Relazione Progetto SE

Report written by Gusmara Andrea (831141).

Consegna del progetto

Si vuole progettare un sottosistema di acquisizione e prima elaborazione dati da sensori per un drone che verrà utilizzato per l'ispezione di torri eoliche. Il drone non necessita di essere certificato per il volo civile e il sottosistema di acquisizione/elaborazione dati deve essere poco costoso. Si è pertanto scelto di basare il sottosistema su un microcontrollore ARM e su sensori facilmente reperibili sul mercato. La versione che progetterete voi sarà basata sul microcontrollore STM32 F767ZI installato sulla scheda di sviluppo Nucleo usata a laboratorio e sui sensori sullo shield di espansione X-NUCLEO-IKS01A2, oltre ad altri apparati sensoriali non gestiti da questa board. Il sottosistema deve funzionare in tempo reale con opportuni vincoli temporali che saranno successivamente specificati. Il sottosistema interagisce con i seguenti sensori dello shield di espansione: accelerometro 1, accelerometro 2, giroscopio, magnetometro, barometro.

Il progetto è stato realizzato nell'ambiente di sviluppo integrato STM32CUbeIDE.

Scelte architetturali

Le scelte architetturali prese su questo progetto ricadono su utilizzo della tecnica del controllo da programma , in cui si presenta un ciclo infinito nel quale viene eseguito un polling per la gestione delle periferiche . Si è scelto inoltre l'utilizzo dei timer , settati per utilizzo con interrupt ad ogni counter overflow , per gestire le tempistiche dei vari compiti (o impropriamente task) del nostro sistema in time sharing . Le comunicazioni tra le schede dei sensori e la scheda Nucleo vengono inviati tramite il protocollo I2C , comunicazione seriale presente in tutte le schede.

Timing dei compiti

Per rispettare i requisiti (costraint di tempo) del sistema richiesti dalla consegna abbiamo deciso di utilizzare la tecnica dell'interrupt perché semplice allo stesso tempo adatta alle nostre esigenze, le quali appunto ricadevano sulla capacità del componente di lanciare un interrupt ad ogni counter overflow. Nel nostro sosftware utilizziamo quindi i Timer : TIM6 (assegnato al timing di acquisizione del data_raw_pressure) settato a 75Hz , TIM4 (assegnato al timing di acquisizione del data_raw_magnetic) settato a 10Hz ,TIM7 (assegnato al timing di acquisizione di data_raw_acceleration e data_raw_angular_rate) settato a 30 Hz, TIM3 (assegnato al timing di stampa su lcd) settato a 10 Hz.

Tenendo presente che il display ha una timing constraint di 1Hz e che la scrittura su display impiega circa 0,65 sec . Ho allora ipotizzato che per stampare i valori del secondo (s) corrente si doveva acquisire tutti i valori delle code in 1-0,65=0,35s. Quindi a una frequenza di 30 Hz possiamo acquisire 10 dati nel caso

dell'accelerometro e giroscopio , $\frac{1}{30} * 10 = 0{,}33\ s$. Ad una frequenza di 10 Hz possiamo acquisire 3 dati nel caso del magnetometro $\frac{1}{10} * 3 = 0{,}3s$. Invece 75Hz frequenza data dalla consegna per il barometro.

Dopo di che abbiamo settato i parametri (Prescaler e Period) di ogni timer in base alla seguente formula :

$$96Hz = x/((prescaler + 1) * (period + 1))$$

Dove x è la frequenza del timer che non vogliamo.

Funzioni User principali

All'interno del programma vengono utilizzate le seguenti funzioni :

inizialization() in cui vengono settato i valori booleani dei vari flag

lsm6dsl_init() , lsm303agr_init_xl() , lsm303agr_init_mg() , lps22hb_init_mg() sono funzioni di inizializzazione dei sensori in cui viene passato un handle e che ritornano una struttura del sensore.

acceleration1_acquisition() , acceleration2_acquisition() : prende in input la struttura dell'accelerometro e un intero che identifica la posizione del buffer in quell'istante; acquisisce i dati dall'accelerometro, poi vengono convertiti nei componenti del vettore di accelerazione sui 3 assi e memorizzati nei buffer corrispondente al sensore.

Magnetometer_acquisition() : prende in input la struttura del giroscopi e un intero che identifica la posizione del buffer in quell'istante; acquisisce i dati dal giroscopio, poi vengono convertiti nei componenti del vettore velocità angolare sui 3 assi e memorizzati nei buffer.

Gyroscope_acquisition() : prende in input la struttura del magnetometro e un intero che identifica la posizione del buffer in quell'istante; acquisisce i dati dal magnetometro, poi vengono convertiti ei componenti sui 3 assi del vettore densità di flusso magnetico e memorizzati nei buffer corrispondente al sensore.

Pressure_acquisition() : prende in input la struttura del barometro e un intero che identifica la posizione del buffer in quell'istante; acquisisce i dati dal barometro , convertito in un valore di pressione e memorizzati nei buffer corrispondente al sensore.

Compute_mean() : prende in input un intero che rappresenta la dimensione del buffer, l'indirizzo dei tre buffer dei dati sui quali calcolare la media e l'indirizzo del buffer dove mettere il risultato; la funzione calcola la media per ogni buffer e i dati finali vengono messi nel buffer dei risultati.

compute_mean_bar () :prende in input un intero che rappresenta la dimensione del buffer, l'indirizzo del buffer dei dati sul quale calcolare la media e l'indirizzo del buffer dove mettere il risultato.

Criticità

Il sistema presenta delle criticità rispetto alla risposta agli impulsi di pressione , si verifica che a volte gli impulsi non rispondono o non rispondono in modo corretto . Suppongo che il problema si

dovuto che la scheda se in fase di scrittura o di cancellazione non invii il segnale al microprocessore , dovuto magari al fatto al bus del clock occupato . Suppongo in oltre che la scheda invii un segnale unico per tutte le volte che è stato schiacciato ma senza essere stato inviato .