



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

“Comunicazione tra dispositivi elettronici tramite luce visibile generata da LED”

Relatore: Prof. Tomasin Stefano

Laureando: Visentin Alessandro

ANNO ACCADEMICO 2022 – 2023

Data di laurea 25/09/2023

Contenuti

Introduzione.....	1
Visible Light Communication	3
2.1 I vantaggi della tecnologia VLC	3
2.2 La modulazione dei sistemi VLC	5
Il sistema VLC sviluppato	7
3.1 Il progetto OpenVCL.....	7
3.2 Le componenti hardware	9
3.3 L'interfacciamento con Arduino Due.....	13
La comunicazione.....	15
4.1 Il canale	15
4.2 La modulazione PPM	16
4.3 La sincronizzazione	18
4.4 La lettura del messaggio.....	20
4.5 La chiusura della trasmissione	21
4.6 La probabilità d'errore.....	22
4.7 Il rapporto segnale/rumore	24
Conclusione.....	27
Bibliografia	29

Capitolo 1

Introduzione

Al giorno d'oggi la porzione dello spettro elettromagnetico maggiormente utilizzata per la trasmissione senza fili è quello delle onde radio, ma data l'enorme quantità di dispositivi connessi e di conseguenza alla banda occupata e quindi non utilizzabile, può risultare utile utilizzare altre bande. In merito a ciò è stata studiata la tecnologia *VLC (Visible Light Communication)*, che consente di far comunicare due o più dispositivi tra di loro tramite segnali luminosi adeguatamente modulati. I vantaggi di questa tecnologia verranno discussi nel capitolo 2, dove verrà confrontata con l'invio di messaggi tramite onde radio. In seguito, nel capitolo 3, si analizzerà il sistema progettato per l'utilizzo di questo sistema di trasmissione, in particolare a livello circuitale, illustrandone i vantaggi e le limitazioni rispetto ad altri progetti già esistenti nello stesso ambito.[1]

Nel capitolo 4, si affronterà il tema della comunicazione tra i due dispositivi di rete, analizzando le problematiche riscontrate e le soluzioni applicate a livello di mezzo trasmissivo, preambolo per la sincronizzazione, lettura del messaggio e chiusura della comunicazione.

Infine, ci si soffermerà sulla probabilità d'errore del segnale inviato, con una breve analisi dei risultati ottenuti dai test pratici svolti e si discuteranno le conclusioni raggiunte.

L'idea di questo progetto è nata dalla ricerca di un sistema più semplice da implementare rispetto a quello analizzato, in modo da poter essere sia meno costoso sia meno complesso. Inoltre lo stesso è stato realizzato in collaborazione con Disarò Alessandro, appartenente allo stesso dipartimento, in particolare per la creazione del circuito e della parte software riguardante la sincronizzazione.

Capitolo 2

Visible Light Communication

2.1 I vantaggi della tecnologia VLC

L'acronimo VLC indica una tecnologia emergente per la trasmissione di informazioni a corto raggio basata sull'utilizzo di impulsi luminosi modulati, spesso generati da *LED (Light Emitting Diodes)*.

Rispetto alla tradizionale trasmissione a corto raggio tramite radiofrequenze, l'utilizzo di segnali luminosi consente di sfruttare uno spettro più ampio di valori, inoltre essendo una tecnologia di recente sviluppo non rischia di avere le problematiche di saturazione dello spettro tipiche delle onde radio.

L'utilizzo della tecnologia VLC diminuisce di molto i rischi collegati all'esposizione prolungata alle radiofrequenze [2] [3] oltre ad altri vantaggi notevoli, ad esempio:

- Maggiore spettro utilizzabile: l'utilizzo massivo di onde radio ha quasi totalmente saturato lo spettro corrispondente (da 3kHz a 300 GHz); ovviamente questo problema non si è ancora presentato per la tecnologia illustrata in quanto di recente impiego, inoltre lo spazio per la luce visibile è nettamente più ampio, da 400 THz a 780 THz.
- Nessuna interferenza elettromagnetica: a differenza dei sistemi RF, i sistemi VLC sono utilizzabili anche negli ambienti suscettibili alle onde elettromagnetiche, come ospedali, aerei e centrali nucleari.
- Facili da implementare: i sistemi VLC possono essere integrati negli attuali sistemi di illuminazione grazie alla loro piccola dimensione. L'apparato di trasmissione può essere integrato in un LED, mentre il sistema di ricezione può essere inserito negli apparati esistenti senza grosse difficoltà.

- Basso costo: collegandosi al punto precedente, i sistemi possono essere inseriti negli apparati esistenti, riducendo sensibilmente i costi.
- Alta efficienza energetica: grazie all'utilizzo di LED ad alta efficienza, è possibile ridurre drasticamente i consumi energetici.
- Sicuri per la salute: a differenza dei laser, i LED a luce visibile non utilizzano dei raggi concentrati, potenzialmente dannosi per la salute.
- Sicurezza delle informazioni: la sicurezza è un aspetto molto importante nell'utilizzo delle RF, in quanto potendo attraversare le superfici solide possono portare al furto di informazioni. I sistemi VLC non soffrono di questo rischio, in quanto non potendo attraversare gli oggetti opachi sono facilmente confinabili.
- Maggiore velocità di trasmissione: dato che i sistemi VLC possono supportare una maggiore larghezza di banda, è possibile raggiungere velocità di trasmissione dei dati molto maggiori rispetto al WI-FI o ai sistemi di reti cellulari.

I sistemi VLC possono comunicare tra di loro in vari modi sia in ambienti interni che esterni, sfruttando i vantaggi appena elencati;

Nei luoghi esterni potremmo avere delle comunicazioni tra due ricetrasmittitori VLC tramite una connessione diretta di tipo punto-a-punto, oppure dei messaggi inviati in broadcast tramite i LED di insegne o segnali, recepibili sia dai dispositivi mobili sia dai sensori solitamente installati sugli autoveicoli.

