



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

**“STUDIO E REALIZZAZIONE DI UN NODO SENSORE PER LA
MISURAZIONE DI PESO PER L'ANALISI DEL TRASPORTO DEI
SEDIMENTI SULLE DUNE COSTIERE”**

Relatore: Prof. Alessandro Pozzebon

Laureando: Cristian Lorenzi

ANNO ACCADEMICO 2021 – 2022

Data di laurea 23/09/2022

Al nonno Agostino

STUDIO E REALIZZAZIONE DI UN NODO SENSORE PER LA MISURAZIONE DI PESO PER L'ANALISI DEL TRASPORTO DEI SEDIMENTI SULLE DUNE COSTIERE

Abstract

Il lavoro di tesi si pone l'obiettivo di progettare e realizzare un sensore per la misurazione del peso, con elevato livello di precisione, necessario per il monitoraggio del fenomeno di erosione delle dune costiere. Lo studio del trasporto dei sedimenti avviene grazie ad una "trappola per la sabbia", una struttura già realizzata dal Dipartimento di Scienze della Terra dell'Università di Pisa. Questo oggetto è un cilindro semiaperto che permette di raccogliere continuamente al suo interno la sabbia trasportata dal vento, grazie ad una banderuola sulla sua sommità che orienta l'apertura in funzione della direzione del vento stesso. Per stabilire la quantità di materiale al suo interno, viene misurato il peso tramite il sensore. Per l'acquisizione dei parametri il microcontrollore utilizzato è l'ATtiny84, programmabile tramite piattaforma Arduino. Viene utilizzata una cella di carico per misurare la forza applicata sul piatto inferiore della trappola: questa è un trasduttore che sfrutta la deformazione meccanica che il peso esercita sul componente per restituire in uscita una tensione elettrica. Infine, viene integrato un sistema di comunicazione LoRa-WAN per la trasmissione dei dati. Poiché il segnale elettrico differenziale prodotto dalla cella è nell'ordine dei mV, sarà necessario un sistema di amplificazione del segnale. Il lavoro di tesi parte dalla scelta della cella di carico e del sistema di amplificazione ritenuti più adatti, in modo da ridurre al minimo l'errore e massi-

mizzarne la precisione. Successivamente verrà programmato il microcontrollore e implementato fisicamente il nodo sensore nella trappola. Si procederà da ultimo con le misurazioni di prova, per verificare l'efficacia e la precisione del dispositivo realizzato.

Indice

1	La struttura	14
1.1	Problema delle dune costiere	14
1.2	Analisi del trasporto	15
1.3	Descrizione struttura sperimentale	15
1.3.1	Struttura meccanica	15
1.3.2	Struttura di rilevamento	16
2	I microcontrollori	18
2.1	Arduino	18
2.1.1	Hardware	19
2.1.2	Software	19
2.1.3	Arduino UNO	20
2.2	Microcontrollori ATtiny	20
2.2.1	Programmare ATtiny tramite Arduino	21
3	Il sensore	23
3.1	Celle di carico	23
3.1.1	Cella di carico pneumatica	23
3.1.2	Cella di carico idraulica	24
3.1.3	Cella di carico capacitiva	24
3.2	Cella di carico ad Estensimetro	25
3.2.1	Estensimetro	25
3.2.2	Ponte di Wheatstone	26
3.2.3	Cella di carico scelta	27
3.3	Amplificatore HX711	28

3.3.1	Libreria e comandi utilizzati	28
4	La trasmissione	30
4.1	Tecnologia di trasmissione LoRa e LoRaWAN	30
4.2	Modulo RFM95	31
5	Il progetto sviluppato	32
5.1	Calibrazione e taratura con Arduino UNO	32
5.1.1	Misurazioni di prova	35
5.2	Implementazione con ATtiny84	37
5.2.1	Test dello strumento	39
5.3	Integrazione del modulo RFM95	39
5.3.1	Test dello strumento con struttura completa	42

Elenco delle figure

1.1	Modello 3D e prototipo della struttura meccanica.	16
1.2	Modello 3D e prototipo della struttura di rilevamento.	17
2.1	Microcontrollori utilizzati.	20
2.2	Pinout ATtiny84.	21
2.3	Connessioni per caricare uno sketch su ATtiny84.	22
3.1	Cella di carico pneumatica.	24
3.2	Cella di carico idraulica.	24
3.3	Cella di carico capacitiva.	25
3.4	Cella di carico ad estensimetro.	26
3.5	Estensimetro e ponte di Wheatstone.	27
3.6	Cella di carico scelta.	28
3.7	Amplificatore HX711	29
4.1	Modulo di trasmissione utilizzato.	31
5.1	Connessioni per calibrazione e taratura.	34
5.2	Misure ottenute.	37
5.3	Connessioni per implementazione su ATtiny84.	39
5.4	Test di precisione.	40
5.5	Struttura completa della trappola assemblata.	42
5.6	Test di precisione della struttura completa.	43

Elenco delle tabelle

2.1	Connessioni per caricare uno sketch su ATtiny84.	22
5.1	Connessioni per calibrazione e taratura.	33
5.2	Connessioni per implementazione su ATtiny84.	40
5.3	Connessioni per implementazione del modulo RFM95.	41

Elenco degli sketch

5.1	Calibrazione e taratura.	33
5.2	Misurazioni di prova.	36
5.3	Implementazione su ATtiny84.	38

Introduzione

Nel **capitolo 1** si presenta brevemente il problema dell'erosione delle dune costiere, esponendone le cause e le possibili soluzioni. È descritta inoltre la struttura fisica della trappola per la sabbia, nelle sue due parti: meccanica ed elettrica.

Nei capitoli seguenti si tratta dei vari componenti utilizzati nel progetto, descrivendone le caratteristiche e la motivazione per cui sono stati scelti. In particolare nel **capitolo 2** vengono affrontati i microcontrollori utili alla realizzazione della tesi, ovvero ATmega328P e ATTiny84.

Nel **capitolo 3** si offre una panoramica generale sui vari tipi di cella di carico, il sensore scelto e il relativo amplificatore HX711.

Nel **capitolo 4** si tratta la tecnologia di comunicazione LoRaWAN e, in particolare, il modulo di trasmissione RFM95 utilizzato per l'invio dei dati rilevati.

Nel **capitolo 5** si ripercorre il lavoro pratico affrontato per la realizzazione del progetto, andando nel dettaglio su come sono state fatte le connessioni, sugli sketch utilizzati e sui test effettuati per verificare l'efficacia e la precisione dello strumento.

Capitolo 1

La struttura

1.1 Problema delle dune costiere

Il trasporto dei sedimenti è una branca della sedimentologia, che studia come questi sono trascinati, distribuiti e depositati. In particolare, in una zona costiera definisce come vengono spostati i granelli di sabbia, come si muovono e quanto materiale è trasportato. Le cause di questo fenomeno possono essere varie, come ad esempio le onde, le maree, l'immissione di fiumi nel mare, il vento. I parametri del trasporto di sabbia sono molto importanti per l'ambiente costiero poiché l'evoluzione del litorale si basa sull'equilibrio tra i sedimenti che abbandonano il sistema e quelli che invece ne entrano a far parte. Nella maggior parte dei casi la degradazione delle zone costiere è dovuta a correnti e onde, che contribuiscono a spostare i granelli dalla spiaggia al fondale marino lasciando questo bilancio in negativo. Ciò porta ad un fenomeno erosivo in quanto non viene immesso nel sistema alcun materiale per compensare a questa perdita. Data l'importanza delle zone costiere, non solo per ragioni ambientali (vegetazione e fauna), ma anche per ragioni economiche (attività commerciali e turismo), si stanno cercando delle soluzioni per contrastare la degradazione delle spiagge. La pratica migliore è considerata la redistribuzione della sabbia, piuttosto che metodi di protezione più invasivi come barriere e frangiflutti. Per operare una redistribuzione è però necessaria una comprensione completa dei parametri del trasporto dei sedimenti, in modo da capire la quantità di materiale richiesto e le zone interessate.

1.2 Analisi del trasporto

Esistono diverse tecniche per quantificare il trasporto dei sedimenti, ognuna con i suoi vantaggi e svantaggi: il tracciamento dei sedimenti, il calcolo di variazioni in volume e l'uso di trappole. Il tracciamento dei sedimenti è utile per ottenere gli spostamenti delle particelle nella spiaggia; sono però conosciute le posizioni iniziali e finali, non le traiettorie percorse dal punto A al punto B. Per il calcolo delle variazioni in volume si usano strumenti accurati: scanner laser o dispositivi multi-raggio se vengono studiate zone specifiche e di ridotte dimensioni, telerilevamento tramite satellite per zone più estese, ma sono necessarie condizioni atmosferiche favorevoli. Per il terzo metodo, quello che verrà sviluppato in questo lavoro di tesi, si utilizzano delle trappole per la sabbia: dispositivi che catturano i sedimenti trasportati e che permettono di misurarli mano a mano che sono raccolti al loro interno. In particolare si utilizzerà una trappola per la sabbia già realizzata dal Dipartimento di Scienze della Terra di Pisa con l'obiettivo di progettare e realizzare il sensore che permette di misurare il peso del materiale trasportato.

