

# UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di laurea in Ingegneria Meccanica e Meccatronica

---

*TESI DI LAUREA TRIENNALE*

## STUDIO DI UNA SCHEDA DI CONTROLLO PER UNA STUFA A PELLETTI

*Relatore:* Prof. Ing. DIEGO DAINESE

*Laureando:* DIEGO FRANCHETTI

Matricola 1048421

ANNO ACCADEMICO 2015-2016



.....Siamo buoni a nulla ma capaci di tutto.....

.....Il più bel regalo che si possa fare ad uno studente è il dubbio.....



## INDICE

INTRODUZIONE .....	7
CAPITOLO 1: IL MICROCONTROLLORE	
1.1 Generalità.....	9
1.2 Componenti microcontrollore.....	11
1.2.1 Processor Core .....	12
1.2.2 Memorie .....	14
1.2.3 Digital I/O .....	17
1.2.4 Analog I/O.....	18
1.2.5 Interrupts .....	18
1.2.6 Watchdog o watchdog timer .....	18
1.2.7 Interfaccia di comunicazione .....	19
1.2.8 Microcontrollore RX1115 .....	22
CAPITOLO 2: COMPONENTI IMPIEGATI	
2.1 Generalità sui sensori.....	27
2.2 Sensore di temperatura .....	28
2.2.1 Descrizione generale PCT2075D.....	30
2.2.2 Interfaccia seriale I <sup>2</sup> C-bus .....	32
2.3 Ricetrasmittitore ad ultrasuoni .....	35
2.3.1 Generalità .....	35
2.3.2 Descrizione generale SRF01 .....	37
2.4 Modulo GPS-GSM .....	39
2.4.1 Generalità .....	39
2.4.2 Descrizione generale SIM908 .....	41
2.5 RTC.....	43
2.5.1 Generalità .....	43
2.5.2 Descrizione generale RTC DS2417 .....	43
CONCLUSIONI.....	45
RINGRAZIAMENTI .....	47
BIBLIOGRAFIA.....	49
APPENDICE.....	51
A1 Differenza tra microcontrollore e microprocessore .....	51
A2 Programma main (piccolo programma che dimostra un'applicazione della scheda) .....	52



## **INTRODUZIONE**

Questa tesi è stata redatta dopo un periodo di stage nello studio E3 srl di Altavilla Vicentina.

Lo stage è stato conseguito con lo scopo di entrare in contatto con il mondo del lavoro, relazionarsi con persone che hanno un'esperienza nel campo dell'ingegneria, per mettere in pratica ciò che si è studiato durante questo triennio di studi universitari e per imparare ciò che non è scritto nei libri mettendosi in gioco.

L'obiettivo di questa trattazione è lo studio degli elementi essenziali per progettare una scheda elettronica di comando per una stufa a pellet, senza entrare nel dettaglio di marchi e comportamenti precisi di tali stufe. Il passo successivo può essere quello di progettare, realizzare e scrivere del software per produrre un prototipo di tale scheda e di considerare la possibilità di comandarla a distanza mediante un'app per smartphone appositamente creata. Ulteriormente si potrebbe iniziare una pre-serie di tali schede da immettere nel mercato.

In questa trattazione quindi si è partiti da una scheda già esistente con vari componenti, fornita dallo studio E3 con il relativo firmware, impiegata per la gestione dei cestini "intelligenti" pubblici, e ricavare da essa tutta la parte componentistica utile per la bozza di una nuova scheda per una stufa a pellet.

Ho avuto modo così di leggere del codice scritto da altri, comprenderlo, verificare e testare le varie funzioni del programma sui vari chip, moduli e sensori a bordo della scheda.

Da questa relazione quindi si farà una trattazione del microcontrollore, il cuore vero e proprio della scheda, sui vari sensori di temperatura e ultrasuoni, sul modulo GSM/GPS e sul chip RTC, fondamentale per ogni applicazione per scandire il tempo. Verrà poi proposto un possibile main di un programma, scritto in linguaggio C, da implementare per la scheda della stufa a pellet.



# MICROCONTROLLORE

## *1.1 Generalità*

Un semplice sistema di microcontrollore è composto da tre tipi di modulo tipicamente connessi come mostrato in Fig.1-1. I componenti sono:

1. Un microprocessore, che contiene la logica di controllo e il sistema dell'unità aritmetica;
2. Una memoria, che contiene il programma e dati;
3. Un sistema input/output (I/O), che contiene una o più porte che connettono dispositivi esterni come terminali, stampanti ecc...

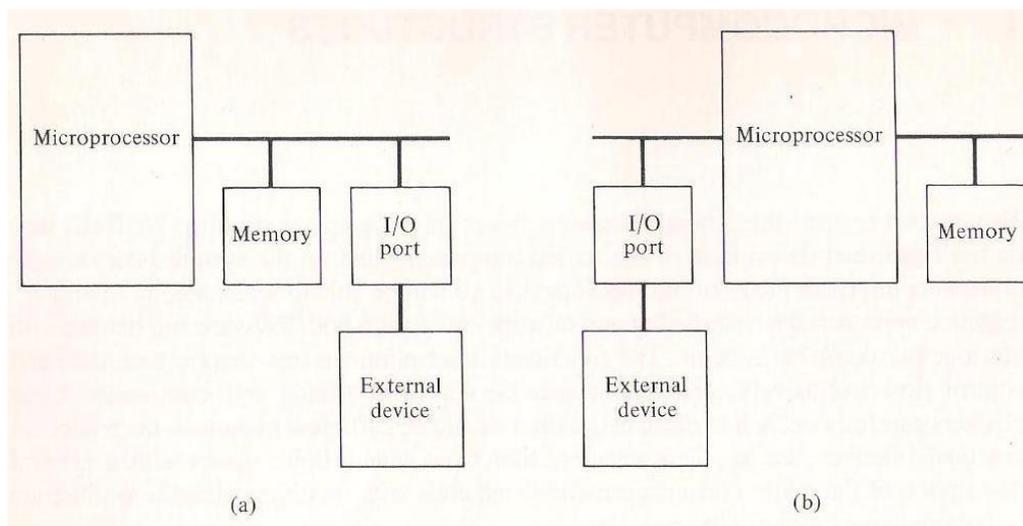


Fig. 1- 1 The basic structure of microcomputer: (a) one path system: (b) two path system.

Come rappresentato in Fig.1-1 sono possibili due macrostrutture di base: un sistema a bus singolo, Fig 1-1(a), attraverso il quale la memoria e il sistema di I/O comunicano con il processore. In questo sistema il processore è il master, esso inizializza tutte le attività nel bus inviando i comandi dalla memoria e dal sistema di I/O. Il bus trasporta solo una transazione alla volta quindi i comandi sono inviati sequenzialmente.

Nel sistema a doppio bus, Fig.1-1 (b), la memoria e il sistema di I/O hanno bus indipendenti per il processore, si ha così la possibilità che due differenti transazioni possono essere attivate nello stesso tempo, una per ogni bus cioè, il processore può inviare comandi alla memoria simultaneamente ai comandi di sistema I/O.

In entrambi i sistemi raffigurati in Fig.1-1 il comportamento del sistema base è lo stesso.

Le interazioni tra il processore e la memoria è tipicamente una ripetizione della sequenza:

1. Prelevare un'istruzione dalla memoria;
2. Esecuzione dell'istruzione (leggendo il dato dalla memoria o scrivendo il dato in essa).

Uno speciale registro, il program counter (PC), controlla l'istruzione successiva da eseguire. L'esecuzione di ciascuna istruzione modifica il PC in un indirizzo prescritto, cioè quando il processore ha eseguito un'istruzione, il PC carica la nuova istruzione da eseguire. L'esecuzione di una singola istruzione, in generale, comporta una o più transizioni del bus addizionale che dipende dall'istruzione eseguita.

Per esempio, il processore potrebbe leggere dalla memoria o scrivere in essa per scambiare dati tra i registri interni e la memoria stessa.

Per le transizioni di I/O il processore potrebbe ottenere stati o dati da una porta di I/O o inviare istruzioni o dati a tale porta. Tuttavia il processore non effettua transizioni mediante bus durante la fase di una istruzione perché potrebbe alterare altri registri interni del PC, che aggiorna ogni istruzione. La Fig.1-2 mostra graficamente una esecuzione di istruzione, includendo la relativa sincronizzazione delle transizioni della memoria e le attività indipendenti del processore.

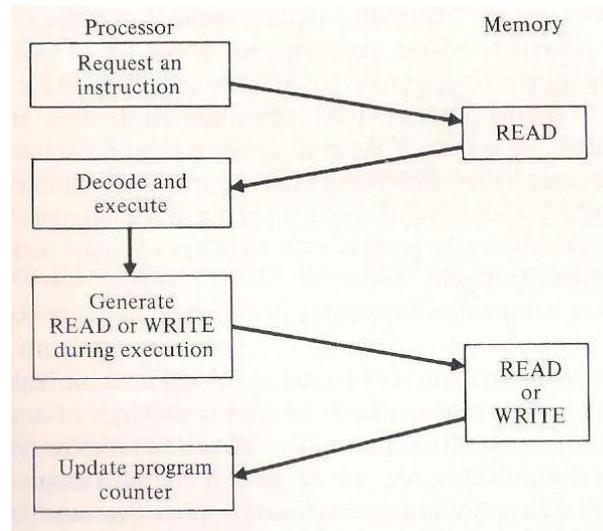


Fig. 1- 2 The time sequence for processor/memory interactions during the execution of a single instruction.

Alcuni computer, come detto, hanno in comune un bus per la memoria e I/O. Ma I/O e la memoria hanno tempi di sincronizzazione diversi, infatti la memoria richiede un tempo fisso massimo entro il quale risponde ad un processo. Tuttavia il sistema I/O ha il controllo degli eventi esterni che sono totalmente indipendenti dalla sincronizzazione del computer.

Quando un problema di processo invia un comando a una porta di I/O per accedere a un dato da una sorgente esterna, questo dato può arrivare in qualsiasi momento e il nuovo stato del processo sarà imprevedibile. Il processore deve essere in grado di sincronizzare le sue attività con gli eventi esterni.

Molte implementazioni supportano generalmente due tipi di sincronizzazione descritte di seguito. Sono:

1. Interrupts;
2. Stati periodici di controllo.

Il sub sistema I/O di sincronizzazione è attivato dal processore quando avvengono eventi esterni attraverso il servizio di richiesta trasmesso al computer attraverso il bus di I/O. Questo segnale è usualmente chiamato richiesta di interrupt e fornisce un mezzo per la sincronizzazione degli I/O. L'altra via di sincronizzazione serve al processo per interrogare le porte per avere informazioni sul loro stato.

L'interrogazione di una porta è normalmente una lettura di trasmissione di I/O nel bus e ciò che la porta ritorna è un dato che ne indica il suo stato, se la porta risponde con un fallimento il processore la interroga nuovamente. Tuttavia, quando il segnale di una porta è un interrupt, il segnale è mantenuto come un segnale attivo in un pin d'ingresso del microprocessore finché sarà riconosciuto dal processore.

Ritornando alla Fig.1-1 la differenza principale in questo sistema è che ci sono una o due connessioni dati al processore. Il sistema con due connessioni può eseguire simultaneamente delle transizioni in entrambe le connessioni risultando così più veloce del sistema con una connessione, di contro è richiesta una logica che consenta di gestire queste multi connessioni che è più complessa rispetto a quella singola. Quindi, il microprocessore in Fig. 1-1(b) è molto più complesso di quello di Fig.1-1(a).

C'è anche un'altra differenza tra i sistemi che è rappresentata in Fig. 1-3. Si può, infatti, notare un modulo separato chiamato DMA controller (direct memory-access) incorporato in un sistema di microcomputer.

Il DMA controller può inviare comandi alla memoria che si comporta esattamente come i comandi provenienti dal processore. In questo senso, il DMA controller è un secondo processore nel sistema, ma è dedicato a una funzione di I/O. Come mostrato in figura, il DMA controller connette uno o più porte I/O

direttamente alla memoria così i dati possono essere trasferiti tra le porte e la memoria senza passare attraverso il processore e senza alcun intervento del programma.

Questo processo sarà molto più veloce ed efficiente perché il canale DMA è specializzato per l'attività di trasferimento di dati. Quando il DMA controller è attivo, il processore deve essere inattivo e viceversa. Il DMA controller è normalmente usato per dispositivi che trasmettono dati ad alte velocità come i dati memorizzati su disco. La frequenza con cui sono trasmessi questi dati non danno la possibilità al microprocessore di controllarne il trasferimento byte a byte ed è per questo motivo che viene disattivato.

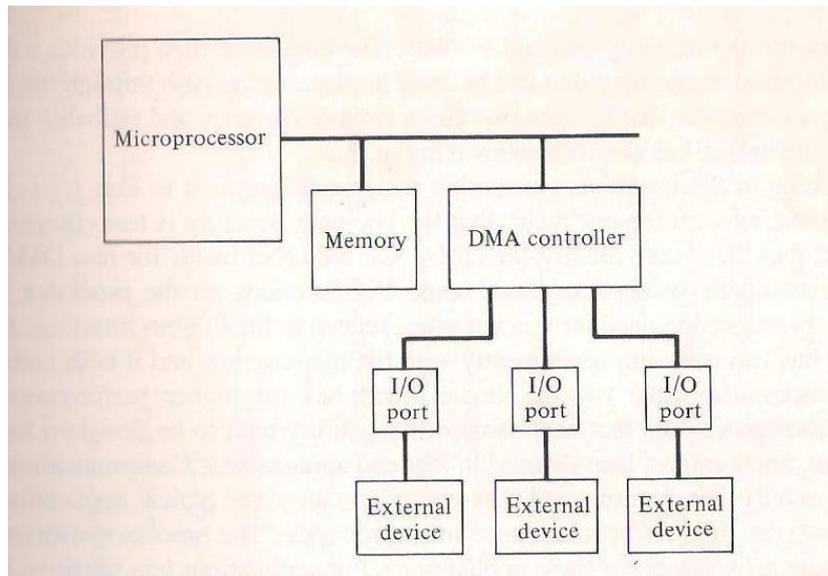


Fig. 1- 3 A microcomputer with a direct memory access controller.

## 1.2 Componenti microcontrollore

Le strutture di base dei vari microcontrollori, come già detto in precedenza, sono piuttosto simili.

La Fig. 1-4 mostra, in modo più complesso, lo schema a blocchi di un tipico microcontrollore: tutti i componenti sono collegati tramite un bus interno e sono integrati in un unico chip e i moduli sono collegati al mondo esterno tramite pin I/O.

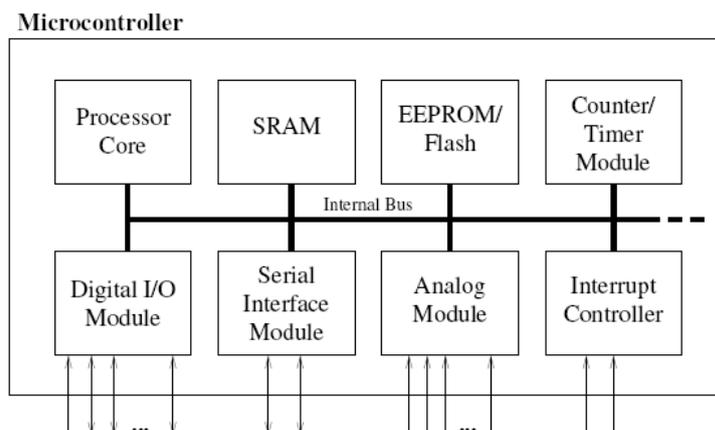


Fig. 1- 4 Basic layout of a microcontroller.

Di seguito si tratterà in dettaglio i vari componenti rappresentati in Fig. 1-4.

## 1.2.1 Processor Core

Il Processor Core è la parte principale di qualsiasi microcontrollore, è l'unità di elaborazione centrale (comunemente chiamata CPU, central processing unit). È detta unità centrale di elaborazione perché coordina in maniera centralizzata tutte le altre unità di elaborazione presenti nelle architetture hardware dei computer di elaborazione delle varie periferiche interne o schede elettroniche.

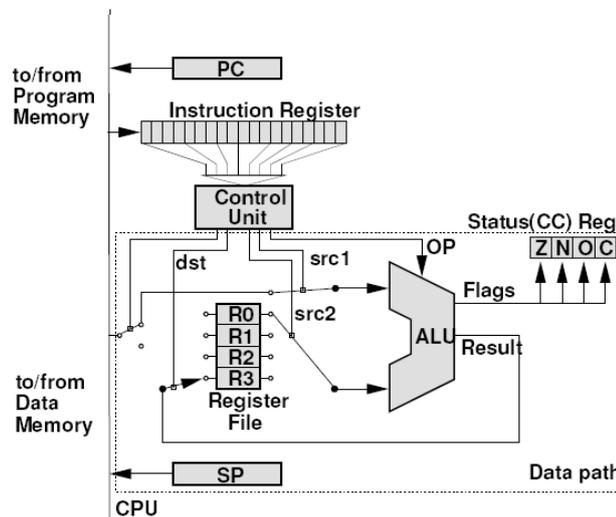


Fig. 1- 5 Basic CPU architecture.