Allo stesso modo il sistema VLC potrebbe essere utilizzato per avere una triangolazione precisa della posizione del dispositivo tramite l'enorme distribuzione di LED nelle strade cittadine.

2.2 La modulazione dei sistemi VLC

Nei sistemi VLC solitamente vengono utilizzati LED e fotodiodi rispettivamente per invio e ricezione delle informazioni trasmesse tramite la luce visibile, ponendo quindi nuove sfide e nuovi vantaggi rispetto all'utilizzo delle onde radio, che si basano solitamente sull'uso di antenne. Solitamente i LED lavorano seguendo il naturale flusso della corrente, mentre i fotodiodi lavorano in modalità di voltaggio inverso; quindi dal momento in cui questi componenti vengono utilizzati per la trasmissione simultaneamente, è necessario studiare accuratamente la loro cromaticità e la non linearità, confrontando le caratteristiche di LED al fosforo o con mixaggio di colori.[4]

Per quanto riguarda la modulazione del segnale, la modalità più semplice e pertanto implementata solitamente è la *PAM (Pulse Amplitude Modulation)* [5], basata sulla suddivisione della banda in più livelli, uno per ogni simbolo da trasmettere, l'intensità del LED verrà quindi adeguata al livello da trasmettere.

Se poniamo $h(t)_{Tx}$ come l'impulso reale ad energia finita e M come il numero di simboli da trasmettere, il set di forme d'onda disponibili diventa:

$$s_n(t) = \alpha_n h_{Tx}(t) \text{ con } \alpha_n = \frac{2n-1}{M}$$

dove $n = 1, 2, \dots, M$. In altre parole, la forma d'onda è ottenuta modulando l'ampiezza della forma d'impulso $h(t)_{Tx}$ con dimensione $I = 1$.

La modulazione PAM può essere implementata anche tramite l'uso di più LED, ognuno dei quali modulato tramite modulazione *OOK (On-Off Keying)*.

Capitolo 3

Il sistema VLC sviluppato

3.1 Il progetto OpenVCL

OpenVLC è un progetto sviluppato presso *IMDEA (Institutos Madrileño de Estudios Avanzados)* per la creazione di una rete di comunicazione tra due o più dispositivi basata sul già precedentemente esposto protocollo VLC.[6] Gli host sono formati da una scheda *BBB (BeagleBone Black)* (figura 3.1) opportunamente configurata e utilizzata per il lato software del sistema e da un ricetrasmittitore fisico utilizzato per la trasmissione e ricezione del segnale. La BBB è un computer a scheda singola, cioè con tutte le componenti installate su una singola scheda di dimensioni ridotte, che utilizza un processore ARM AM335x, 512 MB di memoria RAM e 4 GB di memoria EPROM.

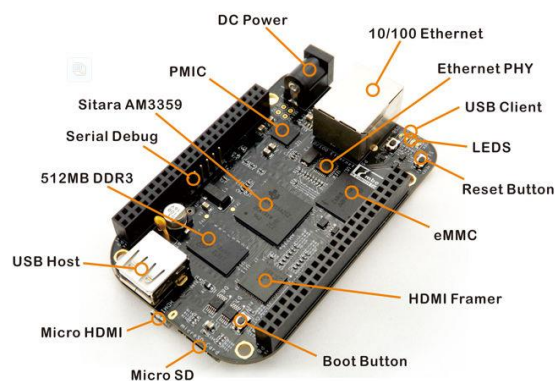


Figura 3.1 BeagleBone Black

Il ricetrasmittitore è una *PCB (Printed Circuit Board)* (figura 3.2), una lastra di vetronite con incise le interconnessioni necessarie tra i vari componenti, che saranno in seguito saldati alla scheda.

Per effettuare la comunicazione viene utilizzato un LED come trasmettitore e un fotodiodo come ricevitore.



Figura 3.2 Apparato OpenVLC 1.3 completo

Il circuito stampato comunica con la BBB tramite dei pin che vengono utilizzati come interfaccia di input e output tra i due componenti; la parte software consiste in un driver per Linux in grado di far comunicare direttamente l'interfaccia esterna front-end con l'interfaccia di rete, difatti il front-end viene implementato a livello software come un'interfaccia di rete connessa a una rete punto a punto. Il sistema operativo utilizzato è Debian [7] in modo da poter sfruttare tutte le funzionalità software necessarie senza dover supportare il peso di un sistema operativo *GUI (Graphic User Interface)* e poter mantenere il sistema finale a basso costo.

Questo progetto è stato la fonte d'ispirazione per il progetto sviluppato, difatti gli host sono composti da una scheda Arduino Due opportunamente programmata e da rispettivamente un trasmettitore composto da un LED e da un ricevitore contenente un fotodiodo. Il segnale viene trasmesso da un computer mittente al trasmettitore, che invia tramite segnale luminoso il messaggio al ricevitore in ascolto e di conseguenza al computer ricevente.

3.2 Le componenti hardware

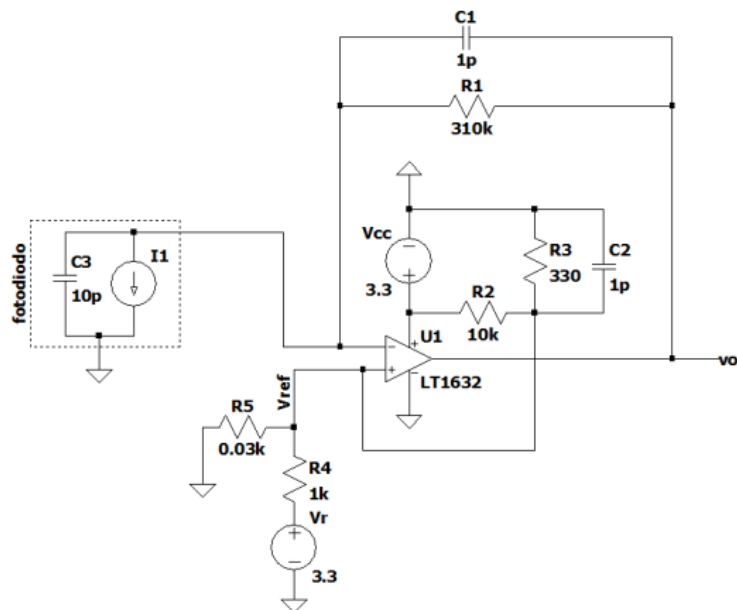


Figura 3.3 Schema elettrico del ricevitore

Il ricevitore (figura 3.3), è alimentato da generatori da 3.3V, così come i punti di massa a 0V, sono forniti da dei specifici pin delle schede Arduino. L'output del circuito (v_0) è connessione ad un pin della scheda impostato in modalità di input.