1.3 Descrizione struttura sperimentale

1.3.1 Struttura meccanica

La struttura meccanica (figura 1.1) permette la raccolta della sabbia trasportata dal vento: il dispositivo è un cilindro di plastica con un'apertura su un lato che permette al materiale di entrare e depositarsi sul fondo, dove è misurato. La struttura è studiata in modo da poter ruotare liberamente sulla sua base e, grazie ad una banderuola (A) posta sulla sommità, orientarsi sempre con l'apertura nella direzione del vento. La struttura rotante (B) è composta da un tubo di PVC di 10 cm di diametro e alto 96 cm, con due aperture rettangolari di 7 cm x 47 cm su due lati, approssimativamente 20 cm sopra la base. Mentre l'apertura davanti è lasciata scoperta, la seconda è chiusa da un velo di poliestere con una maglia di 63 μm che blocca la sabbia, ma permette al vento e a particelle più sottili (limo e argilla) di passare. La base (C) permette di ancorare la struttura nella sabbia, permettendo la rotazione della parte superiore ed evitando attriti o frizioni che possono ostacolarne

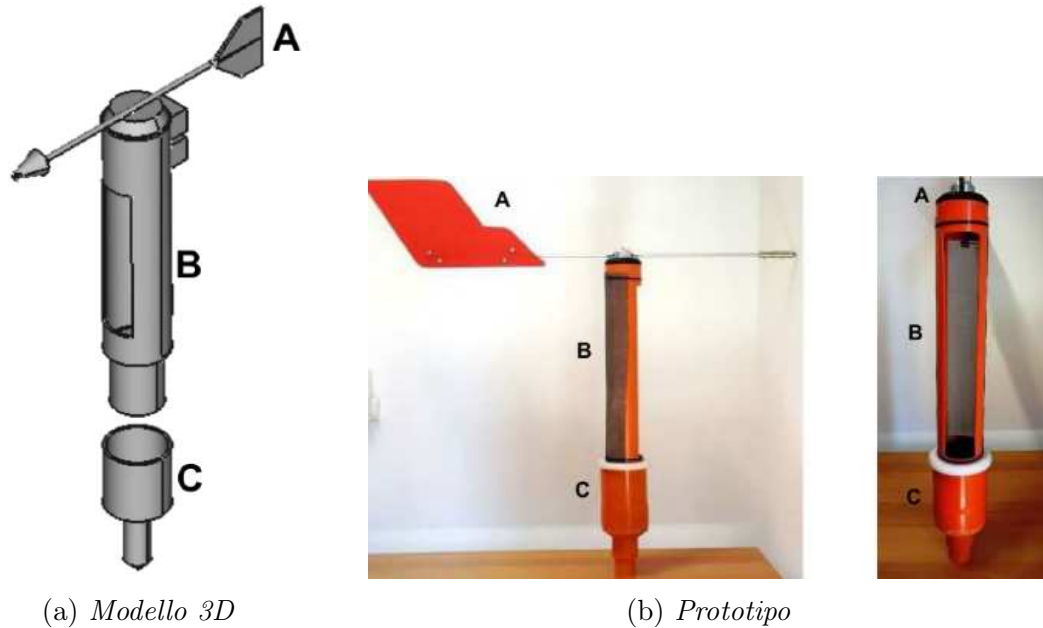
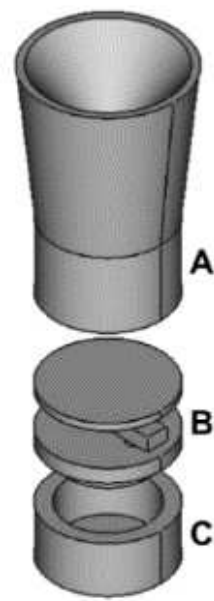


Figura 1.1: Modello 3D e prototipo della struttura meccanica.

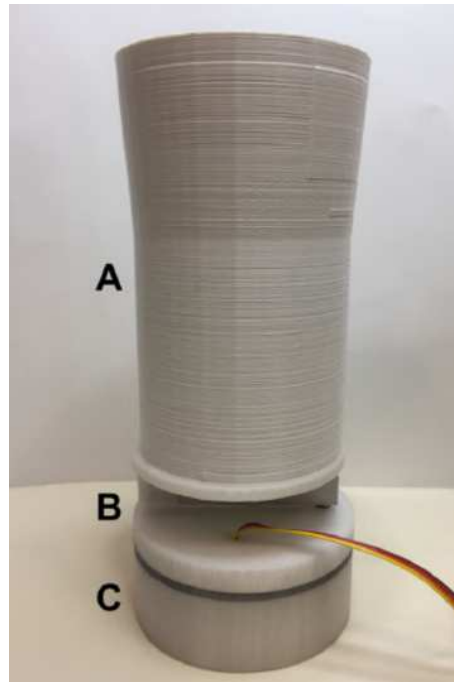
il funzionamento. Anche essa è composta da un tubo di PVC di dimensioni leggermente superiori (105 mm di diametro) connesso ad un altro più piccolo e lungo che permette di ancorarsi in profondità. Quando viene posizionata sulla spiaggia la base è quasi completamente sotterrata, il livello della sabbia è più basso di 1-2 cm.

1.3.2 Struttura di rilevamento

Per misurare la quantità di sabbia raccolta dalla trappola è presente una struttura che integra sia una parte meccanica che una parte elettronica. Questa si trova sul fondo del tubo rotante, è perfettamente aderente ad esso ed è protetta dagli agenti atmosferici. In particolare dalla figura 1.2 è possibile notare la suddivisione in tre sezioni: il contenitore (A), la sezione della cella di carico (B) e l'involucro per l'elettronica (C). Il contenitore è un cilindro alto 15 cm creato con la stampante 3D, svasato verso l'alto e con un imbuto sulla sommità, in modo da indirizzare la sabbia al centro del contenitore ottimizzandone la sua distribuzione. Una volta riempitosi, deve essere svuotato manualmente; durante questa procedura non si interrompe la trasmissione dei dati, tuttavia questi sono erranei in questo lasso temporale. Il contenitore poggia direttamente sul secondo componente della struttura di rilevamento: questo è composto da un supporto inferiore, una cella di carico e un disco superiore. Sia supporto che disco sono stati creati in ABS con la stampante 3D; il primo ha un



(a) *Modello 3D*



(b) *Prototipo*

Figura 1.2: Modello 3D e prototipo della struttura di rilevamento.

diametro di 95 mm, spessore 10 mm ed è provvisto di buchi per far passare i fili nel vano inferiore, mentre il secondo ha diametro di 90 mm e spessore 5 mm. La cella di carico è fissata nel mezzo dei due tramite viti e colla. Anche il vano per l'elettronica è un cilindro in ABS di 95 mm di diametro, 30 mm di altezza e presenta lo spazio necessario a contenere amplificatore, microcontrollore e modulo di trasmissione.

Capitolo 2

I microcontrollori

Un microcontrollore in elettronica digitale è un dispositivo elettronico integrato su singolo circuito, creato per proporre un'alternativa al microprocessore e utilizzato solitamente per applicazioni specifiche di controllo digitale. Oltre a elaborare istruzioni integra fisicamente altre funzioni demandate ad altri componenti elettronici. Ciò in cui si differenzia da un microprocessore è il fatto che si tratta di un sistema completo integrato su un unico chip (tecnologia SoC), interagisce col mondo esterno grazie ad un programma che risiede nella propria memoria interna tramite l'uso di pin specializzati o configurabili dal programmatore. Nel lavoro di tesi proposto ne verranno utilizzati due differenti, ATmega328P (integrato su una board Arduino UNO) e ATtiny84, che verranno presentati nei prossimi paragrafi.

2.1 Arduino

Arduino è una piattaforma hardware e software utilizzata per costruire progetti e prototipi di robotica, elettronica e automazione. Si compone di una scheda programmabile con microcontrollore e di un ambiente di programmazione (IDE), che viene utilizzato per scrivere e caricare il codice con le istruzioni nella scheda stessa. Ciò che rende Arduino ampiamente utilizzato è il fatto che sia open-source, cioè con licenza libera: sia il software che gli schemi circuitali sono di libero accesso alla comunità mondiale garantendo così un processo continuo di miglioramento e di crescita che consentono l'interfacciamento di Arduino con periferiche e componenti di qualsiasi natura. Questo contribuisce inoltre ad una estrema semplicità di uti-

lizzo: grazie alle numerose librerie già create dagli utenti si possono prototipare e progettare idee senza dover disporre di molte conoscenze tecniche.

