ISSN/2499-5347

D. MURRA

Divisione Fisica della Fusione
Laboratorio Sorgenti, Diagnostiche
e Interazione Laser-Materia
Centro Ricerche Frascati, Roma

SCHEDE ARDUINO E SENSORI DI LUCE

Risultato dei test e realizzazione di uno strumento di misura

RT/2021/7/ENEA

AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

D. MURRA

Divisione Fisica della Fusione
Laboratorio Sorgenti, Diagnostiche
e Interazione Laser-Materia
Centro Ricerche Frascati, Roma

SCHEDE ARDUINO E SENSORI DI LUCE

Risultato dei test e realizzazione di uno strumento di misura

RT/2021/7/ENEA

AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

SCHEDE ARDUINO E SENSORI DI LUCE
Risultato dei test e realizzazione di uno strumento di misura

D. Murra

Riassunto
Le schede elettroniche denominate "Arduino" rappresentano una soluzione semplice, a basso costo ed estremamente versatile, che consente di realizzare strumentazione da laboratorio utile sia per la misura di diverse grandezze fisiche sia per l'automazione di alcuni processi.
In questo Rapporto Tecnico verrà dapprima esposta una panoramica delle potenzialità di tali schede, insieme ad una serie di esperimenti in cui sono stati usati con successo degli strumenti fatti ad hoc, per poi passare alla descrizione dell'interfacciamento di Arduino con misuratori di luce a molti sensori. Tali sensori possono essere in configurazione lineare (array) o a matrice (fotocamere di tipo CCD o CMOS) e sono dotati di decine, centinaia o anche migliaia di microelementi di dimensioni che vanno da pochi micron a qualche decina di micron. La gestione e la manipolazione di questi sensori richiede un'elettronica particolarmente accurata nei tempi, poiché funzionano grazie ad impulsi regolari inviati loro dall'esterno, ed una capacità di digitalizzazione ed archiviazione dei segnali di tensione che deve avvenire in modo accurato e veloce.
Infine, sarà illustrato uno strumento di misura completo, realizzato con una di tali schede, in grado di replicare con successo la bussola solare recentemente brevettata dall'ENEA.

Parole chiave: Arduino, fotosensori, rivelatori, attuatori.

Abstract
The electronic boards known as "Arduino" represent a simple, low cost and extremely versatile solution, which allows to realize laboratory instrumentation useful both for the measurement of many physical quantities and for the automation of some processes.
In this Technical Report, an overview of the potentiality of these boards will first be presented, together with a series of experiments in which some tools have been successfully used, to move on to the description of the interfacing of Arduino with light multi-sensors. These sensors can be in linear (array) or matrix (CCD or CMOS type cameras) configuration and are equipped with tens, hundreds or even thousands of microelements whose size ranges from a few microns to a few tens of microns. The management and manipulation of these sensors requires electronics that have high time accuracy, since they work thanks to regular pulses sent by an external source, and a capacity to digitize and store voltage signals that must be carried out accurately and quickly.
Finally, a complete measuring instrument will be shown, made with one of these cards, capable of successfully replicating the solar compass recently patented by ENEA.

Keywords: *Arduino board, photosensors, detector, electronic driver.*

INDICE

Schede Arduino e sensori di luce: risultato dei test e realizzazione di uno strumento di misura

1. Introduzione

La piattaforma hardware denominata ‘Arduino’ è ormai entrata come standard non solo tra gli appassionati di elettronica ma anche nei laboratori di ricerca grazie alla sua semplicità di utilizzo, alla versatilità delle sue applicazioni, alle innumerevoli periferiche che vi possono essere collegate e all’immensa letteratura costituita da codici, librerie, forum ed aiuti di ogni genere che possono essere reperiti in internet [1].

In questo Rapporto Tecnico faremo una breve introduzione sull’hardware che costituisce la scheda e su alcune delle applicazioni utili in un laboratorio di ricerca e ci concentreremo sull’uso di tale scheda nell’interfacciamento con sensori di luce ad array ed a matrice. Il motivo di questa ricerca nasce dalla possibilità di usare un sensore ad array per misurare con estrema accuratezza l’angolo con cui una sorgente di luce viene vista da un determinato punto di osservazione, esattamente ciò che avviene nel caso della bussola solare che l’ENEA ha progettato e brevettato qualche anno fa [2-5]. Nel caso della bussola solare, il sensore di luce era inizialmente costituito da una webcam e il controller da un pc portatile, poi sostituiti rispettivamente da un sensore a matrice e da un microprocessore, i quali comunicano tramite protocollo seriale. La lentezza dell’acquisizione di un fotogramma necessario per misurare la posizione del sole, scaricato ed analizzato in circa 10 secondi, ha portato alla ricerca di eventuali alternative, sia per il sensore che per l’elettronica, eventualmente migliorando anche altri aspetti del dispositivo, quali l’uso di un display grafico o la comunicazione con un cellulare attraverso il bluetooth. La sostituzione dei componenti è stata inoltre dettata dall’impossibilità di reperire gli stessi quando è sorta la necessità della realizzazione di un secondo esemplare della bussola.

Da questo spunto iniziale è scaturita la ricerca ai sensori e alla combinazione tra sensore e scheda elettronica. Da una parte la scelta è caduta su una scheda Raspberry [6]

interfacciata con una fotocamera CCD (Charge Coupled Display), di cui esistono esemplari appositamente dedicati a tale scheda, e dall'altra su una scheda Arduino e sensori lineari. Nel resto del Rapporto ci occuperemo esclusivamente di questa seconda opzione.

2. La piattaforma Arduino

L'idea di realizzare una scheda elettronica con un microprocessore incorporato e in cui fossero presenti un regolatore di tensione, il collegamento ai pin tramite semplici connettori a spina, una porta seriale per l'interfacciamento con un computer e la presenza di un ambiente di sviluppo semplificato con cui poter programmare il processore, nasce ad Ivrea nel 2005 grazie all'intuizione di cinque brillanti inventori, tra insegnanti e studenti, impegnati presso l'Interaction Design Institute, una scuola post-laurea. Il nome della piattaforma prende il nome dall'omonimo bar di Ivrea in cui gli inventori si incontravano.
Il progetto si è da subito rivelato vincente e ad oggi si contano più di 15 tipologie di schede ufficiali Arduino, innumerevoli schede compatibili e una schiera di appassionati e professionisti difficilmente quantificabile. È sufficiente considerare che nel forum ufficiale del sito di Arduino il numero degli argomenti che sono stati trattati dagli iscritti è più di mezzo milione e quello dei messaggi scambiati è superiore a quattro milioni e mezzo!
Il cuore di una scheda elettronica 'intelligente' è il processore e la maggior parte delle schede Arduino sfrutta un microprocessore della famiglia ATmega della Atmel. Questo processore ha una frequenza di clock pari a 16 MHz, lavora ad 8 bit, possiede una memoria flash (l'analogo dell'hard disk di un computer, dove va a risiedere il programma) che può andare da 16 a 256 megabyte ed una memoria ad accesso casuale (cioè la RAM, quella in cui vengono caricati il programma e i dati una volta alimentato il processore) che va da 1 a 8 kilobyte. Attualmente, il processore con maggiori performance di cui è dotata una piattaforma Arduino è l'Atmel SAM3X8E, un processore della famiglia ARM con un clock di 84 MHz, una memoria flash di 512 kilobyte ed una RAM di 96 kilobyte, che costituisce il cuore della scheda Arduino DUE.
Il confronto tra un processore del genere ed una qualsiasi CPU di un moderno computer o smartphone è impietoso ma si deve tener conto che un microprocessore programmabile prevede che giri un solo programma (quello che vi è stato caricato) e che non sia corredato di sistema operativo. Per le applicazioni che vedremo nel seguito del Rapporto, si

apprezzerà il fatto che, nonostante il limitato bagaglio di memoria e la bassa velocità di clock, questi microprocessori sono in grado di eseguire compiti di particolare complessità.
Gli elementi che compongono una scheda Arduino, oltre al microprocessore, sono essenzialmente: un regolatore di tensione, un convertitore usb-seriale per la comunicazione con un computer, una serie di connettori per gli ingressi ed uscite digitali, una serie di connettori per gli ingressi analogici (collegati agli ADC, cioè i convertitori analogico-digitali, del processore), i connettori destinati alla comunicazione via seriale (UART, I^2C e SPI) e infine dei connettori di tensione (5V, 3.3V, tensione di riferimento per gli ADC e massa).
Il numero di connessioni dipende dal tipo di processore e di scheda. Nelle schede più piccole (denominate NANO e UNO) vi sono 14 pin digitali e 6 o 8 pin analogici. In quelle intermedie come la Arduino MEGA2560 e in quella con processore ARM, l'Arduino DUE, vi sono 54 pin digitali e 16 o 12 pin analogici rispettivamente.
In figura 1 è mostrata la foto di una scheda Arduino UNO, in cui sono indicati i vari elementi che la costituiscono.
La programmazione del microprocessore presente sulle schede Arduino è possibile tramite la classica connessione ad un programmatore esterno, come avviene per i comuni PIC (Programmable Interrupt Controller) oppure, più semplicemente, attraverso l'ambiente di sviluppo dedicato che, una volta installato su PC, consente di editare un programma in C/C++ e di inviarlo, una volta compilato, direttamente sulla scheda tramite la porta USB-Seriale.

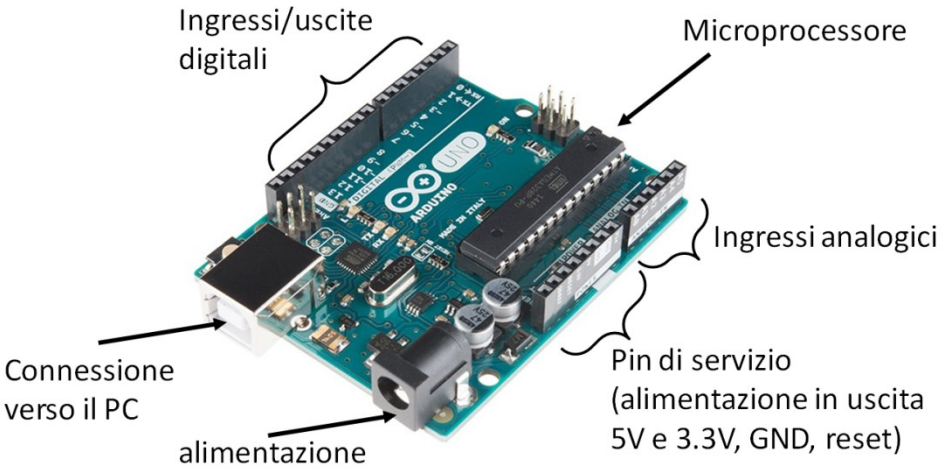


Figura 1: la scheda Arduino UNO con l'indicazione degli elementi principali

Dal numero di porte digitali (input/output) ed analogiche (solo input, tranne che per la scheda DUE, in cui sono anche output) si deduce che le applicazioni in cui sia necessario collegare molti sensori o attuatori non rappresentano un problema.
Nel prossimo paragrafo presentiamo una piccola panoramica dell'uso che si è fatto di questo semplice hardware nel laboratorio Sorgenti, Diagnostiche e Interazione Laser-Materia del Centro ENEA di Frascati e delle possibili applicazioni in un comune laboratorio di ricerca.

3. Applicazioni di Arduino

La versatilità dei microprocessori è tale per cui molte semplici misure possono essere eseguite contemporaneamente da un'unica scheda corredata degli appositi sensori. Ed è altrettanto possibile utilizzare Arduino per agire attivamente sull'esterno, muovendo motori, accendendo luci o inviando impulsi elettrici.
Grazie all'estrema diffusione di tali schede (e di molte altre che sfruttano la stessa idea) sul mercato si trovano decine di dispositivi per le più svariate applicazioni. Si va dai sensori di temperatura, analogici o digitali, ai sensori di luce, dai rivelatori di gas a quelli di umidità, da quelli a effetto Hall a quelli che misurano le vibrazioni e ancora inclinometri, pulsanti a sfioramento, ricetrasmettitori a radiofrequenza e a ultrasuoni, moduli FM, encoder, lettori di chip a radiofrequenza (RFID) e così via. Ed anche gli attuatori non mancano: motori passo-passo o in corrente continua, attuatori lineari, moduli RGB, display alfanumerici e grafici, buzzer, amplificatori, moduli MIDI per la sintesi di suoni, relé, telecomandi a infrarossi…
Per sistemi che richiedono una discreta potenza elettrica, le schede Arduino, che possono erogare al massimo solo circa 40 mA sui pin digitali, vanno collegate a circuiti elettronici pilotati dalla scheda ma alimentati dall'esterno. Molti degli attuatori presenti sul mercato e dedicati ad Arduino, come i driver per i motori passo-passo o i relé, sono già dotati dell'elettronica adatta, così da minimizzare eventuali disegni elettrici aggiuntivi.
Qui accenneremo ad alcuni lavori in cui è stata utilizzata una scheda Arduino nell'ambito di progetti di ricerca.

