

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO di
INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE
"Guglielmo Marconi"
DEI

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA ELETTRONICA

TESI DI LAUREA
in
Sensori e trasduttori per l'industria e l'ambiente M

**OTTIMIZZAZIONE DEL FUNZIONAMENTO
DI UN SENSORE INDUSTRIALE
A TRIANGOLAZIONE OTTICA BASATO SU
TRASDUTTORI LINEARI D'IMMAGINE**

CANDIDATO
Marco Casali

RELATORE
Prof. Ing. Gaetano Pasini

CORRELATORI
Prof. Ing. Pier Andrea Traverso
Ing. Salvatore Bertulu

Anno Accademico 2017/2018

Sessione III

Alla mia famiglia

Indice

1. Introduzione	1
1.1 Il fotodiode.....	3
1.2 I sensori.....	7
1.3 Tipologie di sensori di prossimità	11
1.4 Tipi di ricevitori.....	19
1.5 L'array di fotodiode IC-LFM	33
1.6 Il sistema integrato PSoC.....	35
2. Scansione totale	39
2.1 Programmazione microcontrollore.....	41
2.2 Risultati ottenuti.....	45
3. Scansione parziale	51
3.1 Programmazione microcontrollore.....	51
3.2 Risultati ottenuti.....	55
4. Distanza.....	57
4.1 Approssimazione curva gaussiana	57
4.2 Programmazione microcontrollore.....	61
4.3 Risultati ottenuti.....	67
4.4 Fitting iperbole	69
4.5 Isteresi.....	71
5. Conclusioni e sviluppi futuri.....	75
Bibliografia	77
Ringraziamenti	78

1. Introduzione

Nell'ambito dell'automazione industriale in particolare, e del controllo in generale, si utilizzano sensori allo scopo di trasformare segnali (e quindi informazioni) di natura fisica, in segnali elettrici rendendoli adatti alle successive elaborazioni.

Le informazioni prelevate da un sensore generalmente forniscono indicazioni circa lo stato di un sistema e sono quindi un componente fondamentale per poterlo controllare.

Il sensore restituisce un segnale di tipo elettrico poiché questo genere di segnale può essere ben acquisito, memorizzato ed elaborato.

Medicina, industria e robotica sono i settori in cui essi vengono utilizzati maggiormente.

Ne esistono delle più svariate tipologie a seconda degli utilizzi: sensori di portata, sensori di corrente, sensori di temperatura, sensori di posizione, sensori di accelerazione, sensori di suono, ecc ecc...

Verrà trattato un particolare tipo di sensori di posizione chiamati sensori di prossimità.

Questi sensori vengono molto utilizzati nell'ambito dell'automazione industriale per rilevare la presenza di un oggetto in una specifica posizione.

Essi quindi presentano un'uscita che può essere di solo due tipi (presenza/assenza) ed è per questo che vengono anche chiamati switch di prossimità.

Si differenziano leggermente dunque dai tipici sensori in grado di fornire un'uscita che è funzione continua del misurando.

Ci si concentrerà soprattutto su un particolare tipo di sensore ottico di prossimità chiamato in gergo a soppressione di sfondo (BackGround Suppression, BGS).

Successivamente, si faranno considerazioni e valutazioni su quale tipo di ricevitore utilizzare e si adotterà una tecnologia basata su un array di fotodiodi.

Il presente elaborato di tesi ha come scopo finale lo studio e l'approfondimento dei metodi realizzativi di sensori BGS basati su array di fotodiodi.

L'obiettivo è la rilevazione della presenza di un oggetto in una determinata posizione, attraverso la stima della sua distanza, ottenuta con il miglior compromesso tra accuratezza e velocità di acquisizione ed elaborazione.

L'intera attività è stata svolta presso il gruppo R&D della divisione Sensor and Safety di Datalogic S.r.l., con sede a Monte San Pietro (BO).

L'azienda è leader a livello mondiale nei settori di acquisizione automatica dei dati e di automazione dei processi ed è in particolare specializzata nella progettazione e produzione di lettori di codici a barre, mobile computer, RFID, sensori per la rilevazione, misurazione e sicurezza, sistemi di visione e laser marking.

Di seguito si sviluppano gli elementi teorici utili alla trattazione dell'elaborato.

1.1 Il fotodiode

In elettronica, il fotodiode è il tipico trasduttore che permette di trasformare un segnale ottico in un segnale elettrico [1].

La struttura è concettualmente identica a quella di un normale diode (presenta quindi anch'esso un anodo e un catodo), ma cambia l'asimmetria nel drogaggio della giunzione p-n. Essa è progettata in modo da realizzare una zona svuotata di grandi dimensioni, nella quale le coppie elettrone lacuna possano essere separate efficacemente.

Per massimizzare questa dimensione, si realizzano fotodiode dove tra la zona drogata N e la zona drogata P è interposta una zona intrinseca (fotodiode PIN).

Il suo funzionamento si basa sul principio fotoelettrico: quando un fotone, avente energia compatibile col bandgap tra banda di valenza e banda di conduzione, viene assorbito da un semiconduttore, si genera una coppia elettrone-lacuna.

Dalla curva caratteristica che mostra la sensibilità spettrale relativa di un fotodiode PIN al silicio, si può dedurre come il fotone non viene assorbito se presenta una lunghezza d'onda minore di 400 nm o maggiore di 1110 nm.

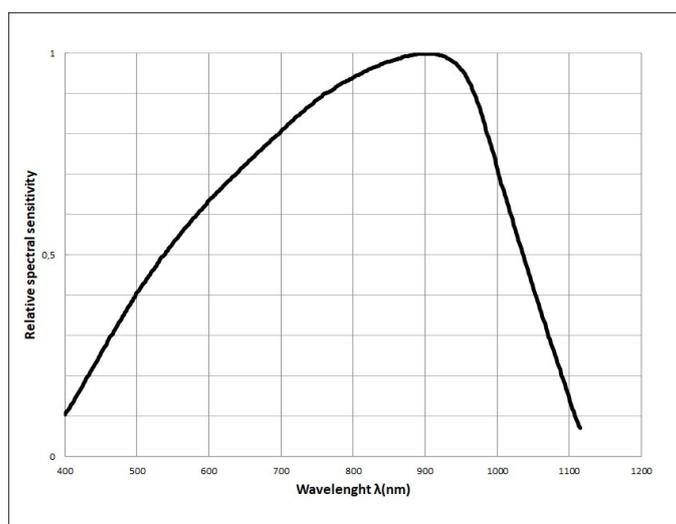


Figura 1.1 Sensibilità spettrale relativa di un fotodiode

Si può ricavare come per il caso di emissione LED rossa ($\lambda \approx 620$ nm), la sensibilità risulta pari al 65%.

Normalmente le coppie elettrone lacuna generate si ricombinano spontaneamente senza dare luogo ad un effetto netto. Se però vengono generate in una zona dove è presente un campo elettrico (così avviene nella zona di svuotamento di una giunzione), esse vengono separate dal campo elettrico stesso dando luogo ad una corrente netta detta fotocorrente i_{ph} .

$$i_{ph} = r_{\varphi} \varphi_e$$

Dove r_{φ} è la sensibilità spettrale e φ_e è il flusso di energia incidente.

Come si vedrà meglio nel modello equivalente più avanti, la corrente totale che attraversa il fotodiode è quindi contributo di due correnti:

- 1) la corrente del diode ideale espressa dalla legge di Shockley chiamata "dark current" perché presente in assenza di illuminazione;
- 2) la fotocorrente i_{ph} sopracitata.

$$I = i_s \left(e^{\frac{V_D}{\eta V_T}} - 1 \right) - i_{ph}$$

- I_s è la corrente di saturazione
- V_D è la tensione applicata al fotodiode
- η è il fattore di idealità della giunzione
- V_T è la tensione termica pari a $\frac{kT}{q}$

La qualità di un fotodiode dipende da quanto è bassa la dark current in modo da massimizzare la sensibilità all'illuminazione.

Si esamina ora la caratteristica I/V di un fotodiode.

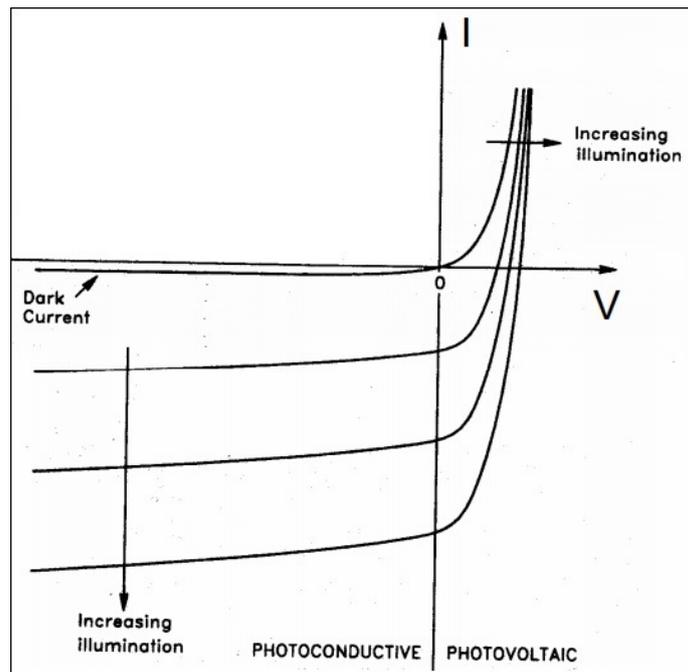


Figura 1.2 Caratteristica I-V del fotodiode

Il fotodiode ha 3 regioni principali di funzionamento:

- polarizzazione diretta (relativa al I° quadrante): se si alimenta il fotodiode con una tensione positiva ($V > 0$), esso si comporta come un comune diode erogando corrente in modo esponenziale. Non è tuttavia concepito per funzionare in questa modalità in quanto la corrente diretta che lo attraversa non è un parametro di progetto e si tende a mantenerla nulla o bassa per evitare surriscaldamenti o altri fenomeni dannosi;

- modalità fotovoltaica (relativa al IV° quadrante): in assenza di una polarizzazione esterna, il fotodiodo genera comunque una corrente i_{ph} proporzionale all'intensità luminosa; tale corrente polarizza direttamente il diodo D in parallelo (si veda figura successiva del modello) generando quindi ai suoi capi una tensione pari a :

$$V = V_T \ln \left(\frac{i_{ph}}{i_D} \right) = \frac{KT}{q} \ln \left(\frac{r_\phi \phi_e}{i_D} \right)$$

La tensione generata è funzione logaritmica della luce incidente (modo di operare tipico delle celle fotovoltaiche);

- modalità fotoconduttiva (relativa al III° quadrante): il fotodiodo è polarizzato in inversa ($V < 0$). In queste condizioni aumenta l'estensione della zona di svuotamento, e quindi aumenta la probabilità che la coppia elettrone lacuna generata da un fotone assorbito possa essere separata generando un contributo alla fotocorrente che sarà proporzionale all'illuminamento (questa è la modalità tipica di utilizzo del fotodiodo).

Di seguito si approfondisce il modello equivalente di un fotodiodo.

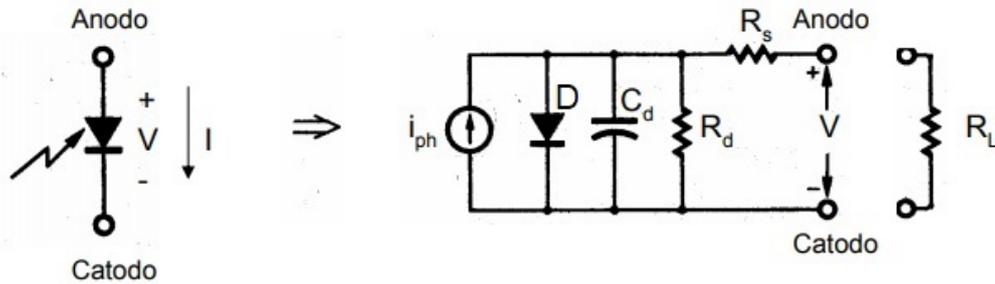


Figura 1.3 Simbolo e modello equivalente di un fotodiodo

Il modello del fotodiodo (per la modalità fotoconduttiva) è costituito fondamentalmente da:

- un generatore di corrente ideale che eroga una corrente i_{ph} proporzionale all'illuminazione, che rappresenta la fotocorrente;
- un diodo ideale D che rappresenta il comportamento della giunzione in caso di polarizzazione diretta;
- una capacità parallelo C_d che indica la capacità giunzione del diodo e che dipende direttamente dalla larghezza della zona di svuotamento e quindi indirettamente dalla tensione inversa: $C_d = \frac{C_{d0}}{\sqrt{1 + \frac{V}{\phi_B}}}$, dove

C_{d0} è la capacità a zero bias, ϕ_B il potenziale di built-in e V la tensione applicata;

- una resistenza di shunt R_d di valore generalmente elevato ma che decresce esponenzialmente all'aumentare dell'illuminazione;
- una resistenza serie R_s di valore generalmente ridotto che descrive la zona di diffusione del semiconduttore e le resistenze di contatto.

Esterna al modello, è riportata la resistenza R_L che rappresenta la resistenza equivalente di ingresso del circuito a valle, tipicamente un amplificatore di transimpedenza che trasforma la corrente del fotodiode in una tensione.

La resistenza in ingresso di un amplificatore di transimpedenza è molto bassa (idealmente nulla). Nel caso ideale (R_s e R_L nulle), tutta la corrente fotogenerata entra nell'ingresso dell'amplificatore.

1.2 I sensori

È innanzitutto necessaria una precisazione riguardo la differenza tra trasduttori e sensori, dato che nel linguaggio comune e nell'uso pratico i due termini vengono utilizzati per indicare lo stesso concetto.

I *trasduttori* sono dispositivi che ricevono informazioni da un segnale in ingresso (detto misurando), costituito da una determinata grandezza fisica, e le restituiscono mediante un segnale di uscita costituito da una grandezza fisica, della stessa specie oppure di specie diversa, più adatta alle successive elaborazioni.

I *sensori* sono l'elemento primario di una catena di misura e convertono la variabile in ingresso in un segnale misurabile.

Essi sono una parte interna dei trasduttori ed hanno il compito di entrare direttamente in contatto con le grandezze fisiche delle quali si vuole ricavare informazioni.

D'ora in avanti ci si uniformerà al gergo comune parlando quindi sia di sensore che di trasduttore per indicare l'oggetto finito.

Per ogni trasduttore si definisce una funzione di conversione che lega il valore del misurando $x(t)$ a quello dell'uscita del trasduttore $y(t)$:

$$y(t) = f [x(t), g_1, g_2, \dots, g_n]$$

Tale funzione è condizionata da parametri non sottoposti a misura detti grandezze di influenza g_n come per esempio temperatura, pressione, umidità, tensione, corrente, vibrazioni, ecc.. la cui influenza sul risultato della misura deve essere annullata o comunque tenuta sotto controllo.

Uno degli obiettivi della progettazione di un trasduttore è l'ottenimento di una funzione di conversione f che sia la più semplice possibile, in particolare con una dipendenza ridotta dai parametri g_n e una dipendenza da $x(t)$ monotona (in molti casi anche la linearità è una caratteristica importante, nei sensori di prossimità in particolare però non è richiesta).

Per questo tipo di trasduttori i principali parametri che ne permettono la classificazione sono: [2]

Linearità: indica quanto la relazione tra misurando ed uscita si discosta dall'andamento lineare;

Risoluzione: è la minima variazione del misurando che provoca una variazione apprezzabile della grandezza di uscita;

Sensibilità: è il valore minimo di misurando che fornisce uno spostamento apprezzabile della grandezza d'uscita rispetto il suo valore minimo;

Stabilità: è la capacità del trasduttore di mantenere inalterate le sue caratteristiche di funzionamento per un certo intervallo di tempo;

Accuratezza: indica lo scostamento del valor medio campionario risultante da una serie di valori rispetto a quello reale (o di riferimento) e dipende sia da errori casuali che sistematici;

Isteresi: è la massima differenza fra i valori dell'uscita corrispondenti allo stesso misurando raggiungendo il valore prima partendo dall'estremo inferiore poi partendo dall'estremo superiore;

Portata: è l'intervallo dinamico di ingresso che può essere convertito dal sensore con un'incertezza prestabilita;

Ripetibilità (o precisione): è l'attitudine del trasduttore a fornire valori dell'uscita poco diversi tra loro quando in ingresso viene applicato lo stesso misurando per più volte consecutive senza modificare le condizioni operative;

Tempo di risposta: è l'intervallo di tempo che intercorre dal momento della rilevazione del target e la conseguente variazione di stato dell'uscita (da on a off o viceversa).

I sensori di prossimità di cui ci si occupa sono una particolare categoria di sensori deputati a rilevare la presenza di un oggetto (detto target) entro una distanza massima dal sensore indicata con d_{switch} .

Dal punto di vista della classificazione, sono sensori di distanza che realizzano una funzione fortemente non lineare che è idealmente del tipo:

$$y(x) = \begin{cases} 1 & x \leq d_{switch} \\ 0 & \text{altrove} \end{cases}$$

dove x è la distanza del target dal sensore.

Per garantire stabilità della rilevazione, in genere a questa funzione è associata una certa quantità di isteresi; in questo modo la relazione che lega lo stato dell'uscita alla distanza è, ad esempio:

$$y(x) = \begin{cases} 1 & \{(x \leq d_{\text{switch-ON}} \cap dx < 0) \cup (x \leq d_{\text{switch-OFF}} \cap dx > 0)\} \\ 0 & \{(x \geq d_{\text{switch-OFF}} \cap dx > 0) \cup (x \geq d_{\text{switch-ON}} \cap dx < 0)\} \end{cases}$$

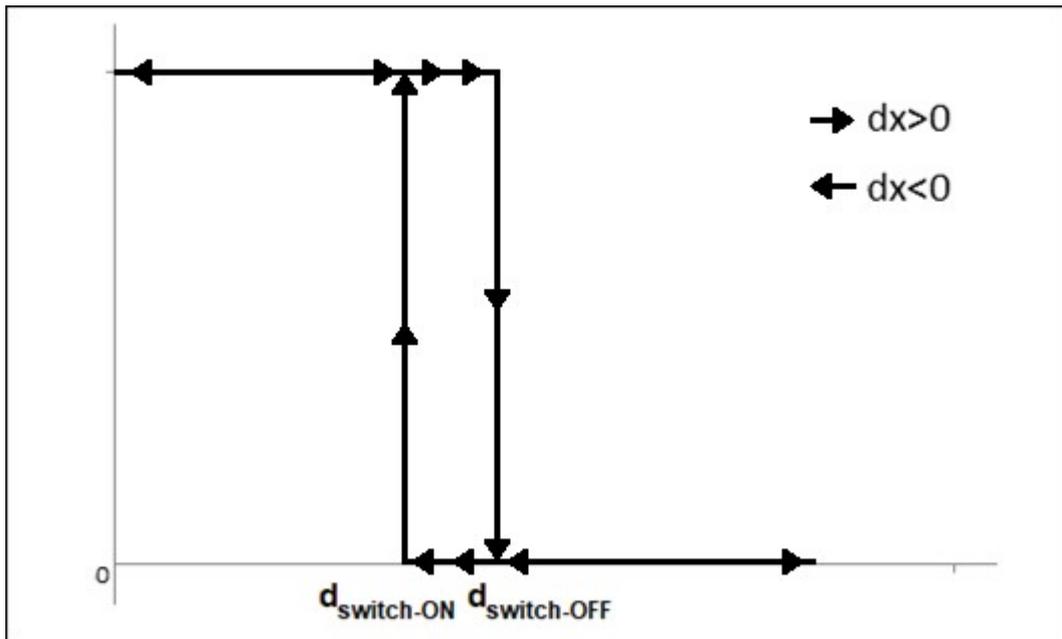


Figura 1.4 Grafico che rappresenta l'isteresi

1.3 Tipologie di sensori di prossimità

I sensori di prossimità trovano applicazione nel mondo dei controlli di processo industriale, robot industriali, macchine utensili, strumenti di misura, linee di montaggio, e così via.

Si elenca qualche applicazione tipica:

- Finecorsa: si utilizza un sensore di prossimità per rilevare la presenza di un oggetto (per esempio un carrello di una macchina utensile) in prossimità dei limiti del suo campo di movimento;
- Presenza: si rileva la presenza di un oggetto in modo utile alla successiva attivazione di un'ulteriore funzionalità della macchina. Per esempio, un macchinario che riempie casse di plastica con bottiglie d'acqua non aziona il meccanismo finché non rileva la presenza effettiva della cassa;
- Rilevazione bordo: si individua il bordo di un oggetto (ad esempio una scatola) perché tale informazioni torna utile come sincronizzazione del processo di posizionamento di un'etichetta al suo centro;
- Livello: si controlla se all'interno di un serbatoio il liquido rimane entro i livelli desiderati;

La tecnologia con la quale sono costruiti, ne identifica le diverse categorie:

- Sensori di prossimità induttivi, sfruttano il fenomeno dello smorzamento di un campo elettromagnetico per effetto delle correnti indotte (correnti di Foucault) in materiali conduttivi posti nelle loro vicinanze. È presente un circuito oscillante con una bobina che genera campo elettromagnetico ad alta frequenza che induce, in azionatori metallici vicini, correnti parassite. Tali correnti provocano una perdita di energia nell'oscillatore, smorzando l'ampiezza del segnale che viene rilevata da un amplificatore e paragonata ad una soglia con isteresi. Sono in grado di rilevare soltanto oggetti di materiali conduttore (acciaio, ottone, alluminio e rame);

- Sensori di prossimità capacitivi, basati sulla variazione di capacità elettrica che si crea tra sensore ed un qualunque potenziale costante. Tale capacità viene perturbata dall'oggetto da rilevare. In corrispondenza di una determinata distanza del target dalla faccia sensibile del sensore, un circuito interno al sensore rileva la variazione di capacità e commuta il segnale d'uscita. Rilevano oggetti anche non conduttori (legno, liquidi e materiali plastici);
- Sensori di prossimità ad ultrasuoni: l'eco di ritorno di un ultrasuono indica la presenza dell'oggetto (il cosiddetto "*Time of flight*"), la distanza dell'oggetto è misurata usando la semplice equazione $d = c \cdot t$, dove c è la velocità dell'onda propagata e t il tempo di volo. Il maggiore vantaggio è che il sistema di rilevamento è indipendente da colore e forma del target, di contro sono richiesti circuiti a radiofrequenza ad elevata sensibilità perciò molto costosi e complicati;
- Sensori di prossimità ottici (o fotoelettrici), basati sulla riflessione o diffusione di un fascio luminoso da parte del target.

L'attenzione viene focalizzata proprio su quest'ultima categoria: i sensori fotoelettrici (detti anche più comunemente fotocellule).

Sono formati da una sorgente ottica che emette un raggio luminoso che colpisce direttamente o indirettamente un ricevitore fotoelettrico.

