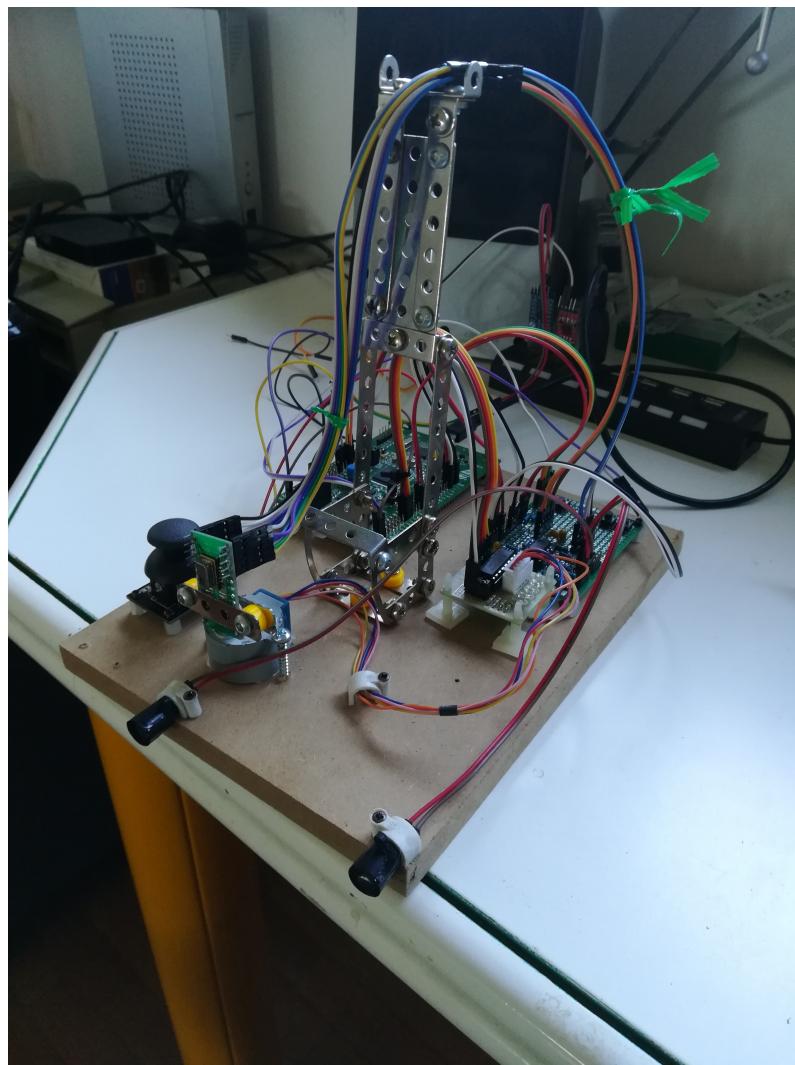


Night Surveillance System: prototipo di camera termica con localizzazione sonora.

Giacomo Turati

May 2022



1 Introduzione

Il report documenta la progettazione e realizzazione di un prototipo di dispositivo di sorveglianza notturna composto da:

- Camera termica. Griglia di sensori IR dotata di lente, con la quale acquisire immagini in notturna o mappe di calore dell'ambiente circostante.
- Motore stepper. Consente l'orientamento della camera a 180° sull'azimuth.
- Coppia di microfoni a elettrete. Posti frontalmente e a distanza reciproca nota, consentono l'acquisizione dell'audio ambientale in stereo tramite conversione ADC.
- Scheda di sviluppo STMicroelectronics STM32F407G-DISC1 (CortexM4).

Il dispositivo ha due modalità di funzionamento:

- *Manuale.* Gestisce simultaneamente streaming real-time dell'immagine termica su porta seriale, acquisizione e playback audio ambientale su uscita analogica in stereo e controllo dell'orientamento manuale della camera tramite joystick analogico.
- *Follow.* Oltre a provvedere a streaming audio e video, il dispositivo reagisce ad un eventi sonori sopra soglia RMS, orientando la camera verso un target termico, possibile fonte di rumore.

Per visualizzare lo streaming d'immagini e l'output testuale di debug si può utilizzare un terminale seriale esterno, mentre per il monitorare il playback audio si può connettere un oscilloscopio o un amplificatore esterno ai canali di uscita del DAC.

Come assunzioni preliminari, si suppone che il target sia identificabile solo sui 180° frontali e che non sia attiva più di una sorgente sonora impulsiva alla volta, dove per impulsiva s'intende di breve durata temporale, come un battito di mani o uno scoppio. L'intero software è stato sviluppato in programmazione bare metal utilizzando l'API Hardware Abstraction Layer (HAL) e compilando e configurando il codice con il supporto dell'IDE STM32CubeIDE alla versione 1.9.0.

1.1 Applicazioni, elementi d'interesse e criticità

Il prototipo parte dall'idea di creare un dispositivo di sorveglianza notturna reattivo agli stimoli sonori, utile per monitorare infrazioni in un luogo sostanzialmente silenzioso e privo di fonti di calore, con la possibilità di integrare dati audio e termici in modo da coprire in maniera più fine il perimetro di sorveglianza.

