

---

**ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA  
SEDE DI CESENA**

**SECONDA FACOLTA' DI INGEGNERIA CON SEDE A CESENA**

**CORSO DI LAUREA MAGISTRALE IN  
INGEGNERIA ELETTRONICA E TELECOMUNICAZIONI PER LO  
SVILUPPO SOSTENIBILE**

**PROGETTO  
DI UN SISTEMA WIRELESS  
A MICROCONTROLLORE  
PER IL MONITORAGGIO  
DI CONSUMI ELETTRICI**

Elaborato in:

**Elettronica dei Sistemi Digitali L-A**

Relatore:

**Prof. Ing. Aldo Romani**

Correlatore:

**Dott. Matteo Filippi**

Presentata da:

**Mattia Paganelli**

---

Terza Sessione

Anno accademico 2011 / 2012

---



# INDICE

INDICE .....	3
INTRODUZIONE .....	5
1 IL SISTEMA .....	7
2 LA RETE DI RACCOLTA DATI .....	9
2.1 POWER MEASUREMENT UNIT .....	9
2.2 MICROCONTROLLORE.....	11
2.3 MODULO WIRELESS LOW POWER.....	11
3 LA RETE DI ARCHIVIAZIONE.....	13
3.1 MICROCONTROLLORE.....	14
3.2 MODULO WI-FI .....	15
3.3 ACCESS POINT .....	15
3.4 DATABASE.....	16
4 LE POLITICHE DI GESTIONE.....	17
4.1 FASE DI RACCOLTA DATI.....	17
4.2 FASE DI SCAMBIO DEI DATI FRA LE DUE RETI.....	20
4.3 FASE DI INVIO DEI DATI AL SERVER.....	21
5 REALIZZAZIONE DEL PROTOTIPO.....	25
5.1 L'HARDWARE .....	25
5.2 IL SOFTWARE.....	29
6 VALIDAZIONE E VERIFICA DEL PROGETTO .....	33
6.1 PRESTAZIONI DELLA RETE DI ARCHIVIAZIONE .....	33
6.2 PRESTAZIONI DEL PROTOCOLLO DI SCAMBIO.....	36
6.3 PRESTAZIONI DELLA RETE DI RACCOLTA .....	37
CONCLUSIONI.....	41
BIBLIOGRAFIA .....	43



## INTRODUZIONE

La presa di coscienza dei governi sui temi ambientali ed energetici ha portato ad una continua incentivazione, con conseguente diffusione, di impianti per l'autoproduzione da fonti rinnovabili [1].

Tali impianti sono per loro natura non programmabili, dato che dipendono dalle condizioni ambientali, e vengono solitamente collocati agli estremi della rete di distribuzione energetica rispetto alle grandi centrali.

Queste circostanze hanno comportato un allontanamento dal classico modello di produzione di energia elettrica centralizzato e controllato.

Mentre le reti di distribuzione tradizionali [2] prevedevano il dispacciamento solo per la parte ad alta ed altissima tensione ed un flusso unidirezionale dell'energia elettrica, i nuovi impianti produttivi rendono necessaria l'implementazione delle *smart grids*, reti "intelligenti" che permettono flussi energetici bidirezionali e sono in grado di adattarsi per far interagire in modo più diretto consumatori e nuovi produttori di energia.

Simili risultati sono possibili grazie all'utilizzo di tecnologie in appoggio alla rete elettrica, strumenti in grado di monitorare ed inviare alla centrale di controllo del distributore energetico i campionamenti temporali dell'energia consumata e/o autoprodotta dai singoli utenti.

Tale compito è affidato allo *smart meter*.

Ad oggi si parla di *smart meter* facendo comunemente riferimento al contatore elettronico [3], che permette il telerilevamento dell'energia utilizzata (e prodotta) dall'intera unità abitativa.

L'obiettivo che questo elaborato si pone è la realizzazione di un prototipo di *smart meter* distribuito che, in modalità Near-Real-Time, acquisisca ed invii ad un database remoto i consumi relativi ad ogni singolo elettrodomestico o, in generale, generico carico elettrico presente nell'abitazione.

In questo modo si rende disponibile all'utente un rilevamento dei consumi più dettagliato, mettendolo in condizione di eliminare gli sprechi e di risparmiare, aiutandolo nella scelta del piano tariffario.

Inoltre, in un eventuale contesto futuro di *smart buildings*, si può ipotizzare un sistema di questo tipo per dialogare con elettrodomestici intelligenti, rendendoli in grado di coordinarsi ed autogestirsi.

Il sistema ideato integra diversi protocolli wireless.

Una rete senza fili a corto raggio, i cui nodi sono attrezzati con Phasor Measurement Units (PMU), si occuperà delle rilevazioni energetiche e del loro invio al nodo centrale.

La comunicazione dei dati da inserire nel database, invece, verrà effettuata collegandosi ad un Wi-Fi Access Point (AI).

Tanto premesso, nei successivi paragrafi verranno trattate in dettaglio le singole componenti del sistema e saranno argomentate le principali scelte progettuali.

# 1 IL SISTEMA

Lo *smart meter* progettato sfrutta due diverse tipologie di reti.

La rete di primo livello ha il compito della raccolta dati, mentre la seconda si occuperà del loro salvataggio su database.

In seguito si farà riferimento alle due indicandole rispettivamente con “Rete di raccolta dati” o “Rete di primo livello” e “Rete di archiviazione” o “Rete di secondo livello”.

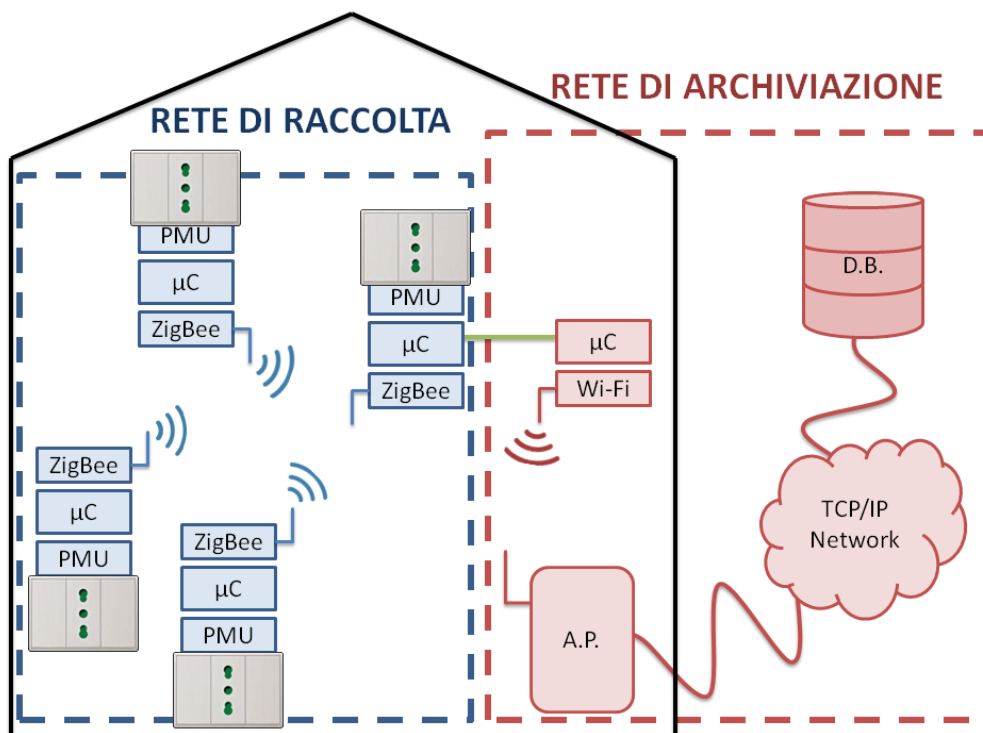


Figura 1.1: Rete di raccolta e di archiviazione dello smart meter progettato.

A ciascun elettrodomestico è assegnato un nodo composto da PMU, microcontrollore e ricetrasmittitore wireless. L'insieme di questi nodi costituisce la rete di raccolta dati wireless a corto raggio.