L'utilizzo di Arduino Due consente di sfruttare l'ADC (*Analog to Digital Converter*) integrato, in modo da rendere più semplice ed economica l'implementazione dello schema elettrico.

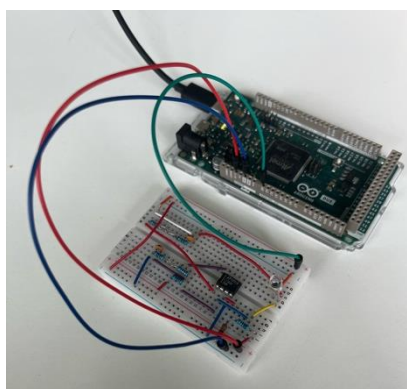


Figura 3.4 Foto del ricevitore

Lo schema mostrato precedentemente in figura 3.3 è stato poi realizzato utilizzando come base una breadboard (figura 3.4)

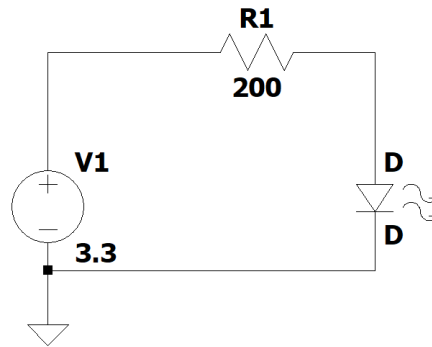


Figura 3.5 Schema elettrico del trasmettitore

Come mostrato dallo schema elettrico in figura 3.5, il trasmettitore è composto da un LED non lineare (con andamento pressoché esponenziale fino al punto di massima potenza) per l'invio del messaggio, alimentato da 0 a 3.3V da un pin di output. Vista l'impossibilità di variare la potenza del LED tramite software (gli unici stati impostabili dal pin sono acceso e spento) si è deciso di variare la distanza tra emittente e ricevente tra i vari test.

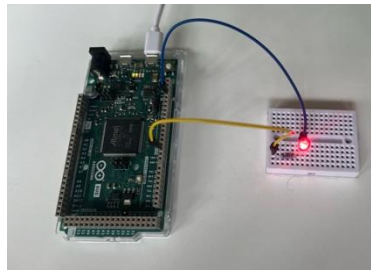


Figura 3.6 Foto del trasmettitore

La tabella 3.1 elenca le componenti utilizzate, oltre a queste sono state impiegati condensatori e resistenze di vari valori, visualizzabili negli schemi elettrici precedenti.

Tabella 3.1 Componenti utilizzati

Codice componente	Descrizione	Quantità
QSD2030-ND	Fotodiodo radiale 880nm	1
1050-1049-ND	Arduino Due	2
296-1395-5-ND	Amplificatore operazionele	1
1125-1075-ND	Led 660nm 50mA	1

Sono state utilizzate due schede *Arduino Due* (figura 3.7), una per il trasmettitore e una per il ricevitore; questo è un microcontrollore, programmabile in un linguaggio C-like, fornito di un processore *Atmel SAM3X8E ARM Cortex-M3* a 32 bit, numerosi pin di input/output analogici e digitali, pin di alimentazione e di messa a terra, oltre a due *DAC (Digital to Analog Converter)* utilizzati per fornire il valore di input ai LED . Si può interfacciare con il PC di riferimento tramite due diverse interfacce, una porta seriale chiamata Programming Port, e una porta USB che supporta un maggior flusso di dati, la USB Port.



Figura 3.7 *Arduino Due*

Per programmare il microcontrollore o leggere i valori forniti in output si utilizza un software specifico chiamato *Arduino IDE* (figura 3.8).

I software *IDE (Integrated Development Environment)* consentono di scrivere, modificare, compilare e verificare del codice, nel caso di *Arduino IDE*, il codice coinvolto è una versione di C opportunamente modificata per agevolare il lavoro a basso livello tramite apposite librerie.



Figura 3.8 Schermata di caricamento di *Arduino IDE*

Inoltre per agevolare la lettura dei valori forniti in output dal ricevitore, è stato utilizzato il software *Termite*[8] che fornisce un terminale per interfacciarsi più semplicemente con le porte COM del PC,

le stesse utilizzate da Arduino. Questa scelta è stata obbligata dalla impossibilità di copiare i valori direttamente dal terminale di Arduino IDE.