I principali elementi d'interesse e tematiche affrontate con questo progetto sono

innanzitutto la gestione in simultanea di diversi sensori ed attuatori in real-time; infatti, mantenere costante il flusso dati in ingresso e uscita entro le deadline imposte dalla conversione audio, dal frame rate di acquisizione della camera termica e dal movimento fluido del motore, implicano l'impossibilità di gestire tutte le periferiche in polling e richiedono il corretto utilizzo di interrupt e trasferimenti DMA. Inoltre, si è voluta approfondire la tematica di progettazione di driver per periferiche basati sull'API HAL, supportando le varie modalità di configurazione e utilizzo dei dispositivi.

Infine, va detto che, nonostante nell'idea iniziale ci fosse l'intenzione di stimare l'angolo d'incidenza della sorgente sonora tramite cross-correlazione, fattori come il poco controllo sulle riflessioni ambientali e imperfezioni del preamplificatore microfonico costruito ad-hoc, hanno reso difficile avere risultati attendibili con una implementazione naïve dell'algoritmo. Si è perciò ritenuto opportuno implementare una variante più semplice, in cui la camera esegue una pan su tutto il campo visivo al rilevamento dell'evento sonoro sopra soglia e si ferma nel momento in cui identifica un target di temperatura superiore ad una soglia di temperatura impostata. Tale metodo, seppur meno preciso, consente comunque di mostrare la logica del processo di automazione in sé e l'utilizzo delle periferiche, eliminando la necessità di implementare una complessa elaborazione dei segnali che si lascia come possibile sviluppo futuro.

2 Hardware

In questa sezione verranno descritte le principali caratteristiche dei componenti hardware utilizzati e la circuiteria di condizionamento del segnale.

2.1 Camera termica

La camera termica utilizzata è il modello AMG8833 della Panasonic. Il dispositivo va alimentato a 3.3V e codifica su 12 bit con segno temperature nel range tra -55° e 125° , ha una risoluzione di 64 px e acquisisce 10 fps o 1 a seconda della modalità configurata. Nonostante risoluzione e frame rate siano troppo bassi per avere uno stream di alta fedeltà, il sensore è stato scelto ai fini della prototipazione, come surrogato di costo nettamente inferiore di una camera termica di alta qualità. Comunica tramite interfaccia I2C in slave mode ed è configurabile, connettendo il pin AD_SELECT alternativamente a massa o a tensione di riferimento, su uno dei due indirizzi a 7 bit 0x68 o 0x69. Il protocollo di comunicazione seriale consente al master di configurare il dispositivo ed acquisire l'immagine termica tramite scrittura/lettura dei registri interni di quest'ultimo connettendo con resistenze di pull-up i pin SDA (traffico dati) e SCL (segnaletica di clock).

Inoltre la camera termica può essere configurata per generare un interrupt sul pin INT nel caso in cui uno qualsiasi dei pixel misuri una temperatura sopra una soglia impostata, assoluta o d'isteresi, al fine di segnalare la presenza di un target termico nel campo visivo. Collegando il pin a una linea di external

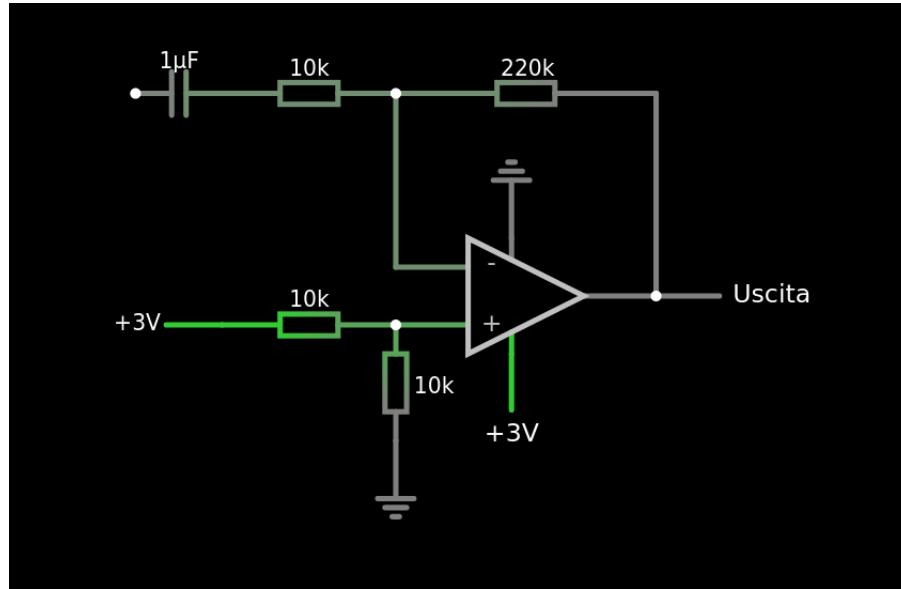


Figure 1: Circuito di preamplificazione invertente basato su chip TLE2022

interrupt, è possibile gestire l'evento di identificazione del target in maniera asincrona.