A seconda della modalità di rilevamento si distinguono 3 principali famiglie di sensori fotoelettrici:

1. a barriera
2. a catarifrangente
3. a diffusione diretta

1. A barriera

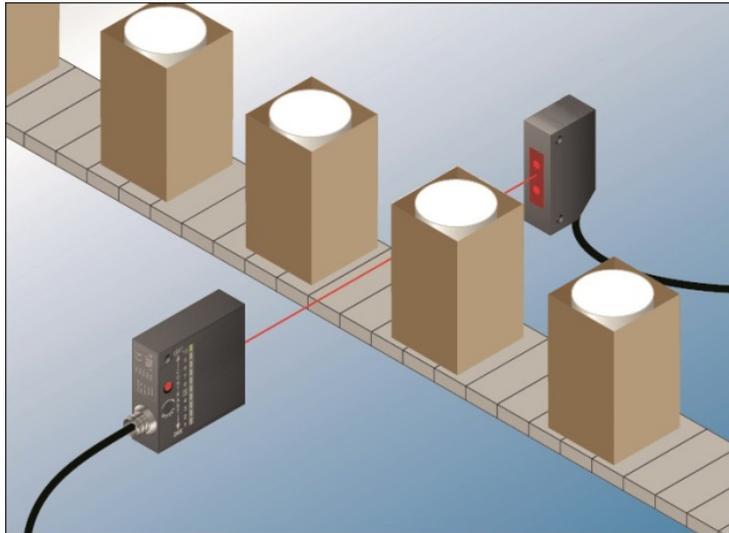


Figura 1.5 Applicazione di un sensore a barriera

È l'unica tipologia in cui emettitore e ricevitore non si trovano incorporati nella stessa unità, ma in due ben distinte.

Esse vengono poste una di fronte all'altra: la rilevazione dell'oggetto avviene mediante l'interruzione del percorso del fascio luminoso da parte del target.

Questa soluzione è molto affidabile, permette di raggiungere campi operativi tra i più elevati ma a costi altrettanto cospicui soprattutto a causa della doppia unità e dell'installazione che deve prevedere il perfetto allineamento dei due elementi.

Un ulteriore svantaggio nell'aver le unità separate risiede nella necessità di doverle sincronizzare rendendo quindi la rilevazione più lenta e meno immune al rumore.

2. A catarifrangente

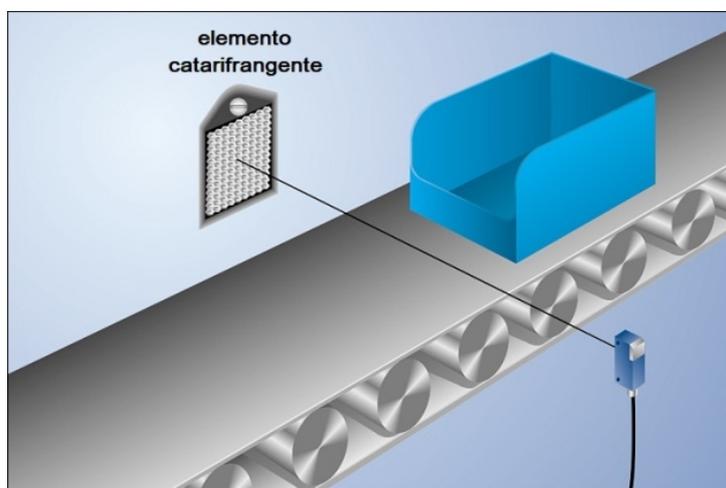


Figura 1.6 Esempio di un sensore a catarifrangente

Emettitore e ricevitore sono integrati nella stessa unità.

Il segnale luminoso proveniente dall'emettitore viene riflesso da un catarifrangente (o un riflettore prismatico) per poi tornare verso il sensore stesso: l'interruzione del fascio luminoso determina la presenza del target.

Il catarifrangente effettua una riflessione del raggio esattamente nella stessa direzione di provenienza, non seguendo quindi la legge di Snell come nei classici casi di riflessione.

Per fare ciò, esso è composto da tre specchi perpendicolari tra loro corrispondenti ad uno spigolo di cubo (da qui il nome corner cube).

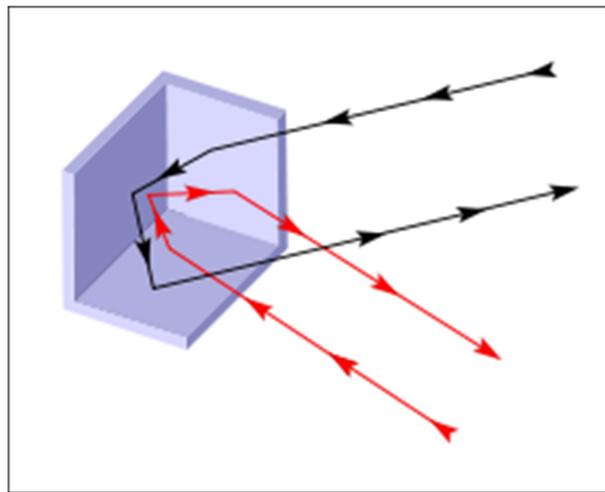


Figura 1.7 Sistema corner cube alla base di un catadiottero

Tale sensore trova impiego soprattutto in quei sistemi in cui non è possibile utilizzare la doppia unità separata come per esempio in caso di spazi ristretti. Questa è la versione classica non polarizzata, che permette distanze di rilevazione moderate ed un allineamento semplice.

Una volta che sensore e riflettore sono allineati, la rilevazione avviene misurando l'energia del fascio ricevuto: se è inferiore ad una certa soglia, significa che un oggetto opaco sta interrompendo il fascio luminoso.

Può nascere un problema di non-riconoscimento di oggetti caratterizzati da superfici molto chiare e/o speculari orientate normalmente all'asse ottico, in quanto non distinguibili, da un punto di vista energetico, dal riflettore.

In questi casi si può realizzare una variante che sfrutta la proprietà del riflettore catadiottrico di ruotare la polarizzazione di un raggio incidente di 90° .

Aggiungendo un polarizzatore lineare al sistema di emissione ed un secondo polarizzatore ruotato di 90° rispetto al primo sistema di ricezione, si può realizzare un sensore che si comporta nel seguente modo:

- catadiottero: la polarizzazione della luce riflessa è ruotata di 90° rispetto all'emissione, passa attraverso il polarizzatore della ricezione e viene rilevata;

- specchio: la polarizzazione del raggio riflesso è la stessa del raggio emesso. Il raggio riflesso viene bloccato dal polarizzatore in ricezione e non viene quindi rilevato (lo specchio viene visto come un oggetto opaco);
- oggetto molto chiaro: la luce diffusa non è in generale polarizzata. Il polarizzatore in ricezione attenua il raggio ricevuto di circa il 50% (quello che basta affinché un oggetto bianco venga rilevato come opaco).

I costi e le prestazioni sono leggermente inferiori rispetto ai sensori a barriera e l'installazione è più semplice.

3. A diffusione diretta

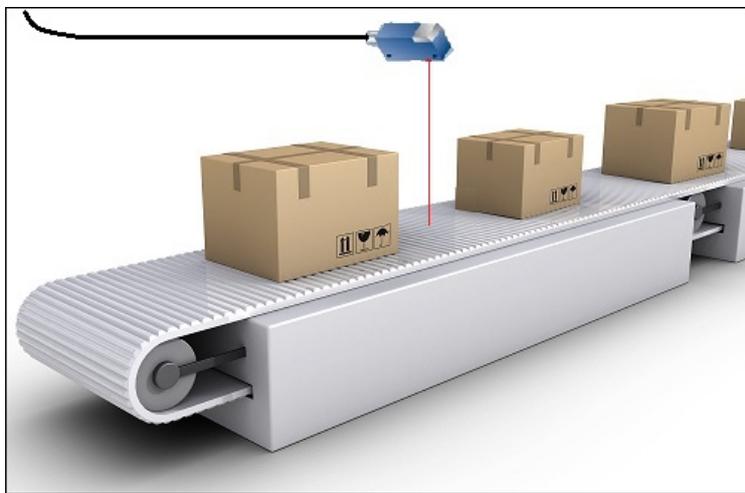


Figura 1.8 Modello di un sensore a diffusione diretta

Emittitore e ricevitore sono incorporati nella medesima unità.

I fasci ottici sono paralleli o leggermente convergenti. La presenza di un target nel campo ottico produce la diffusione e/o la riflessione del fascio luminoso sul ricevitore e la conseguente rilevazione dell'oggetto stesso.

C'è quindi una sostanziale differenza con i 2 casi precedenti: nei sensori a barriera e a catarifrangente la discriminazione target presente/assente è tutto sommato semplice in quanto il ricevitore in un caso rileva un segnale e nell'altro non ne rileva per niente.

Nei sensori a diffusione invece, sul ricevitore è sempre presente un segnale; dovrà essere quindi complicato l'algoritmo di discriminazione presenza/assenza in quanto non vi è una distinzione così netta tra i due casi.

La discriminazione della presenza o meno di un oggetto è demandata nel caso più semplice ad una misura dell'energia riflessa o diffusa dall'oggetto nel ricevitore della fotocellula.

Questo tipo di sensori, dovendo lavorare in condizioni di rapporto segnale rumore peggiorative rispetto ai casi precedenti (il raggio diffuso ha meno intensità di quello riflesso), ha campi di distanza operativa più ridotti che arrivano ad un massimo di 1 m circa (paragonato a qualche decina di metri per un sensore a barriera e 4/6 m per uno a catarifrangente. Il paragone è fatto per sensori “miniaturizzati”, cioè quelli che tipicamente sono inclusi in un parallelepipedo di circa 25x20x10 mm).

Per tutte e 3 le tipologie elencate finora, il ricevitore è composto da un singolo fotodiode che valuta la presenza del target a seconda della quantità di energia che riceve.

Le proprietà riflettenti e di diffusività del target, però, sono determinanti e influenzano la rilevazione e le performance del sensore: un oggetto chiaro riflette verso il ricevitore una maggiore energia, e viene pertanto rilevato ad una distanza maggiore di un oggetto scuro che riflette un’energia inferiore.

Un esempio di ciò che succede è mostrato nel seguente grafico.

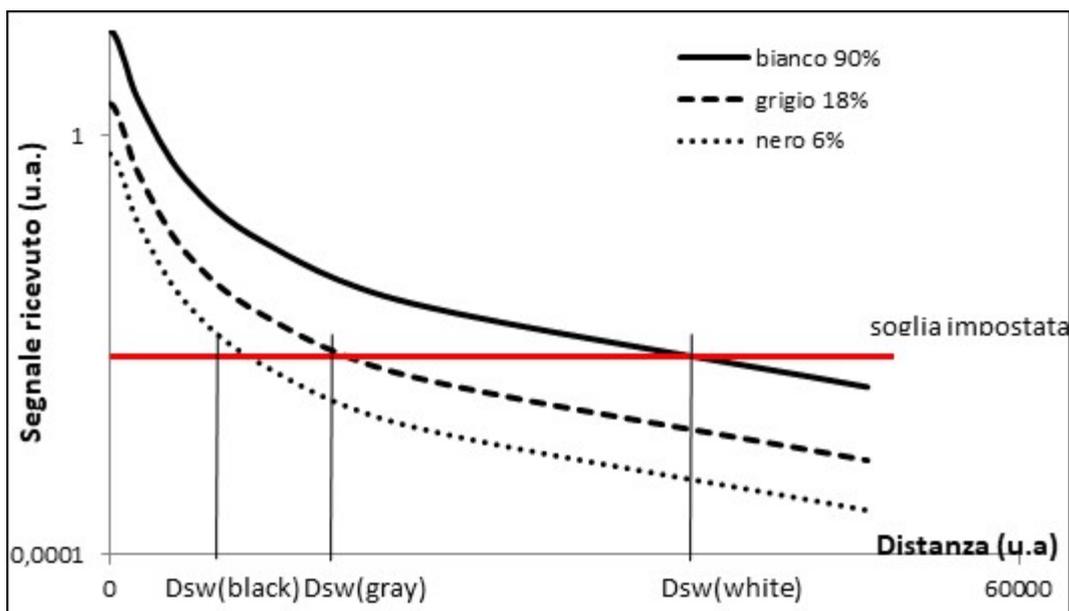


Figura 1.9 Proprietà di diffusività differenti per i 3 target

La funzione di prossimità descritta precedentemente è quindi fortemente influenzata dalla diffusività del target. In applicazioni nelle quali non può essere previsto il colore dell’oggetto da rilevare, questa funzione non offre buone performance di rilevazione.

È pertanto necessario studiare un sistema che rilevi la prossimità di un oggetto indipendentemente dal suo colore. Un metodo per ottenere questo risultato è rilevare la prossimità attraverso una stima di distanza.

La tecnica che si utilizza in questo caso è la triangolazione ottica ed i sensori che ne risultano sono detti a triangolazione o a soppressione di sfondo.

Il sistema di rilevamento è sensibile all'angolo di riflessione, che diminuisce in proporzione alla distanza dell'oggetto. Con semplici regole di trigonometria si riesce a risalire alla distanza del target essendo noti l'angolo di emissione e la distanza tra emettitore e rivelatore.

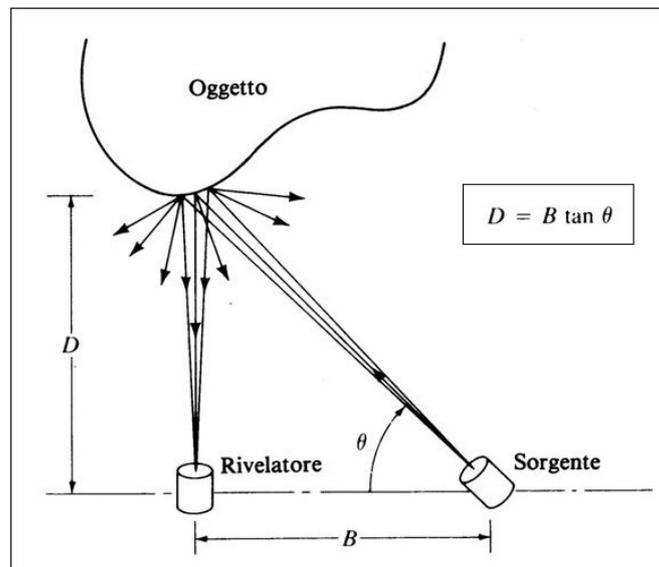


Figura 1.10 Funzionamento del principio della triangolazione

Nel sistema in figura, la sorgente può essere orientata: regolando la sua inclinazione si può fare in modo che il raggio diffuso dall'oggetto raggiunga il rivelatore solo quando la distanza prossima a quella indicata dalla formula in figura.

Con lo stesso angolo del fascio emesso, il raggio diffuso cadrebbe fuori dal rivelatore per distanze dell'oggetto sensibilmente diverse da quella impostata.

Nei sensori considerati, anziché controllare l'angolo di emissione, si tende a utilizzare un emettitore fisso e operare la stima della distanza tramite una regolazione del sistema di ricezione, che diventa pertanto più complesso.

Solitamente in sistemi di questo genere, un ricevitore composto da un singolo fotodiode non permette di ottenere performances che giustifichino il costo della maggiore complessità del sistema: si usano pertanto rivelatori con architetture più complesse.

Implementando la tecnica della triangolazione, si può raggiungere un buon livello di indipendenza della distanza di commutazione dal colore del target.

Tuttavia la distanza operativa risulta ridotta rispetto a quella attesa dai semplici sensori a diffusione.

Generalmente, per sensori con un interasse che misura 10÷15 mm si ottiene una distanza di intervento che si aggira intorno ai 200 mm.

1.4 Tipi di ricevitori

Considerando soltanto i sensori a soppressione di sfondo BGS, ci si concentra ora sul ricevitore e le diverse tecniche per realizzarlo.

- Doppio fotodiode + ottica fissa

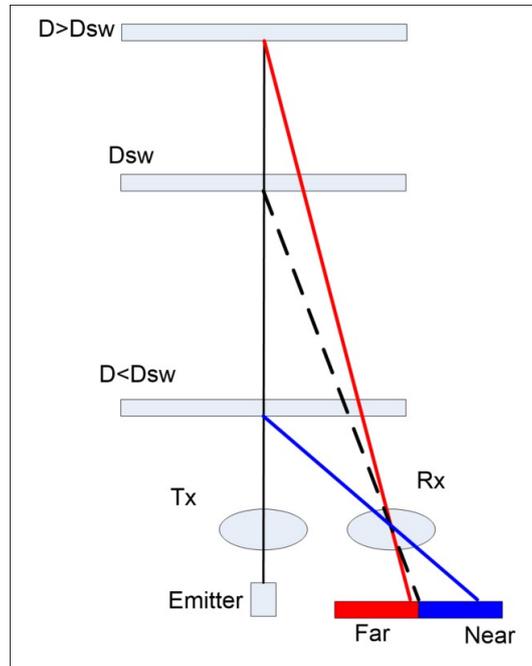


Figura 1.11 Modello di un sensore con lente fissa

Il ricevitore è composto da 2 fotodiodi, uno per la distanza vicina (near) e uno per la distanza lontana (far).

Il sensore presenta una distanza di switching D_{sw} fissa e definita geometricamente dal punto corrispondente all'attraversamento del raggio riflesso del confine tra i due fotodiodi.

In linea di principio, se il raggio riflesso viene rilevato dal fotodiode near, allora il target è ad una distanza minore di D_{sw} . Se il raggio riflesso è ricevuto dal fotodiode far, allora il target è ad una distanza maggiore di D_{sw} .

La stima della distanza in questo caso è operata in relazione alla distanza di switching (maggiore o minore di D_{sw}) e non in termini assoluti.

La distanza viene valutata attraverso la seguente relazione tra i valori di segnale in uscita della coppia di fotodiodi studiata appositamente per fornire un'uscita a gradino (comportamento ideale) :

$$distance = \frac{far - near}{far + near}$$

Questa funzione varia tra -1 (target vicino) e 1 (target lontano).

È in generale una funzione continua della distanza, che può essere resa simile ad una funzione a gradino ottimizzando il design ottico. Il denominatore di questa funzione normalizza il risultato, rendendolo indipendente dall'energia ricevuta (che è una funzione della distanza).

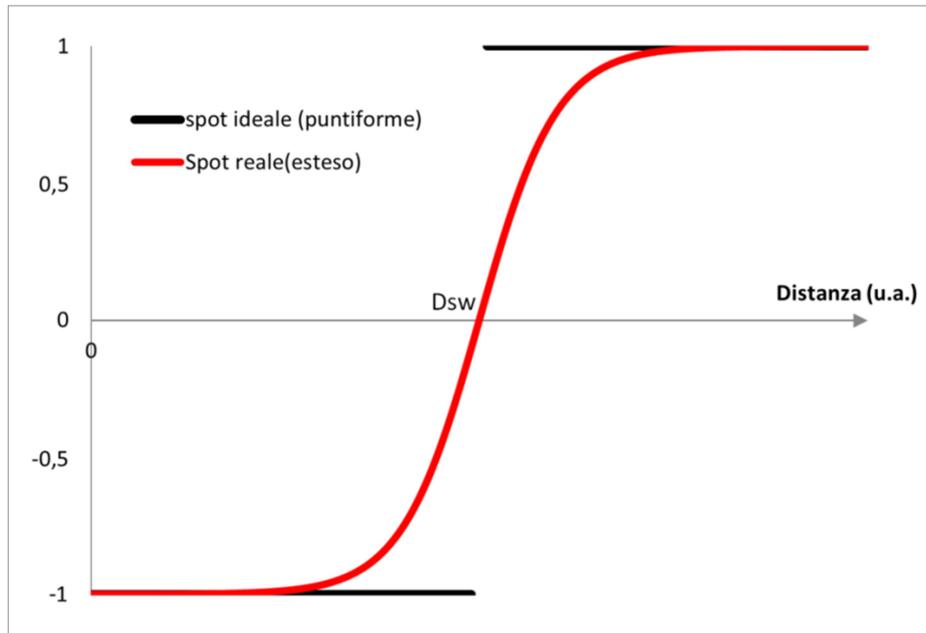


Figura 1.12 Andamento della distanza per uno spot ideale e uno reale

Sotto opportune ipotesi (spot puntiforme, energia sufficiente del segnale ricevuto) il sensore è in grado di fornire una stima della distanza indipendente dalla quantità di luce che investe il rivelatore e indipendente quindi dal colore dell'oggetto da rilevare.

Questo tipo di soluzione permette di essere integrata in sistemi relativamente semplici (e quindi a basso costo) in quanto il processing stesso è poco complesso: in linea di principio è sufficiente amplificare la differenza tra i due segnali in modo da arrivare alla saturazione ed avere distintamente un segnale alto o uno basso, senza dover calcolare il rapporto che in termini di complessità è generalmente oneroso.

In alternativa per avere una stima della distanza si può anche analizzare l'andamento nel tempo dei due segnali provenienti dai fotodiodi: lo spot cade maggiormente su quello che presenta una risposta più veloce.

Con un'approssimazione che talvolta è ragionevole, si può discriminare la distanza analizzando il solo segnale differenza e valutando l'istante (relativamente al segnale di emissione) in cui esso passa per lo zero.

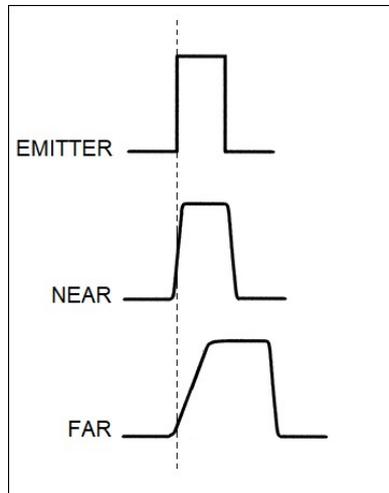


Figura 1.13 Esempio di spot che cade maggiormente sul fotodiodo near

Miglioramenti possono essere eseguiti per quanto riguarda l'ottica al fine di rendere tale transizione più rapida possibile.

Questo tipo di sensore nasce per una precisa distanza di switching: il suo principale svantaggio risiede nell'impossibilità di essere regolato sul campo e sulle limitate possibilità di taratura in fase di produzione.

- Doppio fotodiodo + lente mobile

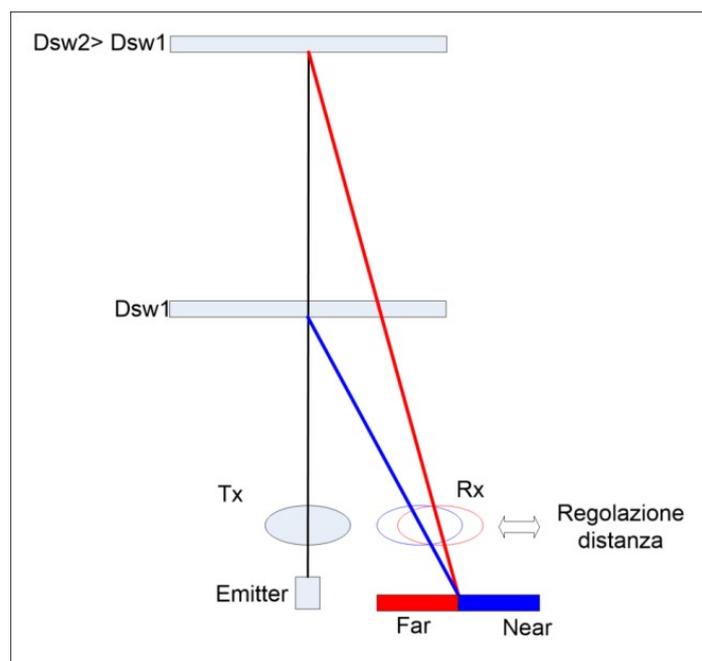


Figura 1.14 Modello di un sensore con lente mobile

Lo stesso principio di rilevazione può essere applicato ad un sistema più complesso, caratterizzato dalla possibilità di movimento relativo tra la lente di ricezione ed il doppio fotodiode. Operando questo movimento, è possibile regolare la distanza di switching, che rimane quella per la quale lo spot sul ricevitore attraversa il confine di separazione tra i due fotodiodi.

