



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA
DELL'INFORMAZIONE

Tesi di Laurea Triennale in

INGEGNERIA DELL'INFORMAZIONE

**Sviluppo di driver per sensoristica
ambientale in scenari
Internet of Things**

Relatore

Leonardo Badia

Correlatore

Riccardo Manfrin

Candidato

Alain Olivier

Anno Accademico 2012/2013

*Ai miei genitori
che sempre mi hanno sostenuto*

Abstract

La rete Internet si evolve sempre di più fino a penetrare in vari aspetti della vita quotidiana, ad esempio, tramite la capacità di interconnettere oggetti di uso comune, la cosiddetta "Internet delle cose" (meglio noto in letteratura nel suo corrispondente in inglese, "Internet of Things"). In questa tesi viene presentata una particolare applicazione di questo paradigma, ovvero, l'utilizzo di reti di sensori wireless per il monitoraggio ambientale. Successivamente, viene descritto un caso pratico consistente nell'implementazione di un *driver* per la lettura del valore della concentrazione di CO₂ ambientale, attraverso un sensore a tecnologia NDIR, da utilizzare in una *board* che andrà a inserirsi in una rete di sensori wireless accessibili, attraverso Internet, tramite protocollo IPv6. In seguito, verranno riportati graficamente dati relativi al funzionamento del sensore, nonché una finestra del campionamento che viene eseguito per effettuare la misurazione. Infine saranno presentati possibili sviluppi futuri del *driver*.

Indice

1	Introduzione	1
1.1	Motivazioni e struttura della tesi	3
2	Stato dell'arte	5
2.1	Internet of Things	5
2.2	Reti di sensori <i>wireless</i> nell'ambito IoT	7
2.3	Attività di ricerca su reti di sensori all'Università di Padova .	11
2.4	Sensori di CO ₂	13
3	Attività svolta presso Patavina Technologies srl	17
3.1	Descrizione preliminare	17
3.2	Implementazione del <i>driver</i>	23
3.2.1	La classe <i>co2_read</i>	24
3.2.2	La classe <i>measure_queue</i>	26
4	Risultati numerici	29
5	Sviluppi Futuri	33
	Bibliografia	37

1 | Introduzione

“L’Internet delle cose ha il potenziale per cambiare il mondo, così come fu per Internet. Forse anche di più”¹ è quanto afferma Kevin Ashton in un suo articolo [18] pubblicato su *RFID Journal* [2]. Può sembrare un’affermazione azzardata, ma si pensi all’impatto che ha avuto Internet nella nostra società. Si consideri ora che Internet of Things, in seguito indicato come IoT, è la naturale evoluzione di Internet che comporta un notevole miglioramento per quanto riguarda il raccoglimento, l’analisi e la distribuzione di dati e informazioni dell’ambiente che ci circonda.

La nascita di IoT va ricercata presso il Massachusetts Institute of Technology (meglio noto come MIT), dove un consorzio di ricerca, fondato nel 1999, l’Auto-ID Center, specializzato nella tecnologia Radio Frequency Identification (RFID, tradotto in italiano come Identificazione a Radio Frequenza), a cui apparteneva lo stesso Kevin Ashton, ne ha delineato l’architettura. IoT può essere definito come il momento storico in cui vi sono più oggetti che persone connessi alla rete Internet, questo secondo quanto stabilito da *Cisco IBSG* (Internet Business Solutions Group) [38].

Nel 2003, la popolazione mondiale era di circa 6.3 miliardi di persone [45] e si contavano 500 milioni di dispositivi connessi a Internet, secondo quanto affermato da George Colony della *Forrester Research* in [14]. Un semplice calcolo, indica che allora c’erano solamente 0.08 dispositivi per ogni individuo. E così, secondo la definizione dell’*IBSG*, IoT ancora non esisteva allora. A ogni modo, l’enorme successo degli *smartphone* e dei *tablet* che si ha avuto negli anni successivi, ha portato il numero di dispositivi connessi alla rete a

¹Originalmente “The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so.”

12.5 miliardi nel 2010, quando la popolazione mondiale contava 6.8 miliardi di individui [45], tant'è che il rapporto di dispositivi per persona ha superato l'unità per la prima volta nella storia. Di conseguenza l'avvento di IoT viene fatto risalire a cavallo degli anni 2008 e 2009. Con l'attuale *trend* di sviluppo, si prevede che nel 2015 i dispositivi connessi saranno 25 miliardi, mentre nel 2020 oltre 50 miliardi. In ogni caso, il numero di dispositivi per persona può sembrare basso, ma questo è dovuto al fatto che il calcolo viene eseguito basandosi sull'intera popolazione mondiale: restringendo l'analisi a coloro che hanno accesso alla rete, il numero di dispositivi per persona aumenta drasticamente. Così, noto che nel 2010 c'erano approssimativamente 2 miliardi di persone aventi accesso a Internet [44], il rapporto di dispositivi per persona supera le 6 unità. È interessante osservare che dai risultati condotti da un gruppo di ricercatori cinesi, la dimensione della rete Internet obbedisce alla legge di Moore: presenta quindi un andamento esponenziale che prevede un raddoppio di dimensioni ogni 5.32 anni [15].

È comunque da notare il fatto che, da quando la rete Internet è nata, al grandissimo sviluppo tecnologico non è corrisposto un cambiamento del paradigma di comunicazione: dalla fase iniziale di ricerca a scopi militari, che ha visto protagonista la Advanced Research Projects Agency NETwork (ARPANET, in italiano Rete dell'Agenzia dei Progetti di Ricerca Avanzata), alla fase attuale del *social web* [32] si ha principalmente una interazione tra persone. L'innovazione che porta con sé IoT consiste nell'introdurre una nuova forma di interazione, non più ristretta alle persone, ma tra persone e oggetti, denotata anche come Man-Machine Interaction (MMI), e pure tra oggetti e oggetti. IoT diventa così la prima vera evoluzione di Internet.

Un settore particolarmente rilevante di IoT è quello legato alle reti di sensori wireless, note in letteratura come Wireless Sensor Networks e in seguito riferite come WSN. Queste saranno trattate estesamente nel capitolo seguente.

1.1 Motivazioni e struttura della tesi

La stesura della presente nasce da un'attività di studio e implementazione di un *driver* atto a pilotare un sensore a tecnologia Non Dispersive InfraRed (in seguito indicata come NDIR, InfraRosso Senza Dispersione), in grado di rilevare la concentrazione ambientale di CO₂, svolta durante il mese di aprile, anno 2013, presso Patavina Technologies srl,² una *spin-off* dell'Università di Padova, impegnata nel settore delle Telecomunicazioni e in particolare nelle WSN. Di conseguenza, i risultati ottenuti possono essere classificati all'interno delle WSN, quindi, più in generale, nell'ambito IoT.

Viene ora delineata la struttura della presente tesi. Nel secondo capitolo sarà data una descrizione più tecnica di IoT e le WSN. Ne verrà data una breve caratterizzazione e in seguito saranno presentati dei casi di utilizzo. In particolare, vi sarà una sezione dove verranno descritti i lavori svolti all'interno dell'Università di Padova inerenti tali reti di sensori. Successivamente verrà analizzata una tipologia di sensori che è quella dei sensori di gas. Nello specifico verrà descritta la tecnologia NDIR, sfruttata dalla maggior parte di questi sensori. Infine saranno presentati casi di utilizzo di WSN dove viene monitorata la presenza di gas. Nel terzo capitolo sarà descritto il lavoro che è stato svolto presso la Patavina Technologies srl. Verrà ripercorso lo studio che è stato svolto per ultimare la realizzazione del *driver* per il pilotaggio del sensore NDIR utile alla misurazione della concentrazione ambientale di CO₂. Segue il quarto capitolo dove verranno presentati risultati numerici caratterizzanti il sensore utilizzato. Si concluderà con il quinto capitolo delle conclusioni dove verrà presentata una serie di possibili sviluppi futuri e applicazioni del *driver* e della *board* su cui viene inserito.

²Patavina Technologies srl. <http://www.patavinatech.com>

2 | Stato dell'arte

2.1 Internet of Things

Nel capitolo precedente è stata data una formale definizione di IoT. Ma cos'è effettivamente IoT? Come suggerisce il nome stesso, si tratta dell'estensione della rete Internet agli oggetti, quindi non più solamente *server*, computer e *smartphone*. Si intende permeare l'ambiente - non necessariamente su larga scala, in quanto si prevedono anche scenari a livello microscopico, come nel campo medico (segue un esempio) - di oggetti interagenti tra loro e con gli umani attraverso Internet [27]. Si tratta di un processo che può aprire a nuovi scenari, in particolare a quello delle città intelligenti. Ciò avviene dotando, prima di tutto, gli oggetti di un modulo per la connettività *wireless*. Una volta che gli oggetti sono collegati alla rete, i dati raccolti potranno essere utilizzati da altri oggetti (attuatori), oppure, organizzati al fine di creare una descrizione fisica del luogo circostante.

Vanno comunque menzionati alcuni ostacoli a questa forma di progresso, principalmente di natura tecnica. Innanzitutto, data la natura eterogenea della rete di IoT, è necessaria una standardizzazione, un insieme di protocolli globalmente accettati, così da costituire un'unica realtà. Così, dato che al giorno d'oggi la rete Internet è essenzialmente basata sul protocollo TCP/IPv4, che, oggi, non risulta essere più adatto a indirizzare il numero attuale di dispositivi, è chiaro come sia necessaria l'adozione di un nuovo *standard*. La scelta ricade su IPv6 che oltre a permettere l'indirizzamento di molti più dispositivi, offre maggiore sicurezza per quanto riguarda lo scambio di informazioni [9]. Altro ostacolo non indifferente è legato al fattore

energetico [4], in modo particolare nel settore dei sensori. Infatti, è requisito necessario che i sensori collegati alla rete siano autosufficienti e quindi debbano essere in grado di autoalimentarsi sfruttando l'ambiente circostante; al limite, saranno alimentati attraverso l'utilizzo di batterie. Ovviamente, in quest'ultimo caso, è ugualmente richiesta una buona efficienza energetica, necessaria per limitare interventi di carica o sostituzione delle batterie.

Di seguito, si analizzano alcuni casi di possibile applicazione di IoT di interesse civile, commerciale e scientifico.

