



**POLITECNICO
DI TORINO**

**Collegio di ingegneria Elettrica
Corso di laurea in ingegneria elettrica**

**Prova finale di laurea
a.a. 2015-/2016**

**DOMOTICA:
normative e protocolli**

Relatore
Prof. Napoli Roberto

Candidato
Gentileschi Matteo
s203546

Sommario

- **Introduzione.....pag. 2**
- **Vantaggi e svantaggi.....pag. 3**
- **Suddivisione.....pag. 4**
- **Breve descrizione del progetto.....pag. 4**
- **Principali differenze tra impianti domotici e tradizionali.....pag. 5**
- **CEI 64-8 (V3) – particolarità della versione 3 di tale normativa.....pag. 5**
- **Campioni e misure di energia termica (MAX35101)pag. 6**
- **UNI EN 15232 –“Prestazione energetica degli edifici.
Incidenza dell’automazione, della regolazione e della gestione tecnica degli
edifici”pag. 7**
- **Alcuni schemi elementari di impianti domotici.....pag. 8**
- **Lo standard KNX (Architettura).....pag. 11**
- **Arduino Due.....pag. 13**
- **TCM320.....pag. 14**
- **Lo standard EnOcean.....pag. 16**
- **Dispositivi Eltako.....pag. 21**
- **Configurazioni dei dispositivi.....pag. 22**
- **Circuiti elettrici.....pag. 23**
- **Software.....pag. 23**
- **Studio dei dispositivi.....pag. 25**
- **il FBH65S-wg (sensore di luminosità e di presenza): utilizzo come sensore di
luminosità.....pag. 24**
- **Comunicazione tra TCM320.....pag. 26**
- **Pannello domotico.....pag. 29**
- **Fonti.....pag. 31**
- **Rigraziamenti.....pag. 31**

Introduzione

Il termine “Domotique”, neologismo introdotto dai francesi, tra i primi ad occuparsi (in termini temporali) di tale settore, discende dal latino “domus” (“casa”) e da “Robotica” ed indica la scienza atta ad coniugare le tecnologie più moderne alla vita sociale dell’uomo, mediante l’integrazione dell’elettronica agli impianti civili e industriali, consentendo di migliorare l’efficienza energetica dell’edificio insieme alla sicurezza.

I primi passi in tale direzione sono stati fatti concretamente nella seconda metà del 900 automatizzando l’accensione e lo spegnimento delle luci e con l’introduzione di sistemi di allarme: un interruttore crepuscolare o un sensore di presenza sono ormai componenti elettrici comuni sul mercato.

L’automazione dell’impianto contribuisce a correggere le cattive abitudini dell’essere umano che portano a consumi non necessari provvedendo ad una gestione ottimale con forti risparmi energetici che porta al miglioramento dell’efficienza energetica dell’edificio.

Un ulteriore punto di forza sta nel fatto di poter modificare il comportamento di un impianto senza modificarne la topologia ad esempio si può eliminare una presa o un punto luce semplicemente disabilitandoli, oppure creare degli “scenari” ovvero delle situazioni domestiche ideali, senza costi aggiuntivi.

Consideriamo ad esempio un ufficio di ampia superficie con più punti luce, in prossimità delle finestre se fuori splende il sole non ha senso mantenere accese al 100% le lampade, con appositi sensori di luminosità distribuiti è possibile miscelare luminosità naturale ed artificiale in modo da creare un ambiente omogeneo addirittura risparmiando energia.

Si tratta di una logica distribuita, ovvero non c’è un dispositivo che accentra la logica del sistema bensì ogni dispositivo è munito di RAM, ROM ed EPROM (dove è caricato il firmware); gli attuatori sono separati galvanicamente dai comandi (aumento sicurezza) e comunicano tra loro mediante un segnale codificato con appositi standard(KNX, ENOCEAN ecc) , ma la vera differenza con un impianto “tradizionale” risiede nella semplicità di implementare logiche più articolate della “negazione” e quindi “acceso-speinto” , ad esempio se si vuole accendere in diverse combinazioni le lampadine a seconda del numero di volte che si preme un pulsante, cosa che risulta banale in domotica ma assai complesso con semplici interruttori.

Non si tratta di un mero miglioramento del comfort casalingo, ma (andando sul filosofico) un modo elegante per migliorare la qualità della vita, sia in termini di impatto ambientale che in termini economici e di sicurezza.

È interessante notare come entrino in contatto due realtà tecnologiche, quella open source, rappresentata ad esempio da Arduino e quella tradizionale, rappresentata ad esempio da Eltako ed EnOcean. Questo è indice del fatto che ci accingiamo a vivere in un’era dove saper programmare sarà come è adesso saper leggere e scrivere. Il numero di processori sulla terra sta superando il numero di esseri umani, questo vuol dire che qualsiasi oggetto avrà almeno un processore, sarà quindi possibile metterlo in rete e farlo interagire con altri sistemi: la domotica sarà dunque una componente essenziale nella vita.

Questo non vuol dire che Arduino sia un valido strumento tecnologico, esso infatti è poco più che un giocattolo più adatto a situazioni sperimentali che a veri e propri utilizzi industriali. Senza sminuire lo studio di tale settore, correndo anche il rischio di essere lapalissiani, possiamo concludere che l’elettronica ha praticamente raggiunto ogni settore della nostra vita e la relativa facilità con cui è possibile accedere a tali sistemi rende necessaria una radicale normazione, per evitare quello che già sta succedendo con i sistemi domotici Apple, ovvero, esistono server dove viene registrato il numero di volte che si accende una lampadina durante il giorno.

Si vuole sottolineare la sostanziale differenza tra un “impianto” domotico che un qualunque appassionato di elettronica potrebbe ideare e un impianto “normato ed efficiente” che solo un professionista può realizzare.

In conclusione, è bene precisare che questo elaborato è caratterizzato da voli Pindarici, ogni singolo aspetto meriterebbe infiniti ulteriori approfondimenti: dalle normative ai protocolli veri e propri, passando per aspetti “filosofici” ed economici.

Vantaggi e svantaggi

I vantaggi sono notevoli, si può monitorare e pilotare l'impianto anche a distanza con una semplice applicazione per smartphone o tablet, si possono impostare delle priorità nel distacco del carico in caso di superamento di una soglia di potenza (distacco sequenziale) nonché implementare una "selettività" degli interruttori dal momento che gli attuatori possono essere muniti di trasduttori di corrente ecc.

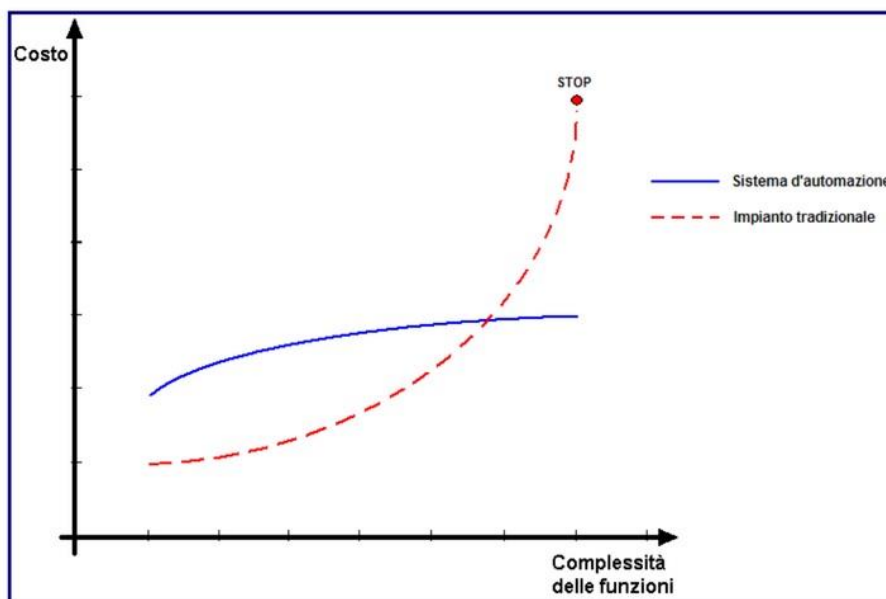
Un dispositivo può svolgere contemporaneamente più mansioni: un sensore ad effetto Hall utilizzato per controllare lo stato di apertura e chiusura di una tapparella, può far parte di un sistema di allarme e contemporaneamente può contribuire al risparmio: In quest'ultimo caso ad esempio, il sistema può spegnere il climatizzatore se l'utente ha dimenticato la finestra aperta, o meglio, chiudere la finestra se lo scenario lo consente, altrimenti può comunicare il "warning" all'utente.

Tra gli svantaggi principali vi è senza dubbio il costo, causa della mancata diffusione di questa disciplina: un banale relè a scatti costa infinitamente meno di un moderno attuatore wifi e, per chi non è addetto ai lavori, svolge la stessa funzione "causa-effetto": se premi il pulsante la lampadina si accende.

L'automazione di un impianto elettrico può sembrare "inutile" in quanto di fatto, un'abitazione civile presenta fondamentalmente tre grossi carichi elettrici: lavatrice, lavastoviglie e frigorifero. Se si sta consumando troppo non si può decidere di spegnere il frigorifero! Si può affermare dunque che in questo caso è l'efficienza stessa dell'elettrodomestico a fare la differenza.

In grossi palazzi invece l'integrazione di un buon controllo luci con lampade ad elevatissima efficienza può portare vantaggi notevoli.