La CPU ha un'architettura base, come rappresentato in Fig 1-5, composta da:

- **ALU (Arithmetic Logic Unit):** il cuore della CPU è l'unità logica aritmetica (ALU), che viene utilizzato per eseguire calcoli. Diverse linee di controllo selezionano le varie operazioni che l'ALU deve eseguire sui dati di input. L'ALU prende due ingressi e restituisce il risultato dell'operazione come uscita. I dati d'ingresso sono prelevati dai registri o dalla memoria. Inoltre, l'ALU memorizza alcune informazioni sulla natura del risultato nel registro di stato (chiamato anche condizione registro codice).
- **Register File:** contiene i registri di lavoro della CPU. Esso può consistere in un insieme di registri di uso generale ciascuno dei quali può essere la fonte o la destinazione di un'operazione, o è costituito da alcuni registri dedicati, ad esempio un accumulatore, che viene utilizzato per operazioni aritmetiche/logiche, o un registro indice, che viene utilizzato per alcune modalità di indirizzamento. In ogni caso, la CPU può prendere gli operandi per l'ALU dai registri, e può memorizzare il risultato dell'operazione in un secondo registro. In alternativa, gli operandi/risultati possono venire da/essere memorizzati nella memoria. Tuttavia, l'accesso alla memoria è molto più lento rispetto all'accesso ai register file per questo motivo, se possibile, è consigliabile utilizzare i registri.
- **Stack Pointer:** è una porzione di memoria situata in un preciso spazio che è usata dalla CPU per caricare indirizzi di ritorno dovuti a subroutine o a interrupt. Vi si accede mediante comandi di PUSH (inserimento del dato nello stack) e POP (prelevare il dato dallo stack) e segue la logica di tipo LIFO (Last In First Out), ovvero che l'ultimo elemento entrante sarà il primo ad uscire. Per memorizzare il livello di riempimento corrente dello stack, la CPU contiene un registro speciale chiamato stack pointer (SP), che indica la parte superiore dello stack e generalmente viene incrementato ogni qual volta vi sia un comando di PUSH e decrementato quando vi sia un comando di POP.

- **Control Unit:** a parte alcune situazioni speciali come il reset, la CPU esegue costantemente istruzioni di programma. È compito della Control Unit determinare quale operazione deve essere eseguita successivamente e configurare di conseguenza il percorso dei dati. Per fare ciò è usato un altro registro speciale, il program counter (PC), che viene utilizzato per memorizzare l'indirizzo della successiva istruzione di programma. La Control Unit carica questa istruzione nel instruction register (IR), decodifica l'istruzione, e imposta il percorso dei dati per eseguirla. La configurazione di percorso dei dati fornisce ingressi appropriati per l'ALU (da registri o della memoria), selezionando l'operazione ALU esatta, e fa in modo che il risultato sia scritto per la corretta destinazione (registro o memoria). Il PC viene incrementato per puntare alla successiva istruzione nella sequenza, o è caricato con un nuovo indirizzo nel caso di un salto o di una chiamata a una subroutine.

I microprocessori possono avere varie architetture a seconda dell'applicazione, dall'operazione da eseguire, dalla complessità richiesta e dal costo stesso. Negli anni si sono sviluppate quindi nuove strutture con varie caratteristiche. Di seguito verranno descritte quelle più importanti:

**CISC:** Un'architettura CISC (Complex Instruction Set Computer) è dotata di un set di istruzioni che consentono di eseguire operazioni anche complesse. I primi microprocessori erano di questa tipologia. I microprocessori con questa architettura erano pensati per essere programmati in assembler e quindi il progettista metteva a disposizione del programmatore istruzioni anche molto complesse che sfruttavano le peculiarità della macchina in modo da generare programmi veloci e che occupassero poca memoria. Queste istruzioni potevano nascondere alcune caratteristiche del processore per rendere la programmazione assembler più semplice. Per esempio potevano contenere istruzioni che formalmente modificavano i dati in memoria senza passare dai registri. In realtà queste istruzioni caricavano il dato in un registro nascosto, modificavano il dato e poi lo risaltavano in memoria. In sostanza creavano un illusorio teatro per il programmatore.

I processori CISC spesso sono dotati di un set di istruzioni definito con codici di lunghezza variabile e sono dotati di un numero ridotto di registri dato. I codici di lunghezza variabile permettono di sfruttare al pieno la memoria disponibile mentre i pochi registri interni derivano dai pochi transistor integrabili negli anni 80 e dal fatto che avendo istruzioni interne molto complesse spesso non sono necessari molti registri per eseguire i programmi.

**RISC:** I microprocessori RISC (Reduced Instruction Set Computer) invece sono basati sulla presenza di un numero ridotto di istruzioni rispetto a un microprocessore CISC e di un ridotto numero di modi di indirizzamento. Questo permetteva di realizzare architetture semplici e con pochi transistor, i transistor rimanenti potevano essere utilizzati per realizzare molti registri o per implementare strutture come le pipeline. Nei microprocessori RISC quasi tutte (se non tutte) le istruzioni erano eseguite in numero costante di cicli di clock (usualmente 1 o 2). Questo a differenza dei microprocessori CISC dove alcune operazioni erano eseguite in modo rapido (1 o 2 cicli di clock) mentre altre richiedevano decine di cicli di clock (usualmente le divisioni).

I processori RISC non erano nati per essere programmati in assembler dato che non implementavano semplificazioni per il programmatore che invece i processori CISC avevano. Questa tipologia di processori era nata per essere programmata con linguaggi ad alto livello che poi opportuni compilatori avrebbero tradotto in linguaggio macchina. I processori RISC normalmente hanno un set di istruzioni di lunghezza fissa per facilitare il caricamento di pezzi di codice ad alta velocità in modo semplice e sono quasi sempre dotati di molti registri (spesso centinaia) dato che per eseguire un programma spesso sono necessarie molte operazioni elementari e quindi molti registri ove salvare i dati intermedi.

## 1.2.2 Memorie

Sono sistemi in grado di conservare informazione e possono essere classificate secondo vari criteri:

- ordine di accesso (memorie ad accesso diretto o memorie ad accesso sequenziale);
- possibilità di scrittura (memorie a lettura-scrittura, memorie scrivibili una sola volta, memorie a sola lettura);
- velocità di lettura e scrittura;
- costo unitario;
- volatilità;
- tecnologia (elettronica, magnetica, ottica, magneto-ottica);
- gerarchia.

La suddivisione per gerarchia determina due grandi insiemi:

- Le memorie principali: sono anche dette memorie veloci per la grande rapidità con cui cedono dati alla CPU, contengono dati ed istruzioni prelevati dalla memoria di massa in attesa che questi siano a loro volta prelevati ed elaborati dal microprocessore, lavorando in modo strettamente accoppiata con esso, sono collocate all'interno del computer e si attivano quando questo viene acceso.
  - la RAM,
  - la ROM,
  - la cache.
- Mentre le memorie di massa: sono utilizzate per immagazzinare grosse quantità di dati in modo permanente. Sono classificate in base al tipo di supporto ed alla tecnologia di registrazione e di recupero dati che utilizzano:
  - le memorie magnetiche, basate sulla tecnologia delle testine magnetiche come i vecchi registratori,
  - le memorie ottiche basate sulla tecnologia laser,
  - le memorie stato solido.

Altra suddivisione degna di nota, come raffigurato in Fig 1-6, è tra:

- Memoria volatile:
  - In assenza di alimentazione perdono l'informazione e tornano in uno stato "vergine",
  - Le memorie possono essere "scritte e riscritte" dall'utente durante il normale funzionamento,
  - Scrittura e lettura sono ragionevolmente simmetriche.
- Memoria non-volatile:
  - Sistemi che conservano l'informazione anche in assenza di alimentazione,
  - L'informazione può essere scritta durante la realizzazione o dall'utente,
  - La scrittura è significativamente più lenta della lettura.

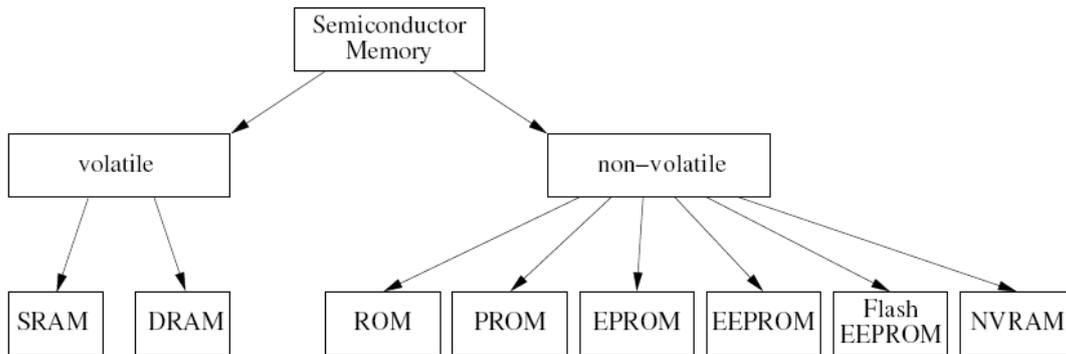


Fig. 1- 6 Types of semiconductor memory.

## Cenni sui vari tipi di memorie

Le memorie volatili, come scritto prima, mantengono il loro dato immagazzinato fino a che il sistema è alimentato, perdendolo in assenza di essa. A differenza delle memorie non volatili sono estremamente più veloci e per questo molto usate.

**SRAM:** acronimo di Static Random Access Memory, è un tipo di RAM volatile che non necessita di refresh. I banchi di memorie SRAM consentono di mantenere le informazioni per un tempo teoricamente infinito, hanno bassi tempi di lettura e bassi consumi, specialmente in condizioni statiche.

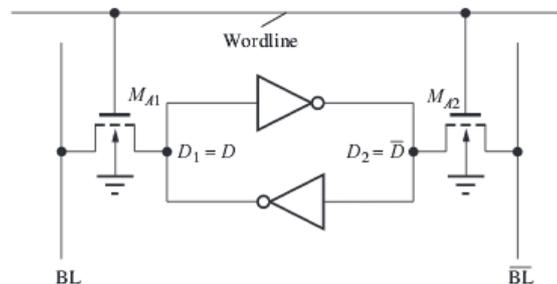


Fig. 1- 7 Basic memory cell formed from the two-inverter latch and two transistors.

Le celle di una SRAM sono costituite da un circuito retroazionato formato da due invertitori logici le cui uscite sono collegate alle due estremità alle linee dei dati tramite due transistor detti porte di trasmissione. Le singole coppie di porte di trasmissione vengono abilitate a seconda della cella su cui deve essere effettuata la lettura o scrittura (Fig. 1-7).

**DRAM:** acronimo di Dynamic Random Access Memory, è un tipo di RAM che immagazzina ogni bit in un diverso condensatore. Il numero di elettroni presenti nel condensatore determina se il bit è 1 o 0. Se il condensatore perde la carica, l'informazione è perduta: nel funzionamento la ricarica avviene periodicamente. Da qui la definizione di memoria dinamica, opposta alle memorie statiche come la SRAM. Viene definita anche volatile per la sua caratteristica di perdere le informazioni in mancanza di energia (Fig. 1-8).

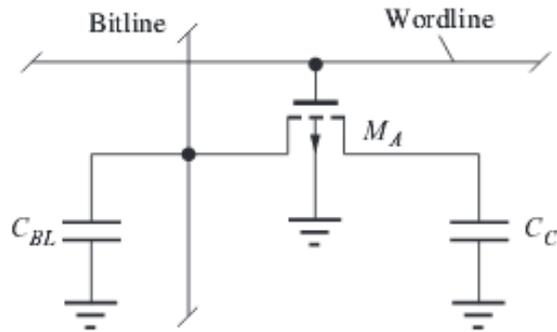


Fig. 1- 8 One-transistor (1-T) storage cell in which binary data is represented by the presence or absence of charge on  $C_C$ .

Ciascuna cella di una DRAM richiede un condensatore allo stato solido per memorizzare il bit di informazione e un transistor MOS per controllare l'accesso alla cella. Per questo motivo le celle DRAM sono più piccole e più economiche delle celle utilizzate per le RAM statiche SRAM, le quali usano latch che richiedono l'uso di quattro o sei transistori MOS.

Le memorie non-volatili conservano l'informazione anche quando non sono alimentate, sono utilizzate comunemente come memorie secondarie per memorizzare dati intermedi o per memorizzare dati da archiviare e sono più lente rispetto alle memorie volatili.

**ROM:** acronimo di Read Only Memory è una memoria di sola lettura (ovvero non modificabile, non scrivibile) e sono le prime memorie non-volatili a semiconduttore, contengono il BIOS, software (detto firmware) scritto dal costruttore e costituito da un insieme di istruzioni necessarie all'accensione del sistema.

**PROM:** acronimo di Programmable Read-Only Memory è una tipologia di memoria informatica, in particolare una tipologia memoria scrivibile una sola volta a stato solido.

**EPROM, EEPROM:** acronimo di Erasable Programmable Read Only Memory, è una memoria di sola lettura cancellabile tramite raggi ultravioletti. La EPROM può essere totalmente cancellata (per un numero limitato ma consistente di volte) e riprogrammata a piacimento. La scrittura avviene forzando una carica elettrica in un piccolo pezzo di materiale policristallino detto gate flottante localizzato nella cella di memoria (sono utilizzati gli FPGA). Usualmente la presenza di carica corrisponde ad uno zero logico e la sua assenza corrisponde ad un uno. La sostanziale differenza tra le memorie EPROM e EEPROM (acronimo di Electrically Erasable Programmable Read-Only Memory) è che le memorie EEPROM possono essere cancellate, per essere riscritte, elettricamente senza dover ricorrere ai raggi UV.

**Flash:** è una tipologia di memoria a stato solido, di tipo non volatile, che per le sue prestazioni può anche essere usata come memoria di lettura-scrittura. Quando viene utilizzata come ROM viene anche chiamata flash ROM. Diversamente dalle tecnologie precedenti, la tecnologia Flash ha reso possibile il salvataggio o la cancellazione di dati in un unico passo, introducendo quindi un incredibile guadagno in velocità

**Cache:** è una memoria utilizzata dalla CPU per ridurre i tempi d'accesso ai dati presenti nella RAM. Quando è necessario l'accesso ad un dato, una copia di questo viene prima cercata nella cache: se è presente e valida, viene utilizzata tale copia, altrimenti il dato viene recuperato dalla memoria principale, e memorizzato nella cache, nel caso possa servire successivamente. La cache è un tipo di memoria piccola, ma molto veloce, che mantiene copia dei dati ai quali si accede più frequentemente. Questo rende i dati ancor più veloci nel passaggio dalla RAM alla CPU. Inoltre funge da adattatore di velocità tra la CPU (più veloce) e la RAM (più lenta).

Le operazioni effettuate sulla memoria di un computer sono fondamentalmente le seguenti:

- *Inizializzazione.* È il trattamento che subisce la memoria prima dell'uso normale. Alcuni tipi di memoria (per esempio la RAM elettronica) non hanno bisogno di inizializzazione; per tali memorie, il contenuto iniziale è imprevedibile. Per le memorie a sola lettura, l'inizializzazione consiste nella scrittura dei dati effettuata in fase di produzione. Per i dischi magnetici, l'inizializzazione consiste nella scrittura di dati che costituiscono un casellario in cui porre le informazioni utili (la cosiddetta formattazione).
- *Scrittura.* È l'operazione di memorizzazione delle informazioni.
- *Letture.* È l'operazione di recupero di informazioni memorizzate. Un esempio di lettura è chiedere alla memoria il contenuto della cella di indirizzo 1000. Si accede inoltre alla memoria e ai suoi contenuti ogni qual volta l'Unità di Controllo (Control Unit) del Processore richiede dati utili su cui eseguire un'operazione di elaborazione. Tutto ciò avviene grazie alle procedure di indirizzamento gestite dall'Address Logic, sottosistema del processore.

Per parlare indifferentemente di lettura o di scrittura, si usa il termine accesso. Per esempio, per indicare che una memoria è veloce sia in lettura che in scrittura, si dice che ha un basso tempo di accesso.

I calcolatori per immagazzinare in memoria dati di dimensioni superiore al byte utilizzano fondamentalmente due modalità differenti conosciute come big-endian e little-endian che definiscono così l'ordine di immagazzinamento dei byte che costituiscono il dato.

- little-endian: memorizzazione che inizia dal byte meno significativo per finire col più significativo, è utilizzata dai processori Intel;
- big-endian: memorizzazione che inizia dal byte più significativo per finire col meno significativo, è utilizzata dai processori Motorola e nei protocolli usati in Internet;

### ***1.2.3 Digital I/O***

Permettono la possibilità di monitorare e controllare l'hardware ed è la principale caratteristica dei microcontrollori. Di conseguenza tutti i microcontrollori possiedono almeno 1-2 digital I/O pin che possono essere connessi direttamente all'hardware. In genere, si possono trovare 8-32 pin e più in alcuni microcontrollori.

I pin di I/O sono generalmente raggruppati in porte di 8 pin, alle quali si può accedere a esse con un singolo byte. I pin possono essere solamente di input o solamente di output o più comunemente bidirezionali cioè hanno la possibilità di essere sia di input che di output. Oltre alla loro capacità di essere pin digitali I/O, molti pin hanno altre funzioni alternative per salvare spazio e snellire la struttura del microprocessore, infatti tutti gli altri moduli del controllore richiedono pin I/O, come il modulo analogico o il timer e usano funzioni alternative dei pin digitali I/O. Il programmatore può selezionare quale funzione deve essere utilizzata per il pin in modo tale da attivare la funzionalità appropriata per il modulo in considerazione. Se un pin è usato per un modulo analogico, lo stesso perderà la sua funzione di pin digitale I/O e viceversa, quindi il progettista dell'hardware dovrà scegliere con cura le funzioni da attribuire ai pin. Quando si parla di digital I/O, si intende che il valore del pin, dal punto di vista del controllore è 1 o 0 dove nella logica positiva 1 definisce il valore alto mentre 0 il valore logico basso.

I digital I/O pin possono essere:

- Data Direction Register (DDR): sono pin bidirezionali che devono essere determinati;
- Port Register (PORT): è usato per controllare il livello di tensione dei pin output;
- Port Input Register (PIN): è generalmente un pin di sola lettura.