La PMU, connessa alla presa di corrente dell'elettrodomestico, campiona la potenza attiva assorbita dallo stesso. I dati così raccolti vengono letti dal microcontrollore, che li elabora e li trasmette via wireless al nodo centrale, seguendo precise politiche di gestione della rete.

Il nodo centrale sarà molto simile a quelli periferici (tanto che potrà essere a sua volta allacciato ad un elettrodomestico); tuttavia, questo nodo avrà il compito aggiuntivo di coordinare la rete di raccolta e di interfacciarla ad un nodo esterno.

Questo nodo comunicherà tramite Wi-Fi con un Access Point che gli permetterà di raggiungere il server: in questo modo, tutti i dati potranno essere salvati su database.

Il *core* dell'intero sistema sarà quindi costituito dalla coppia di microcontrollori – quello del nodo centrale della rete a corto raggio e quello che monta lo shield Wi-Fi –, i quali comunicheranno fra loro attraverso un protocollo seriale.

Ogni nodo verrà alimentato prelevando energia dalla linea elettrica alla quale è connesso.

Nei capitoli a seguire si vedranno con maggior dettaglio tutti i componenti dell'intero progetto.



## 2 LA RETE DI RACCOLTA DATI

La raccolta dei dati sui consumi energetici sarà affidata ad una rete wireless low-power a corto raggio.

Viste le ridotte dimensioni dell'ambiente da coprire (decine di metri) la tipologia di rete a stella appare essere la soluzione ideale.

Il progetto dei nodi che compongono tale rete è curato da UNIBO in collaborazione con STMicroelectronics ed è tuttora in fase di sviluppo.

Importanti specifiche sulla costruzione dei nodi riguardano il loro costo e la loro dimensione (dovranno essere sistemati all'interno della scatola elettrica a muro).

Come anticipato, ciascun dispositivo della rete sarà costituito da: PMU, microcontrollore e ricetrasmittitore wireless. Nei paragrafi che seguono verranno brevemente illustrati tali componenti.

### 2.1 POWER MEASUREMENT UNIT

La misura energetica è affidata all'*STPM01* [4], un circuito integrato di produzione STMicroelectronics che si presenta nella forma di package TSSOP20.

L' *STPM01* è un *energy meter* progettato per lavorare su reti monofase a 50-60 Hz. E' in grado di fornire la misura dell'energia attiva, reattiva e apparente ed i valori efficaci di tensione e corrente.

Tali grandezze vengono calcolate internamente a seguito del campionamento di tensione, corrente e loro sfasamento.

Il valore della tensione viene prelevato dalla linea di alimentazione del carico da monitorare grazie ad un sensore a partitore resistivo. La corrente, invece, può essere misurata attraverso un trasformatore di corrente, una bobina di Rogowski o uno shunt.

Nel nostro circuito verrà utilizzata la prima soluzione.

Internamente al componente, una coppia di convertitori di tipo  $\Sigma\Delta$  si occuperà della conversione A/D dei segnali provenienti dai sensori.

Questi serviranno al calcolo della potenza, dalla cui integrazione verrà ricavata l'energia. Queste operazioni saranno svolte da un DSP, anch'esso integrato nel componente.

Gli integratori sono implementati come contatori up/down a 20 bit, dunque possono subire *rollover*. Ciò obbliga il microcontrollore ad eseguire la lettura dei valori di energia almeno 10 volte al secondo. Per sicurezza, il manuale consiglia di eseguirne almeno 30 al secondo [5].

La comunicazione col microcontrollore avverrà attraverso l'interfaccia SPI, con velocità massima in lettura di 32 MHz (nel nostro caso sarà quindi limitata dalla SPI del microcontrollore a 18 MHz).

Per maggiori dettagli sull'*STPM01* si faccia riferimento al datasheet del componente.

Invece, per la circuiteria esterna (tra cui i circuiti di *power sensing*, *power supply* e l'oscillatore al quarzo) si consulti l'apposito *application note* [6].

## 2.2 MICROCONTROLLORE

Il “cervello” del nodo di *sensing* sarà costituito da un microcontrollore della famiglia *STM32F103xx*<sup>1</sup>, nel package LQFP64, con 128 kB di memoria Flash e 20 kB di SRAM [7].

L’integrato prodotto da STMicroelectronics dispone di diverse interfacce di comunicazione: I2C, USART, CAN, USB 2.0 (full speed) e SPI (18 Mbit/s).

Per questo progetto, una porta SPI verrà impiegata per leggere i valori dei consumi dall’*STPM01*, mentre un'altra porta seriale sarà collegata al modulo wireless. Limitatamente al caso del nodo centrale, sarà necessario allacciarsi anche al microcontrollore del nodo Wi-Fi per poter effettuare il trasferimento dei dati.

Si rimanda al datasheet dell’*STM32F103xx* per una descrizione più dettagliata del dispositivo. Si faccia invece riferimento all’apposito *application note* [8] per le istruzioni sull’interfacciamento con l’*energy meter* e la descrizione delle librerie firmware.

## 2.3 MODULO WIRELESS LOW POWER

Poter evitare la fase di cablaggio riduce sicuramente i costi di installazione di una rete. Per questo motivo si è scelto di utilizzare un canale di comunicazione wireless. Ovviamente saranno necessarie interfacce *low-cost*.

---

<sup>1</sup> Per la produzione dei nodi, al fine di contenere i costi, potrà essere effettuato un facile *porting* alla famiglia *STM32F101xx*.

Tale specifica può essere soddisfatta utilizzando dispositivi basati sullo standard IEEE 802.15.4 [9].

Lo standard regola il livello fisico e MAC di reti a corto raggio, dette “low-rate wireless personal area networks” (LR-WPANs), che operano sulle frequenze radio della banda libera ISM.

Lo standard prevede due diverse tipologie di nodo:

- RFD (Reduced Function Device): dispongono di funzionalità ridotte e possono operare solo ai margini della rete;
- FFD (Full Function Device): hanno piene funzionalità e possono dialogare con più di un dispositivo (coordinatori).

Nella rete in esame, il nodo del centro stella sarà certamente della prima tipologia, mentre i restanti nodi potranno essere dei semplici RFD.

Diverse *suite* di protocolli di comunicazione usano come base questo standard, differenziandosi nella definizione delle specifiche per i livelli ISO/OSI superiori.

Uno di questi protocolli in particolare verrà preso in considerazione per questo progetto: ZigBee.

Le sue specifiche prevedono un data-rate limitato a 250 kbit/s, non sufficiente a smaltire grossissime quantità di traffico.

La soluzione ZigBee potrebbe essere ideale per una applicazione di domotica, ma forse risulterebbe insufficiente per un ambito industriale.

Per sistemi dalle caratteristiche più spinte si potrebbe pensare di migrare al protocollo Bluetooth, basato su standard IEEE 802.15.1 [10]. Questo prevede, nella sua versione 2.0, un data-rate massimo di 3 Mbit/s, a scapito di costo e complessità.

Entrambe le soluzioni (e molte altre) possono essere considerate per la costruzione dello *smart meter* distribuito in esame.

In ogni caso, il *transceiver* dovrà essere controllato dal microcontrollore che risiede sul nodo. Per questa comunicazione verrà utilizzata una delle interfacce seriali disponibili sui pin dell'*STM32*.

### 3 LA RETE DI ARCHIVIAZIONE

Come noto, la seconda parte del sistema si occuperà del salvataggio delle informazioni raccolte su database.

I dati convogliati al nodo centrale dalla rete di primo livello verranno trasferiti sulla rete di archiviazione ad alta velocità. Ciò sarà possibile grazie alla comunicazione, tramite SPI, dei due microcontrollori posti sui nodi che fungono da interfaccia fra le due reti. Tali nodi saranno logicamente separati perché appartenenti a due reti diverse, ma saranno fisicamente collegati, costituendo dunque un *device* unico.

Supponendo che in un'abitazione che impieghi complessi sistemi di domotica sia comunque previsto un Access Point Wi-Fi, risulta conveniente sfruttarne la presenza per allacciare la rete di raccolta ad una rete di tipo TCP/IP (alla quale sarà collegato il server).

Tenuto conto che, per ogni sistema, dovrà essere prodotto un solo pezzo di questo tipo, il nodo che funge da interfaccia fra le due reti avrà specifiche sui costi molto meno restringenti. Potrà quindi essere utilizzato un ricetrasmittitore Wi-Fi.

