IMPIANTO DI ALLARME DOMESTICO

Real-Time Embedded System Project

CANDIDATI
Casalini Andrea
Silvestri Marco

Anno Accademico 2020-2021

DESCRIZIONE

Questo progetto consiste in una simulazione di impianto di allarme domestico attraverso l'uso di componenti hardware specializzati.

Il sistema viene attivato attraverso un bottone e l'utente potrà verificare il corretto azionamento attraverso apposite informazioni sul display LCD e relativo led di segnalazione che diventerà di colore giallo.

L'applicazione quando risulta attiva:

- Attraverso l'utilizzo di un sensore PIR HC-SR501 rileva il movimento all'interno di una stanza
- Con un sensore ultrasonic HC-SR04 rileva la presenza di un intruso in un ambiente esterno, ma comunque privato come può essere un balcone o un terrazzo.
- Mediante l'uso di un bottone viene simulata la forzatura di una finestra.

Nel caso in cui uno dei precedenti sensori riscontrasse una intrusione, viene avviato un segnale acustico attraverso un buzzer.

Sia nel caso in cui il buzzer si fosse attivato sia nel caso opposto, l'utente per disattivare il sistema dovrà inserire correttamente la password preimpostata per tale dispositivo ''1234'' e successivamente premere come tasto di invio ''#''.

Infine, l'utente potrà verificare il corretto passaggio in modalità offline del sistema di allarme attraverso il display LCD ed il led di segnalazione che tornerà del colore blu.

L'ambiente di sviluppo utilizzato è ArduinoIDE con l'ausilio della libreria freeRTOS per schede STM32L abbiamo compilato e caricato il software sulla scheda B-l475e-iot01a2 della famiglia STM32.

TASKS

- Thread_ON (priority 2) : attende che l'utente attivi il sistema attraverso il pulsante USER integrato sulla scheda. Quando si verifica questo evento, questo thread si occupa di rilasciare i semafori che bloccano i thread: Thread_WINDOWS, Thread_PIR, Thread_BALCONY, Thread_DISPLAY e Thread_KEYBOARD. Infine, cambia il colore del led di segnalazione da blu a giallo per indicare il corretto passaggio da OFF ad ON del sistema.

- Thread_WINDOWS (priority 1): viene svegliato da thread_ON e rileva la forzatura di una finestra attraverso l'utilizzo di un bottone. Nel caso in cui tale evento si verificasse viene rilasciato il semaforo sbloccando il Thread_BUZZER e vengono conseguentemente bloccati i seguenti thread: Thread_WINDOWS, Thread_PIR, Thread_BALCONY

- Thread_PIR(priority 1) : viene svegliato da thread_ON e rileva la presenza di un intruso all'interno di una stanza attraverso l'utilizzo di un sensore di movimento PIR HC-SR501.Nel caso in cui tale evento si verificasse viene rilasciato il semaforo sbloccando il Thread_BUZZER e vengono conseguentemente bloccati i seguenti thread: Thread_WINDOWS, Thread_PIR, Thread_BALCONY

- Thread_BALCONY(priority 1) : viene svegliato da thread_ON e rileva la presenza di un intruso in un ambiente esterno attraverso un sensore di distanza ULTRASONIC HC-SR04 . Nel caso in cui tale evento si verificasse viene rilasciato il semaforo sbloccando il Thread_BUZZER e vengono conseguentemente bloccati i seguenti thread: Thread_WINDOWS, Thread_PIR, Thread_BALCONY

- Thread_BUZZER (priority 2) : viene svegliato da uno de seguenti thread: Thread_WINDOWS, Thread_PIR e Thread_BALCONY. Attraverso un buzzer produce un segnale acustico che può essere interrotto solamente dal Thread_KEYBOARD introducendo la corretta password. Nel caso in cui la segnalazione di attivare il buzzer provenisse dal sensore di movimento PIR vengono attesi 10 secondi prima di avviare il segnale acustico in quanto potrebbe essere il proprietario che vuole disattivarlo attraverso il tastierino.

- Thread_DISPLAY (priority 2) : viene svegliato da thread_ON e si occupa di mostrare sul sensore di output LCD le informazioni riguardanti lo stato del sistema (allarme ON e allarme OFF) e l'inserimento della password.
- Thread_KEYBOARD (priority 2) : viene svegliato da thread_ON e attraverso l'uso di un tastierino KEYPAD MODULE rileva l'inserimento della password. Se la password inserita è errata fa ripetere l'inserimento, invece nel caso in cui fosse stata inserita correttamente blocca il sistema, spegne il buzzer nel caso in cui fosse attivo e rilascia il semaforo sbloccando il thread_ON.

SEMAFORI

- xBinSemON : semaforo per l'attivazione dell'allarme.
- xBinSemPIR : semaforo per la gestione del sensore PIR.
- xBinSemBUZZER : semaforo per la gestione del sensore BUZZER.
- xBinSemKEYBOARD : semaforo per la gestione del sensore KEYBOARD.
- xBinSemWINDOWS : semaforo per la gestione del sensore BUTTON che identifica la finestra forzata.
- xBinSemBALCONY: semaforo per la gestione del sensore ULTRASONIC.
- xMutex: semaforo adibito per la mutua esclusione all'accesso di risorse condivise tra i vari threads.
- xBinSemDISPLAY: semaforo per la gestione del sensore LCD DISPLAY.

REQUISITI DELL'APPLICAZIONE

I sensori utilizzati sono: bottone integrato su scheda, bottone, LCD Display, Ultrasonic HC-SR04 , PIR HC-SR501, Buzzer, Keyboard Module e leds integrati su scheda.

1. 1 task per canale input/output: Thread_ON, Thread_PIR, Thread_BUZZER, Thread_KEYBOARD, Thread_WINDOWS, Thread_BALCONY, Thread_DISPLAY.
2. ≥ 3 canali di input: Keyboard Module e 2 bottoni.
3. ≥ 2 sensori: Pir e Ultrasonic.
4. ≥ 2 canali di output : buzzer, display LCD, led Blue e led Yellow.
5. Per la comunicazione dei task molti ad uno sono stati utilizzati dei semafori di sincronizzazione.

REQUISITI ADDIZIONALI

Tra i requisiti addizionali sono stati soddisfatti i seguenti:

1. Utilizzo di GitHub per la condivisione del codice.
2. Esecuzione periodica dei task.
3. Utilizzo del tool CppCheck per verificare il corretto sviluppo del codice software secondo le politiche di MISRA C.
   Un esempio di violazione del codice riscontrato durante la fase di sviluppo è stata la dichiarazione di variabili all'interno dei task, ma al di fuori del while(1).