### 1.2.4 Analog I/O

Si è visto che per acquisire un segnale analogico si utilizzano i pin digitali del microcontrollore e quindi si deve convertire, mediante il modulo analogico del microcontrollore (DAC-ADC), tale segnale in modo da avere una corrispondenza tra il segnale di tensione, in ingresso o in uscita, con il fenomeno esterno sotto esame.

Il microcontrollore equipaggiato con input analogici normalmente ha dai 4 ai 16 canali analogici di input (multiplexer) e sono incanalati in un unico ADC interno. Di conseguenza, i canali analogici non possono essere letti in contemporanea ma devono essere letti sequenzialmente. In pratica questo significa dire che il modulo analogico viene configurato per iniziare la conversione del segnale, una volta conclusa è possibile leggere i valori configurando il modulo per il canale successivo e il trigger per la conversione.

### 1.2.5 Interrupts

È una condizione che consente al microcontrollore di lavorare temporaneamente su una diversa operazione da quella prevista dal programma e poi ritornare al regolare flusso di istruzioni dal quale si era interrotto.

Gli interrupt possono essere interni o esterni. Gli interrupt interni, o "software interrupts", sono innescate da un'istruzione del software e operano un salto. Un interrupt esterno, o "hardware interrupt", è causato da un modulo hardware esterno. Ad esempio, molti sistemi informatici utilizzano interrupt guidati dove premendo un tasto sulla tastiera o clic su un pulsante del mouse attiva un interrupt. Il processore interrompe l'operazione che stava eseguendo, legge l'input da tastiera o dal mouse, e poi ritorna al programma in corso.

La Fig. 1-9 rappresenta cosa accade quando viene attivato un interrupt.

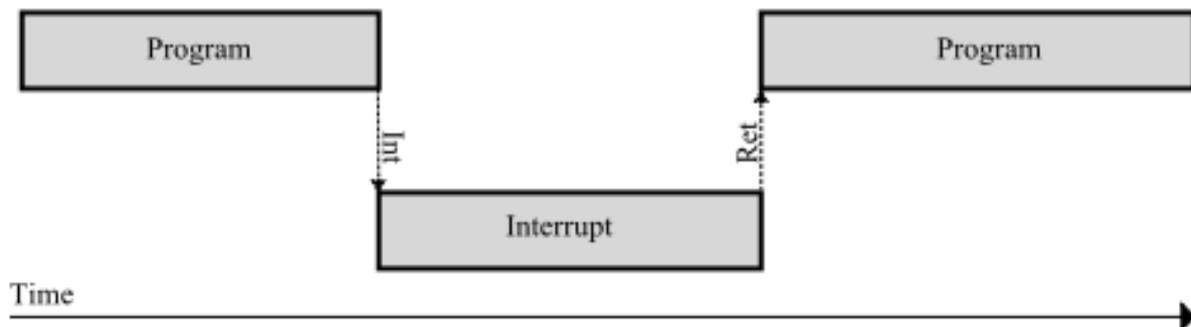


Fig. 1- 9 Activation of an interrupt.

### 1.2.6 Watchdog o watchdog timer

In italiano: "temporizzazione di supervisore" - letteralmente: "cane da guardia" è un sistema di temporizzazione hardware che permette alla CPU la rilevazione di un loop infinito di programma o di una situazione di deadlock.

Tale rilevazione può consentire di prendere dei provvedimenti per correggere la situazione, generalmente effettuando un reset del sistema. Possono essere implementati watchdog più complessi che consentano la memorizzazione delle informazioni di contesto per effettuare il debugging delle applicazioni che lo hanno causato. In questo caso un secondo watchdog più semplice dovrà assicurarsi che, se il primo non portasse a termine il suo compito in un determinato lasso di tempo, il sistema comunque venga resettato.

## 1.2.7 Interfaccia di comunicazione

La trasmissione di dati implica lo scambio di informazioni digitali tra due sistemi o dispositivi.

I microcontrollori contengono in genere diverse interfacce di comunicazione e talvolta anche più istanze di una particolare interfaccia, come due moduli UART. Lo scopo principale di tale interfaccia è quello di consentire al microcontrollore di comunicare con altre unità, come altri microcontrollori, periferiche ecc... L'attuazione di tali interfacce può assumere molte forme, ma fondamentalmente possono essere classificati secondo le loro proprietà: possono essere sia seriali o parallele, sincrone o asincrone, utilizzare una comunicazione bus o point-to-point, essere full-duplex o half duplex, e possono essere basate sia su un principio master-slave o costituiti da partner alla pari.

Un'interfaccia seriale invia i dati in sequenza, un bit alla volta. Chiaramente, questo metodo richiede solo una linea di dati, per cui la comunicazione è conveniente in termini economici. Sul lato negativo, il numero di bit di dati che possono essere inviati al secondo è basso. Una interfaccia parallela, invece, utilizza diverse linee di dati per trasferire più di un bit alla volta. Il numero di bit che vengono trasferiti in parallelo varia.

Asincrono significa, in questo contesto, che i dati sono trasmessi senza l'aggiunta di un segnale di clock, cioè senza alcun segnale comune tra trasmettitore e ricevitore destinato a sincronizzare il flusso di informazioni; ovviamente sia il trasmettitore che il ricevitore devono comunque essere dotati di un proprio clock locale per poter interpretare correttamente i dati.

Una trasmissione è di tipo point-to-point quando nella comunicazione è presente, per ciascun segnale utilizzato, un solo trasmettitore ed un solo ricevitore; tale termine può essere contrapposto a una trasmissione bus, che indica la situazione in cui, a fronte di un trasmettitore, esistono più ricevitori.

Half-duplex e full-duplex, i due termini fanno riferimento alla situazione in cui due dispositivi si scambiano informazioni tra di loro, comportandosi entrambi sia da sorgente di informazioni (cioè da talker o, in sigla, Tx) sia da ricevitore (listener o Rx).

- Half-duplex indica che la trasmissione è bidirezionale ma non contemporanea nei due versi: in un determinato istante uno solo dei due dispositivi emette segnali, l'altro ascolta. Quando è necessario, si scambiano di ruolo.
- Full-duplex indica che la trasmissione è bidirezionale e contemporanea. In questo caso sono necessari due fili, uno per ciascun verso di trasmissione. In alcuni sistemi di comunicazione, quali il comune telefono, possono essere adottati meccanismi che permettono la trasmissione full-duplex con un solo filo.
- Simplex se la trasmissione è sempre in un solo verso.

In Fig. 1-10 sono rappresentati i diagrammi a blocchi della modalità asincrona e sincrona di trasmissione dei bit.

### • Modalità Asincrona



### • Modalità Sincrona

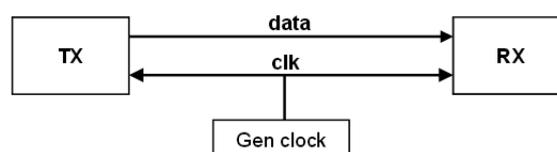


Fig. 1-10 Asynchronous and synchronous mode of data transfer.

Le unità di misura usate per la velocità di trasmissione su una linea seriale sono:

- bps: numero di bit trasmessi per secondo,
- baud: numero di transizioni trasmesse per secondo.

Solo se il segnale trasmesso può assumere due soli possibili stati, allora baud e bps sono equivalenti.

I protocolli di comunicazione dati qui trattati sono:

**SCI(UART):** L'interfaccia di comunicazione seriale (Serial Communication Interface SCI) fornisce una interfaccia di comunicazione asincrona (Universal Asynchronous Receiver Transmitter, UART). Il modulo UART utilizza due fili, una trasmissione (TXD) e una linea di ricezione (RXD), per la comunicazione full o half-duplex.

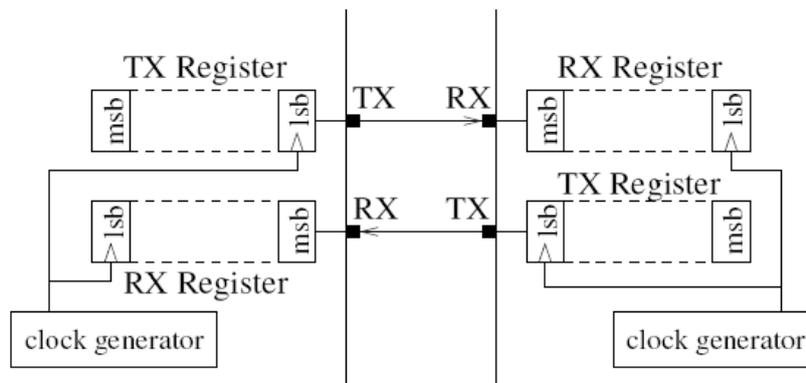


Fig. 1- 11 Basic structure of a UART module.

La Fig. 1-11 mostra la struttura interna di una UART. Fondamentalmente, il modulo è costituito da una trasmissione, una ricezione e un registro per contenere i dati.

L'UART non è un protocollo di comunicazione assistente, ma un modulo che può essere utilizzato per la comunicazione seriale asincrona. Quindi, il modulo UART posto in un microcontrollore consente all'applicazione di controllare gran parte del suo comportamento.

**USART:** Synchronous Asynchronous Receiver Transmitter Universale (USART) estende la funzionalità dell' UART da un modulo di trasmissione sincrona. L' USART ha quindi una terza linea supplementare che porta un segnale di clock. In modalità sincrona, il segnale di clock è generato da uno dei partner di comunicazione ed è usato sia per la trasmissione e la ricezione dei dati. Naturalmente, questa comunicazione sincrona rende il meccanismo sovracampionato quindi la modalità sincrona è più veloce della modalità asincrona. Il modulo USART combina la logica sia per la comunicazione sincrona e asincrona. Se si utilizza la comunicazione asincrona, la linea di clock è libera e può essere generalmente usato come un pin I/O digitale normale.

**SPI:** Serial Peripheral Interface (SPI) è una semplice interfaccia sincrona point-to-point basata su un principio master-slave che permette a un dispositivo definito come master di iniziare una comunicazione con un altro dispositivo definito come slave. Il segnale di clock è generato esclusivamente dal dispositivo master che sincronizza l'intera comunicazione e controlla quando il dato può cambiare e quando può essere pronto per essere letto. I dispositivi slave quindi non possono manipolare il clock e la comunicazione avviene con uno scambio contemporaneo di dati tra i due dispositivi.

Esso fornisce comunicazione full-duplex tra un master (generalmente un controllore) e uno (o più) slave (generalmente periferiche). L'interfaccia consiste in quattro linee single ended:

- MOSI (Master Out, Slave In): Questa linea è usata dal master per trasmettere i dati allo slave,
- MISO (Master In, Slave Out): Questa linea è utilizzata dallo slave per trasmettere i dati al master,
- SCK (System Clock): Questa linea è usata dal master per trasmettere il segnale di clock,
- SS (Slave Select): Questa linea è usata dal master per selezionare uno slave.

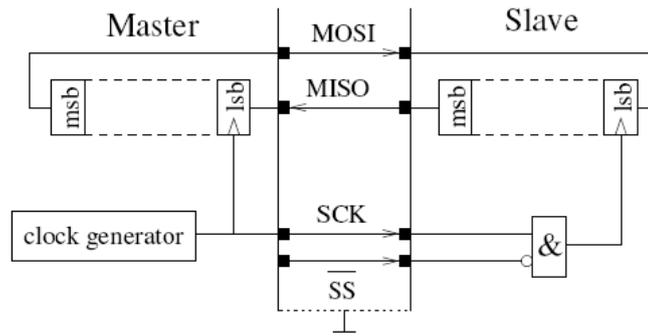


Fig. 1- 12 SPI interface.

La Fig. 1-12 mostra il principio di base di funzionamento dell'interfaccia SPI. Sia il master che lo slave hanno un registro a scorrimento interno, che è gestito dalla SCK. Ad ogni impulso di clock, l'MSB (o LSB, che in genere è configurabile) del registro SPI del master viene spostato sulla linea MOSI e spostato nel registro SPI dello slave come LSB. Allo stesso tempo, msb dello slave viene trasmessa oltre la linea MISO nel registro del master come LSB. Dopo 8 cicli di clock, il master e lo slave si sono scambiati tutti gli 8 bit nei loro registri. In generale quando sono presenti più dispositivi Slave si implementa un ulteriore segnale attivo basso definito come SS (Slave Select) che abilita uno slave alla volta. Se lo slave è unico allora tale linea può anche essere omessa, anche se una rigorosa implementazione di tale protocollo la impiegherebbe. L'impiego principale di questo piedino consiste nel tenere un dispositivo disabilitato fino poco prima dell'invio dei dati in modo da prevenire malfunzionamenti dovuti a segnali spuri o disturbi in genere. Tipicamente il dato presente sulla linea cambia durante il fronte di salita o di discesa del clock, in modo da far avvenire la lettura del dato nel fronte opposto al momento di cambiata. A seconda di questi particolari, si ottengono vari tipi di comunicazione SPI, ognuno dei quali deve essere implementato a seconda del dispositivo che si deve interfacciare.

**I<sup>2</sup>C:** Il bus Inter-IC (detto anche IIC o I2C) è un bus sincrono che opera su un principio master-slave. Esso utilizza due fili single-ended SCL (linea di clock seriale) e SDA (Serial Line dati) per la comunicazione half-duplex. Il protocollo è stato sviluppato da Philips ed è ampiamente utilizzato per la comunicazione tra uno o più controllori e dispositivi periferici posti a breve distanza.

Il protocollo consente la coesistenza di diversi master. Il ruolo del master normalmente è ricoperto dal microcontrollore, con tutti i dispositivi periferici come semplici slave. In un sistema con diversi microcontrollori, si può scegliere per ogni controller se debba essere un master o uno slave. L'unica condizione è che ci deve essere almeno un master nel sistema. Uno dei principali vantaggi del bus I<sup>2</sup>C è la sua facile estensibilità. Nuovi dispositivi possono essere aggiunti al bus semplicemente collegandoli. Non vi è alcun limite specifico al numero di dispositivi collegati al bus finché la massima capacità di bus 400 pF non venga superata.

Per un'ulteriore descrizione sul protocollo I<sup>2</sup>C si rimanda alla parte del sensore di temperatura.

## 1.2.8 Microcontrollore Renesas RX1115

Il microcontrollore preso in esame in questo elaborato appartiene alla famiglia Renesas RX111 in particolare modo il microcontrollore Renesas RX1115R5F51115ADFM34 con un' architettura CISC.

In Fig. 1-13 è definita la legenda con la quale è possibile risalire alle caratteristiche identificative del microcontrollore:

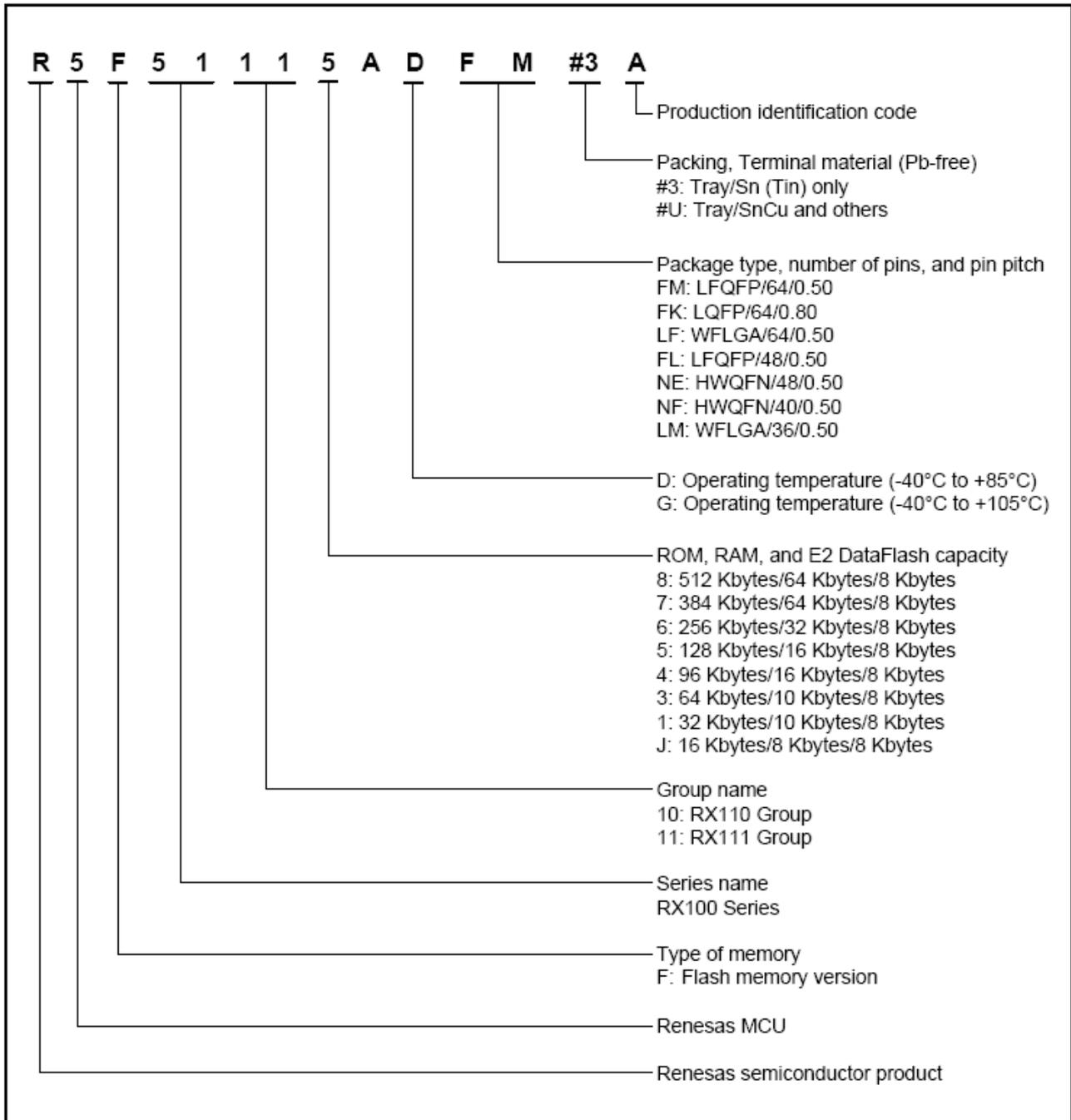


Fig. 1-13 How to read the product part No., memory capacity and package type.