3.1 Arduino come controllore di luminosità e temperatura

Il primo esempio riguarda un esperimento di irraggiamento di materiali per la misura del degrado dovuto all'irraggiamento solare fuori dell'atmosfera [7]. A tale scopo è stata

utilizzata una lampada a mercurio che emette luce ultravioletta (principalmente alla lunghezza d'onda di 254 nm) a simulare la radiazione solare in quella regione ed una camera da vuoto in cui erano inseriti i campioni da irraggiare.
Durante il processo di irraggiamento, durato molti giorni, era indispensabile conoscere sia la temperatura dei campioni (che, secondo la richiesta del committente, non doveva superare i 40 °C) sia l'andamento della potenza luminosa della lampada.
Per far ciò abbiamo utilizzato una scheda Arduino NANO, tre comuni fotoresistenze a semiconduttore ed una sonda digitale di temperatura. Le fotoresistenze sono state poste in 3 punti distinti, fuori dalla camera di irraggiamento, indirizzate verso la sorgente luminosa e schermate da 3 diversi filtri colorati, per verificare che non ci fosse né un abbassamento della luminosità né uno spostamento tra le righe di emissione della lampada. La sonda di temperatura (modello DS18B20) è stata posta sul fondo della camera da vuoto, dove erano collocati i campioni, e i suoi tre terminali sono stati collegati all'esterno tramite un doppio passante di tipo BNC. Ognuna delle fotoresistenze è stata collegata ad circuito simile a quello indicato in figura 2 e la caduta di tensione era controllata dal convertitore analogico-digitale di Arduino. La sonda di temperatura era invece connessa ad un terminale di input/output digitale e comunicava tramite protocollo "*one-wire*" ovvero con richiesta, da parte di Arduino, e risposta, della sonda, viaggianti sullo stesso cavo. Durante l'acquisizione dei dati, la scheda Arduino comunicava ad un computer i risultati dei rilevamenti che venivano poi memorizzati su file.

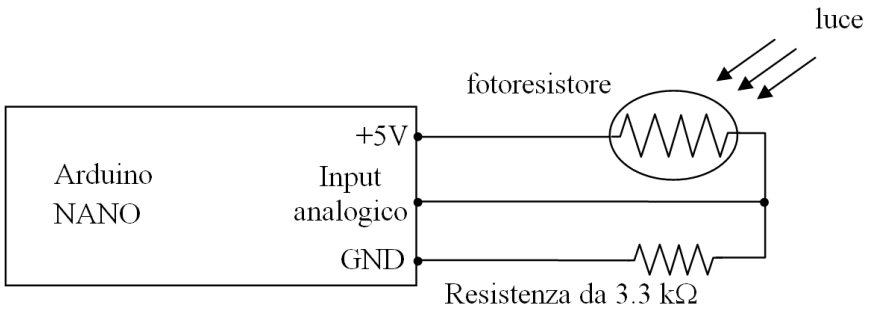


Figura 2: schema del circuito per il controllo dell'intensità luminosa tramite fotoresistori

Questo sistema ha funzionato ininterrottamente, senza alcun problema tecnico, prima per circa 14 giorni e poi per altri 10 giorni, intervallati da quattro giorni in cui l'esperimento è stato fermato per via della chiusura del laboratorio dovuta a delle festività ed i risultati sono stati pubblicati su una rivista internazionale [8].

3.2 Arduino come pilota di motori lineari

Un secondo impiego della piattaforma Arduino è stato quello di driver di attuatori lineari per la simulazione di movimenti di una struttura, operazione utile per verificare il comportamento di alcuni sensori di stress meccanico a fibra ottica.
Una scheda Arduino UNO è stata interfacciata con due motori lineari pilotati tramite un'alimentazione a 24 volt selezionata tramite 4 relé, 2 per ogni motore. Un relé serviva a scollegare o ricollegare l'alimentazione principale mentre l'altro era usato per scambiare i poli dell'alimentazione in modo da poter far muovere l'attuatore sia in avanti che all'indietro. La scheda, inserita in un contenitore in plastica, era controllata tramite 4 pulsanti e mostrava le informazioni su un display alfanumerico. Il programma inserito nel processore di Arduino consentiva di muovere i motori in modo manuale o in modo ciclico, impostando il numero di cicli e la corsa. Dopo il test positivo, il box con gli alimentatori, Arduino e la console di comando è stata consegnata ai colleghi di FSN-TECFIS-MNF che lo hanno usato nell'ambito di un progetto [9] finalizzato allo studio sperimentale di sistemi per l'isolamento sismico di strutture dotati di sensori in fibra ottica (vedi figura 3).

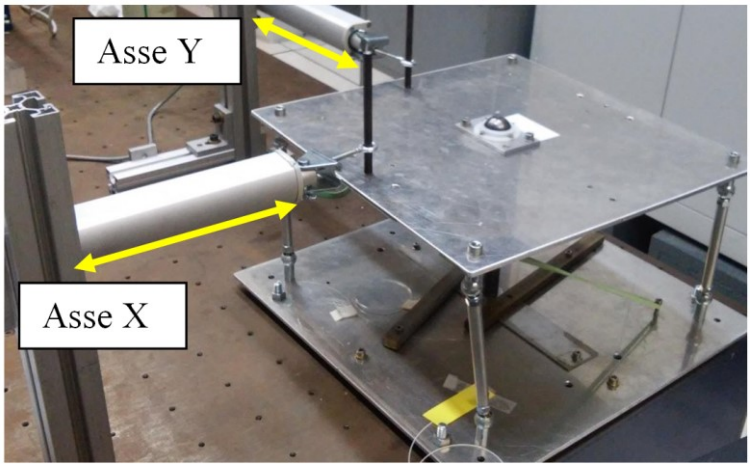


Figura 3: dispositivo per la sollecitazione di una struttura di prova al fine di testare degli isolatori sismici. Il movimento dei bracci dei due motori visibili in figura era comandato da una scheda Arduino.

3.3 Arduino come misuratore di profilo di intensità luminosa

Per un'applicazione legata ad un sistema di sanificazione tramite led UVC, è stata di nuovo utilizzata una scheda Arduino. In questo caso era necessario misurare la distribuzione

angolare dell'energia luminosa emessa da un led e la scelta è caduta su un sistema in cui il led veniva fatto ruotare lungo un asse verticale mentre il rivelatore di luce, un fotodiodo, era fisso. Ad ogni posizione angolare del motore che muoveva il led, il fotodiodo forniva un valore di corrente proporzionale all'intensità luminosa che lo colpiva. Per realizzare tale esperimento è stato sufficiente equipaggiare Arduino con un motore passo-passo e collegare l'uscita del fotodiodo ad una resistenza elettrica ai capi della quale è stato connesso un ingresso analogico della scheda.

Il programma di Arduino muoveva il motore di un determinato angolo e, nello stesso momento, misurava la tensione proveniente dal fotodiodo. Piuttosto che memorizzare internamente il dato si è preferito inviare i valori (angolo e tensione) ad un computer per il loro salvataggio. Il risultato ottenuto ha confermato la curva presente nel datasheet del produttore del led, così come si può apprezzare nel grafico di figura 4.

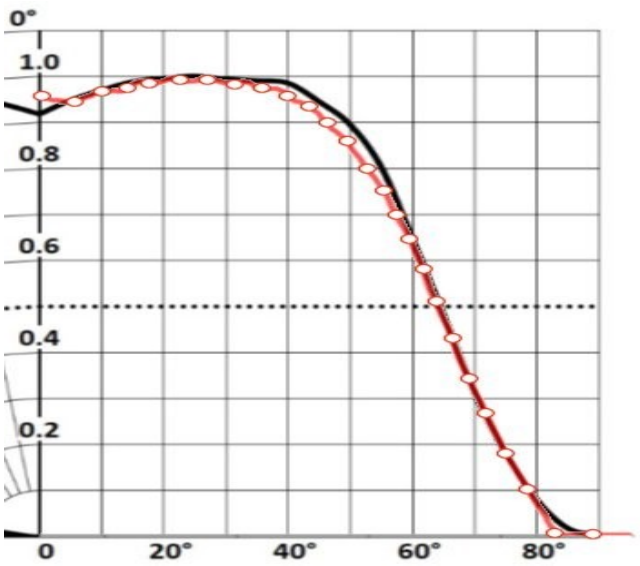


Figura 4: confronto tra la distribuzione angolare teorica (curva nera) e sperimentale (cerchi e curva rossa) del led UV sotto test.

Come è possibile dedurre da questa applicazioni, sono notevoli i vantaggi dell'uso di una scheda programmabile dotata di parecchie interfacce e a cui è applicabile un gran numero di sensori, a cominciare dalla semplicità di realizzazione del circuito elettrico e dell'uso dell'ambiente di programmazione fino alla velocità di acquisizione dei dati ed all'affidabilità di tutto l'apparato di misura uniti ad un costo veramente modesto dei componenti.

Anche quando le esigenze in termini di velocità di clock e di capacità di memoria digitale sono più stringenti, la soluzione semplificata tramite schede Arduino è comunque percorribile. Nei prossimi paragrafi viene illustrata la fattibilità dell'uso di tali schede per pilotare sensori di luce costituiti da centinaia o migliaia di sensori, come negli array lineari o nella matrici. In questi casi, infatti, la necessità di fornire un segnale di clock a decine di kHz, o più, è indispensabile per il corretto funzionamento dei sensori ed inoltre la quantità di dati in arrivo da tali sensori può superare la capacità di memoria degli stessi processori.

4. Tipologie di sensori di luce

Il sensore di luce più semplice lo abbiamo già citato nel precedente paragrafo ed è un fotoresistore, la cui resistenza elettrica varia in funzione dell'energia luminosa che lo colpisce. Una volta inserito tale sensore all'interno di un circuito elettrico, la misura dell'intensità luminosa avviene osservando la caduta di tensione ai capi del foto resistore (vedi fig. 2).

Altri comuni sensori di luce sono i fotodiodi, costituiti da una giunzione a semiconduttore, in cui il segnale luminoso consente il passaggio di corrente attraverso la giunzione che, alimentata da un generatore di tensione, fornisce un segnale da inviare su una resistenza ai capi della quale va misurata la caduta di tensione. Rispetto al fotoresistore il principio di funzionamento è diverso ma, dal punto di vista della misura, è molto simile.

Diversa è la situazione per i sensori ad array o a matrice. In questo caso ci sono centinaia di fotodiodi allineati lungo una riga o migliaia di celle incasellate in una griglia. Tali sensori sono internamente dotati di un circuito elettrico che provvede a 'scaricare' il segnale proveniente dai singoli micro rivelatori e ad inviarlo sequenzialmente al processore esterno.

Nell'introduzione è stato già spiegato che, nel nostro caso, la necessità di usare un sensore di questo tipo è nata per poter misurare con estrema precisione la direzione di arrivo dei raggi solari rispetto ad un piano verticale di riferimento. Da tale misura, unita al calcolo delle effemeridi, è possibile ricavare la direzione del polo Nord geografico e quindi è utile per realizzare una bussola solare.

Accanto all'opzione Raspberry-CCD (con questa accoppiata è già stata realizzata una nuova bussola da usarsi con impianti per lo sfruttamento dell'energia solare a concentrazione) è stata presa in considerazione l'ipotesi di utilizzare la piattaforma Arduino.

Di conseguenza, è stata fatta un'indagine per verificare quali sensori fossero disponibili sul mercato e quale di questi potevano essere compatibili con l'hardware Arduino.
I sensori lineari presi in considerazione sono stati i seguenti modelli:

- TSL1401CL della AMS
- LF1401 della IC HAUS
- S9226 della Hamamatsu
- ILX554A della Sony
- TCD1304DG della Toshiba

È stata anche valutata l'opzione di usare una matrice di sensori, in particolare la fotocamera CMOS (Complementary Metal-Oxide Semiconductor) OV7670 della Omnivision.
Nella tabella 1 sono riportate le caratteristiche essenziali dei vari sensori.

Modello	Numero di pixel	Spaziatura pixel (μm)	Lunghezza area sensibile (mm)	Clock esterno minimo (μs)	Clock esterno massimo (μs)	Costo indicativo (€)
TSL1401CL (AMS)	128	63.5	8.13	0.125	200	10
LF1401 (IC Haus)	128	63.5	8.13	0.2	senza limite	13
S9226 (Hamamatsu)	1024	7.8	8	1.25	100	75
ILX554A (Sony)	2048	14	28.7	0.5	senza limite	10*
TCD1304DG (Toshiba)	3648	8	29.2	0.42	1.25	25
OV7670 (Omnivision)	640x480	3.7 x 3.7	2.36 x 1.76	0.02	1**	10

*Questo chip è fuori produzione ma ancora è reperibile online da venditori di componenti elettronici

**Si riferisce al clock di lettura del chip AL422B, una memoria esterna usata come buffer per i dati. Il clock massimo per la CMOS è di 0.1 μs.

Tabella 1: caratteristiche ottiche, elettriche e costi dei sensori sottoposti a test

Come si può vedere dalla tabella, a parte i sensori della AMS e della IC Haus che sono praticamente identici, le caratteristiche variano molto, sia per quanto riguarda la geometria, in particolare per la dimensione dei pixel, sia per la velocità del clock da inviare per far funzionare il circuito interno del chip che pilota i fotodiodi.

Il costo di questi sensori è sempre piuttosto contenuto, diversi di essi sono reperibili sui cataloghi dei grandi distributori che vendono su internet (RS, Distrelec, Mouser) mentre altri si trovano su siti di vendite online (e-bay, Amazon).
In funzione del tipo di misura che si intende fare, dunque, si ha una discreta possibilità di scelta. Una richiesta di elevata risoluzione porta a scegliere i chip della Sony o della Toshiba, mentre la necessità di usare un clock basso, pur mantenendo l'esecuzione di una misura entro poche frazioni di secondo, richiede un numero non troppo elevato di pixel, per cui la scelta cadrebbe sull'array dell'Hamamatsu o sui due gemelli della AMS o della IC Haus.
Chiaramente, se lo scopo della misura è quello di ottenere un'immagine, o comunque un'informazione bidimensionale, l'unica possibilità è quella di optare per la CMOS camera.
Come vedremo nel prossimo paragrafo, con Arduino è stato possibile pilotare tutti i sensori in tabella tranne quello della Toshiba, probabilmente per problemi di sincronizzazione degli impulsi. In particolare, tramite una scheda Arduino DUE interfacciata al sensore della Hamamatsu è stata realizzata l'elettronica per una bussola solare che è stata consegnata ai colleghi dell'INGV (Istituto Nazionale di Geofisica e Vulcanologia) che si occupano delle misure geomagnetiche.