Al contempo, questo incremento di performance comporta un incremento del costo e della complessità del sistema meccanico.

In alternativa a questa soluzione si potrebbe movimentare il doppio fotodiode e lasciare fissa la lente.

Questa soluzione è raramente utilizzata in quanto complica notevolmente la connessione elettrica al doppio fotodiode, che è un aspetto critico della progettazione soprattutto per quanto riguarda il layout e la schermatura necessari a garantire una buona qualità del segnale. Infatti le correnti generate dal fotodiode possono essere dell'ordine dei nA. In queste condizioni il guadagno dell'amplificatore di transimpedenza può essere dell'ordine delle centinaia di k Ω o alcuni M Ω .

Con guadagni così elevati, è necessario che le possibili interferenze siano minimizzare e in questo contesto, layout e schermature stabili ed affidabili sono una necessità irrinunciabile.

- PSD (Position Sensitive Detectors)

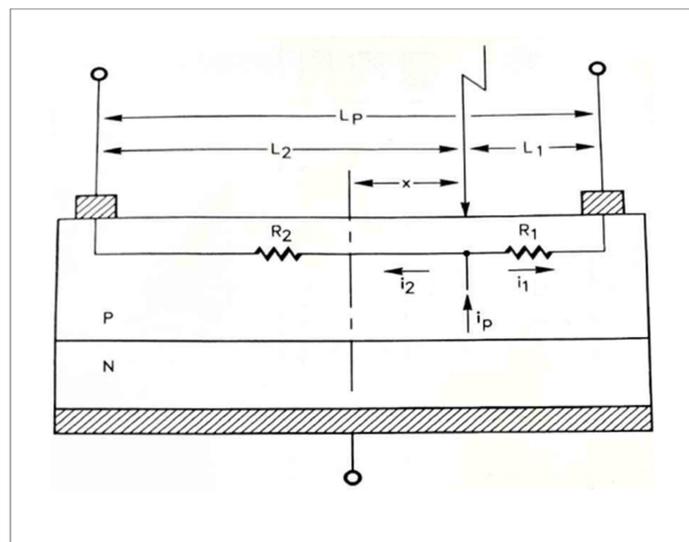


Figura 1.15 Schema di un PSD

Il ricevitore è realizzato come un fotodiode esteso costruito in modo che la resistività della zona P, all'estremità della quale sono collegati due elettrodi distanziati di una distanza L_p (anodi) sia più elevata di quella della zona N (che realizza il catodo).

Il suo funzionamento è basato sull'effetto fotoelettrico laterale. [3][4][5]

Facendo riferimento alla figura 1.14 si suppone che un fotone venga assorbito dal fotodiode ad una distanza x dal suo centro : la fotocorrente i_p generata per illuminazione viene divisa in due percorsi differenti tramite le resistenze R_1 e R_2 .

Si ricava:

$$i_1 = \frac{R_2}{R_{tot}} i_p \qquad i_2 = \frac{R_1}{R_{tot}} i_p$$

Supponendo che nella regione P vi sia una resistività ρ uniforme e che quindi $R_1 = \rho L_1$ e $R_2 = \rho L_2$, si ottiene:

$$i_1 = \frac{L_2}{L_p} i_p \qquad i_2 = \frac{L_1}{L_p} i_p$$

Da cui:

$$x = L_2 - L_1 = \frac{i_2 - i_1}{i_1 + i_2} L_p$$

La formula è la stessa del ricevitore a lente fissa con la differenza che qui si ricava la posizione dello spot sul PSD.

Per trovarla, si è ipotizzato che si voglia rilevare la posizione in cui un fotone incide sul PSD. Si può dimostrare che nel caso di uno spot esteso, vale la stessa formula, dove però x è la posizione del baricentro dello spot.

È possibile ricavare la relazione che lega tale informazione con la distanza del target.

Si schematizza il sistema nel seguente modo:

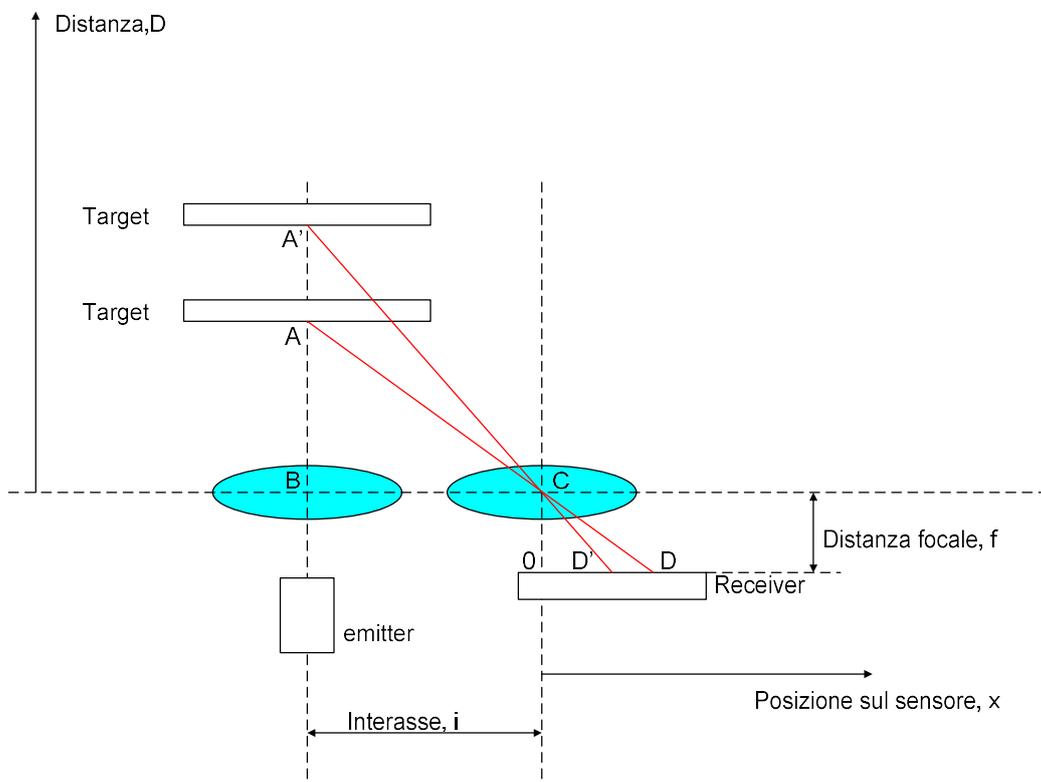


Figura 1.16 Schematizzazione sistema di lenti e ricevitore

Si individuano due coppie di triangoli simili: ABC/C0D e A'BC/C0D'.
 Impostando le dovute proporzioni si ottiene:

$$\frac{D}{i} = \frac{f}{x} \rightarrow x = \frac{fi}{D} \rightarrow x \propto \frac{1}{D} \quad (\text{Eq.1})$$

La posizione dello spot sul ricevitore è proporzionale all'inverso della distanza secondo una relazione iperbolica.

Esiste quindi una relazione continua e monotona che lega la distanza del target alla posizione dello spot sul trasduttore. Il sensore di prossimità può quindi essere realizzato eseguendo una stima continua della distanza del target, e quindi elaborando il risultato, si può costruire lo stato dell'uscita. Uno schema di principio è presentato di seguito:

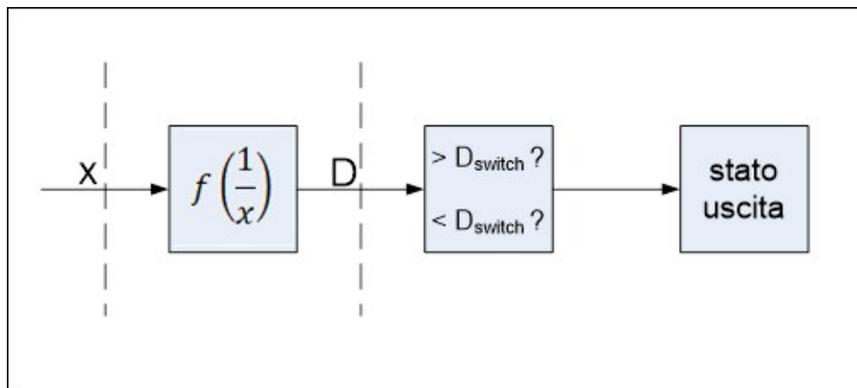


Figura 1.17 Schema di principio di un sensore di prossimità

Un sensore costruito in questo modo non ha parti in movimento e consente al contempo di regolare la distanza di switching.

In particolare, essendo la regolazione puramente elettronica, può essere implementato un algoritmo che implementa la cosiddetta funzione di teach, che consiste nell'auto apprendimento della distanza di regolazione, che dal punto di vista dell'utente, consiste nella semplice pressione di un tasto.

Si calcola ora la precisione della regolazione.

Differenziando ambo i membri si ottiene informazione riguardo la risoluzione:

$$dD = -\frac{fi}{x^2} dx$$

Il segno (-) indica come, nel sistema di riferimento scelto, ci sia una relazione inversa tra D e x: se la distanza aumenta, la posizione dello spot sul ricevitore diminuisce e viceversa.

La risoluzione in distanza è proporzionale a $\frac{1}{x^2}$, quindi:

- per distanze vicine (quando x è grande) si ha una buona risoluzione, cioè si riescono a risolvere piccoli spostamenti;
- per distanze lontane (quando x è piccolo) si ha una scarsa risoluzione, cioè il minimo spostamento che è possibile risolvere aumenta progressivamente con l'avvicinarsi dello spot all'origine, ovvero man mano che la distanza diverge.

Un altro modo di gestire il PSD è attraverso null method : si applicano due guadagni diversi alle correnti in uscita in modo che, alla distanza che interessa, la differenza tra i segnali amplificati sia nulla.

È un metodo più semplice in quanto non passando attraverso il calcolo della distanza, può essere implementato con componenti ed algoritmi relativamente semplificati.

Tuttavia la performance di un prodotto fatto in questo modo è il risultato di una serie di compromessi tra semplicità, accuratezza e regolabilità ed è difficile enfatizzare una caratteristica senza influire sulle altre.

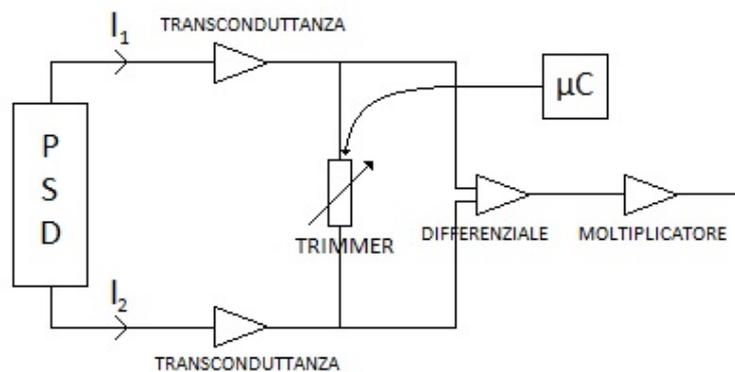


Figura 1.18 Schema a blocchi della gestione null method del PSD

Costo contenuto e capacità di fornire dati in maniera continua costituiscono i maggiori vantaggi del PSD.

Inoltre è in grado di raggiungere una risoluzione teoricamente infinita e limitata solamente dal rumore.

Di contro, per effettuare la stima della distanza è necessario eseguire una divisione. Questa, se eseguita in modo analogico, è difficile e costosa da implementare in modo stabile e con un adeguato range dinamico. Se implementata tramite un microcontrollore, richiede solitamente un core con la divisione (che però tipicamente è costoso), o la realizzazione della divisione via software (che è onerosa in termini di tempo).

- Sensori lineari d'immagine
 - CCD
 - CMOS
 - Array di fotodiodi

Un altro modo per procedere alla stima della distanza è quello di utilizzare un sensore lineare di immagine. Come nel caso del PSD, un sensore di immagine può rilevare la posizione dello spot (e quindi la distanza del target), ma le tecniche che si usano in questo caso sono assimilabili a quelle di elaborazione dell'immagine. È inoltre possibile operare analisi altrimenti irrealizzabili con i sistemi precedenti (ad esempio, stima della forma o della dimensione dello spot, presenza di spot multipli, ecc.) aprendo alla possibilità di sensori con prestazioni avanzate, o con maggiore affidabilità.

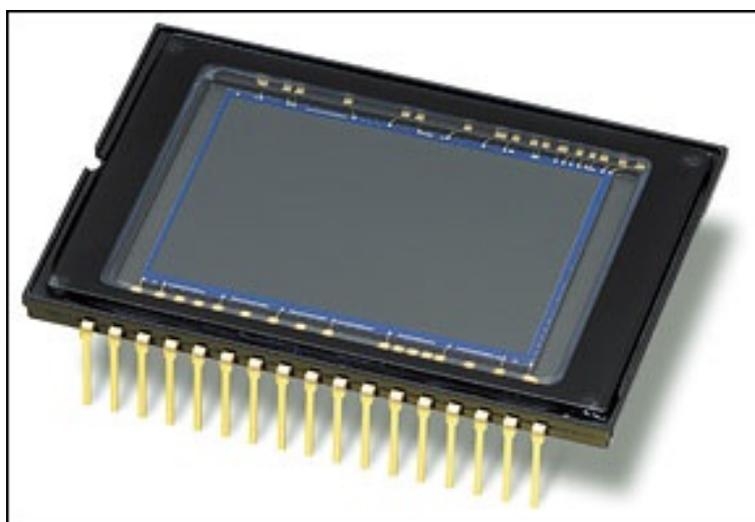


Figura 1.19 Esempio di un CCD in commercio

Il CCD (Charge Coupled Device) è un tipo di sensore lineare d'immagine costituito da una matrice di condensatori MOS.

Una parte di essi è realizzata in modo da generare una carica proporzionale all'energia luminosa ricevuta (questi sono detti pixel) [6].

Tali condensatori sono accoppiati ai rimanenti condensatori MOS, che sono invece opachi(non sono sensibili alla luce) e realizzano una sorta di shift register analogico: tramite un'opportuna serie di impulsi di temporizzazione, è possibile trasferire la carica tra un condensatore e l'altro.

La loro lettura e acquisizione sequenziale permette di ottenere l'immagine.

Per rilevare la distanza è necessario analizzare l'immagine proiettata sulla superficie del CCD : si invia un treno di impulsi che permette lo shift di ogni colonna di pixel in uscita, dove esternamente viene collegato un amplificatore.

Nello specifico esistono 3 metodi differenti di trasferimento della carica che identificano altrettante architetture diverse:

Full frame transfer: il CCD è composto interamente da pixel sensibili alla luce, la lettura dell'immagine avviene mediante trasferimento verticale del contenuto delle righe della matrice e poi tramite shift register si invia l'informazione in uscita. È l'architettura che occupa meno spazio ma allo stesso tempo quella che richiede maggior tempo di readout;

Frame transfer: il comportamento è medesimo con la differenza che il CCD presenta due aree identiche, una sensibile alla luce e l'altra (che funge da memoria) schermata da una lamina metallica quindi l'immagine prima di essere inviata in uscita viene replicata in una regione uguale a quella di illuminamento. L'architettura è molto sensibile alla luce ma richiede attenzione e schermatura durante il trasferimento dei pixel in quanto potrebbero venire "sporcati" da illuminamenti indesiderati.

Interline Transfer: in questa architettura ad ogni colonna della matrice è associata una colonna adiacente non fotosensibile e che ricopre il ruolo di memoria. I pacchetti di cariche vengono spostati nei registri verticali per poi essere trasferiti riga per riga allo shift register di lettura. È l'architettura più veloce ma, a parità d'area, anche quella meno sensibile in quanto solo metà della matrice è adibita a convertire il segnale luminoso incidente.

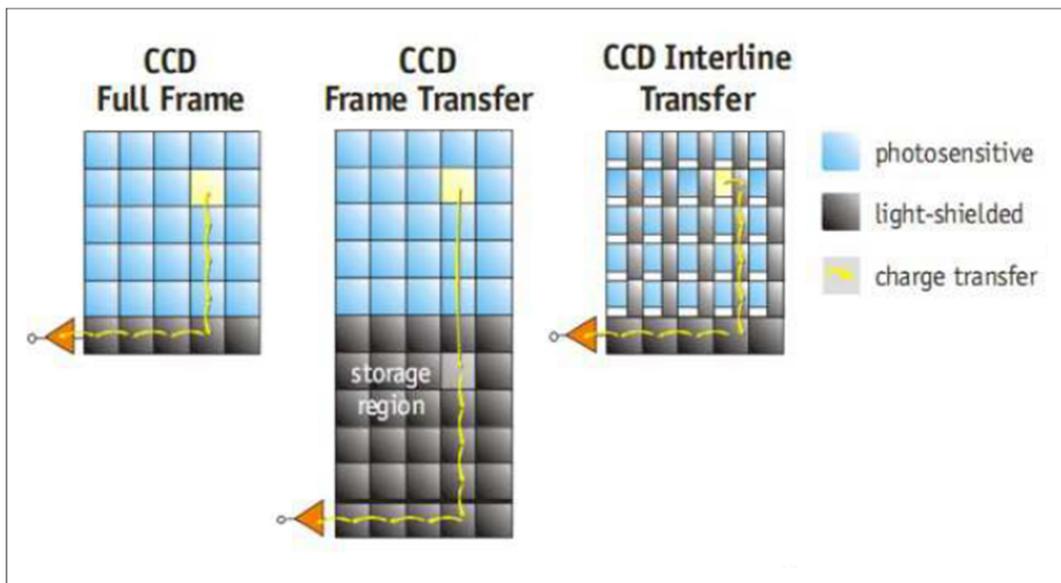


Figura 1.20 Architetture differenti per un sensore d'immagine CCD

Il principale vantaggio risiede nella ridotta capacità d'uscita dovuta alla presenza di un unico stadio di amplificazione, aumentando il rapporto segnale-rumore.

Diverso il discorso per i sensori attivi CMOS, sempre composti da matrici di pixel ma con l'aggiunta per ogni singolo pixel di un amplificatore di segnale integrato. Una volta che la luce raggiunge la matrice, ogni singolo pixel la converte in una tensione amplificata che verrà elaborata dal resto del sensore: l'uscita quindi è una tensione.

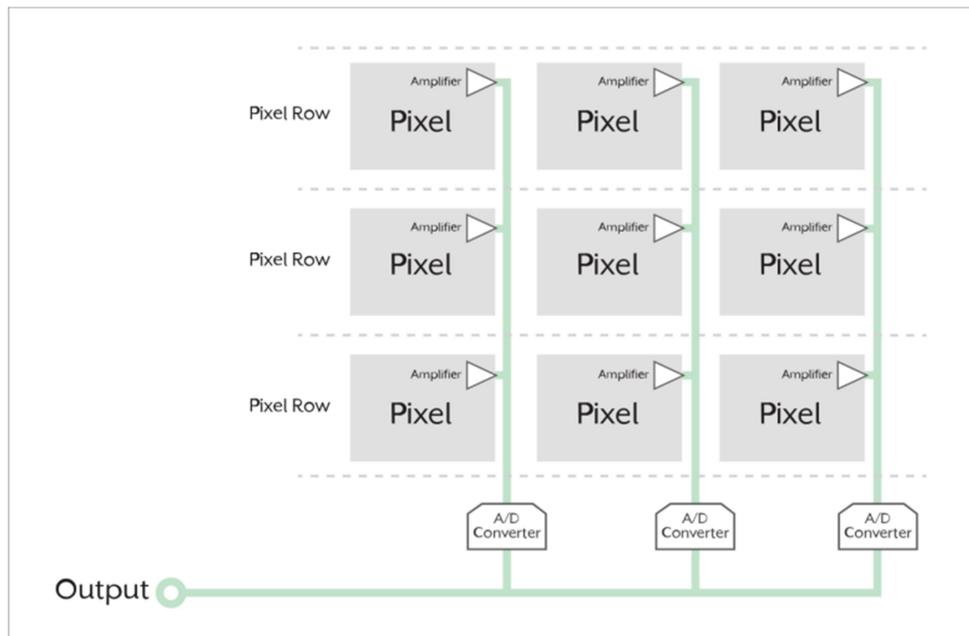


Figura 1.21 Schema a blocchi di un sensore CMOS

Avendo un gestione di tipo parallelo, il tempo di readout dell'uscita è relativamente basso permettendo quindi utilizzi a banda larga. Di contro però si ottiene un'immagine meno uniforme e di qualità inferiore siccome ogni pixel esegue una conversione in loco.

Una sottofamiglia dei sensori CMOS sono gli array di fotodiodi, che, come dice la parola, non sono composti da una matrice di righe e colonne, ma sono costituiti da una singola linea di pixel.

Si arriva a questa soluzione come rimedio al problema meccanico della lente mobile: anziché utilizzare il doppio fotodiodo se ne utilizzano molteplici disposti adiacenti su un array.

Ogni singolo fotodiode (o pixel) fornisce quindi in uscita un valore di tensione proporzionale all'intensità della radiazione elettromagnetica che lo colpisce.

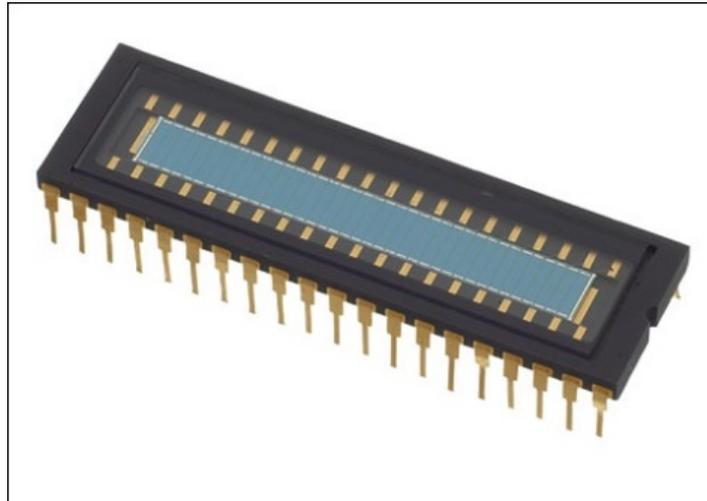


Figura 1.22 Modello di array di fotodiodi in commercio

Il grosso vantaggio qui è che l'uscita è già "digitalizzata" semplificando notevolmente la struttura in cui viene utilizzato l'array: il fulcro della struttura è composto dal microcontrollore che si occupa di gestire i segnali utili al trasmettitore ed elaborare quelli provenienti dall'array.

Riassumendo, in virtù della più semplice e flessibile struttura e della minor difficoltà nel trattare i segnali e quindi di determinare la posizione di caduta del fascio luminoso, si è scelto di approfondire il ricevitore basato su array di fotodiodi.