Si vedrà nei prossimi capitoli che la scelta del database e delle sue modalità di accesso giocheranno un ruolo fondamentale nel determinare le prestazioni dell'intero sistema.

Nel proseguo si intende approfondire la descrizione dei componenti che costituiscono la rete di secondo livello.

### 3.1 MICROCONTROLLORE

Chi si prenderà cura di trasferire i dati dal nodo centrale della rete di raccolta al database è un microcontrollore di produzione Microchip, della famiglia *PIC32MX7XX* [11].

In particolare, per il montaggio su breadboard effettuato nella fase di prototipazione, si è utilizzato un *PIC32MX775F512H-80I/PT*. Dato il package 64-Pin TQFP, è stato necessario inserire un adattatore su PCB.

Per questa famiglia di dispositivi la casa produttrice rende disponibili, con licenza free, le librerie per lo stack TCP/IP.

Si è scelto questo particolare componente perché, integrando una CPU RISC a 32 bit e 80 MHz, fornirà prestazioni abbastanza elevate da avere la certezza che non diventerà il “collo di bottiglia” del sistema. Però, nel caso lo si ritenga opportuno, si potrà eseguire un facile downgrade alle famiglie *PIC24* o *PIC18*, anch’esse in grado di implementare lo stack TCP/IP.

L’integrato è munito delle seguenti interfacce:

I2C, UART, CAN, USB 2.0 (full speed), Ethernet e SPI (fino a 20 Mbit/s).

Una delle SPI verrà impiegata per comunicare con il modulo Wi-Fi, mentre l’altra sarà collegata all’*STM32* per effettuare il trasferimento dei dati (Nel *Capitolo 5.1* viene mostrato lo schema di interfacciamento con il modulo Wi-Fi, attraverso SPI).

Si consiglia di consultare il datasheet del componente per trovare tutte le informazioni che lo riguardano.

## 3.2 MODULO WI-FI

Sullo standard IEEE 802.11 per i livelli fisico e MAC, viene definito il protocollo Wi-Fi [12].

Il chip Microchip **MRF24WB0MA/B** [13] supporta, nello specifico, lo standard 802.11b, con transmission-rate di 1 Mbps e 2 Mbps.

Questo integrato, unito al microcontrollore che implementa lo stack TCP/IP, permette la comunicazione Wi-Fi, sulla quale possono essere impiegate le protezioni di tipo WEP, WPA-PSK e WPA2-PSK.

Nello specifico, si è utilizzata la versione del chip **MRF24WB0MA**, che prevede una antenna PCB integrata.

Il modulo può operare in diverse modalità, tra le quali quella che prevede una rete di tipo *Infrastructure Basic Service Set (BSS)*.

Con questa impostazione, i dispositivi Wi-Fi comunicheranno attraverso un router-Access Point wireless. Nel caso il router sia connesso alla WAN, il nodo Wi-Fi potrà accedere ad internet.

Per il test del circuito si è utilizzata la **MRF24WB0MA Wi-Fi PICtail/PICtail Plus Daughter Board** perché permette il montaggio su breadboard, ma nella versione definitiva su PCB del nodo si utilizzerà il solo chip **MRF24WB0MA**.

## 3.3 ACCESS POINT

Questo componente deve fare da tramite tra il nodo Wi-Fi ed il server contenente il database.

Il dispositivo utilizzato in fase di prototipazione è un comune Router per uso domestico che incorpora la funzionalità di Access Point Wi-Fi.

Come precedentemente anticipato il suo costo non sarà interamente imputabile al sistema poiché lo si considera già installato (per altri scopi) all'interno dell'abitazione.

Dato il basso bit-rate utilizzato dalla rete di raccolta, il traffico generato dal solo *power meter* sarà smaltito con assoluta facilità dal dispositivo, ma anche l'uso congiunto con traffico internet non limiterà le prestazioni del sistema.

Non è quindi necessario un apparecchio con particolari livelli di prestazione.

### **3.4 DATABASE**

In fase di prototipazione si è utilizzato un server web locale con annesso interprete PHP per l'esecuzione dinamica di codice, che si interfaccia ad un *database management system* MySQL. Nel prototipo sviluppato, tale server sarà connesso al router tramite cavo Ethernet. Tuttavia, sarà in generale sufficiente che sia collegato alla medesima rete IP del nodo.

Questo metodo offre prestazioni poco ottimali, in quanto il passaggio attraverso il linguaggio di *scripting* richiede tempi di elaborazione non trascurabili, ma è risultato facilmente implementabile e dunque adatto ad essere utilizzato nella fase di prototipazione del sistema elettronico.

Gli inserimenti su database vengono fatti inviando al server pacchetti HTTP con metodo POST.

Per i test sui tempi di risposta si è invece utilizzato un server Telnet, decisamente più veloce, sul quale veniva eseguita la stampa a video o il salvataggio su file dei valori ricevuti in formato XML.



## **4 LE POLITICHE DI GESTIONE**

Nel presente capitolo si intendono argomentare le principali scelte implementative affrontate nella progettazione del sistema.

Il flusso di dati verrà diviso in tre fasi essenziali:

- Fase di raccolta dati;
- Fase di scambio dei dati fra le due reti;
- Fase di invio dei dati al server.

### **4.1 FASE DI RACCOLTA DATI**

In merito alla fase di raccolta dei dati, si è dettata una politica di minimizzazione del traffico.

Siccome i nodi sono dotati di microcontrollore, essi potranno elaborare le misurazioni effettuate prima di inviarle su rete wireless.

E' dunque possibile variare la frequenza di invio dei dati da parte del nodo di raccolta, a seconda della variabilità della grandezza misurata.

Si fissano dunque due intervalli temporali, che indicheremo da qui in poi come:

- **Tmin**: intervallo minimo di tempo che può intercorrere tra due invii consecutivi da parte di un nodo;
- **Tmax**: intervallo massimo di tempo che può intercorrere tra due invii consecutivi da parte di un nodo.

La specifica su Tmin può essere ricavata tenendo conto della massima capacità di smaltire il traffico della rete, quella su Tmax invece è a discrezione della risoluzione temporale massima che si vuole adottare per il salvataggio dei consumi.

Il nodo di sensing interroga il PMU ad intervalli di tempo che sono un ordine di grandezza inferiori rispetto a Tmin. Questo gli permette di calcolare una derivata locale dell'energia.

Tale valore verrà utilizzato come peso per trovare l'intervallo di tempo **Tinv** che il nodo dovrà aspettare prima di inviare il dato in rete.

Dunque:

$$Tinv|_t = d[E_t] \cdot K$$

Con:

$E_t$ : energia istantanea

$K$ : costante temporale di progetto

Però, considerando le specifiche, questo intervallo avrà limite massimo e minimo rappresentati da Tmin e Tmax.

Quindi:

$$Tinv\ effettivo|_t = \begin{cases} Tmin & se\ Tinv|_t < Tmin \\ Tinv|_t & se\ Tmin < Tinv|_t < Tmax \\ Tmax & se\ Tinv|_t > Tmax \end{cases}$$

Con questa politica, i nodi che registrano una forte variabilità delle misure verranno aggiornati più velocemente, mentre quelli con consumi relativamente costanti non occuperanno la rete per trasmettere informazioni di minore utilità.

Ovviamente, il dato contenuto nel pacchetto dovrà contenere l'intera energia consumata dall'elettrodomestico nell'intervallo di invio.

Siccome il PMU fornisce solamente l'integrale della potenza tra due diverse letture, spetterà al microcontrollore effettuare il calcolo dell'energia consumata fra due invii consecutivi.

Il valore massimo di questo dato influirà sul numero minimo di bit col quale dovrà essere rappresentata l'energia.

Noto dal manuale dell'*SPM01* [5] che, nel caso peggiore, la variabile a 20 bit subirà *rollover* in un tempo di 1,6 secondi. Da una semplice proporzione si deduce che una variabile a 24 bit non verrà riempita prima di 25,6 secondi.

Dunque, se il valore di  $T_{max}$  non supera questa soglia, 24 bit saranno sufficienti a contenere il dato dell'energia.