2.2 Microfoni a elettrete e preamplificazione

I microfoni a elettrete sono simili a microfoni a condensatore, quindi trasducono le onde di pressione in tensione variabile grazie al movimento della piastra mobile più esterna di un condensatore. Quest'ultima, avvicinandosi e allontanandosi dalla piastra fissa, provoca una variazione di capacità consentendo di misurare in uscita un segnale audio. A differenza dei microfoni a condensatore, la piastra mobile dei sensori in questione è costituita da una membrana di elettrete che non necessita di polarizzazione esterna, ma, nonostante ciò, i microfoni vanno alimentato a causa del transistor FET interno, utilizzato per adattare la loro alta impedenza d'uscita. La risposta in frequenza può considerarsi non nulla tra i 20Hz e 20kHz e copre quindi l'intero range di udibilità dell'orecchio umano.

I microfoni a elettrete sono per natura molto sensibili, tuttavia, sviluppando una tensione molto bassa, impongono l'utilizzo di circuito di preamplificazione che condizioni il segnale in modo da coprire tutto il range di conversione del ADC. Infatti, analizzando il comportamento del microfono con l'oscilloscopio, è possibile constatare come senza preamplificazione il microfono in saturazione produca in uscita un segnale di circa 200mV di ampiezza picco-picco, che oscilla attorno a 0V, mentre l'ADC incorporato nella scheda MCU converte su valori a 12 bit tensioni in un range da 0 a Vref (3V c.a. - 2.95V misurati sul contatto Vdd). Si è deciso quindi di progettare un circuito di preamplificazione che avesse

le seguenti caratteristiche:

- alimentazione a 3V presa dal contatto Vdd della scheda.
- punto di lavoro a metà del range di conversione del ADC (1.5V circa). Infatti, per una conversione corretta, è necessario che il segnale audio oscilli attorno a tale valore fissato come 0.
- saturazione a 3V. Per assicurarsi che le tensioni convertite coprano tutto il range di conversione, si può tarare il circuito di preamplificazione in maniera tale da avere un segnale di ampiezza 3V quando il trasduttore è in saturazione(ad esempio soffiandoci dentro da brevissima distanza).

Tra tutte le possibili opzioni di schemi circuitali note per la preamplificazione di un segnale audio, ha prevalso per semplicità e disponibilità di componenti il preamplificatore ad op-amp in configurazione invertente. In tale configurazione, denominate R1 la resistenza in ingresso al terminale invertente e R2 la resistenza di retroazione, si ottiene in uscita lo stesso segnale in ingresso amplificato in ampiezza di un fattore $-R2/R1$, fatto che rende il circuito adatto nel contesto del condizionamento di segnale audio in cui l'inversione di polarità non è considerata un problema. Inoltre, è necessario inserire un condensatore C in ingresso, di capacità $1\mu F$ nell'implementazione, in modo da rimuovere dal segnale la componente continua e lasciare solo la componente alternata, in cui è contenuta l'informazione sonora.

Una volta scelta la configurazione, è stato necessario dimensionare le componenti e apportare opportune modifiche per ottenere una preamplificazione con le caratteristiche desiderate. In primo luogo, si è dovuto utilizzare un op-amp *rail-to-rail*. Questo particolare tipo di amplificatore operazionale ha la caratteristica principale di avere un'uscita che copre l'intero range di tensione di alimentazione (da qui il nome rail-to-rail); l'uscita di un op-amp normale può invece coprire al massimo una frazione di questo intervallo e risulta pertanto incapace di soddisfare i requisiti di progettazione. In secondo luogo, per mantenere il punto di lavoro a metà del range di conversione, è stato inserito un partitore di tensione con resistenze uguali sul terminale non-invertente, al cui ingresso c'è la stessa tensione di alimentazione del chip; pertanto, il partitore fissa la massa virtuale della configurazione invertente a 1.5V circa. Infine, per avere un fattore di amplificazione tale da coprire il range di 3V in saturazione, si è scelto di utilizzare R1 da $10k\Omega$ e R2 di $220k\Omega$, per avere un segnale in uscita di circa 20 volte superiore in ampiezza rispetto a quello in ingresso.

2.3 Motore stepper e joystick

Per il motore stepper è stato utilizzato il modello unipolare 28BYJ-48. Il movimento del motore passo-passo può essere comandato da MCU tramite un chip driver, in questo caso ULN2003, che va alimentato a 5V e che permette l'attivazione delle bobine interne al motore in sequenza, settando le uscite digitali della scheda, a cui il driver è connesso, seguendo un certo pattern di

stepping. Il modello 28BYJ-48 è molto economico e costituisce una scelta classica nei progetti DIY basati su microcontrollore; il rotore principale ha una risoluzione angolare di 11.25° e perciò compie un giro completo in 32 step. Tuttavia, grazie a un *gear ratio* (rapporto tra il numero di denti dell'ingranaggio di input e quello di output) di 1/63.68, la risoluzione totale è di 2038 step per rivoluzione.

Il joystick analogico è invece un componente simile ad un potenziometro, che varia la resistenza ai suoi capi in funzione della sua posizione sui due assi. Si interfaccia con due terminali di alimentazione (GND e V+) e due uscite analogiche in tensione, una per l'asse x e una per l'asse y. Per leggere la posizione del joystick in un dato istante di tempo è sufficiente connetterlo ad un ingresso ADC e leggere il valore di tensione ai capi.

