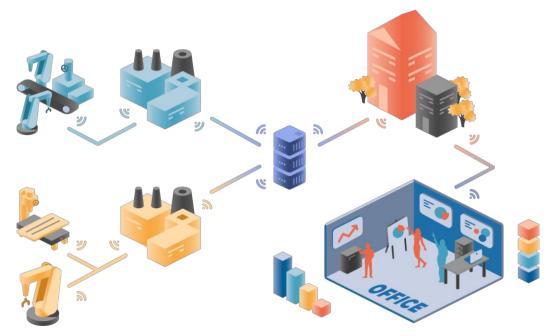
Un ulteriore aspetto che ci offre la connessione la possibilità della creazione e gestione del digital twin. Il digital twin non è altro che un modello gemello digitale, a seconda dell'ambito può avere differenti declinazioni. Ora nell'ambito della produzione, all'entrata in linea i PLC trasmetto al server remoto l'immissione di un nuovo prodotto da lavorare, successivamente all'identificazione in automatico si viene a creare un modello digitale sul quale su un lato si trovano le corrispondenti macchine sulle quali deve transitare e dall'alto lato vengono riportare come una sorta di check le operazioni fate e le caratteristiche specifiche della quale si vuole tenere memoria o venire a conoscenza. Alla fine del processo il prodotto verrà immagazzinato insieme al suo insieme di informazioni all'interno di un archivio. Quest'ultimo a disposizione dei vari reparti di competenza per effettuare le proprie analisi o ricerche.

7.3.2 Libreria MQTT a servizio di software gestionali

Il protocollo MQTT integrato all'interno della produzione può apportare un enorme supporto e tutto quello che è il comparto gestone e qualità.

L'analisi delle grandi quantità di dati raccolti e inviati al server in cloud, dai sensori del reparto di produzione, garantisce la visibilità in tempo reale degli indicatori KPI (Key Performance Indicators). Le organizzazioni utilizzano KPI a più livelli per valutare il loro successo nel raggiungimento di quanto prefissato.

Allo stesso tempo i dati provenienti dalla produzione vengono combinati con i dati operativi di ERP (Enterprise Resource Planning), conosciuto come pianificazione delle risorse d'impresa. L'idea è quella di mettere a disposizione i dati a un software di gestione che li integra con tutti i processi aziendali e tutte le funzioni aziendali rilevanti, ad esempio vendite, acquisti, gestione magazzino, finanza o contabilità. Integra quindi tutte le attività aziendali in un unico sistema per supportare al meglio il management. Attraverso tale sistema, i dati provenienti da molteplici parti dell'azienda vengono raccolti e gestiti in maniera centralizzata.



 $Figura\ 42: Esempio\ di\ comunicazione\ server\ remoti\ e\ software\ gestionali$

L'obiettivo finale della produzione è di migliorare la qualità del prodotto finale monitorando e controllando il processo utilizzato per produrlo. I costi per scarti, rilavorazione e difetti non rilevati, nonché le spese sostenute per ispezioni di qualità, hanno un impatto diretto e immediato sul profitto di un'azienda. In questo caso il protocollo MQTT consente di mettere a disposizione i dati ai software per analisi di qualità e controllo statistico di processo SPC, ottenendo così analisi statistiche per riorganizzare la frequenza delle ispezioni umane e guidare l'azienda nell'ottimizzazione della propria strategia di qualità.

Qualsiasi sia il campo gestionale a cui sono indirizzati i dati, la miglioria apportata sta nel fatto che essi siano centralizzati ed esportabili in maniera veloce sicura e facile. È immediato quindi immaginare come il manager che sta e Torino può visualizzare un dato aggregato appartenente al sito produttivo di un'altra nazione e con un singolo click può passare al dato singolo e vedere tutte le singole specifiche.

I dati possono essere messi a disposizione di un terzo per effettuare delle analisi ulteriori in maniera semplice e immediata. In un futuro con l'implementazione di macchine learning si potrebbero avere delle analisi sui dati e da lì predire delle devianze di produzione.