Siccome le informazioni relative ad un certo intervallo di tempo saranno salvate solamente in un pacchetto, sarà necessario che il sistema garantisca un flusso di dati affidabile, affinché questa logica di invio non influisca sulla corretta rilevazione dei consumi.

Occorre quindi che entrambe le reti si assicurino della avvenuta ricezione dei pacchetti.

Infatti, se un pacchetto relativo ad un elettrodomestico andasse perduto, il sistema interpreterebbe la mancata ricezione come indice della stabilità del vecchio valore di energia assorbita. Si andrebbe in questo modo a registrare valori dei consumi energetici non corretti.

Scelta la politica con la quale si genera il traffico sulla rete, non rimane che scegliere il protocollo che si occuperà di trasportarlo.

Come già specificato, diverse tecnologie sono state vagliate per assolvere questo compito e si sta valutando la possibilità di realizzare diversi prototipi.

Tuttavia, i protocolli che saranno principalmente presi in considerazione sono ZigBee e Bluetooth.

A titolo di esempio, consideriamo in modo più approfondito una di queste tecnologie: il protocollo ZigBee.

Esso è largamente utilizzato per applicazioni distribuite, per via dell'ottimo compromesso fra prezzo del ricetrasmittitore e data-rate effettivo.

I nodi ZigBee possono raggiungere distanze che variano fortemente con la tipologia di ambiente.

Considerando un contesto abitativo *indoor*, ogni nodo arriverà a coprire una distanza di almeno una decina di metri [14], quindi si riuscirà a coprire aree di almeno 200 mq con una semplice configurazione a stella (per ambienti più spaziosi si dovrà ricorrere alla tecnica del *multihopping* oppure utilizzare più di un nodo Wi-Fi). La stella è la rete che garantisce la minima latenza.

Il nodo centrale dovrà essere di tipo FFD e dovrà svolgere il compito di coordinatore della rete. Gli altri nodi potranno essere anche di tipo RFD.

Si possono prevedere diverse procedure di gestione della stella, ma vista la quantità di dati in gioco e la natura intermittente degli stessi, la modalità che nel caso di rete non congestionata dovrebbe essere in grado di offrire prestazioni maggiori, è quella *non-beacon-enabled*.

Un nodo che ha bisogno di trasferire un dato al coordinatore può accedere al canale utilizzando il protocollo CSMA/CA. Se non avvengono collisioni, ed il nodo centrale riceve correttamente il pacchetto, risponderà con un *acknowledge*.

## **4.2 FASE DI SCAMBIO DEI DATI FRA LE DUE RETI**

L'interfacciamento fra il nodo centrale della rete a corto raggio ed il nodo Wi-Fi avviene attraverso protocollo SPI.

La ragione di questa scelta risiede nel fatto che su entrambi i microcontrollori interfacciati era disponibile questo tipo di porta e che permette frequenze di trasmissione elevate.

La scelta poteva ricadere anche sul protocollo USB, che è stato tuttavia escluso in ragione del diverso voltaggio richiesto dal protocollo rispetto a quello presentato dai microcontrollori. L'utilizzo di questo protocollo avrebbe pertanto richiesto di aggiungere al circuito un traslatore di livello, non necessario per l'SPI.

### 4.3 FASE DI INVIO DEI DATI AL SERVER

Per dialogare col server potevano essere considerate almeno due diverse tecnologie ad alta velocità.

Una *power line communication (PLC)* avrebbe permesso il collegamento senza la stesura di altri cavi, poiché utilizza come mezzo trasmissivo i conduttori della rete elettrica. Però sarebbe stato necessario mantenere il server col database all'interno dell'edificio monitorato, ad esempio in prossimità del quadro contatori dell'edificio, dove tutte le utenze elettriche si diramano.

Sfruttando invece l'Access Point presente nell'abitazione, si potrà accedere ad un qualsiasi server collegato ad internet. Anche in questo caso si può evitare il cablaggio, utilizzando un AP con Wi-Fi.

L'adozione del protocollo di trasporto TCP è giustificata dalle scelte fatte in fase di raccolta dati. Infatti, come già accennato, esse hanno come conseguenza diretta, la necessità di disporre di una rete capace di trasmettere dati in maniera affidabile.

Sopra al livello IP, questo servizio è offerto dal protocollo TCP [15].

Affinché i dati trasmessi in Wi-Fi presentino minor overhead, il PIC cercherà di accumulare il maggior numero di informazioni in un unico pacchetto. Ovviamente, tale procedura incontrerà un limite nell'obiettivo prefisso di trasmettere i dati in modalità Near-Real-Time. E' possibile tuttavia prevedere forme di time-out per effettuare trasmissioni nel caso in cui questo costituisse un limite.

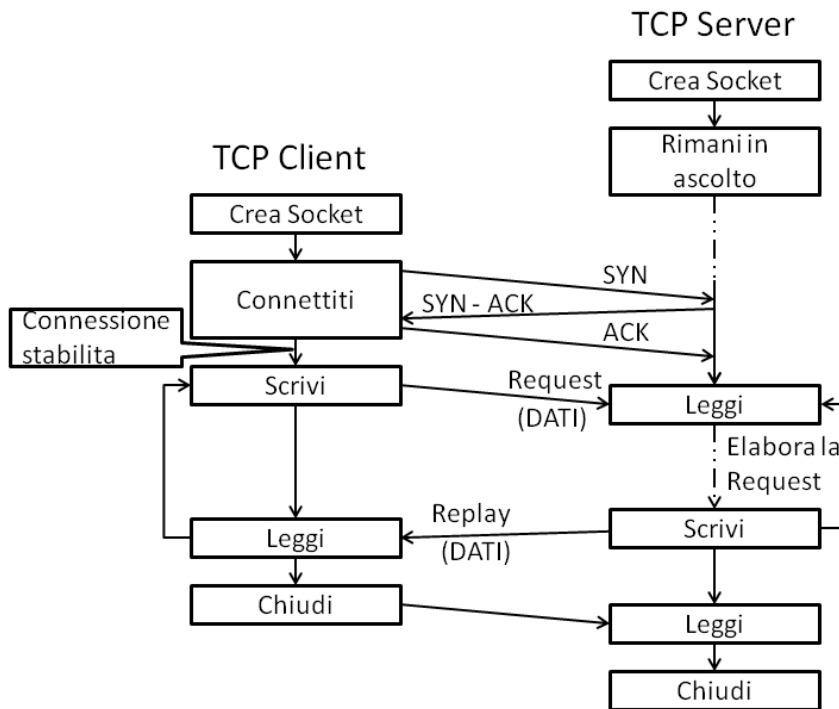
Infine, sempre col fine di risparmiare traffico, si è scelta una specifica politica di gestione della connessione TCP/IP.

In particolare, si è disposto che il socket, una volta aperto, non venga chiuso prima di un certo tempo.

Se il PIC non invia alcun ulteriore dato, il socket verrà chiuso allo scadere del termine stabilito, al fine di liberare risorse scarsamente utilizzate. In caso contrario il time-out verrà rinnovato ed il termine ricomincerà a decorrere dal principio.

In questo modo, quando il PIC invia dati a cadenza ravvicinata, si evita di ripetere continuamente la procedura di apertura del socket, che prevede uno scambio di pacchetti TCP, risparmiando così tempo e traffico. Nel caso però in cui il client non abbia dati da inviare, il socket verrà disattivato dopo pochi secondi, evitando così di occupare risorse sul server.

Questa procedura è dettagliatamente descritta in *Figura 4.3.1*.



**Figura 4.3.1: Diagramma temporale della comunicazione fra Client e Server TCP**

La gestione dei socket lato client è assegnata al microcontrollore PIC.

Nel *Capitolo 5.2* si potrà consultare lo schema della macchina a stati finiti impiegata dal client TCP.

Le operazioni sulle socket sono state effettuate sfruttando le librerie per lo stack TCP/IP messe a disposizione da Microchip.

Per semplificare la stesura del codice è stato utilizzato un socket di tipo bloccante. Socket non bloccanti potrebbero essere utilizzati per aumentare l'efficienza delle fasi di comunicazione e ridurre i relativi tempi. Tuttavia, poiché allo stato attuale il collo di bottiglia del sistema è collocato esternamente al sistema progettato, risulta una scelta valida l'uso di socket bloccanti.