5. Le caratteristiche di Arduino e i requisiti dei sensori lineari

Gli array di fotodiodi sono costituiti dalla parte sensibile alla luce, dove sono allineati centinaia o migliaia di sensori, e da un circuito elettronico che viene comandato esternamente e che provvede a regolare il funzionamento degli elementi elettro-ottici nonché ad inviare all'esterno i dati di tensione proporzionali all'intensità di luce. Tali dati vengono inviati sequenzialmente e la sincronizzazione dei tempi avviene tramite un segnale di clock fornito dal driver esterno. Nel caso della CMOS a matrice, non solo l'invio dei dati di luminosità ma anche il semplice funzionamento dell'elettronica che riceve i comandi di avvio acquisizione o di impostazione dei parametri di lavoro avviene grazie al segnale di clock esterno.
I pin di comunicazione tra driver e sensore sono sempre 3 per tutti i modelli, tranne che per quello della IC Haus e quello della Toshiba. I 3 pin rappresentano il segnale di clock (input), di start (input) e quello di uscita del segnale (output). Nel modello della IC Haus e in quello della Toshiba vi è un ulteriore segnale in input per regolare il tempo di integrazione della luce.

Il grafico temporale che rappresenta la sequenza dei segnali da inviare agli array è rappresentato in figura 5 nella sua forma più generale (la vera sequenza può variare di poco da modello a modello).

Oltre alla stabilità della frequenza di clock e alla necessità di rientrare tra il minimo ed il massimo valore di tempi consentito, vi sono altri requisiti temporali indispensabili affinché la comunicazione tra driver e sensore vada a buon fine. Tali requisiti consistono nella velocità del fronte di salita (e di discesa) del segnale di clock, la distanza temporale tra salita (o discesa) dell'impulso di start rispetto a quello di clock e la velocità di digitalizzazione del segnale di output, che deve concludersi entro la durata di mezzo periodo di clock circa per evitare uno sfasamento tra l'acquisizione di un pixel ed il successivo.

In figura 6 è mostrata, a titolo di esempio, l'indicazione riportata nel datasheet del modello della Hamamatsu a proposito della velocità dei fronti di salita e discesa e dei ritardi tra clock e start.

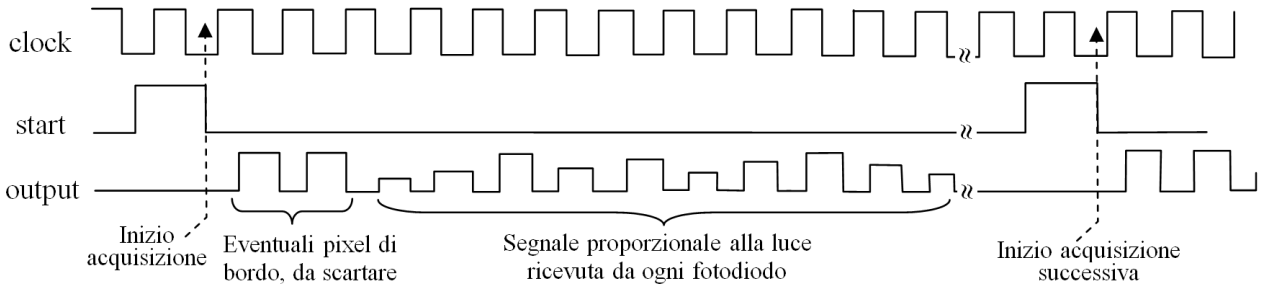
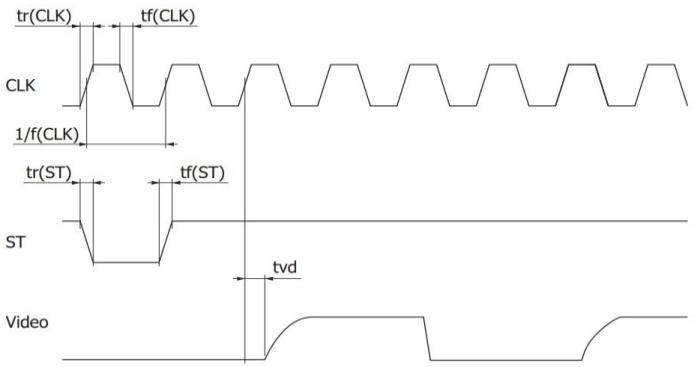


Figura 5: esempio di sequenza degli impulsi di clock, start e output per l'acquisizione del segnale luminoso.



Parameter	Symbol	Min.	Typ.	Max.	Unit
Start pulse interval	tpi(ST)	4104/f(CLK)	-	-	s
Start pulse rise and fall times	tr(ST), tf(ST)	0	20	30	ns
Clock pulse duty	-	40	50	60	%
Clock pulse rise and fall times	tr(CLK), tf(CLK)	0	20	30	ns
Video delay time*7	tvd	10	20	30	ns

*7: Ta=25 °C, Vdd=5 V, V(CLK)=V(ST)=5 V

Figura 6: requisiti relativi alla velocità dei fronti di salita e di discesa nel sensore della Hamamatsu

L'elettronica incaricata di pilotare questi sensori, dunque, benché non si richiedano frequenze elevatissime, deve comunque essere in grado di mantenere entro il limite di poche decine di nanosecondi la transizione alto-basso degli stati logici delle uscite digitali, garantendo allo stesso momento l'invio stabile di un treno di impulsi con un periodo di pochi microsecondi e la contemporanea digitalizzazione e memorizzazione dei segnali analogici in ingresso.
La velocità di clock delle schede Arduino con il processore ATmega è di 16 MHz, quindi il tempo di esecuzione di una operazione elementare è dell'ordine dei 60 ns. Ma, come è ragionevole supporre, un'istruzione informatica richiede numerose operazioni elementari e, di conseguenza, alcuni periodi, o alcune decine di periodi, di tempo di esecuzione di una singola istruzione. Per requisiti temporali più stringenti è possibile optare per la scheda Arduino DUE che ha un clock circa 5 volte più veloce.
Il linguaggio usato per programmare il processore è stato il C/C++ e le istruzioni di alto livello per generare un segnale ad onda quadra sono le seguenti:

```
digitalWrite(10, HIGH);    //porta il pin 10 a livello logico '1'
digitalWrite(10, LOW);     //porta il pin 10 a livello logico '0'
```

Eseguendo un ciclo in cui queste istruzioni vengono inviate continuativamente al processore si ottiene un segnale ad onda quadra il cui periodo è pari a 9 μs. Ciò significa che il numero di cicli di clock del processore per tale istruzione è ben 150! Usando questo segnale come oscillatore per i sensori è chiaro che non sarebbero pilotabili né il chip della Toshiba né la CMOS camera (cfr. tabella 1). Per gli altri sensori questo valore rientra nei limiti e il trasferimento del segnale di tutti i punti avverrebbe in tempi più che accettabili (con un massimo di circa 37 ms per l'Hamamatsu, per il quale la lettura di un pixel avviene ogni quattro cicli di impulsi di clock).
Non è difficile, comunque, aumentare la frequenza dell'onda quadra emessa dai pin digitali di Arduino, ricorrendo alla manipolazione diretta dei registri del processore, senza passare per le istruzioni ad alto livello. Le stesse due istruzioni date in precedenza, se trasformate in quelle a basso livello diventerebbero le seguenti:

```
PORTB = 0b00000100;      //porta il pin 10 a livello logico '1'
PORTB = 0b00000000;      //porta il pin 10 a livello logico '0'
```

Questa sequenza di istruzioni genera un'onda quadra con un periodo di soli 125 ns, ovvero appena due cicli elementari.
In figura 7 sono riportate le forme d'onda, registrate da un oscilloscopio, corrispondenti ad un treno di impulsi inviati da una scheda Arduino UNO utilizzando rispettivamente le istruzioni ad alto e basso livello.

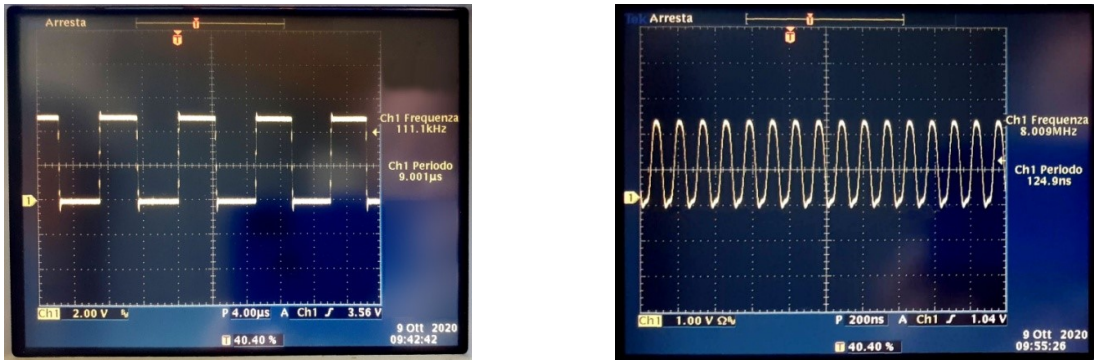


Figura 7: Un'onda quadra emessa da un pin di Arduino UNO al massimo della frequenza. A sinistra il segnale in uscita quando vengono usate istruzioni ad alto livello, a destra quando vengono manipolati direttamente i registri del processore. Da notare che la scala temporale della figura di destra è 40 volte più corta.

Se la velocità di clock sembra più che sufficiente, resta da verificare se anche quella di digitalizzazione del segnale in uscita dal sensore, inviato ad un ingresso analogico della scheda di Arduino, riesce a rientrare nei limiti.
Tale digitalizzazione deve essere inserita necessariamente all'interno del ciclo del clock, poiché, seppur sia possibile utilizzare due *thread* diversi per inviare il clock e per leggere il dato analogico (cioè due sottoprogrammi che lavorano contemporaneamente, uno dei quali si occupa del ciclo infinito di clock e l'altro procede con i cicli di lettura) vi è la necessità di sincronizzarli al livello del microsecondo e la separazione in diversi thread non garantisce il sincronismo.
Il ciclo di clock/lettura, dunque, è qualcosa che deve avvenire secondo questo schema:

- Invio segnale alto di clock
- Lettura e digitalizzazione del segnale analogico in uscita dal sensore
- Invio segnale basso di clock

In questo modo, però, se l'operazione di lettura ha tempi confrontabili con quelli del periodo di oscillazione del clock, gli impulsi non avrebbero più un duty cycle al 50%, ciò che è richiesto (anche se non in modo stringente) dall'elettronica dei sensori. Per riportare il ciclo al 50%, dunque, i tempi vanno necessariamente allungati.

Per conoscere il tempo di digitalizzazione di un segnale tramite istruzioni ad alto livello basta lanciare il seguente ciclo:

```
for (n=1; n<10000; n++)
    int valore=analogRead(pinAnalogico);
```

Questo ciclo di diecimila digitalizzazioni si completa in circa 1.1 secondi, pertanto la singola operazione avviene in 100 microsecondi, una velocità inaccettabile, visto che dovrebbe essere confrontabile almeno con i 9 μs del periodo del clock più lento.
Al di là del fatto che 100 microsecondi siano eccessivi anche per il chip della Hamamatsu (oltre che per quello della Toshiba), la lettura dei 1024 pixel avverrebbe in quasi mezzo secondo, il che sarebbe intollerabile qualora si pretendano diverse letture al secondo.
In realtà, questa velocità è limitata da un'impostazione dell'hardware di Arduino che è possibile modificare tramite poche istruzioni. In pratica, anche l'ADC ha un suo clock, il quale deriva dal clock principale (quello da 16 MHz) scalato di un determinato fattore. Tale fattore è, per impostazione di fabbrica, pari a 128, il quale, unito al numero di cicli usati dall'istruzione di digitalizzazione, che sono 13, porta ai 100 microsecondi misurati in precedenza. L'impostazione del fattore (*prescaler*) con cui dividere il clock principale ad un fattore, per esempio, pari a 16 (che porta il clock dell'ADC al valore di 1 MHz), si ha tramite le seguenti istruzioni [10]:

```
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
sbi(ADCSRA, ADPS2);
cbi(ADCSRA, ADPS1);
cbi(ADCSRA, ADPS0);
```

Impostando il prescaler a 16, il tempo per completare lo stesso ciclo visto in precedenza diventa di 150 ms, ovvero di 15 μs per ogni conversione analogico-digitale.
Se si spinge il clock dell'ADC fino a 2 MHz, un ciclo completo che preveda l'invio del clock al sensore, durante il quale effettuare la lettura e digitalizzazione del segnale di tensione, avrebbe un periodo di circa 16 μs (8 μs per la conversione ed altri 8 per avere un duty cycle al 50%).