Un campo che può trarre benefici da IoT è quello medico. Si pensi, ad esempio, a una pillola in grado di collegarsi a Internet e in grado di aiutare il medico a pervenire a una diagnosi e determinare le cause di certe malattie [10]. Oppure, si immagini un paziente affetto da diabete che, per curarsi, si affida a un sensore impiantato che costantemente monitora la glicemia nel sangue e avverte lui e il suo medico curante, che disporrà di uno *smartphone* con il quale può comunicare con il sensore, qualora ci siano condizioni critiche [6]. Questo dell'Electronic Health-Care (indicato spesso come EHC e letteralmente "cura elettronica della salute") è solamente uno dei campi in cui IoT avrà influenza.

Curioso è il caso di Sparked, una start-up olandese che impianta sensori nelle orecchie delle mucche. In questo caso, gli allevatori possono monitorare la salute e gli spostamenti del bestiame [17].

Altro esempio di applicazione è nel campo civile. Nelle grandi città dei paesi emergenti sono numerosi i problemi dell'approvvigionamento d'acqua. Esempio è la città indiana di Mumbai. Nella città, infatti, esiste una certa disparità del costo dell'acqua, da quartiere a quartiere. IoT può essere d'aiuto per risolvere la situazione. Dotare i contatori dell'acqua di un accesso a Internet, potrebbe senz'altro essere utile alle autorità per monitorare perdite nelle condutture e localizzare fenomeni di evasione, garantendo così un più razionale utilizzo della preziosa risorsa idrica [13].



Figura 2.1: Un bovino equipaggiato con un sensore GPS autoalimentato.

2.2 Reti di sensori *wireless* nell'ambito IoT

Nell'ambito IoT si inseriscono perfettamente le reti di sensori *wireless*, più volte indicate come Wireless Sensor Network (in seguito WSN). Queste reti sono essenzialmente costituite da sensori dotati di accesso a Internet. Possiamo considerare queste reti come un ponte al mondo fisico, un mezzo di comunicazione con la realtà fisica dell'ambiente.

Se lo scopo principale di IoT è quello di costituire una realtà dinamica che rivoluzioni il modo di interagire con gli altri utenti, gli oggetti e soprattutto con l'ambiente è chiaro come le WSN divengano quindi un fattore fondamentale per questo processo. Esse determinano un'estensione consistente dei dispositivi connessi alla rete Internet. Oltretutto, questi sensori sono dotati di capacità computazionali, che, seppur modeste, consentono una certa elaborazione dei dati che raccolgono. In questo modo, i dispositivi di prossima generazione e le persone saranno in grado di comunicare a un livello più profondo con gli oggetti, appunto costituendo la nuova IoT, un'infrastruttura pervasiva, dove nodi di qualsiasi genere concorrono a fornire e scambiare

servizi e informazioni.

Essenzialmente una WSN è in gran parte stazionaria, intendendo che la maggior parte dei sensori che la costituiscono sono ancorati all'ambiente che monitorano. È comunque previsto un insieme di sensori mobili che contribuiscono al raccoglimento delle informazioni. Tali reti devono avere una certa estensione, il che consente di costituire uno *smart environment*, ambiente intelligente, cioè dotato di capacità di elaborazione e interazione [11]. Infine, le WSN devono essere omogenee, ovvero costituite da sensori simili tra loro. È bene osservare che queste reti non necessitano di una stazione base che coordini l'attività di tutti i sensori: sono infatti un caso particolare di quelle che vengono definite reti *ad hoc*, cioè reti decentralizzate [8].

I vantaggi che comporta una WSN rispetto a una rete di sensori cablata sono numerosi. Banalmente, non è necessario un collegamento fisico tra i dispositivi: questo non solo comporta una notevole riduzione del costo economico del sistema, ma permette inoltre di utilizzare i sensori in applicazioni non trattabili con reti cablate (si pensi ad esempio a sistemi in rotazione). Oltretutto, un'eventuale modifica dell'impianto non comporta una sostanziale modifica della rete di sensori, in quanto non vincolati da cablature, e così un'eventuale aggiunta di nuovi sensori alla rete non necessita di posare altri cavi. Si può quindi constatare una certa flessibilità. Altro vantaggio è una minore manutenzione: infatti l'assenza di cavi elettrici evita i problemi ad essi legati, come ad esempio la corrosione, e inoltre è solamente richiesta l'eventuale sostituzione della batteria del sensore. Infine consentono un notevole risparmio energetico, in quanto, come già accennato, i sensori sono autoalimentati oppure utilizzano delle batterie, e quindi, in fase di progetto, particolare attenzione viene prestata all'efficienza del sistema. Un'ulteriore conferma è data dall'adozione di protocolli di trasmissione concepiti appositamente per gestire al meglio il lato energetico. Tra i protocolli maggiormente utilizzati è necessario citare ZigBee e 6LowPan [47] [35].

ZigBee è uno *standard* di protocolli proprietario concepito appositamente per piccoli moduli *wireless* a bassa potenza di trasmissione. Viene utilizzato per creare reti *ad hoc*, con topologia a *mesh*, e caratterizzate da bassi *rate* di trasmissione. Le reti a *mesh* sono caratterizzate dalla presenza di nodi in gra-

do di trasmettere, ricevere e fungere da ripetitori: ciò consente di costituire una rete cooperativa dove ogni nodo si serve degli altri nodi per trasmettere a un qualsiasi altro terminale non direttamente raggiungibile. L'instradamento dei pacchetti, meglio noto come *routing* diventa quindi fondamentale. Una delle tecniche maggiormente utilizzate è il *flooding* [5] che vede la disseminazione dei pacchetti nell'intera rete [37]. Incidentalmente, la *board* per cui è stato sviluppato il *driver* utilizza un modulo ZigBee. Poggia le sue basi sullo *standard* IEEE 802.15.4. Quest'ultimo definisce i primi due livelli del modello ISO/OSI, ovvero il livello fisico (*physical layer*) e il livello *data link*. È stato ideato di proposito per le reti *wireless* personali a basso *rate* di trasmissione, spesso indicate come Low Rate Wireless Personal Area Network (LR-WPAN). ZigBee si occupa di definire i livelli successivi per completare il modello ISO/OSI [48].

6LoWPan deriva da IPv6 over Low power Wireless Personal Area Networks, ovvero IPv6 esteso a reti *wireless* personali a bassa potenza. L'idea che ha portato alla nascita di questo protocollo è quella di estendere IPv6, il quale garantisce ottime capacità di indirizzamento (sono previsti 128 bit per l'indirizzamento, contro i 32 di IPv4), a piccoli oggetti, come possono essere i sensori, dotati di modeste capacità di calcolo, di memoria e velocità di trasmissione. Caratteristica principale di 6LoWPAN è quella di ridurre le dimensioni dell'*header* di ogni pacchetto di trasmissione IPv6 quantificabile in 40 byte. Attraverso una tecnica detta di *stacked headers*, peraltro utilizzata dallo stesso IPv6, la dimensione dell'*header* in un pacchetto 6LoWPAN può essere ridotta fino a 4 byte. Tutto ciò si traduce in un minore dispendio energetico [21]. Altri vantaggi sono una maggiore velocità di connettività e una maggiore compatibilità con le preesistenti architetture; oltretutto, 6LoWPAN permette un rapido sviluppo di applicazioni e la loro integrazione con i servizi Web esistenti basati sulle reti a IP [9].

Sono degni di nota numerosi studi condotti sulle WSN. Molto influente è stato il progetto SENSEI [33]. Nato il primo gennaio 2008 e terminato il 31 dicembre 2010, ha annoverato 19 partner da undici paesi europei. In [28] viene brevemente descritto l'obiettivo preposto dal progetto, ovvero la creazione di un'architettura per costituire una rete sicura e pervasiva. Risultati

tangibili raggiunti dal progetto sono la realizzazione di un'infrastruttura che consente una facile integrazione di un grande numero di WSN in un sistema globale, assicurando oltretutto il supporto per la gestione e la sicurezza della rete e delle informazioni; una certa unificazione per quanto riguarda l'accesso alle informazioni; una ottimizzazione volta al risparmio energetico dei protocolli che gestiscono la rete, tra le altre cose, sviluppando un trasmettitore a bassa potenza (5 nJ/bit); infine, la creazione di una rete estesa a livello europeo che permette lo studio dell'integrazione delle WSN nell'Internet del futuro.

Si riportano, infine, due casi particolari che illustrano le potenzialità delle WSN.

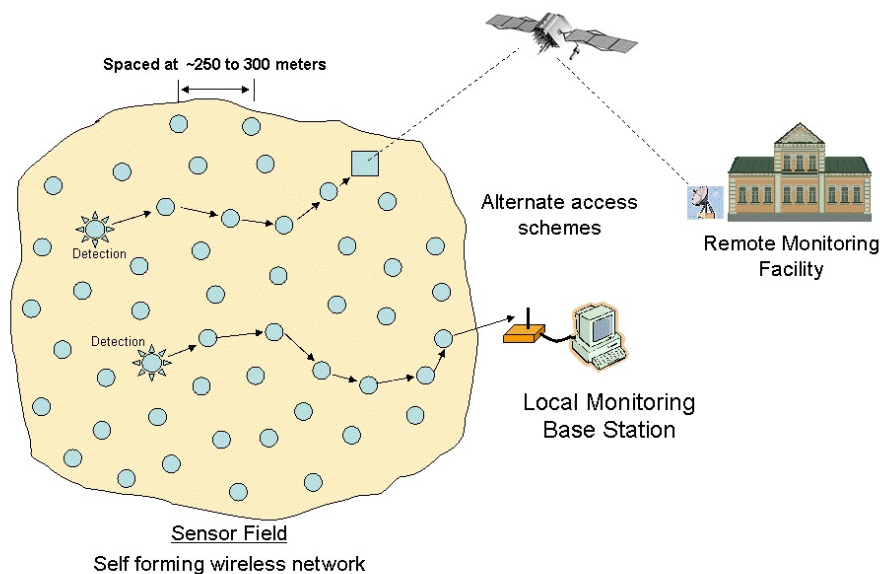


Figura 2.2: Rete di sensori *wireless* installata nel terreno e comunicante con un centro abitato.

Ad oggi, sono in corso numerosi studi su come prevenire i danni del catastrofico Big One, il terremoto che secondo gli esperti dovrebbe colpire la California nelle prossime decenni [7]. Una rete di sensori in grado di monitorare movimenti del terreno, e disseminati nell'intero territorio e a diverse

profondità, può essere in grado di limitare gli effetti che una simile calamità causerebbe. In particolare, l'intera rete sarebbe in grado di localizzare l'epicentro del terremoto e con un tempo inferiore ai trenta secondi sarebbe in grado di allarmare i vicini centri abitati e le persone che vi abitano. Le città intelligenti costituite da edifici di futura generazione, sapranno difendersi al meglio dall'onda sismica, e la gente, una volta allarmata, sarà conscia del pericolo imminente [29]. È chiaro quindi come le WSN e IoT possano realmente migliorare le nostre vite.

