IDA Lab
identification automation

UNIVERSITÀ DEL SALENTO

Progettazione e validazione di un prototipo IoT per la comunicazione con un macchinario industriale basato su protocollo MODBUS/TCP

A.A. 2018/2019

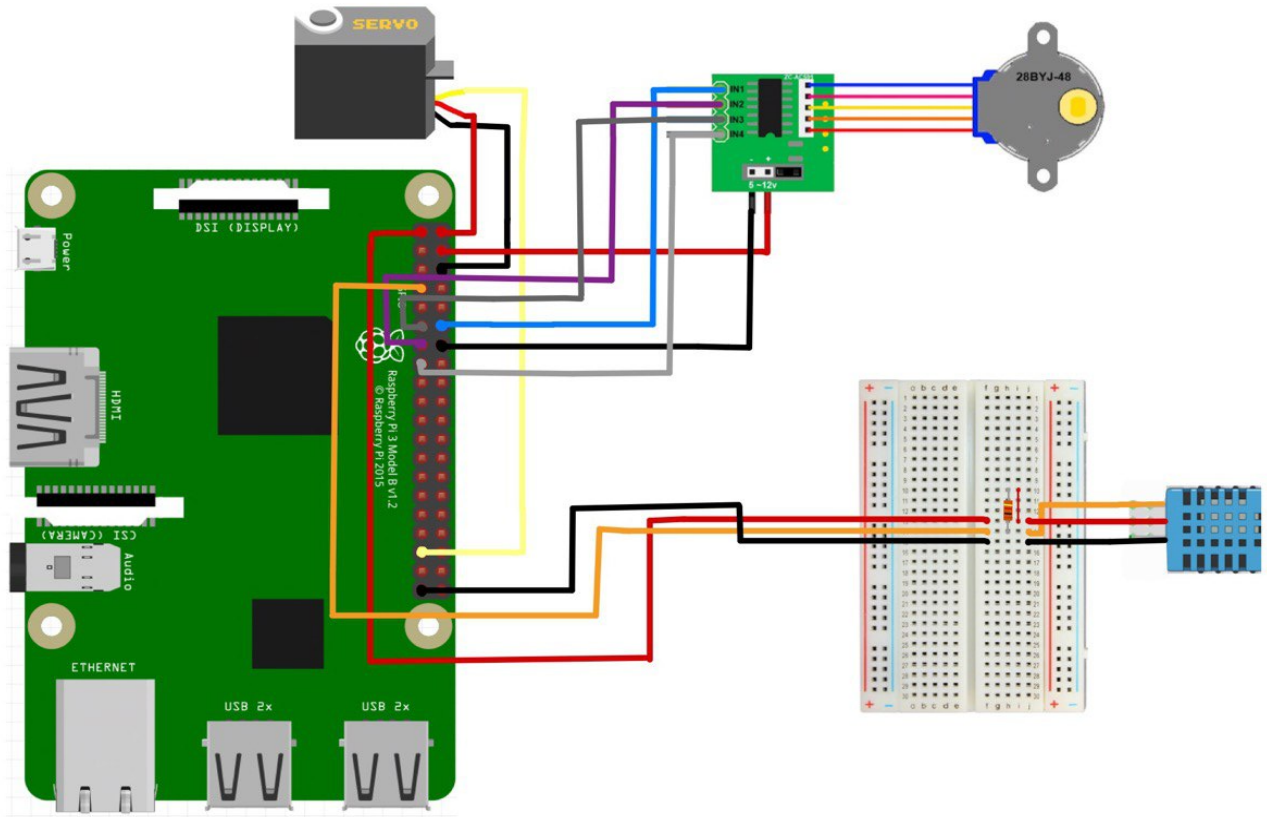
INTRODUZIONE

L'IoT è un neologismo che ha come scopo quello di rappresentare la connessione creatasi tra qualsiasi oggetto o apparato e Internet con lo scopo di fornire ed usufruire al tempo stesso di servizi e dati, è un paradigma che prevede l'utilizzo di oggetti intelligenti sparsi all'interno delle nostre case, al lavoro, nelle città, nella vita di tutti i giorni.
Esso si colloca in quel percorso nello sviluppo tecnologico in base al quale, attraverso la rete Internet, ogni oggetto dell'esperienza quotidiana acquista una sua identità nel mondo digitale.
Gli ambiti di utilizzo delle tecnologie abilitanti IoT sono molteplici e vanno dallo smart building, allo smart healthcare fino ad arrivare anche alla smart factory.
In campo industriale, infatti, le tecnologie denominate "Industrial IoT" hanno come scopo quello di facilitare l'ambiente di produzione connesso in cui la produzione stessa diventa sempre più efficiente e sono tra i protagonisti dell'evoluzione Industry 4.0 in cui un obiettivo fondamentale per poter digitalizzare la produzione industriale è quello di interconnettere gli apparati tra di loro e permettere una comunicazione.
Il principale obiettivo dei più importanti lavori in tale campo è quello di stabilire una comunicazione tra i macchinari industriali con il fine di introdurre nell'azienda/industria alcuni vantaggi: la sicurezza generale dei dipendenti, il monitoraggio remoto della macchina in tempo reale, l'aumento della produzione, la manutenzione predittiva, l'automazione, e l'acquisizione automatica dei dati.

OBIETTIVO

In questo contesto si inserisce il lavoro di tesi che ha come scopo quello di studiare l'interazione tra macchinari industriali con l'acquisizione di dati grazie al protocollo MODBUS/TCP.
Nel presente lavoro di tesi si procederà all'implementazione di un simulatore di macchinario industriale che permetta l'acquisizione di parametri attraverso un sensore, successivamente memorizzati grazie all'implementazione di un MODBUS/TCP Server e visualizzati grazie alla creazione di un'App Mobile che acquisisce i dati tramite REST API opportunamente tradotte in richieste accettabili dal MODBUS/TCP Server.

ARCHITETTURA



TECNOLOGIE

Le tecnologie utilizzate nella realizzazione del progetto sono:

1. Il protocollo MODBUS/TCP usato per la comunicazione;
2. Raspberry Pi;
3. DC Motor Servo Motor;
4. Step Motor;
5. Sensore DHT11;
6. pymodbusTCP;
7. Flask.

TECNOLOGIE

IL PROTOCOLLO MODBUS/TCP - INTRODUZIONE

La soluzione che permette la comunicazione machine-to-machine è senza dubbio il protocollo MODBUS, protocollo di comunicazione basato su una architettura client/server, ove un client avvia la trasmissione e il server può solo rispondere ad una richiesta specifica. L'obiettivo è facilitare una comunicazione affidabile e veloce tra dispositivo di campo e automazione garantendo una trasmissione dati ETHERNET ultraveloce ed una struttura dati indipendente dal produttore che consente la comunicazione tra dispositivi di produttori diversi.

PARADIGMA MASTER - SLAVE

MODBUS si basa sul paradigma Master/Slave, in cui il Master (client) manda delle richieste e lo Slave (server) le riceve e le soddisfa mandando una risposta al client. Inoltre se la parte slave è separata e si verifica un problema, il Master può diagnosticarlo e, una volta riparato il guasto, la rete può essere collegata automaticamente; pertanto il protocollo MODBUS è affidabile.

FUNZIONAMENTO

Esistono varie modalità operative di MODBUS: RTU, ASCII e TCP; Nel lavoro di tesi è stato utilizzato il protocollo MODBUS/TCP, il quale consente comunicazioni su reti internet/intranet.

Importante è il campo Fcode presente nel frame del messaggio, il quale contiene un numero in esadecimale a cui corrisponde un'area di accesso tra output coils, input coils, holding registers e input registers, e un'azione tra leggere, scrivere singole celle e scrivere più celle.

Coil/Register Numbers	Data Addresses	Type	Table Name
1-9999	0000 to 270E	Read-Write	Discrete Output Coils
10001-19999	0000 to 270E	Read-Only	Discrete Input Contacts
30001-39999	0000 to 270E	Read-Only	Analog Input Registers
40001-49999	0000 to 270E	Read-Write	Analog Output Holding Registers

Figura 1 Associazione tra aree di accesso, azioni permesse e indirizzo di riferimento

TECNOLOGIE

RASPBERRY PI

Il Raspberry Pi può essere considerato un computer in miniatura, un intero ecosistema hardware raccolto in un unico board; è considerato un dispositivo di fast-prototyping ovvero un dispositivo che permette la realizzazione di un prototipo in tempi molto rapidi.
Importante componente è il GPIO costituito da 40 pin a cui collegare diversi componenti seguendo l'opportuno datasheet associato al Raspberry Pi.