Il microcontrollore Renesas RX115 è caratterizzata dal fatto di avere le seguenti caratteristiche descritte in Tab. 1-1.

<b>Basic Information</b>	
Product Status	Mass Production
Family	RX
Series	RX100
Group	RX111

<b>Memory</b>	
Program memory	128 KB Flash memory
RAM	16 KB
Data Flash	8 KB

<b>CPU (Bit, Clock, DMA, External Bus etc)</b>	
CPU	RX (32-bit)
Max. Frequency	32 MHz
Sub-Clock (32.768 kHz)	YES
On-chip Oscillator Freq.	32 MHz
PLL	YES
Real-Time Clock (RTC)	YES
Power-On Reset	YES
Low Voltage Detection	YES
DMA Remarks	DTV only
External Interrupt Pins	9
I/O Ports	50

<b>Timers</b>	
16-bit Timers	8 ch
Watchdog Timers	1 ch
PWM Output	16
3-Phase PWM Output Function	YES

<b>Analog</b>	
A/D Converters	12-bit x 14 ch
D/A Converters	8-bit x 2 ch

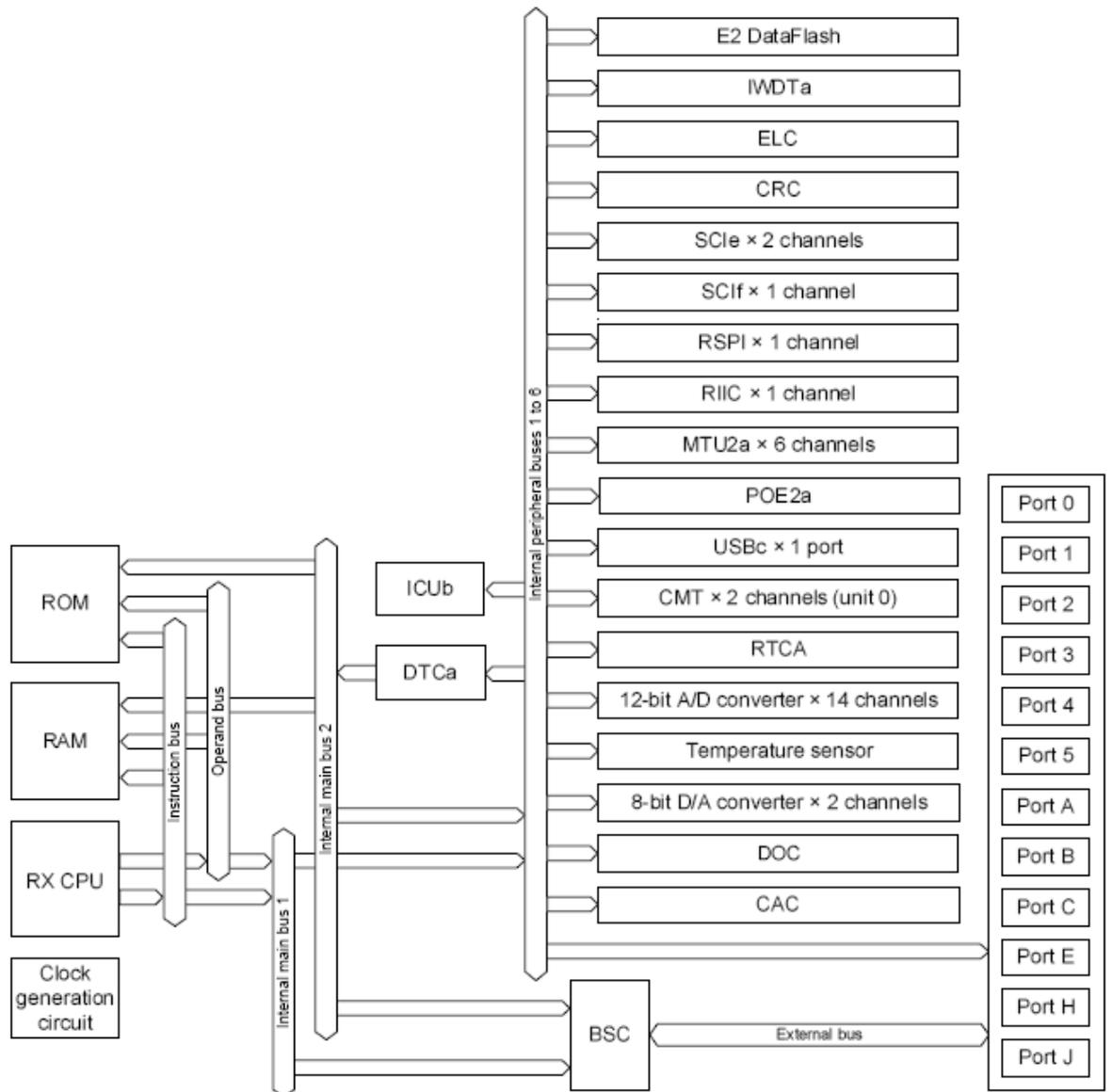
<b>Interfaces</b>	
USB Ports	1
USB Host	YES
USB Peripheral	YES
USB Remarks	USB2.0 Full Speed & USB OTG (On-The-Go) & Battery Charging Support
CSIs	3 ch
SPIs	4 ch
UARTs	3 ch
I2Cs	4 ch

<b>Operating Conditions</b>	
Operating Voltage	1.8 to 3.6 V
Operating Temperature	-40 to 85 °C (Ambient Temperature)

Tab. 1-1 Features.

Di seguito, in Fig. 1-14, è raffigurato il microcontrollore RX1115 tramite diagrammi a blocchi.



- |            |  |        |  |
|------------|--|--------|--|
| ICUb:      | Interrupt controller                     | MTU2a: | Multi-function timer pulse unit 2            |
| DTCa:      | Data transfer controller                 | POE2a: | Port output enable 2                         |
| IWDTa:     | Independent watchdog timer               | USBc:  | USB 2.0 host/function module                 |
| ELC:       | Event link controller                    | CMT:   | Compare match timer                          |
| CRC:       | CRC (cyclic redundancy check) calculator | RTCA:  | Realtime clock                               |
| SCle/SCIf: | Serial communications interface          | DOC:   | Data operation circuit                       |
| RSPI:      | Serial peripheral interface              | CAC:   | Clock frequency accuracy measurement circuit |
| RIIC:      | I <sup>2</sup> C bus interface           |        |  |

Fig. 1-14 Block diagram .

Come descritto precedentemente, il programmatore per comunicare con il microcontrollore, e viceversa, utilizza i pin I/O messi a disposizione dalla casa costruttrice del micro tramite opportuni registri.

Nel RX1115 sono presenti i seguenti principali registri:

- Port Direction Register (PDR): PDR sono utilizzati per selezionare la direzione di ingresso o uscita per ogni singolo pin per ogni porta  $m$  quando i rispettivi pin sono configurati come pin di I/O generali.  
Ogni bit di `PORT.DR` corrisponde ad ogni pin della porta  $m$ ; la direzione I/O può essere specificata in unità di 1-bit.  
Il bit `PORT3.PDR.B5` è riservato, in quanto il pin P35 è solo di ingresso. Un pin riservato viene letto come 0.
- Port Output Data Register (PODR): PODR contengono i dati in uscita dai pin utilizzati per le porte di uscita. Il bit `PORT3.PODR.B5` è riservata, in quanto il pin P35 è solo di ingresso. Un pin riservato viene letto come 0.
- Port Input Data Register (PIDR): PIDR indicano i singoli stati dei pin delle porte  $m$  e possono essere letti con il comando `PORT.PDR`, indipendentemente dai valori di `PORTm.PDR` e `PORTm.PMR`. Un pin riservato viene letto come indefinito e non può essere modificato.

Altra operazione importante implementata nel presente microcontrollore è la capacità di avere la funzione del risparmio di energia grazie alle modalità stand-by e sleep. Queste modalità permettono al microcontrollore di ridurre al minimo le proprie funzionalità, risparmiando in termini di energia di alimentazione e di dissipazione di calore. Una volta che il microcontrollore riceverà un segnale di interrupt dall'esterno o un interrupt previsto dall'operatore questi riprenderà a pieno regime le sue funzioni.



## COMPONENTI IMPIEGATI

### *2.1 Generalità sui sensori*

I sensori sono elementi fondamentali di un qualsiasi sistema di acquisizione dati o di misurazione. Dalla loro scelta o progettazione possono dipendere le caratteristiche e le prestazioni dell'intero sistema. I sensori oggi sono conosciuti e disponibili in commercio sono numerosi e diffusissimi. Tra i principali settori e campi di impiego nei quali sono più utilizzati si ricorda l'automazione industriale e degli edifici, il monitoraggio di grandezze fisiche ambientali, il controllo di grandezze in sistemi complessi come automobili, velivoli e satelliti e le misurazioni in ambito medicale.

Il termine sensore è spesso confuso con quello di trasduttore. Entrambi infatti identificano attività molto simili tra loro e spesso sono da ritenersi del tutto equivalenti. Da un punto di vista misuristico, il trasduttore è il dispositivo che effettua una trasformazione da una forma di energia all'altra. Il sensore è invece il dispositivo che fornisce in uscita un segnale elettrico legato alla grandezza in ingresso ma non necessariamente alla sua energia. I sensori sono quindi da ritenersi un sottoinsieme dei trasduttori.

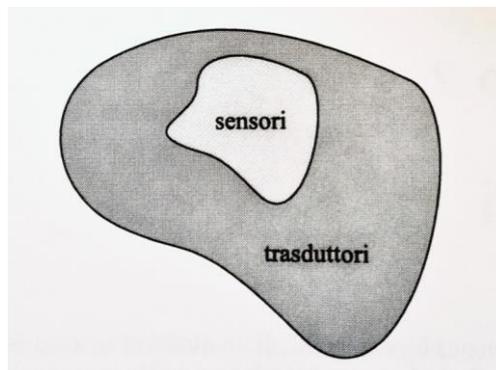


Fig. 2-1 Subset of the sensors

Un qualsiasi sensore, indipendente dalle sue caratteristiche e funzionalità, può essere rappresentato come in Fig. 2-2, vale a dire come una scatola nera, o black box, nella quale non si conosce nulla fuorché il legame tra l'ingresso  $x$  e l'uscita  $y$ .

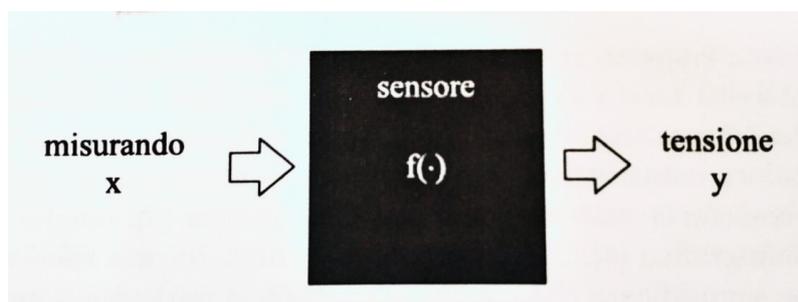


Fig. 2- 2 Sensor like a black box.

In un sensore,  $x$  è una qualsiasi grandezza fisica (misurando) che il sensore rileva e trasforma in una tensione elettrica  $y$  in uscita, normalmente disponibile sotto forma di differenza di potenziale tra due conduttori. Il legame ingresso-uscita è descritto dalla funzione caratteristica che associa a ciascun valore di  $x$  nell'intervallo  $(x_{\min}, x_{\max})$  un corrispondente valore di  $y$  nell'intervallo  $(y_{\min}, y_{\max})$ .

## 2.2 Sensore di temperatura

I sensori di temperatura utilizzati oggi sono molteplici, vengono classificati secondo le loro caratteristiche, il loro impiego e il costo. Di seguito si farà una breve presentazione dei sensori di temperatura classificandoli secondo i materiali utilizzati.

### Sensori di temperatura resistivi

**Sensori a filamento metallico:** spesso indicati con la sigla RTD (Resistance Temperature Detector) e chiamati anche termoresistori, basano il loro funzionamento sulla dipendenza della propria resistività dalla temperatura. In un materiale conduttore un aumento della temperatura provoca un aumento dell'agitazione termica degli atomi, che a sua volta ostacola il movimento degli elettroni di conduzione. La temperatura influenza anche le dimensioni geometriche del materiale in esame, provocando quindi un'ulteriore variazione di resistenza, che può essere non trascurabile e definita con l'espressione:

$$R = \frac{\rho l}{s} \quad (2.1)$$

dove  $\rho$  è la resistività elettrica del materiale,  $l$  è la lunghezza del conduttore e  $s$  la sua sezione.

Affinché l'influenza della temperatura sia sufficientemente marcata in modo da poter utilizzare questi dispositivi come sensori di temperatura, vengono utilizzate per la loro realizzazione particolari leghe, o miscele, che conferiscono le prestazioni che interessano, sia come sensibilità alla temperatura, sia come intervallo di temperatura nel quale possono essere impiegati. Nelle applicazioni in cui non viene richiesta un'accuratezza molto spinta e il campo di temperatura è abbastanza limitato, si può considerare il comportamento del RTD lineare con la temperatura. In questo caso si ha:

$$R_t = R_o[1 + \alpha\Delta T] \quad (2.2)$$

Dove  $R_o$  è la resistenza del RTD alla temperatura  $T_o$  e  $\Delta T = T - T_o$  è lo scostamento della temperatura rispetto a  $T_o$  e  $\alpha$  è un parametro che assume valori che dipendono dal tipo di materiale con cui è realizzato il termoresistore.

**Termistore:** è un sensore che trasforma variazioni di temperatura in variazioni di resistenza, ma a differenza delle termoresistenze, l'elemento sensibile è un semiconduttore. La dipendenza della resistenza di un termistore dalla temperatura è influenzata non solo dalla variazione delle sue dimensioni fisiche, ma anche da altri fattori quali il tipo di drogante e l'entità della drogatura cui il materiale semiconduttore è sottoposto. Con un'opportuna scelta della drogatura è possibile aumentare la sensibilità del termistore, ottenendo così una sensibilità alla temperatura molto maggiore di quella ottenibile con un metallo. I termistori possono essere divisi in due categorie, basate sull'andamento della resistenza all'aumentare della temperatura:

- NTC (Negative Temperature Coefficient),
- PTC (Positive Temperature Coefficient).

### Sensore di temperatura a semiconduttore

I sensori di temperatura a semiconduttore basano il loro funzionamento sulle proprietà di giunzioni a semiconduttore (diodi e transistor) di avere una tensione o corrente fortemente dipendente dalla temperatura. La più semplice realizzazione di un sensore a semiconduttore è ottenuta mediante l'impiego di un diodo. Mediante vari passaggi nelle equazioni corrente-tensione che caratterizzano un diodo è possibile ottenere un'equazione tensione-temperatura.

In un diodo polarizzato direttamente la relazione corrente-tensione è:

$$I = I_s e^{\frac{V}{nV_T}} = I_s e^{\frac{qV}{nkT}} \quad (2.3)$$

Dove:

$I_s$  è la corrente inversa di saturazione (dell'ordine dei  $10^{-6} \div 10^{-9}$  A) dipendente dalla temperatura;  
 $n$  è una costante empirica dipendente dal tipo di semiconduttore usato ( $n=2$  per Si,  $n=1$  per Ge);  
 $V_T$  è la tensione termica (25mV alla temperatura di 27 °C);  
 $k$  è la costante di Boltzmann ( $1.38 \cdot 10^{-23}$  J/K);  
 $q$  è la carica dell'elettrone ( $1.6 \cdot 10^{-19}$  C);  
 $T$  è la temperatura in K.

Ricordando che  $I_s$  dipende dalla temperatura possiamo ricavare l'equazione tensione-temperatura definita come:

$$V = \frac{E_g}{q} - \frac{kT}{q} * (\ln C - \ln I) \quad (2.4)$$

Dove:

$E_g$  è l'energy gap del materiale;  
 $C$  è la costante del materiale indipendente dalla temperatura.

Pertanto, quando la giunzione è alimentata a corrente costante, la tensione varia linearmente con la temperatura.

## Termocoppie

Si considerino due conduttori realizzati con metalli (o leghe metalliche) diversi, indicati genericamente con A e B e si supponga di realizzare un circuito chiuso utilizzando questi materiali, come indicato in Fig. 2-3. Si vengono così a formare due giunzioni  $J_1$  e  $J_2$ . Si verifica sperimentalmente che, se queste due giunzioni sono portate a due temperature diverse  $T_1$  e  $T_2$ , nel circuito chiuso si manifesta una circolazione di corrente. La corrente diventa nulla se si rendono uguali le due temperature. Questo fenomeno è chiamato effetto Seebeck in onore di Thomas Johann Seebeck, un fisico di provenienza estone, che nel 1821 ne scoprì il fenomeno.

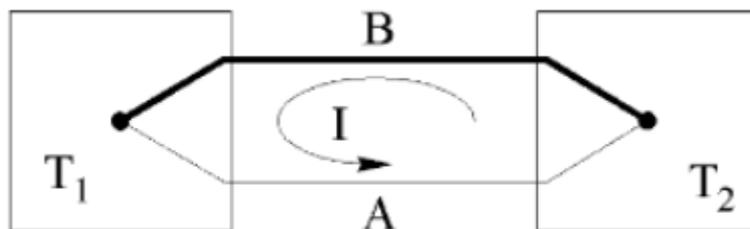


Fig. 2-3 Representation of Thermocouple's circuit.