Ulteriore applicazione delle WSN è nel campo dei sistemi di trasporto intelligente, Intelligent Transportation System (in seguito ITS). Gli attuali ITS sono in grado di localizzare un veicolo in una posizione fissata e inoltre il costo dei cavi di trasmissione e di alimentazione implicano un elevato costo di costruzione e manutenzione. Le WSN, grazie al basso consumo energetico e la già citata flessibilità, permettono di risolvere questi problemi. Si immagini infatti che ogni veicolo sia dotato di un sensore in grado di rilevarne la posizione. Questi sensori, interagendo tra loro, possono aiutare a rispettare distanze di sicurezza, gestire incolonnamenti e le percorrenze di strade congestionate [19].

2.3 Attività di ricerca su reti di sensori all'Università di Padova

L'ateneo patavino è impegnato nella ricerca in questo settore. Un progetto degno di nota è il Wireless Sensor networks for city-Wide Ambient Intelligence (riferito in seguito con il suo acronimo WISE WAI) [41]. Lo scopo è quello di sviluppare e definire soluzioni per dotare l'ambiente di un'intelligenza, essenzialmente creare delle città intelligenti. Per fare questo, WISE WAI si occupa principalmente nello sviluppo di una rete di sensori *wireless* e attuatori a larga scala. Molta importanza viene data al consumo energetico e all'efficienza. Il progetto è strutturato in quattro principali aree tecniche: lo studio dei problemi legati alle applicazioni e l'infrastruttura della rete; lo sviluppo di *hardware* a basso consumo energetico, e in modo particolare di

trasmettitori CMOS a radiofrequenze, il che rappresenta senza dubbio una grande sfida in quanto è bene ricordare che una WSN è costituita da numerosi sensori che dovranno trasmettere in contemporanea, causando inevitabilmente dei problemi di interferenza; lo sviluppo di protocolli e algoritmi adattativi, infatti una WSN richiede particolari accorgimenti per quanto riguarda il *routing* e il Medium Access Control (MAC), e inoltre si rende necessario un particolare protocollo che renda possibile una diffusione efficiente dei dati raccolti dal complesso dei sensori; la creazione di esperimenti e dimostrazioni [8].

All'interno di WISE WAI, un gruppo di ricerca sta realizzando una WSN che sia facilmente estensibile e costituita da tre tipi di sensori: base station, nodi mobili e nodi specializzati. La rete è attualmente installata nei tre edifici che costituiscono il Dipartimento di Ingegneria dell'Informazione (DEI)¹: essa conta oltre 200 nodi statici e altri 100 mobili. L'obiettivo ultimo non è tanto rappresentato dalla costituzione della rete, bensì dalla sua ottimizzazione. A questo scopo ogni nodo è accessibile dall'esterno attraverso protocollo IPv6/UDP e comunica attraverso un trasmettitore a connettività IEEE 802.15.4, particolarmente indicata per applicazioni che richiedono un basso consumo di energia. I nodi offrono servizi Web, servizi di localizzazione, monitoraggio dei parametri ambientali (intensità luminosa, temperatura e umidità, ma anche parametri di funzionamento del sensore stesso come il livello della batteria e la potenza in trasmissione), la sperimentazione di protocolli, nonché alcune funzionalità di basso livello come ad esempio la possibilità di cambiare l'applicazione installata, la programmazione e l'aggiornamento del *software on air* attraverso un particolare protocollo chiamato SYNAPSE++ [31] e infine la possibilità di resettarsi in caso di malfunzionamenti. I servizi Web vengono definiti come tecniche per lo sviluppo di applicazioni sfruttando *standard* del Web come HTTP. Si evidenzia la presenza di due approcci diversi nello sviluppo di servizi Web: uno è l'approccio basato sul protocollo Simple Object Access Protocol (meglio noto come SOAP), che sfrutta lo scambio di messaggi per l'invocazione di servizi remoti; il secondo è l'approccio basato sull'architettura Representational State Transfer (RE-

¹<http://www.dei.unipd.it/wdyn/?IDsezione=4506>

ST), dove ogni sensore rappresenta una risorsa indirizzabile univocamente tramite Uniform Resource Locator (più conosciuto come URL) e accessibile attraverso i tipici metodi di *HTTP* come *GET*, *POST*, *PUT* e *DELETE*. [9].

Altro studio interessante, condotto da ricercatori dell'Università di Padova, e che sfrutta le WSN, riguarda una rete di sensori *wireless* installata su un piccolo edificio residenziale di due piani situato a Padova. In particolare, ogni sensore è in grado di monitorare temperatura, umidità, irraggiamento solare e così, globalmente, l'intera rete rende possibile l'identificazione termodinamica dell'edificio, fattore che oggigiorno diventa sempre più importante [39].

2.4 Sensori di CO₂

Attualmente, per quanto riguarda la misurazione della concentrazione di CO₂ esistono diverse tecnologie sfruttate dai sensori: una è quella basata sull'assorbimento infrarosso, mentre le altre sono basate su sensori a elettrolita. Un esempio di quest'ultimo tipo sono i sensori *Severinghaus* potenziometrici. Sono essenzialmente costituiti da un elettrodo di vetro riempito da una soluzione acquosa di bicarbonato e ricoperto da una membrana permeabile al gas ma non all'acqua. La misura della concentrazione di anidride carbonica avviene valutando il pH della soluzione acquosa, in quanto la CO₂, in acqua, forma acido carbonico, H₂CO₃, secondo la reazione all'equilibrio $\text{CO}_2(aq) + \text{H}_2\text{O}(l) \rightleftharpoons \text{H}_2\text{CO}_3(aq)$ [12] e di conseguenza il pH è direttamente proporzionale alla concentrazione del gas. Questo genere di sensori, oltre ad avere elevati costi di manutenzione, si basa su una misura indiretta e inoltre, il pH della soluzione può variare anche a causa della presenza di altri gas [25].

Un altro metodo per misurare la concentrazione è la spettrografia di massa, dove, attraverso un campo magnetico, si riescono a separare le molecole di un campione [20]. Il metodo, benché efficace, è altamente costoso, e oltretutto non si è ancora in grado di realizzare un apparato di ridotte dimensioni. Altro metodo è la cromatografia, che rispetto alla spettrografia di massa, è molto meno costosa, ma presenta un tempo nettamente superiore per effettuare l'a-

nalisi, ma, così come per il metodo precedente, ancora non è disponibile un sistema di dimensioni ridotte [26].

Sono questi i motivi per cui ai nostri giorni la maggior parte dei sensori sul mercato sfruttano la tecnologia ad assorbimento infrarosso, specialmente la tecnologia Non Dispersive InfraRed (in seguito indicata come NDIR, InfraRosso Senza Dispersione) [42].

Il principio di funzionamento di questi sensori è basato sulla spettroscopia, in particolare, sull'assorbimento energetico caratteristico della CO_2 nella zona infrarossa. È noto infatti che ogni elemento presenta spettri di assorbimento caratteristici, determinati dai livelli energetici tipici di ogni molecola: la quantizzazione dell'energia secondo la teoria dei quanti, e l'equazione che rappresenta l'energia di un fotone, emesso o assorbito, in funzione della sua lunghezza d'onda ν , $E = h\nu$ (dove $h = 6.62606957 \times 10^{-34} \text{ m}^2 \cdot \text{kg/s}$) [1], implicano l'assorbimento e l'emissione di radiazione a determinate lunghezze d'onda. Nello specifico caso dell'anidride carbonica, nella regione infrarossa, si rilevano assorbimenti di radiazioni con lunghezza d'onda di 2.7 μm , 4.7 μm e 15 μm .

La struttura di un sensore NDIR risulta essere particolarmente semplice. La luce infrarossa, necessaria per la spettroscopia, viene generata da un emettitore monocromatico e viene convogliata in due tubi identici. Uno è denominato di riferimento, ed è riempito di gas che non presenta assorbimento nell'infrarosso, come, ad esempio, l'azoto, N_2 . Il secondo tubo, invece, contiene il gas da analizzare, e in particolare assorbirà più o meno della radiazione infrarossa a seconda della concentrazione del gas. Così, un rilevatore confronta l'energia del fascio proveniente da entrambi i tubi: la differenza è proporzionale alla quantità di gas [25] [42].

I sensori NDIR commercializzati, presentano, tra i vari *pin*, sia un accesso al rilevatore relativo al tubo di riferimento, che a quello attivo. Non fa differenza il sensore di CO_2 Alphasense IRC-A1 che abbiamo utilizzato. Il nostro compito è stato scrivere un *driver* per microcontrollore, atto a valutare i segnali dei *pin* citati e calcolare così la concentrazione del gas. È bene notare come la misurazione sia relativa e determinata da una differenza di energie, quindi non assoluta. Questo implica necessariamente un processo di

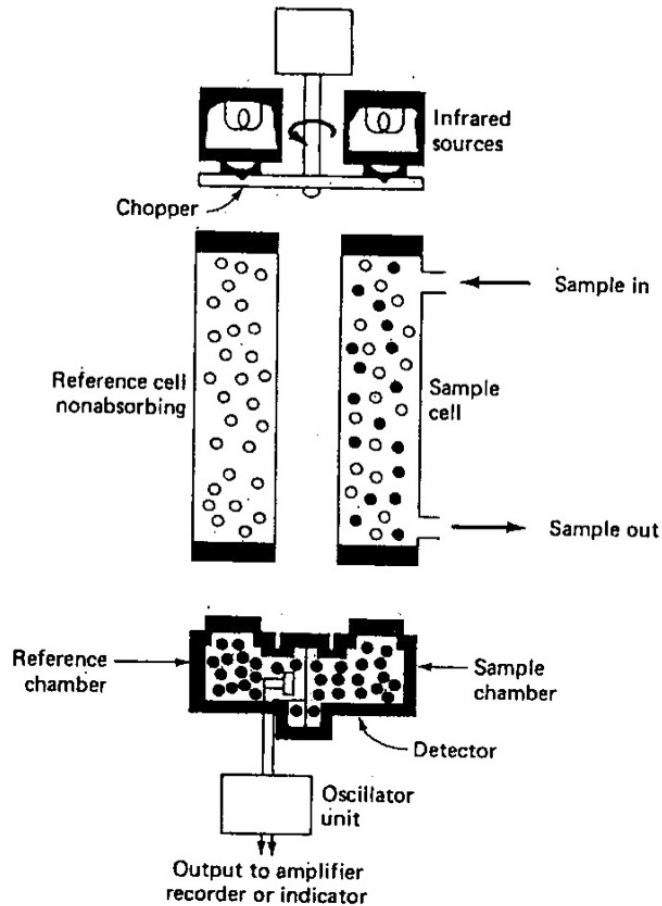


Figura 2.3: Struttura di un sensore NDIR. Si distinguono chiaramente la cella attiva e quella di riferimento.

calibrazione.