Alcuni slides di presentazione della ABB riportano questo interessante grafico che riassume alcuni concetti:



Se l'impianto non è particolarmente complesso, fino a che i costi non scendono, conviene quasi installare un impianto tradizionale. Man mano che la complessità aumenta, i costi degli impianti tradizionali crescono in maniera considerevole, mentre i costi dell'impianto domotico restano quasi invariati dopo un certo grado di complessità. Per fare un esempio concreto, l' FIC61NP-230V della Eltako (in seguito verrà analizzato) può essere associato ad un crepuscolare con sensore di movimento, ma anche solamente ad un interruttore, il costo del dispositivo non varia nei due casi (ovviamente un crepuscolare tradizionale costa meno di un crepuscolare domotico).

Inoltre, come si vedrà più avanti, gli impianti di illuminazione "automatizzati" presentano rendimenti migliori.

Suddivisione

La domotica si divide in due grandi categorie in funzione delle applicazioni, “Home Automation” per l’ambito civile e “Building Automation” in ambito industriale, mentre si possono distinguere tre classi di applicazione:

Classe 1

- Comandi : illuminazione, riscaldamento, ventilazione, condizionamento, attuatori;
- Allarmi: soccorso, antintrusione, fughe di gas, incendio, allagamento ,tecnic;
- Diffusione sonore: controllo;

Classe 2

- Diffusione sonora : altoparlanti;
- Comunicazione: Telefono, citofono;

Classe 3

- Comunicazione: Video a larga banda;

Le modalità di controllo per un sistema domotico possono essere raggruppati in tre categorie automatico, manuale e misto. Il controllo Manuale (che richiede componenti passivi come interruttori o terminali di ingresso) consente all’utente di interagire con il sistema a differenza di quello automatico (che richiede dispositivi attivi quali sensori ecc.) dove appunto è il sistema a prendere decisioni. Una terza modalità di controllo è quella mista, che consente di intervenire manualmente anche se è presente un controllo automatico nel caso fosse necessaria una condizione “straordinaria” ovvero non prevista negli scenari prestabiliti (ad esempio in un corridoio dove la luce viene controllate per mezzo di un sensore di movimento e temporizzata, se è necessario stazionarvi è bene che la luce resti accesa tutto il tempo necessario e ciò può richiedere un controllo manuale che “forzi“ lo scenario).

Breve descrizione del progetto

Siamo giunti in un era in cui qualsiasi cosa può essere collegata ad un PC e di conseguenza in rete. L’intenzione di questo documento è appunto quella di evidenziare come con pochi componenti si possa creare un “centro di controllo per abitazione”, nella fattispecie si utilizza Matlab come linguaggio di programmazione, attuatori e sensori della Eltako che comunicano utilizzando il protocollo EnOcean. è previsto un Arduino Due a fare da interfaccia tra impianto e PC (tramite porta seriale).

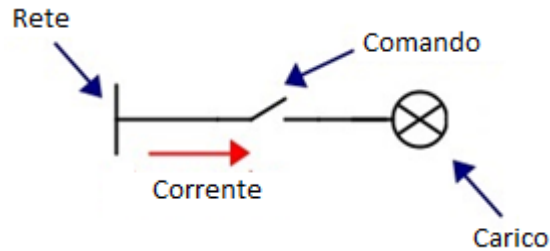
La scelta dell’ambiente di programmazione è del tutto casuale ed è dovuta al fatto che al momento in cui è

partito il progetto si stava utilizzando matlab per altri scopi (Questo linguaggio utilizza praticamente la sintassi del C, è stato dunque molto facile capire le meccaniche necessarie). I componenti Eltako e i TCM320 invece sono una gentile concessione del dipartimento.

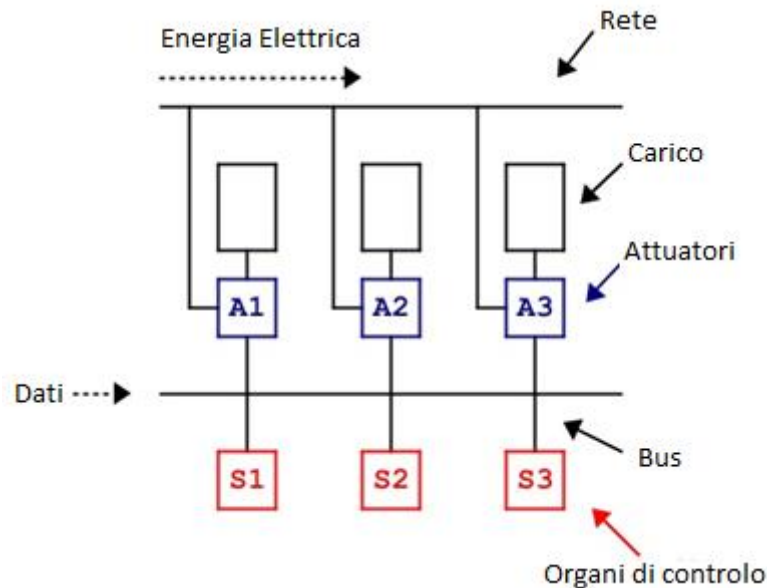


Principali differenze tra impianti domotici e tradizionali

Negli impianti tradizionali flussi elettrici e comandi non sono galvanicamente disaccoppiati, come si evince dal seguente schema unifilare:



Negli impianti domotici invece, comando e flusso energetico non si trovano sulla stessa linea:



In linea di principio, ogni dispositivo sul bus è dotato di un indirizzo. Gli organi di controllo (sensori, pulsanti ecc) mandano sul bus delle stringhe contenenti indirizzo dell'attuatore da comandare e l'oggetto (funzione da fargli eseguire), mentre gli attuatori riconoscono il comando a loro rivolto e lo eseguono. Il supporto fisico del BUS verrà analizzato più avanti.

Essendo un cablaggio di tipo software, dove quindi si possono cambiare le funzioni degli attuatori e dei sensori (nel limite del possibile), si ha quindi un cablaggio "flessibile".

Se ad esempio si vuole aggiungere un eventuale pulsante per pilotare uno dei carichi, non è necessario procedere con dispendiose opere murarie, basta aggiungere un ulteriore dispositivo al BUS.

Esistono in commercio pulsantiere che trasmettono dati via wireless che non necessitano di alimentazione in quanto prelevano l'energia necessaria attraverso fenomeni di natura piezoelettrica (alcuni cristalli se deformati meccanicamente generano una differenza di potenziale), questo rappresenta un grosso vantaggio in termini pratici: poter evitare di "scavare tracce" per il passaggio dei cavi in un edificio storico può dare la differenza.

CEI 64-8 (V3) – particolarità della versione 3 di tale normativa

Il capitolo 37 ("Ambienti residenziali Prestazioni dell'impianto") della normativa è specifico per gli impianti residenziali, in questo modo viene normata quella che è la "prestazione" (e quindi il comfort) degli impianti residenziali: si può parlare quindi del "passo zero" verso quella che è la formalizzazione del concetto di domotica.

L'impianto elettrico è classificato su tre livelli in base al numero di circuiti terminali e dotazioni funzionali tenendo conto anche della metratura dell'edificio (calpestabile).

La norma è pensata in modo tale da evitare le classiche “ciabatte” sparse per la casa, per tale motivo fornisce un numero minimo di prese a seconda del tipo di ambiente.

Livello 1: Il minimo livello richiesto, si tratta infatti di semplici impianti rispettando quella che è “la regola dell’arte” ovvero sicuro ed efficiente.

Livello 2: Rispetto al livello uno prevede un maggior numero di prese corrente, videocitofono e controllo carichi.

Livello 3: Funzioni domotiche.

Per ambiente		livello 1				livello 2				livello 3 ⁽¹⁾			
		Punti prese ⁽¹⁾	Punti luce ⁽²⁾	Prese radio/TV	Prese telefono e/o dati	Punti prese ⁽¹⁾	Punti luce ⁽²⁾	Prese radio/TV	Prese telefono e/o dati	Punti prese ⁽¹⁾	Punti luce ⁽²⁾	Prese radio/TV	Prese telefono e/o dati
Per ogni locale, ad esclusione di quelli sotto elencati in Tabella, (ad es. camera da letto, soggiorno studio, ...)	8 m ² < A ≤ 12 m ²	4	1			5	2			5	3		
	12 m ² < A ≤ 20 m ²	5	1	1	1	7	2	1	1	8	3	1	1
	20 m ² < A	6	2			8	4			10	4		

Ad esempio, per un locale di 10mq

-4 prese e 1 punto luce per classificarlo come livello 1

-5 prese e 2 punti luce per classificarlo come livello 2.

Affinchè la casa sia “domotica” è necessario che siano integrate almeno quattro funzioni che devono essere “integrate e realizzate con un controllo Domotico”.

Alcuni esempi di funzioni:

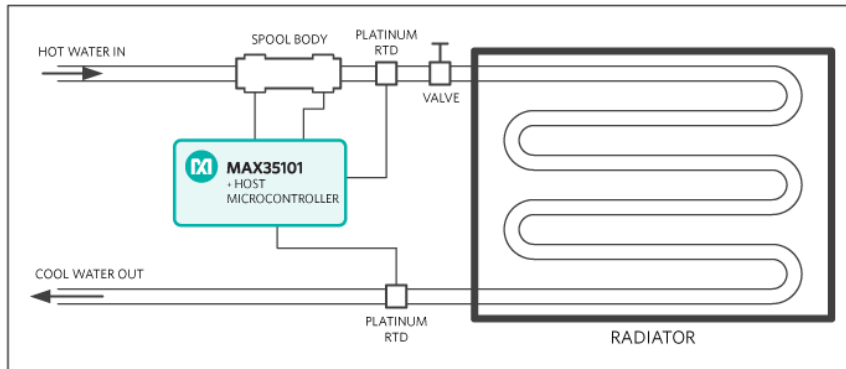
- Allarme;
- Controllo carichi;
- Gestione comando luci;
- Gestione temperatura;
- Gestione scenari;
- Controllo remoto;
- Impianto audio;
- Antincendio;
- Rilevatore di gas e allagamento;

Campioni e misure di energia termica

Le funzioni di automazione deputate al risparmio energetico sono in genere abbastanza semplici ed economiche rispetto alle più complesse e costose funzioni di “comfort”.