7.3.3 Liberia MQTT a servizio della manutenzione predittiva

La manutenzione predittiva è un tipo di manutenzione preventiva che viene effettuata a seguito dell'individuazione di uno o più parametri che vengono misurati ed analizzati utilizzando appropriati modelli matematici allo scopo di individuare il tempo residuo prima del guasto. Sono svariate le metodologie, come ad esempio analisi tribologiche sui lubrificanti, la misura delle vibrazioni, la termografia, l'analisi delle correnti assorbite, il rilievo di vibrazioni anomale e altre.

In questo contesto il protocollo MQTT si inserisce come tramite tra segnali rilevati e software di analisi predittiva. Alimenta quella che possiamo definire un'intelligenza artificiale che processa i vari dati secondo modelli, trova le corrispondenze di failure, stima lo stato di degrado e da qui la vita residua del componente.

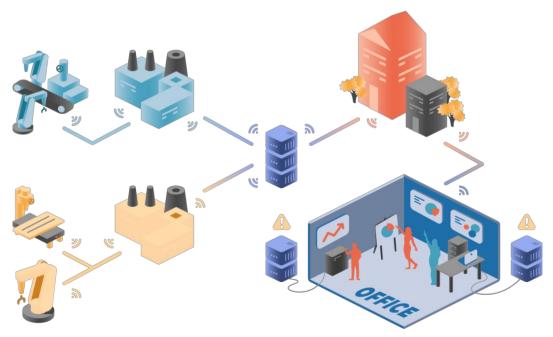


Figura 43: Esempio di comunicazione macchine con software AI per manutenzione predittiva

La manutenzione preventiva consente quindi di programmare i fermi macchina in tempi convenienti per la produzione, e di pianificare per tempo l'approvvigionamento di tutti i pezzi di ricambio e il reperimento del personale necessario. Tuttavia, questa strategia richiede un'accurata pianificazione e, se non ottimizzata, la frequenza degli interventi finisce per essere troppo elevata o troppo bassa, rispetto alle effettive necessità delle attrezzature.

Un altro strumento facilmente implementabile può essere quello di poter ricevere un avviso anche su smartphone ogni qualvolta all'interno della linea si verifica un'anomalia. Prendiamo l'esempio di una linea con vari macchinari, uno dei quali si arresta a causa di una semplice avaria macchina, dal momento che si arresta al momento in cui viene avvisato l'addetto per la riparazione l'informazione deve passare per più livelli, andando così aumentare il tempo di fermo macchina. Con il protocollo MQTT e con una semplicissima applicazione si ha la possibilità di visualizzare lo stato attuale delle macchine. Nel caso di avaria può far arrivare un avviso a tutte le persone di competenza, abbattendo così le tempistiche.

7.3.4 Libreria MQTT a servizio delle analisi e test sperimentali

Il plc possedendo dei tempi di campionamento molto bassi, potrebbe per questo essere sfruttato per analisi di tipo sperimentale in laboratorio. Un esempio potrebbe essere un processo fisico della quale vorremmo analizzare la sua dinamica. Collegando i sensori al PLC vi è la possibilità di campionare il dato e spedirlo real time in Cloud dal quale si possono creare delle interfacce grafiche oppure fruirlo in maniera pressoché istantanea a un centro di analisi distante dal laboratorio dove sta avvenendo l'esperimento o il test.

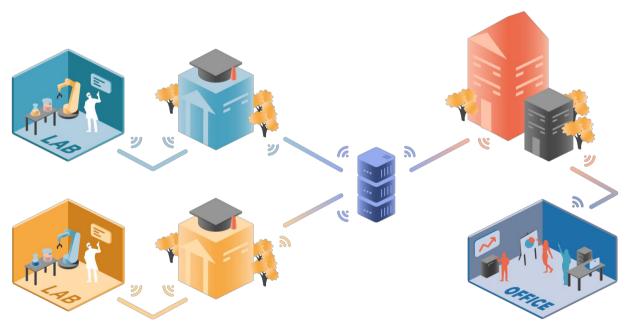


Figura 44: Esempi comunicazione test sperimentali e uffici vari

Cap 8: Conclusioni

Il risultato raggiunto è stata la realizzazione della prima versione della libreria di comunicazione. Allo stato attuale il programma può essere implementato su PLC, in parallelo al programma macchina ed è in grado di stabilire la connessione al broker remoto con il quale trasmettere e ricevere dati in Real time. I vari test confermano e delineano un'ottima affidabilità e velocità di invio anche in presenza di PLC carichi.