Rispetto ai sensori e metodi suddetti, i vantaggi della tecnologia NDIR sono un costo minore, dimensioni inferiori, tempi di risposta ridotti, la possibilità di misurare continuamente e infine un dispendio energetico inferiore [34]. Sono questi i principali motivi per cui i sensori a tecnologia NDIR sono preferibili per quelle applicazioni che richiedono una misura in tempo reale della concentrazione di CO_2 con il vincolo ulteriore di basso consumo energetico (requisito per le WSN). Vi sono comunque degli svantaggi da riportare: variazioni di temperatura, illuminazione, presenza di vapore o monossido di

carbonio modificano i livelli energetici infrarossi, causando quindi variazioni nella misura [46].

Per quanto riguarda le possibili applicazioni che può avere un sensore di CO_2 , e in modo particolare nell'ambito WSN nell'architettura IoT, esse spaziano in numerosi campi. Oggigiorno, il surriscaldamento globale rappresenta una seria minaccia per il pianeta. È per questo motivo che la riduzione dei gas emessi in atmosfera è un obiettivo sempre più ricercato. WSN che monitorano la concentrazione di anidride carbonica, possono quindi essere d'aiuto per quanto riguarda il monitoraggio della qualità dell'aria [40]. Si pensi, ad esempio, a come possa essere utilizzato per monitorare la concentrazione del gas all'interno di una stanza quale può essere un'aula universitaria: attraverso un sistema centralizzato che monitora il livello di CO_2 nei locali e un'opportuna rete di attuatori, sarebbe possibile ventilare più o meno l'aula così da alleggerire l'aria. Oppure si pensi ad applicazioni industriali, come ad esempio la produzione di etanolo e urea, dove è richiesto un ambiente con una concentrazione ben precisa del gas. Un'applicazione più critica può essere un rilevatore d'incendio basato su sensore NDIR di CO_2 [43]. È comprensibile come sensori di questo tipo consentano lo sviluppo di città intelligenti: se ogni edificio fosse dotato di una serie di sensori antincendio costituenti una rete estesa a livello cittadino, i vigili del fuoco sarebbero immediatamente avvertiti di un eventuale rogo e della sua posizione.

Altra applicazione di sensori di CO_2 si trova a livello industriale nel campo agroalimentare. Si pensi a possibili applicazioni dove è richiesta la misurazione della concentrazione del gas, come, ad esempio, il monitoraggio di biogas, di scambi di gas, oppure nei processi di distillazione alcolica. Il confezionamento degli alimenti, molto spesso, avviene in ambiente con specifiche composizioni di ossigeno, anidride carbonica e azoto così da prevenire la formazione di muffe. Rilevazioni e misurazioni di livelli eccessivi di CO_2 possono garantire la sicurezza degli agricoltori che si trovano a lavorare in ambienti esposti al gas come ad esempio granai e silos. È chiaro quindi come la richiesta di sensori di anidride carbonica sia consistente anche nel settore alimentare, per il controllo di sistemi di ventilazione, trattamento del cibo, trasporto e analisi della qualità [25].

3 | Attività svolta presso Patavina Technologies srl

3.1 Descrizione preliminare

L'attività svolta durante il mese di Aprile 2013, presso il laboratorio di Patavina Technologies srl, è stata lo sviluppo di un *driver software* per la lettura del valore della concentrazione ambientale di CO₂, da effettuarsi attraverso un sensore a tecnologia NDIR. Il sensore, IRC-A1 [16], prodotto da Alpha-sense, deve essere pilotato secondo le linee guida riportate in [23] e inoltre il dato utile della concentrazione ambientale deve essere estrapolato attraverso più passi che verranno delineati in seguito. In particolare, si rende necessario operare un campionamento della tensione in uscita al trasduttore. Da qui l'esigenza di sviluppare un *driver* che gestisca sia il pilotaggio che la raccolta dei dati e la loro elaborazione per ultimare la lettura.

Lo sviluppo del *driver* ha richiesto innanzitutto una fase preliminare di studio, a carattere prettamente tecnico, caratterizzata dalla comprensione del funzionamento dell'ambiente di sviluppo *software*, la consultazione delle *application notes* relative al sensore e lo studio della relativa scheda su cui viene montato, nonché della *board* sulla quale viene programmato il *driver*. Quest'ultima, sviluppata da Eutech Electronics srl,¹ e denominata EUTECH2, come accennato in precedenza, presenta un certo grado di modularità in quanto può essere utilizzata per il controllo di sensori diversi dal nostro, come, ad esempio, sensori di monossido di carbonio, CO, oppure di umidità.

¹Eutech Electronics srl. <http://www.eutechelectronics.it/contatti.htm>

EUTECH2 monta un microprocessore della famiglia AVR, sviluppato da Atmel: è l'ATxMEGA256A3 [3]. Si tratta di un microprocessore a 16 bit, 64 *pin*, che viene utilizzato per pilotare il sensore esterno alla scheda, quale può essere quello di CO₂, nonché altri sensori disponibili in EUTECH2 come ad esempio l'igrometro. Il microcontrollore svolge oltretutto funzioni di *power management*, ovvero di gestione delle alimentazioni dei moduli collegati alla scheda. Degna di nota è la presenza del modulo *wifi* ZigBee interno alla scheda e sempre pilotato dal microcontrollore che permette l'interfacciamento alla rete Internet attraverso IPv6, di fatto, realizzando il paradigma dell'IoT. È anche da segnalare la presenza di un alloggiamento per le batterie, il che permette di posizionare l'intero modulo lontano da una sorgente energetica.

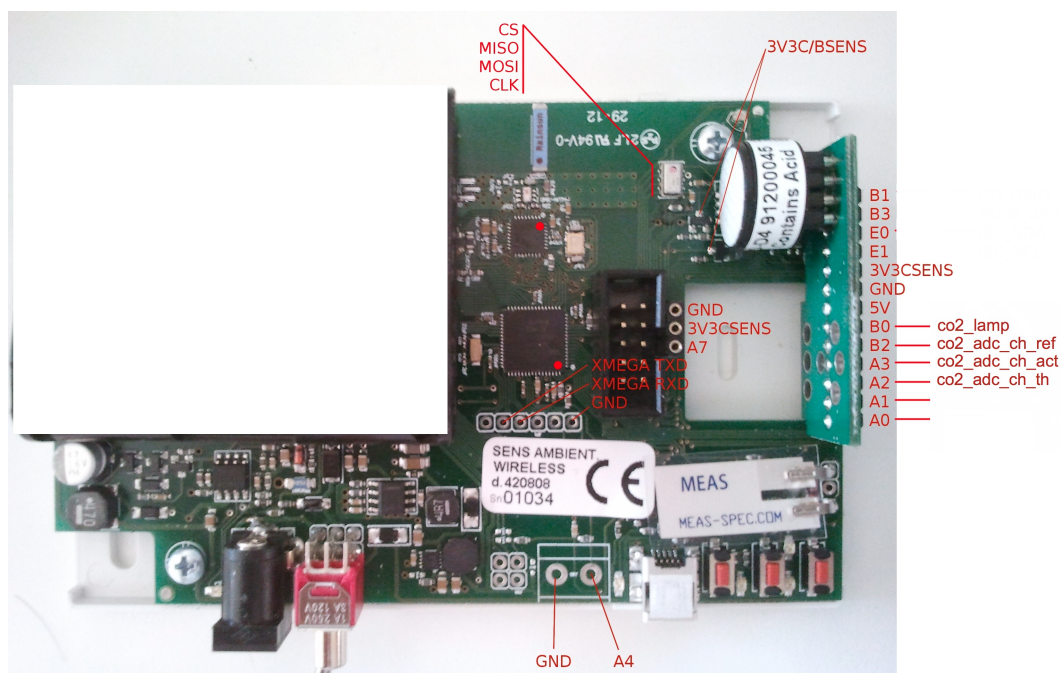


Figura 3.1: La scheda EUTECH2. Si può osservare sulla sinistra il modulo con il sensore, al centro, invece, il microcontrollore.

La programmazione del microcontrollore della scheda richiede l'installazione di una *toolchain* in ambiente Linux, che raccoglie gli strumenti utili a compilare il *driver* e a *flashare* il microprocessore. Applicazioni degne di

3. Attività svolta presso Patavina Technologies srl

nota, interne alla *toolchain*, sono *gcc* per la compilazione, *avrdude* (strumento reso disponibile dalla stessa Atmel per *flashare* i propri microcontrollori) e *gcc*, *debugger* necessario per testare la correttezza del *driver*. In particolare, quest'ultimo è stato implementato in linguaggio C++, linguaggio di programmazione evoluzione del C, alla quale è stata aggiunta la possibilità di realizzare applicazioni attraverso il paradigma della programmazione orientata agli oggetti, nonché particolarmente indicato per questo tipo di applicazioni che richiedono un controllo totale dell'indirizzamento della memoria del microprocessore [36].

La stesura del *driver* parte da una solida base *software* per la gestione del microcontrollore e della scheda, interamente realizzata da Patavina Technologies srl. Attraverso opportune modifiche ad alcuni file - nello specifico ai Makefile, in cui si può specificare il *TARGET* (scheda su cui verrà caricata l'applicazione) - che comportano una diversa compilazione dei *file* sorgente, è inoltre possibile riutilizzare lo stesso codice per programmare altre schede diverse da EUTECH2.

In particolare, come viene illustrato nella figura 3.2, l'ambiente è organizzato nel seguente modo: si differenziano due cartelle, *system* e *apps*. La prima contiene quanto necessario per gestire le varie schede utilizzabili, i *driver* sviluppati per altri sensori, e infine tutte le interfacce e le classi utili a gestire i servizi del microcontrollore come ad esempio il modulo per la conversione da analogico a digitale (Analog to Digital Conversion e nel seguito indicato come ADC), gli *interrupt*, le porte di ingresso e uscita, meglio note come I/O (Input/Output). La seconda, invece, consta di numerose applicazioni che utilizzano quanto definito nella cartella *system*. Così, ad esempio, il *driver* che abbiamo sviluppato andrà a inserirsi in *system*, mentre l'applicazione di test del *driver* sarà posizionata nella directory *apps*.