È bene ricordare che il settore domestico è piuttosto energivoro e assorbe una buona fetta dell’intero fabbisogno nazionale, introdurre strumenti come la domotica può impattare notevolmente nei bilanci, anche se a scorgiare tale investimento sono i notevoli costi di partenza .

Per il monitoraggio dei consumi, sono necessarie misure accurate (riconducibili all’SI) facenti riferimento ad un campione. Servono anche campioni di trasmittanza termica per valutare le perdite. Esistono in commercio dispositivi che forniscono la misura dell’energia ceduta all’ambiente, ad esempio il MAX35101 della MAXIM INTEGRATED du cui viene riportato lo schema applicativo (che si può trovare sul sito della casa madre) :



Il principio di funzionamento può essere riassunto come segue: L'energia fornita dal radiatore all'ambiente circostante è data dalla seguente formula:

$$E = c(\theta) * \dot{m} * \Delta\theta$$

Dove E è l'energia scambiata, $c(\theta)$ il calore specifico del fluido, \dot{m} è la portata massica e $\Delta\theta$ è la differenza di temperatura tra ingresso e uscita. La portata massica è una misura indiretta eseguita misurando la velocità del fluido: conoscendo il diametro del tubo si risale ad una portata in volume per poi arrivare alla portata massica tramite la densità. Sia la densità che il calore specifico cambiano con la temperatura dunque è necessario applicare delle "tabelle di conversione" che legano volume e massa in funzione della temperatura.

Per la misura delle temperature vengono utilizzate due tipi di trasduttore a seconda del range di temperatura: termocoppia per le alte temperature e RTD (termoresistenza: sfrutta la variazione di resistività del materiale con la temperatura) per le basse temperature.

La velocità del flusso viene misurate utilizzando un "ultrasonic flow meter".

UNI EN 15232 –“Prestazione energetica degli edifici. Incidenza dell'automazione, della regolazione e della gestione tecnica degli edifici”

Un limite pratico alla diffusione della Domotica sta nella possibilità di quantizzare in termini economici i vantaggi legati all'efficacia dell'investimento ed eventualmente confrontare più investimenti "ex ante" (a differenza dell'adozione di componenti di miglior efficienza come lampadine a basso consumo ecc. il cui "ROE" è ben definito.

La normativa definisce quattro classi convenzionali (simili all'etichettatura energetica degli elettrodomestici) chiamate "classi di efficienza BAC" acronimo di Building Automation and Control":

- Classe D : Nessun livello di automazione (impianti elettrici tradizionali) e dunque a bassissima efficienza energetica (i nuovi impianti non dovrebbero essere progettati in classe C mentre quelli esistenti andrebbero ammodernati);
- Classe C : Impianti dotati di sistemi di controllo senza supervisione centrale, rappresenta la classe di riferimento (BAC standard);
- Classe B: Impianti dotati di un buon livello di automazione e supervisione centralizzata;
- Classe A: Come la classe B ma con alti livelli di precisioni e completezza che garantiscono prestazioni elevate, l'impianto risulta ad altissima efficienza energetica.

Viene messa a disposizione una tabella che riporta le funzioni di automazione richieste per rientrare nelle categorie sopracitate, anche se a volte possono essere delle "forzature" : Aspirare alla classe A, un edificio deve essere provvisto di controllo automatico della luce naturale, ma se tale edificio si trova sotto terra tale funzione non può essere implementata per ovvi motivi.

Per classificare un intero edificio si fa riferimento all'impianto con a classe più bassa.

La normativa prevede due procedure per il calcolo dell'impatto sulle prestazioni dell'edificio a seconda se si conosce in dettaglio il sistema o si vuole effettuare una stima "ex ante".

- **Calcolo dettagliato**

Si basa su simulazioni analitiche (il metodo del calcolo è descritto nella EN ISO 13790) dunque rispecchiano la realtà fino a che il sistema non porti ad una variazione dello stato dell'ambiente in tempi minori rispetto agli step di simulazione. È necessario che il sistema sia completamente noto, se si conosce la potenza P necessaria e il rapporto tra tempo di utilizzo e tempo totale (Fc) si può scrivere che l'assorbimento E vale:

$$E = P * t * Fc$$

È bene precisare che se il dispositivo non è di tipo on/off ma la potenza modulare, Fc è pari al tempo operativo "equivalente".

In alternativa, se si conosce il consumo nel caso di riferimento E_{CR} , si può dire che l'assorbimento E vale:

$$E = E_{CR} * X_c$$

Dove X_c è un coefficiente correttivo che dipende dal sistema e dalle condizioni ambientali.

- **Fattori BAC**

Il metodo dei fattori BAC si basa su coefficienti di tipo statistico, ovvero sono stati presi in considerazione dei sistemi reali e in seguito a misure e adeguati calcoli si è giunto a tali fattori (il gran numero di test eseguiti fa sì che siano dati attendibili).

Alcuni schemi elementari di impianti domotici

Termoregolazione

Affinché un terminale termico si presti all'implementazione di una funzione di automazione è necessario che la sua inerzia termica sia bassa per consentire veloci regolazioni di temperatura e con una piccola isteresi, per cui i ventil-convettori si prestano particolarmente a questo scopo.

Un aspetto critico sta nel riscaldamento e raffrescamento degli ambienti, accorgimenti particolari possono impattare positivamente sui consumi:

- Divisione dell'ambiente in microzone indipendenti con cronotermostati (di cui un master e più slave) ed elettrovalvole, comando automatico i funzione della presenza o meno di persone e attenuazione dell'apertura di porte e finestre con ripristino automatico di chiusura ;
- Funzione di protezione dei dispositivi elettromeccanici (antisticking) o complete aperture e chiusure periodiche per evitare l'incollaggio delle valvole dovuto a lunghi periodi di chiusura/apertura;
- "indoor air quality" (IAQ) ovvero la presenza di un sensore che analizza la qualità dell'aria che consente di far attivare il sistema di ricambio dell'aria se necessario o in orari non troppo freddi in inverno e non troppo caldi in estate per evitare sprechi;
- Integrazione di stazione meteo per consentire l'azionamento di tende da sole o tapparelle;
- Funzione antigelo e protezione arredi e impianti per ambienti ad occupazione saltuaria;

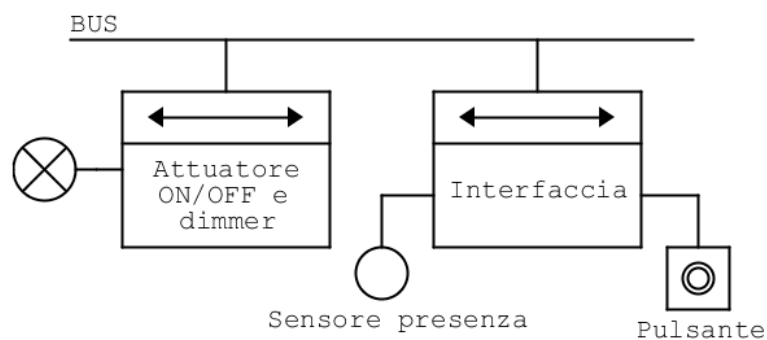
Illuminazione

Le principali funzioni di automazione che portano alla riduzione dei consumi agendo sull'illuminazione possono essere:

- Accensione/spengimento di apparecchi di illuminazione temporizzati e sensori di presenza (sensori di movimento nelle aree di passaggio quali corridoi, ripostigli ecc).
- Regolazione dell'intensità luminosa sfruttando al massimo quella naturale con integrazione di quella artificiale solo se strettamente necessario;
- Il miscelamento tra luce naturale e quella artificiale che può avvenire in due modi , agendo direttamente sul sistema di illuminazione o su sistemi di schermatura (tapparelle, tende ecc.).
- Alternanza programmata tra diversi gruppi di apparecchi con lo scopo di aumentarne la vita utile.
- Accensione con sensore crepuscolare di lampade segnaposto e led di segnalazione integrati nei pulsanti e nelle centraline (se possibile utilizzare un singolo dispositivo di segnalazione per un gruppo di più dispositivi);

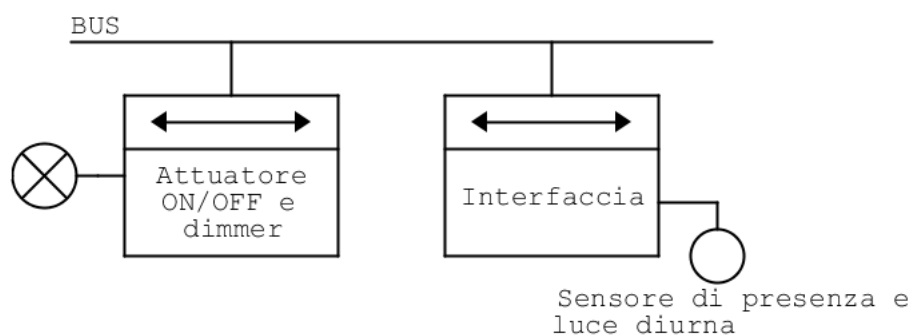
Di seguito sono riportati alcuni schemi semplici per l'automazione dell'illuminazione residenziale.