La circolazione di tale corrente è dovuta al generarsi di una forza elettromotrice  $E_{AB}$ , che è funzione sia della natura dei materiali A e B, sia della differenza di temperatura  $\Delta T = T_1 - T_2$ .

## Sensore di temperatura integrato

Partendo dalle configurazioni, descritte precedentemente, e sfruttando le proprietà dei semiconduttori vengono realizzati sensori di temperatura più complessi “IC sensor” (Integrated Circuit sensor) che contengono specifici circuiti integrati. La realizzazione in forma integrata rende possibili numerosi vantaggi, rispetto agli altri sensori esaminati: in particolare, il costruttore può prevedere sul chip tutti i circuiti di condizionamento richiesti, tra i quali la preliminare amplificazione del segnale, rendendo questi amplificatori molto semplici da usare. Come si è visto in precedenza si possono realizzare trasduttori estremamente lineari con uscita in corrente o tensione in funzione alla temperatura a cui sono posti.

### 2.2.1 Descrizione generale sensore PCT2075D

Si prenderà, adesso, in esame il sensore di temperatura utilizzato in questa esperienza che è il PCT2075D rappresentato in Fig. 2-4.

Questo sensore fa parte della famiglia dei sensori di temperatura integrati visti precedentemente.

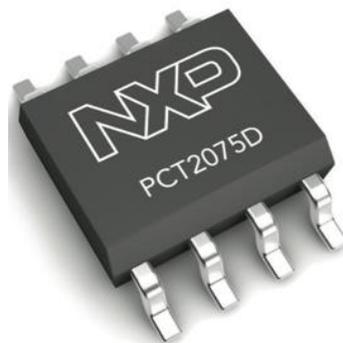


Fig. 2- 4 NXP PCT2075D IC-sensor.

Il PCT2075 è un convertitore di temperatura-digitale con un'accuratezza di  $\pm 1^{\circ}\text{C}$  e un input range da  $-25^{\circ}\text{C}$  a  $+100^{\circ}\text{C}$ . Esso utilizza un sensore di temperatura e un convertitore A-D Sigma-Delta e presenta un dispositivo di rilevamento di sovratemperatura che sostituisce il sensore termico LM75. Il dispositivo contiene un numero elevato di registri: Configuration register (Conf) per memorizzare le impostazioni del dispositivo come la modalità di funzionamento del dispositivo e la modalità di funzionamento del sistema operativo; temperature register (Temp) per memorizzare il valore di temperatura digitale, set-point register (Tos e Thyst) per memorizzare i limiti di arresto sovratemperatura e isteresi programmabili e il tempo di campionamento del sensore di temperatura programmabile Tidle, che possono essere comunicati da un controller tramite la comunicazione seriale a due fili chiamata I<sup>2</sup>C-bus.

Il PCT2075 include anche un'uscita open-drain (OS), che si attiva quando la temperatura supera i limiti programmati. L'uscita OS opera in due modalità selezionabili:

- modo comparatore,
- modalità di interruzione.

La PCT2075 può essere configurato per diverse condizioni di esercizio. Esso può essere impostato in modalità normale per controllare periodicamente la temperatura ambiente o in modalità di arresto dell'impianto per minimizzare il consumo di energia.

Il registro di temperatura memorizza sempre 11-bit dando una risoluzione termica di  $0,125^{\circ}\text{C}$ . Questa elevata risoluzione di temperatura è particolarmente utile nelle applicazioni che richiedono un'alta precisione di misurazione.

Il PCT2075 vanta anche una velocità I<sup>2</sup>C-bus più veloce, fino a 1 MHz, che consente una risposta più rapida sul bus con molti slave I<sup>2</sup>C. Tipici dispositivi standard del settore industriale consentono solo otto dispositivi sullo stesso segmento di I<sup>2</sup>C-bus senza schemi di multiplexing elaborati, ma il PCT2075 permette fino a 27 dispositivi sullo stesso segmento I<sup>2</sup>C-bus usando solo tre pin del dispositivo. Questa caratteristica è particolarmente utile quando si determina gradienti di temperatura su una superficie.

## Caratteristiche

- pin-per-pin sostituto di serie LM75, ma permette fino a 27 dispositivi sul bus,
- range di alimentazione da 2,7 V a 5,5 V,
- range di temperatura da -55°C a +125°C,
- range di frequenza da 20kHz a 1Mhz,
- 11-bit ADC, che offre una risoluzione di temperatura di 0.125 ° C,
- precisione della temperatura di:
  - ± 1 ° C (max.) Da -25 ° C a +100 ° C,
  - ± 2 ° C (max.) Da -55 ° C a +125 ° C;
- punti di soglia di temperatura e di isteresi programmabili durante il funzionamento,
- corrente di alimentazione di <1,0 µA in modalità di arresto per il risparmio energetico,
- funzionamento autonomo come termostato all'accensione.

## Possibili applicazioni

- Sistema di gestione termica;
- Personal Computer;
- Apparecchiature elettroniche;
- Controller industriali.

In Fig. 2-4 è rappresentato un diagramma a blocchi del PCT2075D mentre in Fig. 2-5 la disposizione dei pin del chip con relativa tabella descrittiva dei vari pin (Tab. 2-1).

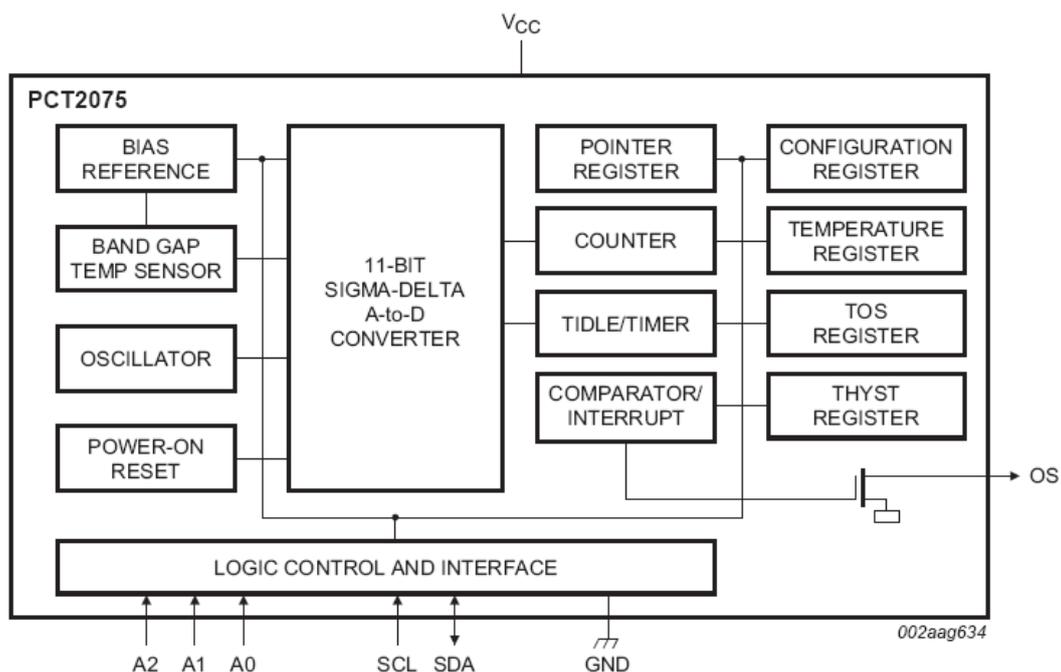


Fig. 2- 5 Block diagram of PCT2075.

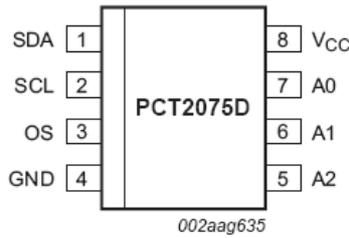


Fig. 2- 6 Pin configuration of PCT2075D.

Symbol	Pin	Description
SDA	1	Digital I/O. I <sup>2</sup> C-bus serial bidirectional data line; open drain.
SCL	2	Digital input. I <sup>2</sup> C-bus serial clock input.
OS	3	Overtemp Shoutdown output; open-drain.
GND	4	Ground. To be connected to the system ground.
A2	5	Digital input. User-defined address bit 2.
A1	6	Digital input. User-defined address bit 1.
A0	7	Digital input. User-defined address bit 0.
V <sub>cc</sub>	8	Power supply.

Tab 2-1 Description pin of PCT2075D.

### 2.2.2 Interfaccia seriale I<sup>2</sup>C-bus

La comunicazione fra due o più dispositivi elettronici si realizza attraverso un insieme di linee, chiamate bus. Affinché gli elementi interagiscano correttamente, è necessario stabilire delle specifiche di comunicazione. L'insieme delle regole di trasmissione prende il nome di protocollo. La Philips ha brevettato un protocollo di comunicazione, chiamato I<sup>2</sup>C, il quale consente il colloquio e lo scambio di informazioni fra più dispositivi collegati fra loro attraverso un bus composto da due linee come in Fig.2-6.

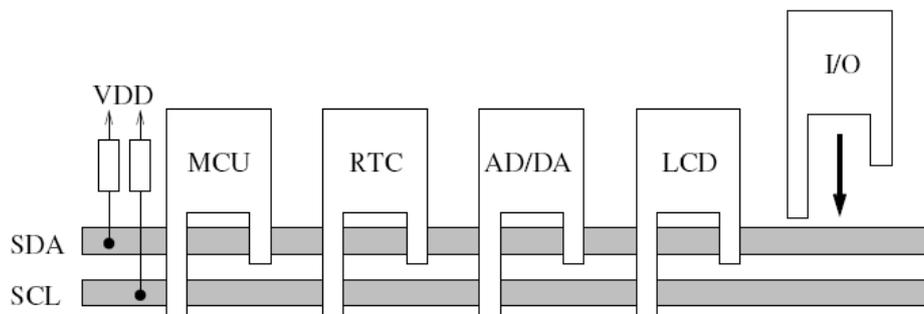


Fig. 2- 6 Basic configuration of the IIC bus.

Il protocollo I<sup>2</sup>C prevede l'utilizzo di un bus formato da due linee bidirezionali. Le due linee, chiamate "scl" e "sda" rispettivamente, trasportano la tempistica di sincronizzazione (chiamata anche "clock") e i dati. I segnali che transitano sulla linea hanno valore '1' o '0' e le tensioni che li rappresentano sono quelle d'alimentazione e di massa rispettivamente. Le due linee non sono lasciate ad un valore indefinito, ma vengono collegate all'alimentazione attraverso una resistenza di pull-up. In questo modo le due linee permangono ad un valore "alto-debole", facilmente modificabile da un dispositivo.

Ogni dispositivo collegato a queste linee è dotato di un indirizzo univoco, di 7 o 10 bit e può agire sia da master che da slave, secondo le funzioni previste al suo interno. Il master si occupa di iniziare la trasmissione e di generare la tempistica del trasferimento, mentre lo slave è quello che riceve una richiesta. Entrambe le due categorie appena descritte possono assumere il ruolo di trasmittente o ricevente.

La trasmissione inizia con la generazione di un segnale di start, immesso sulla linea dal master, dopo un controllo sull'occupazione del bus. La condizione di start consiste nel lasciare la linea "scl" allo stato "alto", mentre la linea "sda", subisce una transizione dallo stato '1' allo stato '0'.

Le modalità di trasferimento dati possono essere riassunte come segue:

D) A trasmette dati a B

1. A (master) spedisce l'indirizzo di B (slave) sul bus,
2. A (master-transmitter) trasmette i dati a B (slave-receiver),
3. A termina il trasferimento.

II) A vuole ricevere dati da B

1. A (master) spedisce l'indirizzo di B (slave) sul bus,
2. B (slave-transmitter) spedisce i dati ad A (master-receiver),
3. A termina il trasferimento.

Dopo la generazione del segnale di start inizia la trasmissione dei dati vera e propria.

Il primo byte trasmesso è quello composto dall'indirizzo dello slave con l'aggiunta di un ulteriore valore che indica al ricevente quale è l'operazione a lui richiesta. In base allo standard previsto dal protocollo questo bit può assumere due valori con i seguenti significati:

- '0': ricezione dati;
- '1': trasmissione dati;

L'ordine di trasmissione dei bit è quello dal più significativo al meno significativo. Le successive comunicazioni seguono sempre quest'ultimo criterio. Dopo la trasmissione d'ogni byte chi trasmette ha l'obbligo di lasciare la linea "sda" allo stato "alto", in modo da permettere a chi riceve dei dati, di darne conferma tramite il meccanismo dell'acknowledgement: esso consiste nell'abbassare la linea "sda" in corrispondenza del nono impulso presente sulla linea "scl".

Se l'ack non venisse generato, chi ha iniziato la trasmissione può interrompere la comunicazione, utilizzando la condizione di stop. Un dispositivo può non generare il segnale di acknowledgement ad esempio, quando è inabilitato a ricevere, perché sta eseguendo delle funzioni in real-time, oppure quando non può più immagazzinare altri dati. Il segnale di stop è sempre generato dal master, come pure quello dello start. Esso consiste nel far variare la linea sda dallo stato "basso" a quello "alto" in corrispondenza del periodo "alto" della linea "scl".

Di seguito si descriverà il protocollo da seguire per scrivere e leggere nei registri del PTC2075 tramite I<sup>2</sup>C.

1. Prima di una comunicazione l'I<sup>2</sup>C-bus deve essere libero e non occupato. Questo significa che le linee SCL e SDA devono essere entrambe liberate dai dispositivi che occupano il bus e devono avere il valore HIGH grazie al circuito di pull-up.
2. L'host deve fornire impulsi di clock alla linea SCL necessari per la comunicazione. I dati sono trasferiti in una sequenza di nove impulsi di clock di SCL per ogni byte di dati a 8 bit seguito da un bit di stato di riconoscimento.
3. Durante il trasferimento dati, tranne per i segnali di START e STOP, il segnale SDA deve essere stabile mentre il segnale SCL è HIGH. Ciò significa che il segnale SDA può essere modificata solo durante la durata LOW della linea SCL.
4. S: segnale di START, avviato dall' host per avviare una comunicazione, la SDA passa da uno stato HIGH a uno stato LOW mentre il SCL è HIGH.

5. RS: segnale di RE-START, uguale al segnale di START, per avviare un comando di lettura che segue un comando di scrittura.
6. P: segnale di STOP, generato dall'host per fermare una comunicazione, la SDA passa da LOW a HIGH mentre il SCL è HIGH. In seguito il bus si libera.
7. W: scrivere un bit quando il write/read bit = LOW in un comando di scrittura.
8. R: leggere un bit quando il write/read bit = HIGH in comando di lettura.
9. A: dispositivo riconosce il bit restituito dal dispositivo. Il valore del bit è LOW se il dispositivo funziona correttamente altrimenti sarà HIGH. L'host deve rilasciare la linea SDA durante questo periodo per dare la linea SDA al dispositivo di controllo.
10. A': il master riconosce il bit, non restituito dal dispositivo, ma impostato dal master o host nella lettura dei dati 2 byte. Durante questo periodo di clock, l'host deve impostare la linea SDA per LOW per notificare il dispositivo che il primo byte è stato letto per il dispositivo fornire il secondo byte sul bus.
11. NA: non riconosce bit. Durante questo periodo di clock, sia il rilascio del dispositivo e l'host la linea SDA alla fine di un trasferimento di dati, l'host viene quindi attivato per generare il segnale di arresto.
12. In un protocollo di scrittura, i dati vengono inviati dall'host al dispositivo e l'host controlla la linea SDA, tranne durante il periodo di clock quando il dispositivo invia al dispositivo segnale di conferma al bus.
13. In un protocollo di lettura, i dati vengono inviati al bus dal dispositivo e l'host deve rilasciare la linea SDA durante il tempo che il dispositivo sta fornendo dati sul bus e controllo la linea SDA, tranne durante il periodo di orologio quando il master invia il maestro segnale di conferma al bus.

La Fig. 2-7 mostra graficamente un esempio dei vari passaggi, appena descritti, da seguire, per leggere la temperatura dal registro Temp del sensore utilizzato.

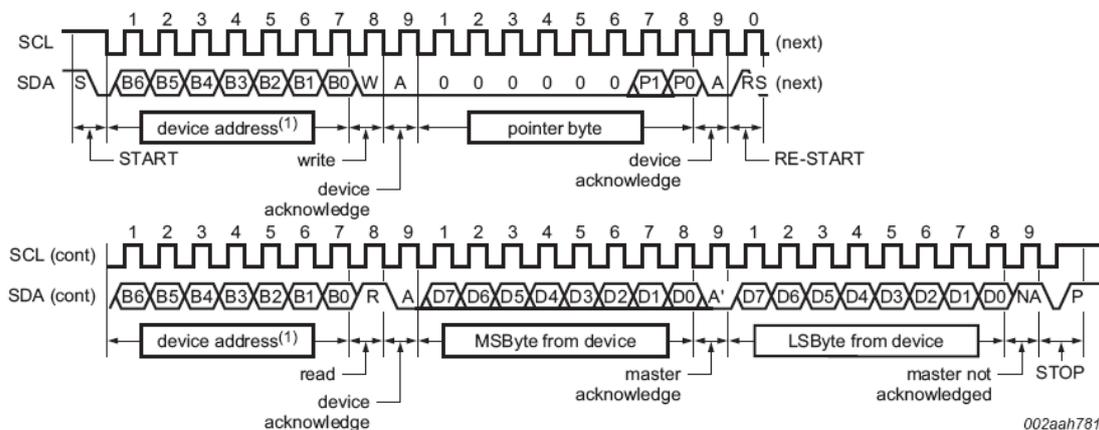


Fig. 2-7 Read Temp, Tos or Thyst register including ppointer byte (2-byte data).