3 Software

Di seguito verrà presentata l'architettura software del progetto. A più alto livello, consiste di un loop principale infinito che gestisce il flusso dati acquisiti in Input e Output, la loro elaborazione e il comando coerente dell'attuatore. Lo fa verificando lo status dei flag settati dalle ISR che gestiscono eventi asincroni, come lo scadere di timer o il termine di trasferimenti DMA ed eseguendo in funzione di essi una determinata sequenza di operazioni.

Verranno di seguito illustrate:

- Le interfacce di collegamento utilizzate e il setup delle periferiche.
- Le librerie driver scritte per astrarre l'utilizzo delle componenti.
- Il loop principale in termini di sequenza di esecuzione.

3.1 Setup interfaccia e periferiche

Di seguito si elencano le principali configurazioni dell'interfaccia tra il processore e le periferiche esterne, indicandone l'utilizzo, la configurazione dei pin di collegamento e le modalità di trasferimento dati/comandi.

3.1.1 I2C1

- Utilizzo: interfaccia di comunicazione con termocamera AMG8833.
- Pin: PB6 SCL (Serial Clock Line) e PB7 SDA (Serial Data Line).
- Modalità di trasferimento: DMA da periferica a memoria in normal mode.

3.1.2 USART3

- Utilizzo: output immagine termica verso monitor seriale.
- Pin: PB10 Tx e PB11 Rx.

- Configurazioni: modalità asincrona con baud rate 115200 bit/sec senza bit di parità.
- Modalità di trasferimento: DMA da memoria a periferica in normal mode.

Considerando che il sensore è utilizzato in modalità di acquisizione 10 fps, l'applicazione deve leggere la matrice di temperatura nel buffer di memoria di 128 byte e trasmetterlo in uscita sulla porta UART gestendo uno streaming video di bit rate pari a

$$\text{bit rate} = 128 \cdot 8 \cdot 10 = 10.24 kbps$$

3.1.3 ADC1,ADC2

- Utilizzo: conversione simultanea dei dati audio acquisiti dai microfoni a elettreti.
- Pin: PB0, canale 8 ADC1, e PB7, canale 7 ADC2.
- Modalità di trasferimento: due stream DMA configurati per trasferire in circular mode unità da mezza parola (16 bit) nei buffer d'ingresso dei due microfoni. La conversione di entrambi gli ADC è triggerata dalla TRGO line del Timer 2, il quale, essendo configurato con il clock a $25MHz$ (AHPB1 timers clock), $prescaler = 1$ e $period = 255$ triggerà la conversione con frequenza di campionamento

$$F_{tim2} = F_{clock}/(prescaler+1) \cdot (period+1) = 25 \cdot 10^6 / 2 \cdot 256 = 48.828 kHz$$

Tale frequenza rispetta il teorema di Nyquist considerando la risposta in frequenza del microfono nulla sopra i $20kHz$. Per effettuare il trasferimento continuo con DMA dal Data Register dell'ADC a buffer di memoria bisogna configurare i convertitori con il flag di DMA continous request abilitato.

3.1.4 DAC

- Utilizzo: uscita analogica su due canali separati (uno per microfono), per il playback dell'audio acquisito.
- Pin: PA4 e PA5 per i canali 1 (audio ADC1) e 2 (audio ADC2).
- Modalità di trasferimento: DMA in circular mode per trasferire i dati da buffer di uscita a periferica. Anche la conversione DA viene triggerata dall'update event del timer 2 ed è quindi simultanea alla conversione AD.

L'applicazione principale deve trasferire i campioni audio da buffer d'ingresso a buffer di uscita mentre gli stream DMA trasferiscono i dati a/da i buffer stessi in modalità continua per avere il playback in real-time; in questo è cruciale la scelta della taglia dei buffer per stabilire entro che tempi l'applicazione debba

completare il loop di playback per non perdere campioni in riproduzione e avere una latenza controllata. Infine, considerando che i valori di tensione in uscita dai due preamplificatori microfonici sono convertiti nel range 0 – 3V su 12 bit senza segno e trasferiti in memoria come interi a 16 bit, si può calcolare il bit rate dello streaming audio come

$$\text{bit rate} = 48.828 \cdot 16 \cdot 2 = 781 \text{ kbps}$$

3.1.5 GPIO Output

- Utilizzo: controllo driver del motore stepper.
- Pin: PD1-4.
- Modalità di utilizzo: set/reset delle uscite digitali in sequenza secondo un determinato pattern di stepping, di cui si parlerà in seguito.

3.1.6 ADC3

- Utilizzo: lettura della posizione del joystick su un solo asse, per comandare il movimento verso sinistra/destra del motore in modalità manuale.
- Pin: PA1 in analog mode.
- Modalità di trasferimento: lettura periodica in polling utilizzando il canale 1 del convertitore in single channel-single conversion mode.

3.1.7 EXTI LINES 2,4,5

- Utilizzo: pulsanti di toggle della modalità e single step left/right del motore.
- Pin: PE2-4 in external interrupt mode con rising edge detection.
- Modalità di utilizzo: l'interrupt generato dalla pressione di un tasto attiva un timer di debounce, allo scadere del quale lo stato del pin che ha generato l'interrupt verrà rivalutato per stabilire se il bottone sia stato o meno premuto.