Controllo presenza

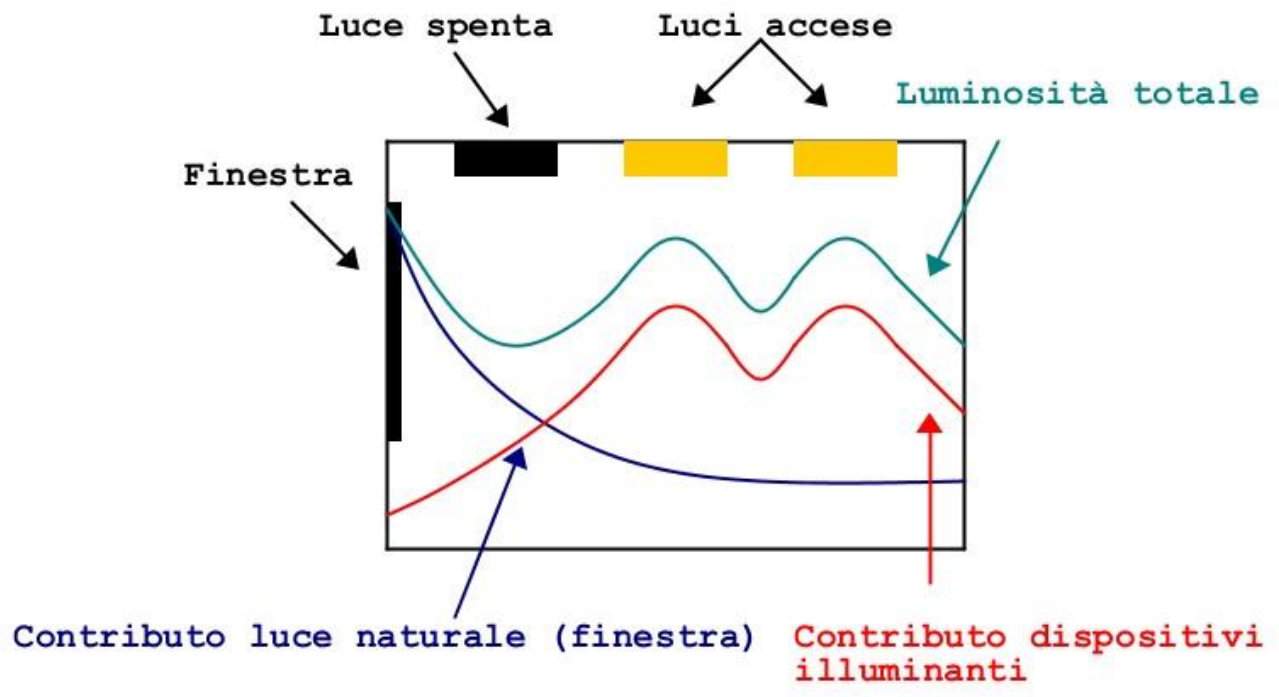


Questo sistema rientra nella Classe A poiché presente interruttore manuale, segnale di estinzione graduale (implementato software) rilevamento di presenza, auto-on, auto-off e dimmer (dalle tabelle si risale alla classe). L'interruttore manuale o il sensore di presenza accendono la luce, se dopo un certo intervallo di tempo non viene rilevata la presenza di persone il sistema abbassa l'illuminazione ad un valore non superiore al 20% per poi provvedere a spegnerlo dopo un ulteriore intervallo di tempo.

Controllo luce diurna



La luce viene regolata in funzione della presenza di persone e in base alla luce proveniente dall'esterno.



Dalla figura si evince che in prossimità delle pareti finestate il contributo della luce naturale è sufficiente da garantire da sola la luce ambientale, allontanandosi invece ciò non è più verificato. È necessario dunque accendere la seconda e terza fila di luci a destra per avere un ambiente ottimale.

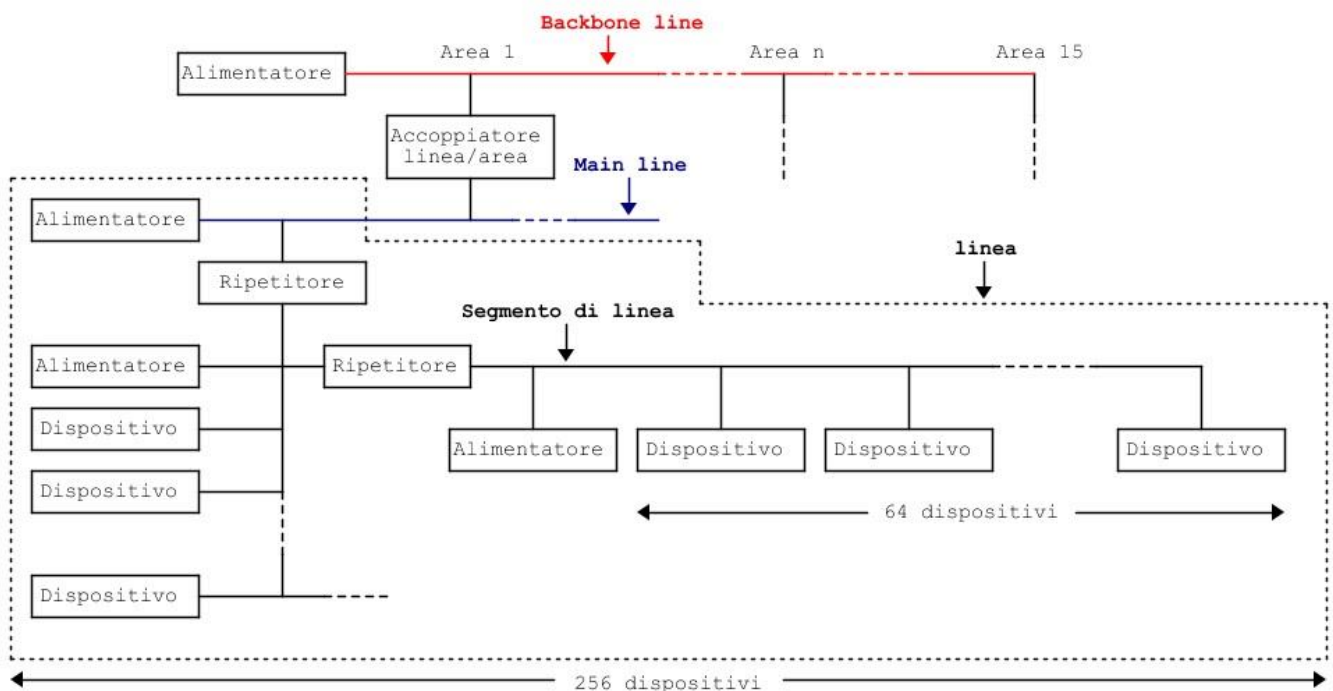
Lo standard KNX (Architettura)

Lo standard knx è un protocollo per la gestioni di impianti decentralizzati.

L'Associazione KNX è la realtà che, nel 1999, ha dato vita allo standard KNX come protocollo di comunicazione dei dispositivi per il funzionamento e il controllo di impianti domotici di Home e Building Automation, approvato come standard mondiale (ISO /IEC 14543-3).

Si tratta, quindi, di uno standard aperto che garantisce l'interoperabilità tra dispositivi di diversi costruttori, evitando in tal modo che l'installatore e soprattutto il proprietario dell'impianto restino vincolati a un unico costruttore per sempre.

Un impianto domotico presenta una topologia ad albero o a stella ma non ammette il "loop" per motivi elettrici. Esso di compone di un bus , accoppiatori o ripetitori, alimentatori moduli di ingresso/uscita:



Partendo dall'alto, il protocollo consente di dividere l'impianto in un massimo di 15 aree collegate tra loro mediante la backbone (dorsale) da cui dipartono le linee dove sono collegati i dispositivi, fino ad un massimo di 15 linee per area.

Su ogni linea possono essere collegati un massimo di 256 dispositivi divisi in gruppi da 64 distribuiti sui vari segmenti di linea, ogni nodo ha un indirizzo a 16bit dunque in linea teorica possono essere montati in un impianto 64.000 dispositivi.

Ogni dispositivo (nodo) ha un proprio indirizzo ed è univoco in tutto l'impianto:

area[0-15].linea[0-15].dispositivo[0-255]

Ad esempio 2.6.26 è il ventiseiesimo dispositivo della sesta linea relativa alla seconda area.

Alimentatore

Ogni linea (e relativi segmenti) deve essere collegata ad un alimentatore. Gli alimentatori forniscono una tensione di 29V d.c. direttamente ai morsetti del bus, ciascuno può alimentare fino a 64 dispositivi tenendo conto degli assorbimenti energetici. La distanza massima di cavo bus tra un dispositivo e il relativo alimentatore deve essere minore di 350m, superato tale limite bisognerà introdurre dei ripetitori/accoppiatori di linea e creare nuovi segmenti di linea o linee, mentre la distanza minima tra due alimentatori è di 250m.

Accoppiatore/ripetitore

L'accoppiatore separa galvanicamente le linee bus impedendo la propagazione di un guasto elettrico e rigenera il segnale.