## 5 REALIZZAZIONE DEL PROTOTIPO

Per poter verificare che le scelte progettuali fino ad ora illustrate permettano di soddisfare le specifiche, è necessario realizzare un prototipo del sistema.

La realizzazione dei componenti della rete di primo livello è affidata ad UNIBO in collaborazione con STMicroelectronics, dunque verrà di seguito trattata solamente la prototipazione del nodo che offre il collegamento Wi-Fi con l'AP.

Come ormai ben noto, questa parte di sistema si interfacerà alla rete di raccolta per ricevere i dati sui consumi dell'abitazione, per poi eseguire i salvataggi sul server.

Il nodo dovrà quindi svolgere il compito di client TCP/IP.

Il circuito è stato montato su breadboard ed il firmware caricato sul microcontrollore PIC è stato scritto in linguaggio C.

### 5.1 L'HARDWARE

In *Figura 5.1* viene mostrato lo schematico che tratta l'interfacciamento del microcontrollore Microchip *PIC32MX775F512H-80I/PT* con il modulo Wi-Fi.

La comunicazione avviene attraverso protocollo SPI.

Nello schema sono indicati anche i componenti utilizzati per interagire col sistema in fase di test (pulsanti, LED e display LCD).

Per dare un'idea delle dimensioni dei componenti, si mostra anche una immagine fotografica del circuito montato su breadboard (*Figura 5.2*).

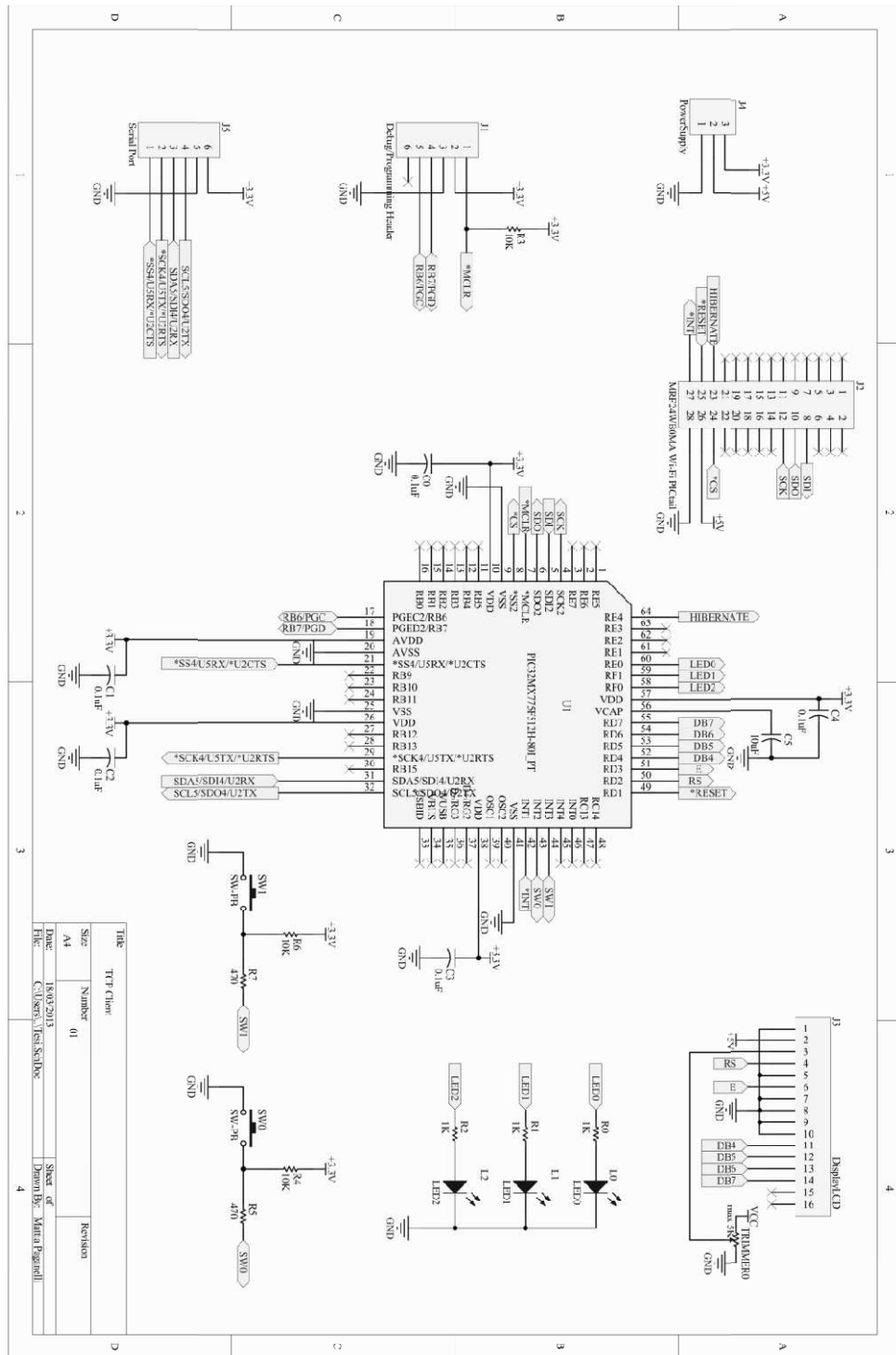


Figura 5.1.1: Schematico del circuito di test.

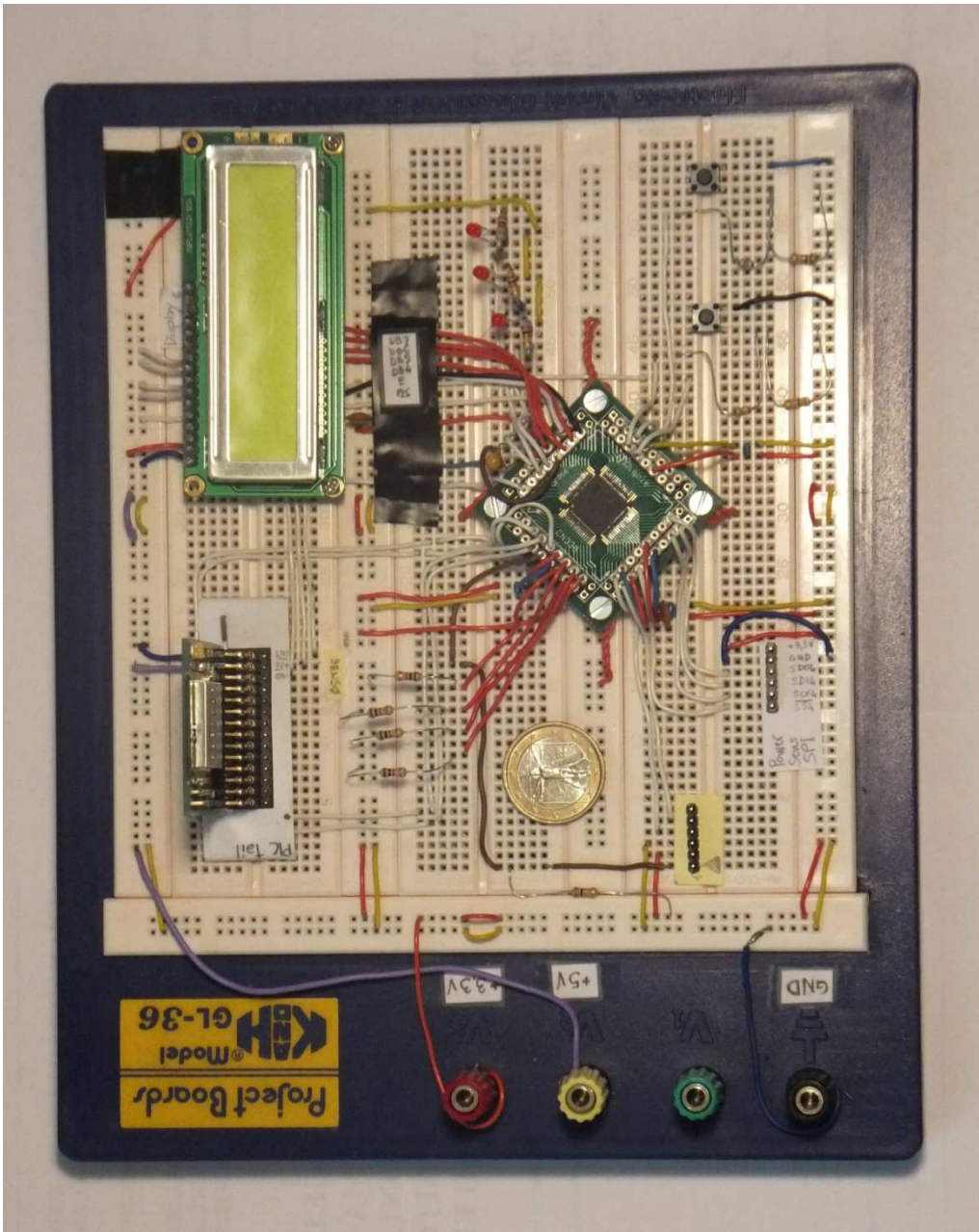


Figura 5.1.2: Circuito di test montato su breadboard.