Si prosegue allora la trattazione analitica considerando solamente l'array di fotodiodi come tipo di ricevitore e prendendo come sistema di riferimento quello indicato in figura 1.15.

In particolare infatti in un array uniforme $x = kp$ con k intero e compreso tra 1 e il numero di pixel scanditi nel percorso completo del fascio.

Allora si può riscrivere eq.1 come:

$$D = \frac{fi}{kp}$$

In un sistema realizzato con sensori di immagine, la posizione dello spot sul sensore è "quantizzata", ovvero è possibile identificare tra quali pixel cade lo spot a meno del pitch.

Si supponga di avere un array uniforme avente N pixels ognuno con pitch costante pari a p : la lunghezza totale dell'array è pari a Np .

Nel sistema di riferimento descritto, è possibile definire una funzione di quantizzazione q come segue:

$$\begin{aligned} q_x: [0, Np] &\rightarrow \{1, \dots, N\} \\ x \in (kp, (k+1)p] &\rightarrow k \end{aligned}$$

La quantizzazione della stima della posizione dello spot nel sensore, induce una quantizzazione nella stima della distanza in virtù di

$$D = \frac{fi}{kp}$$

Che corrisponde a

$$\begin{aligned} q_D: [0, \infty] &\rightarrow \{D_1 \dots D_n\} \\ D \in \left(\frac{fi}{(k+1)p}, \frac{fi}{kp} \right] &\rightarrow D_k = \frac{fi}{kp} \end{aligned}$$

Come si può notare, mentre la successione delle posizioni quantizzate k dello spot sull'array è lineare e a gradino (con gradini costanti), la successione delle distanze quantizzate D_k non è lineare e l'ampiezza dei gradini è inversamente proporzionale a k .

In questo sistema, è definita una distanza massima per $k=1$ pari a $D_1 = \frac{fi}{p}$ ed una distanza minima per $k=N-1$ pari a $D_{N-1} = \frac{fi}{(N-1)p}$.

L'ipotesi di spot puntiforme implica due considerazioni:

- è sempre possibile definire in quale pixel è caduto lo spot. Poiché la dimensione dello spot è infinitesima rispetto alla dimensione del pixel, sono impossibili configurazioni dove lo spot cade tra due pixel adiacenti, oppure dove lo spot occupa più di un pixel;
- in virtù di quanto detto sopra, non sarà mai possibile interpolare la posizione dello spot all'interno di un pixel.

Lo spot uniforme nella pratica è irrealizzabile e quindi tale ipotesi non si verifica mai, ma è utile per costruire un modello matematico.

Nella pratica lo spot sarà esteso e sarà necessario valutarne il baricentro. Questo renderà possibile anche l'interpolazione della posizione all'interno di un singolo pixel.

Nell'ipotesi di spot puntiforme, si può calcolare quale sia il minimo spostamento apprezzabile dal sensore

$$\Delta D_k = D_k - D_{k+1} = \frac{fi}{p} \left(\frac{1}{k} - \frac{1}{k+1} \right) = \frac{fi}{p} \frac{1}{k(k+1)} \quad (\text{Eq.2})$$

Si noti che per $k=1$ si ha

$$D_1 = \frac{fi}{p}$$

$$\Delta D_1 = \frac{fi}{2p}$$

Tale coppia di equazioni indica come in assenza di interpolazione, la massima distanza rilevabile può essere misurata con un'accuratezza del 50%.

Nella pratica questa performance può essere migliorata agendo sulla dimensione dello spot e facendo in modo che alla massima distanza esso interessi più pixel. In questo modo tramite l'interpolazione la risoluzione può essere migliorata.

Le stesse equazioni riscritte per $k=N$ forniscono:

$$D_N = \frac{fi}{Np}$$

$$\Delta D_N = \frac{fi}{p} \frac{1}{N(N+1)}$$

Nella pratica si ha che $D_N=D_{\min}$ può essere dell'ordine di alcuni mm, mentre N può valere tra alcune decine e un centinaio di pixel.

Risulta quindi che, nei dintorni della minima distanza, il minimo spostamento apprezzabile risulta dell'ordine di $10^{-3} \div 10^{-4}$ volte D_{\min} .

Seppur notevole, tale performance nella pratica è di scarsa utilità. L'ottima precisione teorica che si può raggiungere con un sensore realizzato in questo modo, viene completamente vanificata dalle seguenti considerazioni:

- il sensore deve essere montato su una macchina automatica. Esso può essere sottoposto a vibrazioni, e la sua performance non ne deve essere influenzata → una risoluzione eccessiva può rendere il sensore sensibile agli spostamenti indotti dalle vibrazioni generando false commutazioni;
- le tolleranze costruttive del sensore (costruzione ed assemblaggio dei componenti optomeccanici che lo compongono) sono di alcuni ordini di grandezza superiori a ΔD_N : la risoluzione calcolata, nella pratica non potrà mai essere raggiunta.

L'elevata risoluzione dell'array da vicino si traduce quindi nella pratica, nella necessità di acquisire ed elaborare un numero elevato di pixel il cui contenuto informativo è poco utilizzabile.

Il sistema di misura quindi presenta una risoluzione scarsa per distanze elevate e sovrabbondante per distanze ridotte.

1.5 L'array di fotodiodi IC-LFM

Il modello di array di fotodiodi utilizzato è IC-LFM prodotto dall'azienda tedesca IC-Haus.

È un sensore lineare d'immagine ad alta sensibilità composto da 64 fotodiodi (chiamati pixel) disposti in un'unica riga [7].

I fotodiodi presentano un pitch (larghezza lungo l'asse dell'array) di $63,5 \mu\text{m}$ permettendo così una risoluzione di 400 DPI (Dots Per Inch).

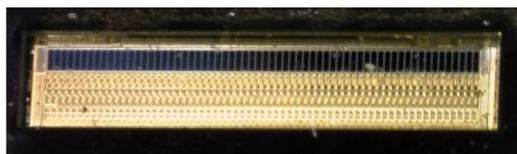


Figura 1.23 Vista microscopica dell'array IC-LFM

Per convertire il segnale ottico in segnale elettrico, essi attuano un'integrazione nel tempo della radiazione elettromagnetica che li colpisce.

Le principali caratteristiche del sensore sono:

- elevata frequenza di funzionamento fino a 5 MHz;
- tempo di integrazione flessibile e definibile dall'utente;
- presenza di un amplificatore di uscita push-pull che rende l'uscita analogica glitch-free;

Di seguito si riporta lo schema a blocchi del circuito interno.

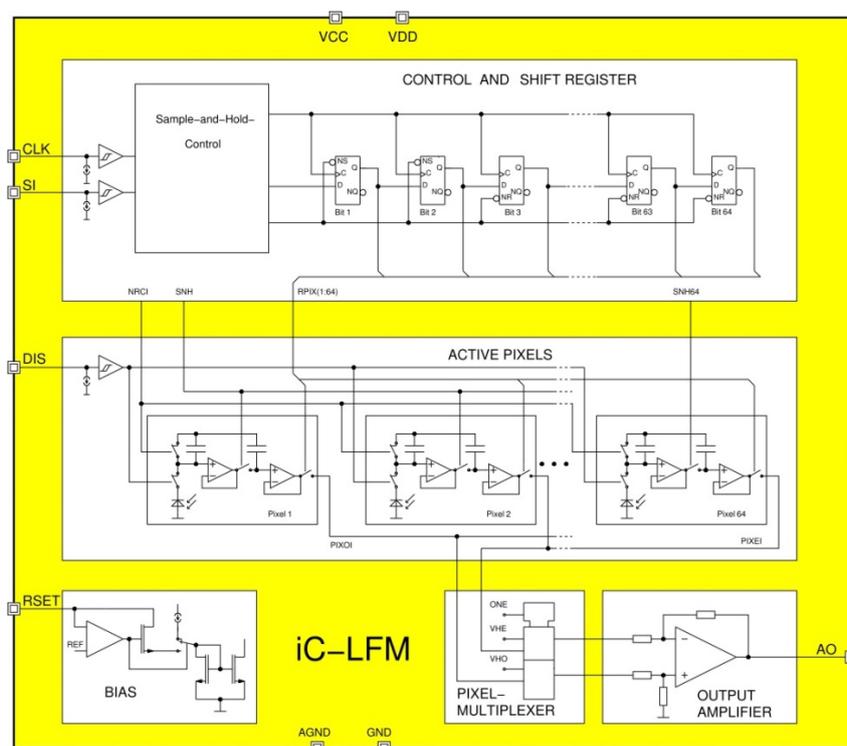


Figura 1.24 Schema a blocchi del circuito interno del sensore

All'interno è presente:

- un controllo sample & hold che permette di decidere se campionare un nuovo valore dei pixel o mantenere il valore attuale dei pixel;
- uno stage speculare per ogni pixel composto da operazionali in configurazione integratore e condensatori utili al sample & hold;
- un circuito di bias per la gestione del reset;
- un multiplexer che permette l'invio sequenziale di ogni pixel in uscita;
- un amplificatore d'uscita che gestisce il valore analogico risultante.

La configurazione dei pin invece risulta:

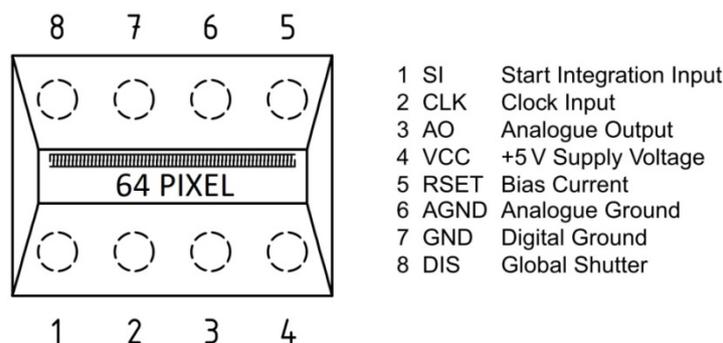


Figura 1.25 Pin-out del sensore IC-LFM

La temporizzazione dell'intero sistema è cadenzata dal segnale di clock; con il segnale $\overline{\text{DIS}}$ si definisce l'ampiezza della finestra d'integrazione mentre con il segnale SI si dà il via ad un ciclo di readout, disponibile dal fronte di discesa del CLK successivo all'impulso di SI.

Sul segnale AO di uscita quindi si prelevano i valori di ogni pixel in maniera sequenziale, come mostrato in figura:

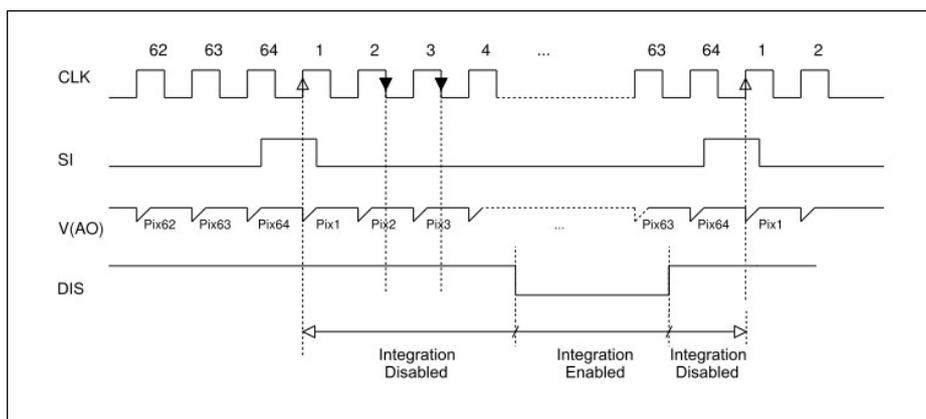


Figura 1.26 Forme d'onda caratteristiche dell'array

È quindi intrinseco che la durata di un ciclo di lettura è vincolata a 64 colpi di clock e questo, come si vedrà più avanti, rappresenta il limite principale della tecnologia ad array di fotodiodi.

1.6 Il sistema integrato PSoC

La gestione dei segnali utili all'array di fotodiodi e più in generale la programmazione dell'intero sistema è stata affidata al development kit PSoC della Cypress [8].

Più precisamente è stata utilizzata la versione di 4^a generazione denominata CY8CKIT-042.

Grazie alla combinazioni di blocchi analogici e digitali programmabili e configurabili, il PSoC è un sistema estremamente flessibile e, quindi, indicato per una vasta gamma di applicazioni embedded.

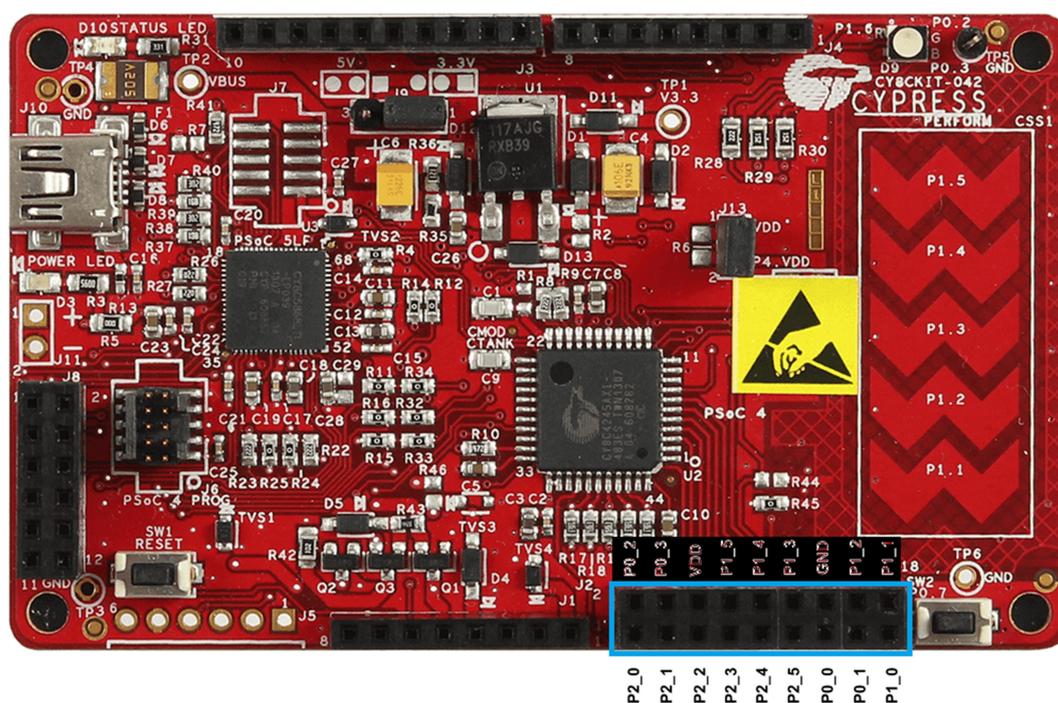


Figura 1.27 Aspetto della scheda PsoC 4

Nel riquadro in blu sono indicati i pin di I/O coi rispettivi nomi che verranno utilizzati per interfacciare il sensore alla scheda PSoC.

Tra le principali periferiche integrate nel microcontrollore, si evidenzia:

- un microprocessore da 32 bit Arm Cortex-M0 a 48Mhz;
- un ADC ad approssimazioni successive da 12 bit;
- due amplificatori operazionali a banda elevata;
- due comparatori a basso consumo;
- blocco per la comunicazione seriale (UART, SPI o I2C);
- quattro contatori PWM da 16-bit.

Per programmare la scheda, Cypress fornisce un sistema di sviluppo integrato (IDE) proprietario chiamato “PSoC Creator”.

Oltre ad aver il classico compilatore per la programmazione del microcontrollore, estremamente interessante all’interno del progetto è la finestra denominata “TopDesign”: qui è consentito istanziare e configurare i blocchi rappresentanti le periferiche interne al microcontrollore.

Il sistema di sviluppo genera automaticamente i driver delle periferiche (API), permettendo quindi di programmarle più semplicemente in via grafica ed evitando numerose righe di codice.

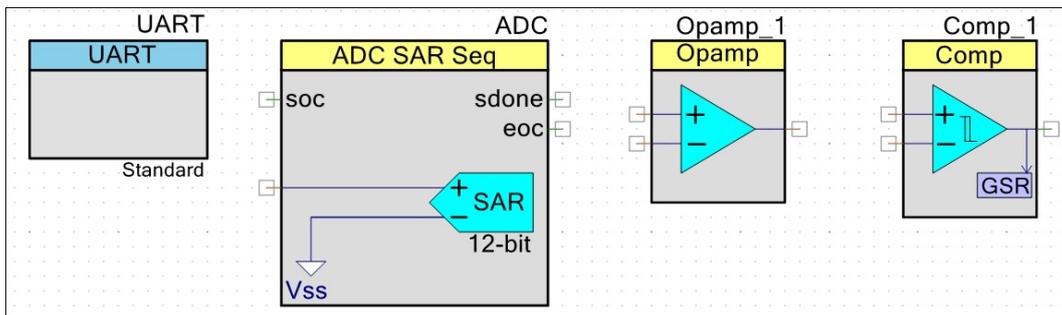


Figura 1.28 Esempio di API a disposizione

È presente inoltre una finestra per assegnare graficamente ogni variabile o pin di input/output utilizzata al pin fisico della scheda.

La scheda è collegata al computer tramite connessione USB e si utilizzerà una comunicazione UART per prelevare i dati dall’array e elaborarli su foglio elettronico.

Per usufruire del protocollo UART, è richiesto un collegamento che metta in comunicazione PSoC 4 con PSoC 5LP integrato (usato per l’interfacciamento tra PSoC 4 e porta USB): si crea un ponte tra il pin P0[5] UART TX di PSoC4 e P12[6] UART RX di PSoC 5LP e tra P0[4] UART RX di PSoC 4 e P12[7] UART TX di PSoC 5LP, come mostrato in figura:

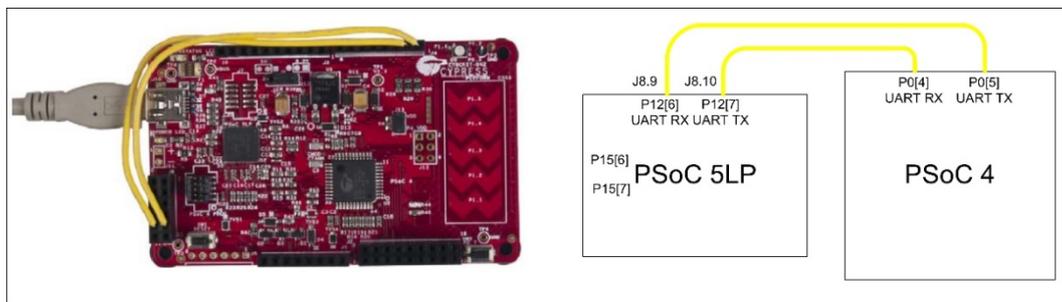


Figura 1.29 Collegamento a ponte per comunicazione UART

Si passa ora al lavoro vero e proprio svolto sull'array, andando ad esporre la serie di prove, misure e considerazioni eseguite per realizzare un sensore di prossimità BGS che fornisca una stima della distanza del target col miglior compromesso velocità/accuratezza.

Da specificare che l'array è stato in precedenza integrato in una struttura (di tipo tubolare) che comprende l'emettitore LED e una coppia di lenti (una per la trasmissione e una per la ricezione) aventi caratteristiche ottiche appositamente studiate per questo tipo di applicazione.

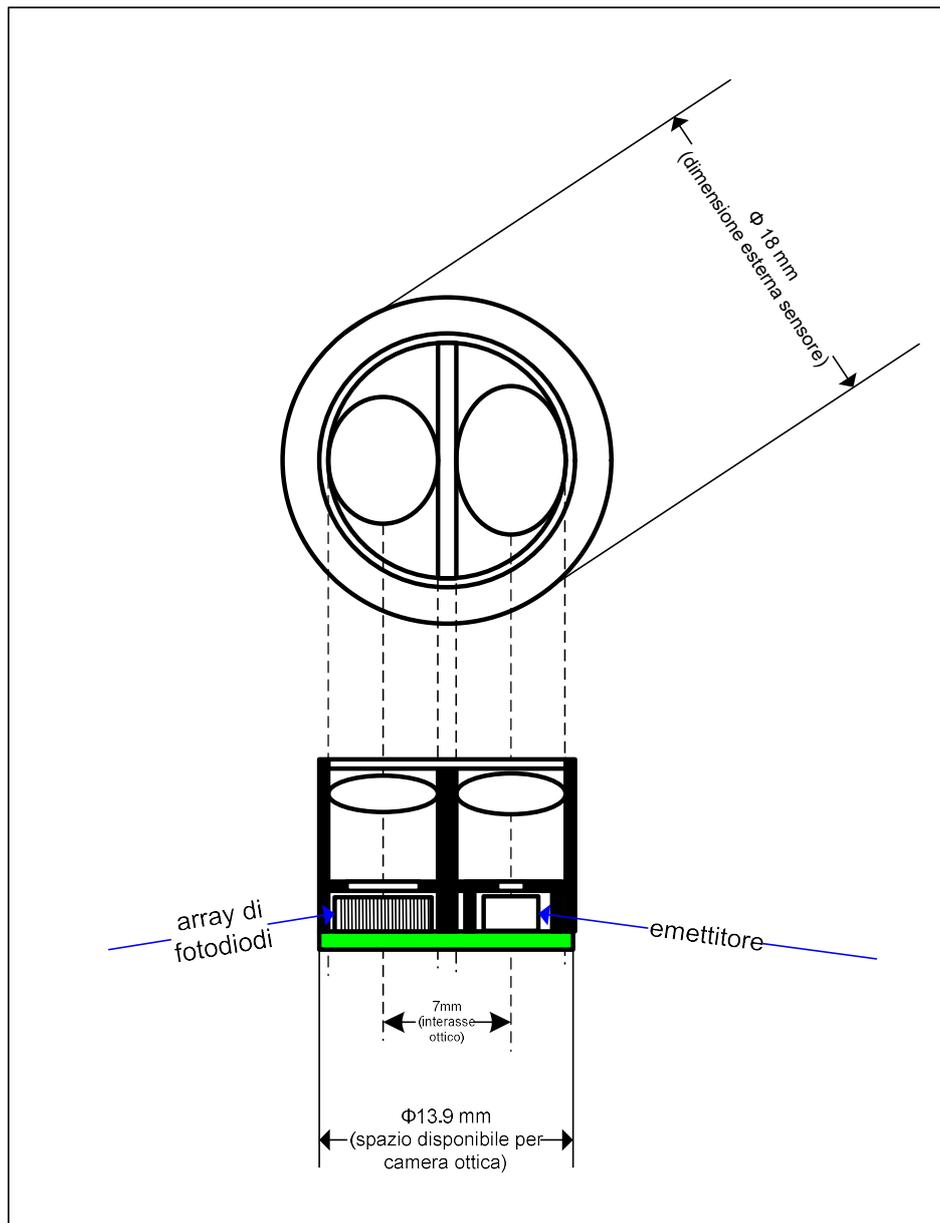


Figura 1.30 Vista frontale e in sezione della struttura in cui viene integrato l'array

Come si può notare anche dalle quote riportate, la struttura non presenta ampi gradi di libertà: vincoli ottici e meccanici (il sensore ha un diametro totale di soli 18 mm) fanno sì che la posizione dell'array in ricezione non possa essere modificata.

2. Scansione totale

Il primo passo è stato effettuare una scansione totale dell'array di fotodiodi allo scopo di comprendere a pieno il suo funzionamento.