- Accoppiatore area/area : collega tra loro le aree lungo la dorsale (backbone);
- Accoppiatore linea/area: collega tra loro le linee in un'area lungo la linea principale (main line);
- Ripetitore: collega tra loro due segmenti di linea (rigenerando il segnale);
- Filtro: Si può impostare il dispositivo accoppiatore in modo da "bloccare" il passaggio ad alcune informazioni per evitare che vengano inviate inutilmente (per migliorare la capacità di comunicazione);

Mezzi trasmissivi

Nel sistema Bus lo scambio dei dati tra attuatori e comandi avviene tramite una linea dedicata separata dalla linea di potenza che viene collegata unicamente ai carichi. Questa caratteristica si riflette sulla flessibilità del sistema in quanto è possibile portare delle modifiche senza intervenire sul cablaggio ma semplicemente agendo sul firmware (esistono appositi moduli per l'interfacciamento con il PC mediante una porta seriale USB).

Esistono diversi mezzi trasmissivi:

- **Cavi in rame "twistati" (twisted pair, TP):** si tratta di cavi bipolari ritorti schermati e con doppio isolamento (isolati fino a 4000V) che garantiscono una velocità di trasmissione di 9.600 bit/s. I cavi sono di colore nero e rosso chiamati rispettivamente "BUS-" e "BUS+", mentre il cavo a quattro fili consente di alimentare applicazioni aggiuntive SELV (separate extra low voltage) mediante due ulteriori cavi (giallo e bianco). Il cavo può essere installato a diretto contatto con i cavi di potenza, sempre che venga mantenuto l'isolamento ;
- **Power line (PL) :** Le onde ad alta frequenza che codificano le informazioni vengono sovrapposte alla tensione di rete (la comunicazione utilizza le preesistenti linee di forza motrice);
- **Onde radio (Radio frequenza):** Le informazioni viaggiano per mezzo di onde radio con portante a 868.30 MHz, (gli stati logici 1 e 0 vengono generati dalla deviazione rispetto alla frequenza portante) . La velocità di trasmissione è 16.32bit/s.
- **Ethernet:** I dati vengono trasferiti con protocollo IP (internet protocol). Nei grossi impianti la rete internet viene utilizzata come dorsale ad alta velocità (fast-backbone).

Arduino Due

Si tratta di schede elettroniche a microcontrollore che possono essere programmate ad alto livello senza l'ausilio di programmatori, bensì possono essere collegate al PC tramite USB e gestite da appositi IDE. Arduino offre uno ambiente di programmazione basato sul C che quindi offre sintassi semplici e funzionali che consentono di snellire la programmazione. Per questo progetto si è scelto "Arduino Due" in quanto possiede porte logiche TTL 3.3V, dunque non necessita interfacciamenti ulteriori con il TCM320 (descritto in seguito) . Inoltre possiede ben 4 moduli seriali integrati , si presta molto bene alla realizzazione di questo progetto. I dati trasmessi dal TCM320 vengono letti dal Arduino che provvede a trasmetterli al PC (e viceversa).



Firmware Arduino

Per quanto riguarda Arduino le funzioni che consentono di inviare e ricevere dati tramite seriale sono:

- **Serial.begin(BAUD_RATE)**
Che inizializza la seriale (Serial2.begin(), Serial3.begin()) permettendo di impostare il baud rate, i bit di start e stop, di parità ed il numero di bit per dato.
- **Serial.write('dato_da_inviare')**
Invia il dato.
- **Serial3.available()**
I dati vengono stoccati grazie a dei buffer seriali. Questa funzione restituisce il numero di byte non letti presenti nel buffer che vengono letti (in maniera sequenziale) con la seguente funzione:
- `byte_ricevuto = Serial3.read()`
Legge il dato nel buffer seriale.

La prima inizializza le porte seriali e vuole come argomento il baud rate, di default sono impostati 8bit di dato, nessun bit di parità e un bit di stop (per cambiarli è sufficiente inserire i parametri dopo il baud rate). (Si ricorda che

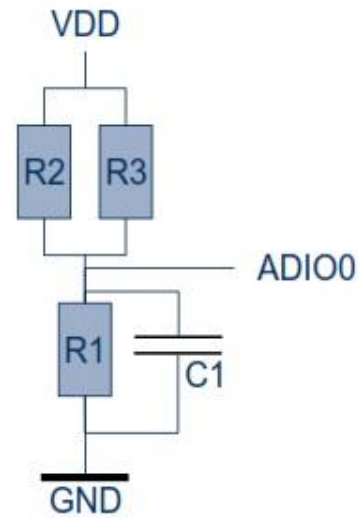
Esempio di programma

```
int byte_ricevuto;

void setup() {
  Serial.begin(9600); //Settaggio seriale principale; baud rate: 9600
  Serial3.begin(9600); //seriale 3 (RX sul pin 15);
}

void loop() {
```


Mode	ADIO0 (MODE_SEL) input voltage range	Proposed component values
0	0% to 3.99% VDD	R1: 0 Ohm R2: leave open R3: leave open C1: leave open
1	4% to 11.99% VDD	R1: 1k2 ±1% R2: 15k ±1% R3: 150k ±1% C1: 100p
2	12% to 19.99% VDD	R1: 2k2 ±1% R2: 12k ±1% R3: 270k ±1% C1: 100p
3	20% to 27.99% VDD	R1: 3k9 ±1% R2: 15k ±1% R3: 68k ±1% C1: 100p
4	28% to 35.99% VDD	R1: 4k7 ±1% R2: 12k ±1% R3: 56k ±1% C1: 100p
5	36% to 39.99% VDD	R1: 5k6 ±1% R2: 10k ±1% R3: 56k ±1% C1: 100p



In

particolare si sono utilizzate le modalità 0 e 1:

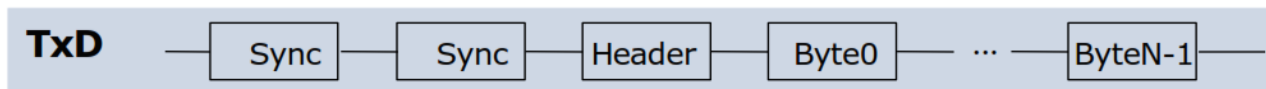
Mode	Function	Output signal description	No. of channels
0	Unidirectional serial interface compatible with TCM 220C, no teach-in capability	SER_TX: UART output, supplies standard data blocks of information from all received EnOcean radio telegrams (9600 bps; 8 data bits, no parity bit, one start bit, one stop bit). For further information see chapter A.1	
1	Bidirectional serial interface, teach-in capability for up to 30 entries ¹	SER_RX, SER_TX: Asynchronous bidirectional Interface, supplies standard data blocks of information from all received EnOcean radio telegrams (9600 bps; 8 data bits, no parity bit, one start bit, one stop bit). For further information see chapter A.1 LRN_TOGGLE: Learning mode status indicator	

Lo standard EnOcean

L'EnOcean è uno standard di comunicazione wireless nato per ridurre i consumi energetici, basa il suo funzionamento sul "recupero energetico" infatti molti dispositivi sono muniti di piccoli pannelli fotovoltaici oppure sfruttano l'energia cinetica dell'operatore grazie a fenomeni di natura piezoelettrica (pulsanti).

Non necessita di particolari manutenzioni, quindi i dispositivi possono essere posizionati in punti critici quali ad esempio scatole di derivazione. Le frequenze usate come portanti sono 868MHz, 315 MHz, 902 MHz (introdotta appositamente per l'America del nord) e 928 MHz a seconda delle applicazioni.

La struttura di un telegramma ESP (EnOcean serial protocol):



Message format for asynchronous serial communication

Bit 7	Bit 0
SYNC_BYTE1 (A5 Hex)	
SYNC_BYTE0 (5A Hex)	
H_SEQ	LENGTH
ORG	
DATA_BYTE3	
DATA_BYTE2	
DATA_BYTE1	
DATA_BYTE0	
ID_BYTE3	
ID_BYTE2	
ID_BYTE1	
ID_BYTE0	
STATUS	
CHECKSUM	

SYNC_BYTE1 (Byte di sincronizzazione) 0xA5

SYNC_BYTE0 (Byte di sincronizzazione) 0x5A

H_SEQ (3bit) = tipo di telegramma (tabulato)

LENGTH (5 bit) = numero di ottetti successivi (11)

(H_SEQ + LENGTH = HEADER)

ORG : Tipo di telegramma

DATA_BYTE : dati del telegramma

ID_BYTE : Byte di identificazione

STATUS: stato

CHECKSUM: Si tratta di un byte di controllo. In particolare è il byte meno significativo della somma

I byte di sincronizzazione servono appunto per sincronizzare il dispositivo, per L'ESP (qui si analizza l'ESP2) questi byte devono essere 0xA5 e 0x5A.

Esistono 4 tipi di telegramma:

- RRT - Receive Radio telegrams (from AIR to Serial)
- TRT - Transmit Radio telegrams (from serial to air)
- RCT - Receive Command Telegram (between module and host only serial)
- TCT - Transmit Command Telegram (between module and host only serial)

L'header (identificativo) è composto da due parti: la prima, di tre bit, è H_SEQ che identifica il tipo di telegramma (vedere seguente tabella). La seconda, di 5 bit, contiene la lunghezza del telegramma, ovvero in numero di ottetti successivi all'HEADER (in questo caso 11).