3.1.8 EXTI LINE 15

- Utilizzo: gestione degli interrupt triggerati sul pin INT dalla camera termica.
- Pin: PE15 in external interrupt mode con rising edge detection.
- Modalità di utilizzo: set del flag di target identificato nella ISR di gestione dell'interrupt.

3.1.9 USART6

- Utilizzo: output print di debug.
- Pin: PC6 Tx e PC7 Rx in modalità asincrona con baud rate 115200 bit/sec, senza bit di parità.
- Modalità di trasferimento: trasferimento di stampe di debug dal buffer dei messaggi alla periferica UART.

Infine, è bene sottolineare come non ci sia uno stream di DMA responsabile del trasferimento dati per più di una periferica contemporaneamente e pertanto non risulta necessario stabilire una gerarchia di priorità intra-stream.

3.2 Device driver

In questa sezione vengono illustrati i device driver sviluppati per interagire con le periferiche. In particolare, sono state implementate strutture dati e funzioni per astrarre l'accesso alle funzionalità principali dei dispositivi messi in campo e si lascia come sviluppo futuro il completamento dell'interfaccia software con le funzionalità aggiuntive che nell'ambito di questo progetto non sono state utilizzate.

3.2.1 AMG8833

Per la termocamera è stata implementata una struttura dati che contiene un intero a 16 bit per l'indirizzo I2C e un puntatore a una struttura dati di tipo I2C_HandleTypeDef, che mantiene i parametri della porta I2C configurata come descritto nella sezione precedente. Inoltre, sono definiti come costanti gli indirizzi dei registri interni del sensore indicati sul datasheet e sono state implementate le funzioni che astraggono l'utilizzo base del sensore, ossia il reset, il test della presenza sulla linea I2C all'indirizzo configurato e le funzioni di lettura dell'immagine termica in polling e DMA mode. Tra le altre feature implementate utilizzate dall'applicazione troviamo l'abilitazione della interrupt mode, lo switching tra modalità di interrupt con soglia assoluta o con isteresi, l'impostazione i valori di soglia e la lettura della interrupt table, tabella di 64 bit che indica quali pixel sono stati misurati sopra/sotto le soglia nell'ultimo frame. Tutte queste funzioni consistono nella lettura/scrittura di registri tramite protocollo I2C e possono essere chiamate con un parametro di *max_retry* per ritentare fino a un massimo numero di volte la transazione I2C in caso di fallimento.

3.2.2 Motore

La struttura dati del motore contiene invece un array di 4 interi a 16 bit per memorizzare gli output pin di controllo e un campo per il riferimento alla porta. Viene inizializzata con una tra le modalità WAVE e FULL che determinano il pattern di stepping, che consentono il movimento del motore con risoluzione 0.1766° per step. Di seguito nelle tabelle 1 e 2 vengono indicati i due pattern di

attivazione sequenziale delle uscite di controllo per compiere uno step in un senso o nell'altro. Una volta impostata la modalità, ad ogni chiamata della funzione *step* il microcontrollore setta i pin di output in corretta sequenza grazie al campo *cur_idx* della struct che, incrementato/decrementato circolarmente da 0 a 3 ad ogni step in un verso o nell'altro, permette di conoscere in ogni momento a quale punto della sequenza il motore è arrivato e quali sono gli stati logici dei pin per compiere il prossimo step.

Inoltre, la struttura dati memorizza anche un indice angolare che, insieme al parametro di risoluzione, permette di conoscere a che angolo in gradi si trova il motore in un dato istante rispetto a un offset iniziale (ultima volta in cui è stato settato l'indice angolare=0 via software), consentendo l'implementazione di una funzione di movimento verso un angolo di destinazione espresso in gradi sia in modalità bloccante che non-bloccante (interrupt mode). Per la seconda modalità, è necessario che la struttura dati sia inizializzata con un timer la cui frequenza di reset determina la velocità di movimento e che venga chiamata la funzione *stepIT* nella di callback di gestione dell'interrupt di time elapsed.

3.2.3 Joystick

La struttura dati che astrae la periferica joystick è molto semplice: contiene un riferimento a una ADC_HandleTypeDef del convertitore a cui il joystick è collegato. Esporta al client solo una funzione per controllare se, al momento della chiamata, il joystick si trova in posizione centrale, destra o sinistra, leggendo in polling il valore di tensione in ingresso all'ADC.

3.3 Loop di controllo

Una volta documentate le configurazioni e le API sviluppate per interagire con la camera e il motore, si può passare all'analisi del loop principale di controllo. A livello più alto il dispositivo si trova a ciclare in 2 stati, uno di controllo manuale, in cui è possibile vedere l'immagine termica, monitorare il playback

pin0	pin1	pin2	pin3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Table 1: Wave stepping

pin0	pin1	pin2	pin3
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1

Table 2: Full stepping

sull'uscita audio e direzionare la camera, e l'altro di follow, in cui reagisce a un impulso sonoro sopra soglia cercando localizzare la sorgente, mantenendo attivo lo stream audio/video. Chiamando i due stati rispettivamente M ed F, nella figura 2 è illustrato il loop principale di controllo; per switchare da una modalità di funzionamento all'altra è sufficiente premere il bottone di toggle mode. Nella progettazione, si è cercato di mantenere il codice delle ISR il più possibile efficiente e si è quindi scelto di mantenere un meccanismo coerente di gestione degli interrupt per cui nel codice degli handler vengono settati bit di controllo mappati nella bit band region e nel loop principale si monitora lo status di questi flag per gestire la sequenza di esecuzione corretta. Durante la modalità manuale, è previsto che la camera venga allineata con i microfoni, in modo da settare l'offset angolare corretto per la modalità follow.