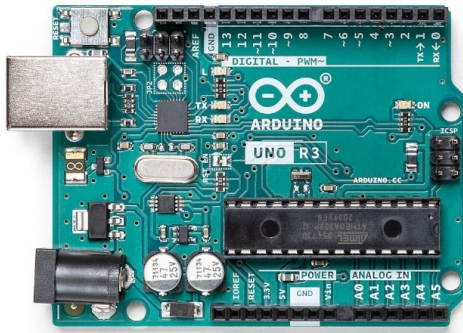
2.1.1 Hardware

Per quanto riguarda il lato hardware, Arduino è una piccola scheda elettronica dotata di un microcontrollore centrale e di altri componenti elettronici che facilitano il collegamento con dispositivi esterni. Per caricare sulla scheda il codice realizzato, che prende il nome di sketch, si utilizza la connessione USB. La board, se opportunamente programmata, è in grado di leggere segnali di input provenienti dai sensori ad essa collegati e di convertirli in segnali di output per controllare attuatori che agiscono sull'ambiente circostante. Un sensore è un componente elettronico che permette di convertire un fenomeno fisico proveniente dall'esterno in segnali elettrici, mentre un attuatore è responsabile della conversione del segnale elettrico in lavoro meccanico.

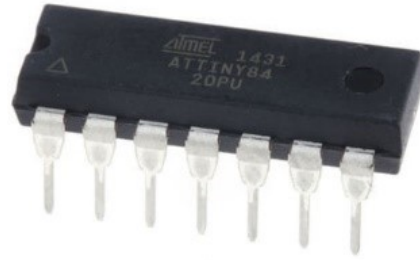
2.1.2 Software

Per quanto riguarda il software, Arduino è fornito di un ambiente di sviluppo semplice e intuitivo che permette di scrivere, compilare e caricare il programma. L'IDE include un editor di testo per consentire la stesura del codice di programmazione e le numerose librerie open source facilitano la gestione di alcuni componenti. Per vedere i risultati di un programma dall'IDE si può attivare un monitor seriale, su cui vediamo comparire le istruzioni seriali tramite l'istruzione *Serial.print()* inserito nello sketch stesso. Il linguaggio di programmazione utilizzato deriva da C e C++ ed è comunemente chiamato Wiring. Ogni sketch Arduino è composto di due strutture fondamentali:

- *void setup()*: in cui si inserisce il codice da eseguire una sola volta e che non verrà modificato fino al caricamento di un nuovo sketch. Qui solitamente si fanno le inizializzazioni dei pin e della comunicazione seriale.
- *void loop()*: contiene il codice del programma che viene ciclicamente ripetuto fino a che la scheda non è spenta o riprogrammata.



(a) *Arduino UNO*



(b) *ATtiny84*

Figura 2.1: Microcontrollori utilizzati.

2.1.3 Arduino UNO

Esiste un vasto numero di board differenti: nel caso di questa tesi è stata scelta una scheda Arduino Uno (figura 2.1a). Tale scheda consiste in un microcontrollore a 8 bit Atmega328P, un regolatore di tensione lineare a 5 V, un oscillatore a cristallo a 16 MHz, una memoria flash, una memoria EEPROM e una memoria SRAM. Inoltre possiede 14 pin digitali di input/output, 6 dei quali possono produrre segnali PWM (modulazione della larghezza d'impulso) e 6 pin analogici di ingresso che acquisiscono un dato di tensione che varia in maniera continua da 0 V a 5 V e lo convertono con una logica a 10 bit (ovvero in 1024 livelli discreti). Infine ci sono 5 pin per l'alimentazione (POWER): un'uscita stabilizzata di tensione a 5 V, una a 3.3 V, due messe a terra (0 V) e un pin V_{in} che fornisce una tensione di input nel caso di alimentazione esterna. Per alimentare la board ci sono tre possibili modi: tramite la porta USB (stabile 5 V), tramite il connettore di alimentazione (AC/DC) o con una batteria esterna connessa ai pin GND e V_{in} della board.

2.2 Microcontrollori ATtiny

Oltre ai microcontrollori ATmega168/328 utilizzati dalle schede Arduino, esiste un'altra famiglia di microcontrollori: gli ATtiny. Questi sono microcontrollori Atmel a 8 bit e con dimensioni ridotte ("tiny" significa piccolo in inglese) molto utilizzati per le loro caratteristiche. Sono infatti l'ideale per progetti in cui lo spazio disponibile è minimo, in cui è richiesto un basso dispendio di energia e non si necessita

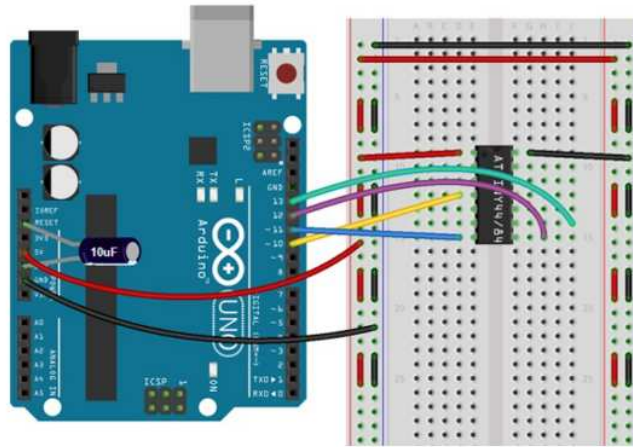


Figura 2.3: Connessioni per caricare uno sketch su ATtiny84.

Tabella 2.1: Connessioni per caricare uno sketch su ATtiny84.

Arduino UNO	ATtiny84
PIN10	PIN4 (RESET)
PIN11	PIN7 (MOSI)
PIN12	PIN8 (MISO)
PIN13	PIN9 (SCK)
+5V	PIN1 (VCC)
GND	PIN14 (GND)

Capitolo 3

Il sensore

3.1 Celle di carico

Una cella di carico è un componente elettronico, in particolare un trasduttore, capace di misurare una forza applicata ad un corpo. È infatti in grado di convertire una forza come una tensione, una compressione, una pressione o un momento torcente in un segnale elettrico che può essere misurato: il segnale elettrico varia proporzionalmente alla forza applicata. Esistono diversi tipi di celle di carico che sfruttano vari principi fisici: le più diffuse sono quelle pneumatiche, idrauliche, capacitive e ad estensimetro.

3.1.1 Cella di carico pneumatica

Una cella di carico pneumatica (figura 3.1) sfrutta la pressione dell'aria: è composta da un diaframma elastico attaccato ad una superficie, sopra alla quale viene posto il peso da misurare. Sono presenti inoltre un regolatore di aria che controlla il flusso all'interno del sistema e un manometro. Il funzionamento è il seguente: quando un oggetto è posto sulla base il sistema inietta aria pressurizzata per bilanciare il suo peso e la misura è effettuata in base alla quantità di aria necessaria al bilanciamento. Il manometro converte in segnale elettrico la pressione.

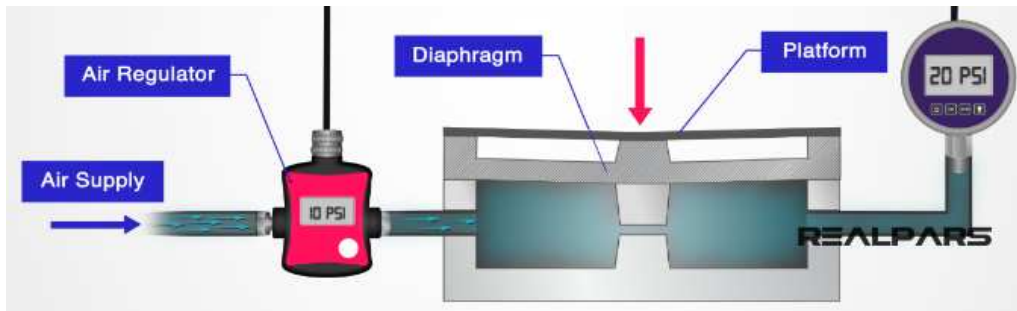


Figura 3.1: Cella di carico pneumatica.

3.1.2 Cella di carico idraulica

Una cella di carico idraulica (figura 3.2) è simile a quella pneumatica, ma è utilizzato un liquido in sostituzione dell'aria (di solito acqua o olio). Anch'essa è composta da un diaframma elastico sul quale poggia un pistone con la superficie di carico e un manometro a tubo di Bourdon. Il peso dell'oggetto fa aumentare la pressione del liquido, ne viene identificata la variazione tramite il manometro e questa viene poi convertita in segnale elettrico.