Poiché il sensore di temperatura misura la propria temperatura che viene trasferita al suo corpo, la temperatura del corpo del dispositivo deve essere stabilizzata per poter fornire letture più vere possibili. La precisione della misura dipende maggiormente dalla temperatura ambiente, che è influenzata da diversi fattori: disposizione del sensore nel circuito stampato su cui è montato; il flusso d'aria a contatto con il corpo del dispositivo e la differenza di temperatura tra scheda e ambiente (se la temperatura dell'aria ambiente e la temperatura della scheda a circuito stampato sono molto diversi la misurazione non può essere stabile a causa dei diversi percorsi termici).

## 2.3 Ricetrasmittitore ad ultrasuoni

### 2.3.1 Generalità

Gli ultrasuoni sono delle onde meccaniche sonore. A differenza dei fenomeni acustici propriamente detti, le frequenze che caratterizzano gli ultrasuoni sono superiori a quelle mediamente udibili da un orecchio umano. La frequenza convenzionalmente utilizzata per discriminare onde soniche da onde ultrasoniche è fissata in 20-kHz.

Come ogni altro tipo di fenomeno ondulatorio gli ultrasuoni sono soggetti a fenomeni di riflessione, rifrazione e diffrazione e possono essere definiti mediante parametri quali la frequenza, la lunghezza d'onda, la velocità di propagazione, l'intensità (misurata in decibel), l'attenuazione (dovuta all'impedenza acustica del mezzo attraversato).

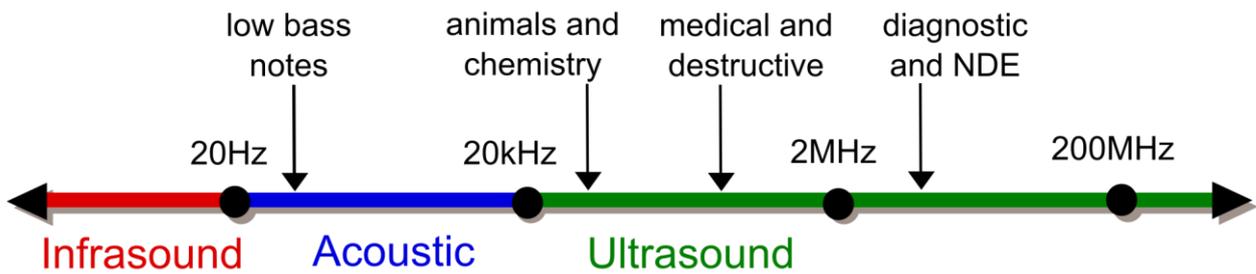


Fig. 2- 8 Spectral range of ultrasound.

Gli ultrasuoni trovano utilizzo per lo più in campo medico ed industriale essendo ampiamente utilizzati nelle ecografie, nei controlli non distruttivi e in molti apparecchi utilizzati per la pulizia superficiale di oggetti di piccole dimensioni.

Anche il sonar impiega intervalli di frequenze che non di rado sconfinano nella gamma degli ultrasuoni.

Gli ultrasuoni vengono generati per mezzo di materiali con particolari caratteristiche meccanico-elettriche, i materiali piezoelettrici. Questi particolari materiali come ad esempio il quarzo o titanato di bario hanno la caratteristica di generare una differenza di potenziale se compressi o stirati in senso trasversale; viceversa, se applicata una differenza di potenziale ai loro estremi, questi si comprimono o dilatano in senso trasversale. Proprio quest'ultima caratteristica viene sfruttata per generare queste onde meccaniche sopra il campo dell'udibilità (ultrasuoni). In base al materiale scelto avremo quindi diverse frequenze di ultrasuoni, diverse propagazioni nei materiali e quindi diverse caratteristiche di potenza delle macchine generatrici.

La misurazione della distanza ad ultrasuoni si basa sulla proprietà della velocità del suono. Il sistema trasmette più onde sonore che viaggiano nell'aria. Queste onde sonore si riflettono su tutti gli oggetti incontrano nella propria strada e ritornano come un'eco alla posizione da cui hanno avuto origine. Il sistema rileva queste onde sonore riflesse (cioè, echi). Il tempo tra la trasmissione dell'onda e il rilevamento dell'eco è misurato come mostrato in Fig. 2-9.

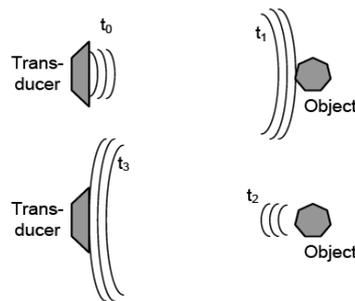


Fig. 2- 9 Ultrasonic distance measurement.

Per calcolare la distanza percorsa dall'onda si moltiplica la velocità dell'onda per il tempo impiegato da quest'ultima da quando viene sparata a quando viene rilevata nel segnale di ritorno, poi si divide per due per ottenere la distanza dal sensore trasmettitore all'ostacolo (si veda l'equazione 2.5):

$$s = \frac{v * t}{2} \quad (2.5)$$

Dove  $s$  è la distanza tra il trasduttore e l'oggetto rilevato,  $v$  è la velocità del suono, e  $t$  è il tempo misurato tra la trasmissione delle onde sonore e rilevamento dell'eco.

La velocità del suono dipende da molti fattori tra cui la temperatura, la densità e la composizione dell'aria. Generalmente la velocità del suono ad una temperatura di 23°C dell'aria è di circa 345 m/s.

In molte applicazioni il tipo di trasduttore ad ultrasuoni più utilizzato è racchiuso (Fig. 2-11), cioè è chiuso in un guscio protettivo ermetico per evitare qualsiasi contaminazione che ne ridurrebbe la sua affidabilità. Inoltre in queste applicazioni sono ricetrasmittitori, cioè possono convertire l'energia elettrica in energia sonora (onda) e viceversa. Pertanto possono creare e rilevare ultrasuoni.

Questo tipo di trasduttori (integrati) sono più affidabili e robusti con un compromesso in termini di prestazioni, infatti tutto il sistema di trasmissione e ricezione è unito in un unico contenitore, cosa che non accade per i trasduttori equivalenti aperti (Fig. 2-10), e quindi ci possono essere dei disturbi che il ricevitore acquisisce dal trasmettitore.



Fig. 2-10 transceiver and receiver.



Fig. 2-11 Integration ultrasonic transceiver and receiver.

Alcune importanti proprietà dei trasduttori ad ultrasuoni sono:

- Frequenza di risonanza (tipicamente di 40 kHz),
- Massima tensione applicabile (tipicamente 100 V<sub>pp</sub>),
- Tempo di trasmissione (in genere di 1 ms),
- Impedenza nominale di risonanza,
- Capacità nominal,
- Angolo del fascio.

Come in ogni sensore è previsto un range massimo di rilevamento (Fig 2-12). Il campo di rilevamento desiderato per un sistema varia a seconda dell'applicazione e la portata massima dipende principalmente dai seguenti fattori:

- La quantità di energia utilizzata per guidare il trasduttore,
- Il rapporto del guadagno del segnale e segnale-rumore (SNR) del circuito utilizzato per misurare il segnale di ritorno,
- Le proprietà del trasduttore stesso.

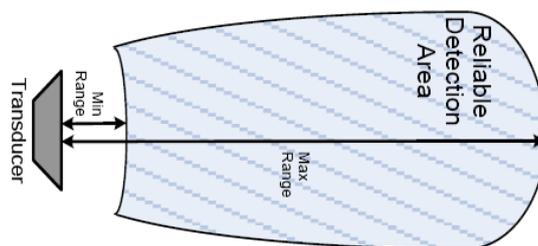


Fig. 2-12 Object detection range.

Un sistema di misurazione della distanza ad ultrasuoni di solito ha anche un campo minimo nel quale gli oggetti all'interno di questo range non possono essere rilevati in modo affidabile (Fig. 2-12).

L'intervallo minimo è un aspetto indesiderato del sistema causata dall'uso di trasduttori ricetrasmittitore. Un trasduttore ricetrasmittitore trasmette il suono e lo riceve. Pertanto, c'è un periodo dopo la trasmissione durante il quale il sistema deve attendere perché il trasduttore smetta di emettere. Se il sistema iniziasse immediatamente ad acquisire, prima che il trasmettitore si spenga, misurerebbe sempre una distanza pari a zero perché rileverebbe il segnale inviato direttamente dal trasmettitore. Pertanto, il sistema, prima di rilevare, deve attendere che il trasduttore abbia finito di trasmettere. Il tempo di attesa è direttamente proporzionale alla distanza minima di rilevamento.

Il circuito interno di un sensore ad ultrasuoni tipicamente è composto da:

- Generatore d'onde a 40 kHz,
- Segnale digitale de-multiplexer,
- Segnale analogico multiplexer,
- Programmable Gain Amplifier (PGA),
- Filtro passa basso a 40 kHz,
- Riferimento,
- Comparatore,
- Timer a 16 bit,
- CPU: gestisce i componenti hardware del sistema e agisce l'esecuzione della fase di misura del sistema,
- Interfaccia di comunicazione: l'interfaccia di comunicazione maggiormente utilizzata è I<sup>2</sup>C.

### 2.3.2 Descrizione generale SRF01

Il sensore ad ultrasuoni che si è preso in esame è il sensore SRF01.

L'SRF01 è un sensore ad ultrasuoni a singolo trasduttore, con un circuito miniaturizzato delle stesse dimensioni del trasduttore. La connessione dati è costituita da un solo pin. L'interfaccia seriale è a livello TTL con formato UART e può essere connesso direttamente al microcontrollore. Si possono connettere insieme fino a 16 SRF01 su un singolo bus seriale. La misurazione può avvenire in cm o in pollici.

Il sensore SRF01 ha anche la modalità risparmio (sleep mode), quando è posto in modalità di sospensione, tutte le attività sul processore del SRF01 cessano. Il modulo si spegne e il consumo energetico viene ridotto al minimo. È però sempre presente un passaggio di corrente dovuta alla dispersione del processore tipicamente pari a 55µA. Per riattivare il SRF01 dalla modalità sleep si deve inviare il comando 0xFF che permette una transizione sulla linea Rx che risveglia tutti i SRF01 del bus seriale. Dopo il risveglio del

SRF01, è necessario attendere 2 ms perchè gli orologi del processore si stabilizzino prima di inviare i comandi.

Nella Tab. 2-2 sono descritte le principali caratteristiche.

Caratteristiche tecniche	
Tensione operativa	3.3 - 5.5 V
Corrente operativa	11 mA (25 mA durante il rilevamento)
Tipica frequenza	40 kHz
Portata	18 cm – 6 m (0 cm – 600 cm calibrazione richiesta)
Interfaccia	Segnale seriale TTL
Portata bus	16 sensori
Unità di misura	Centimetri o pollici
Peso	2.7 grammi

Tab. 2-2 Description features.

## 2.4 Modulo GSM-GPS

### 2.4.1 Generalità

#### Sistema GSM

In telecomunicazioni il GSM, sigla di Global System for Mobile Communications (in origine «Groupe spécial mobile»), è lo standard 2G (2ª generazione) di telefonia mobile cellulare e attualmente il più diffuso del mondo. In particolare il GSM è uno standard aperto sviluppato dal CEPT e finalizzato dall'ETSI e mantenuto dal consorzio 3GPP (di cui l'ETSI fa parte).

L'introduzione del GSM ha rappresentato una vera e propria rivoluzione nell'ambito dei sistemi di telefonia cellulare. Fondamentalmente i numerosi vantaggi rispetto ai precedenti sistemi cellulari sono stati:

- interoperabilità tra reti diverse che fanno capo ad un unico standard internazionale,
- comunicazione di tipo digitale.

L'introduzione di una trasmissione di tipo digitale a sua volta porta con sé tre grosse e importanti conseguenze:

- maggiore velocità di trasmissione grazie alle tecniche di compressione dati proprie della codifica di sorgente (codifica LPC),
- nuovi e più ampi servizi (es. SMS) grazie all'aumento della velocità di trasmissione,
- funzioni di sicurezza in termini di cifratura della comunicazione.

La tecnologia alla base del GSM è significativamente diversa dalle precedenti (es. TACS) soprattutto per il fatto che sia il canale di identificazione che quello di conversazione supportano una comunicazione digitale. Per questo motivo il nuovo standard è stato lanciato sul mercato come sistema di telefonia mobile di seconda generazione (o più sinteticamente 2G).

Questa caratteristica di base significa che la possibilità di scambiare dati, oltre che conversazioni, è già stata implementata fin dall'inizio dello sviluppo del nuovo sistema.

La diffusione universale dello standard GSM ha fatto sì che la maggior parte degli operatori internazionali di telefonia mobile stipulassero fra di loro accordi per l'effettuazione del cosiddetto roaming (commutazione automatica fra diverse reti) grazie all'interoperabilità offerta dallo standard stesso.

Il maggior punto di forza del sistema GSM è stata la possibilità, da parte degli utenti, di accedere a tutta una serie di nuovi servizi a costi molto contenuti. Ad esempio lo scambio di messaggi testuali (SMS) è stato sviluppato per la prima volta in assoluto in ambito GSM.

Uno dei principali vantaggi per gli operatori è stato, invece, la possibilità di acquistare infrastrutture ed attrezzature a costi resi bassi dalla concorrenza fra i produttori sullo standard comune imposto. Per contro, una delle limitazioni più serie è derivata dal fatto che le reti GSM impiegano la tecnologia di accesso radio di tipo TDMA, considerata meno avanzata ed efficiente rispetto alla concorrente tecnologia CDMA. Tuttavia le prestazioni effettivamente riscontrate sul campo non sono molto diverse tra loro.

Pur essendo lo standard in costante evoluzione, i sistemi GSM hanno sempre mantenuto la piena compatibilità (o retrocompatibilità) con le precedenti versioni.

Successivi sviluppi del GSM sono stati infatti il GPRS (2.5G), che ha introdotto la commutazione di pacchetto e la possibilità di accesso ad Internet, e l'EDGE (2.75G) che ha incrementato ulteriormente la velocità di trasmissione del GPRS. Per utilizzare GPRS ed EDGE, che attualmente convivono con il GSM, la rete GSM necessitava di un upgrade software di alcuni apparati e di poche modifiche hardware, cosa che ha reso questi nuovi standard particolarmente appetibili per le compagnie di telefonia mobile, le quali hanno potuto implementare servizi di accesso ad Internet affrontando costi nettamente inferiori rispetto all'UMTS (3G) che ha richiesto invece sostanziali modifiche agli apparati della rete GSM.

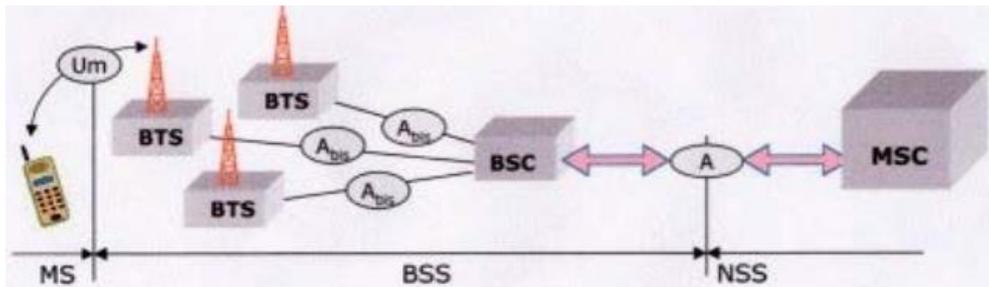


Fig. 2-13 The structure of a GSM network.

Il sistema GSM è composto da tre sottosistemi principali interconnessi tra loro come rappresentato nella figura soprastante di cui si dà un breve cenno nelle righe successive:

- Base station subsystem (BSS) + mobile station (MS),
- Network and switching subsystem (NSS),
- Operation & maintenance center (OMC).

Base station sub system (BSS):

- E' il sottosistema che gestisce l'accesso radio: comprende le unità (stazioni base) che consentono di fornire la copertura radio di un'area costituita da una o più celle,
- Una BSS comprende un certo numero di BSC che si connettono ad un singolo MSC,
- La stazione base è composta di due unità: una Base Transceiver Station(BTS) e una Base Station Controller(BSC),
- La mobile station fa funzionalmente parte del BSS.

Base transceiver station (BTS):

- Una BTS è composta da antenna (il cui numero dipende se si implementa diversità spaziale o no), trasmettitore, ricevitore e poco più,
- Comunica con BSC attraverso l'Interfaccia Abis(poco standardizzata). Le connessioni tra BSC e BTS possono essere su cavo o collegamento radio, dipende da dove sono localizzate le BTS,
- Tra BTS e BSC ci sono canali a 13 (banda occupata da comunicazioni vocali GSM) o a 64 kbit/s banda ISDN).

Base station controller (BSC)

- BSC sono l'intelligenza del sottosistema radio. Ciascuna BSC controlla fino ad un centinaio di BTS,
- Tra le funzioni di una BSC:
  - Assegnare i canali ad un utente che ne faccia richiesta,
  - Gestire handover tra BTS che controlla,
  - Gestire procedure di paging delle MS,
  - Gestire la comunicazione con MSC attraverso interfaccia A (pienamente standardizzata).

L'operation and maintenance center (OMC) è il terzo sottosistema che compone l'architettura GSM e ha due funzioni principali:

- Manutenimento e controllo di ogni elemento della rete,
- Gestione delle procedure di charging&billing.