H_SEQ	Meaning	Mode
0b000	<ul style="list-style-type: none"> Unknown transmitter ID received (serial telegram only if no ID has been learned so far!) For RPS also: <ul style="list-style-type: none"> Known transmitter ID and unknown rocker U-message from known transmitter ID received For HRC also: <ul style="list-style-type: none"> Known transmitter ID and unknown rocker Scene switch command (last three bits of ID 0b111) from known transmitter ID (only first 29 bits are compared!) 	Operating Mode
0b001	<ul style="list-style-type: none"> For 1BS and 4BS: Known transmitter ID received For RPS: Known transmitter ID and at least 1 known rocker (1 or 2 rockers operated) For HRC: Known transmitter ID and known rocker 	Operating Mode
0b010	New transmitter learned (If a switch telegram is received (RPS or HRC), the rocker code (RID) is stored together with the ID.)	Learn Mode
0b110	Transmitter just deleted (If a switch telegram is received (RPS or HRC), the rocker code (RID) and module ID are checked. The entry is only deleted if module ID and rocker are known.)	Learn Mode
0b011	Transmit radio Telegram (TRT) Host -> Module -> Air	Operating Mode
0b100	Receive Command Telegram (RCT) Module -> Host	Operating Mode
0b101	Transmit Command Telegram (TCT) Host -> Module	Operating Mode

ORG è il tipo di telegramma, può assumere i valori 0x05, 0x06 e 0x07, rispettivamente se il telegramma è RPS, 1BS o 4BS.

Il checksum è il byte di controllo, ovvero il byte meno significativo della somma tra tutti gli ottetti esclusi quelli di sincronizzazione e il checksum stesso.

Il manuale suggerisce di verificare che non vi siano errori durante la trasmissione dei dati poiché potrebbero danneggiarsi, specialmente nella trasmissione full duplex. Di seguito vengono riportati alcuni dei telegrammi di

risposta:

ERR_SYNTAX

Bit 7	Bit 0
0xA5	
0x5A	
0x8B	
Field	
X	
X	
X	
X	
X	
X	
X	
X	
ChkSum	

This telegram is sent automatically through the serial port after the TCM has detected a syntax error in a TCT telegram. Errors can occur in the H_SEQ, LENGTH, ORG or CHKSUM fields/bytes.

Field code:
H_SEQ=0x08 ORG=0x0B
LENGTH=0x09 CHKSUM=0x0A

OK

Bit 7

Bit 0

0xA5
0x5A
0x8B
0x5B
X
X
X
X
X
X
X
X
X
X
X
ChkSum

Standard message used to confirm that an action was performed correctly by the TCM.
In case of full duplex communication it may happen that serial telegrams get corrupted and lost. Therefore it is recommended to check for "OK" where applicable.

ERR

Bit 7

Bit 0

0xA5
0x5A
0x8B
0x19
X
X
X
X
X
X
X
X
X
X
ChkSum

Standard error message response if after a TCT command the operation could not be carried out successfully by the TCM.

Dispositivi Eltako

La Eltako è l'azienda tedesca che si occupa di apparecchiature elettriche per impianti, sia tradizionali che domotici. Di seguito vengono presentati brevemente gli attuatori e i sensori in esame.

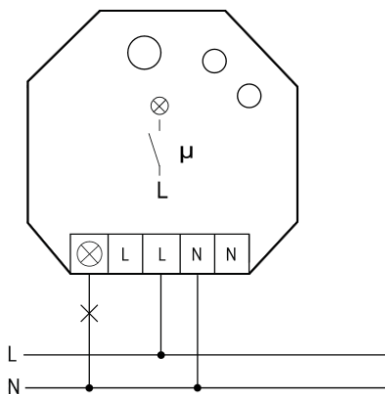
- **FMH4-ws : Mini telecomando wireless**

Costituito da quattro pulsanti esso sfrutta l'energia cinetica fornita dall'operatore per inviare i telegrammi (fenomeni piezoelettrici)



- **FSR61VA-10A : Attuatore wireless**

Relè passo-passo e monostabile con misurazione di corrente.



- **FIC61NP-230V : Controllore luce**

Modalità di funzionamento:

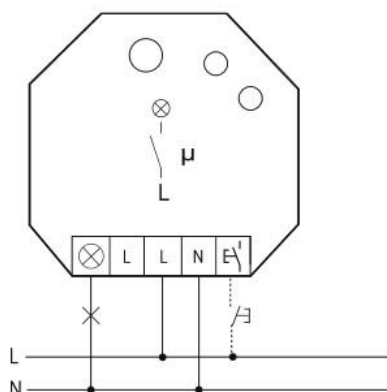
ES(V)+TLZ = Relè passo-passo e temporizzato 0-60 minuti.

Auto 1 : Semiautomatico con movimento, si spegne senza movimento e dopo il tempo di ritardo impostato.

Auto 2 : Semiautomatico con movimento e luminosità. Si spegne senza movimento raggiunta la luminosità impostata e dopo il tempo di ritardo impostato.

Auto 3 : Automatico, si accende dopo che la luminosità dell'ambiente scende sotto la soglia impostata e dopo il tempo di ritardo impostato.

Auto 4 : Automatico, si accende dopo che la luminosità dell'ambiente scende sotto la soglia impostata e si spegne senza movimento o raggiunta la luminosità impostata e dopo un certo tempo.



- **FBH65S-wg : Rilevatore di movimento e luminosità**

Sensore di luminosità e di movimento wireless.



Tale sensore misura l'intensità luminosa da 0 a 2000 lux e trasmette il proprio telegramma ogni 100 secondi se la luminosità è cambiata di almeno 10 lux dall'ultimo invio. Se rileva movimento invece, il telegramma viene inviato immediatamente. Se lo stato non cambia (nessuna variazione di intensità luminosa e nessun movimento rilevato) lo stato viene inviato ogni 20 minuti.

Configurazioni dei dispositivi

Scaricando appositi software, caricando le librerie dei dispositivi in uso reperibili presso il sito della casa produttrice, si possono configurare i dispositivi in S-MODE (sistema), ovvero si ha la totale gestione e la massima flessibilità dell'impianto. La S-MODE è utilizzata nella Building Automation a causa del grande numero di dispositivi e della complessità dell'impianto ed offre potenti sistemi di diagnostica.

Se si necessita di funzionalità base si può configurare l'impianto in E-MODE (semplice) in cui dispositivi contengono al loro interno le configurazioni possibili.

La Configurazione più semplice è quella A-MODE ovvero il "plug&play" destinato a piccole applicazioni come ad esempio far accendere la lampadina con lo smartphone.

I dispositivi in esame sono configurabili in E-MODE, infatti a seconda della posizione dei potenziometri al momento del "teaching-in" (ovvero della sincronizzazione) essi svolgono le funzioni prestabilite.

Di seguito viene riportata la procedura di teaching-in di un FSR61VA-10A :

Pulizia memoria : Posizionare il potenziometro superiore su CLR e prima che passino dieci secondi ruotare il potenziometro inferiore verso destra (fino a fine corsa) e tornare indietro per tre volte. Quando il led smette di lampeggiare la memoria è resettata.

Reset singolo comando : Posizionate il potenziometro superiore su CLR e inviare il comando da cancellare (ad esempio se si era sincronizzato con un pulsante, per farli disaccoppiare è sufficiente fare questa operazione).

teaching-in: posizionare il potenziometro superiore su "LRN" e quello inferiore in "right stop", poi si invia il segnale dal sensore (es. pulsantiera), in questo modo il pulsante viene associato dall'attuatore come un pulsante di OFF.

-----ULTERIORI OPERAZIONI-----

```
fclose(s3);  
delete(s3);  
clear s3;
```

Una volta nominata la seriale in questione come 's3' (ovviamente qualsiasi altro nome va bene), si procede con il settaggio dei parametri quali baud rate, numero di bit per dato ecc, poi si "apre" la porta con `fopen(s3)`: Questa operazione risulta essere particolarmente delicata in quanto una volta che matlab ha l'accesso alla porta, se il programma viene interrotto prima che questa viene chiusa opportunatamente con `fclose(s3)`, lanciando successivamente il programma questo non riuscirà ad accedervi. Ciò è dovuto al fatto che essendo una trasmissione asincrona (ovvero la sincronizzazione è affidata ai dati trasmessi), se il dispositivo esterno continua a trasmettere e il software non riceve si perde la sincronizzazione. La lettura del buffer invece avviene con il comando:

```
dato = fgets(s3);
```

Il programma gira fino ad arrivare a questa istruzione e vi rimane fino a che il dato seriale non viene letto. In realtà non rimane in attesa all'infinito: va settato il tempo di attesa come parametro 'Timeout':

```
set(s3, 'Timeout', x);
```

Questo comando fa sì che il programma resti in attesa del dato per x secondi. Se una volta trascorsi x secondi non viene ricevuto alcun dato l'istruzione viene saltata.

```
clc %pulisce lo script  
n=1;
```

```
s3 = serial('COM3'); %determinazione seriale  
set(s3, 'BaudRate', 9600);  
set(s3, 'DataBits', 8);  
set(s3, 'StopBits', 1);  
set(s3, 'Parity', 'none');  
set(s3, 'Timeout', 1000);  
fopen(s3);
```

```
while n<15
```

```
    dato = fscanf(s3); %lettura buffer
```

```
    switch n %a seconda dell'indice il telegramma contiene un tipo di dato
```

```
        case 5  
            data_byte_3 = dato;  
        case 6  
            data_byte_2 = dato;  
        case 7  
            data_byte_1 = dato;  
        case 8  
            data_byte_0 = dato;  
        case 9  
            id_byte_3 = dato;  
        case 10  
            id_byte_2 = dato;  
        case 11  
            id_byte_1 = dato;
```

```

        case 12
            id_byte_0 = dato;
        end

        n = n+1;
    end
fclose(s3);
delete(s3);
clear s3;
format hex %forza lettura dati come esadecimali (L'ID è in esadecimale)

ID = strcat(id_byte_3,id_byte_2,id_byte_1,id_byte_0)%ricostruzione ID mediante
%concatenazione

```

Quest'ultima funzione provvede a concatenare i quattro caratteri.