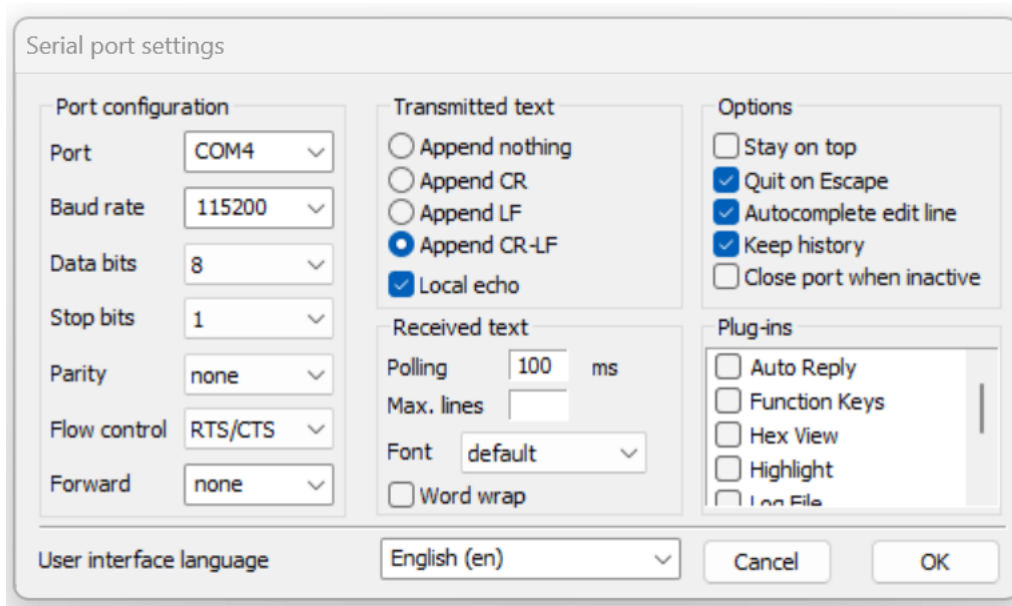


Figura 3.9 Schermata di configurazione di Termit

3.3 L'interfacciamento con Arduino Due

Per inviare il messaggio in output è stato impostato un pin (pin 2) in output, che accenderà e spegnerà il LED tramite l'impostazione di due stati, cioè LOW (0V) e HIGH (3.3V), correttamente alternati durante il periodo di simbolo in quanto la modulazione scelta è una *PPM (Pulse Position Modulation)*.

Al fine di trovare il rapporto segnale/rumore ottimale, vista l'impossibilità di variare il valore HIGH trasmesso in output dal pin, si è deciso di variare la distanza tra ricevitore e trasmettitore. Il ricevitore utilizza la libreria *DueTimer* [9] in modo da gestire autonomamente la lettura degli input senza dover sfruttare il metodo `delay()` che occuperebbe inutilmente il processore, infatti mentre il metodo `delay` interrompe l'esecuzione del processore per un determinato periodo di tempo, l'utilizzo dei timer della libreria citata consente di eseguire uno o più metodi (ad esempio la lettura dell'input) ad intervalli regolari, tramite l'utilizzo del processo di schedulazione. La lettura dei valori in input viene effettuata tramite i metodi della libreria *DueAdcFast* [10] che consentono di leggere i dati in ingresso a velocità superiori rispetto ai metodi di default di Arduino, oltre ad altri vantaggi.

Capitolo 4

La comunicazione

4.1 Il canale

Il canale trasmissivo è ovviamente la luce in quanto si tratta di un sistema VLC, inoltre possiamo considerarlo come un canale *AWGN* (*Additive White Gaussian Noise*) in quanto il rumore, dovuto soprattutto ad eventuali rumori nell'amplificatore e a disturbi dovuti a fenomeni elettromagnetici o termici, è additivo al segnale trasmesso. Altre interferenze sono dovute alle fonti luminose presenti nella stanza, sia di intensità fissa sia con forti variazioni.

L'utilizzo di una modulazione PPM rende la nostra trasmissione più robusta nei confronti di eventuali rumori quindi possiamo avere un margine segnale/rumore minore.

Il nostro canale avrà inoltre capacità:

$$C = \frac{1}{T} \log_2 \left(1 + \frac{\sigma_s^2}{\sigma_w^2} \right)$$

Con T periodo del simbolo, σ_s^2 potenza del segnale trasmesso e σ_w^2 potenza del rumore trasmesso.

4.2 La modulazione PPM

La modulazione PPM si basa sulla suddivisione del periodo di simbolo T in n sottoperiodi, i quali possono assumere un valore uguale all'ampiezza massima A del segnale o zero, in base alla successione di valori otteniamo un determinato simbolo, deciso a priori nel codice. A livello di ricevitore avviene una procedura sostanzialmente simile in quanto il segnale ricevuto viene proiettato sulla base corrispondente e analizzato, dopo essere stato sincronizzato tramite l'uso di un preambolo, come spiegato successivamente. Il vantaggio principale dell'uso della PPM rispetto alla PAM è una maggiore resistenza al rumore, in quanto avendo valori ben lontani (0 e A), anche nel caso di rumore il segnale dovrebbe essere abbastanza leggibile, mentre lo svantaggio principale è la minore larghezza di banda dovuta ai limiti fisici e tecnologici della strumentazione che influiscono sulla frequenza massima raggiungibile. Per ovviare a questa difficoltà si è deciso di optare per una 4-PPM con il seguente codice, costruito in modo da poter trasmettere 4 valori:

0 → 1000

1 → 0100

2 → 0010

3 → 0001

Il codice così creato è stato trasmesso tramite il metodo mostrato in figura 4.1, dove il pin di output viene impostato con valore di uscita HIGH in corrispondenza del bit 1, e LOW in corrispondenza del bit 0, e il valore di output viene mantenuto fisso tramite un comando di attesa.

```
void write(int simbol){
  for(counter=0; counter < simbol; counter++){
    digitalWrite(PIN_OUT, LOW);
    delay(PERIOD/CARDINALITY);
  }
  digitalWrite(PIN_OUT, HIGH);
  counter++;
  delay(PERIOD/CARDINALITY);
  for(; counter < CARDINALITY; counter++){
    digitalWrite(PIN_OUT, LOW);
    delay(PERIOD/CARDINALITY);
  }
}
```

Figura 4.1 Porzione di codice del trasmettitore

Al ricevitore il segnale verrà letto tramite un metodo della libreria DueAdcFast, che come spiegato precedentemente consente di leggere valori in modo molto rapido e facilita le interazioni con il sistema.