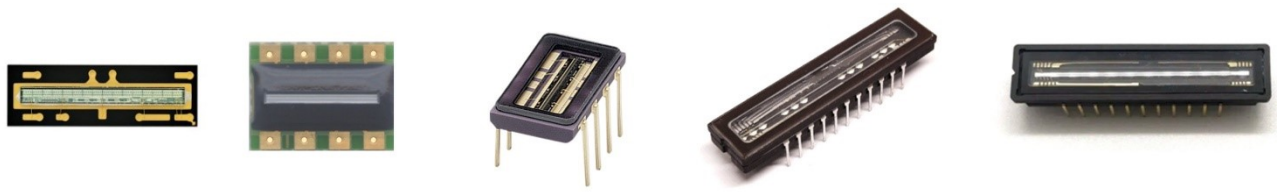
Utilizzando questo periodo, in tabella 2 sono riportati i tempi di una scansione completa per ognuno dei sensori lineari (anche per quelli che non funzionerebbero con un clock così lento). Per completezza si riportano le stesse informazioni qualora si utilizzi la scheda Arduino DUE (ottimizzando i tempi dell'ADC).
Se l'interfacciamento con un tale sensore di luce non richiede una velocità di refresh particolarmente elevata, già la scheda con il processore a 16 MHz è sufficiente allo scopo.

Modello	Numero di pixel	Cicli di clock per leggere un pixel	Arduino UNO Tempo di acquisizione (ms)	Arduino UNO Frequenza di acquisizione (Hz)	Arduino DUE Tempo di acquisizione (ms)	Arduino DUE Frequenza di acquisizione (Hz)
S9226	1024	4	66	15	12.3	81.4
TSL1401CL	128	1	2	500	0.4	2500
LF1401	128	1	2	500	0.4	2500
ILX554A	2048	1	33	30	6.2	162.8
TCD1304D	3648	4	233	4	87.6	11.4

Tabella 2: tempi di acquisizione per uno scan completo degli array da testare, utilizzando un clock di 16 μs per la scheda UNO e il minor clock possibile (circa 3 μs) per la scheda DUE.

6. Esito delle prove fatte sui sensori lineari

Tutti i sensori presenti nella tabella 4.1, e mostrati in figura 8, sono stati interfacciati ad una scheda Arduino per verificare la capacità di tale scheda di pilotare i sensori e di acquisire i dati, nonché per valutare l'affidabilità dell'apparato quale sistema di misura.



(a) (b) (c) (d) (e)

Figura 8: fotografia dei cinque sensori interfacciati con schede Arduino.
Da sinistra: a) TSL1401CL, b) LF1401, c) S9926, d) ILX554A, e) TCD1304D.

6.1 TSL1401CL

Il primo sensore ad essere stato sottoposto a test è stato il modello TSL1401CL dell'AMS [11]. Questo sensore, predisposto per la saldatura superficiale, è stato saldato a dei fili semirigidi e inserito su una scheda prototipale. Per il test la scheda è stata poi inserita in un contenitore che schermava la luce tranne per una fenditura da circa 2 mm di larghezza centrata sopra il sensore a circa 2 cm di distanza. I requisiti temporali richiesti da questo array sono piuttosto blandi ed il numero di pixel è basso per cui si è scelto di lavorare ad una frequenza di clock non particolarmente elevata, utilizzando esclusivamente istruzioni ad alto livello.
La sequenza degli impulsi inviati al sensore è illustrata nel grafico della figura 9.
Quando viene dato il via all'acquisizione, Arduino invia un segnale di START, in modo da far scaricare il contenuto che i fotodiodi hanno accumulato in precedenza e poter ripartire da zero. Al secondo segnale di START la luce che viene trasformata in segnali di tensione è quella accumulata tra i due impulsi di START. In questo sensore il tempo minimo di integrazione della luce è pari a 110 volte il tempo di clock a cui vanno aggiunti 20 microsecondi. Usando istruzioni ad alto livello, quindi con un clock minimo di 9 μs, il minimo tempo di integrazione è pari a 1 ms.

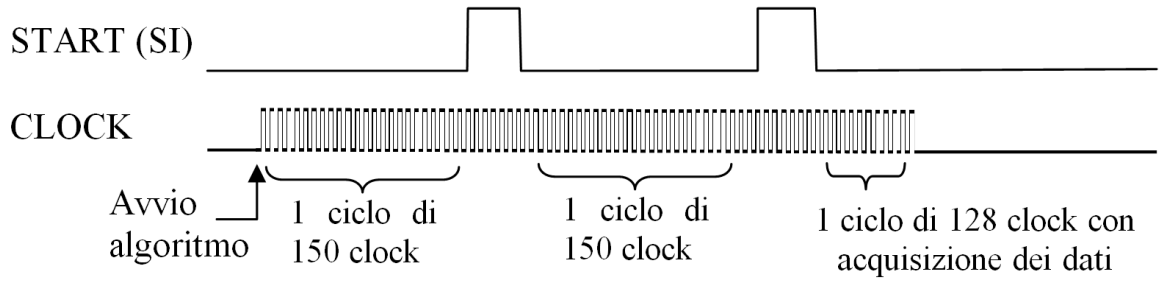


Figura 9: sequenza degli impulsi inviati da Arduino al sensore TSL1401CL per l'acquisizione del segnale dei 128 pixel

In totale, la durata di una acquisizione è pari a 2 cicli di 150 segnali di clock, più un altro ciclo di 128 impulsi con la lettura della tensione. Per poter regolare la quantità di luce registrata dal sensore, in modo da sfruttare il massimo del rapporto segnale/rumore, nel programma è stato inserito un parametro che consente di aumentare o diminuire il periodo

del clock e quindi modificare il tempo di integrazione del segnale. In questo modo, il tempo per un'acquisizione completa va da un minimo di 16 ms ad un massimo di 40 ms.
Questo sistema ha subito dato dei buoni risultati per cui, tramite stampa 3D, è stato realizzato un contenitore in cui è stata inserita una scheda NANO, un sensore GPS, uno scomparto per batterie e un punto di ancoraggio per la scheda prototipale su cui era stato montato il sensore TSL1401. Il sensore era inclinato a circa 45° rispetto all'orizzontale e, a pochi millimetri, sulla copertura, anch'essa a 45° è stata predisposta una fenditura di circa 250 μm. In questo modo, è stata praticamente realizzata una bussola solare portatile.
In figura 10 sono riportate due fotografie di tale prototipo.

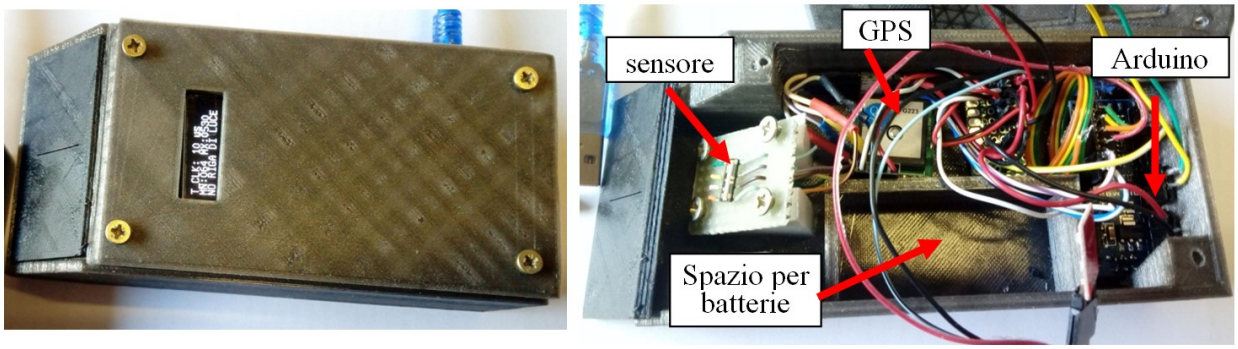


Figura 10: un prototipo di bussola portatile realizzata con stampante 3D. A sinistra, la scatola è chiusa, si vede il display e sulla parete inclinata di sinistra, la fenditura. A destra è stato rimosso il coperchio e sono visibili il sensore di luce, il sensore GPS, lo spazio per le batterie e la scheda Arduino NANO. Le dimensioni della scatola sono 13x6x4 cm (lunghezza, larghezza, altezza).

Affinché la bussola dia risultati affidabili occorre tararla ed il procedimento è piuttosto laborioso. A parte l'accuratezza dell'indicazione del Nord, comunque, era importante verificare che il segnale ricevuto dal sensore, una riga di luce filtrante attraverso la fenditura, fosse ben distinto, riproducibile e la cui elaborazione desse un valore di centro riga ripetibile e con un basso errore.
La riprova della buona risposta del sensore è illustrata in figura 11 dove, da una parte, è mostrato il profilo della riga di luce registrato dal sensore e, dall'altra, lo scarto del valore corrispondente al centro riga (misurato matematicamente come baricentro della curva), in 60 acquisizioni di una riga di luce fornita da una sorgente fissa. Lo scarto è espresso in primi d'arco e, come si può apprezzare, l'errore è contenuto entro +/- 1'.
In conclusione, l'array TSL1401CL non presenta problemi nel suo pilotaggio tramite schede Arduino dotate di processore ATmega328, con frequenze di lettura che, senza necessità di

utilizzare codice a basso livello, possono arrivare a oltre 60 Hz.

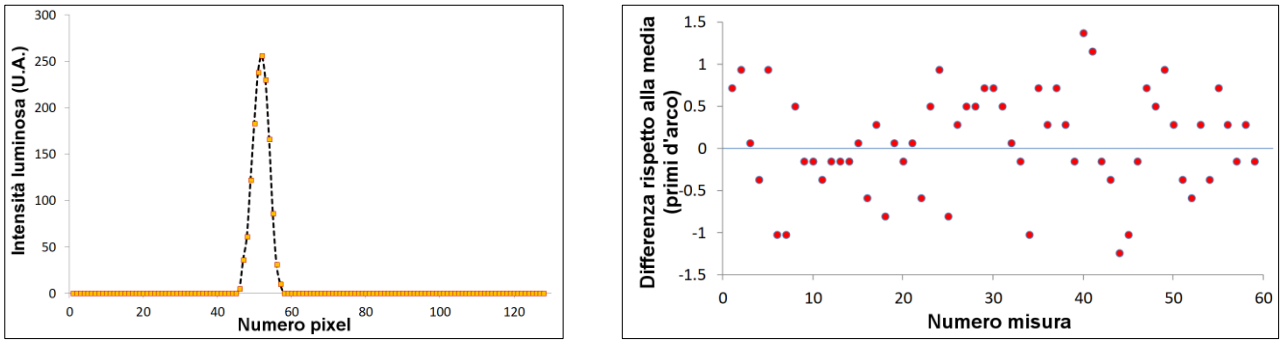


Figura 11: a sinistra, profilo della riga di luce registrato dal sensore TSL1401 e, a destra, scarto, in primi d'arco, su 60 valori di centro riga misurati con la bussola portatile mostrata in figura 10.

6.2 LF1401

Il discorso per il sensore della IC Haus [12] è sostanzialmente lo stesso di quello della AMS. I due chip, infatti, sono identici sia come dimensioni dei pixel, sia per le caratteristiche elettriche dell'elettronica. Anche questo sensore è stato testato e ha dato risultati confrontabili con quelli del TSL1401, utilizzando le stesse istruzioni.

Il chip della IC Haus, però, ha degli innegabili vantaggi rispetto a quello della AMS. Innanzitutto il sensore è disposto su una piastra dotata di piedini con dei fori, sui quali poter saldare i contatti elettrici. A differenza del gemello TSL1401, quindi, non c'è la necessità di fare saldature superficiali ed il rischio di distacco dei contatti è quasi azzerato.

Inoltre, nell'LF1401 il tempo di integrazione del segnale può essere regolato in modo più diretto, utilizzando un ingresso apposito. Quando tale ingresso è collegato a massa è attiva l'integrazione, quando è connesso all'alimentazione l'integrazione è inibita. In questo modo, il tempo di integrazione può essere regolato fino a zero, cosa che può essere utile in caso di sorgenti di luce particolarmente intense, e aumentato a piacimento senza dover incidere sul periodo del clock.

6.3 S9226

Il sensore della Hamamatsu [13] è simile ai due precedenti per dimensioni ma contiene un numero di pixel 8 volte superiore. Tra tutti i sensori presi in considerazione in questo

Rapporto è il più costoso e neppure quello con le caratteristiche più interessanti ma è un dispositivo con qualità e affidabilità pregevoli.
A differenza dei dispositivi a 128 pixel, questo viene fornito in un package che assomiglia a quello di un comune circuito integrato (come pure i sensori della Sony e della Toshiba di cui si parla in seguito). È dotato, infatti, di piedini che si possono inserire in un comune zoccolo da 8 pin, per cui non c'è la necessità di eseguire saldature.
Il numero di pixel, seppur non al livello del modello della Sony o della Toshiba, è comunque molto elevato e la loro dimensione è la più piccola, tra gli array lineari. Una piccola dimensione dei pixel favorisce l'accuratezza della misura di una linea di luce in quanto, a parità di larghezza, essa intercetta un numero maggiore di pixel e quindi diminuiscono eventuali errori dovuti a rumore di fondo o a disuniformità di risposta.
Questo chip impone un valore massimo al periodo di oscillazione del clock pari a 100 μs, il che è pari alla durata dell'operazione di digitalizzazione su una scheda con ATmega qualora si usino istruzioni ad alto livello. Inoltre, alcuni requisiti riguardanti i tempi di salita/discesa degli impulsi e del duty cycle (vedi fig. 6) sono tali per cui la scelta migliore per usare questo chip è quella di pilotarlo attraverso la scheda Arduino DUE. L'S9226, tra l'altro, può essere alimentato a 3.3 V (la tensione di lavoro della scheda DUE) e il tempo di digitalizzazione di Arduino DUE è di soli 4 μs (usando istruzioni ad alto livello), oltre al fatto che è possibile aumentare la risoluzione della conversione fino a 12 bit.
Con 1024 pixel, considerando che l'elettronica di questo chip fornisce un dato ogni 4 cicli di clock, il tempo minimo di acquisizione è poco superiore a 32 ms, senza andare a manipolare i registri del processore.
Come nel caso degli array da 128 pixel, anche con il chip della Hamamatsu i primi test hanno dato un esito confortante per cui si è deciso di progettare e realizzare la nuova bussola solare con questo sensore ed i dettagli sono illustrati nel paragrafo 8.