3.3.1 Display immagine termica

Come mostrato in figura 3, lo stream dell'immagine termica è gestito in maniera asincrona tramite i bit di status AMG_RD_START, AMG_RD_CPLT e AMG_OUT_CPLT. Viene implementata una macchina a stati finiti che verificando lo stato di questi bit esegue la seguente sequenza di controllo:

- se $\text{AMG_RD_START}=1$, resettalo e comanda la lettura dell'immagine termica in DMA mode dal sensore AMG8833 al buffer di 128 byte *img_buf* allocato in memoria RAM.
- se $\text{AMG_RD_CPLT}=1$, resettalo e comanda il trasferimento del buffer sulla seriale in DMA mode
- se $\text{AMG_OUT_CPLT}=1$, attiva un timer al cui scadere verrà ripetuta la sequenza.

Il primo dei tre flag ha come stato iniziale 1, mentre gli altri 0, dopodiché AMG_RD_CPLT viene settato dalla funzione di callback *HAL_I2C_MemRxCpltCallback* eseguita al termine del trasferimento I2C, AMG_OUT_CPLT dalla ISR associata al termine del trasferimento DMA su seriale *HAL_UART_TxCpltCallback*, mentre AMG_RD_START dalla callback eseguita allo scadere del timer6, impostato per avere generare un interrupt dopo 0.025s,

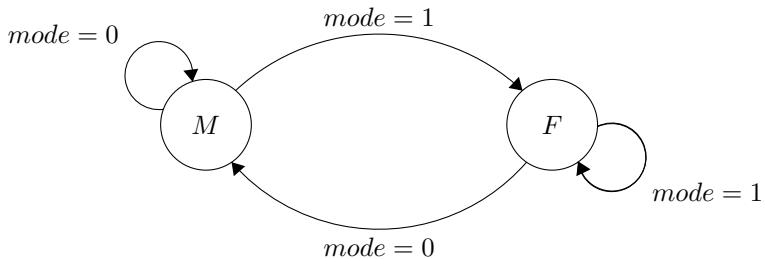


Figure 2: Main loop Finite State Machine

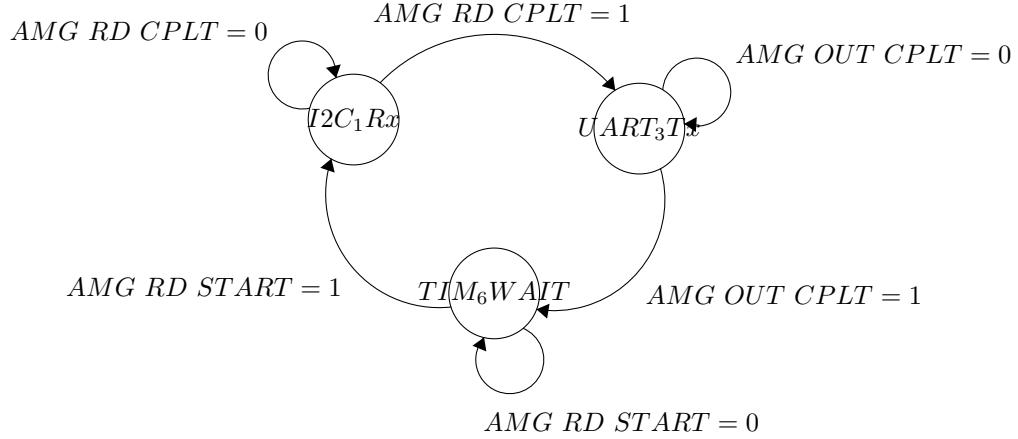


Figure 3: FSM stream immagine termica

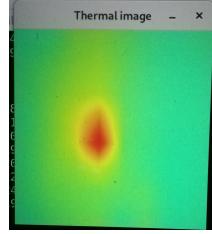


Figure 4: Immagine termica di un accendino di fronte alla camera

tempo dopo il quale verrà letto un nuovo frame dal sensore e prodotto in output. I risultati sperimentali dimostrano che con questo metodo è possibile gestire lo streaming video senza perdita di frame a 10 fps, consentendo alla CPU di occuparsi simultaneamente di playback audio e controllo motore.