SERVO MOTOR

Un servomotore è un motore elettrico rotante che consente di controllare con precisione la posizione angolare, la velocità e l'accelerazione, permettendo di impostare il perno di trasmissione della rotazione su angoli esatti compresi in un certo intervallo che di solito va da 0 a 180 gradi.
Il servo motore presenta tre cavi, uno (nero) per l'alimentazione, uno (bianco) per la messa a terra e uno (rosso) per il passaggio dei dati e la sua rotazione si controlla attraverso un circuito di controllo che ne regola l'angolo e questo controllo viene ottenuto regolando la lunghezza di un impulso ad onda quadra inviato al servo motore la quale si definisce tramite il PMW (Pulse Modulation Width). Questo treno di impulsi si caratterizza dal duty cycle, cioè dal tempo occupato dall'impulso rispetto all'intero periodo destinato ad un singolo segnale.

STEP MOTOR

Lo step motor viene utilizzato per controllare la rotazione e la velocità precise con il controllo ad anello aperto., per questi motivi si trova in molte applicazioni delicate. Un altro vantaggio dell'utilizzo dei motori passo-passo è la loro capacità di mantenere il carico stabile una volta raggiunta la posizione richiesta e inoltre, sono ampiamente utilizzati in molte applicazioni industriali, specialmente nelle applicazioni che richiedono una posizione di controllo ad alta precisione.

DHT11

Il DHT11 è un sensore digitale di umidità e temperatura dell'aria costituito da una parte che si occupa della rilevazione dell'umidità e da un'altra parte che rileva la temperatura, entrambe gestite da un microcontrollore che è parte integrante del sensore.

TECNOLOGIE

LA LIBRERIA pymodbusTCP

La libreria pyModbusTCP fornisce l'accesso al server MODBUS/TCP tramite l'oggetto ModbusClient.. Importante è il modulo ‘utils’ che fornisce alcune funzioni e il modulo ‘client’ la cui classe ModbusClient presenta metodi utili per scrivere e leggere su Coils, su Registers, possibili Files e per ottenere delle informazioni diagnostiche.

FLASK

Flask è un framework Python per lo sviluppo di applicazioni web e viene fornito con funzionalità e requisiti minimi incorporati, e grazie a Flask si ha la libertà di personalizzare la maggior parte degli aspetti dell'applicazione, tra cui l'integrazione del database, gli account e l'autenticazione. Il modulo Python Flask contiene tutte le classi e le funzioni necessarie per la costruzione di una web app e per istanziare delle REST API.

Le API permettono a prodotti o servizi di comunicare con altri prodotti o servizi senza che sia necessario sapere come vengono implementati, semplificando così lo sviluppo delle app e consentendo un netto risparmio di tempo e denaro; con REST invece, vengono identificate delle risorse, ossia degli aggregati di informazioni che il servizio web offre; tali risorse vengono condivise in rete mediante protocollo che al suo interno possiede già tutto ciò di cui uno scambio di informazioni necessita pertanto non c’è bisogno di creare alcuna ulteriore sovrastruttura. Sfruttando gli elementi interni a questo protocollo potremo instaurare un dialogo tra il client ed il server. Il protocollo HTTP funziona mediante un modello richiesta/risposta: un client invia una richiesta al server, il server risponde. Importante è il campo metodo di HTTP al cui interno si inseriscono per di più quattro metodi GET, POST, PUT, DELETE.

DETTAGLI

Per la realizzazione del prototipo è stato istanziato e attivato un Server MODBUS/TCP grazie alla libreria pymodbusTCP impostando un socket con host di default e porta effimera 1502.

L'implementazione del Client consiste nella creazione di uno script in python costituito da varie funzioni ognuna delle quali istanzia un Client MODBUS/TCP che instaura una connessione e legge dei dati dal Server, successivamente, in base ai dati letti, controlla il movimento dei componenti o restituisce i dati.

È stato implementato il gateway attraverso la libreria Flask, grazie alla quale è stato istanziato un Client MODBUS/TCP che scrive i dati nei registri del Server e successivamente, in base all'azione selezionata, invoca la funzione apposita per leggere i dati e controllare i componenti.