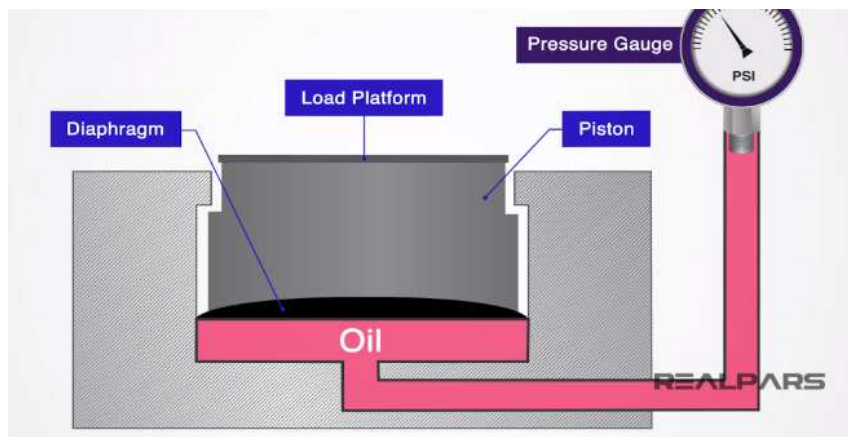


Figura 3.2: Cella di carico idraulica.

3.1.3 Cella di carico capacitiva

Una cella di carico capacitiva (figura 3.3) si basa sulla capacità elettrica, ovvero la proprietà di un oggetto di immagazzinare carica. La cella è composta da due piatti piani paralleli ai quali viene applicata una corrente in modo da caricare il condensatore. Sul piatto superiore poggia la superficie di carico. La capacità dipende dalla distanza dei due piatti: quando un oggetto è posto sulla superficie di carico

questo spazio diminuisce e ciò si traduce in una variazione di capacità del sistema che può essere convertita in una misura di peso.

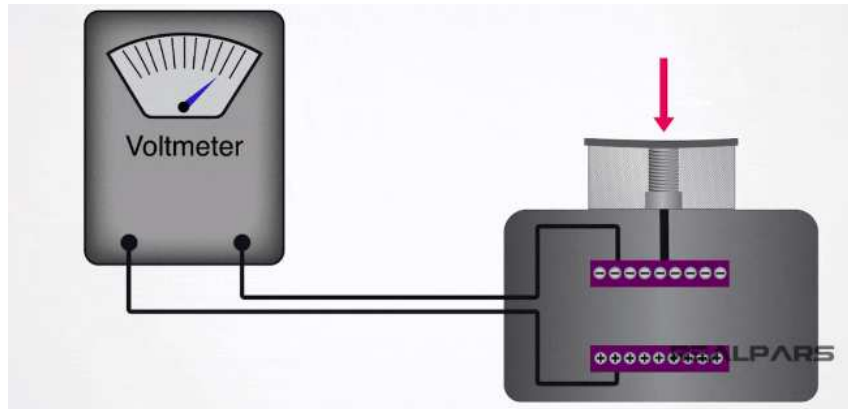


Figura 3.3: Cella di carico capacitiva.

3.2 Cella di carico ad Estensimetro

Un altro tipo di cella di carico, solitamente la più utilizzata e quella che viene usata in questo progetto, è la cella di carico ad estensimetro (figura 3.4). Il principio di funzionamento su cui si basa è la variazione di resistenza elettrica di un estensimetro quando è applicata una deformazione. La resistenza è proporzionale alla sollecitazione meccanica subita dalla cella e questo rende di facile calibrazione il sistema, in modo da produrre misure molto accurate. La cella di carico è costituita da una barretta di alluminio piena, eccetto nel mezzo dove è presente un foro (due circonferenze unite): in questo punto, che subisce la deformazione, sono incollati uno o più estensimetri. Una estremità è fissata ad un supporto, mentre sull'altra viene posta la base su cui pesare l'oggetto.

3.2.1 Estensimetro

Un estensimetro è un sottile foglio flessibile di materiale isolante (solitamente plastica) sul quale è presente una griglia di filo metallico con alle estremità due connettori (figura 3.5a A). Esistono estensimetri di varie grandezze che possono variare da pochi micron ad alcuni centimetri e con diverse disposizioni della resistenza. Esso viene incollato sulla superficie del corpo in modo che il filo segua le deformazioni

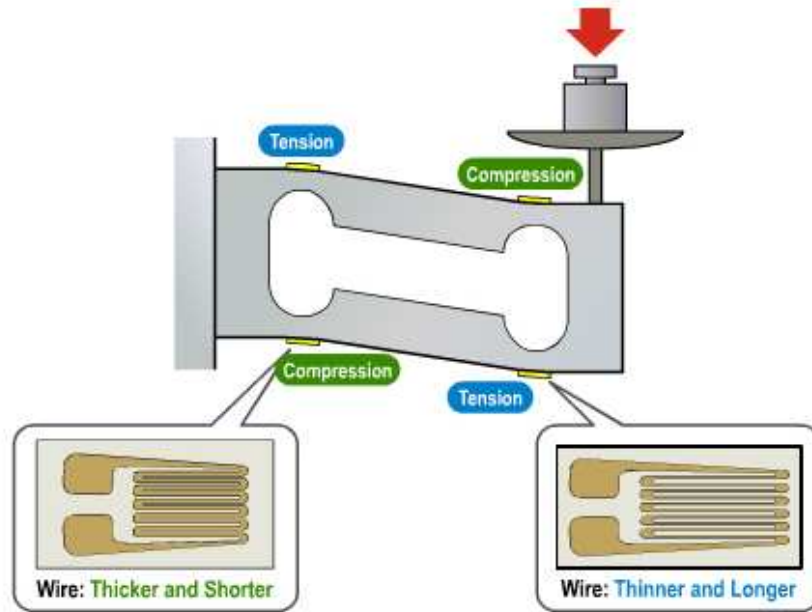


Figura 3.4: Cella di carico ad estensimetro.

della superficie, allungandosi ed accorciandosi insieme ad essa. Ricordando la formula della resistività elettrica, in cui si può notare che la resistenza di un conduttore R è direttamente proporzionale alla sua lunghezza l , se la sua resistività ρ e la sua sezione S rimangono costanti:

$$R = \rho \frac{l}{S}$$

si può dedurre che questi mutamenti dimensionali causano una variazione della resistenza del filo. In particolare se avviene un allungamento la resistenza aumenta, perché cresce l'area dell'estensimetro e quindi la lunghezza del conduttore (figura 3.5a B), al contrario se vi è compressione il filo si accorcia, l'estensimetro diminuisce la sua area, e quindi la resistenza cala (figura 3.5a C).

3.2.2 Ponte di Wheatstone

I microcontrollori sono in grado di misurare la differenza di potenziale ma non la resistenza. Si inserisce quindi l'estensimetro in un ponte di Wheatstone con cui si riesce a convertire la variazione di resistenza (anche piccola) in una variazione di differenza di potenziale. La configurazione (figura 3.5b) si compone di tre resistenze R_1 , R_2 e R_3 note e di uguale valore, pari a quello dell'estensimetro a riposo, che occupa la quarta posizione nel ponte. Si applica una tensione tra i nodi A e C e si misura

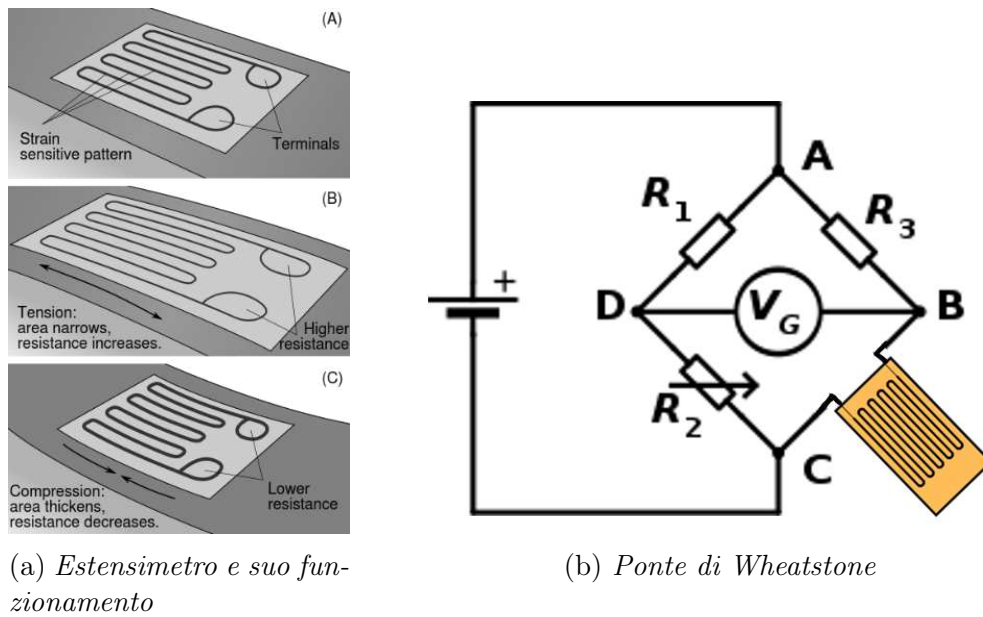


Figura 3.5: Estensimetro e ponte di Wheatstone.

la differenza di potenziale tra D e B. Se il ponte è in equilibrio, ovvero le quattro resistenze sono uguali, V_{cd} è nulla; questo accade quando nessuna forza è applicata all'estensimetro. Se invece la resistenza dell'estensimetro aumenta o diminuisce, V_{ab} sarà minore o maggiore di 0. La configurazione a ponte di Wheatstone risolve inoltre un ulteriore problema, ovvero la variazione di temperatura. Infatti è noto come la resistività di un materiale sia direttamente proporzionale ad essa, quindi le misure prese dall'estensimetro da solo non sono costanti in diverse condizioni. Se inserito in un ponte di Wheatstone invece tutte e quattro le resistenze subiranno la medesima variazione dovuta alla temperatura, mentre solo l'estensimetro subirà quella dovuta alla forza ad esso applicata, quindi la misura terrà conto solo di quest'ultima.