Per facilitare le acquisizioni della distanza, si è fissata la scheda PSoC ad un supporto collegato ad un calibro digitale, permettendo di effettuare misure per una distanza massima di 190 mm.

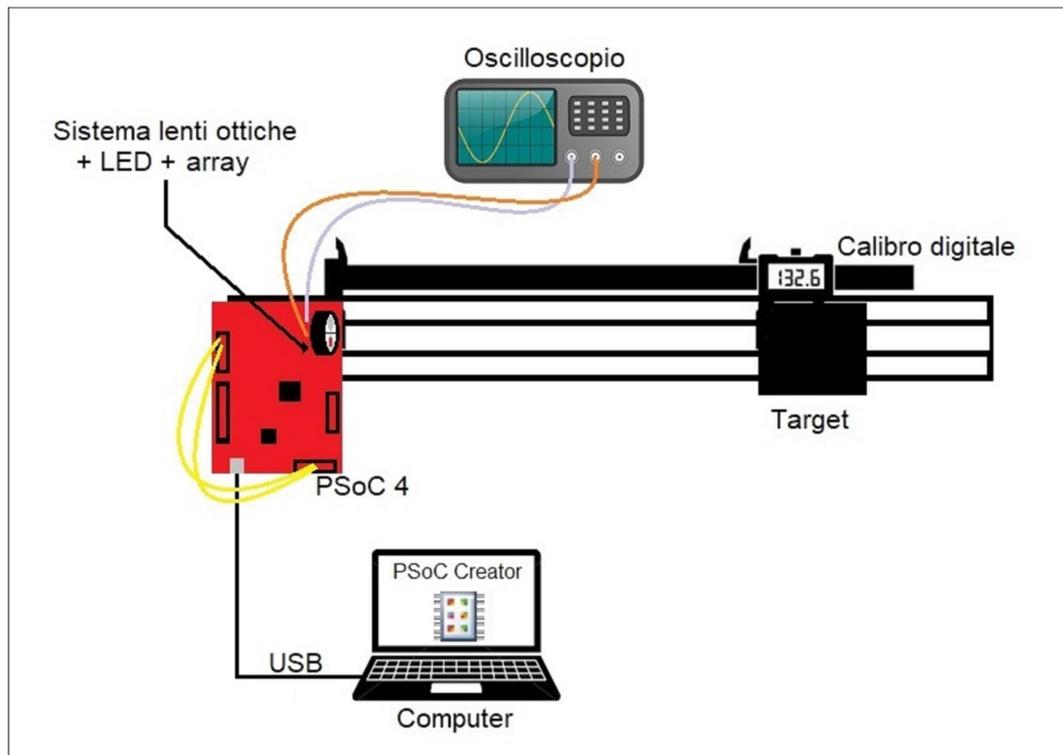


Figura 2.1 Sistema di misura

Il resto del sistema di misura comprende l'utilizzo di un oscilloscopio digitale per valutare l'andamento nel tempo dei segnali desiderati e l'interfacciamento USB tra scheda PSoC e PC con doppia funzione: programmazione scheda PSoC e scambio dati (tramite protocollo UART).

Per questo tipo di misure vengono tipicamente utilizzati 3 target standardizzati che rappresentano le casistiche limite di un generico oggetto dal punto di vista cromatico : bianco90%, grigio18% e nero6%.

La percentuale rappresenta la quantità di luce che il target è in grado di riflettere. Con questi target è anche possibile caratterizzare il comportamento di un sensore in modo univoco.

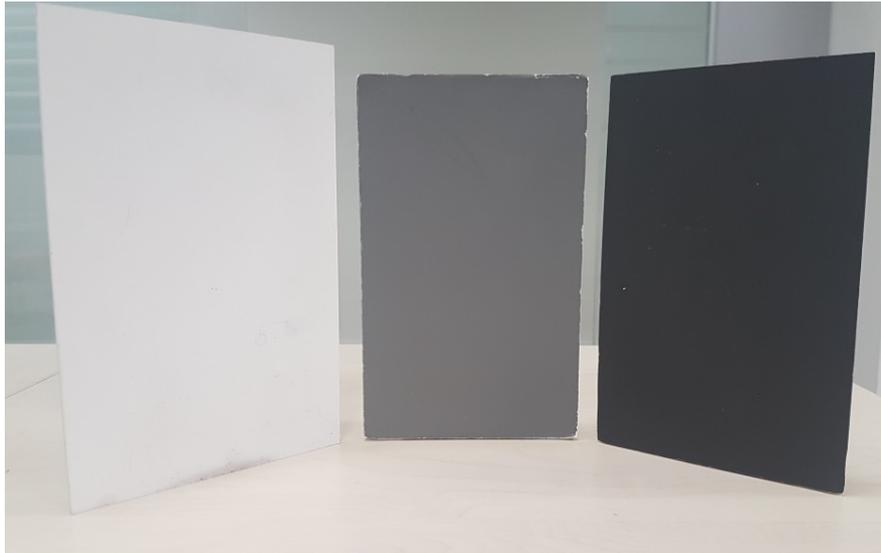


Figura 2.2 Target B90%, G18% e N6%

Effettuando una prima verifica approssimativa dell'andamento dell'uscita dell'array, si è notato che intorno al valore di 2V essa subisce un netto troncamento finendo, quindi, in saturazione, rendendo impossibile l'esecuzione di una misura analogica.

Per evitare che l'uscita saturi o che assuma valori troppo ridotti per poter essere elaborati, si è reso necessario prevedere che il LED di trasmissione emetta con 3 step diversi, a seconda del livello massimo del segnale d'uscita: 2 μ s, 4 μ s e 12 μ s. Per eseguire questa regolazione si è implementata la macchina a stati rappresentata schematicamente di seguito.

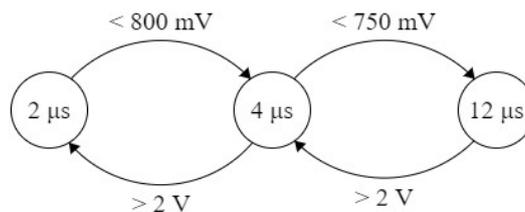


Figura 2.3 Diagramma a stati dei valori assunti da Ton

In altre parole, più il target è distante, meno è il segnale che torna indietro all'array, più l'emettitore deve aumentare il tempo di emissione e viceversa.

2.1 Programmazione microcontrollore

Di seguito si riporta il “TopDesign” per la programmazione del microcontrollore utile alla gestione dei segnali dell’array di fotodiodi.

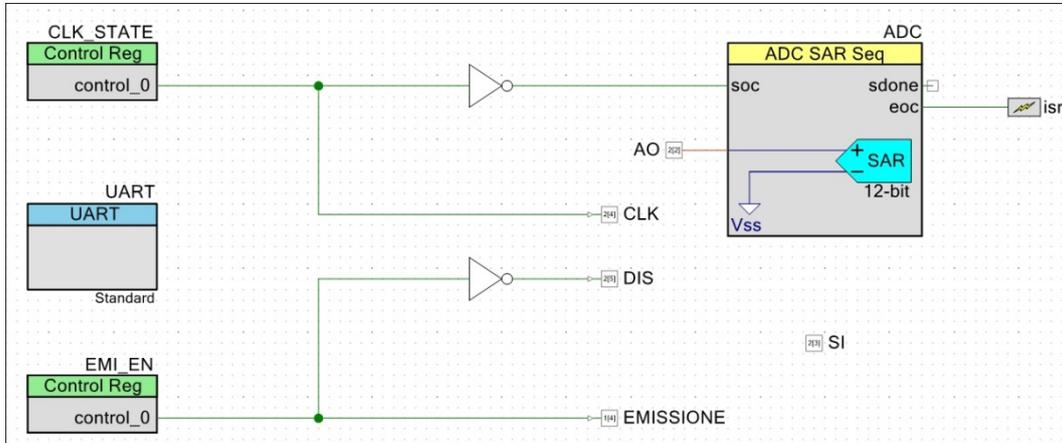


Figura 2.4 TopDesign per la scansione totale

Come anticipato, si utilizzerà la comunicazione UART, quindi è necessario inserire il rispettivo blocchetto impostando modalità di sola trasmissione, 115200 bps di baud rate , 8 bit di dato, nessuno di parità e uno di stop.

Tutti i segnali vengono generati via software: si aggiungono i registri per il controllo dell’emissione del LED e del clock.

Il segnale $\overline{\text{DIS}}$ (attivo basso) viene collegato al segnale di emissione interponendo una NOT: la finestra di integrazione corrisponde al periodo di emissione.

Infine si aggiunge l’ADC ad approssimazioni successive per campionare la tensione analogica AO in uscita su ogni fronte di discesa del clock.

Il termine della conversione (eoc) da parte dell’ADC attiva l’interrupt che permette di salvare il valore convertito su un array di interi creato appositamente.

Per completezza, si riporta una tabella che indica per ogni segnale a quale pin fisico della scheda PsoC (si veda figura 1.26) viene assegnato.

Questa configurazione rimane immutata per tutte le prove effettuate.

EMI	P1[4]
DIS	P2[5]
CLK	P2[4]
SI	P2[3]
AO	P2[2]

Tabella 1 Configurazione pin

Si prosegue ora con il codice di programmazione del microcontrollore.

```

1  #include "project.h"
2  #include <stdio.h>
3  #include <math.h>
4
5  void CLOCK1();
6  void CLOCK_rpt();
7  void checks();
8  void find_max();
9
10 int i=0,v_max=0,pix_max=0,emi=1;
11 int result[64]; //array che contiene il risultato dei campionamenti
12 int voltage[64]; //array che contiene le tensioni dei pixel campionati
13 char v_buff[256]; //stringa di caratteri per l'invio UART
14 char i_buff[120]; //stringa di caratteri per l'invio UART
15 char str0[40];
16 char str1[9]="Il pixel ";
17 char str2[6]=" vale ";
18 char str3[7]="Picco ";
19 char str4[10]=" @ pixel ";
20 char new[19]="\r\nNew emission Ton=";
21
22 int flag_UART = 0 ;//0=NO UART 1=SI UART
23
24 //PARAMETRI FOTOCELLULA
25 int Ton=2;
26
27 //Interrupt per leggere ADC
28 CY_ISR(adcint)
29 {
30     result[i-1]=ADC_GetResult16(0); //prelevo risultato del campionamento ADC
31     voltage[i-1]=ADC_CountsTo_mVolts(0,result[i-1]); //converto il risultato in mV
32     find_max();
33     if (flag_UART){ //procedura per rendere le stringhe per invio UART della stessa lunghezza
34         if(voltage[i-1]<99){
35             v_buff[(i*4)-3]='0';
36             v_buff[(i-1)*4]='0';
37             sprintf(&v_buff[(i*4)-2],"%d",voltage[i-1]);
38         }
39         else if(voltage[i-1]<999){
40             v_buff[(i-1)*4]='0';
41             sprintf(&v_buff[(i*4)-3],"%d",voltage[i-1]);
42         } else{
43             sprintf(&v_buff[(i-1)*4],"%d",voltage[i-1]);
44         }
45         if(i<11){
46             sprintf(&i_buff[i-1],"%d",i);
47         } else{
48             sprintf(&i_buff[2*i-11],"%d",i);
49         }
50     }
51 }
52
53 int main()
54 {
55     CyGlobalIntEnable; /* Enable global interrupts. */
56     //inizializzo adc, uart e interrupt
57     ADC_Start();
58
59     isr_Enable();
60     isr_StartEx(adcint);
61
62     if(flag_UART) UART_Start();
63
64     for(;;)
65     {
66         v_max=0;
67         pix_max=0;
68
69         if(flag_UART) {
70             memset(str0,0,sizeof(str0));
71             snprintf(str0,sizeof(str0),"%s%d%s",new,Ton,"\r\n");
72             UART_UartPutString(str0);
73         }
74
75         i=1;
76         EMI_EN_Write(1);
77         CyDelayUs(Ton); //impongo la durata d'emissione pari a Ton
78         EMI_EN_Write(0);

```

```

78 |         SI_Write(1);
79 |         CLOCK1();
80 |
81 |         while(i<64){           //ripetere per i 64 pixel
82 |             i++;
83 |             CLOCK_rpt();
84 |         }
85 |         if(flag_UART) {       //procedura d'invio UART
86 |             int j=0;
87 |             memset(str0,0,sizeof(str0));
88 |             for(;j<9;j++){
89 |                 sprintf(str0,sizeof(str0),"%s%c%s%c%c%c%s",str1,i_buff[j],str2,v_buff[4*j],
90 |                     v_buff[4*j+1],v_buff[4*j+2],v_buff[4*j+3],"\r\n");
91 |                 UART_UartPutString(str0);
92 |             }
93 |             for (;j<64;j++){
94 |                 sprintf(str0,sizeof(str0),"%s%c%s%c%c%c%c%s",str1,i_buff[2*j-9],i_buff[2*j-8],
95 |                     str2,v_buff[4*j],v_buff[4*j+1],v_buff[4*j+2],v_buff[4*j+3],"\r\n");
96 |                 UART_UartPutString(str0);
97 |             }
98 |             memset(str0,0,sizeof(str0));
99 |             sprintf(str0,sizeof(str0),"%s%d%s%d%s",str3,v_max,str4,pix_max,"\r\n");
100 |             UART_UartPutString(str0);
101 |         }
102 |         checks();
103 |     }
104 | }
105 |
106 | //generazione primo impulso di clock con discesa di SI
107 | void CLOCK1(){
108 |     CLK_STATE_Write(1);
109 |     SI_Write(0);
110 |     CLK_STATE_Write(0);
111 |     CyDelayUs(1);
112 | }
113 |
114 | //generazione restanti fronti di clock
115 | void CLOCK_rpt(){
116 |     CLK_STATE_Write(1);
117 |     CLK_STATE_Write(0);
118 |     CyDelayUs(1);
119 | }
120 |
121 | //ricerca valore massimo
122 | void find_max(){
123 |     if (voltage[i-1]> v_max){
124 |         v_max=voltage[i-1];
125 |         pix_max=i;
126 |     }
127 | }
128 |
129 | //controllo emissione
130 | void checks(){
131 |     if (v_max>2000 && emi>1){
132 |         emi-=1;
133 |     }else if (v_max<750 && emi==2){
134 |         emi=3;
135 |     } else if (v_max<800 && emi==1){
136 |         emi=2;
137 |     }
138 |     switch(emi){
139 |         case 1: Ton=2;
140 |             break;
141 |         case 2: Ton=4;
142 |             break;
143 |         case 3: Ton=12;
144 |             break;
145 |     }
146 | }

```

Si è reso necessario inserire un flag UART per attivare o disattivare rapidamente la comunicazione seriale.

Nel primo caso sarà utile per una successiva elaborazione dei dati mentre il secondo caso permette di valutare gli effettivi tempi di acquisizione, senza i ritardi introdotti dalla trasmissione UART.

2.2 Risultati ottenuti

La prima analisi è stata eseguita sull'andamento delle forme d'onda dei principali segnali per verificare il corretto pilotaggio dell'array.

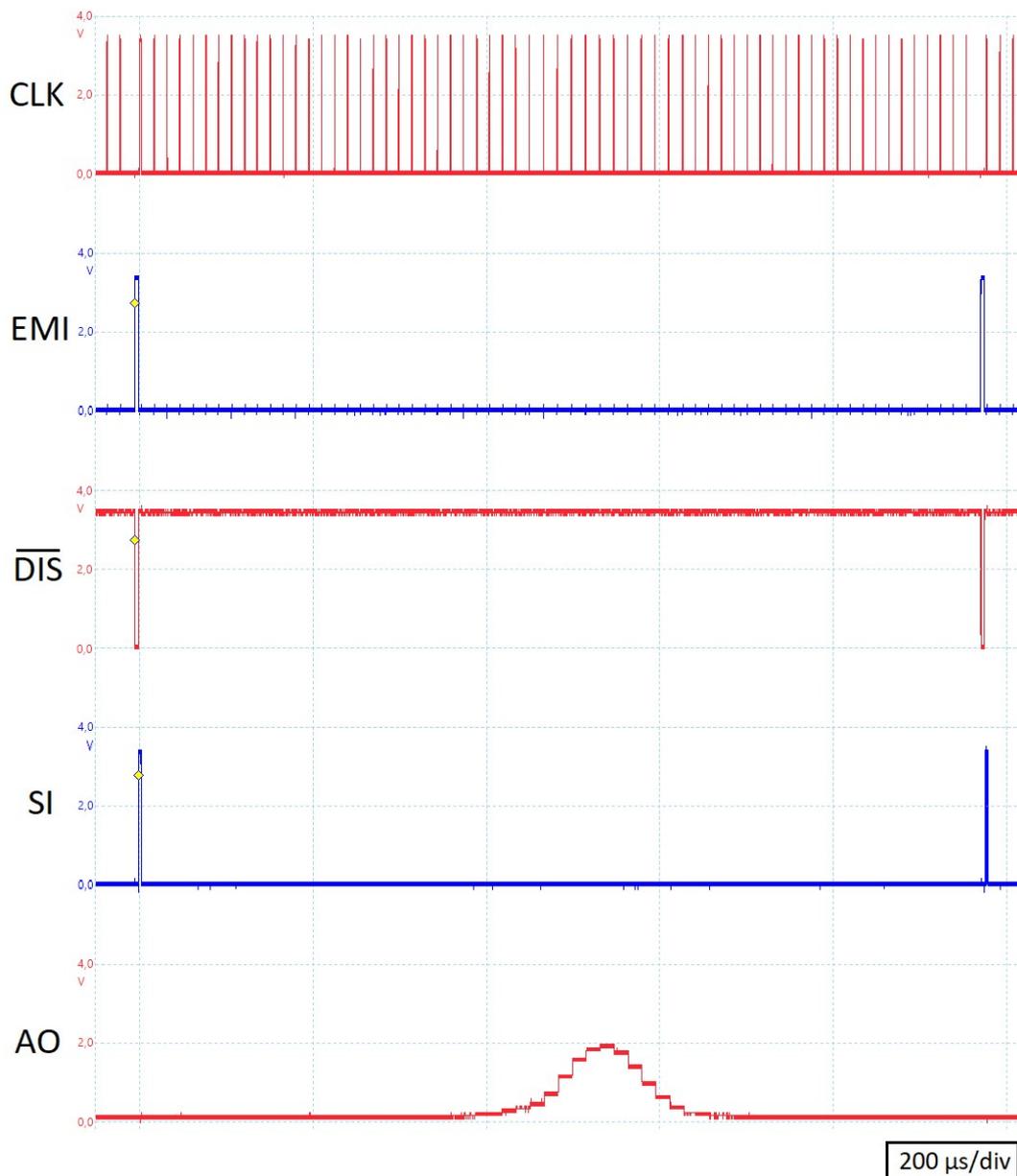


Figura 2.5 Andamento nel tempo dei principali segnali

Si può notare come l'impulso del segnale EMI indichi l'inizio di un nuovo ciclo di lettura, $\overline{\text{DIS}}$ segua correttamente l'andamento invertito di EMI mentre l'impulso SI avvia il ciclo di read out dell'uscita AO.

Per i successivi 64 cicli di clock, si campiona il segnale di uscita AO ed infine, prima di iniziare un nuovo ciclo di lettura, si effettua il controllo sul valore massimo dell'uscita, modificando il Ton di conseguenza.

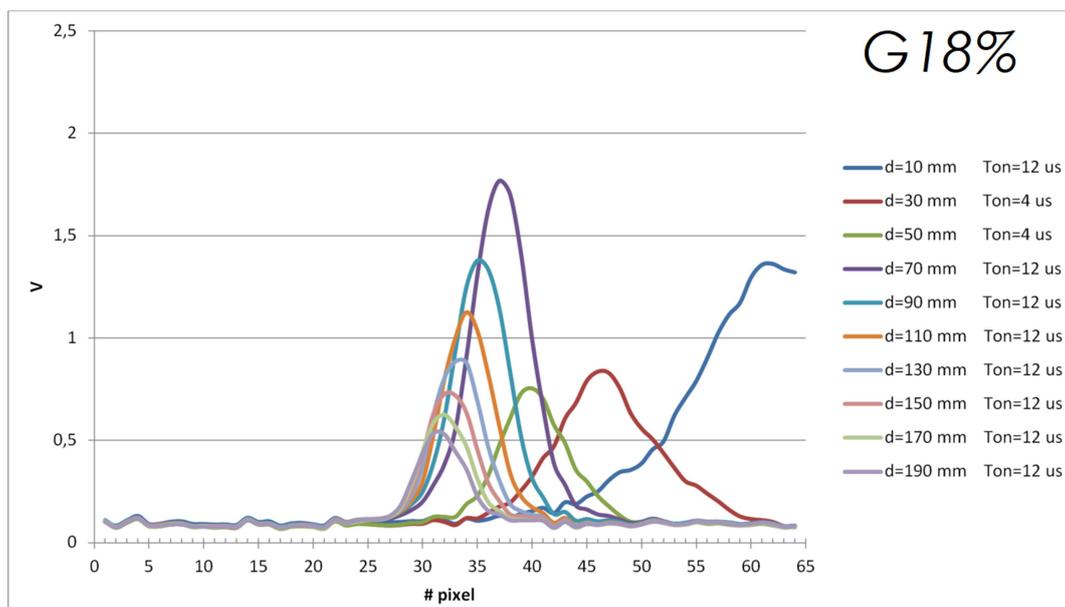
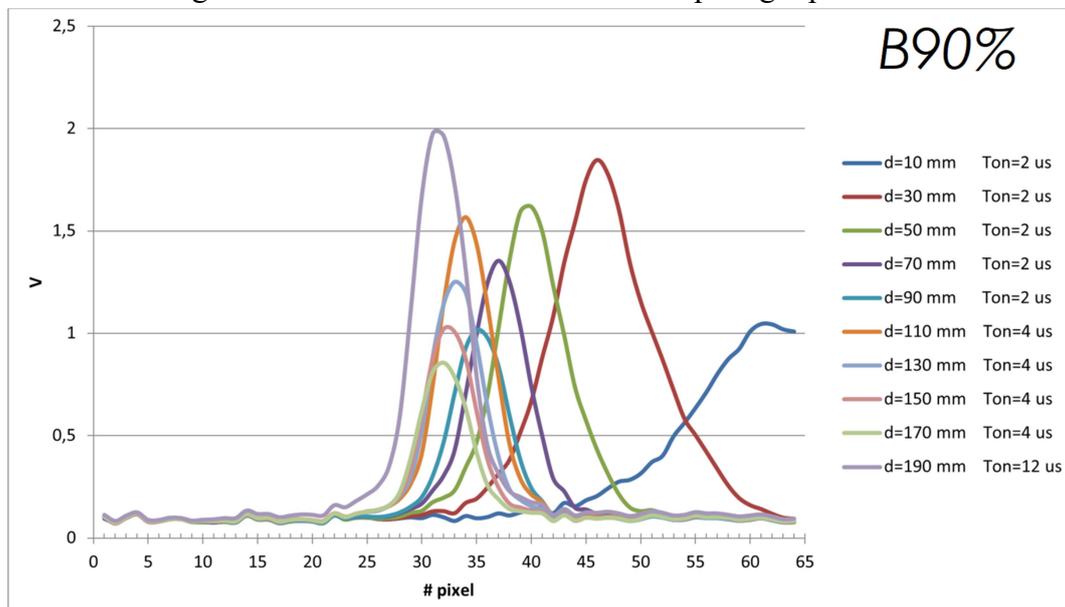
Il clock generato presenta un T_{on} pari a 625 ns ed un t_{off} pari a 14,4 μs che è un valore elevato in quanto include il tempo di campionamento e conversione dell'ADC.