## 5.2 IL SOFTWARE

Per la stesura del codice che sfrutta lo stack TCP/IP, si sono utilizzate le librerie messe a disposizione da Microchip.

Il modello di esecuzione dello stack prevede che, conclusa la fase di inizializzazione, il programma entri in un *loop* infinito, nel quale vengono eseguiti in continuazione tutti i task relativi all'implementazione dello stack. Anche le procedure riguardanti l'applicazione sviluppata dovranno essere inserite all'interno di questo ciclo.

E' previsto che tali processi siano cooperativi, dunque questo meccanismo impone che ogni task ritorni al *main program* entro tempi brevissimi, in modo da lasciare la CPU libera di eseguire le altre procedure.

Per rispettare queste specifiche sarà necessario frammentare le procedure più lunghe.

Il codice che implementa il client TCP dovrà tenere conto di questo aspetto, quindi è stato scritto seguendo alcuni accorgimenti.

In *Figura 5.2.1* è rappresentato lo schema della macchina a stati finiti (MSF) che regola le fasi di dialogo col server.

La chiamata che manderà in esecuzione questo processo è stata collocata all'interno del ciclo infinito del *main program*, quindi verrà lanciata in modo ripetitivo.

E' importante costruire gli stati della MSF in modo che eseguano brevi operazioni e poi ritornino.

Solo in questo modo lo stack potrà essere gestito correttamente, poiché la CPU avrà la possibilità di essere equamente assegnata ai task cooperativi.

La gestione delle socket viene eseguita dalla MSF come descritto nel *Capitolo 4.3*.

A seguito dell'inizializzazione del sistema, che prevede la fase di connessione all'AP, il client TCP si troverà nello stato di "Wait" (W). Qui rimarrà in attesa della richiesta di trasferimento dei dati.

Quando questi saranno disponibili sul nodo centrale della rete di raccolta, verrà eseguito il trasferimento con protocollo SPI.

Recuperati i dati verrà effettuata la richiesta di connessione al server, reiterandola fino a che non sarà ottenuta.

Non appena la connessione viene accettata si invierà un pacchetto TCP contenente i consumi raccolti.

Conclusa la trasmissione ci si troverà in uno stato clone del “Wait” iniziale. Questo sarà necessario per escludere una nuova procedura di apertura connessione nel caso si debba inviare entro breve un nuovo pacchetto.

Dunque, se entro lo scadere del Time Out il sistema avrà necessità di spedire altri dati, si sfrutterà la connessione lasciata aperta. Altrimenti verrà chiusa e lo stato tornerà ad essere il “Wait” di partenza.

Una volta qui, per effettuare un nuovo trasferimento, sarà necessario richiedere una nuova connessione.

Questa procedura è stata implementata allo scopo di liberare risorse sul server se queste sono poco utilizzate.

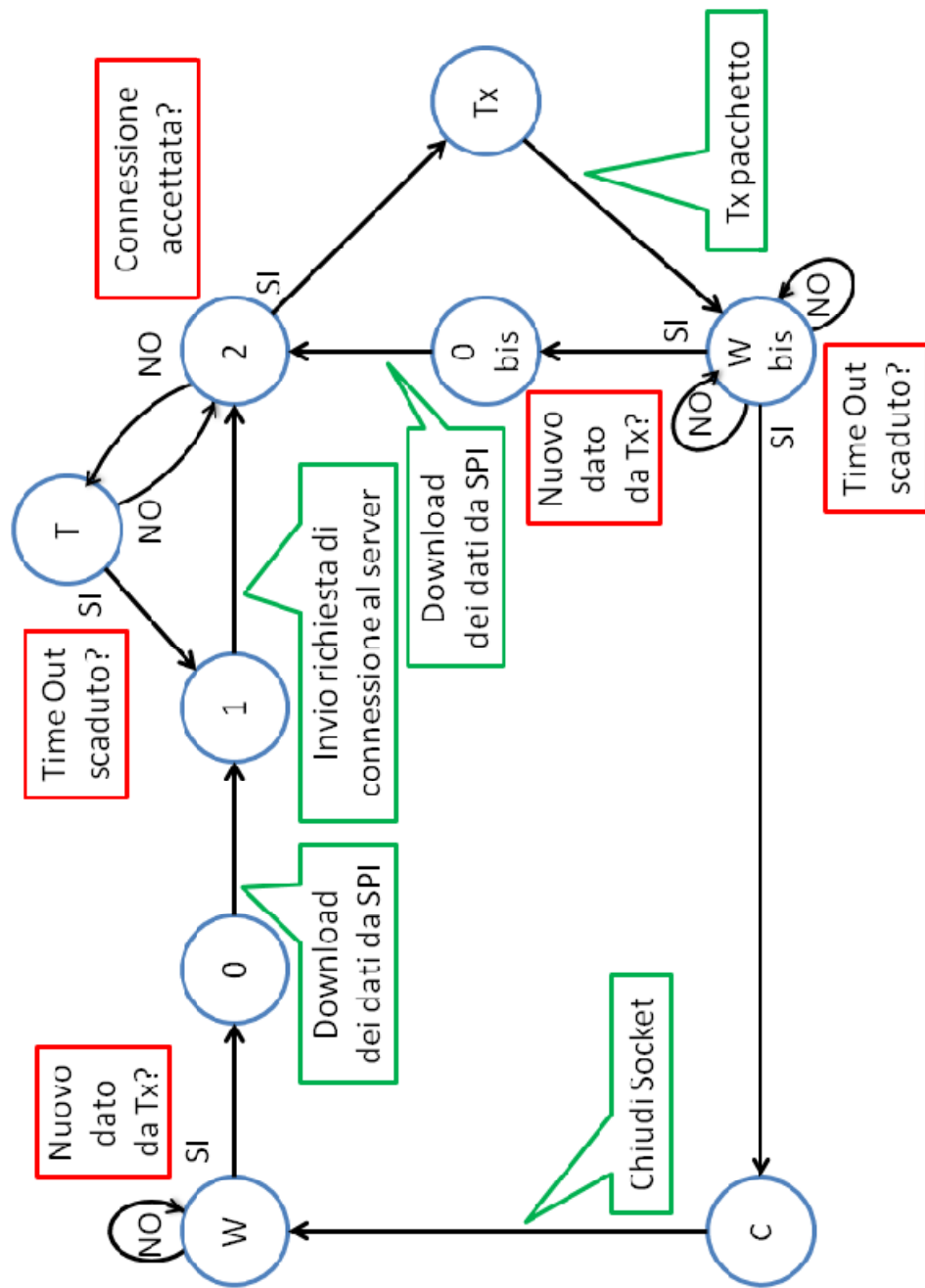


Figura 5.2.1: Macchina a stati finiti implementata dal Client TCP.





## **6 VALIDAZIONE E VERIFICA DEL PROGETTO**

Realizzato il sistema, è necessario verificare che rispetti le specifiche imposte. Può però risultare interessante testarne anche le potenzialità.

Spingere il sistema al limite permetterà di individuarne le criticità ed aiuterà a capire su quali punti si potrà agire per poter soddisfare specifiche sempre più restrittive.