3.2.3 Cella di carico scelta

La cella di carico scelta (figura 3.6) per questo progetto ha 4 estensimetri posti a configurazione di ponte di Wheatstone. Se il ponte è alimentato ad una certa tensione, con l'esercizio di una forza, si produrrà una differenza di potenziale di pochi millivolt all'altra coppia di terminali del ponte. Sono presenti 4 fili: rosso e nero per alimentazione e terra, verde e bianco la differenza di potenziale in uscita dalla cella. Il quantitativo massimo di sabbia che si accumula nella gabbia è di circa 700 g a cui si devono sommare il peso del contenitore che raccoglie i sedimenti (circa

400 g) e del piatto (circa 25 g); è quindi stata scelta una cella di carico da 3 kg.



Figura 3.6: Cella di carico scelta.

3.3 Amplificatore HX711

Poiché la tensione in uscita dalla cella di carico è nell'ordine di grandezza dei millivolt è necessario passare attraverso un amplificatore prima di inviare il segnale al microcontrollore. Per questo lavoro si è scelto di utilizzare un HX711 (figura 3.7). L'HX711 è un convertitore analogico/digitale di precisione a 24 bit, progettato per bilance e applicazioni di controllo industriali, in modo da interfacciarsi direttamente con una cella di carico ad estensimetro. Da un lato sono presenti sei pin per la connessione con la cella di carico: E+, E- per alimentazione e terra e le due coppie A+, A- e B+, B- che corrispondono a due possibili canali di ingresso per il segnale della cella di carico (verrà utilizzato un solo canale, quindi solo i pin A+ e A-). Dall'altro lato invece ci sono quattro pin per l'interfacciamento con il microcontrollore: GND e VCC per terra e alimentazione, DT per i dati e SCK per il clock.

3.3.1 Libreria e comandi utilizzati

Esistono diverse librerie che permettono di interfacciare l'HX711 con Arduino. La libreria scelta per questo progetto è la seguente:

https://github.com/zmaker/arduino_cookbook/blob/5f0099f38a5164848f63d856e154d9882bde97e1/215-cella-carico-calibra/HX711.zip.

Prima di procedere con la scrittura degli sketch è necessario importarla nell'IDE di

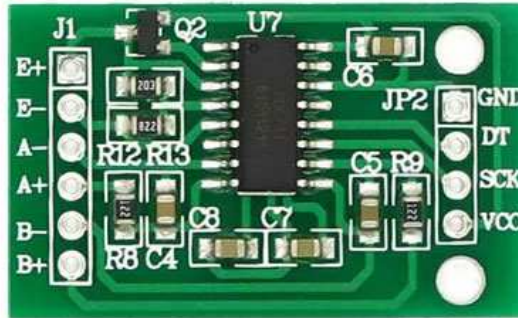


Figura 3.7: Amplificatore HX711

Arduino. Questa libreria funziona normalmente anche su ATtiny84. Di seguito sono riportati i comandi più importanti utilizzati nel progetto:

- "*HX711 scale(DT, SCK)*": crea un oggetto HX711 di nome 'scale' che utilizza il pin DT per i dati e il pin SCK per il clock;
- "*scale.set_scale()*": inizializza l'oggetto 'scale';
- "*scale.tare()*": esegue una taratura, ovvero azzerà il peso;
- "*scale.get_units()*": restituisce il valore della misurazione;
- "*scale.get_units(int)*": restituisce il valore della media di 'int' misurazioni;
- "*scale.set_scale(cal_factor)*": imposta come fattore di calibrazione della cella di carico 'scale' il parametro 'cal_factor'.

Capitolo 4

La trasmissione

Per la trasmissione dei dati di questo progetto si è deciso di utilizzare un modulo radio RFM95 che utilizza la modulazione LoRa.

4.1 Tecnologia di trasmissione LoRa e LoRaWAN

LoRa (da Long Range) è una tecnica di modulazione che consente una comunicazione wireless adatta ad inviare pacchetti di dimensioni ridotte, ma a grandi distanze. I punti di forza dell'uso di questo tipo di trasmissione wireless sono il range elevato, il basso consumo energetico, e la semplicità di utilizzo rispetto a tecnologie come WiFi o Bluetooth. Si può infatti raggiungere un raggio di copertura di vari chilometri in contesto urbano, mentre fino a 15-20 km in ambiente rurale. Queste prestazioni sono ottenute grazie alla modulazione in frequenza che prende il nome di Chirp Spread Spectrum (CSS), che si avvale di diversi segnali chirp estesi nel tempo, ognuno dei quali trasporta informazioni criptate grazie a “salti di frequenza” dei segnali stessi. Grazie a questa caratteristica si ha un aumento della capacità del ricevitore poiché i segnali sono inviati su un'ampia banda di frequenze, riducendo così i consumi e aumentando l'immunità alle attenuazioni. A seconda dell'area geografica, opera su frequenze di 433 MHz oppure 868/915 MHz.

LoRaWAN (“LoRa Wide Area Network”) definisce il livello MAC (Media Access Control) del modello OSI (Open Systems Interconnection); in particolare delinea il protocollo di livello di rete, per la gestione delle comunicazioni tra gateway LPWAN e dispositivi end-node, gestisce le frequenze di comunicazione, la velocità dei dati e

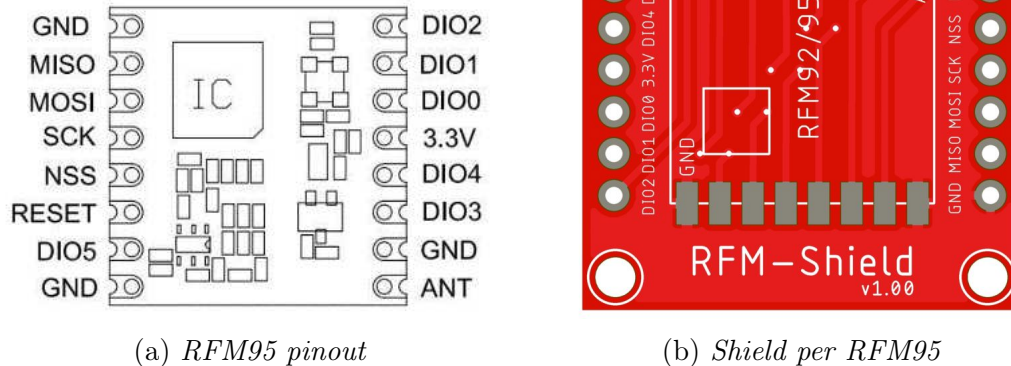


Figura 4.1: Modulo di trasmissione utilizzato.

l'alimentazione dei nodi.

4.2 Modulo RFM95

Il modulo di trasmissione scelto per questo progetto è l'RFM95 (figura 4.1a). Il ricetrasmittente possiede un modem a lungo raggio LoRa™ che permette uno spettro molto ampio e una grande immunità alle interferenze, minimizzando il consumo di corrente. Questa tecnica di modulazione permette una sensibilità di oltre -148 dBm, fino a 8 dB meglio rispetto ad altre tecniche come FSK e OOK. Il modulo trasmette il dato che si desidera inviare; ogni altro ricevitore sintonizzato alla stessa frequenza e con la stessa chiave di crittografia può ricevere e leggere il dato e se necessario inviare una risposta. I moduli RFM9x si occupano di tutti gli aspetti della comunicazione: trasformano i dati in pacchetti, correggono gli eventuali errori in trasmissione e se necessario possono anche auto-ritrasmettere i dati. Il basso consumo energetico è garantito dalla possibilità di programmare il modulo per trasmettere solo quando necessario e restare in “sleep” per il resto del tempo. Per questo progetto il modulo è saldato ad una shield per facilitare le connessioni, in particolare a una shield della Diycon (figura 4.1b).

Capitolo 5

Il progetto sviluppato

In questo capitolo verranno presentate passo passo tutte le operazioni effettuate per la realizzazione di questo progetto, a partire dalla calibrazione e taratura della cella di carico tramite Arduino UNO, la successiva implementazione con AT-tiny84 e stampa su Serial Monitor, fino a concludere con l'integrazione del modulo di trasmissione RFM95.