L'azione opportuna viene selezionata grazie ad un'App Web implementata in HTML e collegata con il gateway istanziato precedentemente.

RISULTATI

Viene illustrato un caso d'uso in cui viene ruotato il Servo Motor di 90 gradi.
Operazioni di pre-test:

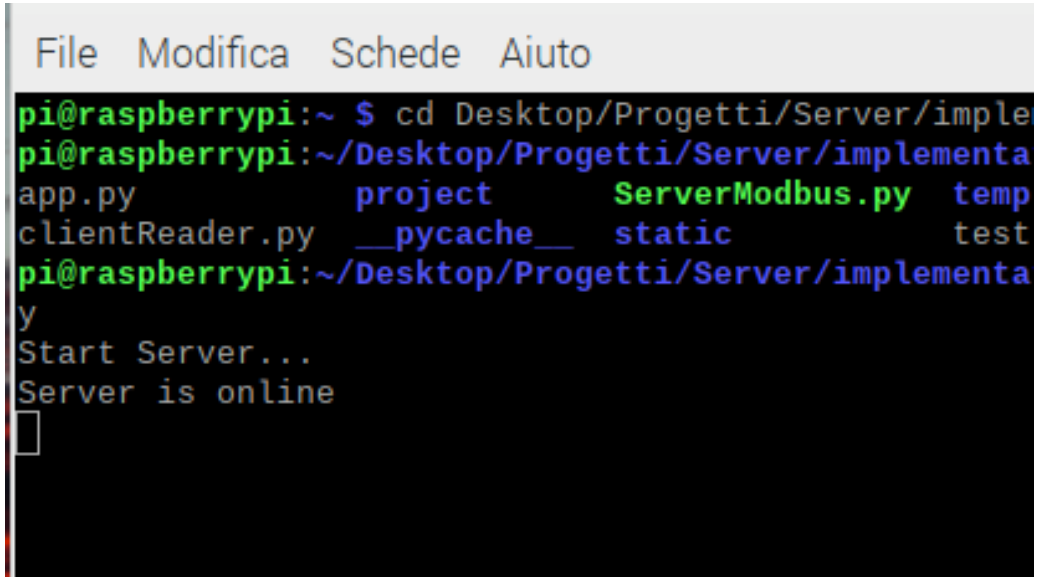


Figura 2 Attivazione del Server MODBUS/TCP

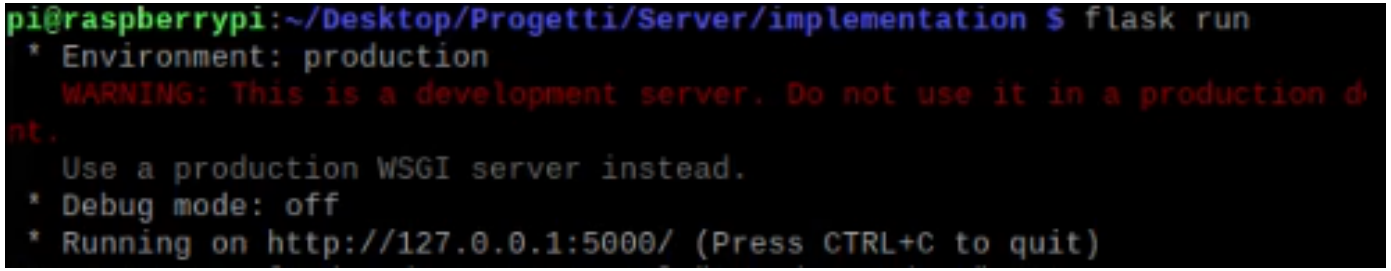
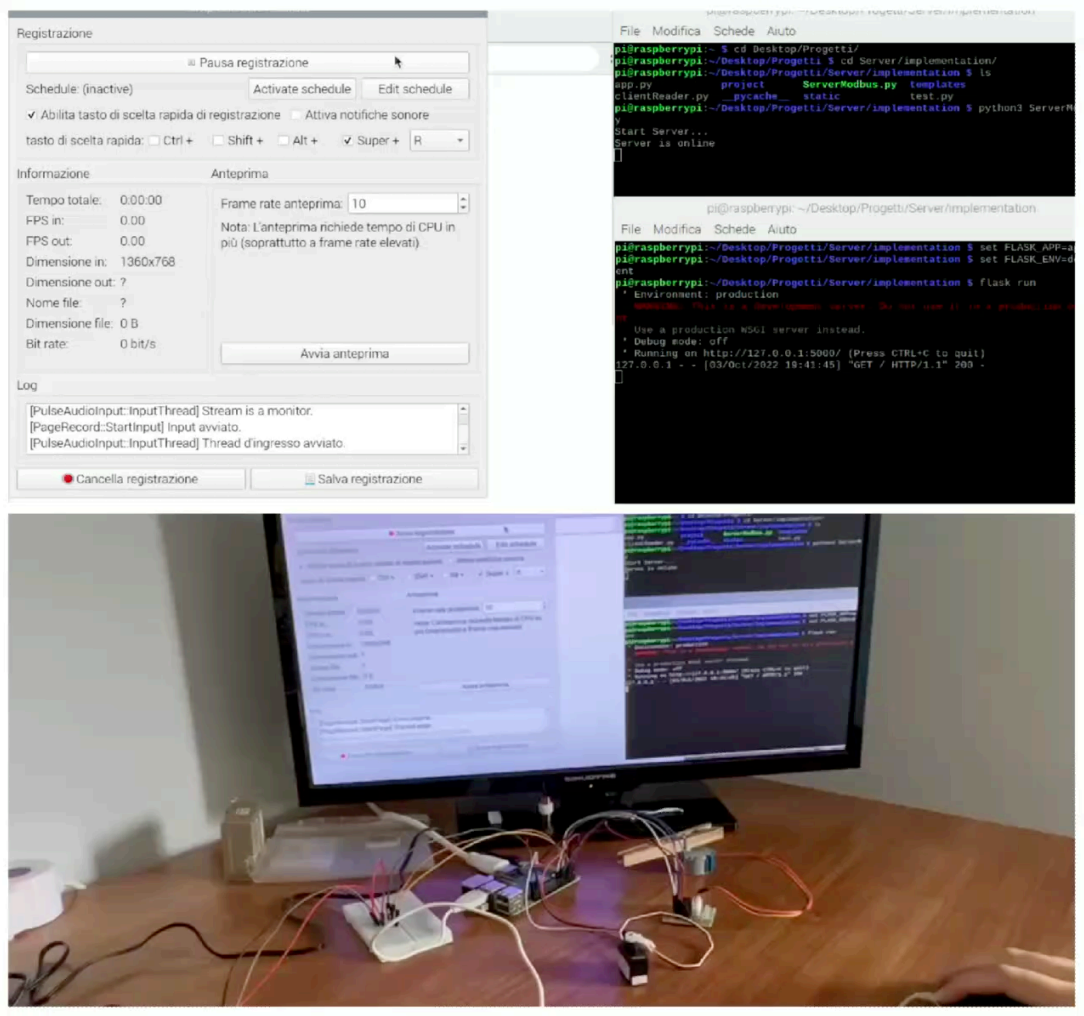


Figura 3 Attivazione dell'App Web sul link http://127.0.0.1:5000/

RISULTATI



Come si può vedere dal video, una volta avviata l'App W
basta scegliere l'azione desiderata, nel caso specifico
'rotazione di 90 gradi', per controllare il servo motor
e farlo ruotare dell'angolo scelto

CONCLUSIONI E SVILUPPI FUTURI

Il lavoro svolto nella presente tesi ha avuto come obiettivo l'esplorazione del mondo dell'IoT applicato al campo dell'industria, con lo scopo di far interagire e comunicare i macchinari industriali attraverso il protocollo MODBUS per far fronte alle varie problematiche che si possono affrontare in questo campo. Dopo aver analizzato alcune tecnologie già presenti che si basano sul suddetto protocollo, è stato elaborato un prototipo che permette di far interagire l'utente con dei componenti che simulano le macchine industriali e permette di ricavare parametri che caratterizzano l'ambiente circostante. Eventuali sviluppi futuri possono essere effettuati verificando l'utilità del prototipo presentato precedentemente, eventualmente inserendo il sistema all'interno del campo industriale; Inoltre si potrebbe migliorare l'implementazione software cercando di portare allo stato dell'arte l'app Web e migliorandola.