Del nostro sistema, purtroppo, si è potuta testare solo la rete di secondo livello, poiché i nodi della rete di raccolta sono ancora in fase di sviluppo.

### **6.1 PRESTAZIONI DELLA RETE DI ARCHIVIAZIONE**

Le prime prove sono state eseguite in locale, effettuando le scritture su database attraverso l'interprete PHP.

Sul microcontrollore del nodo Wi-Fi è stato caricato un programma appositamente scritto in C che implementa un client HTTP

Tale client, una volta aperto il socket, invia a ripetizione dei pacchetti HTTP indirizzati alla porta 80 del server. Questi pacchetti, contengono i campi ed i valori da inserire nelle righe del database relazionale.

Lo script PHP, estrapolate le informazioni necessarie dal pacchetto HTTP, eseguirà la vera e propria operazione di INSERT nel database

E' bene ricordare che, affinché il server mantenga attiva la connessione dopo la ricezione del pacchetto, occorre specificare il metodo CONNECTION con l'informazione "Keep-alive".

La richiesta al server usa il metodo POST, da preferirsi a GET, in quanto non limita il numero di caratteri da inserire.

Con questa configurazione si sono registrati tempi di inserimento nel database dell'ordine delle centinaia di millisecondi.

Considerato che il nodo Wi-Fi può integrare in un unico pacchetto i dati raccolti dall'intera rete di primo livello, si desume che i dati sui consumi dell'abitazione verranno aggiornati alcune volte al secondo.

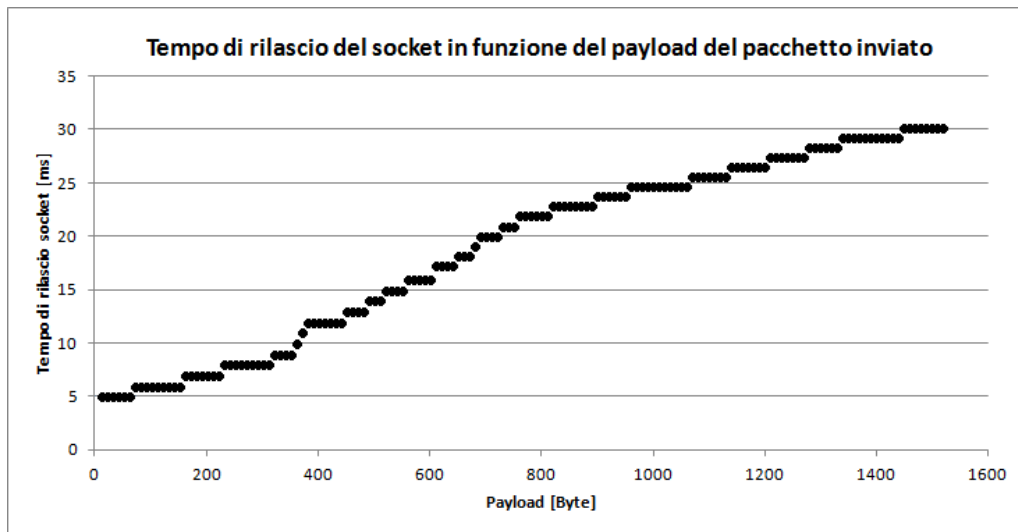
Essendo un sistema che deve interagire con l'uomo, queste tempistiche sono più che sufficienti per soddisfare le specifiche. Però, ci si aspettava che non fosse questo collegamento la parte limitante del sistema, bensì la rete di raccolta.

Dunque, per approfondire il problema, si sono svolti altri test utilizzando un server telnet.

Si è quindi configurato il client per inviare, sulla relativa porta 23, dei dati in formato XML che il server dovrà salvare su file di testo.

Con queste impostazioni si sono registrati netti miglioramenti. Sono quindi imputabili ai ritardi introdotti dall'interprete PHP le scarse prestazioni rilevate con la precedente configurazione.

I risultati ottenuti con server Telnet sono nell'ordine delle decine di microsecondi, come si può notare dal grafico in *Figura5.1.1*.



**Figura 5.1.1: Tempo di rilascio della socket in funzione del payload del pacchetto inviato.**

Si evince dunque che la fase di accesso al database dovrà essere affrontata con cura, perché potrà influire molto sulla capacità del sistema di smaltire i dati.

In un ottica di miglioramento del progetto, si potrebbe pensare di realizzare la scrittura su database direttamente dal client, bypassando quindi l'interpretazione PHP ed il web server stesso.

Se invece si volesse svincolare completamente il sistema dalle tempistiche di risposta del server, basterebbe utilizzare, al posto del TCP, il protocollo UDP. Tale modifica però porterebbe ad avere un collegamento non affidabile, ovvero non si avrebbe la certezza dell'effettivo arrivo a destinazione dei datagrammi. Con queste premesse occorrerebbe tuttavia cambiare le politiche di raccolta dati, perché quelle precedentemente descritte richiedono la condizione di certa ricezione per poter ricostruire in modo corretto l'andamento dei consumi.

Una nuova politica adatta all'utilizzo di UDP potrebbe, ad esempio, prevedere l'invio dell'intera energia raccolta fino al momento della trasmissione, anziché solo quella utilizzata tra due successivi intervalli.

Con questo metodo, la perdita di un pacchetto influenzerà solo la frequenza di aggiornamento dei consumi, ma non causerà alcun errore sull'energia complessivamente misurata.

Si dovrà però aumentare la dimensione della variabile contenente il valore dell'energia<sup>2</sup>.

Concludendo, se la perdita di dati non dovesse influenzare il corretto funzionamento del sistema, si potrebbe prendere in considerazione anche l'impiego del protocollo UDP.

Un'altra controindicazione all'uso del protocollo UDP si avrebbe nel caso di saturazione e congestione del mezzo trasmissivo, condizioni che rendono più probabile la perdita di pacchetti dati da parte degli apparati di rete.

## 6.2 PRESTAZIONI DEL PROTOCOLLO DI SCAMBIO

Per quanto riguarda la comunicazione fra le due tipologie di reti, si può far notare che la frequenza di clock del protocollo SPI sarà limitata dal microcontrollore *STM32* a 18 MHz (visto che il *PIC32* renderebbe possibile una comunicazione fino a 20 MHz).

Il Bus dati sarà molto corto, dato che i componenti sono fisicamente molto vicini, dunque non si avranno problemi di attenuazione dei segnali.

La scelta della tipologia di socket utilizzata nella comunicazione col server influirà anche su questa fase. Per semplicità, sul prototipo sono state

---

<sup>2</sup>Una variabile a 36 bit potrà contenere l'intero valore di energia consumato in una giornata dall'elettrodomestico. Sono 12 bit in più rispetto al caso considerato precedentemente. Anche per una rete a basso data-rate è un aumento di traffico più che trascurabile.

utilizzate socket di tipo bloccante, ma la modalità non bloccante porterebbe sicuri miglioramenti.

Con questa modifica, infatti, una volta inviato il pacchetto, verrebbe restituito il controllo al programma, cosicché il *PIC32* potrebbe tornare ad effettuare i trasferimenti seriali mentre aspetta la risposta del server.

### 6.3 PRESTAZIONI DELLA RETE DI RACCOLTA

I nodi che compongono la rete di primo livello non sono ancora disponibili, dunque non è stato possibile testare fisicamente questa rete.

Ci si è quindi accontentati di effettuare una valutazione a grandi linee del numero di dati che una ipotetica implementazione sarebbe in grado di gestire.

I calcoli effettuati utilizzano metodi approssimati, quindi non permettono di determinare con accuratezza il limite della rete, ma solamente una stima di prima approssimazione della sua capacità.

Si consideri per la seguente trattazione una rete ZigBee, con topologia a stella *non-beacon-enabled* e composta da  $N$  nodi che richiedono ciascuno l'invio di  $M$  pacchetti al secondo.

Ipotizzando che il dato inviato da ogni nodo sensore sia composto da 24 bit contenenti il campionamento dell'energia e da 64 bit di variabile temporale<sup>3</sup>, il *payload* di ogni pacchetto occuperà 11 byte.

---

<sup>3</sup> Non è necessario esplicitare anche l'identificativo del nodo poiché potrà essere ricavato dall'indirizzo MAC del pacchetto.