5.1 Calibrazione e taratura con Arduino UNO

Ogni cella di carico ha un parametro caratteristico, unico e diverso da tutte le altre. Questo viene chiamato fattore di calibrazione e determina come avviene la conversione tra valore di uscita analogico e peso in grammi: si ottiene dal rapporto tra il primo e il secondo. Per ottenere questo parametro ed eventualmente correggerlo è stato creato uno sketch (5.1) da caricare sulla scheda Arduino UNO; è necessario trovarlo una sola volta, in quanto non cambia se si utilizza la stessa cella di carico. Il codice come prima cosa inizializza l'oggetto "scale", con parametri i pin 2 e 3 di Arduino, e la variabile float "calibration", che è il fattore di calibrazione. All'interno del `'void setup()'` lascia all'utente 10 secondi di attesa per effettuare la taratura: infatti per misurare correttamente il peso della sabbia è necessario che sia rimosso quello del piatto superiore e del contenitore di raccolta. Una volta completata la taratura lo sketch chiede di appoggiare nel contenitore un oggetto di peso conosciuto e di mandare da Serial Monitor questa misura in grammi. Viene calcolato successivamente il fattore di calibrazione facendo il rapporto tra il valore

Tabella 5.1: Connessioni per calibrazione e taratura.

Cella di carico	HX711	Arduino UNO
FILO ROSSO	PIN E+	-
FILO NERO	PIN E-	-
FILO BIANCO	PIN A-	-
FILO VERDE	PIN A+	-
-	GND	GND
-	DT	PIN DIGITALE 2
-	SCK	PIN DIGITALE 3
-	VCC	+5V

ricavato con `'scale.get_units(10)'` ("cal") e il valore inserito manualmente ("peso"): $calibration = cal/peso$. Nel `'void loop()'` viene settato questo parametro di calibrazione e viene continuamente stampato a monitor il peso rilevato dallo strumento. Per una maggior precisione è stato inserito un meccanismo per perfezionare il fattore di calibrazione: inviando da Serial Monitor '+' o '-' lo si può aumentare o diminuire di una unità. Per affinare il parametro sono state effettuate prove con oggetti di pesi noti diversi, verificando che la misura rilevata si avvicinasse il più possibile al valore reale e modificando il fattore di calibrazione all'occorrenza. Le connessioni effettuate si possono trovare nella tabella 5.1 (figura 5.1).

Sketch 5.1: Calibrazione e taratura.

```
1 #include <HX711.h>
2
3 HX711 scale(2, 3);
4 float calibration;
5
6 void setup() {
7   Serial.begin(9600);
8   scale.set_scale();
9   Serial.println("Taratura in 10 secondi");
10  delay(10000);
11  scale.tare();
```

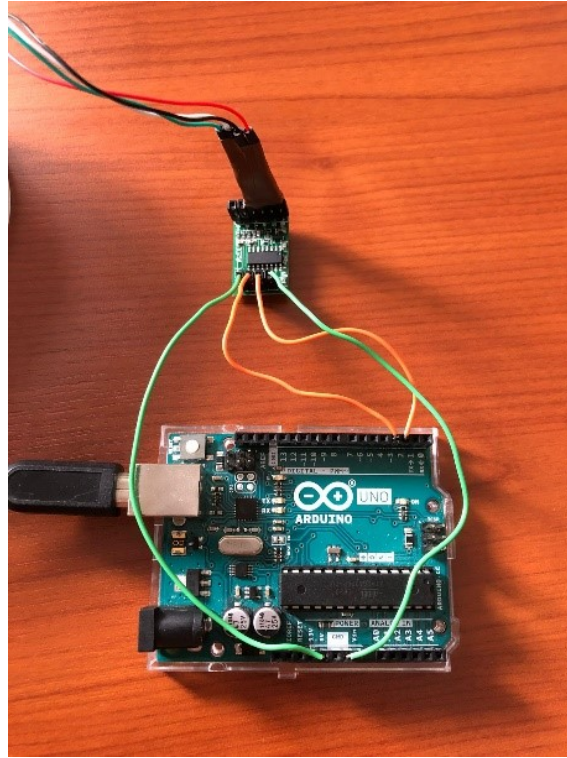


Figura 5.1: Connessioni per calibrazione e taratura.

```
12  Serial.println("Taratura completa");
13  Serial.println("Appoggia un peso conosciuto e manda il
    peso da serial monitor");
14  while (!Serial.available()) {}
15  float temp;
16  if (Serial.available() > 0) {
17      temp = Serial.parseFloat();
18  }
19
20  float cal = scale.get_units(10);
21  float peso = temp;
22  calibration = cal / peso;
23  Serial.println("Cal:");
24  Serial.println(calibration);
25 }
26
27 void loop() {
```

```
28  scale.set_scale(calibration);
29  float units = scale.get_units();
30  Serial.print(units);
31  Serial.print("g.Cal=");
32  Serial.println(calibration);
33  if (Serial.available()) {
34      char temp = Serial.read();
35      if (temp == '+') {
36          calibration += 1;
37      } else if (temp == '-') {
38          calibration -= 1;
39      }
40  }
41 }
```

Il valore trovato dopo vari aggiustamenti è -778.0.

5.1.1 Misurazioni di prova

Per verificare la correttezza del fattore di calibrazione trovato sono state effettuate delle misurazioni di prova: utilizzando uno sketch semplificato (5.2), sono stati registrati i valori analogici di output che la cella restituisce con 50 pesi noti differenti (figura 5.2). Lo sketch utilizzato si occupa solamente di stampare tramite Serial Monitor il valore analogico vero e proprio restituito dalla cella di carico (media di 100 misurazioni), infatti non è stato impostato nessun fattore di calibrazione. Utilizzando poi un semplice foglio di calcolo Excel, dopo aver trovato il fattore di calibrazione per ognuno dei pesi noti, è stata calcolata la media sperimentale con la formula:

$$\overline{cal} = \frac{1}{50} \sum_{k=1}^{50} cal_k = -778,58464$$

con cal_k il fattore di calibrazione trovato per ogni singola misurazione. Come si può vedere la media si avvicina molto al fattore di calibrazione trovato al paragrafo 5.1. Per verificare quanto i valori calcolati si discostano dalla media è stata calcolata

la varianza:

$$s^2(cal_k) = \frac{1}{49} \sum_{k=1}^{50} (cal_k - \overline{cal})^2 = 0,6801 \quad (5.1)$$

da cui si ottiene la deviazione standard

$$s(cal) = \sqrt{s^2(cal_k)} = 0,8247 \quad (5.2)$$

Infine è stata calcolata l'incertezza di tipo A:

$$u(cal) = \frac{s(cal)}{\sqrt{50}} = 0,1166 \quad (5.3)$$

Si può quindi concludere che il fattore di calibrazione trovato al paragrafo 5.1 è da considerarsi abbastanza preciso, sarà quindi utilizzato nelle implementazioni successive.

Sketch 5.2: Misurazioni di prova.

```
1 #include <HX711.h>
2
3 HX711 scale(2, 3);
4
5 float calibration;
6
7 void setup() {
8   Serial.begin(9600);
9   scale.set_scale();
10  Serial.println("Taratura in 10 secondi");
11  delay(10000);
12  scale.tare();
13  Serial.println("Taratura completata");
14 }
15
16 void loop() {
17   float units = scale.get_units(100);
18   Serial.println(units);
```

19 }

PESO(g)	OUTPUT	CAL. FACTOR	PESO(g)	OUTPUT	CAL. FACTOR
52,12	-40523	-777,4942441	716,04	-557642	-778,7861013
96,92	-75235	-776,2587701	758,89	-590915	-778,6569859
147,98	-114939	-776,719827	790,05	-615196	-778,6798304
182,02	-141342	-776,5190638	807,9	-629215	-778,8278252
221,15	-171954	-777,544653	840,04	-654045	-778,5879244
277,09	-215705	-778,4654805	857,68	-668314	-779,2113609
292,88	-227599	-777,1066648	874,95	-681283	-778,6536374
304,7	-236720	-776,8953069	910,1	-708571	-778,5638941
321,32	-250367	-779,1827462	928,99	-723875	-779,20645
338,6	-263538	-778,3165978	950,21	-739827	-778,5931531
351,17	-273060	-777,5721161	964,65	-751805	-779,3552066
371,89	-289855	-779,4105784	984,75	-767602	-779,4892105
387,32	-301613	-778,717856	1020,24	-794651	-778,8863405
408,76	-318279	-778,6451708	1037,12	-807393	-778,4952561
421,8	-328942	-779,8530109	1061,92	-827628	-779,369444
448,23	-348909	-778,4150994	1097,88	-855219	-778,9731118
467,12	-364219	-779,7118513	1106,26	-861615	-778,8539765
491,82	-383105	-778,9536822	1141,72	-889455	-779,048278
523,98	-407937	-778,5354403	1154,95	-899922	-779,1869778
555,86	-433553	-779,9679775	1173,84	-914423	-779,0013971
574,34	-446838	-778,0025769	1205,2	-938959	-779,0897776
600,91	-467681	-778,28793	1248,18	-972560	-779,1824897
630,24	-490637	-778,4923204	1321,34	-1029066	-778,8048496
654,12	-508951	-778,0697731	1369,3	-1066108	-778,5788359
682,09	-531405	-779,0834054	1630,45	-1270002	-778,92729

Figura 5.2: Misure ottenute.