6.4 ILX554A

Questo chip della Sony [14] ha l'unico inconveniente di non essere più in produzione, benché, alla data del presente Rapporto, molti esemplari siano ancora nelle scorte di parecchi rivenditori internazionali.
Rispetto ai sensori appena illustrati, l'ILX554A ha il notevole vantaggio di avere un numero molto più consistente di fotodiodi (2048) una dimensione del singolo fotosensore molto

piccola (14 μm invece che 63.5 dei sensori a 128 pixel e confrontabile con i 7.8 dell'Hamamatsu) ed un'area sensibile molto più estesa (28.7 mm contro 8 mm circa) il che lo rende più adatto a misure di precisione quali potrebbero essere la lettura di codici a barre o la misura di spettri elettromagnetici.

Di contro, il gran numero di pixel non è adatto ad una velocità modesta per il clock e anche la memorizzazione di 2048 valori richiede un hardware che va oltre le possibilità delle schede Arduino più semplici. Il convertitore analogico-digitale di Arduino, infatti, lavora a 10 bit, il che implica l'uso di variabili a 2 byte (a meno di non voler perdere in risoluzione o di fare complicati algoritmi per impacchettare numeri di 10 bit in variabili da 1 byte). Quindi, memorizzare 2048 numeri richiede una RAM di 4096 byte, il doppio di quella a disposizione per le schede UNO e NANO e la metà di quella della scheda MEGA2560.

Uno stratagemma, utile almeno per testare l'accoppiata Arduino-ILX554A, è quello di memorizzare solo una parte dell'array, oppure quello di inviare i dati, così come vengono digitalizzati, ad un'interfaccia esterna come un display o un computer collegato via seriale. Quest'ultimo sistema consente anche di memorizzare i dati (lo farebbe il computer) ma ha lo svantaggio di allungare i tempi di acquisizione in quanto tra una digitalizzazione e l'altra occorre inserire l'invio del dato sulla porta seriale.

Altra possibilità è quella di usare la scheda Arduino DUE, la quale non solo è dotata di 96 kB di memoria, ma ha anche una superiore velocità di clock. Di contro, la tensione di lavoro di Arduino DUE, di 3.3 V, è tecnicamente incompatibile con quella del chip della Sony che è di 5 V. Per usare questa scheda, dunque, serve un adattatore di segnali in tensione.

Da un punto di vista elettrico, questo sensore è appena più esigente dei due visti in precedenza. Il clock, in effetti, non pone un limite superiore al periodo di oscillazione mentre, al contrario, i tempi di salita e discesa devono essere contenuti entro 100 ns ed il duty cycle al massimo può variare dal 40% al 60%.

Con queste premesse, dunque, abbiamo deciso di verificare la possibilità di pilotare questo sensore tramite una scheda Arduino UNO, usando istruzioni a basso livello per avere un clock veloce e memorizzando i 1500 pixel centrali usando un byte per pixel. In effetti, in caso di test positivo, un eventuale prototipo che sfrutti un Arduino per pilotare l'ILX554A può essere interfacciato ad una memoria esterna (una EEPROM o una scheda SD) per salvare tutti i 2048 valori in tempi ragionevolmente brevi oppure, se l'obiettivo della misura fosse quello di trovare un singolo picco, dopo aver verificato la presenza di un picco (senza memorizzare i dati) si potrebbe restringere un secondo scan a quei pixel che si trovano intorno al picco stesso.

L'algoritmo per pilotare e scaricare i dati dal sensore ILX554A non cambia granché rispetto a quello usato per il TSL1401, a parte l'uso di istruzioni a basso livello e con un prescaler del convertitore analogico-digitale portato da 128 a 32 (vedi paragrafo 4), per una frequenza di clock pari a 0.5 MHz. Il fatto di dover rispettare il duty cycle al 50%, però, impone che i due semiperiodi di clock, durante il primo dei quali viene eseguita la digitalizzazione, debbano avere una durata simile. Con le istruzioni utilizzate, il tempo di digitalizzazione era di 28 μs, per cui ogni semioscillazione del clock in cui non c'era digitalizzazione è stata allungata a 20 microsecondi tramite un ritardo software (con un duty cycle di 20/48 = 42%). Il tempo globale di acquisizione, tenendo conto di un periodo di 48 μs, di un tempo di integrazione di 20 cicli di clock, 2088 cicli per uno scan completo (40 pixel otticamente inattivi e 2048 pixel attivi), era pari a circa 100 ms.

Un problema di non poco conto che va tenuto in considerazione quando si testano sensori di questo tipo, estremamente sensibili alla luce, è la difficoltà di distinguere un malfunzionamento dell'intero sensore dalla semplice saturazione dovuta all'eccessiva intensità luminosa.

I fotodiodi di questo sensore saturano con un'intensità luminosa di 4 millilux per secondo. In un tempo di integrazione di circa mezzo millisecondo bastano $2 \cdot 10^{-6}$ lux per arrivare alla saturazione!

Una quantità esagerata di luce rispetto a questo limite, che colpisca anche solo una parte del sensore, non comporta semplicemente un segnale massimo (che, per questo sensore, corrisponde ad una tensione minima di uscita), ma un comportamento anomalo di tutti i pixel, alla stregua di un sensore rotto.

Per limitare al minimo questo problema e verificare il corretto funzionamento del chip, abbiamo schermato quasi tutto il sensore con del cartoncino, lasciando scoperti pochi millimetri, ed abbiamo inserito il sensore in una scatola di cartone chiusa completamente tranne che per un foro di circa 6 mm^2 da cui poteva filtrare la luce, non direttamente affacciato sul sensore.

In figura 12 si riportano le immagini del segnale acquisito all'oscilloscopio in queste condizioni, lo stesso segnale in cui i pixel centrali risultano debolmente saturati ed infine il segnale ottenuto quando la scatola che conteneva il sensore è stata aperta da un lato.

Con gli accorgimenti previsti per evitare la saturazione del sensore, dunque, il test di pilotaggio dell'ILX554A tramite Arduino UNO ha avuto senz'altro esito positivo.

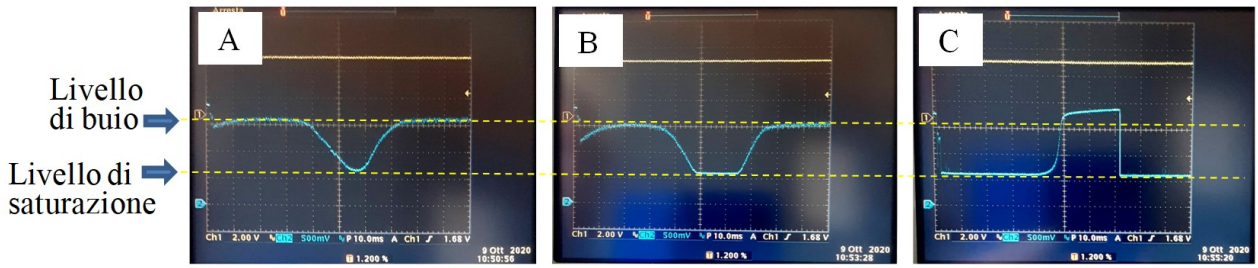


Figura 12: il segnale di luce misurato dal sensore ILX554A, pilotato da un Arduino UNO e registrato dall'oscilloscopio. A) segnale in condizioni normali (curva che scende al minimo) B) segnale in condizioni di saturazione debole C) segnale in condizioni di saturazione forte. La situazione C) potrebbe essere scambiata per un malfunzionamento del sensore, dal momento che sembrerebbe saturare in una zona non illuminata e dare segnale nullo (anzi, anche sotto il minimo) nella parte investita dalla luce.

6.5 TCD1304DG

Questo sensore della Toshiba [15] appare ancora più promettente di quello della Sony qualora si necessiti di fare misure di elevata accuratezza. I suoi 3648 pixel di 8 μm di larghezza, infatti, se utilizzati per misurare l'angolazione di un raggio solare che passa attraverso una fenditura, come avviene nel caso della bussola solare, rappresentano un sistema di misura in grado di coprire un angolo maggiore di 70° con una risoluzione di 2/100 di grado per pixel. Tale sensore, inoltre, viene tuttora prodotto dalla casa madre ed il suo costo è decisamente contenuto.

Purtroppo, questo è anche stato l'unico sensore che non è stato possibile pilotare con Arduino. Probabilmente, la richiesta di un clock molto elevato (frequenza minima 800 kHz) unita ad un requisito piuttosto impegnativo riguardante la sincronizzazione tra impulso di integrazione e impulso di start (non superiore al microsecondo) non ha consentito di ottenere alcun risultato utile. Il segnale in uscita dal sensore, infatti, non ha mai presentato un andamento proporzionale all'intensità di luce. Anche l'uso della scheda Arduino DUE non ha dato esiti positivi. L'unica soluzione percorribile, al momento, sembrerebbe quella di avere un'elettronica aggiuntiva che invii al chip il clock e sincronizzi gli altri due segnali (pin SH e ICG) e deputare Arduino alla sola digitalizzazione del segnale.

7. Arduino e un sensore a matrice di punti

L'interfacciamento di un sensore a matrice di punti è notevolmente diverso da quello di un array lineare. I sensori a matrice (CCD o CMOS) sono essenzialmente usati per fare *imaging*, cioè la riproduzione di un soggetto tramite l'uso di un obiettivo e, se possibile, garantendo una velocità di elaborazione dell'immagine tale per cui sia possibile visualizzare un filmato, quindi con non meno di 15 fotogrammi al secondo.

A differenza dei sensori visti in precedenza, qui il segnale è digitalizzato in partenza e il valore corrispondente alla luminosità che colpisce il singolo pixel viene inviato ad ogni clock tramite una connessione parallela costituita da 8 pin. Ad ogni ciclo di clock, dunque, il microprocessore deve leggere contemporaneamente 8 ingressi digitali. Sebbene il collegamento in parallelo elimini il problema della lentezza dei tempi della comunicazione seriale, una fotocamera con risoluzione VGA (640×480 pixel) utilizzata per riprendere video a colori da 15 fotogrammi al secondo deve inviare 640×480×2×15=9.216 milioni di pixel in un secondo (il sensore usa 2 byte per l'informazione di crominanza e luminanza), il che richiede, almeno teoricamente, una velocità di clock non superiore a 0.1 μs. Questo requisito è estremamente stringente e per una scheda da 16 MHz non può essere soddisfatto. Anche usando un solo byte per pixel (ottenendo un'immagine in bianco e nero) il clock sarebbe eccessivamente corto. Inoltre, i 300 mila byte che compongono la matrice necessitano di una memoria che va oltre qualsiasi scheda di Arduino cosa che impedisce la memorizzazione per una successiva elaborazione direttamente sull'hardware della scheda ma diventa indispensabile l'uso di una memoria esterna.

Al problema della velocità di clock, ad ogni modo, c'è un rimedio in quanto la CMOS presa in considerazione in questo Rapporto, la OV7670 [16], è presente sul mercato non solo nella versione semplice, quella da interfacciare direttamente con il processore, ma anche in una versione in cui, sulla stessa scheda del sensore, è presente una memoria da 384 kB ed un oscillatore preposto ad inviare il clock veloce alla CMOS. Su tale versione, il processore si interfaccia con questa memoria, il cui clock è meno esigente (si può arrivare a 1 μs), può inviare ugualmente i comandi direttamente al sensore e non necessita di conservare i dati per l'elaborazione in quanto può sfruttare questa memoria esterna per leggere più volte il fotogramma già acquisito. Lo svantaggio è un piccolo incremento di prezzo (di pochi euro) e un allungamento dei tempi, poiché c'è un passaggio in più, ma l'hardware è semplificato

e comunque per i nostri scopi l'uso di un clock al microsecondo garantisce un tempo di acquisizione (circa 300 ms) più che accettabile.

La versione in questione è una camera OV7670 con memoria AL422B (vedi figura 13).

Per ragioni di chiarezza, si è anche tentato di utilizzare una OV7670 senza l'AL422B ma il risultato non è mai stato ben riproducibile quando si è cercato di sfruttare la massima risoluzione. I problemi sono scaturiti essenzialmente da rumore elettromagnetico che interferiva sul segnale del clock (inviato da Arduino e che camminava lungo cavi schermati per circa 30 cm) per cui le immagini ottenute dalla CMOS erano spesso affette da segnali spuri.

L'uso dello stesso sensore, dotato di memoria esterna, ha invece dato risultati molto più soddisfacenti.

Rispetto ai sensori lineari appare evidente che i collegamenti tra scheda e sensore in questo caso sono molto più numerosi. Innanzitutto, come si è già detto, il segnale è inviato su 8 linee in parallelo piuttosto che su una sola seriale. Poi ci sono i due canali di comunicazione (I^2C) per l'invio dei settaggi e quindi i vari segnali di clock e reset. Dei 22 pin ne abbiamo utilizzati il minimo indispensabile, ovvero 17, quelli indicati nella tabella 3.