La libreria quindi rappresenta uno strumento per decentralizzare il dato della macchina automatica da locale ad uno spazio virtuale, semplificando e velocizzando così i tempi di accesso ai dati. A livello produttivo notiamo un aumento di efficienza e resa, questo deriva dal fatto che riusciamo a controllare il processo in real time e quindi dal punto di vista gestionale si possono visualizzare le deviazioni nel processo prima che esso esca dal range delle tolleranze.

L'implementazione della libreria non mette esclusivamente in collegamento solo una serie di macchine automatiche e uno o più server, ma permette risvolti pratici di interconnessione effettivamente tra due settori diversi quale l'automazione e l'informatica. Come corollario la connessione di macchine facilita altresì la comunicazione tra produzione e, controllo qualità e tutti gli altri dipartimenti che possono beneficiare di una maggior accessibilità al dato.

In conclusione l'elaborato presenta una soluzione applicativa di industria 4.0 che tramite la libreria si uniscono due settori di studi differenti automazione e IT al fine di per decentralizzare il dato a livello di macchina e centralizzarlo a livello di ditta. Permettendo di agevolare e migliorare sia la comunicazione tra dipartimenti tipicamente distanti e tutti quei processi aziendali che possono beneficiare della disponibilità del dato univoco e centralizzato.

Corollario del beneficio del dato centralizzato applicato a piccole PMI, è la facilitazione ad accedere a strumenti di analisi predittiva tipicamente riservati a grandi compagnie. Questo perché la centralizzazione del dato permette di sfruttare strumenti condivisi e di nuovo sviluppo, come analisi basata su algoritmi di autoapprendimento (Ai) ad un costo minore.

Bibliografia e Sitografia

Hwang, H. C., Park, J., & Shon, J. G. (2016). Design and implementation of a reliable message transmission system based on MQTT protocol in IoT.

Kraijak, S., & Tuwanut, P. (2015, October). A survey on internet of things architecture, protocols, possible applications, security, privacy, real-world implementation and future trends. In Communication Technology (ICCT), 2015 IEEE 16th International Conference on IEEE.

Siemens, MQTT Client for SIMATIC S7-1500 and S7-1200

https://support.industry.siemens.com/cs/document/109748872/use-the-simatic-controller-as-an-mgtt-client?dti=0&lc=en-WW

Flaviana Calignano, Appunti Sistemi di produzione, 2021.

Articoli web:

Perché si dice 4.0 e che cosa vuol dire 1.0, 2.0 e 3.0?

https://www.economyup.it/innovazione/perche-si-dice-4-0-e-che-cosa-vuol-dire-1-0-2-0-e-3-0/

Industria 4.0, una definizione per tutte

https://www.expomagazine.eu/2021/07/22/industria-4-0/

Cyber physical systems (CPS), cosa sono, come stanno rivoluzionando il mondo industriale

https://www.internet4things.it/industry-4-0/cyber-physical-systems-cps-cosa-sono-come-stanno-rivoluzionando-il-mondo-industriale/

Ten Years of Industrie 4.0

https://www.mdpi.com/2413-4155/4/3/26

The concise PLC History

http://www.machine-information-systems.com/PLC_History.html

The Evolution of PLCS

https://www.plctechnician.com/news-blog/evolution-plcs

Cyber hardware software

http://bigportale.altervista.org/cyberhw-sw/index.php/9-informatica/networking/3-protocolli-di-comunicazione-cosa-sono-tipi-ed-esempi

Elenco URL foto perese dal web:

- [a] https://www.innovaimpresa-cnaumbria.it/transizione-4-0/
- [b] Ten Years of Industrie 4.0: https://www.mdpi.com/2413-4155/4/3/26
- [c] https://laptrinhx.com/artificial-intelligence-what-it-is-and-how-is-it-useful-185134004/
- [d] https://www.pngwing.com/en/search?q=simatic+S5+Plc
- [e] https://www.pngegg.com/it/png-cgxyu/download
- [f] https://autoware.com/wp-content/uploads/2018/01/Il_Padre_dei_PLC.pdf
- [g] https://i0.wp.com/www.noeju.com/wp-content/uploads/2016/09/modicon084.jpg
- [h] https://us.profinet.com/profinet-rt-vs-irt/
- [i] https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/