Noti questi dati, è possibile calcolare in modo semplicistico la quantità di traffico gestibile dalla rete, così da poter stimare quali siano i massimi valori per  $N$  ed  $M$ .

Per determinare la dimensione dei pacchetti, si è fatto riferimento alle specifiche dello standard IEEE 802.15.4 e si è considerato il *payload* già calcolato.

Le lunghezze massime di “DataFrame” e “AckFrame” saranno rispettivamente di 336 e 88 bit.

Ora, in modo empirico, (nella speranza di modellare una rete molto congestionata) si ipotizza che andrà a buon fine solo una trasmissione su 10.

Dunque ogni nodo, per effettuare una trasmissione corretta (senza collisioni), dovrà spedire 10 volte lo stesso pacchetto, occupando il canale per un tempo pari a:

$$\begin{aligned} \text{Tempo tot di trasmissione e ritrasmissione} &= \\ &= 10 \cdot (\text{lunghezza DataFrame}) \cdot \text{tempo di bit} = 13.4[\text{msec}] \end{aligned}$$

$$\text{con: tempo di bit} = 1/\text{datarate} = 1/250000 [\text{sec}]$$

A questo tempo va aggiunta l'occupazione del canale da parte del centro stella per l'invio del relativo Ack:

$$\begin{aligned} \text{Tempo di trasmissione dell'Ack} &= \\ &= (\text{lunghezza AckFrame}) \cdot \text{tempo di bit} = 0.4[\text{msec}] \end{aligned}$$

In questo modo si otterrà:

$$\text{Tempo di invio di un dato corretto} = 13,8[\text{msec}]$$

Dunque, per non avere completa congestione della rete, ci si dovrà trovare nella condizione di avere  $N$  nodi ed  $M$  invii al secondo tali per cui:

$$N \cdot M \cdot 13,8[\text{msec}] < 1 [\text{s}]$$

Si conclude quindi che una rete con  $N=50$  potrà tranquillamente permettere ai suoi nodi tempi di invio inferiori al secondo:

$$50 \cdot 1 \cdot 13,8[\text{msec}] = 0.69 [\text{s}]$$

La stima effettuata è sicuramente peggiorativa, in quanto la rete non potrà trovarsi in condizione di forte congestione, poiché il traffico offerto è più di 10 volte inferiore alla capacità massima della rete.

Per stime più accurate si dovrà ricorrere a metodi analitici più complessi, basati sulla teoria delle code, oppure si dovranno effettuare delle simulazioni al calcolatore.

Si è quindi riuscito a mostrare che, anche per reti di buone dimensioni, sarà possibile far registrare al sistema un andamento dei consumi con risoluzione temporale inferiore al secondo.

Tempistiche più che sufficienti per soddisfare le specifiche che un sistema di questo tipo richiede.





## CONCLUSIONI

Nel presente elaborato si è realizzato un prototipo di *smart meter* di nuova concezione.

In questo sistema il rilevamento dei consumi non viene effettuato per l'intera unità abitativa (come avviene oggi con il contatore elettronico), ma si sposta al livello del singolo elettrodomestico, permettendo un più dettagliato controllo dei consumi.

Per la raccolta dei dati si sono sfruttate tecniche di trasmissione wireless low-power, che prevedono l'uso di ricetrasmittitori low-cost. In questo modo, i nodi sensore saranno di facile installazione e l'intero apparato avrà costi e consumi facilmente compensabili con i benefici che esso stesso apporterà in termini di risparmio energetico.

Ritenendo altamente probabile la presenza di una rete Wi-Fi nella casa "intelligente" di un prossimo futuro, si è scelto di sfruttare questo protocollo per salvare i dati raccolti su un server TCP.

Il compito di client TCP per la comunicazione col server sarà dunque svolto da un microcontrollore con shield Wi-Fi.

Questo componente è stato programmato in linguaggio C, sfruttando le librerie che implementano lo stack TCP/IP.

Test ed analisi hanno determinato che il sistema qui progettato è in grado di rispettare le specifiche che un sistema di questo tipo prevede.

Tuttavia, verifiche prestazionali, hanno rivelato quali potessero essere i punti critici del sistema, aiutando a capire quali componenti o procedure dovranno essere modificate per rendere il progetto ancora più prestante.

In particolare, si è palesato che, qualora i salvataggi su database avvengano con procedure snelle, il collo di bottiglia dell'intero sistema sarà determinato dalla rete di raccolta.

Dunque, allorché venissero definite specifiche più restrittive, basterà sostituire le interfacce della rete low-power con dispositivi più prestanti (ma di conseguenza più costosi).

La modularità del progetto aiuta ad ottenere un sistema versatile che ben si presta ad integrare nuove funzionalità.

Sviluppi futuri del progetto, infatti, potranno aggiungere la mansione di rilevazione guasti o addirittura completarlo rendendolo in grado di controllare il dispositivo elettrico al quale è collegato.

Con queste modifiche, il sistema assumerebbe a tutti gli effetti un ruolo attivo nella sicurezza domestica e nella lotta agli sprechi energetici.

## BIBLIOGRAFIA

- [1]. Parlamento europeo, «Approvato il pacchetto clima-energia, obiettivo: 20/20/20» 17 dicembre 2008.  
<http://www.europarl.europa.eu/sides/getDoc.do?type=IM-PRESS&reference=20081216IPR44857&language=IT>
- [2]. Autorità per l'energia elettrica e il gas, «Delibera ARG/elt 160/11 - Avvio di procedimento per la formazione di provvedimenti in materia di regolazione del servizio di dispacciamento» 18 novembre 2011.  
<http://www.autorita.energia.it/it/docs/11/160-11arg.htm>
- [3]. Enel Distribuzione, «Il contatore elettronico» Accesso: 17 marzo 2013.  
[http://www.enel.it/it-IT/reti/enel\\_distribuzione/qualita/progetti\\_contatore\\_elettronico/](http://www.enel.it/it-IT/reti/enel_distribuzione/qualita/progetti_contatore_elettronico/)
- [4]. STMicroelectronics, «STPM01 - PROGRAMMABLE SINGLE PHASE ENERGY METERING IC WITH TAMPER DETECTION» Giugno 2011.  
[http://www.st.com/web/en/catalog/sense\\_power/FM1963/SC397/SS1214/PF81930](http://www.st.com/web/en/catalog/sense_power/FM1963/SC397/SS1214/PF81930)

- [5]. STMicroelectronics, «UM0125 - STPM01 USER MANUAL (Single Phase with CT and Shunt) - Release 1.0», pag.15
- [6]. STMicroelectronics, « AN2317 - Application Note (STPM01 Programmable, Single-Phase Energy Metering IC External Circuits)» Aprile 2006  
[http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application\\_note/CD00091951.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/CD00091951.pdf)
- [7]. STMicroelectronics, « STM32F103xx datasheet» Aprile 2011  
<http://www.st.com/web/en/resource/technical/document/datasheet/CD00191185.pdf>
- [8]. STMicroelectronics, « AN2799 - Application note : Measuring mains power consumption with the STM32x and STPM01», Novembre 2008  
[http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application\\_note/CD00203032.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/CD00203032.pdf)
- [9]. IEEE Standard Association, « IEEE Std 802.15.4», 2011  
<http://standards.ieee.org/about/get/802/802.15.html>
- [10]. IEEE Standard Association, « IEEE Std 802.15.1», 2005  
<http://standards.ieee.org/about/get/802/802.15.html>
- [11]. Microchip, « PIC32MX5XX/6XX/7XX Family Data Sheet», 2011  
<http://ww1.microchip.com/downloads/en/DeviceDoc/61156G.pdf>

- [12]. IEEE Standard Association, « *IEEE Std 802.11* », 2012  
<http://standards.ieee.org/about/get/802/802.11.html>
- [13]. Microchip, « *MRF24WB0MA/MRF24WB0MB Data Sheet* », 2011  
<http://ww1.microchip.com/downloads/en/DeviceDoc/70632B.pdf>
- [14]. Verdone R., Dardari D., Mazzini G., Conti A., « *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design* », Londra: Academic Press, 2008
- [15]. Tananebaum A. S., Wetherall D. J., « *Computer Networks* », edizione 5, Boston: Prentice Hall, 2011