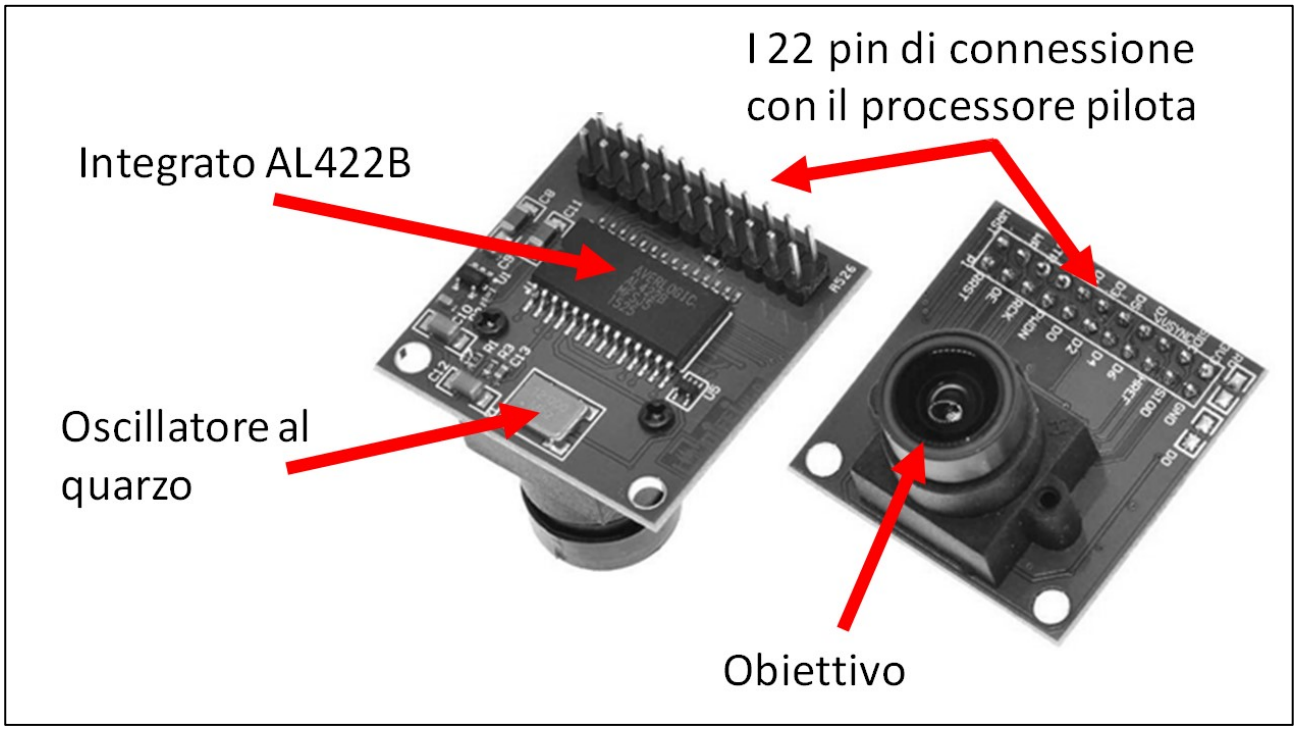


Figura 13: la fotocamera OV7670 nella versione con la memoria AL422B. Nella versione classica non è presente l'integrato sul lato posteriore e i pin di collegamento sono 18 invece di 22.

PIN	3V3	SIOC	VSYNC	D7	D5	D3	D1	RST	STR	WR	WRST
uso	Tensione alimentaz.	Clock I²C	Sincron. verticale	Bit 7	Bit 5	Bit 3	Bit 1	→ 3.3 V	Non usato	Abilita in scrittura	Reset scrittura

PIN	GND	SIOD	HREF	D6	D4	D2	D0	PWDN	RCK	OE	RRST
uso	Massa	Data I²C	Non usato	Bit 6	Bit 4	Bit 2	Bit 0	Non usato	Clock lettura byte	→GND	Reset lettura

Tabella 3: i pin della scheda con la CMOS OV7670 e la memoria AL422B con l'indicazione di quelli effettivamente usati e la loro funzione.

La scheda utilizzata è stata una Arduino DUE e le istruzioni, sia per il clock che per la lettura dei pin digitali, sono state scritte pilotando direttamente i registri del processore. In figura 14 c'è la foto della scheda con i collegamenti fisici verso la CMOS camera.

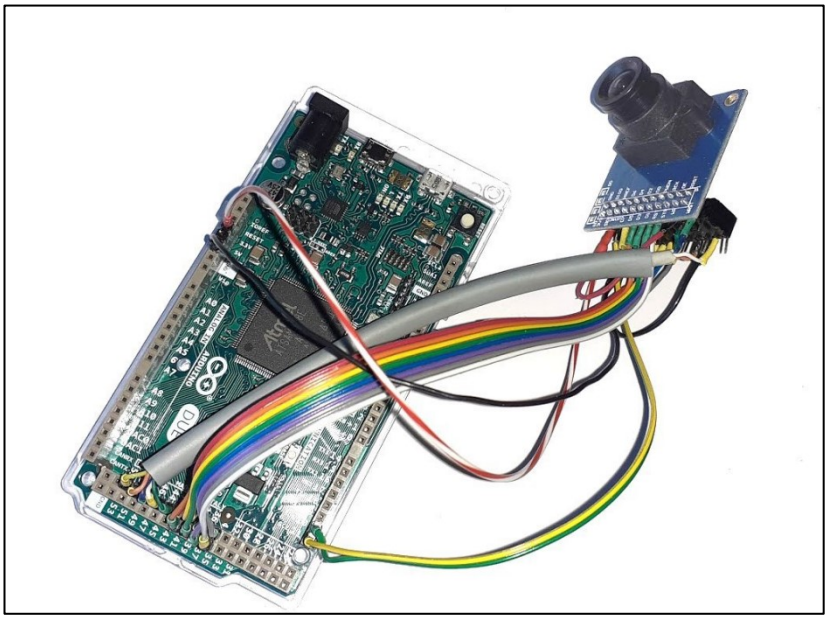


Figura 14: La scheda Arduino DUE interfacciata alla telecamera OV7670. La piattina a 8 cavi colorati porta il segnale digitalizzato dei pixel, il cavo grigio porta i cinque segnali delle sincronizzazioni e del clock, quello giallo e verde i segnali di comunicazione I²C ed infine l'alimentazione avviene tramite i cavi bianco-rosso e nero.

La sequenza di operazioni da eseguire prima di 'scattare una foto' è piuttosto complessa e riguarda il settaggio dei registri della CMOS. Tali registri sono ben 201 e buona parte di essi va modificata rispetto al valore di default affinché tutta la catena dell'acquisizione di un'immagine vada a buon fine. Fortunatamente, su internet si trovano già algoritmi in cui

sono presenti le istruzioni per impostare tali registri che si possono adattare in funzione delle necessità senza dover leggere decine di pagine di manuale. In appendice, a beneficio di chi volesse avventurarsi nell'interfacciare una scheda Arduino con la OV7670, sono riportate le istruzioni per il settaggio iniziale dei registri da modificare e che, nel nostro caso, hanno avuto esito positivo.

Questa CMOS è impostabile con diversi gradi di risoluzione, da 120x160 a 320x240 fino alla massima, 640x480. Considerando che la risoluzione è un elemento importante in qualsiasi tipo di misura e che il tempo di acquisizione, purché fosse almeno sotto 1 secondo, non era determinante, abbiamo impostato il programma in modo che la memoria acquisisse esclusivamente un'immagine a risoluzione VGA. Inoltre, poiché un'immagine in bianco e nero era più che sufficiente per i nostri scopi, abbiamo ridotto ad un byte il segnale di ogni pixel. Anche con un solo byte per pixel, comunque, una matrice di 640x480 byte eccede i limiti di memoria dell'Arduino DUE pertanto, per verificare la corretta acquisizione di un fotogramma VGA c'erano solo due possibilità: dividere il fotogramma in settori e di inviarli ad un computer che poi ricostruiva l'immagine intera a posteriori, oppure rinunciare a memorizzare i byte in Arduino, inviandoli direttamente al computer.

Inizialmente abbiamo controllato il corretto funzionamento dell'elettronica tramite l'acquisizione di una piccola porzione dell'intero fotogramma, dopodiché abbiamo optato per l'invio diretto dei byte al computer. Questa soluzione è potenzialmente a rischio in quanto l'invio di un byte tramite porta seriale comporta un forte allungamento del tempo di clock. Poiché anche l'AL422B pone dei limiti nel tempo di oscillazione del clock di lettura (un microsecondo al massimo, da datasheet) l'inserimento di un comando per l'invio dei dati su porta seriale poteva comportare dei malfunzionamenti.

La durata dell'oscillazione di un impulso di clock, in effetti, è risultata pari a 40 μs, di cui più di 39.5 a carico dell'uso della seriale. Nonostante ciò, il test è andato a buon fine ed è stato possibile ottenere l'immagine VGA di figura 15, in bianco e nero, in circa 12 secondi.

La sequenza delle istruzioni da inviare al chip, dopo aver impostato i parametri, è piuttosto semplice ed è qui riassunta in pseudo-codice:

- Attendi l'impulso di sincronizzazione verticale (inizio fotogramma)
- Invia un impulso di reset al pin WRST
- Abilita la scrittura sulla memoria alzando il pin WR
- (in questo momento la memoria AL422B acquisisce l'immagine dalla fotocamera)
- Attendi l'impulso di sincronizzazione verticale (fine fotogramma)
- Disabilita la scrittura sulla memoria abbassando il pin WR

- Abbassa il pin RRST per portare il puntatore di lettura all'inizio del fotogramma
- Attendi pochi cicli di clock e poi alza il pin RRST
- Ora, ad ogni ciclo di clock sui pin D0-D7 arriva il byte relativo alla luminosità di ogni pixel, in modo sequenziale

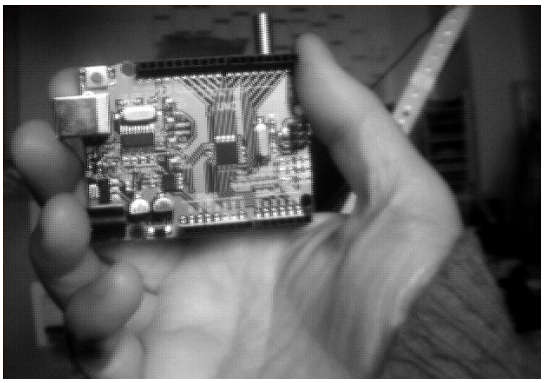


Figura 15: immagine con risoluzione 640x480 ottenuta con la telecamera OV7670 interfacciata con un Arduino DUE

Senza inserire l'invio della seriale tra un clock e l'altro, la velocità di acquisizione di un intero fotogramma, considerando i tempi di attesa della sincronizzazione verticale, è di circa 300 millisecondi. Eventuali elaborazioni matematiche sull'immagine, dunque, possono avvenire in tempi estremamente rapidi, fermo restando che è possibile memorizzare su Arduino solo una porzione dell'intero fotogramma. La presenza della memoria esterna AL422B, comunque, consente di poter acquisire un fotogramma e poi scaricare a piacere varie porzioni della stessa immagine più volte in istanti successivi.

8. Progettazione e costruzione di una bussola solare con sensore lineare

La Bussola Solare ENEA è uno strumento di misura brevettato nel 2012 [2] basato, da una parte, su un originale algoritmo semplificato per il calcolo delle effemeridi e, dall'altra, sulla misura della direzione di vista del Sole ottenuta con un innovativo dispositivo elettro-ottico che fa uso di una fenditura e di un sensore CCD.

Il principio di funzionamento di una bussola solare è basato sulla conoscenza della posizione del Sole nel cielo e quindi delle coordinate con cui esso è traguardato da un particolare punto di osservazione sulla Terra. Tali coordinate sono costituite dall'elevazione

del Sole sopra l'orizzonte e dal suo azimut, ovvero l'angolo che il piano verticale contenente il Sole forma con il piano meridiano (quello che contiene i due Poli) passante per il punto di osservazione. Misurata la direzione di vista del Sole rispetto ad un piano verticale in cui si trovi un obiettivo da traguardare (o più semplicemente un piano verticale di riferimento) di conseguenza si conosce l'angolo che la direzione di vista di quell'obiettivo fa rispetto al Nord geografico. La misura della direzione del Sole viene eseguita calcolando il baricentro della linea di luce proiettata sul sensore (a matrice o lineare) rispetto alla colonna o al pixel di riferimento.

Questo strumento si è rivelato uno dei metodi più precisi per la determinazione del Nord geografico (con un'accuratezza entro 1/100 di grado RMS) ma l'esemplare che è stato realizzato non è replicabile in quanto alcuni componenti non sono più reperibili. Poiché per la misura della linea di vista del Sole può essere sufficiente un sensore lineare, dopo i primi test eseguiti sul chip della Hamamatsu si è deciso di progettare e realizzare una nuova bussola basata su questo sensore e su una scheda Arduino.

L'elettronica a corredo della bussola non è solo costituita da questi due elementi ma anche da un ricevitore GPS, un display alfanumerico, un lettore di schede SD e da un modulo bluetooth. Il tutto (tranne il GPS) inserito in una console in cui trovano posto, internamente, anche un pacco batterie e dei moduli di regolazione della tensione ed esternamente dei pulsanti, dei led e diversi connettori. Il sensore lineare è collocato, come nella bussola originale, in una testa di alluminio che va collocata sopra un sistema di puntamento quale può essere un teodolite.

Sulla parete affacciata verso il sole, inclinata a 45°, trova posto un vetrino in cui è ricavata una fenditura alta 4 cm e larga circa 100 μm, realizzata dall'Istituto di Fotonica e Nanotecnologie del CNR di Roma depositando un sottile strato di metallo sul vetro e rimuovendone una parte tramite tecnica litografica.

Testa della bussola ed elettronica sono connesse attraverso un comune cavo ethernet a 8 poli.