Il dato più interessante è il tempo che intercorre tra un impulso di emissione ed il successivo che indica, quindi, la durata di un ciclo di lettura delle uscite dell'array: $\approx 972 \mu s$.

Per le applicazioni industriali adatte a questo tipo di tecnologia, tale valore è decisamente troppo elevato, indizio dunque che l'acquisizione va senz'altro velocizzata.

Successivamente sono state effettuate misure per ognuno dei 3 target a disposizione nel range di distanza 10 mm \div 190 mm con uno step di 20 mm.

I campioni dell'uscita analogica dell'array sono stati raccolti in un foglio elettronico: si grafica il valore di tensione dell'uscita per ogni pixel.



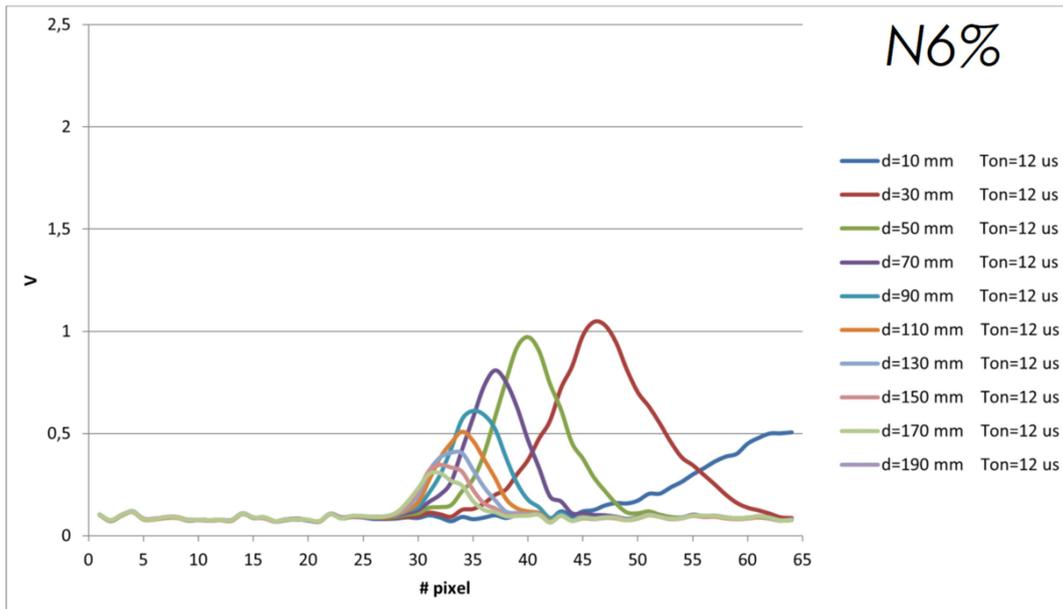


Figure 2.6 a-b-c Grafici dell'uscita per i 3 target

Dalla legenda si nota come ad aumentare della distanza, il periodo di emissione aumenta.

Inoltre si può osservare come i valori massimi assunti dall'uscita sono adeguati alla riflettanza di ogni target.

Interessante più di tutto è però l'andamento di tipo gaussiano che assume l'uscita. Tale comportamento è dovuto al tipo di emissione anch'essa di tipo gaussiana: lo spot presenta un punto centrale ad intensità massima e code via via sempre meno luminose.

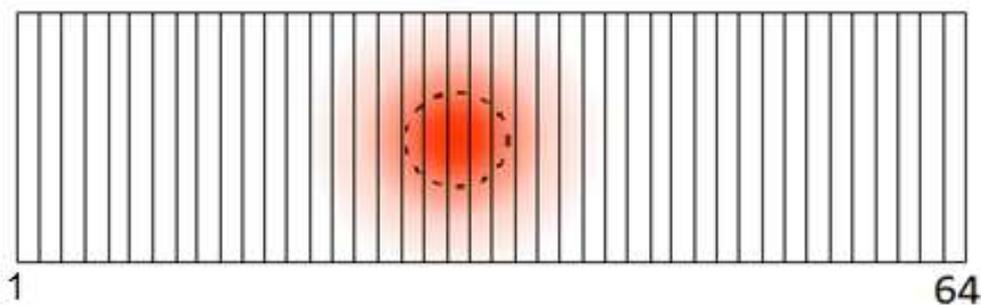


Figura 2.7 Emissione di tipo gaussiana

L'andamento gaussiano non si manifesta per le misure con il target posto ad una distanza di 10 mm: in questa situazione lo spot esce dall'array non permettendo più una corretta acquisizione.

Si può concludere che 10 mm rappresenta la cosiddetta "blind zone" del sensore che si è realizzato.

Molto importante è osservare come si sposta sull'array lo spot, ovvero il picco della gaussiana, in funzione della distanza (viene riportato un grafico unico in quanto il comportamento è identico per i 3 target):

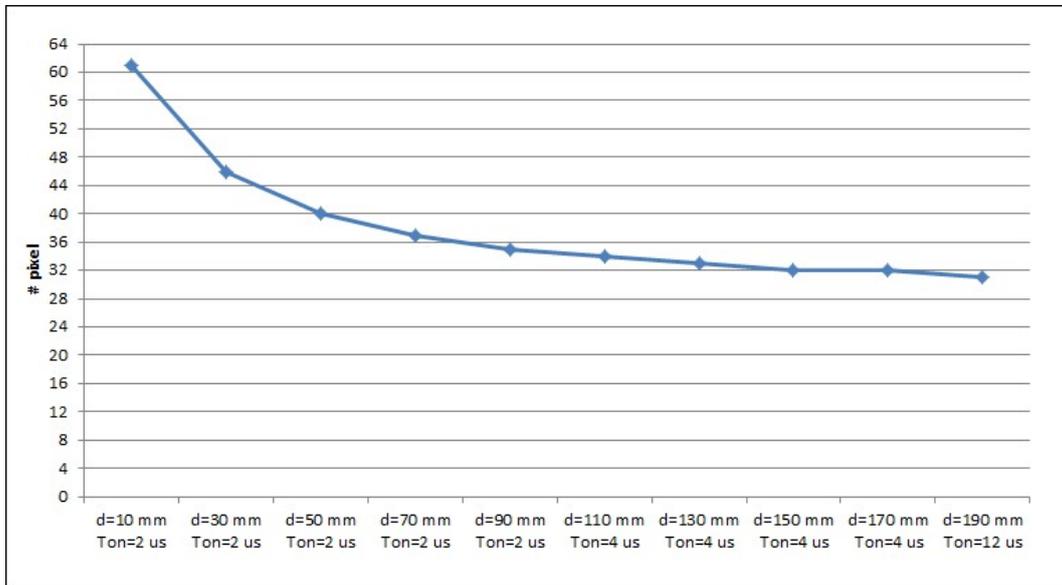


Figura 2.8 Posizione dello spot sull'array in funzione della distanza

Questo grafico fornisce due importanti constatazioni:

1. l'andamento è iperbolico (come ci si aspettava) e lo spot si muove molto da vicino ($d < 50$ mm circa) e poco da lontano;
2. l'iperbole tende ai pixel 31/32 pertanto la prima metà dell'array è come se non fosse utilizzata.

Quest'ultima considerazione la si poteva già notare dalle figure 2.6 a-b-c in quanto i valori della tensione d'uscita per i pixel inferiori al 30° si mantengono dell'ordine di pochi mV.

Il motivo di questo comportamento risiede nel tipo di struttura in cui è stato inserito l'array: è una struttura tubolare che non permette angoli di riflessione del raggio incidente tali da irradiare la prima metà dell'array.

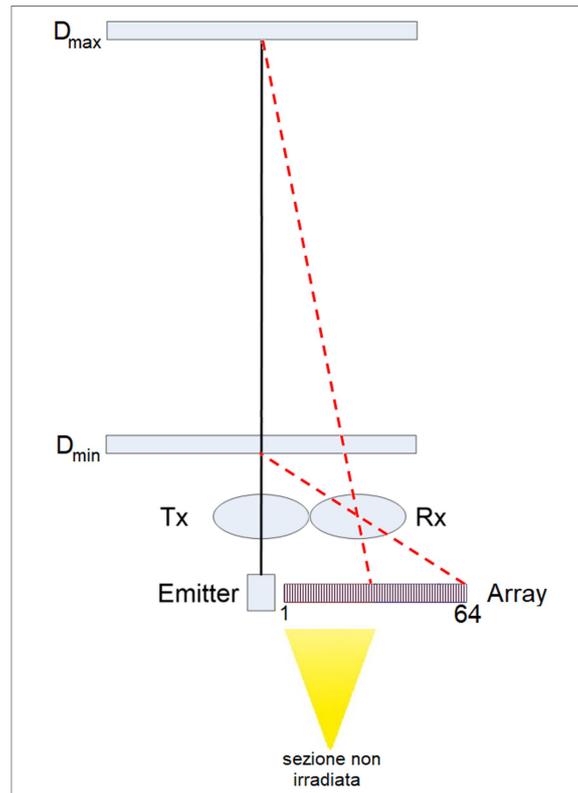


Figura 2.9 Sezione dell'array non colpita da illuminamento

Queste importanti osservazioni, portano alla definizione di un metodo che permette di ottenere un tempo di acquisizione più veloce.

Ripartendo dall'eq.2 di pag.30, si suppone di voler mantenere $\Delta D_k = cost = \Delta D \forall k$ e si ipotizza di poter ottenere tale risultato modulando la lunghezza del pixel p_k .

Si ottiene quindi:

$$p_k = \frac{fi}{\Delta D} \frac{1}{k(k+1)} \quad (\text{Eq.3})$$

Realizzando un array lineare avente pitch costante, si può definire $p_k = m_k p$ dove p è la lunghezza del pixel e m_k è un intero.

Allora si trova che:

$$m_k = \left\lceil \frac{fi}{p \Delta D} \frac{1}{k(k+1)} \right\rceil$$

La risoluzione ΔD è garantita acquisendo i pixel di ordine m_k .

Da evidenziare che comunque la risoluzione a distanza elevata rimane scarsa: per k piccolo (distanza elevata) potrebbe risultare $p_k < p$.

In questo caso la situazione migliore è quella di acquisire tutti i pixel ed interpolare la posizione dello spot.

Diventa dunque necessario fare in modo che lo spot non sia puntiforme ma la reale forma sarà data da un tradeoff tra risoluzione e complessità.

Per ridurre il tempo di acquisizione un'alternativa valida è quella di creare un array ad hoc, ovvero un array non lineare avente pitch variabile.

La regola di dimensionamento della lunghezza del pixel per tale array potrebbe essere quella espressa in eq.3.

Si utilizzano pitch ampi per i pixel che riguardano le distanze vicine (dove è sufficiente una risoluzione inferiore siccome lo spot si muove molto) e, al contrario, occorrono pitch ridotti per i pixel destinati alle distanze lontane dove è necessaria una risoluzione maggiore in quanto lo spot si muove poco.

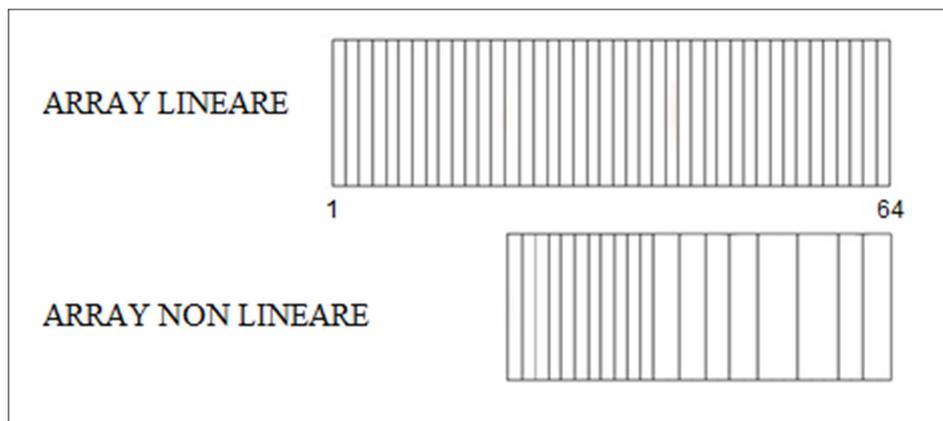


Figura 2.10 Differenza concettuale tra array lineare e non lineare

Presentando un numero inferiore di pixel, è lecito pensare che i tempi di acquisizione per un array non lineare siano inferiori rispetto a quelli per un array lineare.

Non riuscendo ad approfondire gli studi e il progetto di un array non lineare, si è scelto di mantenere l'array lineare di partenza, proseguendo con una scansione parziale.

3. Scansione parziale

Partendo dalle considerazioni effettuate nel paragrafo precedente e avvalendosi di prove empiriche basate su esperimenti pratici, dei 64 pixel totali si è scelto di andarne a campionare soltanto 17.

Di seguito l'elenco dei pixel campionati per la scansione parziale dell'array: 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 40, 42, 44, 46, 52, 58, 64.

Si procede con la programmazione del microcontrollore al fine di valutare tempi e sviluppo dei segnali principali (non verrà valutato l'andamento dell'uscita per ogni pixel campionato: è gaussiano come già precedentemente mostrato per la scansione totale).

3.1 Programmazione microcontrollore

Si riporta innanzitutto il TopDesign.

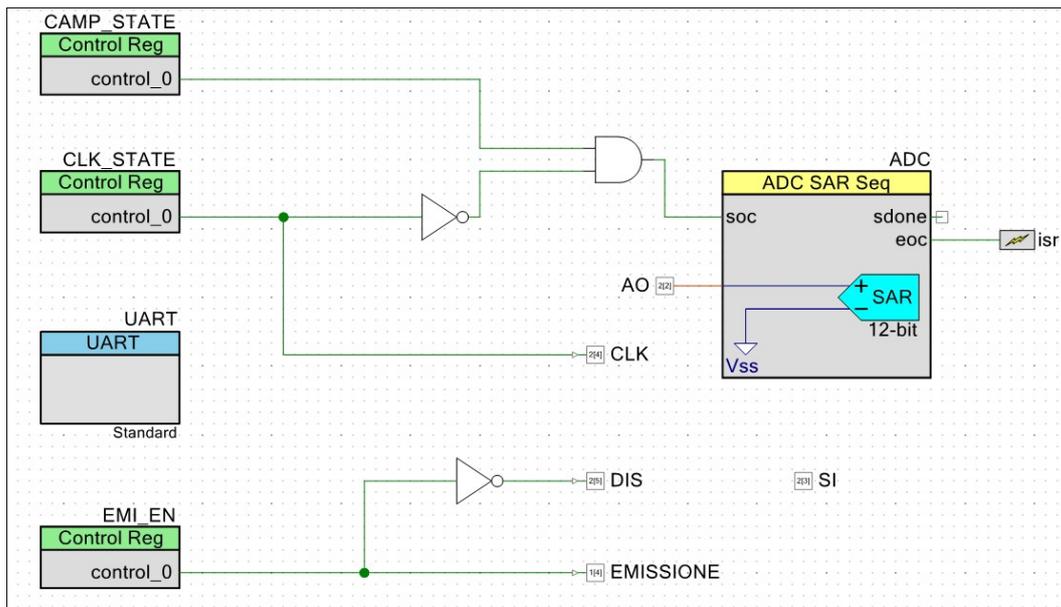


Figura 3.1 TopDesign per la scansione parziale

L'unica modifica rispetto al caso della scansione totale è stata l'aggiunta di un segnale interno CAMP, necessario ad attivare il campionamento soltanto per i pixel desiderati.

Per i restanti 47 pixel non si effettua il campionamento ma si è obbligati comunque ad eseguire la scansione tramite un colpo di clock, rallentando quindi inevitabilmente l'acquisizione.

Di seguito, il codice di programmazione del microcontrollore.

```

1 #include "project.h"
2 #include <stdio.h>
3 #include <math.h>
4
5 void CLOCK1();
6 void CLOCK_rpt();
7 void checks();
8 void find_max();
9
10 int i=0,v_max=0,pix_max=0,emi=1,m=0;
11 int result[64]; //array che contiene il risultato dei campionamenti
12 int voltage[64]; //array che contiene le tensioni dei pixel campionati
13 char v_buff[256]={0}; //stringa di caratteri per invio ad UART
14 char i_buff[120]={0}; //stringa di caratteri per invio ad UART
15 char str0[24]={0};
16 char str1[]="I1 pixel ";
17 char str2[]=" vale ";
18 char str3[]="Picco ";
19 char str4[]=" @ pixel ";
20 char new[]="\r\nNew emission Ton=";
21
22 int flag_UART = 0 ;//0=NO UART 1=Si UART
23
24 //PARAMETRI FOTOCELLULA
25 int Ton=2;
26
27 //Interrupt per leggere ADC
28 CY_ISR(adcint)
29 {
30     result[m-1]=ADC_GetResult16(0); //prelevo risultato del campionamento ADC
31     voltage[m-1]=ADC_CountsTo_mVolts(0,result[m-1]); //converto il risultato in mV
32     find_max();
33     if(flag_UART){ //procedura per rendere le stringhe per invio UART della stessa lunghezza
34         if(voltage[m-1]<99){
35             v_buff[(m*4)-3]='0';
36             v_buff[(m-1)*4]='0';
37             sprintf(&v_buff[(m)*4-2],"%d",voltage[m-1]);
38         }
39         else if(voltage[m-1]<999){
40             v_buff[(m-1)*4]='0';
41             sprintf(&v_buff[(m)*4-3],"%d",voltage[m-1]);
42         }else{
43             sprintf(&v_buff[(m-1)*4],"%d",voltage[m-1]);}
44         if(i<11){
45             i_buff[2*(m-1)]='0';
46             sprintf(&i_buff[2*m-1],"%d",i);
47         } else{
48             sprintf(&i_buff[2*(m-1)],"%d",i);
49         }
50     }
51 }
52
53 int main()
54 {
55     CyGlobalIntEnable; /* Enable global interrupts. */
56     //inizializzo adc, uart e interrupt
57     ADC_Start();
58
59     isr_Enable();
60     isr_StartEx(adcint);
61
62     if(flag_UART) UART_Start();
63
64     for(;;)
65     {
66         m=0;
67         v_max=0;
68         pix_max=0;
69
70         if(flag_UART) {
71             memset(str0,0,sizeof(str0));
72             sprintf(str0,sizeof(str0),"%s%d%s",new,Ton+2,"\r\n");
73             UART_UartPutString(str0);
74         }
75         i=1;
76         CAMP_STATE_Write(0);
77         EMI_EN_Write(1);

```

```

78 | CyDelayUs(Ton); //impongo la durata di emissione pari a Ton
79 | EMI_EN_Write(0);
80 | SI_Write(1);
81 | CLOCK1();
82 |
83 | while(i<64){ //ripetere per i 64 pixel
84 |     i++;
85 |     if(i==27 || i==28 || i==29 || i==30 || i==31 || i==32 || i==33 || i==34 || i==36 ||
86 |        i==38 || i==40 || i==42 || i==44 || i==46 || i==52 || i==58 || i==64){
87 |         m++;
88 |         CAMP_STATE_Write(1);
89 |         CAMP_STATE_Write(0);
90 |     }else CAMP_STATE_Write(0);
91 |     CLOCK_rpt();
92 | }
93 | if(flag_UART) { //procedura d'invio UART
94 |     int j=0;
95 |     memset(str0,0,sizeof(str0));
96 |     for(;j<m;j++){
97 |         snprintf(str0,sizeof(str0),"s%c%c%s%c%c%c%s",str1,i_buff[2*j],i_buff[2*j+1],
98 |            str2,v_buff[4*j],v_buff[4*j+1],v_buff[4*j+2],v_buff[4*j+3],"r\n");
99 |         UART_UartPutString(str0);
100 |     }
101 |     memset(str0,0,sizeof(str0));
102 |     snprintf(str0,sizeof(str0),"s%d%s%d%s",str3,v_max,str4,pix_max,"r\n");
103 |     UART_UartPutString(str0);
104 | }
105 | checks();
106 | }
107 | }
108 |
109 | //generazione primo impulso di clock con discesa di SI
110 | void CLOCK1()
111 | {
112 |     CLK_STATE_Write(1);
113 |     SI_Write(0);
114 |     CLK_STATE_Write(0);
115 | }
116 |
117 | //generazione restanti fronti di clock
118 | void CLOCK_rpt()
119 | {
120 |     CLK_STATE_Write(1);
121 |     CLK_STATE_Write(0);
122 | }
123 |
124 | //ricerca valore massimo e rispettivo pixel
125 | void find_max()
126 | {
127 |     if (voltage[m-1]> v_max){
128 |         v_max=voltage[m-1];
129 |         pix_max=i;
130 |     }
131 | }
132 |
133 | //controllo saturazione e controllo valore minimo
134 | void checks()
135 | {
136 |     if (v_max>2000 && emi>1){
137 |         emi-=1;
138 |     }else if (v_max<750 && emi==2){
139 |         emi=3;
140 |     } else if (v_max<800 && emi==1){
141 |         emi=2;
142 |     }
143 |     switch(emi){
144 |         case 1: Ton=2;
145 |             break;
146 |         case 2: Ton=4;
147 |             break;
148 |         case 3: Ton=12;
149 |             break;
150 |     }
151 | }

```

Rispetto al codice della scansione totale, è stata aggiunta la condizione che pone al livello logico 1 il segnale CAMP quando l'iterazione corrente riguarda uno di quei pixel che si vuole campionare.

3.2 Risultati ottenuti

Si è analizzato l'andamento nel tempo dei principali segnali.

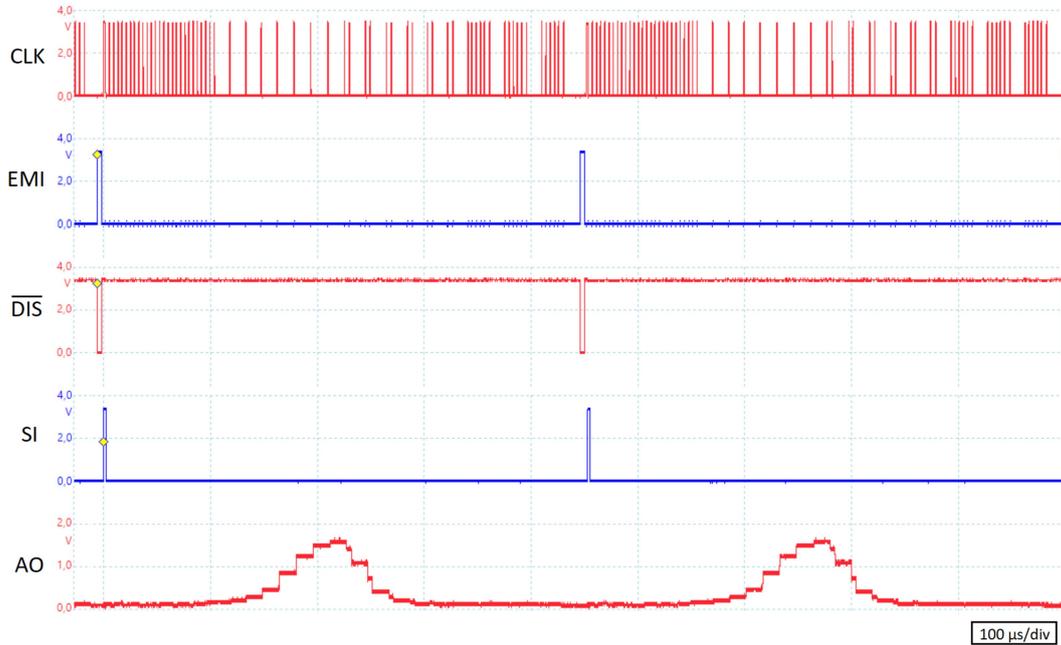


Figura 3.2 Andamento nel tempo dei principali segnali

L'impulso del segnale EMI coincide con l'inizio di un nuovo ciclo di lettura. Da notare l'andamento irregolare del clock: esso presenta in principio 26 cicli di clock ravvicinati che rappresentano la semplice scansione dei primi 26 pixel. Successivamente si può osservare la sequenza dei pixel campionati, in quanto gli impulsi di clock risultano più distanziati a causa del delay introdotto dal campionamento.