5.2 Implementazione con ATtiny84

Dopo aver ottenuto il fattore di calibrazione della cella di carico si può procedere con l'implementazione con ATtiny84. Invece che passare direttamente alla versione finale con anche il modulo di trasmissione LoRa si è preferito prima testare il funzionamento della libreria dell'HX711 su ATtiny84 utilizzando uno sketch (5.3) che non sarà quello definitivo. È stata utilizzata anche la libreria SoftwareSerial, in modo da avere un riscontro immediato su Serial Monitor, utilizzando Arduino UNO come tramite per la comunicazione con il computer e per l'alimentazione. Per prima cosa, seguendo la procedura presentata al paragrafo 2.2.1, è stato caricato lo sketch su ATtiny. Il codice, oltre a creare un oggetto "scale", crea un oggetto "SoftwareSerial", ovvero una porta seriale che permette al microcontrollore di comunicare tramite le porte Rx e Tx di Arduino. Viene settato il fattore di calibrazione trovato al paragrafo 5.1 e viene eseguita una taratura dopo 5 secondi dall'avvio. Il

programma procede poi con la stampa continua su Serial Monitor del valore del peso rilevato dal sensore in grammi. Le connessioni effettuate si possono trovare nella tabella 5.2 (figura 5.3).

Sketch 5.3: Implementazione su ATtiny84.

```
1 #include <SoftwareSerial.h>
2 #include <HX711.h>
3
4 HX711 scale(2, 3);
5 SoftwareSerial MySerial(10, 9);
6 float calibration = -778.00;
7
8 void setup() {
9   MySerial.begin(9600);
10  scale.set_scale();
11  delay(5000);
12  scale.tare();
13  MySerial.println("Taratura completata");
14 }
15
16 void loop() {
17  scale.set_scale(calibration);
18  float units = scale.get_units(10);
19  MySerial.print(units);
20  MySerial.println("\ng");
21 }
```

Facendo partire lo sketch e provando a misurare diversi oggetti di pesi differenti, si può notare come la libreria dell’HX711 funzioni anche su ATtiny84, nonostante non sia nativamente creata per esso. La misura rilevata ha una buona stabilità nel tempo: avvengono delle oscillazioni ma queste sono lievi, nell’ordine dei centigrammi.

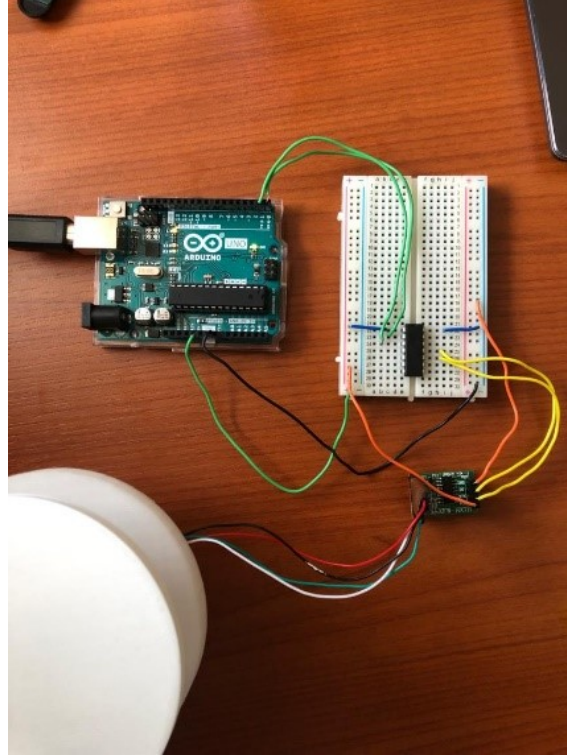


Figura 5.3: Connessioni per implementazione su ATtiny84.

5.2.1 Test dello strumento

Per verificare la correttezza delle misure sono stati effettuati test con 50 pesi differenti senza tutta la struttura meccanica (figura 5.4). Sempre tramite Excel è stato calcolato l'errore di misura per ognuno di questi ed infine lo scarto quadratico medio tra gli errori:

$$s = \sqrt{\frac{1}{50} \sum_{k=1}^{50} (p_k - \hat{p}_k)^2} = \sqrt{\frac{1}{50} \sum_{k=1}^{50} (e_k)^2} = 0,248781028 \quad (5.4)$$

con p_k il peso noto, \hat{p}_k il peso misurato e e_k la differenza tra questi, quindi l'errore. Si può concludere che in media l'errore di misura è leggermente inferiore a $\pm 0,25$ g.

5.3 Integrazione del modulo RFM95

Come ultimo step è stato integrato il modulo di trasmissione RFM95: questo permette al sistema di comunicare con il server dell'Università di Siena tramite un gateway presente al DEI, utilizzando una trasmissione LoRa. Dopo aver effettuato le connessioni come in tabella 5.3 e caricato lo sketch su ATtiny84 come presentato

Tabella 5.2: Connessioni per implementazione su ATtiny84.

Cella di carico	HX711	ATtiny84	Arduino UNO
FILO ROSSO	PIN E+	-	-
FILO NERO	PIN E-	-	-
FILO BIANCO	PIN A-	-	-
FILO VERDE	PIN A+	-	-
-	GND	-	GND
-	DT	PIN11	-
-	SCK	PIN10	-
-	VCC	-	+5V
-	-	PIN14(GND)	GND
-	-	PIN1(VCC)	+5V
-	-	PIN2	PIN0(R_x)
-	-	PIN3	PIN1(T_x)

PESO NOTO (g)	PESO MISURATO (g)	PESO NOTO (g)	PESO MISURATO (g)
19,03	19,33	694,08	694,2
54,76	54,25	713,78	713,38
102,84	102,58	747,49	747,64
147,92	147,65	775,45	775,76
170,12	170,27	794,96	794,62
190,79	190,51	810,91	810,68
217,13	217,02	831,12	830,98
236,01	235,68	854,57	854,66
267,67	267,49	875,35	875,55
292,15	292,3	903,34	903,47
314,24	314,39	915,15	915,07
339,01	339,25	946,02	946,16
359,32	359,05	978,03	977,85
381,02	380,71	993,1	993,23
418,79	418,49	1018,75	1018,84
438,13	437,96	1042,6	1042,95
465,01	464,86	1057,14	1057,48
499,98	499,47	1074,98	1075,31
524,01	523,82	1102,34	1102,48
551,56	551,24	1142,57	1142,97
580,43	580,75	1173,26	1173,05
601,24	601,13	1235,16	1235,34
626,12	626,26	1335,37	1335,48
650,04	649,84	1427,56	1427,32
673,67	673,83	1631,75	1631,56

Figura 5.4: Test di precisione.

al paragrafo 2.2.1 è possibile accedere al sito ad una pagina di consultazione dati su cloud. Dopo aver selezionato il numero identificativo del nodo, nel caso di questo progetto è il nodo C, impostato grazie ai comandi

```
1 unsigned char NwkSkey[16] = { 0x01, 0x02, 0x03, 0x04, 0
    x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0
    x014, 0x15, 0x16 };
2 unsigned char AppSkey[16] = { 0x01, 0x02, 0x03, 0x04, 0
    x05, 0x06, 0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0
    x014, 0x15, 0x16 };
3 unsigned char DevAddr[4] = { 0x00, 0x00, 0x00, 0x0C };
```

si può leggere il contenuto dei pacchetti ricevuti in tempo reale. Il primo pacchetto ricevuto contiene il payload "START—" che conferma l'avvenuta inizializzazione, successivamente sono lasciati 5 secondi per effettuare la taratura e alla fine di essa è ricevuto il pacchetto con payload "TARATURA". Da questo momento in poi sono ricevuti continuamente pacchetti contenenti "*peso* g", dove *peso* sta per la misura rilevata in cifre. I pacchetti sono ricevuti ad intervalli di 20 secondi.

Tabella 5.3: Conessioni per implementazione del modulo RFM95.

Cella di carico	HX711	ATtiny84	Alimentazione	RFM95
FILO ROSSO	PIN E+	-	-	-
FILO NERO	PIN E-	-	-	-
FILO BIANCO	PIN A-	-	-	-
FILO VERDE	PIN A+	-	-	-
-	GND	-	GND	-
-	DT	PIN11	-	-
-	SCK	PIN10	-	-
-	VCC	-	+5V	-
-	-	PIN14(GND)	GND	-
-	-	PIN1(VCC)	+5V	-
-	-	PIN2	-	DIO0
-	-	PIN3	-	NSS
-	-	PIN4(RESET)	-	RST
-	-	PIN7(MOSI)	-	MISO
-	-	PIN8(MISO)	-	MOSI
-	-	PIN9(SCK)	-	SCK
-	-	-	+3.3V	+3.3V
-	-	-	GND	GND

Si può quindi procedere, tramite l'utilizzo di un'alimentazione esterna ad inserire l'elettronica nell'apposito vano alla base della trappola (figura 5.5).