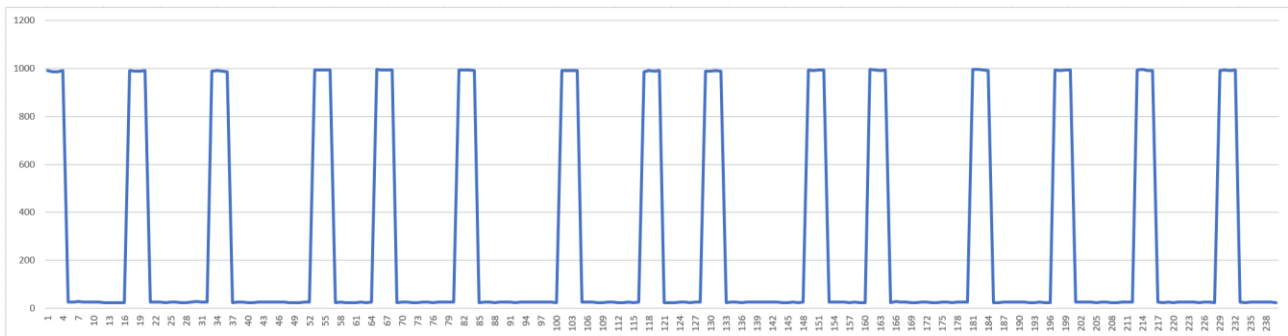


Figura 4.2 Preambolo al ricevitore (1 ciclo)

Nella figura 4.2 possiamo vedere un esempio di segnale ricevuto dal nostro sistema e non ancora proiettato sulla rispettiva base in quanto non sincronizzato; il segnale è stato generato ad una distanza di 30 cm, con un voltaggio in ingresso nel LED di 3.3 V; il segnale è stato

4.3 La sincronizzazione

Per sincronizzare trasmettitore e ricevitore e quindi permettere al secondo di interpretare correttamente il segnale ricevuto e tradurlo nei corrispondenti simboli, viene inviato un preambolo di 45 simboli, creato tramite procedura *MLS (Maximum-Length Sequence)* [11], in questo modo la sequenza ottenuta è a media nulla ed è ottima per la fase successiva, e cioè la convoluzione al ricevitore tra tutti i valori ricevuti e il preambolo stesso; in questo modo è stato possibile ottenere un picco utilizzabile e corrispondente al momento in cui è avvenuta la sincronizzazione.

La sequenza utilizzata per la creazione del preambolo è stata costruita nel seguente modo:

$$n(k + 3) = n(k) \oplus n(k + 2)$$

Formula 4.1

Dove $n(k)$ è l'elemento k -esimo della sequenza MLS, l'operazione utilizzata per il calcolo della sequenza è un'OR esclusivo.

Pertanto, la sequenza ottenuta è 000100110101111, ripetuta 3 volte in modo da creare un preambolo più lungo e facilitare la sincronizzazione tramite la tecnica della convoluzione, in quanto ci si aspetterà tre picchi del segnale, di ampiezza crescente, in corrispondenza di ogni trasmissione dei 15 simboli.

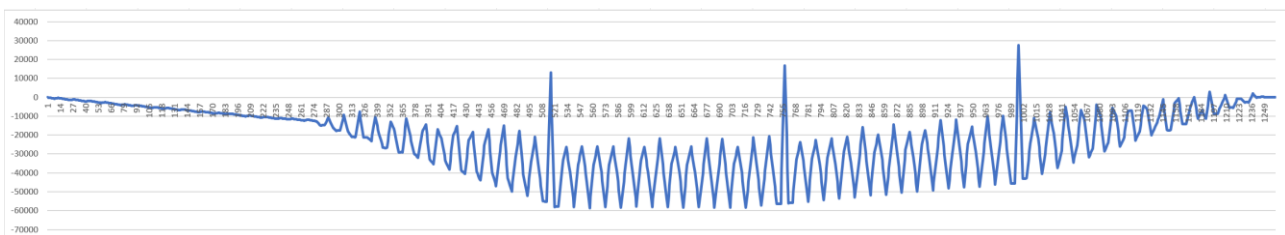
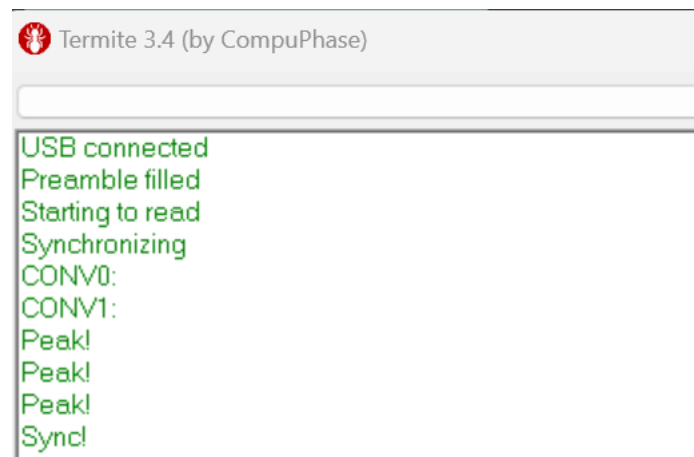


Figura 4.3 Grafico della convoluzione tra i valori ricevuti e il preambolo

Nella figura 4.2 si può notare uno dei segmenti della sequenza utilizzata come preambolo di sincronizzazione e correttamente ricevuta dal fotodiode, mentre nella figura 4.3 possiamo vedere come i tre picchi della corretta sovrapposizione dei segnali siano nettamente visibili.