Una volta verificato il funzionamento di tutte le interfacce connesse alla scheda Arduino, il problema più arduo da risolvere è stato quello di tarare il dispositivo elettro-ottico. In particolare occorreva conoscere con estrema accuratezza la distanza tra sensore e fenditura (D), il numero del pixel che rappresenta la proiezione della fenditura sul sensore (XR0), i due angoli di rotazione del sensore rispetto al piano contenente la fenditura (α) e alla direzione della fenditura stessa (β) ed infine lo scostamento (ϕ), nel piano orizzontale,

tra la direzione del piano di riferimento rispetto a quello visualizzato dal sistema di puntamento del sostegno che sorregge la testa della bussola (vedi figura 16).
La taratura è avvenuta nel seguente modo: la bussola è stata collocata sopra ad un teodolite dotato di livella a bolla e di goniometro ad alta precisione. Quindi, di fronte alla bussola sono stati posti dei led lungo una linea perfettamente verticale onde simulare il sole a diverse altezze. Per ognuno di questi led è stato misurato il corrispondente baricentro delle linea di luce formata sul sensore e ciò per diverse posizioni del goniometro.
Graficando i valori del centro riga in funzione dell'angolo del goniometro, il risultato che si ottiene è una serie di punti che si mettono lungo diverse rette, ognuna corrispondente ad una diversa altezza della sorgente luminosa.

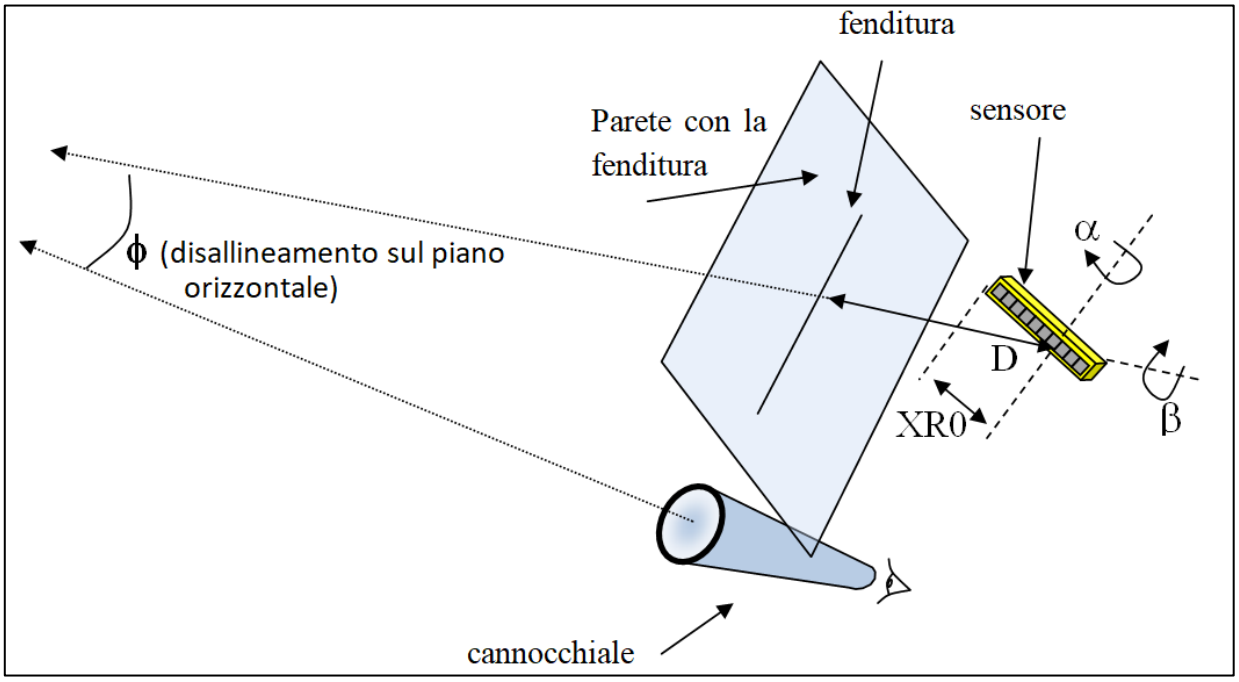


Figura 16: rappresentazione delle variabili e dei parametri da tenere in conto per la taratura della bussola

L'inclinazione e l'intercetta di tali rette dipendono dai parametri di costruzione della bussola, per cui agendo su tali parametri si può ottimizzare la corrispondenza tra la curva teorica e i dati sperimentali. Supponendo che il sensore sia stato montato in modo perfettamente parallelo alla parete con la fenditura e ortogonale alla fenditura stessa (cioè con $\alpha = 0$ e $\beta = 90°$), l'equazione che determina il punto x_a colpito da un raggio di luce proveniente da una sorgente posta ad elevazione φ e ad un azimut θ è la seguente:

$$x_a = D\sqrt{2}\frac{\sin(\vartheta)}{\cos(\vartheta)+\tan(\vartheta)} + XR0 \qquad (1)$$

La distanza x_a (misurata rispetto al bordo dell'array, cioè rispetto al primo pixel) vale esattamente XR0 quando $\theta = 0$, cioè quando la luce è ortogonale al sensore.

Sperimentalmente si ricavano i valori di φ, θ e x_a mentre vanno determinati D e XR0 affinché la differenza tra i due membri dell'equazione (1) risulti minimizzata.

In figura 17 si può apprezzare graficamente la sensibilità ai parametri osservando la differenza tra i punti sperimentali e la retta che si ottiene attraverso l'equazione (1) per un particolare valore di elevazione φ e per diversi valori dei parametri D e XR0.

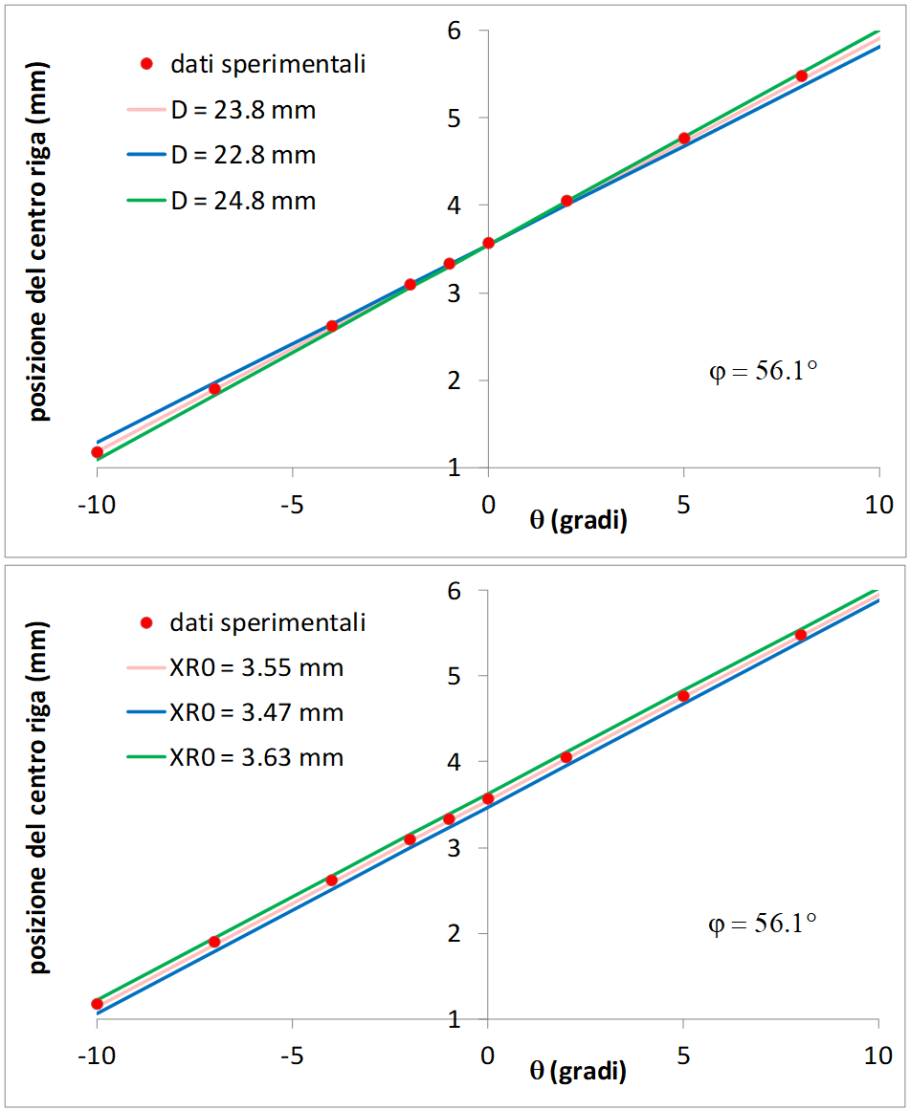


Figura 17: confronto tra i punti di centro riga misurati sull'array, ponendo una sorgente di luce a 56.1° di elevazione e variandone l'azimut, con i corrispondenti valori teorici calcolati per diverse distanze D (figura in alto) e XR0 (figura in basso). Si può vedere come piccolissime differenze nei parametri portino facilmente ad una discordanza con i dati sperimentali.

Una volta determinati i parametri di calibrazione, o almeno un intervallo di valori entro cui ragionevolmente sono contenuti i valori corretti, il secondo passo è quello di verificare il comportamento della bussola una volta esposta al sole. La correttezza dei parametri si manifesta osservando i valori di azimut forniti a bussola ferma per un certo periodo di tempo. Tale azimut, infatti, dovrebbe risultare costante in quanto, mentre il Sole si muove del suo moto apparente, la bussola punta sempre nella stessa direzione. Un eventuale andamento diverso da una serie di valori oscillanti intorno ad un valor medio, ad esempio dei valori che tendono a crescere, a diminuire o a formare una parabola, va confrontato con il comportamento simulato della bussola, in cui sia possibile variare i parametri di costruzione per aggiustare i parametri stessi.

A tal fine, abbiamo eseguito diverse scansioni con la bussola posta in orizzontale e illuminata dal sole ed abbiamo registrato una serie di valori di ora e centro riga. Quindi, abbiamo realizzato un programma in C++, tramite Microsoft Visual Studio Community 2019, in cui, inserendo i dati geografici (data, ora, latitudine, longitudine) viene calcolato l'azimut del Sole e il valore dell'azimut puntato da una bussola in cui sia stata registrata quella serie di valori di centro riga, utilizzando i parametri costruttivi inizialmente determinati tramite la procedura illustrata in precedenza.

In questo modo, innanzitutto si verifica la correttezza di tali parametri in quanto la sequenza dei valori di azimut teorici si dovrebbe distribuire come i dati sperimentali e, in secondo luogo, si può raffinare la stima dei parametri rendendo costante tale distribuzione (a sensore fermo, come già detto, l'azimut della bussola deve rimanere fisso). Il programma, tra l'altro, è in grado di ottimizzare i parametri in automatico, sebbene sia possibile cambiarli in modo manuale. In figura 18 è mostrato l'azimut misurato dalla bussola con i parametri non ottimizzati, accanto al calcolo dell'azimut risultato dalla simulazione con questo programma e, a destra, il risultato dopo l'ottimizzazione.

Una volta ottimizzati D e XR0, rimane da inserire nei calcoli lo scostamento tra il piano di riferimento della bussola e la direzione traguardata dal sistema di puntamento su cui la bussola è stata collocata. Per determinare quest'ultimo parametro è indispensabile conoscere l'azimut vero di una qualsiasi direzione. Sapendo qual è l'azimut tra i punti A (osservatore) e B (traguardo), infatti, è sufficiente puntare il punto B da A, misurare l'azimut fornito dalla bussola e calcolare la differenza tra tali valori (a tal fine, a meno che non si possieda già una bussola accurata, per conoscere l'azimut tra A e B occorre sapere le coordinate geografiche dei due punti ed utilizzare un metodo che tiene conto della curvatura della Terra, vedi [17]). Ovviamente il sistema puntatore-bussola dovrà essere solidale o, se

la bussola fosse removibile, quest'ultima dovrà essere sempre riposizionata nello stesso modo, senza introdurre rotazioni o traslazioni.