Studio dei dispositivi

Lanciando il programma e aspettando che il sensore di luminosità mandi il proprio telegramma, il risultato è il seguente:

ID_CREPUSCOLARE = 18F8A94

Ciò prova che il sistema è ben oliato in quanto l'ID corrisponde a quello scritto sul retro del sensore: Analogamente si risale all'ID della pulsantiera:

ID_PULSANTIERA = FEFD4B61

(Questo è ovviamente univoco per tutti e quattro i pulsanti, anche durante il rilascio)

Aggiungendo il seguente comando risaliamo al dato trasmesso dalla pulsantiera:

```
DATO = strcat(data_byte_3,data_byte_2,data_byte_1,data_byte_0)
```

Pulsantiera FMH4-ws

	DATO
Pulsante 1	30000
Rilascio pulsante 1	0000
Pulsante 2	70000
Rilascio pulsante 2	60000
Pulsante 3	10000
Rilascio pulsante 3	0000
Pulsante 4	50000
Rilascio pulsante 4	0000

il FBH65S-wg (sensore di luminosità e di presenza) – utilizzo come sensore di luminosità

Di seguito un estratto dei manuali Eltako, in particolare la struttura del telegramma inviato dal FBH65S-wg.

```
ORG =          0x07
Data_byte3 =   -
Data_byte2 =   brightness 0 – 2048 lux, linear n = 0x00 – 0xFF
Data_byte1 =   -
Data_byte0 =   DBO_Bit3 = LRN Button
                (0 = teach-in telegram, 1 = data telegram)
                DBO_Bit1 = motion
                (0 = motion , 1 = no motion)
                for data telegram: 0x0D (motion), 0x0F
                (no motion)
                for teach-in telegram: 0x85
```

Nel DATA_BYTE2 viene riportata la luminosità in lux (Il lux è l'unità di misura dell'illuminamento accettata dall' SI e equivale ad un lumen su un metro quadrato) da 0 a 2048 lux con scala lineare. Ovviamente il dato costituito da 8 bit va da 0 a 255 unità ergo è necessaria una proporzione per riportare il dato in lux:

$$DATA_BYTE2: 255 = [LUX]: 2048$$

$$[LUX] = \frac{DATA_BYTE2 * 2048}{255}$$

Il sensore invia un telegramma ogni 100 secondi se la luminosità è scesa di minimo 10 lux, oppure quando rileva una presenza.

L'idea è quella di monitorare la luminosità dell'ambiente esterno per verificare se ad esempio una pianta avrebbe il quantitativo di luce giornaliero necessario. In pratica si è posizionato il sensore nel balcone memorizzando i telegrammi (in particolare il DATA_BYTE2) per qualche ora. Il codice sorgente è una variante di quello precedente:

codice Arduino:

```
void loop() {
  if (Serial3.available() > 0) { // dato seriale disponibile
    telegramma[i] = Serial3.read(); //lettura dato seriale
    i++;
  }
  if(i==14){ //fine del telegramma
    i=0; //reset contatore
    Serial.println(telegramma[5],DEC); //Manda solo il quinto Byte : Data_Byte2 in decimale : da 0 a 255 (8bit)
  }
}
```

codice Matlab:

```
clc
n=2;

s3 = serial('COM3'); %determinazione seriale
set(s3, 'BaudRate', 9600);
set(s3, 'DataBits', 8);
```

```

set(s3, 'StopBits', 1);
set(s3, 'Parity', 'none');
set(s3, 'Timeout', 1000000);

fopen(s3);

while 1
    dato = fscanf(s3)%lettura buffer
    lux = str2num(dato)
    data = clock;
    ore = data(4);
    minuti = data(5);
    secondi = data(6);

    riga=int2str(n);
    indice=strcat('A', riga);
    xlswrite('C:\Users\Matteo\Desktop\tesi\seriale_matlab\lux.xlsx', lux, 1, indice);

    riga=int2str(n);
    indice=strcat('B', riga);
    xlswrite('C:\Users\Matteo\Desktop\tesi\seriale_matlab\lux.xlsx', ore, 1, indice);

    riga=int2str(n);
    indice=strcat('C', riga);

    xlswrite('C:\Users\Matteo\Desktop\tesi\seriale_matlab\lux.xlsx', minuti, 1, indice)
;

    riga=int2str(n);
    indice=strcat('D', riga);

    xlswrite('C:\Users\Matteo\Desktop\tesi\seriale_matlab\lux.xlsx', secondi, 1, indice)
);

n = n+1;
end

fclose(s3);
delete(s3);
clear s3;

```

Il comando per la scrittura su foglio excel è il seguente:

```
xlswrite('Address', dato, 1, indice);
```

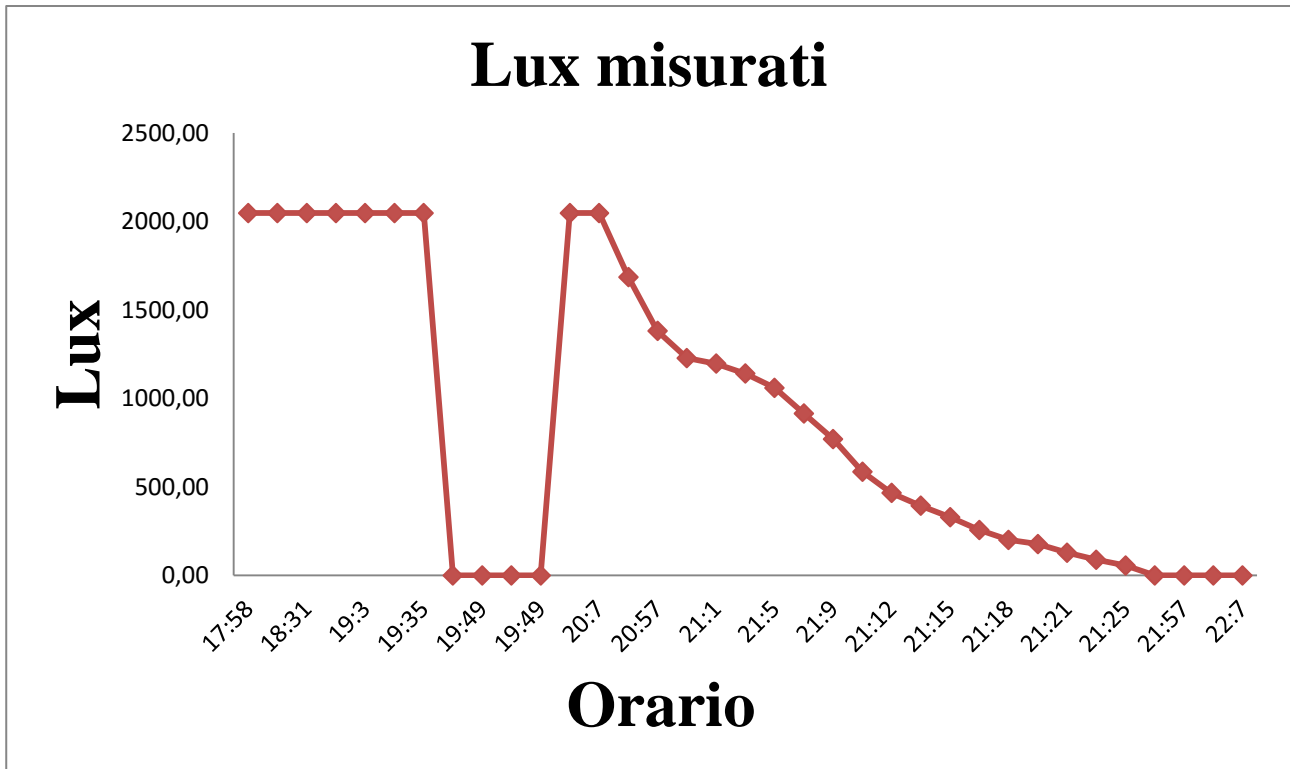
Scrive “dato” sul file di cui è indicato l’indirizzo nella casella “indice”. L’indice va creato di volta in volta in quanto è una stringa contenente il nome della cella:

```
indice=strcat('A', riga);
```

In questo caso l’indice corrisponde a ‘An’ dove n è il numero delle riga che viene incrementata ad ogni lettura (poi trasformata in carattere con `riga=int2str(n)`).

La distinzione tra stringa e numero non è superflua, una cattiva gestione dei tipi di dato porta a errori considerevoli.

Una volta acquisiti i dati sono stati elaborati in excel, da cui è stato estratto il seguente grafico:



Ci si sarebbe aspettato un andamento lineare, questo dati sono stati prelevati in una giornata nuvolosa.