La sequenza ha una durata totale di 45 simboli, campionati quindi in 720 valori da parte del ricevitore vista la scelta di eseguire un sovracampionamento di 4 volte. Quindi ogni simbolo trasmesso tramite modulazione PPM, composto da 4 bit, verrà sovracampionato

di 4 volte per 4 bit, con un totale di 16 bit per simbolo; da ciò possiamo facilmente derivare il fatto che ogni sequenza MLS venga campionata al ricevitore in 16 bit per 15 simboli, quindi in 240 bit. Il segnale campionato viene registrato in uno spazio di memoria apposito, che svolge la funzione di buffer. Una volta memorizzati i bit, il buffer viene sostituito con un altro vuoto che verrà riempito a sua volta, mentre i valori del primo verranno convoluti con il preambolo atteso in modo da non rischiare di perdere dei valori trasmessi a causa delle operazioni svolte successivamente; per effettuare questa operazione il segnale è stato invertito nel tempo e reso a media nulla sostituendo il valore 0 con il valore -1 all'interno del codice del ricevitore. Durante la convoluzione si cercano i tre picchi già illustrati precedentemente, che indicano la perfetta sovrapposizione dei segnali, e da quel punto inizia la lettura del messaggio a gruppi di 16 bit.



```
Termite 3.4 (by CompuPhase)
USB connected
Preamble filled
Starting to read
Synchronizing
CONV0:
CONV1:
Peak!
Peak!
Peak!
Sync!
```

Figura 4.4 Schermata di Termite relativa alla sincronizzazione con messaggi di debug

4.4 La lettura del messaggio

Una volta sincronizzato il ricevitore, è possibile passare alla lettura del messaggio; per fare ciò i valori verranno suddivisi in gruppi di 4 (a causa del sovracampionamento) e ne verrà calcolata la media (figura 4.5), poi i valori ottenuti verranno a loro volta raggruppati in gruppi di 4, così da ottenere un potenziale simbolo.

```
while(b1.read_pos < b1.position){
    for(int j = 0; j < CARDINALITY; j++){
        int mid_value = 0;
        for(int i = 0; i < OVERSAMPLING;i++){
            mid_value += b1.buffer[b1.read_pos++];
        }
        mid_value /= OVERSAMPLING;
        sampled_values[j] = mid_value;
    }
}
```

Figura 4.5 Codice per il passaggio da 16 a 4 valori per simbolo

I vettori della base sono già ortonormali, quindi i valori precedentemente ottenuti verranno proiettati sulla base al fine di ottenere il simbolo a maggiore somiglianza, secondo il criterio *ML* (*Maximum Likelihood*) [12] (figura 4.6), il quale verrà successivamente stampato.

Una volta letto un intero buffer il sistema aspetterà il riempimento del successivo, e procederà nello stesso modo fino alla ricezione dell'intero messaggio.

```
for(int i = 0; i < CARDINALITY; i++){
    projections[i] = sampled_values[i] * table [i][0] + sampled_values[i] * table [i][1] + sampled_values[i] * table [i][2] + sampled_values[i] * table [i][3];
    if(projections[i] > projections[max_index])
        max_index = i;
}
```

Figura 4.6 Codice per la proiezione sulla base

4.5 La chiusura della trasmissione

Per terminare la trasmissione il mittente si limiterà a non inviare più segnali luminosi, infatti il ricevitore dopo un tempo deciso a priori di mancata ricezione, ovvero dalla ricezione di valori in input troppo bassi e quindi assimilabili al rumore di fondo (figura 4.7), dichiarerà la trasmissione conclusa.

Ovviamente una volta terminata la trasmissione del messaggio sarà possibile inviarne altri, ripartendo con il preambolo come indicato precedentemente, in quanto il ricevitore si resetterà e tornerà in ascolto.

```
if(projections[max_index] < min_value * CLOSING_FACTOR)
    closing_counter++;
```

Figura 4.7 Controllo della chiusura della trasmissione

4.6 La probabilità d'errore

I test condotti con il sistema ideato hanno purtroppo avuto dei risultati abbastanza altalenanti e una probabilità d'errore eccessivamente alta, causata probabilmente dall'inesattezza del processo successivo alla sincronizzazione, come verrà successivamente discusso nel capitolo 5.

I seguenti grafici illustrano come la probabilità d'errore non è variata significativamente con il variare della distanza di trasmissione, se non a causa dei limiti fisici di trasmettente e ricevente (poca potenza del LED, altissima direzionalità del fotodiode).

I test successivi sono stati quindi condotti variando la frequenza di trasmissione, al fine di controllare se il problema fosse un numero troppo elevato di operazioni da eseguire, anche se data la frequenza di lavoro del dispositivo sembrava abbastanza improbabile.