## Sistema GPS

Il sistema GPS invece, è il sistema di posizionamento globale (in inglese: Global Positioning System, in sigla: GPS, a sua volta abbreviazione di NAVSTAR GPS, acronimo di NAVigation Satellite Timing And Ranging Global Positioning System o di NAVigation Signal Timing And Ranging Global Position System) è un sistema di posizionamento e navigazione satellitare civile che, attraverso una rete dedicata di satelliti artificiali in orbita, fornisce ad un terminale mobile o ricevitore GPS informazioni sulle sue coordinate geografiche ed orario, in ogni condizione meteorologica, ovunque sulla Terra o nelle sue immediate vicinanze ove vi sia un contatto privo di ostacoli con almeno quattro satelliti del sistema. La localizzazione avviene tramite la trasmissione di un segnale radio da parte di ciascun satellite e l'elaborazione dei segnali ricevuti da parte del ricevitore.

Il sistema GPS è gestito dal governo degli Stati Uniti d'America ed è liberamente accessibile da chiunque sia dotato di un ricevitore GPS. Il suo grado attuale di accuratezza è dell'ordine dei metri, in dipendenza dalle condizioni meteorologiche, dalla disponibilità e dalla posizione dei satelliti rispetto al ricevitore, dalla qualità e dal tipo di ricevitore, dagli effetti di radiopropagazione del segnale radio in ionosfera e troposfera (es. riflessione) e dagli effetti della relatività.

Il sistema di posizionamento si compone di tre segmenti: il segmento spaziale (space segment), il segmento di controllo (control segment) ed il segmento utente (user segment). L'Aeronautica militare degli Stati Uniti sviluppa, gestisce ed opera il segmento spaziale ed il segmento di controllo.

Il segmento spaziale comprende da 24 a 32 satelliti. Il segmento di controllo si compone di una stazione di controllo principale, una stazione di controllo alternativa, varie antenne dedicate e condivise e stazioni di monitoraggio. Il segmento utente infine è composto dai ricevitori GPS.

Attualmente sono in orbita 31 satelliti attivi nella costellazione GPS (più alcuni satelliti dismessi, alcuni dei quali riattivabili in caso di necessità). I satelliti supplementari migliorano la precisione del sistema permettendo misurazioni ridondanti. Al crescere del numero di satelliti, la costellazione è stata modificata secondo uno schema non uniforme che si è dimostrato maggiormente affidabile in caso di guasti contemporanei di più satelliti.

### 2.4.2 Descrizione generale SIM 908

Nel caso in esame si è utilizzato il modulo cellulare GSM/GPRS/GPS SIM908 (Fig. 2-14).



Fig. 2-14 Module SIM908.

Compatta scheda contenente il modulo GSM/GPRS/GPS SIM908 della SIMCom, due connettori d'antenna UFL, una porta SIM card e uno strip maschio 2x10 poli a passo 2mm. Il connettore trasporta l'alimentazione ( $V_{cc}$ , contatti 17 e 19) oltre alla linea di controllo dell'accensione (ON/OFF) le linee di comunicazione seriale da e verso il modulo GSM (TXD ed RXD), ma anche la massa (GND, contatti 18 e 20) ed il Ring Indicator. La linea ON/OFF serve al microcontrollore per comandare l'accensione e lo spegnimento del modulo GSM1, che comunque rimane sempre sotto la tensione fornita dalla linea  $V_{cc}$  ai piedini 17, 19 (3,2÷4,8Vdc); PWR è provvista di una resistenza di pull-up interna ed è attiva a zero logico, quindi il microcontrollore, per accendere il modulo cellulare, deve porre a livello logico alto il terminale ON/OFF (contatto 1 del connettore). Il controllo del reset avviene all'accensione, quindi non è prevista alcuna linea di reset. In

alternativa è possibile eseguire questa operazione inviando mediante seriale il comando AT dedicato. Sulla scheda è previsto un jumper (J2) che, se chiuso, permette di monitorare il pin STATUS che fornisce le indicazioni relative allo stato del modulo. Il modulo SIM908 dispone di due seriali distinte per la sezione cellulare e quella GPS: per la prima è disponibile un UART con le linee TXD, RXD, DTR, che vanno all'esterno mediante i contatti 12, 14, 10 del connettore; per il GPS, invece, la seriale fa capo a GPSTXD (contatto 4) e GPSRXD (contatto 6). La seriale del cellulare è in realtà destinata al controllo globale del SIM908, quindi da essa è possibile anche configurare il ricevitore GPS per ottenere i dati sulla localizzazione, sul numero di satelliti agganciati e farli acquisire al microcontrollore della scheda di controllo. Dalla seriale GPSTXD/GPSRXD escono continuamente le stringhe in formato NMEA standard del sistema GPS, ma in ogni caso è possibile ottenere questi dati inviando specifici comandi AT sulle linee TXD/RXD. Oltre alle linee seriali, dal modulo è possibile prelevare il segnale di RI, che esce dal contatto 8 del connettore, utilizzabile per informare il microcontrollore della scheda di controllo dell'arrivo delle telefonate. L'audio, facente capo a due contatti per il microfono (ad ingresso differenziale) ed altrettanti per l'altoparlante, transita dai contatti 15, 13, 9, 11, che corrispondono rispettivamente a MIC1P e MIC1N (positivo e negativo del microfono) ed SPK1N e SPK1P (rispettivamente negativo e positivo dell'altoparlante). L'antenna del modulo GSM si collega direttamente al connettore previsto allo scopo sul modulo; il SIM908 ha un secondo connettore per connettere l'antenna del ricevitore GPS. È possibile collegare antenne GPS sia passive che attive; in questo caso è necessario chiudere il jumper J1 per consentire al modulo SIM908 di alimentare l'antenna. Il pin 3 del connettore fa capo alla linea LED, con la quale il microcontrollore si informa sull'eventuale presenza della rete GSM e sullo stato di connessione del modulo (assenza segnale di rete, presenza rete ecc.).

Nelle Fig. 2.15, 2-16 viene posto una rappresentazione dell'hardware del modulo con una Tab. 2-3 che ne descrive i vari pin.

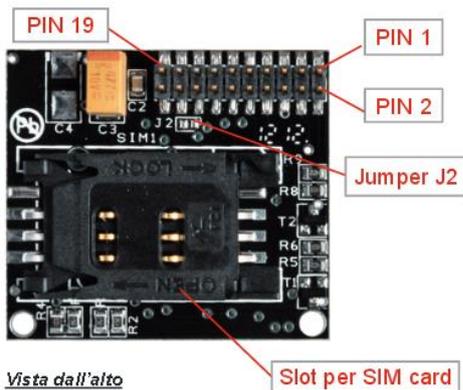


Fig. 2-15 Top view of SIM908.

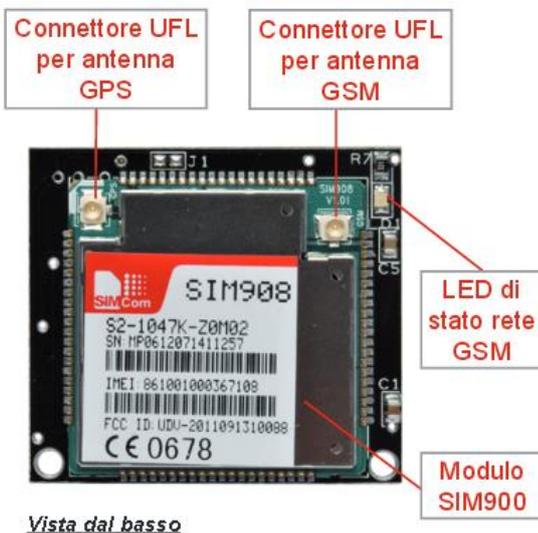


Fig. 2-16 Down view of SIM908.

Pinout modulo	
PIN	Descrizione
1	ON/OFF
2	TEMP BATT
3	LED
4	GPS TXD
5	VRTC
6	GPS RXD
7	VCHG
8	RI
9	SPK1N
10	DTR
11	SPK1P
12	TXD
13	MIC1N
14	RXD
15	MIC1P
16	STATUS
17	VCC
18	GND
19	VCC
20	GND

Dimensioni modulo:  
41 x 35 x 10mm.

Tab. 2-3 Description pin.

## 2.5 RTC

### 2.5.1 Generalità

Un real-time clock (RTC), orologio in tempo reale, è un dispositivo con funzione di orologio, solitamente costituito da un processore a circuito integrato specializzato per questa funzione, il quale conteggia il tempo reale (anno, mese, giorno, ore, minuti e secondi) anche quando l'utilizzatore viene spento. Viene usato in molti tipi di computer ed è presente in tutti i moderni PC. L'RTC è presente anche in molti sistemi embedded nonché viene utilizzato anche in circuiti elettronici dove è necessario avere un preciso riferimento temporale. Per poter mantenere il conteggio del tempo anche a circuito non alimentato, i real-time clock hanno un oscillatore al quarzo a loro dedicato e sono alimentati da una speciale batteria autonoma rispetto all'alimentazione principale; in alcuni tipi anche il quarzo è installato all'interno del package. Al contrario, i clock che non sono real-time non funzionano quando il dispositivo è spento.

Gli RTC non devono essere confusi con il real-time o il clock della CPU.

### 2.5.2 Descrizione generale RTC DS2417

Nel presente caso si è utilizzato un RTC DS2417 1-Wire Time Chip with Interrupt.

Il DS2417 contiene una ROM al laser unico e un orologio/calendario, in tempo reale, implementato con un contatore binario. È richiesto solo un pin per la comunicazione con il dispositivo (Fig.2-17 e Tab. 2-4). Utilizzando una fonte di energia di backup, i dati non verranno persi una volta che verrà a mancare l'alimentazione principale della scheda (i dati sono non volatili) e consente il normale funzionamento dell'integrato (funzionamento stand-alone).

Le caratteristiche DS2417 possono essere utilizzati per aggiungere funzioni come il calendario, data e ora e diario di bordo di qualsiasi tipo di dispositivo elettronico o applicazione incorporato che utilizza un microcontrollore.

Il DS2417 è dotato di due componenti principali:

1. 64-bit lasered ROM,
2. real-time clock counter.

L'orologio in tempo reale utilizza un oscillatore on-chip che è collegato a un cristallo esterno di 32.768kHz. Per ogni operazione il master del bus deve prima fornire una delle quattro funzioni ROM:

1. Leggere ROM,
2. Partita ROM,
3. Ricerca ROM,
4. Salta ROM.

Dopo che un comando funzione ROM viene eseguito con successo, le funzioni dell'orologio in tempo reale diventano accessibili e il master può quindi fornire uno dei comandi di funzione del real-time clock counter. Tutti i dati vengono letti e scritti dal bit meno significativo.

### Caratteristiche

- Orologio in tempo reale (RTC) pienamente compatibile con interfaccia 1-Wire® MicroLAN,
- Utilizza la stessa rappresentazione binaria per l'ora e la data come il DS2404, ma con risoluzione di 1 secondo,
- precisione dell'orologio  $\pm 2$  minuti al mese a 25 °C,
- Uscita di interrupt programmabile per sveglia sistema,
- Comunica a 16.3 kbits al secondo,
- Unico, assicura l'assoluta tracciabilità perché non ci sono due parti uguali nello stesso modo,

- Garantisce la compatibilità con altri prodotti MicroLAN,
- Opera in un intervallo di tensione  $V_{DD}$  ampia: da 2,5 V a 5,5 V, da  $-40^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ ,
- Bassa potenza, 200 nA in genere con oscillatore in funzione,
- Compatto, a basso costo a 6 pin.

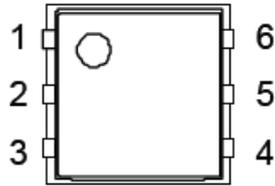


Fig. 2-17 Pin assignment.

PIN DESCRIPTION		
PIN	SYMBOL	DESCRIPTION
Pin 1	GND	Ground Pin
Pin 2	1-Wire	Data input/output Open drain.
Pin 3	$\overline{INT}$	Interrupt pin Open drain.
Pin 4	$V_{DD}$	Power input pin. 2.5V to 5.5V.
Pin5,6	X1, X2	Crystal pin. Connessioni per un cristallo di quarzo standard 32.768kHz, EPSON numero di parte C-002RX o C-004R (assicurarsi di chiedere 6pF capacità di carico).

Tab. 2-4 Pin description.

A solo scopo illustrativo viene di seguito posto la rappresentazione del diagramma a blocchi del chip RTC DS2417 1-Wire Time Chip with Interrupt (Fig 2-17).

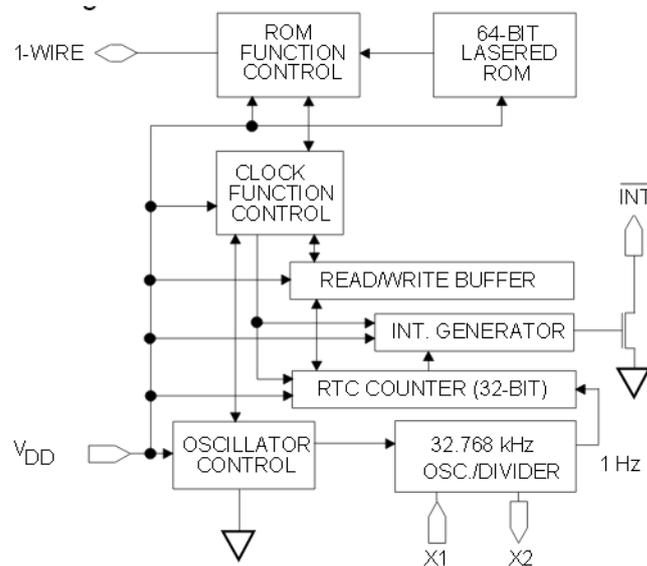


Fig. 2-17 Block diagram.

## Conclusioni

Questa tesi nasce dopo un periodo di stage nello studio E3 srl e ha il compito di poter realizzare quantomeno un prototipo di una scheda di controllo per una stufa a pellet. Questa scheda può essere inizialmente assemblata semplicemente con Arduino, che essendo Open Source è possibile trovare in rete con molti progetti e soluzioni inerenti al controllo di una stufa.

Dopo queste informazioni di base ricevute dalla trattazione, per quanto riguarda la funzione dei componenti principali si deve considerare la struttura nel complesso di un sistema e verificare che soddisfi le esigenze particolari e scegliere i componenti più adatti; in questo caso non si è badato a questa affermazione essendo i componenti facente parte a una scheda e a un progetto totalmente differenti. Infatti in molti casi la miglior soluzione dipende dall'applicazione e differenti approcci devono essere usati per differenti situazioni.

Anche la tecnologia gioca un significativo ruolo nelle decisioni. Quel che è il migliore nel 2010 non lo sarà nel 2020 e quindi il progettista deve essere molto sensibile e aggiornato ai nuovi mezzi, sistemi, componenti e tecniche per avere alla fine un prodotto valido e competitivo sia sotto l'aspetto economico sia tecnologico.

Nel nostro caso è stato usato il microcontrollore RX1115 perché l'applicazione precedente aveva bisogno di una porta USB con cui comunicare e in più è programmabile tramite il linguaggio C molto noto e diffuso nel campo informatico. Il microcontrollore RX1115 è caratterizzato da molti I/O i quali però possono risultare superflui per una piccola implementazione della scheda per la stufa a pellet.

Il sensore di temperatura PCT2075 invece, è un buon componente da usare perché ha una risoluzione molto elevata e per la nostra applicazione la misura della temperatura ambiente è molto importante.

Altro sensore di rilevante importanza è il sensore ad ultrasuoni. In questo caso si è deciso di utilizzare un sensore che ha incorporato in un unico corpo chiuso sia il trasmettitore sia il ricevitore per motivi di spazio e perché, essendo posto in un luogo dove è presente il pellet e di conseguenza polvere e truciolo, è più protetto e quindi ha una vita maggiore rispetto all'alternativa di sensore aperto; è possibile pensare di posizionare il trasduttore nella parte inferiore del coperchio del serbatoio del pellet. Si deve stare molto attenti che il sensore oltre ad avere un range massimo di misura ha anche un range minimo che è indicato nei datasheet e deve essere rispettato per avere una misurazione di livello per il pellet corretta.

Il modulo SIM908 è molto utile perché come si è visto, con la funzione GSM, permette la comunicazione tra la stufa e l'utente finale il quale è avvisato delle condizioni di temperatura ambiente e di livello del pellet all'interno del serbatoio. Il modulo SIM908 ha anche la funzione del GPS che potrebbe essere usata per ricavare dati, secondo le leggi e le regole della privacy e nell'anonimato dell'utente finale, e creare una mappatura termica del territorio verificando quante volte in una stagione invernale viene accesa la stufa ed è possibile fare una stima sul consumo del pellet.

Il componente RTC in qualsiasi scheda e applicazione che necessita di scandire il tempo è fondamentale. In questo caso con il chip DS2417 è possibile anche avere un calendario e redigere un diario di bordo così da avere uno storico facilmente consultabile.

Come affermato in precedenza il microcontrollore è stato programmato con il linguaggio C e così ho avuto modo di vedere come si programma un microcontrollore e di utilizzare le conoscenze ottenute nei vari corsi universitari di informatica svolti; infatti il compito dell'esperienza è stato anche quello di testare le varie funzioni e di trovare eventuali bug presenti nel codice: per esempio testando i vari sensori ho riscontrato degli errori, di conversione tra input e output, nella lettura della temperatura.



## **Ringraziamenti**

Desidero ringraziare il Prof. Diego Dainese per avermi seguito in questo periodo di redazione della tesi.

Ringrazio inoltre tutte le persone che lavorano nello studio E3 srl che mi hanno ben accolto, mi hanno dato l'opportunità di fare un'esperienza nel mondo del lavoro e la possibilità di redigere questa tesi utilizzando il loro tempo e strumenti; in particolare vorrei ringraziare il sig. Fabio Cisco il mio tutor aziendale con cui ho passato bei momenti.