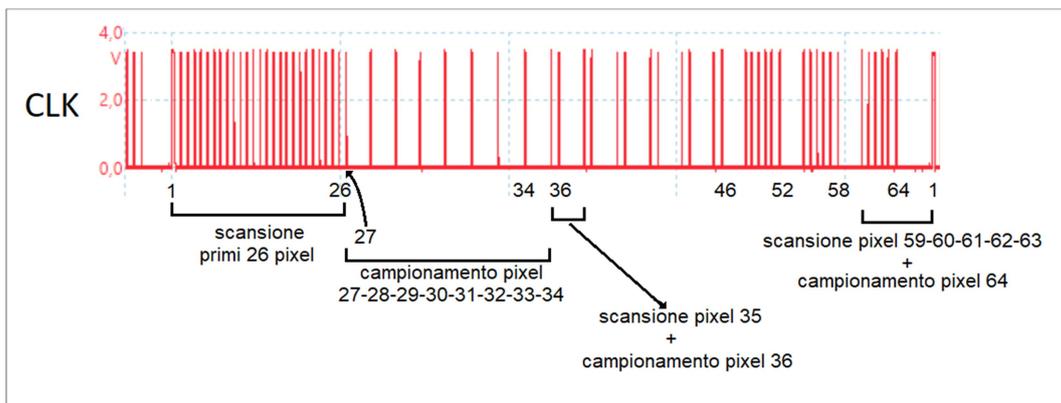


Figura 3.3 Zoom del segnale di clock per evidenziarne la particolarità

Gli altri segnali si comportano come in precedenza, con $\overline{\text{DIS}}$ che segue l'andamento invertito del segnale di emissione e SI che dà il via al ciclo di read-out dell'uscita che, come si vede, ha l'andamento gaussiano previsto.

Si è ottenuto un dimezzamento della durata del ciclo di lettura (misurata tra un impulso del segnale EMI ed il successivo) che risulta pari a 449 μ s rispetto ai 972 μ s della scansione totale.

È un risultato che, in realtà, potrebbe essere migliorato ulteriormente agendo soprattutto sul clock.

Il clock generato presenta un T_{on} pari a 624 ns ed un t_{off} pari a 3.3 μ s, con una frequenza quindi di circa 260 kHz: decisamente inferiore (≈ 20 volte) rispetto la frequenza massima fornita dal datasheet pari a 5MHz.

Il motivo di questa lentezza del segnale è da imputare al setup di misura: la generazione software tramite PSoC non permette di avere un clock più veloce.

In particolare, il problema risiede nell'eccessivo tempo t_{off} che compromette la semplice scansione dei 47 pixel da non campionare che, idealmente, dovrebbe essere la più veloce possibile.

4. Distanza

Una volta ottenuta un'acquisizione più veloce possibile compatibilmente con il setup di misura utilizzato, si vuole ricavare informazioni sulla distanza del target. Il modo per risalire ad una stima della distanza con le informazioni a disposizione risiede proprio nell'iperbole di figura 2.8: se si è in grado di conoscere per ogni distanza su quale pixel cade lo spot ovvero il picco della gaussiana (si trova quindi l'ordinata), è sufficiente invertire la funzione per ricavare l'ascissa, cioè la distanza.

Bisogna però fare attenzione: eseguendo una scansione parziale di soli 17 pixel, non si ha la certezza che, per ogni misura, il picco sia sempre campionato.

È necessario allora un algoritmo che permetta con sicurezza di sapere la posizione del picco sull'array, nonostante si stia eseguendo una scansione parziale.

4.1 Approssimazione curva gaussiana

Avendo a che fare con una funzione complessa come quella gaussiana e dovendo cercare di ricavarla attraverso una serie ridotta di punti, si è scelto di approssimarla tramite una funzione matematica più semplice come la parabola. Come si può vedere dalla figura, tale approssimazione è opportuna nell'intorno del picco (che è la zona di nostro interesse) mentre perde di significato se si considerano le code della funzione.

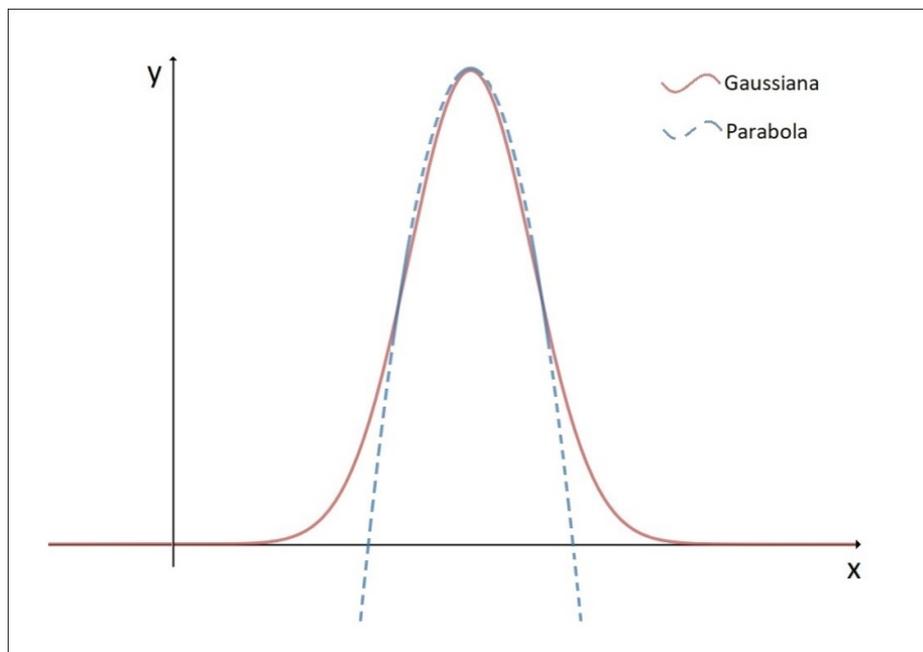


Figura 4.1 Andamento di una gaussiana e parabola generici

È noto dall'algebra che per ricavare l'equazione di una parabola con asse verticale in maniera univoca sono sufficienti 3 punti.

Tra i 17 pixel campionati, per il calcolo della parabola, si considerano soltanto quello a valore più alto e i due adiacenti.

Si è in possesso delle 3 coordinate di tali punti dove la x indica il numero del pixel mentre la y indica il valore in volt: con un sistema si può ricavare l'equazione della parabola.

Una volta trovata, con le formule canoniche della parabola, è semplice ricavare le coordinate del vertice (in particolare si è interessati alla sola coordinata x).

Di seguito, si mostra la serie di calcoli che permettono di trovare l'ascissa del vertice V_x (e quindi la sua posizione sull'array) partendo dalle 3 equazioni della parabola per ogni punto generico indicato con il pedice 1, 2, 3.

$$\begin{cases} y_1 = a x_1^2 + b x_1 + c \\ y_2 = a x_2^2 + b x_2 + c \\ y_3 = a x_3^2 + b x_3 + c \end{cases} \quad \begin{cases} c = y_1 - a x_1^2 - b x_1 \\ y_2 = a x_2^2 + b x_2 + y_1 - a x_1^2 - b x_1 \\ y_3 = a x_3^2 + b x_3 + y_1 - a x_1^2 - b x_1 \end{cases}$$

$$\begin{cases} b(x_2 - x_1) = y_2 - a x_2^2 - y_1 + a x_1^2 \\ y_3 = a x_3^2 + b x_3 + y_1 - a x_1^2 - b x_1 \end{cases} \quad \begin{cases} b = \frac{y_2 - a x_2^2 - y_1 + a x_1^2}{x_2 - x_1} \\ y_3 = a x_3^2 + b x_3 + y_1 - a x_1^2 - b x_1 \end{cases}$$

$$y_3 = a x_3^2 + \frac{x_3 y_2 - a x_2^2 x_3 - x_3 y_1 + a x_3 x_1^2}{x_2 - x_1} + y_1 - a x_1^2 - \frac{x_1 y_2 - a x_1 x_2^2 - x_1 y_1 + a x_1^3}{x_2 - x_1}$$

$$y_3 x_2 - y_3 x_1 = a x_2 x_3^2 - a x_1 x_3^2 + x_3 y_2 - a x_2^2 x_3 - x_3 y_1 + a x_1^2 x_3 + y_1 x_2 - x_1 y_1 - a x_1^2 x_2 + a x_1^3 - x_1 y_2 + a x_1 x_2^2 + x_1 y_1 - a x_1^3$$

$$a(x_1 - x_2)(x_1 - x_3)(x_2 - x_3) = x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1)$$



$$a = \frac{x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}$$

Si riscrive b:

$$b = \frac{y_2 - a x_2^2 - y_1 + a x_1^2}{x_2 - x_1} = \frac{y_2 + a(x_1^2 - x_2^2) - y_1}{x_2 - x_1}$$

Si sostituisce a:

$$b = \frac{y_2 + \frac{x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}(x_1^2 - x_2^2) - y_1}{x_2 - x_1} =$$

$$= \frac{x_1^2(y_2 - y_3)(x_2 - x_1) + x_2^2(y_3 - y_1)(x_2 - x_1) + x_3^2(y_1 - y_2)(x_2 - x_1)}{(x_2 - x_1)(x_1 - x_2)(x_2 - x_3)(x_1 - x_3)}$$



$$b = \frac{x_1^2(y_2 - y_3) + x_2^2(y_3 - y_1) + x_3^2(y_1 - y_2)}{(x_1 - x_2)(x_2 - x_3)(x_1 - x_3)}$$

Si riscrive c:

$$c = y_1 - a x_1^2 - b x_1$$

Sostituendo a e b si ottiene:

$$c = \frac{x_2(x_1^2 y_3 - x_3^2 y_1) + x_2^2(x_3 y_1 - x_1 y_3) + x_1 x_3 y_2(x_3 - x_1)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}$$

Si ricava ora la coordinata x del vertice:

$$V_X = -\frac{b}{2a} = -\frac{x_1^2(y_2 - y_3) + x_2^2(y_3 - y_1) + x_3^2(y_1 - y_2)}{2[x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1)]}$$



$$V_X = \frac{x_1^2(y_3 - y_2) + x_2^2(y_1 - y_3) + x_3^2(y_2 - y_1)}{2[x_1(y_3 - y_2) + x_2(y_1 - y_3) + x_3(y_2 - y_1)]}$$

(Eq.4)

Successivamente si è valutata l'accuratezza della misura eseguendo 198 letture consecutive e trovando media, varianza e deviazione standard sul valore V_X della posizione dello spot.

Da precisare che i calcoli di media e varianza sono stati eseguiti a livello software ed in virgola mobile comportando un maggior impatto computazionale nell'elaborazione dei dati.

Si è scelto, quindi, di calcolare la deviazione standard a parte, su foglio elettronico, siccome contenente un'operazione come la radice quadrata che avrebbe peggiorato ulteriormente l'onere del calcolo.

$$\mu = \frac{1}{n} \sum_{i=1}^n X_i \quad \sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \mu)^2 \quad (\text{Eq.5})$$

4.2 Programmazione microcontrollore

Si riporta il TopDesign.

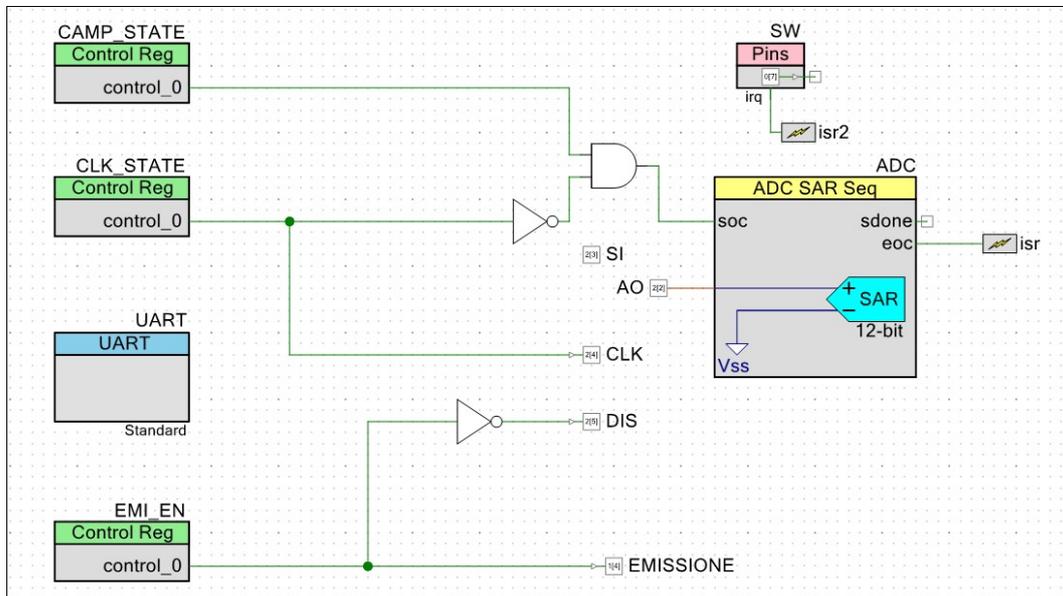


Figura 4.2 TopDesign per la programmazione della scheda PSoc

L'unica modifica rispetto al caso della scansione parziale riguarda l'aggiunta di un interrupt **isr2** col quale si dà inizio alle 198 letture consecutive tramite un pulsante.

Di seguito, il codice di programmazione del microcontrollore.

```

1  #include "project.h"
2  #include <stdio.h>
3  #include <math.h>
4
5  void CLOCK1();
6  void CLOCK_rpt();
7  void checks();
8  void cerca_max();
9
10 float vertice[198]={0},somma=0,media=0,varianza=0,sommal=0,distanza=0;
11 int i=0,emi=1,m=0,xp1=0,yp1=0,xp2=0,yp2=0,xp3=0,yp3=0,nsample=0;
12 float vertix=0.00;
13 int result[64]; //array che contiene il risultato dei campionamenti
14 int voltage[64]; //array che contiene le tensioni dei pixel campionati
15 int indici[64]; //array che contiene gli indici
16 char v_buff[256]={0}; //stringa di caratteri per invio ad UART
17 char i_buff[120]={0}; //stringa di caratteri per invio ad UART
18 char str0[24]={0};
19
20 int flag_UART = 0 ;//0=NO UART 1=SÌ UART
21
22 //PARAMETRI FOTOCELLULA
23 int Ton=2;
24
25 //interrupt per l'acquisizione di 198 misure consecutive
26 CY_ISR(swint){
27     nsample=198;
28
29     for(int n=0;n<198;n++)
30     {
31         m=0;
32         yp2=0;
33         xp2=0;
34         xp1=0;
35         yp1=0;
36         xp3=0;
37         yp3=0;
38         somma=0;
39         media=0;
40         varianza=0;
41         sommal=0;
42
43         i=1;
44         CAMP_STATE_Write(0);
45         EMI_EN_Write(1);
46         CyDelayUs(Ton); //impongo la durata d'emissione pari a Ton
47         EMI_EN_Write(0);
48         SI_Write(1);
49         CLOCK1();
50
51         while(i<64){ //ripetere per i 64 pixel
52             i++;
53             if(i==27 || i==28 || i==29 || i==30 || i==31 || i==32 || i==33 || i==34 || i==36
54                 || i==38 || i==40 || i==42 || i==44 || i==46 || i==52 || i==58 || i==64){
55                 m++;
56                 CAMP_STATE_Write(1);
57                 CAMP_STATE_Write(0);
58             }else CAMP_STATE_Write(0);
59             CLOCK_rpt();
60         }
61         cerca_max();
62         vertice[n]=(float) ((xp1*xp1) * (yp3-yp2) + (xp3*xp3) * (yp2-yp1) + (xp2*xp2) * (yp1-yp3)) /
63             (2 * (xp1 * (yp3-yp2) + xp2 * (yp1-yp3) + xp3 * (yp2-yp1))); //calcolo del vertice
64
65         checks();
66     }
67
68     for (int z=0; z<nsample;z++){
69         somma+=vertice[z];
70     }
71     media=(float) ((float)somma/198); //calcolo della media
72
73     memset(str0,0,sizeof(str0));
74     sprintf(str0,"\r\navg = %.6f \r\n", (float)media);
75

```

```

78     UART_Start();
79     UART_UartPutString(str0);
80
81     for(int z=0;z<nsample;z++){
82         sommal+=(float)((float)vertice[z]-(float)media)*((float)vertice[z]-(float)media));
83     }
84     varianza=(float)sommal/(nsample-1); //calcolo della varianza
85     memset(str0,0,sizeof(str0));
86     sprintf(str0,"var = %e \r\n", (float)varianza);
87     UART_Start();
88     UART_UartPutString(str0);
89
90     SW_ClearInterrupt();
91 }
92
93 //Interrupt per leggere ADC
94 CY_ISR(adcint)
95 {
96     result[m-1]=ADC_GetResult16(0); //prelevo risultato del campionamento ADC
97     voltage[m-1]=ADC_CountsTo_mVolts(0,result[m-1]); //converto il risultato in mV
98     indici[m-1]=i;
99     if(flag_UART){ //procedura per rendere le stringhe per invio UART della stessa lunghezza
100         if(voltage[m-1]<99){
101             v_buff[(m*4)-3]='0';
102             v_buff[(m-1)*4]='0';
103             sprintf(&v_buff[((m)*4)-2],"%d",voltage[m-1]);
104         }
105         else if(voltage[m-1]<999){
106             v_buff[(m-1)*4]='0';
107             sprintf(&v_buff[((m)*4)-3],"%d",voltage[m-1]);
108         } else {
109             sprintf(&v_buff[(m-1)*4],"%d",voltage[m-1]);
110         }
111         if(i<11){
112             i_buff[2*(m-1)]='0';
113             sprintf(&i_buff[2*m-1],"%d",i);
114         } else {
115             sprintf(&i_buff[2*(m-1)],"%d",i);
116         }
117     }
118 }
119
120 int main()
121 {
122     CyGlobalIntEnable; /* Enable global interrupts. */
123     //inizializzo adc, uart e interrupt
124     ADC_Start();
125
126     isr_Enable();
127     isr_StartEx(adcint);
128     isr2_StartEx(swint);
129
130     if(flag_UART) UART_Start();
131
132     for(;;)
133     {
134         m=0;
135         yp2=0;
136         xp2=0;
137         xpl=0;
138         ypl=0;
139         xp3=0;
140         yp3=0;
141         vertix=0;
142
143         i=1;
144         CAMP_STATE_Write(0);
145         EMI_EN_Write(1);
146         CyDelayUs(Ton); //impongo la durata d'emissione pari a Ton
147         EMI_EN_Write(0);
148         SI_Write(1);
149         CLOCK1();
150
151         while(i<64){ //ripetere per i 64 pixel
152             i++;
153             if(i==27 || i==28 || i==29 || i==30 || i==31 || i==32 || i==33 || i==34 || i==36
154                 || i==38 || i==40 || i==42 || i==44 || i==46 || i==52 || i==58 || i==64){

```

```

155         m++;
156         CAMP_STATE_Write(1);
157         CAMP_STATE_Write(0);
158     }else CAMP_STATE_Write(0);
159     CLOCK_rpt();
160 }
161 cerca_max();
162 vertix=(float) ((xp1*xp1) * (yp3-yp2)+(xp3*xp3) * (yp2-yp1)+(xp2*xp2) * (yp1-yp3))/
163 (2*(xp1*(yp3-yp2)+xp2*(yp1-yp3)+xp3*(yp2-yp1)));
164 checks();
165 }
166 }
167
168 //generazione primo impulso di clock con discesa di SI
169 void CLOCK1()
170 {
171     CLK_STATE_Write(1);
172     SI_Write(0);
173     CLK_STATE_Write(0);
174 }
175
176 //generazione restanti fronti di clock
177 void CLOCK_rpt()
178 {
179     CLK_STATE_Write(1);
180     CLK_STATE_Write(0);
181 }
182
183 //ricerca del massimo e dei 3 punti per definire la parabola
184 void cerca_max(){
185     for(int k=0;k<m;k++){
186         if(voltage[k]>yp2){
187             yp2=voltage[k];
188             xp2=indici[k];
189             if (xp2==indici[0]){
190                 xp1=indici[k+2];
191                 yp1=voltage[k+2];
192             }else{
193                 xp1=indici[k-1];
194                 yp1=voltage[k-1];
195             }
196             if (xp2==indici[m-1]){
197                 xp3=indici[k-2];
198                 yp3=voltage[k-2];
199             }else{
200                 xp3=indici[k+1];
201                 yp3=voltage[k+1];
202             }
203         }
204     }
205 }
206
207 //controllo saturazione e controllo valore minimo
208 void checks()
209 {
210     if (yp2>2000 && emi>1){
211         emi-=1;
212     }else if (yp2<750 && emi==2){
213         emi=3;
214     } else if (yp2<800 && emi==1){
215         emi=2;
216     }
217     switch(emi){
218         case 1: Ton=2;
219                 break;
220         case 2: Ton=4;
221                 break;
222         case 3: Ton=12;
223                 break;
224     }
225 }

```

Le righe di codice aggiunte rispetto la scansione parziale riguardano appunto l'interrupt isr2: esegue un loop di 198 iterazioni all'interno del quale ogni volta vengono calcolati il vertice (tramite eq.4), media e varianza (tramite eq.5).

4.3 Risultati ottenuti

Si riportano in tabella i valori risultanti per ognuno dei 3 target.