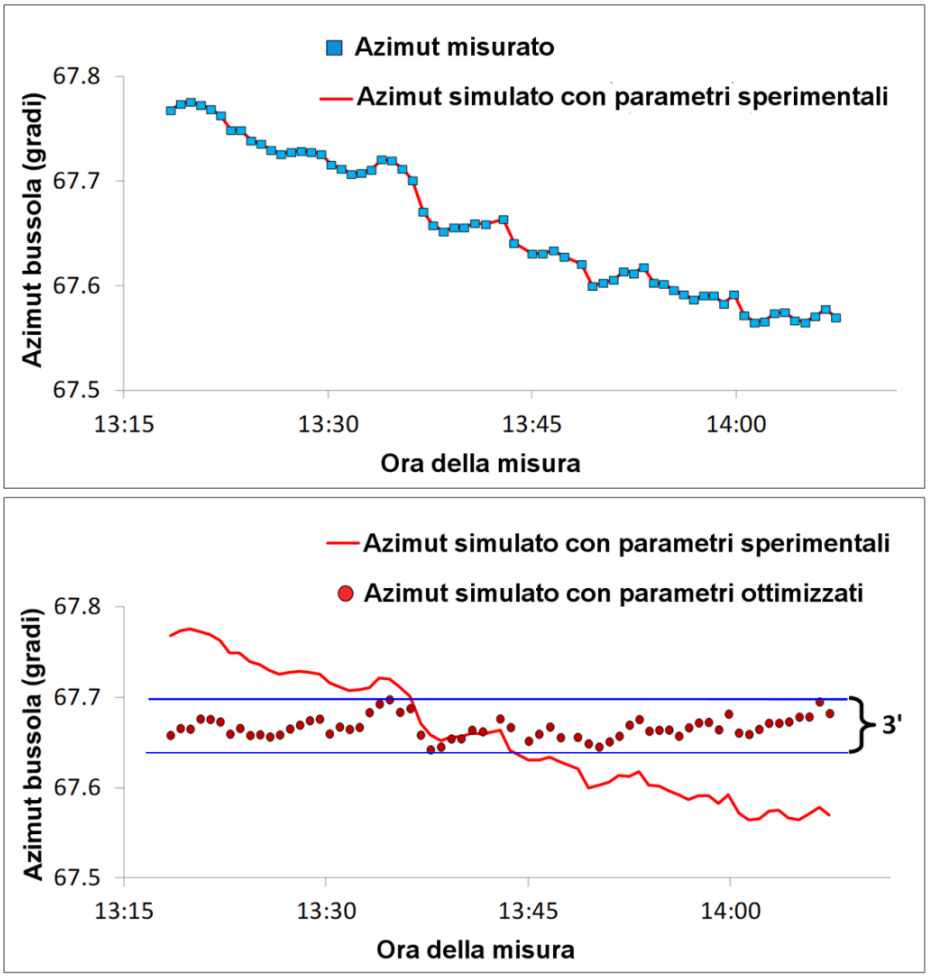


Figura 18: in alto, i punti relativi alla misura sperimentale dell'azimut a bussola ferma (quadrati) e gli stessi valori calcolati dal programma (curva continua) in cui sono stati inseriti i parametri di costruzione della bussola e i valori di centro riga risultati dalla misura. In basso, la stessa curva simulata con i parametri sperimentali e i nuovi punti calcolati usando i parametri ottimizzati. Con l'ottimizzazione, variando il solo parametro D da 22.43 mm a 22.83 mm i punti si dispongono intorno a un valore costante entro ±1.5 primi d'arco.

Conclusa quest'ultima fase di taratura, la bussola è pronta per essere usata come strumento di misura. Nel caso in questione, alcuni giorni dopo aver completato tutta la procedura di taratura, abbiamo voluto verificare se la bussola continuava a fornire i corretti valori di azimut. Abbiamo, pertanto, collocato la bussola in un punto di osservazione nel Centro ENEA di Frascati da cui conosciamo i valori di azimut rispetto ad alcuni punti di riferimento ed abbiamo eseguito delle misure, i cui risultati sono riportati in tabella 4.

	Cupola di San Pietro	Antenna RAI su Monte Mario	Campanile di Santa Maria Maggiore
Numero di misure	5	5	3
Media	116.6956°	120.4162°	120.894°
Scarto massimo tra le misure	0.46’	0.43’	0.19’
Deviazione standard	0.37’	0.29’	0.17’
Valore aspettato	116.6891°	120.4103°	120.8946°
Differenza	0.39’	0.35’	-0.04’

Tabella 4: valori di azimut misurati con la bussola composta dalla scheda Arduino DUE e il sensore S9226 dell’Hamamatsu, alcuni giorni dopo aver completato il processo di taratura. Dai dati presenti è possibile apprezzare sia la precisione che l’accuratezza di questo strumento di misura. Si noti che le differenze tra teoria ed esperimento sono espresse in primi d’arco.

Da questi dati si può dedurre che un sensore lineare può sostituire egregiamente un sensore a matrice di punti e che la coppia Arduino-Hamamatsu è un’ottima soluzione per realizzare l’elettronica di una bussola solare (e non solo).

9. Conclusioni

La versatilità e l’affidabilità delle schede Arduino sono ben note e la possibilità di reperire sul mercato una notevole quantità di sensori a poco prezzo e su internet i codici per il loro interfacciamento le rende un dispositivo utilizzabile anche in un laboratorio di ricerca.
In questo Rapporto abbiamo illustrato dei test di sensori di luce, sia lineari che a matrice, connessi a delle schede Arduino per verificarne la compatibilità e la possibilità di usare tali sistemi come strumenti di misura.
Abbiamo preso in considerazione ben cinque sensori lineari, fabbricati dalla AMS, IC Haus, Hamamatsu, Sony e Toshiba, costituiti da un minimo di 128 ad un massimo di 3648 pixel ed una fotocamera CMOS della Omnivision da 640x480 pixel. In tutti i test, tranne che per il sensore della Toshiba, il test è stato positivo ed abbiamo potuto verificare come l’abbinamento Arduino/Fotosensore possa essere un valido congegno per misure che coinvolgono la luce, come potrebbe essere uno spettrometro o un lettore di codice a barre o anche per sistemi di imaging.

In particolare, con una scheda Arduino DUE e il chip della Hamamatsu è stata progettata e realizzata una bussola solare, ovvero uno strumento che misura la direzione del Nord geografico sfruttando le equazioni del moto terrestre, con dei risultati che sono alla pari con quelli ottenuti con la bussola solare ENEA brevettata alcuni anni fa e che sono confrontabili con i migliori apparecchi in commercio, adatti per tali misure, i cui costi sono almeno di due ordini di grandezza più elevati.

10. Appendice: impostazioni dei registri della CMOS camera OV7670 per l'interfacciamento con Arduino

Le variabili di sinistra hanno lo stesso nome dei registri così come indicati nella tabella 8.2 del documento "OV7670/OV7671 CMOS VGA (640x480) CameraChip Implementation Guide" (versione 1.0, 2 settembre 2005), reperibile su internet all'indirizzo:
https://usermanual.wiki/Pdf/OV767020Implementation20Guide20V10.1347697000/view
Quando i registri sono impostati secondo questi valori, ogni pixel registra un solo byte relativo alla luminosità e l'acquisizione di un fotogramma da parte della memoria AL422 avviene in circa 300 millisecondi. Il tempo di scarico del fotogramma da parte di Arduino dipende, invece, dalla velocità del clock inviato sul pin RCK della scheda con la fotocamera.

COM7 = 0x80 //serve per resettare tutti i parametri prima di reimpostarli
CLKRC = 0x01
COM7 =0x05
COM3 = 0x00
COM14 = 0x00
SCALING_XSC = 0x3A
SCALING_YSC = 0x35
SCALING_DCWCTR = 0x11
SCALING_PCLK_DIV = 0xF0
SCALING_PCLK_DELAY = 0x02
TSLB = 0x01
COM17 = 0x00
AEW = 0x95
AEB = 0x33
HAECCn = 0x78, 0x68, 0xD8, 0xD8, 0xF0, 0x90, 0x94 (con n=1,2,3,4,5,6,7)
SATCTR = 0x60
ADCCTR1 = 0x02
ADCCTR2 = 0x91
ADC_add = 0x37
ACOM = 0x71
OFON = 0x2A
ABLC1 = 0x0C
THL_ST = 0x82

11. Bibliografia

[1] Il sito ufficiale della piattaforma Arduino è http://www.arduino.cc

Su Arduino sono stati scritti molti articoli su riviste scientifiche, ad esempio:

A. Kurniawan, F. T. Dwi Atmaji, J. Alhilman: “Design of remote temperature monitoring system on automatic filling R125 Shinva machine using LM35 sensor and Arduino Uno micro controller”, Int. J. of Integrated Engeneering, vol. 12 n. 7 (2020), pp. 280-290

K. Krishnamurthi, S. Thapa, L. Kothari, A. Prakash: “Arduino Based Weather Monitoring System”, Int. J. of Engineering Research and General Science, Vol. 3, Issue 2, (2015) ISSN 2091-2730

T. W. Schubert, A. D’Ausilio, R. Canto: “Using Arduino microcontroller boards to measure response latencies”, Behav. Res. 45, 1332–1346 (2013). https://doi.org/10.3758/s13428-013-0336-z

E ci sono decine di libri reperibili sul mercato, come ad esempio:

J. Blum: “Exploring Arduino: Tools and Techniques for Engineering Wizardry” (ed. John Wiley & Sons, 2019)

M. Banzi, M. Shiloh: “Arduino: La guida ufficiale” (ed. Tecniche Nuove, 3ª edizione, 2015)

[2] F. Flora, S. Bollanti, D. De Meis, P. Di Lazzaro, A. Fastelli, G. P. Gallerano, L. Mezi, D. Murra, A. Torre, and D. Vicca: “High precision electronic solar compass,” PCT patent WO 2014102841 A1 (2013)

[3] S. Bollanti, D. De Meis, P. Di Lazzaro, F. Flora, G.P. Gallerano, L. Mezi, D. Murra, A. Torre, D. Vicca: ‘Electrooptical sun compass with a very high degree of accuracy’. Opt. Lett. 40 (2015), pp. 3619–3622

[4] S. Bollanti, D. De Meis, P. Di Lazzaro, F. Flora, G.P. Gallerano, L. Mezi, D. Murra, A. Torre, D. Vicca:‘Performance of an electro-optical solar compass in partially obscured Sun conditions’. Appl. Opt. Vol. 55 n. 12 (2016), pp. 3126-3130

[5] D. Murra, S. Bollanti, F. Andreoli, L. Cafarella, D. De Meis, P. Di Lazzaro, D. Di Mauro, F. Flora, G.P. Gallerano, L. Mezi, L. Murra, D. Vicca, A. Zirizzotti: “A Solar Compass for enhancing the Efficiency of Concentrating Solar Power Plants” (Advanced Photonics Congress OSA, Montreal 2020, on-line conference)

[6] Le schede Raspberry sono dei microcomputer, normalmente utilizzati con il sistema operativo Linux, vedi il sito ufficiale: https://www.raspberrypi.org

[7] P. Di Lazzaro, D. Murra, S. Bollanti, F. Flora, L. Mezi, B. Alpat, T. Kaplanoglu: “Test di substrati per telescopi spaziali: irraggiamenti nell’ultravioletto di film poliammide in aria, vuoto e atmosfera controllata”, Rapporto Tecnico RT/2017/9/ENEA

[8] B. Alpat, M.A. Gulgun, G. Çorapcıoglu, M.M. Yildizhan, P. Di Lazzaro, D. Murra, T. Kaplanoglu, V. Postolache, S. Mengali, M. Simeoni and A. Urbani: “Testing of substrates for flexible optical solar reflectors: irradiations of nano-hybrid coatings of polyimide films with 20 keV electrons and with 200–400 nm ultraviolet radiation”, Journal of Instrumentation, Volume 14, June 2019, https://doi.org/10.1088/1748-0221/14/06/T06003

[9] Si tratta del progetto ISOFIBRA, finanziato nell’ambito dei POR FESR Lazio 2014-2020 (https://www.regione.lazio.it/binary/rl_attivitaproduttive_rifiuti/tbl_news/008bis_I_progetti_vincitori_dei_primi_4_bandi_per_la_reindustrializzazione.pdf, pag. 78)

[10] L’impostazione del *prescaler* avviene inserendo un valore composto da 3 bit nel registro ADCSRA. A seconda del valore di questi 3 bit si ha un diverso fattore secondo la seguente tabella (per semplicità si riportano solo alcune combinazioni):

registro	valori				
ADCSRA, ADPS0	1	1	0	0	1
ADCSRA, ADPS1	1	0	0	1	0
ADCSRA, ADPS2	1	1	1	0	0
Frequenza clock (MHz)	0.125	0.500	1	2	8

Per impostare i 3 bit si utilizzano le istruzioni ‘*cbi*’ per indicare uno ‘0’ e ‘*sbi*’ per indicare un ‘1’. Ad esempio, per impostare un clock di 1 MHz (sequenza 1-0-0) le istruzioni da dare sono:

```
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
sbi(ADCSRA, ADPS2);//1
cbi(ADCSRA, ADPS1);//0
cbi(ADCSRA, ADPS0);//0
```

[11] https://ams.com/tsl1401cl

[12] https://www.ichaus.de/product/iC-LF1401

[13] https://www.hamamatsu.com/jp/en/product/type/S9226-03/index.html

[14] Il sensore della Sony non è più in produzione, per cui non si trovano informazioni direttamente sul sito della casa madre. Il datasheet dell'ILX554A è reperibile su molti siti, ad esempio https://www.alldatasheet.com/datasheet-pdf/pdf/47514/SONY/ILX554A.html, mentre una versione simile a questo sensore viene prodotta dalla ditta statunitense Maxwell-Hiqe vedi http://maxwell-hiqe.com/imagica.

[15] https://toshiba.semicon-storage.com/ap-en/semiconductor/product/linear-image-sensors/detail.TCD1304DG.html

[16] Il modello OV7670 non è più presente tra i prodotti della Omnivision ma vi sono modelli equivalenti come l'OV7725 (https://www.ovt.com/sensors/OV7725). La OV7670 è comunque facilmente reperibile sul mercato, sia nella versione classica che in quella dotata di memoria interna AL422B.
Il datasheet si può scaricare qui: https://datasheetspdf.com/datasheet/OV7670.html
Il datasheet della memoria Al422B si trova qui:
https://pdf1.alldatasheet.com/datasheet-df/view/1169327/AVERLOGIC/AL422B.html

[17] S. Bollanti, D. De Meis, P. Di Lazzaro, A. Fastelli, F. Flora, G.P. Gallerano, L. Mezi, D. Murra, A. Torre, D. Vicca: "Calcolo analitico della posizione del Sole per l'allineamento di impianti solari ed altre applicazioni", Rapporto Tecnico ENEA RT/2012/24/ENEA (2012).

ENEA
Servizio Promozione e Comunicazione
www.enea.it

Stampa: Laboratorio Tecnografico ENEA - C.R. Frascati
maggio 2021