La fase iniziale dello sviluppo del *driver* ha richiesto, pertanto, lo studio della già presente base *software*, con il fine ultimo di integrare al meglio il *driver* nel sistema, quindi, utilizzando metodi e interfacce già disponibili e limitando l'implementazione di nuove classi e metodi. Ad esempio, data la necessità di campionare la tensione in uscita al sensore, è stata sfruttata la già presente classe dedicata alla gestione del modulo ADC del microcontrollore.

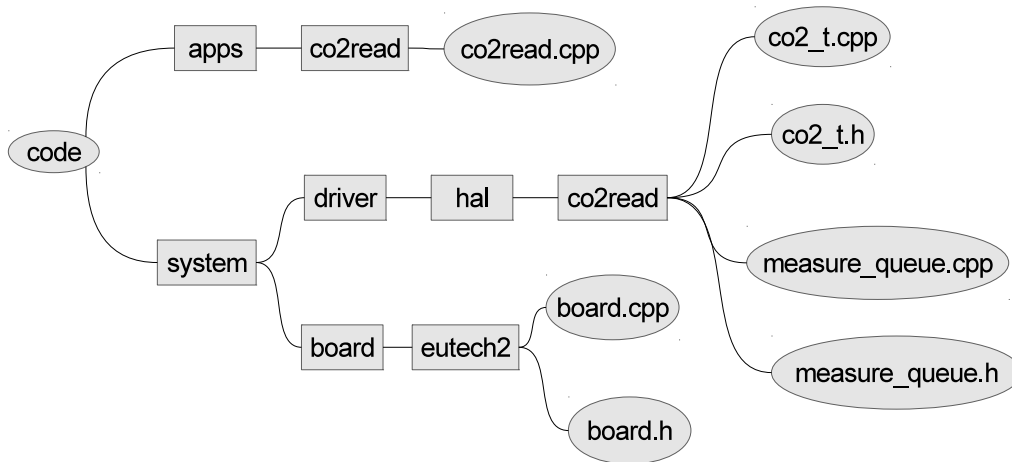


Figura 3.2: L'organizzazione dell'intero sistema *software*

Dopo questa prima fase, siamo passati a studiare quale fosse la migliore strategia per eseguire la lettura della concentrazione di CO₂ ambientale. Il sensore IRC-A1 viene montato su una scheda apposita che viene poi collegata a EUTECH2. Dal *datasheet* del sensore [16] si contano otto *pin*, di cui tre particolarmente rilevanti nella lettura della concentrazione. Si tratta, in particolare, dei *pin* 1, 4, 5, rispettivamente: *Lamp Return* che pilota l'emettitore infrarosso; *Detector Output* (nel seguito indicata come *ACT*); *Reference Output* (nel seguito *REF*). Gli ultimi due sono i due terminali citati nella sezione 2.4. *Lamp Return* è un *pin* necessario ad alimentare l'emettitore infrarosso del sensore. Viene controllato da un *transistor* NMOS sottoposto a un segnale rettangolare avente una frequenza di 2.5 Hz e generato dal microcontrollore, così come viene descritto in [23]. A causa di questa intermittenza dell'emettitore, consegue una variazione della tensione ai *pin* *ACT* e *REF*, che nella pratica mostrano un carattere sinusoidale isofrequenziale con il segnale che pilota *Lamp Return*. La misurazione della concentrazione del gas richiede essenzialmente la conoscenza dell'ampiezza dei due segnali sinusoidali in uscita ai *pin* di *ACT* e *REF* nonché lo svolgimento un calcolo

3. Attività svolta presso Patavina Technologies srl

particolarmente laborioso e pesante per il microcontrollore: in modo particolare, sono previsti elevamenti a potenza e logaritmi, così come descritto in [24]. Problemi di carattere tecnico, relativi all'ambiente di sviluppo, non permettono di integrare queste funzioni di tipo *float* all'interno del progetto. Così abbiamo scelto, drasticamente, di fornire i dati essenziali per il calcolo della concentrazione che sono l'ampiezza del segnale di *ACT* e di *REF*. Nello specifico, la misura della percentuale di CO₂ nell'aria è funzione del rapporto di queste due grandezze. Diamo ora una breve descrizione su quali siano i passaggi matematici da svolgere per arrivare al risultato utile della concentrazione (per ulteriori dettagli si indirizza a [24]). Innanzitutto viene definita una grandezza *ABS*, indice dell'assorbimento della radiazione infrarossa nel sensore, nel seguente modo:

$$ABS = 1 - \frac{I}{I_0} \quad (3.1)$$

dove I è il rapporto tra le due grandezze *ACT* e *REF* prima introdotte e rese disponibili dal *driver*, mentre I_0 è il rapporto tra le stesse eseguito in fase di calibrazione. Viene ora riportata la relazione che fornisce il valore della concentrazione percentuale di CO₂:

$$x = \left[-\frac{\ln\left(1 - \frac{ABS}{SPAN}\right)}{b} \right]^{\frac{1}{c}} \quad (3.2)$$

SPAN è un altro parametro che può essere calcolato in fase di calibrazione, mentre b e c sono due parametri il cui valore è tabulato nella Tabella 1 consultabile in [24]. Altrimenti, i due parametri possono essere estrapolati attraverso regressione sulle curve Absorbance vs. CO₂ Volume così come viene descritto nel documento [22]. Effetti non lineari e variazioni della misura dovute alla temperatura che si possono avere, tipici per questo genere di sensori, e che vengono accennati secondo quanto descritto in [24] e [22], non sono stati presi in considerazione. La ragione di tale scelta sarà chiarita nel seguente capitolo 4. Ci limitiamo a dire che il grado di precisione otte-

nibile senza operare alcuna compensazione è sufficientemente elevato per le applicazioni a cui è destinata la lettura.

Il sensore necessita di un circuito di adattamento, in particolare di un filtro passa banda tra i 2 e 3 Hz, che comprende la frequenza del segnale con la quale viene pilotato l'emettitore. Uno schema pratico, che effettivamente viene riprodotto nella scheda ausiliaria su cui viene montato il trasduttore può essere reperito in [22]. A causa di problemi tecnici, la scheda ausiliaria, sulla quale alloggia il trasduttore e il relativo circuito adattativo, si è rivelata essere inutilizzabile. Di conseguenza abbiamo dovuto riprodurre lo stesso schema su *breadboard*.

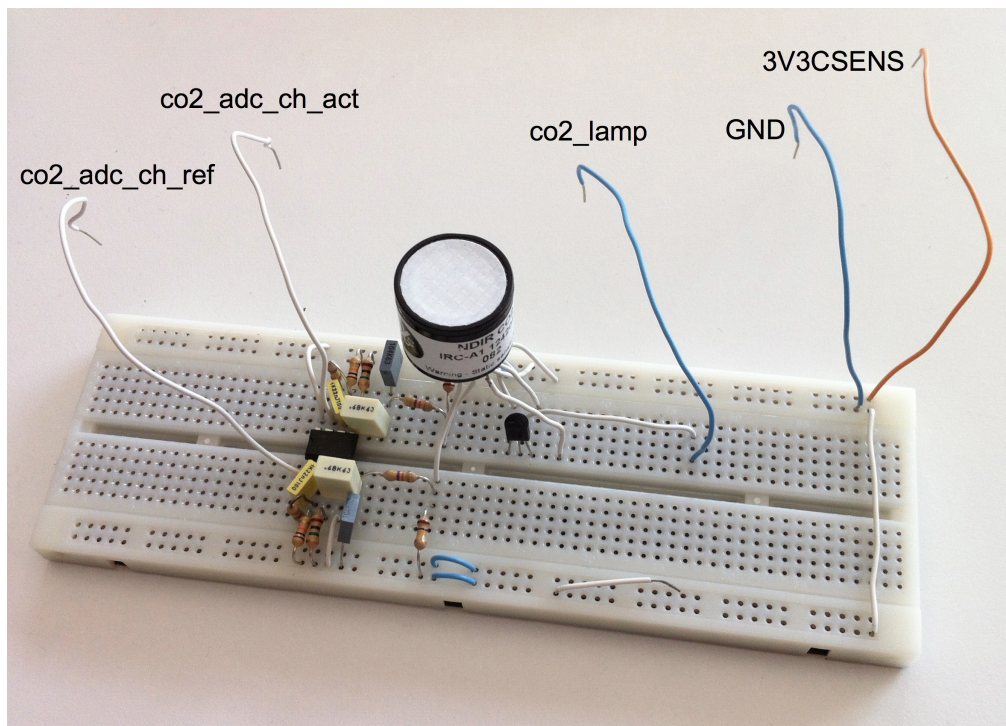


Figura 3.3: Il sensore Alphasese IRC-A1 e il relativo circuito di adattamento riprodotto su *breadboard*

3.2 Implementazione del *driver*

Come anticipato, il *driver* è stato realizzato in modo tale da inserirsi nel già presente sistema sviluppato da Patavina Technologies srl. Per quanto detto in precedenza, risulta chiara la necessità di utilizzare un campionamento ADC per i segnali di *ACT* e *REF*, nonché di un temporizzatore abbastanza preciso, al fine di generare l'onda rettangolare utile a pilotare l'emettitore infrarosso del sensore. La gestione di questi servizi è già presente nel sistema: è stato sufficiente istanziare gli oggetti utili a svolgere un campionamento ADC, e impostare i vari *interrupt* necessari a temporizzare il campionamento e lo *switch* on/off di una porta del microcontrollore, così da generare l'onda rettangolare. Rispettando il paradigma della programmazione a oggetti, con la quale è sviluppato l'intero sistema *software*, è stata creata una classe *co2_t*, che rende disponibili i metodi per la lettura e il *power management*. La gestione di quest'ultimo è stata comunque implementata nonostante la versione sperimentale di EUTECH2 a nostra disposizione, inibisca il funzionamento del *transistor* NMOS, necessario per svolgere il *power management*, attraverso un cortocircuito. La classe *co2_t* si avvale di una classe ausiliaria creata *ad hoc*, di nome *measure_queue* che si occupa di memorizzare i dati relativi al campionamento, in particolare, implementando un *array* statico che viene riempito ciclicamente; inoltre rende disponibili i metodi utili a ispezionare tali dati, come ad esempio la ricerca del massimo e del minimo valore nell'*array*: questi permettono di completare la raccolta dei dati utili per la misura del valore di CO₂. Durante la scrittura del *driver*, si è anche pensato a un metodo alternativo per la determinazione del massimo e minimo valore dell'ampiezza dei due segnali *ACT* e *REF*. L'implementazione di un *trigger* via *software* è una valida soluzione che oltretutto risulta essere molto meno dispendiosa dal punto di vista di occupazione della memoria rispetto a quella da noi adottata. È stata comunque constatata la presenza di disturbi legati alla misura, che di fatto rendono particolarmente complessa la determinazione del fronte di salita e discesa dei due segnali. Soluzione a questo problema sarebbe l'implementazione di un filtro digitale a modello Auto Regressive Moving Average (ARMA, in italiano Auto Regressivo a Media Mobile). Benché

non ci siano particolari problemi realizzativi la soluzione è stata scartata, in quanto eccessivamente complessa e dal punto di vista computazionale.