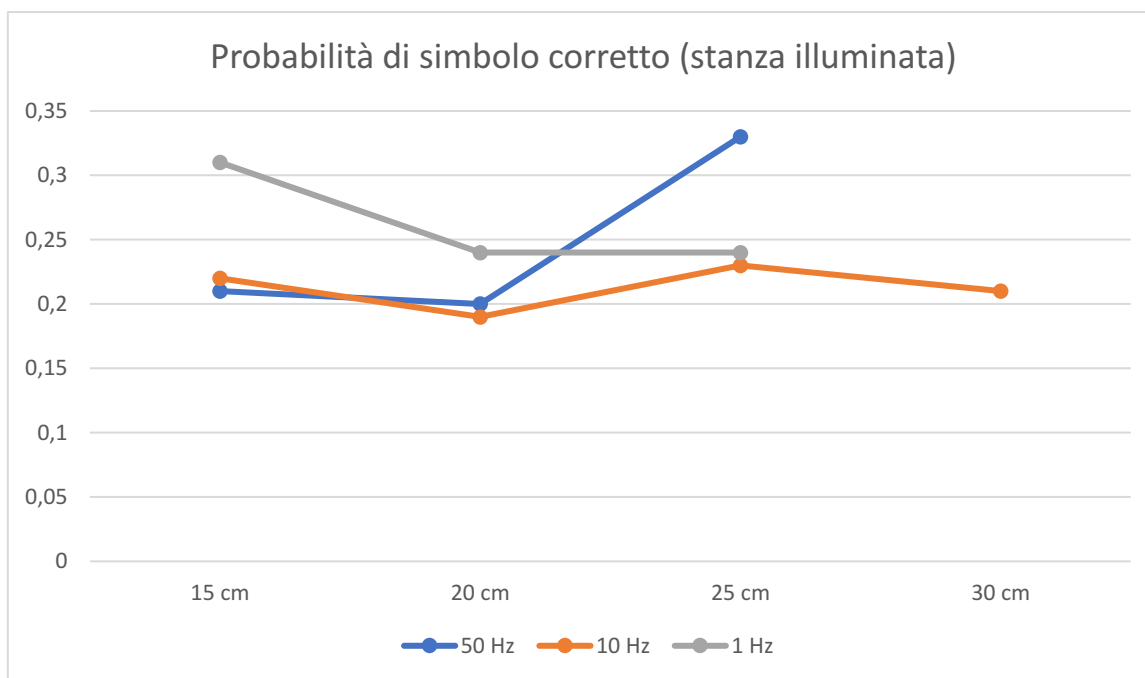


Grafico 4.1

Tutti i test illustrati sono stati condotti a frequenze diverse (da 1 a 50 Hz) e a distanze diverse (da 15 a 30 cm) così da poter notare come variasse la qualità della trasmissione e allo stesso tempo poter trovare il rapporto segnale/rumore ottimo per la connessione creata.

Alcuni tentativi sono stati scartati in quanto i due dispositivi non sono riusciti a sincronizzarsi nonostante i numerosi tentativi, ciò è accaduto sempre e solo nel caso della distanza maggiore; quindi, probabilmente è dovuto alla poca potenza del LED utilizzato.

Il primo gruppo di tentativi è stato condotto in un ambiente illuminato in condizioni normali da luce naturale, come si può notare la trasmissione con la più bassa probabilità di errore è stata ottenuta con la frequenza maggiore e alla distanza maggiore, però l'unico segnale trasmissibile alla distanza massima di 30 cm è stato generato alla frequenza di 10 cicli al secondo, ed è stato inoltre il segnale con la probabilità d'errore più stabile tra quelli analizzati.

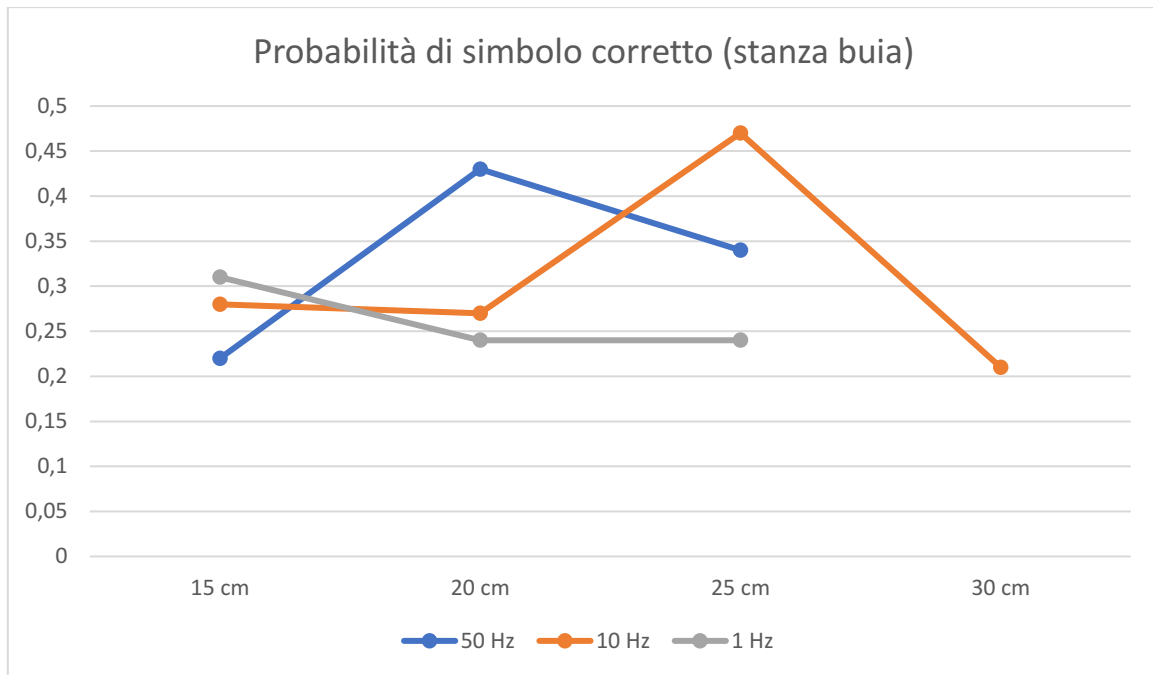


Grafico 4.2

Il secondo gruppo di test è stato condotto in una stanza completamente buia, e ciò ha ridotto il rumore ambientale dovuto alla luminosità di fondo.

Come si può notare dai risultati ottenuti, il miglioramento delle prestazioni è stato notevole, mostrando ancora un buon risultato della trasmissione a 50 Hz, anche se ad una distanza minore rispetto a quella precedente.

Il risultato migliore è stato ottenuto dalla trasmissione a 10 Hz, che oltre ad aver ottenuto la minor probabilità d'errore, è stata ancora l'unica ad aver raggiunto la distanza maggiore.

4.7 Il rapporto segnale/rumore

Il sistema di input utilizzato al ricevitore registra valori compresi tra 0 e 4095, quindi si è deciso di utilizzare questa scala per calcolare la potenza del segnale, in quanto essendo un segnale ottico il suo valore minimo sarebbe sensatamente 0. [13]

Il segnale trasmesso su canale AWGN è $x(t)$, che a ricevitore diventerà $r(t) = s(t) + w(t)$, dove $w(t)$ è il rumore additivo.