Comunicazione tra TCM320

Firmware Trasmettitore

```
//Dichiarazioni variabili
```

```
int SYNC_BYTE1 = 0xA5;
```

```
int SYNC_BYTE2 = 0x5A;
```

```
int H_SEQ_LENGTH = 0x6B;
```

```
int ORG = 0x05;
```

```
int DATA[8]={0x0,0x0,0x0,0x0,0x0,0x0,0x0,0x0}; //Nel vettore DATA sono contenuti anche i byte di ID
```

```
int STATUS = 0x30;
```

```
int CHECKSUM;
```

```
int i =0;
```

```
void setup() {
```

```
  Serial.begin(9600); //Settaggio seriale principale; baud rate: 9600
```

```
  Serial3.begin(9600); //Settaggio seriale numero 3; baud rate: 9600
```

```
  //-----Calcolo del CHECKSUM-----//
```

```
  for(i=0;i<8;i++){
```

```
    CHECKSUM = CHECKSUM + DATA[i];
```

```
  }
```

```
  CHECKSUM = lowByte(H_SEQ_LENGTH + ORG + CHECKSUM + STATUS);
```

```
  //-----
```

```
  delay(10);
```

```
}
```

```

void loop() {
//----- trasmissione telegramma -----
Serial3.write(SYNC_BYTE1);
Serial3.write(SYNC_BYTE2);
Serial3.write(H_SEQ_LENGTH);
Serial3.write(ORG);
Serial3.write(DATA[0]);
Serial3.write(DATA[1]);
Serial3.write(DATA[2]);
Serial3.write(DATA[3]);
Serial3.write(DATA[4]);
Serial3.write(DATA[5]);
Serial3.write(DATA[6]);
Serial3.write(DATA[7]);
Serial3.write(STATUS);
Serial3.write(CHECKSUM);
Serial.println("-----"); // per non confondere una trasmissione dall'altra
delay(6); //il datasheet consiglia di attendere almeno 5 ms prima che il modulo fornisca una risposta
//----- lettura risposta del TCM320-----
if (Serial3.available() > 13 ) {
for(i = 0; i<14; i++){
Serial.println(Serial3.read(),HEX);
}
}
//-----
delay(1000); //per la verifica si manda un telegramma al secondo
}

```

Firmware Ricevitore

```

long telegramma[14];
long ID;
long DATA;
long i;

void setup(){
Serial.begin(9600);
Serial3.begin(9600);

}

void loop() {

if(Serial3.available(>13){
for(i=0;i<14;i++){
telegramma[i]=Serial3.read();
}
}

```

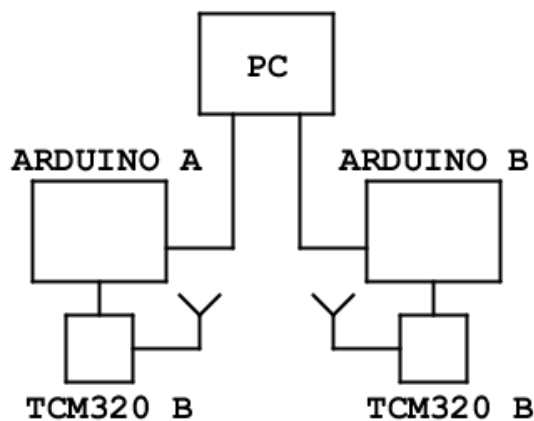
```

ID = (telegramma[8]<<24)+(telegramma[9]<<16)+(telegramma[10]<<8)+telegramma[11];
DATA = (telegramma[4]<<24)+(telegramma[5]<<16)+(telegramma[6]<<8)+telegramma[7];
Serial.println(ID,HEX);
Serial.println(DATA,HEX);

delay(1000);
}
}

```

Per quanto riguarda gli ottetti di identificazione (nell'esempio precedente gli ultimi 4 byte del vettore DATA[14]) essi non sono settabili dall'utente. Per verificare ciò si è proceduto come segue:



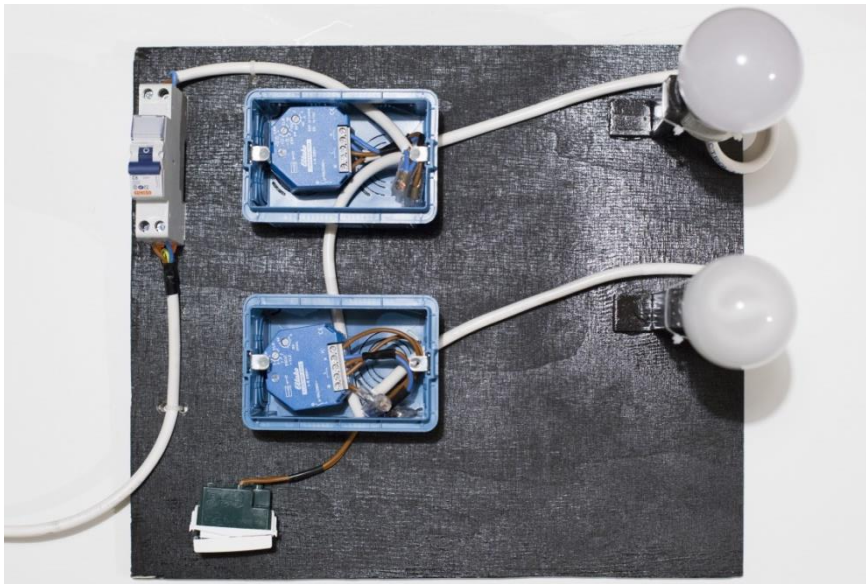
Si è caricato il firmware_trasmettitore all'Arduino A e il firmware_ricevitore all'Arduino B. Il vettore DATA è stato posto pari a zero per semplicità di notazione. Il pacchetto dati ricevuto dall'Arduino B presenta il seguente id :

ID_TCM32_A= Ox87C392 (da A a B)

Invertendo i ruoli invece (ovvero semplicemente scambiando i firmware) il pacchetto ricevuto da A (che in questo caso funge da ricevitore) presenta il seguente id:

ID_TCM32_B = Ox87D14D2 (da B a A)

Pannello domotico



Come risultato finale, il seguente Firmware consente di pilotare il pannello tramite PC (o da Arduino):

```
int SYNC_BYTE1 = 0xA5;
int SYNC_BYTE2 = 0x5A;
int H_SEQ_LENGTH = 0x6B;
int ORG = 0x05;
int PULSANTIERA[2][8]={
{0x30,0x0,0x0,0x0,0x0,0x0,0x0,0x0},
{0x70,0x0,0x00,0x00,0x0,0x0,0x0,0x0}
};
int i;
int STATUS = 0x30;
int CHECKSUM=0;
int TELEGRAMMA[8];

void setup() {
  Serial.begin(9600);
  Serial3.begin(9600);
}

void loop() {

  if(Serial.available(>0){
    switch (Serial.parseInt()) {
    case 1:
      //pulsante 1
      for(i=0; i<8;i++){
        TELEGRAMMA[i] = PULSANTIERA[0][i];
      }
      break;
```

```

case 2:
//pulsante 2
for(i=0; i<8;i++){
TELEGRAMMA[i] = PULSANTIERA[1][i];
}
break;
}

for(i=0;i<8;i++){
CHECKSUM = CHECKSUM + TELEGRAMMA[i];
}

CHECKSUM = lowByte(H_SEQ_LENGTH + ORG + CHECKSUM + STATUS);

Serial3.write(SYNC_BYTE1);
Serial3.write(SYNC_BYTE2);
Serial3.write(H_SEQ_LENGTH);
Serial3.write(ORG);
Serial3.write(TELEGRAMMA[0]);
Serial3.write(TELEGRAMMA[1]);
Serial3.write(TELEGRAMMA[2]);
Serial3.write(TELEGRAMMA[3]);
Serial3.write(TELEGRAMMA[4]);
Serial3.write(TELEGRAMMA[5]);
Serial3.write(TELEGRAMMA[6]);
Serial3.write(TELEGRAMMA[7]);
Serial3.write(STATUS);
Serial3.write(CHECKSUM);
CHECKSUM=0;
}

if(Serial3.available(>13){
Serial.println("Risposta del TCM320:");
for(i=0; i<14;i++){
Serial.println(Serial3.read(),HEX);
}
Serial.println("-----");

}
}

```

Il codice non è completo (ma comunque funzionante, possiamo quindi dire che è fine a se stesso) in quanto in questa sede ci interessa solo evidenziare alcuni aspetti dei protocolli EnOcean. In PULSANTIERE[8][8] sono contenuti i Byte dati “clonati” dalla pulsantiera Eltako a disposizione (FMH4-ws). I byte di ID sono posti a 0x00 in quanto questo campo viene completato dai moduli stessi (l’identificativo dipende appunto dal modulo stesso).

I Dati inviati dalla pulsantiera dipendono ovviamente dal pulsante premuto (premuta, rilasciato, premuto per qualche secondo ecc.), il vettore PULSANTIERA[2][8] contiene i Byte dati relativi a pulsante 1 e pulsante 2 premuti.

Da monitor seriale vengono (utilizzando appunto l'IDE di Arduino) inviati i dati da tastiera, "1" o "2" (che corrispondono rispettivamente pulsante 1 e pulsante 2).

Si procede dunque con teaching-in del FSR61VA-10A, si pone il potenziometro superiore su LRN e il potenziometro inferiore in "right stop" e si invia il comando "1". Analogamente, si procede allo stesso modo mettendo il potenziometro superiore su LRN e quello inferiore in "left stop", mandando in seguito il comando "2".

Ora è possibile attaccare o staccare carichi elettrici mediante PC, applicazioni per smartphone, dispositivi di controllo carichi ecc.

Fonti

- Internet;
- "Guida alle novità della norma CEI 64-8 7° edizione (ABB)";
- User manual TCM320 v1.43 (EnOcean);
- "Efficacia energetica negli edifici-il contributo della domotica e della building automation" (Baggini-Marra; Editoriale Delfino)

Ringraziamenti

"Grazie di cuore a chiunque mi abbia in qualche modo aiutato a vincere l'inerzia della quotidianità fornendomi energia. Ancora una volta, si chiude un'era e se ne apre un'altra."

Torino, 2016