Figura 5.5: Struttura completa della trappola assemblata.

5.3.1 Test dello strumento con struttura completa

Un test della precisione finale della trappola è stato effettuato con la struttura completa per verificare se il tubo interferisce con le misurazioni della cella di carico. Sono state effettuate 50 misure utilizzando la sabbia (figura 5.6): con un piccolo recipiente vengono prelevate piccole quantità di materiale e, dopo essere state pesate, sono fatte cadere all'interno della struttura. In questo processo si è partiti con il contenitore di raccolta vuoto, fino a riempirlo completamente, arrivando ad un peso complessivo di circa 700 g. Tramite un foglio Excel è stato calcolato l'errore di misura per ognuno di questi ed infine lo scarto quadratico medio tra gli errori, come al paragrafo 5.2.1:

$$s = \sqrt{\frac{1}{50} \sum_{k=1}^{50} (p_k - \hat{p}_k)^2} = \sqrt{\frac{1}{50} \sum_{k=1}^{50} (e_k)^2} = 0,403251782 \quad (5.5)$$

con p_k il peso noto, \hat{p}_k il peso misurato e e_k la differenza tra questi, quindi l'errore. Si può concludere che in media l'errore di misura è di poco superiore a $\pm 0,4$ g.

PESO NOTO (g)	PESO MISURATO (g)	PESO NOTO (g)	PESO MISURATO (g)
16,98	16,45	389,26	389,54
33,97	34,25	400,05	400,58
46,89	46,31	408,02	408,37
62,23	62,87	417,49	417,74
74,97	74,6	429,47	429,72
82,24	82,65	445,45	445,04
98,78	98,32	456,69	456,98
124,88	124,43	462,35	462,16
144,59	144,15	479,87	479,35
157,04	157,39	491,56	491,23
171,12	170,89	500,46	500,84
181,14	181,45	512,09	512,59
192,57	192,17	527,45	527,15
209,01	209,35	543,45	543,87
221,67	221,15	569,12	569,67
243,54	243,19	581,34	581,67
259,89	260,33	593,38	593,75
271,54	271,13	608,12	608,43
285,97	286,24	621,54	621,12
292,02	292,23	638,43	638,67
303,11	302,86	649,21	649,93
318,98	319,22	658,67	658,25
338,13	338,4	673,09	673,89
357,25	357,53	693,83	694,14
377,87	378,09	698,87	699,23

Figura 5.6: Test di precisione della struttura completa.

Conclusioni ed implementazioni future

Lo scopo di questo lavoro di tesi è stato quello di realizzare un prototipo per la rilevazione in tempo reale dell'accumulo dei sedimenti, con requisiti fondamentali l'acquisizione remota dei dati e l'elevata precisione nelle misure.

Per quanto riguarda il primo, la richiesta è stata soddisfatta, in quanto la trasmissione wireless dei dati tramite LoRa funziona molto bene, senza interferenze. La frequenza con cui i pacchetti vengono inviati si può aumentare o ridurre a piacimento: in implementazioni future si potrebbe impostare l'invio di dati ad intervalli di alcuni minuti, in modo tale da risparmiare ancora più energia, mandando il modulo RFM95 in modalità di sleep e attivandolo solo il tempo necessario per l'invio. Questo non influenzerebbe di molto le registrazioni dei dati, poiché i cambiamenti di peso all'interno della trappola non avvengono istantaneamente, ma sono molto lenti nel tempo.

Due problemi riscontrati sono il fatto di dover svuotare il contenitore una volta riempitosi e di dover cambiare le batterie quando si scaricano. Una possibile soluzione al primo potrebbe essere la realizzazione di una trappola di dimensioni maggiori, così da riuscire a contenere più materiale, mentre per il secondo problema potrebbe essere l'utilizzo di un pannello solare per l'alimentazione dell'elettronica della struttura. Per quanto riguarda la precisione nelle misure, l'errore trovato dai test può considerarsi buono: senza la struttura esterna è di poco inferiore a $\pm 0,25$ g, mentre di poco superiore a $\pm 0,4$ g con essa. Si può quindi notare che il tubo influenza la misura, probabilmente a causa dell'attrito tra contenitore e pareti interne; in futuro si potrebbe ridurre l'errore anche della trappola completa riducendo il contatto tra le due parti.

È emerso dai test, inoltre, che la misura del peso è molto stabile nel tempo: ci sono delle piccole oscillazioni, ma sono nell'ordine dei centigrammi.

Poiché il peso della sabbia varia a seconda della quantità di acqua presente al suo interno, potrebbe essere implementato in futuro un sensore per rilevare l'umidità del materiale che finisce all'interno del contenitore di raccolta. In questo modo si avrebbe un monitoraggio più preciso del fenomeno.

Bibliografia

- [1] Alessandro Pozzebon, Irene Cappelli, Alessandro Mecocci, Duccio Bertoni, Giovanni Sarti, and Fernanda Alquini. A wireless sensor network for the real-time remote measurement of aeolian sand transport on sandy beaches and dunes. *Sensors*, 18(3):820, 2018.
- [2] Irene Cappelli, Alessandro Mecocci, and Alessandro Pozzebon. Studio e progettazione di una rete di sensori wireless per l'analisi del trasporto dei sedimenti sulle dune costiere. Master's thesis, Università di Siena, dipartimento di ingegneria dell'informazione e scienze matematiche, 2014/2015.
- [3] Microcontrollore. <https://it.wikipedia.org/wiki/Microcontrollore>. Consultato: 17-08-2022.
- [4] Arduino uno. https://store.arduino.cc/products/arduino-uno-rev3/?gclid=Cj0KCQjw6_CYBhDjARIsABnuSzpA6mm4Zaca-fFX5MRuUfnWzvm43mRPuY0Y2pyRD6hNZ8-KiT3\YR0gaAvpjEALw_wcB. Consultato: 17-08-2022.
- [5] Atmega328p datasheet. http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. Consultato: 17-08-2022.
- [6] Attiny84 datasheet. http://ww1.microchip.com/downloads/en/devicedoc/Atmel-7701-Automotive-Microcontrollers-ATtiny24-44-84_Datasheet.pdf. Consultato: 18-08-2022.
- [7] Load cell. https://en.wikipedia.org/wiki/Load_cell. Consultato: 19-08-2022.

- [8] What is a load cell? <https://realpars.com/load-cell/>. Consultato: 19-08-2022.
- [9] Strain gauge. https://en.wikipedia.org/wiki/Strain_gauge. Consultato: 19-08-2022.
- [10] Wheatstone bridge. https://en.wikipedia.org/wiki/Wheatstone_bridge. Consultato: 20-08-2022.
- [11] Hx711 datasheet. https://homotix_it.e-mind.it/upld/repository/File/hx711.pdf. Consultato: 21-08-2022.
- [12] Libreria hx711. https://github.com/zmaker/arduino_cookbook/blob/5f0099f38a5164848f63d856e154d9882bde97e1/215-cella-carico-calibra/HX711.zip. Consultato: 21-08-2022.
- [13] Lora. <https://it.wikipedia.org/wiki/LoRa>. Consultato: 22-08-2022.
- [14] Rfm95. <https://www.hoperf.com/modules/lora/RFM95.html>. Consultato: 24-08-2022.
- [15] Rfm95 datasheet. <https://datasheetpdf.com/pdf-file/748022/HOPERF/RFM95/1>. Consultato: 24-08-2022.

Ringraziamenti

Mi è doveroso dedicare questa sezione a tutti coloro che hanno contribuito al completamento del mio percorso.

In primis un ringraziamento speciale al mio relatore Pozzebon, per avermi accompagnato nel passo più importante, con la sua enorme disponibilità e i suoi consigli. Grazie a lei ho accresciuto le mie conoscenze e le mie competenze.

Un grazie ai miei compagni di corso, per esserci stati nei momenti del bisogno e per aver reso questi tre anni molto piacevoli. In particolare ringrazio Edoardo e Leonardo, con cui sono stato in costante contatto durante tutta la stesura del mio elaborato.

Ringrazio la mia famiglia, mamma, papà e i miei fratelli Camilla e Alessandro, per aver creduto in me fin dall'inizio e per avermi sempre spinto a dare il meglio.

Un grazie alla mia ragazza Anna, con cui ho condiviso tutti i momenti, sia di gioia che di sofferenza, per essere sempre stata al mio fianco, per aver sopportato i miei sfoghi con infinita pazienza e per avermi dimostrato un amore immenso.

Infine un ringraziamento ai miei amici, per essermi sempre stati vicini e per tutti i momenti di spensieratezza vissuti assieme.