B90%										
	d=10 mm	Ton=4 us	d=30 mm	Ton=2 us	d=50 mm	Ton=2 us	d=70 mm	Ton=2 us	d=90 mm	Ton=2 us
media	//		47,625034		40,876526		38,117085		36,274937	
varianza	//		2,976232E-05		8,944577E-05		1,596910E-05		1,397031E-05	
dev st	//		5,455485E-03		9,457577E-03		3,996136E-03		3,737688E-03	
G18%										
	d=10 mm	Ton=12 us	d=30 mm	Ton=4 us	d=50 mm	Ton=12 us	d=70 mm	Ton=12 us	d=90 mm	Ton=12 us
media	//		47,851208		40,971390		38,176399		36,364361	
varianza	//		6,801051E-05		5,639181E-05		1,184654E-05		2,041923E-05	
dev st	//		8,246848E-03		7,509448E-03		3,441880E-03		4,518764E-03	
	d=110 mm	Ton=4 us	d=130 mm	Ton=4 us	d=150 mm	Ton=4 us	d=170 mm	Ton=4 us	d=190 mm	Ton=4 us
media	35,038441		34,174278		33,471996		32,783192		32,428486	
varianza	6,458451E-05		4,773813E-05		3,615806E-04		1,415217E-04		2,154887E-04	
dev st	8,036449E-03		6,909279E-03		1,901527E-02		1,189629E-02		1,467953E-02	
	d=110 mm	Ton=12 us	d=130 mm	Ton=12 us	d=150 mm	Ton=12 us	d=170 mm	Ton=12 us	d=190 mm	Ton=12 us
media	35,141975		34,132946		33,504044		32,922634		32,498718	
varianza	6,148179E-05		9,463813E-05		5,379972E-04		1,147942E-04		4,025395E-04	
dev st	7,841032E-03		9,728213E-03		2,319477E-02		1,071421E-02		2,006339E-02	
N6%										
	d=10 mm	Ton=12 us	d=30 mm	Ton=12 us	d=50 mm	Ton=12 us	d=70 mm	Ton=12 us	d=90 mm	Ton=12 us
media	//		47,870838		40,928226		38,152485		36,404903	
varianza	//		3,434586E-05		1,341380E-04		2,534158E-05		8,257430E-05	
dev st	//		5,860534E-03		1,158180E-02		5,034042E-03		9,087040E-03	
	d=110 mm	Ton=12 us	d=130 mm	Ton=12 us	d=150 mm	Ton=12 us	d=170 mm	Ton=12 us	d=190 mm	Ton=12 us
media	35,139462		34,045753		33,254852		32,785690		32,446247	
varianza	2,951294E-04		5,621256E-04		4,021380E-04		4,630015E-04		6,714643E-04	
dev st	1,717933E-02		2,370919E-02		2,005338E-02		2,151747E-02		2,591263E-02	

Figura 4.3 Tabella dei valori di media, varianza e deviazione standard per ogni target

Per la distanza di 10 mm il valore della media non è riportato in quanto risultante maggiore di 64: ciò conferma che per quella distanza lo spot cade all'esterno dell'array.

Si grafica ora, per ogni target, come varia la posizione dello spot a seconda della distanza: ci si aspetta di ottenere un'iperbole identica a quella di figura 2.8.

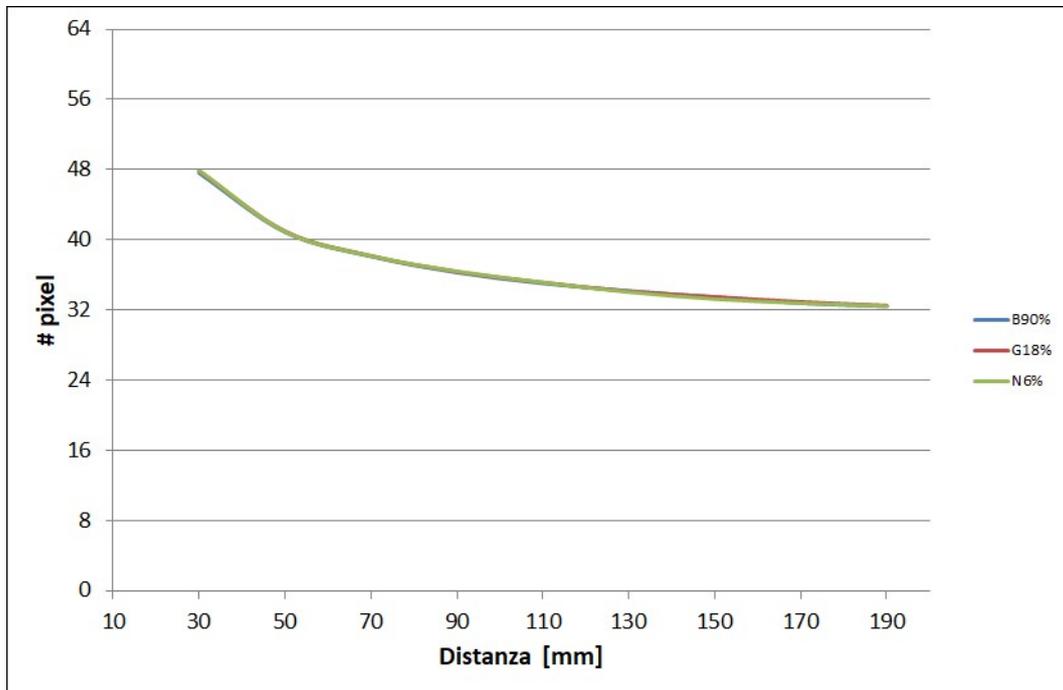


Figura 4.4 Relazione iperbolica tra distanza e posizione dello spot

Le 3 curve sono coincidenti: d'ora in avanti se ne considererà una unica soltanto.

4.4 Fitting iperbole

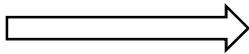
Come già detto, una volta ottenuta l'informazione sulla posizione dello spot, per trasformarla in indicazioni sulla distanza del target bisogna invertire la funzione iperbolica.

Per fare ciò, si è reso necessario trovare una funzione matematica che abbia un andamento il più fedele possibile all'iperbole:

$$y = \frac{K_1}{x+K_2} + K_3 \quad (\text{Eq.6})$$

I valori di K_1 , K_2 e K_3 sono stati trovati tramite un sistema a 3 equazioni, imponendo a x tre valori di distanza (minima, media e massima) e sostituendo a y il valore del pixel su cui cade il picco, ricavato dai valori della tabella precedente per il target bianco90%:

x	y
30	47,6
110	35
190	32,4



$$\begin{cases} 47,6 = \frac{K_1}{30+K_2} + K_3 \\ 35 = \frac{K_1}{110+K_2} + K_3 \\ 32,4 = \frac{K_1}{190+K_2} + K_3 \end{cases}$$

Risolvendo il sistema, si ottiene:

$$\begin{cases} K_1 = 796,723 \\ K_2 = 11,6 \\ K_3 = 28,448 \end{cases}$$

Dal confronto grafico tra l'iperbole originaria e quella ricavata col procedimento appena descritto si nota come i due andamenti siano pressoché coincidenti.

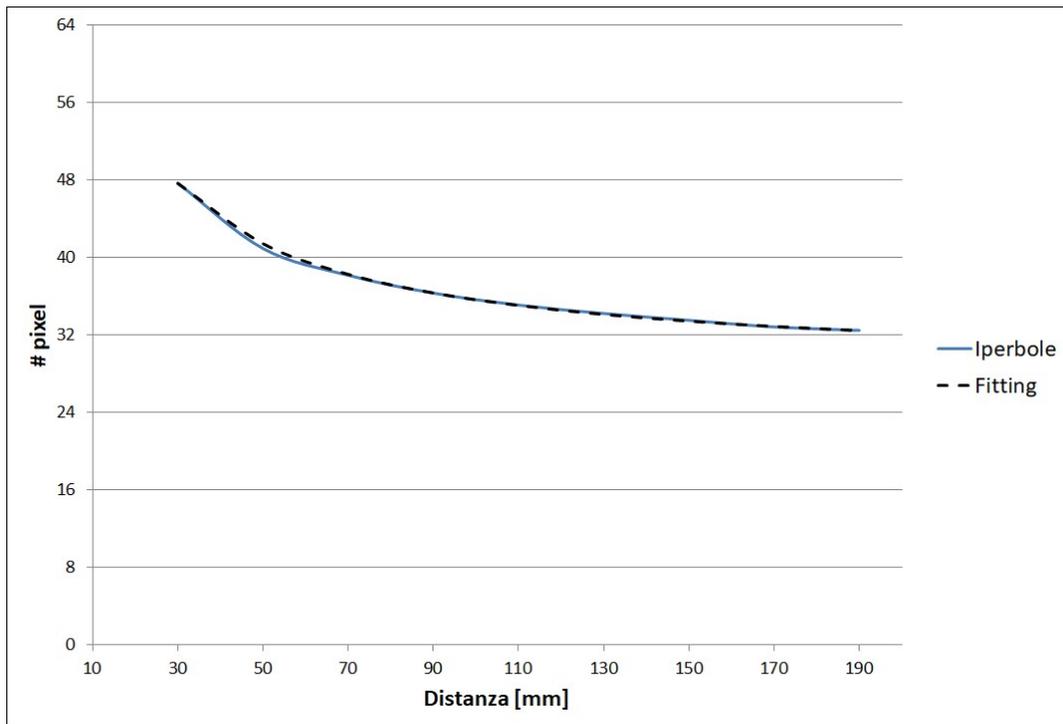


Figura 4.5 Fitting dell'iperbole

Dall'eq.6 si ricava che la distanza (ovvero x) è pari a:

$$x = \frac{K_1 + K_2 K_3 - y K_2}{y - K_3} \quad (\text{Eq.7})$$

4.5 Isteresi

Si vuole ricavare quanto l'incertezza sul valore del numero del pixel di picco ($\Delta\#pix$) si ripercuote in incertezza sul valore della distanza (Δd).

Essendoci una relazione iperbolica, è già facilmente intuibile come per distanze vicine < 90 mm (dove l'iperbole ha una buona pendenza), una piccola incertezza sul valore del pixel corrisponda ad un'altrettanta ridotta incertezza sul valore della distanza. Discorso opposto invece per le distanze lontane > 90 mm dove, siccome l'iperbole ha un andamento molto più piatto, anche un piccolo $\Delta\#pix$ impatta maggiormente su Δd . Per gestire la variabilità dei dati statistici acquisiti, si utilizza il metodo "sei sigma" cioè si ammette un'escursione di tolleranza intorno al valore medio pari appunto a 6σ .

Si calcola :

$$\left. \begin{aligned} \#pix_{max} &= \mu + 3\sigma \\ \#pix_{min} &= \mu - 3\sigma \end{aligned} \right\} \Delta pix = \#pix_{max} - \#pix_{min}$$

Per trasformare questa incertezza in un errore di misura di distanza in mm si utilizza eq.7:

$$\Delta d = \frac{K_1 + K_2 K_3 - \Delta pix K_2}{\Delta pix - K_3}$$

Per ricavare il valore di isteresi in percentuale, si divide il valore dell'incertezza per la distanza di misura:

$$isteresi\% = \frac{\Delta d}{d}$$

In tabella si mostrano i valori ottenuti al variare della distanza ottenuti per i 3 target.

B90%	30 mm	50 mm	70 mm	90 mm	110 mm	130 mm	150 mm	170 mm	190 mm
#pix_min	47,609	40,848	38,105	36,264	35,014	34,154	33,415	32,748	32,384
#pix_max	47,641	40,905	38,129	36,286	35,063	34,195	33,529	32,819	32,473
$\Delta\#pix$	0,033	0,057	0,024	0,022	0,048	0,041	0,114	0,071	0,088
Δd (mm)	0,071	0,293	0,204	0,292	0,885	1,007	3,602	3,026	4,429
isteresi %	0,236%	0,585%	0,292%	0,324%	0,804%	0,775%	2,401%	1,780%	2,331%
G18%	30 mm	50 mm	70 mm	90 mm	110 mm	130 mm	150 mm	170 mm	190 mm
#pix_min	47,826	40,949	38,166	36,351	35,118	34,104	33,434	32,890	32,439
#pix_max	47,876	40,994	38,187	36,378	35,165	34,162	33,574	32,955	32,559
$\Delta\#pix$	0,049	0,045	0,021	0,027	0,047	0,058	0,139	0,064	0,120
Δd (mm)	0,105	0,229	0,174	0,345	0,837	1,439	4,338	2,558	5,846
isteresi %	0,349%	0,458%	0,248%	0,383%	0,760%	1,107%	2,892%	1,505%	3,077%
N6%	30 mm	50 mm	70 mm	90 mm	110 mm	130 mm	150 mm	170 mm	190 mm
#pix_min	47,853	40,893	38,137	36,378	35,088	33,975	33,195	32,721	32,369
#pix_max	47,888	40,963	38,168	36,432	35,191	34,117	33,315	32,850	32,524
$\Delta\#pix$	0,035	0,069	0,030	0,055	0,103	0,142	0,120	0,129	0,155
Δd (mm)	0,074	0,355	0,256	0,686	1,834	3,618	4,149	5,468	7,752
isteresi %	0,248%	0,711%	0,365%	0,762%	1,667%	2,783%	2,766%	3,216%	4,080%

Figura 4.6 Tabella riassuntiva isteresi per i 3 target

Graficando l'andamento dell'isteresi in funzione della distanza si ottiene:

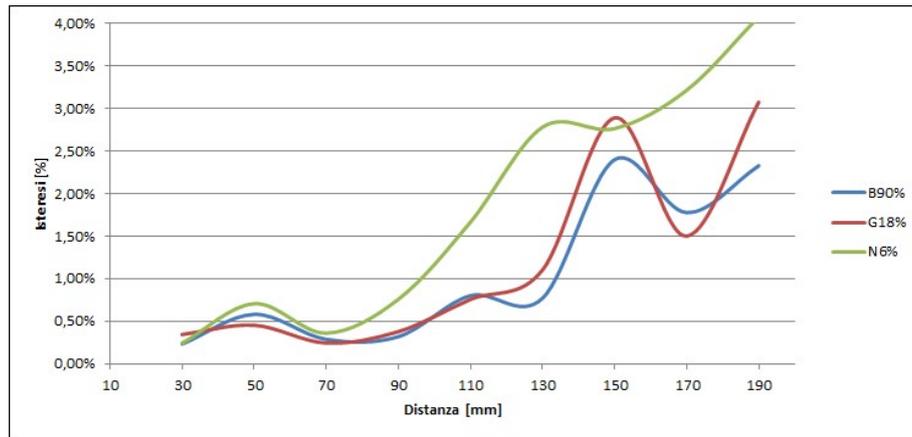


Figura 4.7 Andamento dell'isteresi al variare della distanza per i 3 target

Il grafico conferma ciò che analiticamente si era già dimostrato nella parte introduttiva riferita al PSD.

Successivamente si è fatto un tentativo per cercare di migliorare l'isteresi soltanto sulla distanza lontana (> 130 mm). Si sono campionati di nuovo 17 pixel, ma questa volta concentrati solo sui pixel in cui cade lo spot per le distanze lontane.

La programmazione del microcontrollore è identica con la sola differenza che si campionano i pixel dal 25 al 41.

Il procedimento e i calcoli sono gli stessi effettuati precedentemente.

Si ottiene:

B90%	130 mm	150 mm	170 mm	190 mm
#pix_min	34,153	33,466	33,014	32,545
#pix_max	34,204	33,560	33,063	32,651
Δ#pix	0,052	0,094	0,049	0,106
Δd (mm)	1,601	3,422	2,007	4,935
isteresi %	1,232%	2,282%	1,180%	2,597%
G18%	130 mm	150 mm	170 mm	190 mm
#pix_min	34,382	33,395	32,851	32,392
#pix_max	34,525	33,545	32,930	32,525
Δ#pix	0,142	0,149	0,079	0,133
Δd (mm)	4,133	5,483	3,377	6,417
isteresi %	3,179%	3,655%	1,986%	3,377%
N6%	130 mm	150 mm	170 mm	190 mm
#pix_min	34,488	33,148	32,669	32,361
#pix_max	34,609	33,307	32,828	32,534
Δ#pix	0,121	0,159	0,159	0,173
Δd (mm)	3,435	6,237	7,084	8,391
isteresi %	2,642%	4,158%	4,167%	4,416%

Figura 4.8 Tabella riassuntiva isteresi per i 3 target

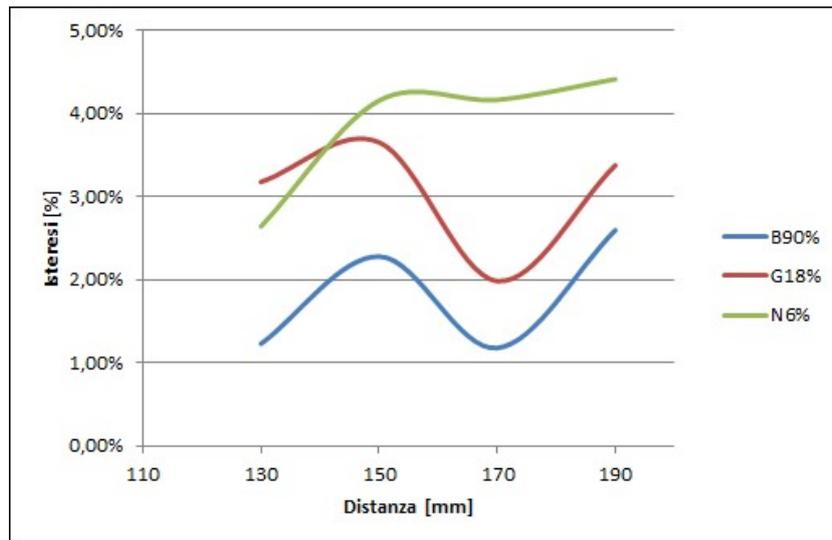


Figura 4.9 Andamento dell'isteresi solo sulle distanze lontane per i 3 target

Il picco di isteresi è ancora intorno al 4% per la distanza di 190 mm con target nero6%.

La situazione non è cambiata ma era ciò che ci si aspettava dalla teoria e si è voluto dimostrare con un'applicazione pratica: l'isteresi dipende da $\frac{1}{x^2}$ e non dalla scelta dei pixel da campionare.

5. Conclusioni e sviluppi futuri

Il presente elaborato di tesi aveva come scopo finale lo studio e l'approfondimento dei metodi realizzativi di sensori BGS basati su array di fotodiodi.

L'obiettivo era rilevare la presenza di un oggetto in una determinata posizione, attraverso la stima della sua distanza, ottenuta con il miglior compromesso tra accuratezza e velocità di acquisizione ed elaborazione.

Per arrivare ad un risultato soddisfacente sono state eseguite prove intermedie.

Si è innanzitutto approfondito il comportamento di un ricevitore composto da array di fotodiodi partendo dalla scansione totale, tramite la programmazione di un microcontrollore che ne gestisce il funzionamento.

Tali misure hanno poi permesso di fare considerazioni sull'elaborazione, la velocità di acquisizione e sulle soluzioni per velocizzarla: le applicazioni dei sensori di prossimità richiedono infatti acquisizioni piuttosto veloci.

Da precisare che l'obiettivo non era l'ottimizzazione del tempo di acquisizione; si è cercato il limite di velocità compatibile col setup di misura e sono state fatte considerazioni per eventuali miglioramenti.

Un modo per velocizzare l'acquisizione è stato quello di procedere con una scansione parziale dell'array, considerando soltanto 17 pixel su 64.

Nonostante siano stati campionati solo circa $\frac{1}{4}$ dei pixel, la velocità di acquisizione non ha subito un miglioramento delle stesse proporzioni: il maggior vincolo della tecnologia ad array di fotodiodi infatti è che, anche se non si è interessati al valore di tutti i pixel, si è comunque obbligati a scansionare la totalità dell'array, allungando inevitabilmente i tempi di acquisizione.

Una volta raggiunto il limite di velocità compatibile col setup di misura, si è proceduto allo studio di un metodo per valutare la distanza dell'oggetto, in modo da rendere il sensore di prossimità indipendente dal colore del target.

La stima della distanza ottenuta presenta un'accuratezza massima del 4% rilevata per oggetti posti a distanze lontane, dove inevitabilmente le prestazioni sono inferiori.

Per ottenere tale stima, si è reso necessario approssimare una curva gaussiana con una funzione parabolica: un primo interessante sviluppo futuro potrebbe essere quello di ricercare un algoritmo di approssimazione più complesso e valutarne eventuali benefici sull'accuratezza della stima della distanza.

Sempre con lo stesso scopo, si potrebbe anche studiare e progettare un array non lineare, col quale è possibile variare il pitch dei fotodiodi a seconda delle necessità.

Questa soluzione si pensa possa risultare molto più efficiente e con un impatto maggiore sul tempo di acquisizione rispetto la scansione parziale adottata nell'attività di tesi.

Un ulteriore approfondimento andrebbe fatto sui metodi di generazione del clock cercando di trovare un setup e un interfaccia migliore tra array e microcontrollore che permetta di ottenere un segnale più veloce.

Il sistema di misura implementato nell'attività di tesi infatti, ha permesso di generare un clock utile alla scansione dell'array di circa 20 volte inferiore rispetto il limite massimo indicato da datasheet.

Bibliografia

[1] Graeme J., *Photodiode amplifiers: op amp solutions*, New York, McGraw Hill Professional, 1996.

[2] Fraden J., *Handbook of modern sensors: physics, designs, and applications*, San Diego CA, Springer, Novembre 2003

[3] Application note, *Using photodetectors for position sensing*, United Detector Technology, Dicembre 1988

[4] Application note, *Position Sensing Detectors*, Osi Optoelectronics

[5] Application note, *Optical position sensing using silicon photodetectors*, Osi Optoelectronics

[6] Baccarani G., Rudan M., *Sensori di immagini a stato solido*, Bologna, Alta Frequenza, Vol. 48, No.1, Gennaio 1979

[7] datasheet array IC-LFM di IC-Haus

https://www.ichaus.de/upload/pdf/LFM_datasheet_A1en.pdf

[8] guida PSoC 4

<http://www.cypress.com/file/46056/download>

Ringraziamenti

Ci siamo! Stavolta è davvero finita! Mai più esami, mai più tesi, mai più università!! E cavoli, quasi quasi, mi tocca ammettere che questi 2 anni a Bologna sono “praticamente” volati!!!

Come sempre, tutto questo è stato possibile grazie al sostegno di ognuno di voi.

Grazie mamma e grazie babbo! Non mi avete mai fatto mancare niente e mi avete sempre fatto sentire il vostro supporto. Gli sforzi che avete fatto per me in questi anni spero un giorno di saperli ricambiare. Vi voglio bene e tutto questo, ancora una volta, è per voi.

Grazie Silvietta perché sei sempre presente, ti preoccupi sempre per me e mi sostieni sempre in tutto quello che faccio. Sai sempre essere di conforto anche se sono lontano. So che la distanza non ti fa piacere (e qui nota di merito a Teo che ti sopporta!) ma noi rimarremo sempre vicini! Ti voglio bene lo sai.

Grazie Valentina perché più di tutti mi sei stata vicino e mi hai sempre sostenuto in questo traguardo così importante.

Grazie Salvatore perché mi hai supportato per tutto il tempo della mia esperienza positivissima con voi, permettendomi di lavorare con tranquillità e serenità e con spesso il sorriso sulle labbra. Sono onorato di essere stato il tuo (credo!) ultimo tesista.

Grazie mbare, Dani, Piero, Tobia e tutti i dataregaz perché mi avete accolto alla grande, facendomi sentire uno dei vostri fin da subito. Le nostre strade purtroppo si divideranno, ma spero che la nostra amicizia rimanga intatta.

Grazie Michele e Mattia perché del gruppo dei cervesi siete senz'altro quelli più presenti e quelli che si interessano di più. E questo lo apprezzo veramente tanto.