3.2.1 La classe *co2_read*

Viene ora analizzato il file di *header* della classe *co2_t*

```
#define LENGTH 80    //50*20ms=1s, two periods in the entire
                    //vector

class co2_t: private power_manageable_if_t{
private:
    power_manager_t pwrmgr;
    power_manager_t *externalpwemgr;
    adc_channel_if_t *adc_ch_act;
    adc_channel_if_t *adc_ch_ref;
    gpio_if_t *lamp;
    timer_alarm_t alarm_measure;
    timer_alarm_t alarm_lamp;    //timer to pilotate the lamp
                                //driver
    measure_queue measures;
private:
    // power_manageable_if_t methods
    virtual void on();
    virtual void off();
public:
    co2_t(power_manager_t *pwmcontroller);
    void init(adc_channel_if_t *adc_ch_act,
              adc_channel_if_t *adc_ch_ref,
              adc_channel_if_t *adc_ch_th, gpio_if_t *enable);
    measure read();
    static void lamp_task(timer_alarm_t* alarm);
    static void measure_task(timer_alarm_t* alarm);
};
```


3. Attività svolta presso Patavina Technologies srl

measures è un oggetto di classe *measure_queue* che semplicemente contiene tutti i dati relativi al campionamento. I dettagli sono trattati nella seguente sezione 3.2.2.

LENGTH è una costante che fissa la dimensione dell'*array* contenente i dati campionati. Ovviamente deve avere un valore tale da permettere la memorizzazione dei campioni di almeno un periodo (con un periodo di campionamento di 5 ms, per campionare il periodo di un segnale con frequenza di 2.5 Hz è necessario un valore di *LENGTH* di almeno 80 unità). I metodi *on()* e *off()* sono destinati alla gestione del *power management*. Di fatto, abilitano e disabilitano il sensore controllandone l'alimentazione. Va comunque ricordato che la scheda EUTECH2 inibisce questa funzione a causa del sopracitato problema riguardante la cortocircuitazione sul *transistor*.

Il metodo *co2_t(power_manager_t *pwmcontroller)* è il costruttore della classe. Viene utilizzato nel file *board.cpp* per inizializzare l'oggetto *co2r* (di classe *co2_t*) che realizza il *driver*. Il metodo *init(adc_channel_if_t *adc_ch_act, adc_channel_if_t *adc_ch_ref, adc_channel_if_t *adc_ch_th, gpio_if_t *enable)* inizializza l'oggetto della classe *co2_t*. Viene applicato a *co2r* nel file *board.cpp*. Semplicemente va a creare i riferimenti alle porte GPIO (I/O), ADC che verranno utilizzate dall'oggetto di classe *co2_t* per completare la misura di CO₂.

Il metodo *lamp_task(timer_alarm_t* alarm)* è una *callback* che viene passata come argomento nell'inizializzazione di un *timer* settato a 200 ms. In particolare, nel metodo, viene acceso/spento, a ogni visita, il led che si occupa del pilotaggio dell'emettitore infrarosso del sensore. Si noti che spegnere e accendere ogni 200 ms implica un segnale d'uscita rettangolare con *duty cycle* del 50% e frequenza di 2.5 Hz, che rientra nei parametri di funzionamento del sensore.

Il metodo *measure_task(timer_alarm_t* alarm)* è una *callback* che viene passata come argomento nell'inizializzazione di un *timer* settato a 5 ms. All'interno del metodo si esegue la lettura vera e propria dell'ADC, sia del segnale di *REF* che di *ACT*. I due dati vengono poi inseriti nell'oggetto *measures*, precedentemente descritto, attraverso il relativo metodo *enqueue(uint16_t act, uint16_t ref)* (vedi sezione 3.2.2).

Il metodo *read()* permette la lettura del valore di ampiezza dei due segnali *ACT* e *REF*. Si avvale della variabile, interna alla classe, *measures* a cui viene applicato il metodo *get_measure()*. Il metodo restituisce così un valore di tipo *measure* in cui è contenuta l'ampiezza dei due noti segnali.

3.2.2 La classe *measure_queue*

Segue il file *header* della classe *measure_queue*.

```
struct measure{
    uint16_t act;
    uint16_t ref;
};
class measure_queue{
private:
    measure *measures;
    int length;
    int end;
private:
    int findmin();
    int findmax();
public:
    measure_queue();
    measure_queue(int length);
    void enqueue(uint16_t act, uint16_t ref);
    measure getmeasure();
};
```

È stata creata una struttura *measure* che contenga sia il dato relativo al segnale di *ACT* che quello di *REF*. La classe *measure_queue* si occupa di creare un'array di valori *measure* che contenga tutti i dati dei campionamenti. Questo array viene riempito ciclicamente, di fatto sovrascrivendo eventuali *entry* salvate. È per questo motivo che l'array deve avere una dimensione sufficientemente grande: deve infatti contenere abbastanza campioni da rappresentare un periodo del segnale di *ACT/REF*. La dimensione dell'array

3. Attività svolta presso Patavina Technologies srl

può essere arbitraria e viene scelta nel momento della creazione dell'oggetto di classe *measure_queue* attraverso il costruttore a cui viene passato il parametro *length*. All'interno della classe *co2_t* viene principalmente impiegato il metodo *enqueue(uint16_t act, uint16_t ref)* per salvare i valori misurati, e il metodo *getmeasure()* che fornisce un valore di tipo *measure* contenente l'ampiezza dei segnali *ACT* e *REF*, ottenuta attraverso i metodi *findmin()* e *findmax()* e una successiva operazione di differenza.

4 | Risultati numerici

La raccolta dei dati campionati avviene attraverso il riempimento ciclico di un *array* statico (sezione 3.2.2). Segue un'immagine dei dati raccolti durante un test del funzionamento del *driver*. La ricerca del valore massimo e minimo

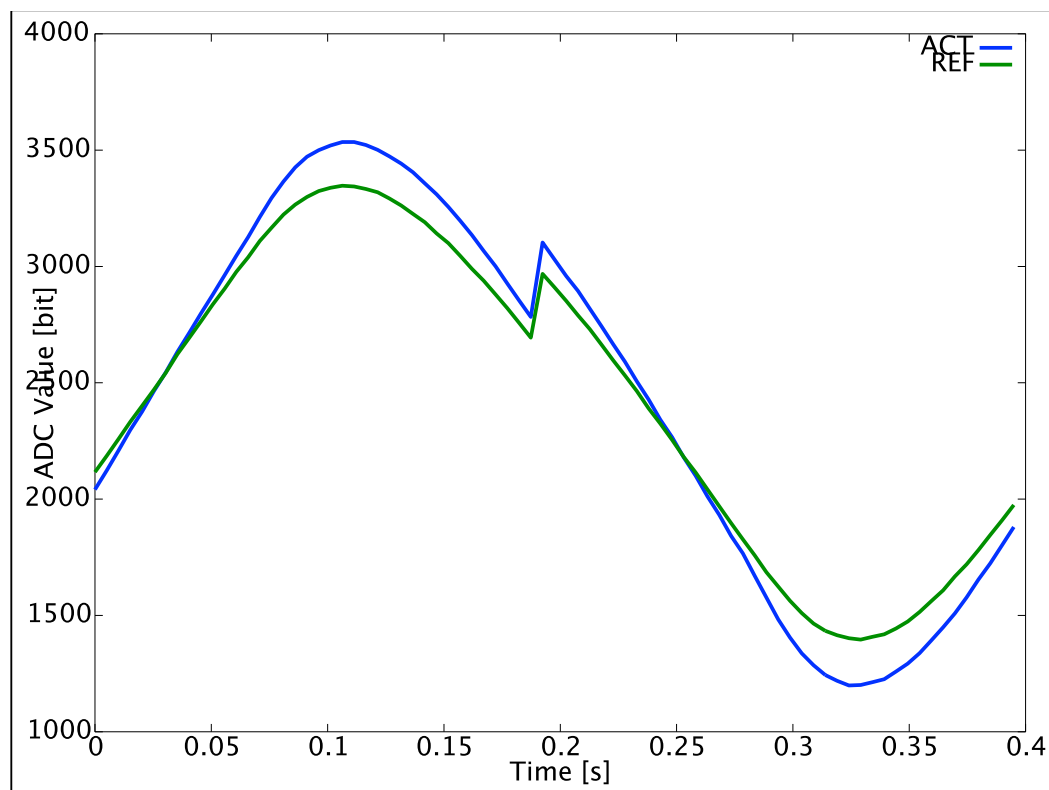


Figura 4.1: Risultato di un campionamento. Si osservi come l'ampiezza delle due sinusoidi sia diversa. Si può anche osservare la presenza di una discontinuità dovuta al riempimento ciclico dell'array statico. Il valore in ordinata viene letto dal registro della misura dell'ADC, la cui dimensione è impostata a 12 bit, il che implica un valore massimo di $2^{12} = 4096$.

di entrambi i segnali all'interno dell'*array* completa il *driver*, in quanto la loro differenza rende disponibile il dato I .

Durante alcune prove effettuate sono emersi fenomeni di saturazione in uscita all'amplificatore operativo del circuito di adattamento. La modifica di una resistenza dell'amplificatore combinata a un aumento della frequenza del segnale utile a controllare l'emettitore (in questo modo viene modificato il guadagno del circuito di adattamento, che presenta una caratteristica passa banda piuttosto stretta), è stata sufficiente per risolvere il problema.

Il sensore di CO₂ a tecnologia NDIR risulta essere sensibile alle variazioni della temperatura. Le relazioni che modellano l'influenza della temperatura sulla misura sono reperibili in [23]. In particolare, la funzione x introdotta nell'equazione (3.2) sarà funzione della temperatura. Quindi, la concentrazione x sarà denotata in seguito come x_T per evidenziare questa dipendenza. Quest'ultima viene modellata modificando l'equazione (3.1) nel seguente modo:

$$ABS = \left(1 - \frac{I}{I_0}\right) \cdot (1 + \alpha \cdot (T - T_{cal})) \quad (4.1)$$

Ne è stato studiato l'effetto attraverso uno *script octave* dove abbiamo assunto $T_{cal} = 25 \text{ }^\circ\text{C}$ e $\alpha = 0.0007$, secondo quanto suggerito da [23]. Ne è emerso che per il sensore utilizzato, funzionante in un *range* di concentrazione di CO₂ da 0% a 5%, l'effetto della temperatura non è così marcato. Di conseguenza, la compensazione degli effetti legati alla temperatura, che può essere eseguita attraverso misurazione della temperatura del sensore valutando l'uscita TH dello stesso, non è stata implementata. Si riporta di seguito il grafico di $x_T(I/I_0)$, risultante dallo studio del fenomeno attraverso la simulazione con *octave*.