3.3.2 Playback audio

Ricordando quanto detto nella sottosezione 3.1, lo stream audio è mantenuto facendo lavorare l'ADC1, l'ADC2 e il DAC triggerati dal medesimo timer (timer 2) a 48 kHz circa. I trasferimenti da periferica a buffer di memoria e viceversa sono gestiti dai DMA, i quali lavorano rispettivamente su un buffer d'ingresso e uno di uscita. Il loop principale si occupa del trasferimento dei campioni tra i due buffer di I/O, in modo da disaccoppiare le periferiche di acquisizione e riproduzione, con la tecnica del ping-pong buffer (o double buffer), il cui funzionamento è schematizzato in figura 5. In questa tecnica, per gestire due buffer di N campioni a 12 bit (codifica su unsigned short), ne vengono allocati due di dimensione 2N, divisi in higher half e lower half. Viene comandato il trasferimento DMA continuo di 2N campioni audio in circular mode da periferica a buffer (ADC) e da buffer a periferica (DAC) e si implementa un processo di

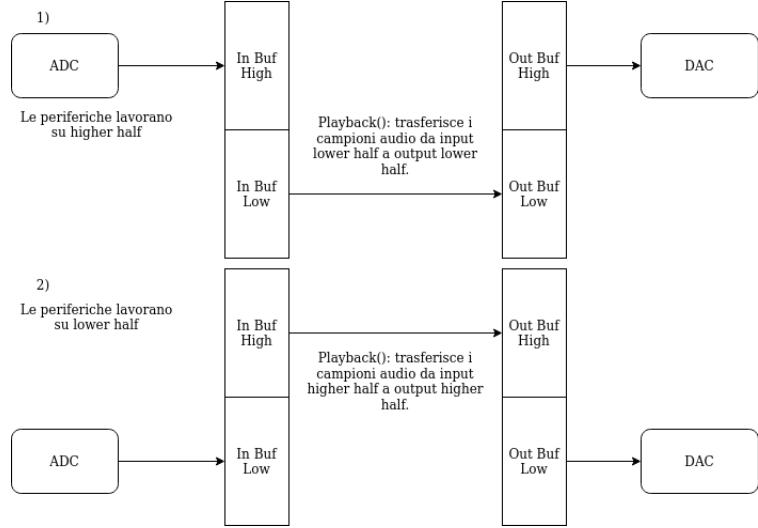


Figure 5: Schema logico ping-pong buffer

playback che trasferisce N campioni da higher half del buffer di input a higher half del buffer di output, mentre ADC e DAC mettono/prendono campioni dalla lower half. Quando il trasferimento DMA genera l'interrupt di metà trasferimento, vengono cambiati i puntatori utilizzati dal loop di playback, facendo sì che questa volta gli N campioni vengano portati da ingresso a uscita nella higher half, mentre le periferiche trasferiscono dalla lower half. All'interrupt di completamento del trasferimento DMA, i puntatori vengono cambiati di nuovo, per ricominciare dal primo punto; da qui la denominazione *ping-pong* buffer. Il loop che trasferisce i campioni da buffer d'ingresso a buffer di uscita viene eseguito solo quando sono settati entrambi i flag BUF1_CPLT e BUF2_CPLT dalle callback di metà/termine del trasferimento DMA, evento che indica la disponibilità di un nuovo buffer d'ingresso da mandare in playback.

Nell'applicazione si è scelto di utilizzare buffer da 512 campioni, che moltiplicati per la frequenza di campionamento generano interrupt di buffer complete (disponibilità di nuovi dati in input) ogni 0.01s, tempo entro il quale qualunque elaborazione audio deve terminare per evitare la sovrascrittura di buffer non ancora prodotti in output. La corretta acquisizione è stata testata osservando con l'oscilloscopio la conversione di un tono sinusoidale ad 1kHz.

3.3.3 Controllo manuale del motore

La routine di controllo manuale del motore è la più semplice; il timer7, impostato per generare un interrupt ogni 2 millisecondi, setta un flag che comanda la lettura in polling della posizione del joystick, la verifica dello stato di pressione dei bottoni per il posizionamento fine, ossia che comandano un solo step, e il conseguente movimento del motore in una direzione o nell'altra.

3.4 Modalità follow

Come anticipato in precedenza, in questa modalità il dispositivo reagisce ad un evento sonoro sopra una soglia RMS facendo una pan su tutto il suo campo visivo e fermando il motore contestualmente all'interrupt di target identificato generato dalla camera termica. Tarando queste soglie è possibile regolare la sensibilità al rumore del prototipo di sorveglianza e targettizzarlo per l'identificazione di un particolare tipo di sorgente di calore: se fissiamo ad esempio una soglia di temperatura a 28° possiamo identificare un umano a pochi metri di distanza dalla camera in absolute threshold interrupt mode, mentre se portiamo il valore a 40° possiamo calibrare il dispositivo per localizzare un oggetto in fiamme, utile in un ipotetico scenario di monitoraggio di esplosioni o incendi.

Per calcolare l'RMS è necessario preprocessare i campioni nel buffer d'ingresso sottraendo il DC offset per avere un segnale audio a media nulla: a tal fine viene mantenuto un array di 100 valori contenenti il valor medio degli ultimi 100 buffer elaborati (corrispondenti a circa 1s di audio), in modo da ottenere una stima affidabile del DC offset sui due canali calcolando la media degli ultimi 100 valori medi. Per evitare che picchi troppo brevi attivino la modalità di localizzazione, al rilevamento di almeno due buffer consecutivi sopra soglia, l'applicazione comanda il movimento interrupt mode del motore prima verso l'estremo più vicino del campo visivo e poi verso quello più lontano, in modo da non interrompere lo streaming audio/video durante la pan. Il movimento del motore è interrotto dalla ISR associata al pin INT della camera termica che, in seguito all'identificazione della possibile sorgente sonora nell'ambiente circostante, chiama la funzione *stop* del driver del motore stepper che blocca il timer di movimento. Una volta fermato il motore, il dispositivo stampa sulla porta di debug a che angolo del campo visivo è stata rilevato l'obiettivo.