Ringrazio tutta la mia famiglia che ha creduto in me e mi ha permesso di raggiungere questo mio obiettivo, traguardo e nuovo punto di partenza aiutandomi anche nei momenti difficili.

E ringrazio tutte quelle persone che mi hanno fatto crescere e diventare quello che sono.



# **Bibliografia**

## **MICROCONTROLLORE**

- Datasheet microcontrollore Renesas RX111 Group, Rev. 1.21, Dec 09, 2014.
- Datasheet microcontrollore Renesas RX1115.
- Gunther Gridling - Bettina Weiss, Introduction to Microcontrollers, Vienna University of Technology-Institute of Computer Engineering, Embedded Computing Systems Group, 26 February 2007.
- HAROLD S. STONE ,Microcomputer Interfacing, Addison-Wesley publishing company, University of Massachusetts, Amherst, 1982.
- Richard C. Jaeger – Travis N. Blalock, Microelectronic Circuit Design, McGraw-Hill, 4th-edition, 2011.

## **SITI WEB**

- [http://www.uniroma2.it/didattica/MSM-S/deposito/Esercitazione\\_1\\_-\\_Seriali.pdf](http://www.uniroma2.it/didattica/MSM-S/deposito/Esercitazione_1_-_Seriali.pdf)
- <http://www.vincenzov.net/tutorial/rs232/seriale.htm>
- [https://en.wikibooks.org/wiki/Microprocessor\\_Design/Interrupts](https://en.wikibooks.org/wiki/Microprocessor_Design/Interrupts)
- [https://it.wikibooks.org/wiki/Architetture\\_dei\\_processori/Set\\_di\\_istruzioni](https://it.wikibooks.org/wiki/Architetture_dei_processori/Set_di_istruzioni)
- <https://it.wikipedia.org/wiki/DRAM>
- [https://it.wikipedia.org/wiki/Memoria\\_%28informatica%29](https://it.wikipedia.org/wiki/Memoria_%28informatica%29)
- [https://it.wikipedia.org/wiki/Memoria\\_flash](https://it.wikipedia.org/wiki/Memoria_flash)
- [https://it.wikipedia.org/wiki/Ordine\\_dei\\_byte](https://it.wikipedia.org/wiki/Ordine_dei_byte)
- <https://it.wikipedia.org/wiki/SRAM>

## **SENSORE DI TEMPERATURA**

- A. Picco, Analisi sui sensori: i sensori di temperatura, Tesi, 26 Marzo 2010.
- Datasheet PCT2075.
- F.Piccolo, Simulazione di una comunicazione fra dispositivi che utilizzano il protocollo I2C, Tesi.
- M.Bertocco – A. Sona, Introduzione alle misure elettroniche, Ed. Febbraio 2010-rev. 2.

## **SENSORE ULTRASUONI**

- Datasheet SRF01

## **SITI WEB**

- <https://it.wikipedia.org/wiki/Ultrasuoni>
- [http://www.robotdomestici.it/joomla/components/com\\_virtuemart/shop\\_image/product/SRF04\\_\\_da\\_3\\_cm\\_\\_4c993b7c1de84.jpg](http://www.robotdomestici.it/joomla/components/com_virtuemart/shop_image/product/SRF04__da_3_cm__4c993b7c1de84.jpg) (Immagine).
- [https://upload.wikimedia.org/wikipedia/commons/thumb/7/74/Ultrasound\\_range\\_diagram.svg/2000px-Ultrasound\\_range\\_diagram.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/7/74/Ultrasound_range_diagram.svg/2000px-Ultrasound_range_diagram.svg.png) (Immagine).

## **GSM-GPS**

## **SITI WEB**

- [http://www.dii.unisi.it/~abrardo/dispense\\_moretti.pdf](http://www.dii.unisi.it/~abrardo/dispense_moretti.pdf)
- [https://it.wikipedia.org/wiki/Global\\_System\\_for\\_Mobile\\_Communications](https://it.wikipedia.org/wiki/Global_System_for_Mobile_Communications)
- [https://it.wikipedia.org/wiki/Sistema\\_di\\_posizionamento\\_globale](https://it.wikipedia.org/wiki/Sistema_di_posizionamento_globale)

## **RTC**

- Datasheet DS2417

### **SITI WEB**

- [https://it.wikipedia.org/wiki/Real-time\\_clock](https://it.wikipedia.org/wiki/Real-time_clock)

## **APPENDICE**

### **SITI WEB**

- <http://it.emcelettronica.com/differenze-tra-microprocessore-e-microcontrollore>
- [https://it.wikipedia.org/wiki/MISRA\\_C](https://it.wikipedia.org/wiki/MISRA_C)

## Appendice

### *A.1 Differenza tra microcontrollore e microprocessore*

Quali sono le differenze tra microprocessore e microcontrollore? Il primo potrebbe essere definito come il "cuore" di un computer, fisso o portatile che sia. Dal punto di vista tecnico è l'integrazione di una serie di funzioni in un unico pacchetto IC ed è un componente che ha bisogno di numerose integrazioni esterne aggiuntive per poter funzionare, tra le quali la memoria, un oscillatore di clock, periferiche di ingresso e uscita. Inventato nel 1971, il microprocessore è attualmente l'implementazione fisica più comune di una CPU, ed è utilizzato su quasi tutti i computer e i dispositivi digitali come telefoni cellulari e scanner.

Un microcontrollore, invece, riunisce tutti gli elementi all'interno di un unico piccolo contenitore, e in teoria non ha bisogno di altri componenti esterni per poter funzionare. Tutto è infatti racchiuso in un unico chip, compresa la memoria per il programma, la memoria RAM, l'oscillatore di clock, il circuito di reset e le periferiche.

Le capacità di calcolo di un microcontrollore sono estremamente ridotte. Ad esempio la memoria RAM è formata da qualche centinaio di celle, e di solito non è espandibile. I microprocessori, al contrario, possono essere usati per effettuare elaborazioni complesse su grandi quantità di informazioni. Tra le tipiche applicazioni di un microcontrollore ci possono essere gli antifurti, gli strumenti di misurazione, quelli per la regolazione della luminosità, i carica batterie e i trasmettitori/ricevitori. Per questi motivi, i microcontrollori sono progettati per eseguire un piccolo insieme di funzioni specifiche, ad esempio nel caso di un Digital Signal Processor, che svolge un piccolo insieme di funzioni di elaborazione del segnale ed è ampiamente utilizzato per regolare i freni su tutte e quattro le ruote, o per regolare l'aria condizionata in auto.

Invece un microprocessore esegue una vasta gamma di compiti in un PC e può essere considerato il "motore" di elaborazione dei dati, racchiuso in un unico circuito integrato, capace di leggere, elaborare e scrivere informazioni in una memoria o in altri dispositivi digitali. La costruzione dei microprocessori fu resa possibile dall'avvento della tecnologia LSI, fondata sulla nuova tecnologia "Silicon Gate Technology" sviluppata nel 1968 da un italiano, Federico Faggin della Fairchild. Con l'integrazione di una CPU completa in un solo chip permise di ridurre di molto il costo dei computer. Secondo la legge di Moore, il numero di transistor integrabili sullo stesso chip dovrebbe raddoppiare ogni 18 mesi e l'evoluzione del microprocessore ha seguito con buona approssimazione questa regola.

Il microcontrollore, quindi, è un dispositivo elettronico integrato su singolo chip e viene utilizzato generalmente in sistemi cosiddetti "embedded", cioè per applicazioni specifiche di controllo digitale. È progettato per interagire direttamente con il mondo esterno tramite un programma residente nella propria memoria interna e mediante l'uso di pin specializzati o configurabili dal programmatore. I microcontrollori sono disponibili in tre fasce di capacità elaborativa: 8 bit, 16 bit e 32 bit. Per i microcontrollori sono rilasciati sistemi di sviluppo amatoriali e professionali anche in modalità open source. A differenza dei microprocessori, il microcontrollore è progettato per avere la massima autosufficienza funzionale ed ottimizzare il rapporto tra il costo e le prestazioni in uno specifico settore di utilizzo.

Anche l'esecuzione delle applicazioni si basa su un'architettura hardware diversa da quella che utilizzano i microprocessori. Mentre questi ultimi eseguono i programmi applicativi sfruttando dispositivi di memoria di massa o a memoria volatile, i microcontrollori eseguono il programma applicativo che è solitamente memorizzato su un dispositivo di memoria ROM.

## *A.2 Programma main*

Per scrivere il codice si è cercato di seguire le istruzioni estratte dal MISRA C che è un insieme di linee guida di sviluppo software per linguaggio di programmazione informatica C sviluppato da MISRA<sup>1</sup> (Motor Industry Software Reliability Association);

Il suo scopo è di facilitare la sicurezza, la portabilità e l'affidabilità del codice nel contesto dei sistemi embedded, specificatamente quei sistemi programmati in ISO C. C'è anche un insieme di linee guida per MISRA C++. MISRA si è evoluto come un modello accettato ampiamente di buone pratiche da sviluppatori del settore aerospaziale, delle telecomunicazioni, strumenti medicali, della difesa, ferroviario, e altri. MISRA C non è uno standard open, i documenti delle linee guida devono essere comprati dagli utenti.

---

<sup>1</sup> MISRA (in italiano: Associazione dell'affidabilità del software nel settore degli autoveicoli), è una collaborazione tra costruttori di veicoli, fornitori di componenti e società di consulenza di ingegneria che cerca di promuovere le migliori pratiche nello sviluppo di sistemi elettronici legati alla sicurezza nei veicoli stradali e di altri sistemi embedded. A tal fine MISRA pubblica documenti che forniscono informazioni accessibili per gli ingegneri e la gestione, e organizza eventi per consentire lo scambio di esperienze tra gli operatori.

## PROGRAMMA

Viene qui proposto, a solo titolo di esmpio, il pezzo di codice più significativo del main dove vengono descritte le possibili operazioni di controllo in grado di utilizzare tutti i componenti descritti nella trattazione. In questo caso tutto dipende dal livello del pellet.

Prima del ciclo while sono state definite le variabili:

```
S16 gTemp;
U16 gDist;

U8 ySMSEscape=1;
U8 gStatoGSM;
U8 gNumero[]="328xxxxxxx";
U8 gMessaggio[BUFF_SIZE];
U8 gValueBattery;
U8 gAlarmBatt;

S8 gSig_Quality;

LibRtc_Date gDate;
```

e vengono dichiarate varie funzioni per inizializzare e configurare i vari componenti:

```
//Setto temperatura
LibTemp_tConfig zTempCfg;

zTempCfg.devMode = LibTemp_eNormal;//LibTemp_eShutDown;
zTempCfg.OSmode = LibTemp_eComparator;
zTempCfg.OSpolarity = LibTemp_eOSlow;

LibTemp_Init(&zTempCfg);

//Inizializzo distanza
LibULT_Init();
EN_GPRS ALIM_ON;

//setto RTC
DS4217_Init();

gDate.seconds;
gDate.minutes=54;
gDate.hours=15;
gDate.day=18;
gDate.month=12;
gDate.year=2015;

LIBRTC_SAVEDATE(&gDate);

//setto GSM
LibGSM_Init();
```

```

while(1)
{
  if(Cmt_1ms_Handler_Request())
  {
    //GSM
    switch(gStatoGSM)
    {
      case 0:
        //controllo presenza SIM
        if(LibGSM_SIM_Inserted() != eLibGSM_NoError)
        {
          break;
        }
        gStatoGSM++;
        break;

      case 1:
        //controllo network
        if(LibGSM_NetworkStatus() != eLibGSM_NoError)
        {
          gStatoGSM=0;
          break;
        }
        gStatoGSM++;
        break;

      case 2:
        //controllo qualità GSM
        LibGSM_SignalQuality(&gSig_Quality);
        gStatoGSM=0;

        break;

      case 3:
        //invio SMS
        sprintf(gMessaggio,"Temp: %d,%d C,\nBattery: %d mV,\nData: %d/%d/%d\nLivPellet:
        %d"/"/"100", gTemp/10,gTemp%10, gValueBattery, gDate.day,gDate.month,gDate.year, gDist);
        LibGSM_SendSMS(gNumero, gMessaggio, strlen(gMessaggio));
        gStatoGSM=0;
        ySMSEscape=0;

        break;
    }

    //leggo temperatura
    if( LibTemp_ReadTemp(&gTemp) != LibErr_eNoError )
    {
      LibTemp_Init(&zTempCfg);
    }

    //leggo RTC
    LIBRTC_READDATE(&gDate);

    //leggo batteria
    AplSub_Battery_Handler(&gAlarmBatt, &gValueBattery);
  }
}

```

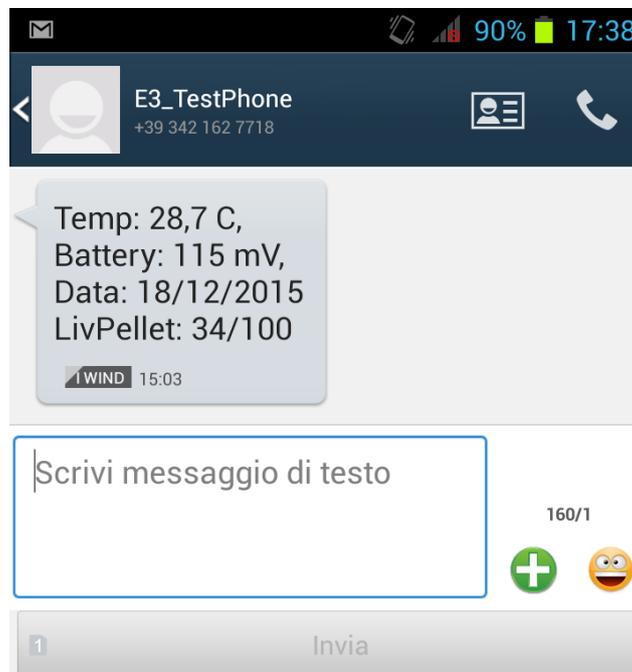
```

//leggo livello
LibULT_Read_Int(&gDist);

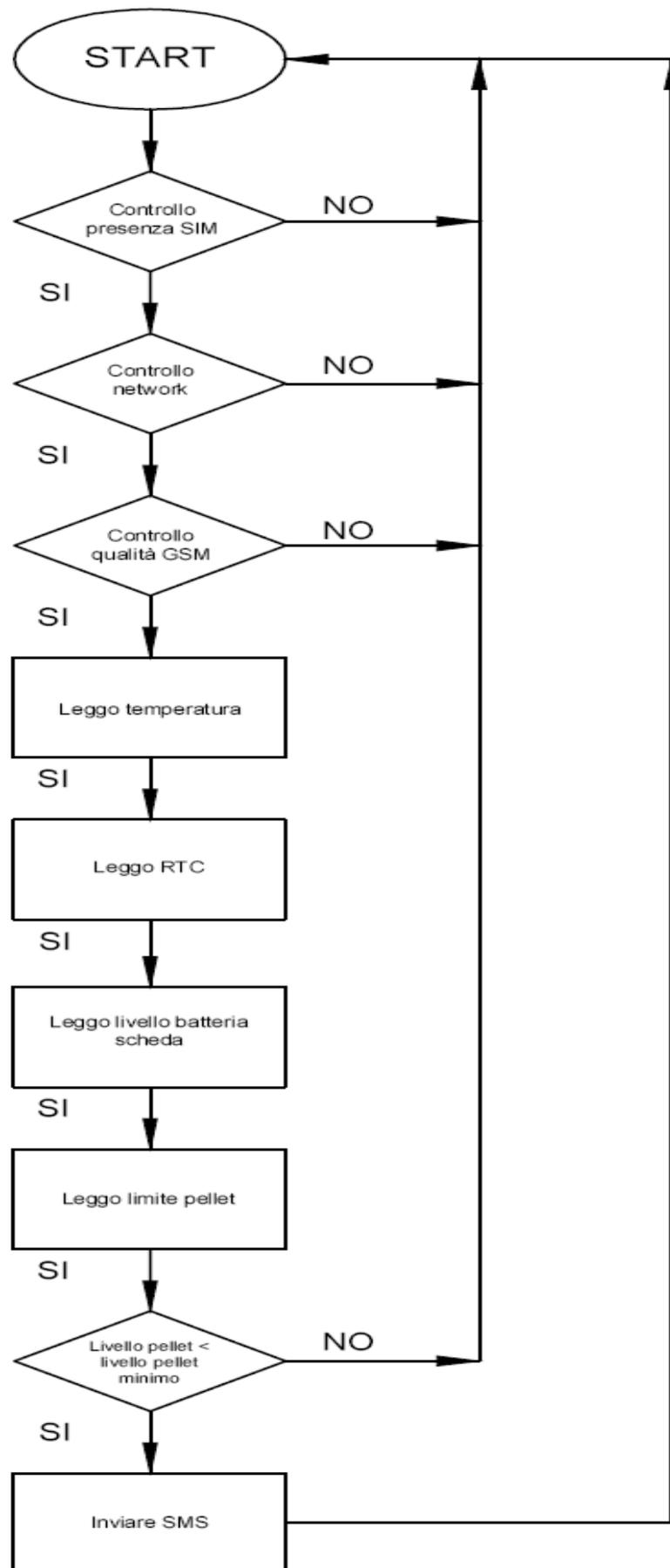
//mando 1 volta SMS di avviso per il livello pellet troppo basso
if(ySMSEscape==1)
{
  if(gDist<MIN_DIST_LEVEL)
  {
    gStatoGSM=3;
  }
}
}
}

```

Viene proposto in Fig. A.2.2 un diagramma a blocchi che rappresenta in modo grafico le operazioni svolte dal programma e nella Fig. A.2.1 il risultato del programma in caso di livello del pellet basso. Essendo il programma un ciclo while infinito non è presente il FINE ciclo.



A.2.1 Example of a view.



A.2.2 Block diagram.