Dal grafico della funzione $x_T(I/I_0)$, si evidenzia il seguente comportamento: all'aumentare della concentrazione di CO₂ il rapporto $I = ACT/REF$ deve calare (I_0 , come descritto nella sezione 3.1) infatti è una costante positiva determinata in fase di calibrazione). Ulteriore conferma riguardo la lieve variazione nella misura, a causa della variazione della temperatura, è visibile in figura 4.3. Attraverso un altro *script octave* è stata simulata la funzione x_T variando la temperatura e mantenendo costante il valore dell'assorbimen-

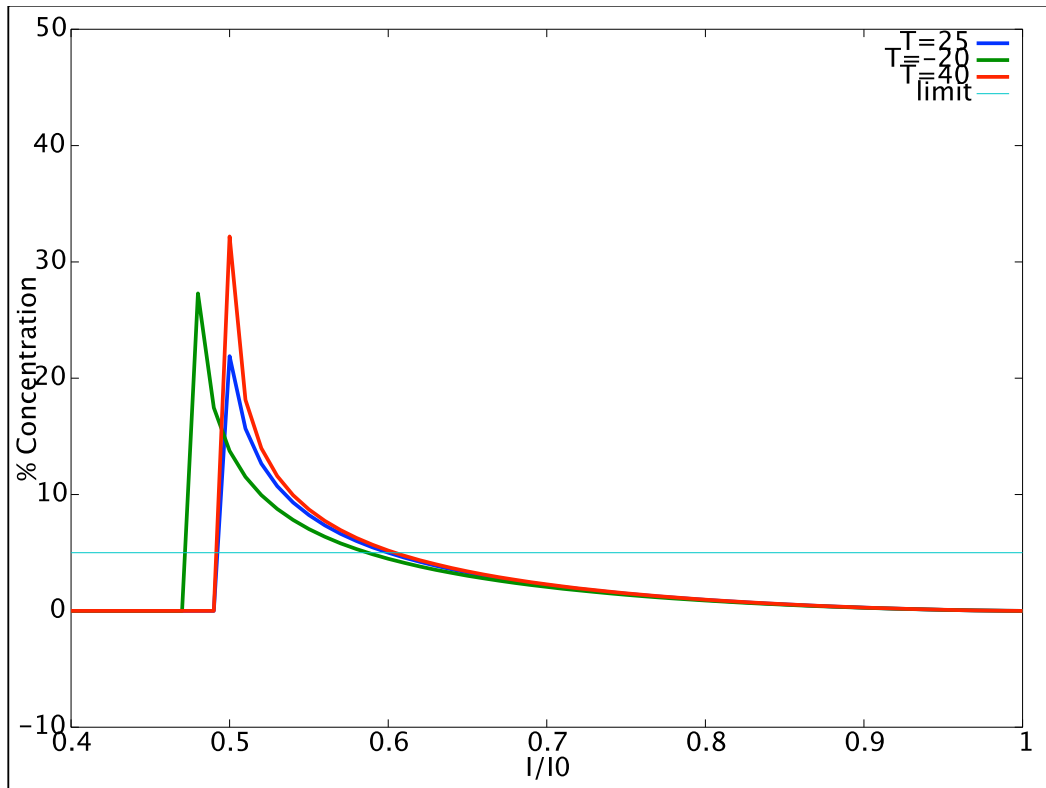


Figura 4.2: Caratteristica CO_2 % Concentration – ABS.

to *ABS* per tre valori, 0.1, 0.3 e 0.4. In particolare si evidenzia il seguente comportamento: l'errore percentuale cresce all'aumentare della concentrazione del gas ed è direttamente proporzionale alla temperatura T . Si può notare come, a temperature di funzionamento proibitive, l'errore percentuale rimanga confinato al 10%.

Per ottenere un risultato numerico preciso della concentrazione di CO_2 servirebbe una fase iniziale di calibrazione, che esula dagli scopi di questo lavoro. Tale fase, che sarebbe utile a determinare parametri come il valore I_0 di riferimento, richiederebbe una serie di misure in ambiente con concentrazione del gas nota, attraverso apparecchiature complesse non disponibili per l'attività di questa tesi. Tuttavia, siamo riusciti a constatare sperimentalmente che il comportamento descritto dalla funzione $x_T(I/I_0)$ viene rispettato.

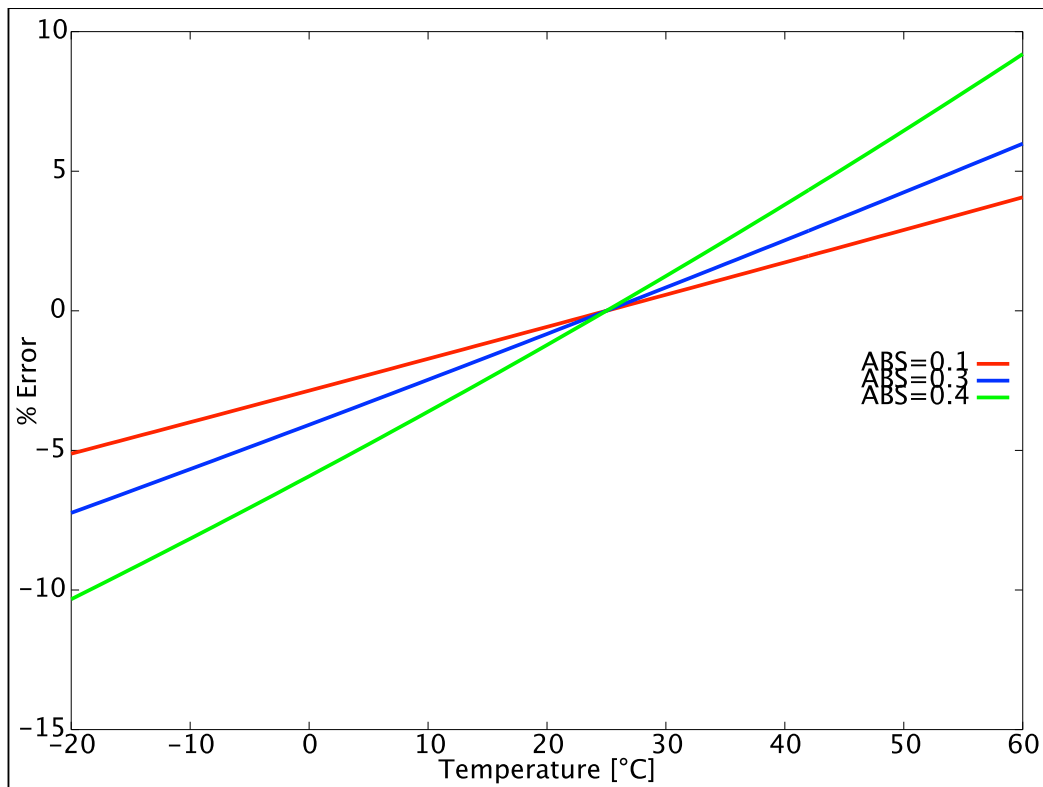


Figura 4.3: Errore percentuale dovuto alla variazione della temperatura per tre valori di ABS . Si noti che, secondo l'equazione (4.1), a un valore di $ABS = 0.4$ corrisponde un rapporto I/I_0 pari a 0.6 che, secondo quanto si può ricavare dall'immagine 4.2, implica una concentrazione di CO_2 del 5% circa.

5 | Sviluppi Futuri

Il funzionamento del *driver* è stato verificato e le variazioni nella concentrazione del gas sono certamente rilevabili. Ciò che manca è la possibilità di valutare numericamente il valore della concentrazione del gas. Una possibile estensione, atta allo scopo, è la seguente: si può pensare di abilitare la connettività *wireless* della *board*, realizzando così un servizio Web di tipo REST che renda accessibili i due dati significativi della misura, *ACT* e *REF*. Uno *script* presente nel terminale che ha accesso ai dati potrà così ultimare il calcolo per valutare la concentrazione del gas.

In ogni caso, questa mancanza non risulta così limitante. Un'applicazione particolarmente semplice che non richiede la conoscenza del valore preciso della concentrazione di CO₂ può essere quella di un allarme che segnala valori particolarmente elevati del gas. La naturale concentrazione ambientale di CO₂ è del 0.03%; un valore del 3%, per brevi periodi di tempo, comporta mal di testa, palpitazioni, aumento della frequenza respiratoria, mentre può risultare mortale se l'esposizione è prolungata oltre la mezz'ora [30]. E così, dopo una prima calibrazione per stabilire quale sia il rapporto I/I_0 corrispondente a tale concentrazione, si può pensare di utilizzare il sensore con il *driver* realizzato per segnalare tale situazione critica.

Uno sviluppo futuro del *driver* prevede lo svolgimento della calibrazione, utile a estrapolare i parametri descritti nella sezione 3.1. L'operazione prevede di valutare la caratteristica Absorbance vs. CO₂ Volume. A questo scopo è necessario valutare il rapporto I/I_0 a determinate concentrazioni di CO₂. Ciò implica la necessità di avere a disposizione un ambiente dove la concentrazione del gas sia nota.

Nel capitolo 4 si è descritto come la temperatura possa influenzare la

misurazione della concentrazione. Si è anche visto che, per applicazioni che prevedono misure di concentrazione inferiore al 5%, tali effetti non sono particolarmente rilevanti. Qualora sia necessaria una maggiore precisione della misura, gli effetti vanno presi in considerazione. Per valutarli è necessario conoscere la temperatura del sensore che può essere calcolata attraverso il *pin TH*, così come viene descritto in [23]. Nello specifico, al *pin TH* è collegato il terminale di un termistore interno al sensore e connesso a massa. Valutando, attraverso il microcontrollore e il modulo ADC, la tensione di un partitore resistivo, ottenuto ponendo in serie al termistore una resistore dal valore noto, è possibile determinare il valore della resistenza del termistore e quindi la temperatura, secondo quanto tabulato in [23].