Il rapporto segnale rumore, solitamente indicato come *SNR (Signal to Noise Ratio)*, viene calcolato grazie alla potenza del segnale trasmesso e alla potenza del rumore, che essendo una variabile aleatoria gaussiana a media nulla è uguale alla varianza del rumore stesso (equazione 4.3).

$$P_s([0, T]) = \frac{E_s}{T} = \frac{\int_0^T s(t)^2 dt}{T} = \frac{A_s^2 T}{4T} = \frac{A_s^2}{4}$$

Equazione 4.2

L'equazione 4.2 mostra la potenza per simbolo del segnale trasmesso, dove E_s è l'energia del segnale trasmesso, T il periodo del segnale trasmesso e A la sua ampiezza; la potenza è divisa per 4 in quanto è la cardinalità della PPM creata.

$$P_w([0, T]) = \sigma^2$$

Equazione 4.3

Come ipotizzato nel sottocapitolo 4.6, la combinazione che ha portato ai risultati migliori è stata una frequenza di trasmissione di 10 Hz ad una distanza di 25 cm; il segnale trasmesso ha avuto un'ampiezza massima media di 576, e quindi potenza 82944, mentre il rumore ha avuto una potenza media di 125.

L'SNR ottenuto è quindi:

$$SNR = 10 \log_{10} \frac{P_s}{P_w} = 10 \log_{10} \frac{82944}{125} = 28.2 \text{ dB}$$

Equazione 4.4

Un altro risultato interessante ai fini dello studio effettuato è stato quello del segnale trasmesso alla frequenza di 50 Hz e alla distanza di 20 cm; il fotodiodo con i parametri impostati in questo modo

riceve un segnale di ampiezza massima media di 2140, e quindi potenza 1.144.900 e un rumore con potenza 16900.

L'SNR ottenuto è quindi:

$$SNR = 10 \log_{10} \frac{P_s}{P_w} = 10 \log_{10} \frac{1144900}{16900} = 18.3 \text{ dB}$$

Equazione 4.5

Come si può notare dai risultati ottenuti, l'SNR relativo ai parametri che hanno consentito di ottenere una trasmissione più corretta ha un valore nettamente più alto del secondo miglior caso, mostrando un aumento del 55%.

Capitolo 5

Conclusione

I risultati illustrati nel sottocapitolo 4.6 dimostrano come il metodo trasmissivo funzioni correttamente ma sia migliorabile, in particolare il sistema creato dimostra di avere un'alta probabilità d'errore, che però si mantiene costante al variare della distanza di trasmissione, fino ai limiti fisici imposti dai componenti utilizzati; in particolare per aumentare la distanza massima di trasmissione si sarebbe potuto utilizzare un LED ad alta potenza al posto di un semplice LED, anche se in questo caso potrebbero acuitizzarsi problemi relativi al fenomeno definito "sfarfallamento" [14] e di conseguenza aumentare il rumore e la probabilità d'errore all'aumentare della frequenza di lavoro.

Il paragrafo 4.7 ha sostenuto ciò che era stato dedotto nel paragrafo 4.6, infatti i segnali che hanno mostrato empiricamente di avere la minore probabilità di errore sono gli stessi che hanno mostrato il miglior SNR.

Il software non presenta criticità, anche se potrebbe successivamente essere ottimizzato o migliorato in alcune parti, in particolare sulla parte successiva alla sincronizzazione, dato che si ipotizza essere il punto più delicato del codice per quanto riguarda gli errori sui simboli.

Sono state provate altre frequenze trasmissive superiori a 50Hz, ma il fotodiodo non riconosceva la sequenza di preambolo in quanto troppo rapida per la lettura.

Bibliografia

1 - Q. Wang, D. Giustiniano, D. Puccinelli, "OpenVLC: Software-Defined Open Architecture for Embedded Visible Light Networks", Demo at ACM VLCS workshop, ACM Mobicom 2014.

2 - D. Giustiniano, N. O. Tippenhauer and S. Mangold, "Low-complexity Visible Light Networking with LED-to-LED communication," 2012 IFIP Wireless Days, Dublin, Ireland, 2012, pp. 1-8, doi: 10.1109/WD.2012.6402861.

3 - H. Burchardt, N. Serafimovski, D. Tsonev, S. Videv and H. Haas, "VLC: Beyond point-to-point communication," in IEEE Communications Magazine, vol. 52, no. 7, pp. 98-105, July 2014, doi: 10.1109/MCOM.2014.6852089.

4 - <https://www.energy.gov/sites/prod/files/2016/08/f33/led-color-characteristics-factsheet.pdf>

5 - N. Benvenuto, M. Zorzi. (2011). Digital Modulation System. In "Principle of Communication Network and Systems" (pp 259-372). Wiley.

6 - "IEEE Standard for Local and Metropolitan Area Networks--Part 15.7: Short-Range Wireless Optical Communication Using Visible Light," in IEEE Std 802.15.7-2011, vol., no., pp.1-309, 6 Sept. 2011, doi: 10.1109/IEEESTD.2011.6016195.

7 - <https://www.ucamco.com/en/gerber>

8 - https://www.compuphase.com/software_termite.htm

9 - <https://github.com/ivanseidel/DueTimer>

10 - <https://github.com/AntonioPrevitali/DueAdcFast>

11 - <https://www.commsp.ee.ic.ac.uk/~mrt102/projects/mls/MLS%20Theory.pdf>

12 - Fisher, R. A. "On an Absolute Criterion for Fitting Frequency Curves." *Statistical Science*, vol. 12, no. 1, 1997, pp. 39–41. *JSTOR*, <http://www.jstor.org/stable/2246266>

13 - M.D. Audeh, J.M. Kahn, J.R. Barry, "Performance of Pulse-Position Modulation on Measured Non-Direct Indoor Infrared Channels", in *IEEE Transactions on Communications*, vol. 44, no. 6, pp.654-659, June 1996.

14 - Wilkins, Arnold & Veitch, Jennifer & Lehman, Brad. (2010). LED lighting flicker and potential health concerns: IEEE standard PAR1789 update. 2010 IEEE Energy Conversion Congress and Exposition, ECCE 2010 - Proceedings. 171 - 178. 10.1109/ECCE.2010.5618050.