4 Valutazione performance: tempi di esecuzione e consumi

Per cronometrare il tempo di esecuzione delle routine implementate è stata utilizzata la tecnica hardware che prevede il collegamento dell'oscilloscopio ad un'uscita digitale della scheda. Settando il pin HIGH prima dell'esecuzione e LOW al termine, possiamo misurare con precisione il tempo di esecuzione per verificare che i processi del main loop rispettino tutte le deadline imposte dalle periferiche real-time. La tecnica è stata preferita all'approccio software che utilizza il DWT data la disponibilità dello strumento di misura e il fatto che sia una tecnica consenta una stima precisa e allo stesso tempo semplice. Per farlo sono state definite delle macro di debug; decommentandole e compilando il codice, possiamo registrare il tempo di esecuzione delle funzioni d'interesse collegando lo strumento all'uscita PE11.

In primo luogo, è stato misurato ogni quanto il loop di controllo completa un

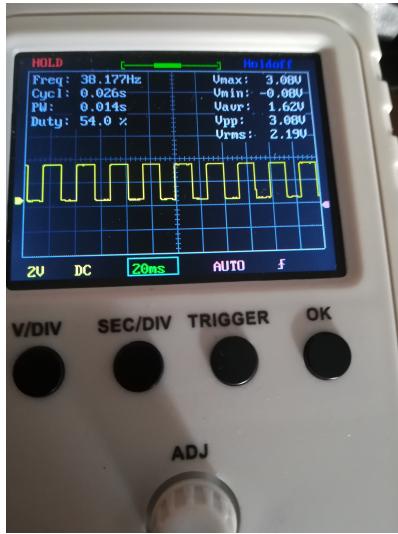


Figure 6: Streaming video misure di tempo

ciclo di acquisizione dell'immagine e trasmissione su UART per verificare che entro il decimo di secondo venga sempre letto almeno un frame dal sensore. La misurazione mostra che ogni 25 ms viene triggerato un nuovo ciclo di acquisizione, pertanto ogni frame acquisito dal sensore viene ridondato circa quattro volte in uscita; sebbene la ridondanza provochi un aumento di consumi, fa in modo che ci sia margine per tollerare eventuali errori di trasmissione. In secondo luogo, sono stati misurati i loop di playback e preprocessing audio (playback+rms trigger) per verificare la compatibilità con i tempi del DMA. Misurando la conversione, viene confermata l'ipotesi secondo cui abbiamo un interrupt dal DMA ogni decimo di secondo e resta quindi da verificare che le routine eseguano in un tempo inferiore per assicurarsi che la riproduzione audio in uscita sia corretta: le misurazioni confermano le ipotesi mostrando che il processo di playback impiega 1ms a trasferire i campioni da buffer d'ingresso a uscita, così come quello di preprocessing che ne impiega circa 2ms, visto che cicla una volta in più sull'array che contiene gli ultimi 100 valori medi dei buffer per calcolare il dc offset.

Per misurare i consumi si è scelto di utilizzare collegare l'amperometro al connettore Idd e misurare il consumo di corrente: la misurazione ha mostrato che vengono consumati circa 20 mAmp di corrente in entrambe le modalità. L'ottimizzazione energetica può essere migliorata, ma non si ritiene un grave problema visto che nel contesto dell'applicazione si ritiene ragionevole che il dispositivo sia alimentato dalla rete elettrica.

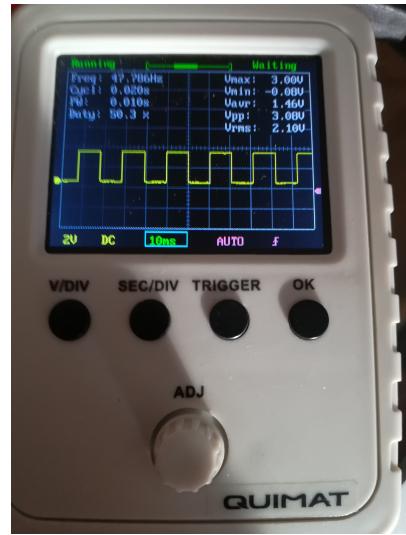


Figure 7: Interrupt DMA conversione audio

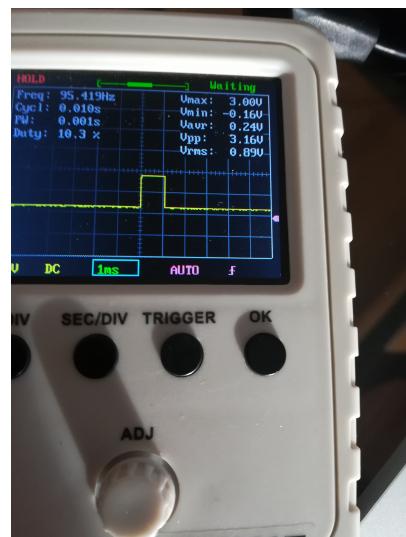


Figure 8: Playback audio misure di tempo