Il *driver* può essere integrato in applicazioni di monitoraggio dei parametri ambientali. Nel paragrafo 3.1 si è infatti parlato dei vari sensori disponibili su EUTECH2, come, ad esempio, il sensore di luminosità e l'igrometro. A questo punto, si pensi alla costituzione di una rete di questi sensori, tutti in grado di comunicare l'uno con l'altro. Lo scenario è molto simile a quello descritto nella sezione 2.3 della rete costituita presso il Dipartimento di Ingegneria dell'Informazione a Padova, ma si può immaginare applicato ovunque: in una serra, al fine di migliorare la resa delle piantagioni; in aule ospedaliere, scolastiche o d'ufficio per migliorare le condizioni dell'aria, magari attraverso un impianto condizionatore e ventilatore, a sua volta, connesso alla rete; negli impianti di produzione dove è richiesta una concentrazione particolare del gas (vedi l'esempio nella sezione 2.4).

Con queste premesse, si può pensare di estendere la rete su larga scala: si immagini a una rete costituita da questi sensori distribuiti in diverse zone della città. Sarebbe così possibile raccogliere informazioni utili riguardanti qualsiasi luogo cittadino. Ciò apre a diverse possibili applicazioni. Si pensi, ad esempio, a una amministrazione che vuole ridurre l'inquinamento cittadino. Attraverso la mappatura della CO₂ potrebbe selettivamente limitare il traffico degli automezzi nelle strade che presentano livelli critici del gas. O ancora, si immagini alla seguente applicazione. Il *jogging* è un'attività sportiva che può essere praticata ovunque, anche in città, ma, ovviamente, sarebbe preferibile praticarlo in presenza di aria pulita: un'applicazione per

smartphone che si appoggia sulla rete succitata può calcolare un percorso con i minori livelli di CO₂ possibili e garantire una corsa più salubre. Sono solo alcuni semplici esempi di applicazione delle WSN, o più in generale di IoT, ma danno comunque prova dell'efficacia e delle potenzialità di questa tecnologia emergente.

Bibliografia

- [1] Marcelo Alonso e Edward Finn. *Physics*. 1970.
- [2] Kevin Ashton. “That ‘Internet of things’ thing. In the real world, things matter more than ideas.” In: *RFID Journal* (2009). URL: <http://www.rfidjournal.com/articles/view?4986>.
- [3] *ATxMEGA256A3 Datasheet doc8068*. URL: <http://www.atmel.com/Images/doc8068.pdf>.
- [4] Luigi Atzori, Antonio Iera e Giacomo Morabito. “The Internet of Things: A survey”. In: *Computer Networks* 54.15 (2010), pp. 2787–2805. ISSN: 1389-1286. DOI: <http://dx.doi.org/10.1016/j.comnet.2010.05.010>. URL: <http://www.sciencedirect.com/science/article/pii/S1389128610001568>.
- [5] Nevio Benvenuto e Michele Zorzi. *Principles of Communications Networks and Systems*. Wiley, 2011, p. 738. ISBN: 978-0-470-74431-4.
- [6] Nicola Bui e Michele Zorzi. “Health care applications: a solution based on the internet of things”. In: ISABEL ’11 (2011), 131:1–131:5. DOI: 10.1145/2093698.2093829. URL: <http://doi.acm.org/10.1145/2093698.2093829>.
- [7] Working Group on California Earthquake Probabilities. “Earthquake Probabilities in the San Francisco Bay Region: 2000 to 2030 - A Summary of Findings”. In: (1999). Open File Report 99-517, Online Version 1.0. URL: <http://geopubs.wr.usgs.gov/open-file/of99-517/of99-517.pdf>.

- [8] Paolo Casari et al. “The “Wireless Sensor Networks for City-Wide Ambient Intelligence (WISE-WAI)” Project”. In: *Sensors* (2009).
- [9] A.P. Castellani et al. “Architecture and protocols for the Internet of Things: A case study”. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*. 2010, pp. 678–683. DOI: 10.1109/PERCOMW.2010.5470520.
- [10] Michael Chorost. “The Networked Pill”. In: (March 20 2008). MIT Technology Review. URL: <http://www.technologyreview.com/biomedicine/20434/?a=f>.
- [11] Diane Cook e Sajal Das. *Smart Environments: Technology, Protocols and Applications (Wiley Series on Parallel and Distributed Computing)*. Wiley-Interscience, 2004. ISBN: 0471544485.
- [12] A. Earnshaw e Norman Greenwood. “Chemistry of the Elements, Second Edition”. In: Butterworth-Heinemann, 1997, p. 310.
- [13] Dave Evans. “How the Next Evolution of the Internet Is Changing Everything”. In: (April 2011).
- [14] “Forrester CEO forecasts web services storm”. In: (2003). URL: <http://www.computerweekly.com/news/2240049850/Forrester-CEO-forecasts-web-services-storm>.
- [15] Zhang Guo-Qing et al. “Evolution of the Internet and its Cores”. In: *New Journal of Physics* 10 123027 (2008).
- [16] *IRC-A1 CARBON DIOXIDE INFRARED SENSOR*. URL: http://www.alphasense.com/industrial-sensors/pdf/NDIR_carbon_dioxide_IR.pdf.
- [17] Duncan Jefferies. “How the ‘internet of things’ could radically change local government”. In: (2011). URL: http://www.theguardian.com/local-government-network/2011/aug/18/internet-of-things-local-government?utm_source=Newsletter%3A+Week+1+%28Sensors%29&utm_campaign=Newsletter+1&utm_medium=archive.

BIBLIOGRAFIA

- [18] *Kevin Ashton the internet of things and other things*. URL: <http://kevinjashton.com/about/>.
- [19] Rahat Ali Khan et al. “Wireless Sensor Networks: A Solution for Smart Transportation”. In: *Journal of Emerging Trends in Computing and Information Sciences* (April, 2012).
- [20] M. Leuenberger et al. “CO₂ concentration measurements on air samples by mass spectrometry”. In: *Rapid Communications in Mass Spectrometry* 14 (2000).
- [21] Geoff Mulligan. “The 6LoWPAN architecture”. In: *Proceedings of the 4th workshop on Embedded networked sensors*. EmNets '07. Cork, Ireland: ACM, 2007, pp. 78–82. ISBN: 978-1-59593-694-3. DOI: 10.1145/1278972.1278992. URL: <http://doi.acm.org/10.1145/1278972.1278992>.
- [22] *NDIR: Determination of Linearisation and Temperature Correction Coefficients*. URL: http://www.alphasense.com/industrial-sensors/pdf/NDIR/AAN_203-04.pdf.
- [23] *NDIR: Electronic Interface*. URL: http://www.alphasense.com/industrial-sensors/pdf/NDIR/AAN_202-03.pdf.
- [24] *NDIR Gas Concentration Calculation Overview*. URL: http://www.alphasense.com/industrial-sensors/pdf/NDIR/AAN_201-05.pdf.
- [25] S. Neethirajan, D. S. Jayas e S. Sadistap. “Carbon Dioxide (CO₂) Sensors for the Agri-food Industry—A Review”. In: *Springer* (2008).
- [26] Donald L. Pavia et al. *Introduction to Organic Laboratory Techniques*. Thomson Learning, 2006, pp. 36–37.
- [27] P.C.Jain, Arti Noor e Vinod Kumar Sharma. “Internet of Things – An Introduction”. In: (2011). 4th Annual Seminar on C-DAC Noida technologies (ASCNT-2011).
- [28] M. Presser et al. “The SENSEI project: integrating the physical world with the digital world of the network of the future”. In: *Communications Magazine, IEEE* 47.4 (2009), pp. 1–4. ISSN: 0163-6804. DOI: 10.1109/MCOM.2009.4907403.

- [29] C.S. Raghavendra, Krishna M. Sivalingam e Taieb Znati. *Wireless Sensor Networks*. Springer, 2004.
- [30] Susan A. Rice. *Health Effects of Acute and Prolonged CO₂ Exposure in Normal and Sensitive Populations*. URL: <http://seca.doe.gov/publications/proceedings/03/carbon-seq/PDFs/173.pdf>.
- [31] M. Rossi et al. “SYNAPSE++: Code Dissemination in Wireless Sensor Networks Using Fountain Codes”. In: *Mobile Computing, IEEE Transactions on* 9.12 (2010), pp. 1749–1765. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.109.
- [32] Johnny Ryan. *A History of the Internet and the Digital Future*. Reaktion Books, 2010. ISBN: 978 1 86189 777 0.
- [33] *SENSEI: Integrating the physical with the digital world of the network of the future*. URL: <http://www.ict-sensei.org/>.
- [34] Yi S.H., Park Y.W. e Han S.O. “Novel NDIR CO₂ sensor for indoor Air quality monitoring”. In: (2005). The 13th International Conference on Solid-State Sensors, Actuators and Microsystems.”.
- [35] Z. Shelby e C. Borman. *6LoWPAN: The Wireless Embedded Internet*. Wiley, 2009.
- [36] Bjarne Stroustrup. *The C++ Programming Language*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000. ISBN: 0201700735.
- [37] Zheng Sun et al. “A Routing Protocol Based on Flooding and AODV in the ZigBee Network”. In: *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*. 2009, pp. 1–4. DOI: 10.1109/IWISA.2009.5072672.
- [38] *The Internet of Things*. URL: <http://share.cisco.com/internet-of-things.html>.
- [39] E. Toffoli et al. “Thermodynamic Identification of Buildings using Wireless Sensor Networks”. In: (2007).

BIBLIOGRAFIA

- [40] Yanfeng Wang et al. “The NDIR CO₂ Monitor With Smart Interface for Global Networking”. In: *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 54, NO. 4* (2005).
- [41] *WISE-WAI project web site*. URL: <http://cariparo.dei.unipd.it>.
- [42] Jacob Y. Wong. “NDIR Gas Sensor.” Patent US5444249 (US). 1995.
- [43] Jacob Y. Wong. “Ultra low power NDIR carbon dioxide sensor fire detector”. Patent US7214939 B1 (US). 2007.
- [44] *World Internet Stats: Usage and Population Statistics*. June 30, 2010. URL: <http://www.internetworldstats.com/stats.htm>.
- [45] *World Population*. URL: http://www.census.gov/population/international/data/worldpop/table_population.php.
- [46] Tomomi Yasuda, Seiichiro Yonemura e Akira Tani. “Comparison of the Characteristics of Small Commercial NDIR CO₂ Sensor Models and Development of a Portable CO₂ Measurement Device”. In: *Sensors* (2012).
- [47] *ZigBee Alliance*. URL: <http://www.zigbee.org/About/AboutAlliance/TheAlliance.aspx>.
- [48] *ZigBee Specification Overview*. URL: <http://www.zigbee.org/Specifications/ZigBee/Overview.aspx>.