***A.A. 2020-2021***

*Candidati*

*Daniela Koreta  
Fedy Haj Ali*

Home Security system

Real-Time Embedded Systems Project

## Descrizione

Il progetto “Home Security System” consiste nella realizzazione di un prototipo di un sistema di allarme, pertanto il dispositivo viene pensato per un utilizzo domestico. L’applicazione mira quindi a simulare il funzionamento di un allarme antifurto mediante l’utilizzo di componenti hardware. In breve: l’apertura/chiusura delle finestre di un’ipotetica casa è stata simulata tramite l’utilizzo di pulsanti e led. L’utilizzo di un sensore di distanza ad ultrasuoni HC SR04 ha permesso il rilevamento di movimenti “sospetti” all’interno della casa mentre un segnalatore acustico permette l’emissione del suono di allarme. Il display LCD viene impiegato per mostrare messaggi e notifiche ed un sensore ad infrarossi, consente – tramite l’apposito telecomando – l’inserimento del codice per la disattivazione dell’allarme.

## Software tools

Gli strumenti software utilizzati per la realizzazione del progetto sono: Visual Studio Code come ambiente di sviluppo, Arduino IDE – con la libreria FreeRTOS per Arduino – per la compilazione e il caricamento del programma sulla scheda ESP32 e git per la condivisione di codice.

## tasks

Task dell’applicazione:

* TaskSlave: sono i 3 task che rappresentano le finestre della casa, e dal punto di vista hardware il loro funzionamento è messo in atto da pulsanti e led. Le finestre sono inizialmente tutte chiuse. Premendo un pulsante viene simulata l’apertura di una finestra che provoca lo scattare dell’allarme.
* TaskAlarm:
* TaskMain:
* TaskConnection:
* TaskMovementDetection:
* TaskDisplay:
* TaskPassword:
* TaskSetAlarmButton:

## Semaphores

Semafori e mutex:

* mutex\_home
* mutex\_alarm
* mutex\_movement
* mutex\_password
* mutex\_ldc

## application requirements

I requisiti dell’applicazione sono stati soddisfatti nel seguente modo:

* I canali dell’applicazione sono: un sensore di prossimità ad ultrasuoni HC SR04, 3 pulsanti, 4 led e 1 sensore ad infrarossi.
* 1 task come canale di input/output: ?
* ≥ 3 canali di input: telecomando (per inserimento codice), applicazione MQTT, pulsanti;
* ≥ 2 sensori oppure 1 sensore + 1 dispositivo di rete: sensore a infrarossi, sensore di prossimità ad ultrasuoni HC SR04, Wi-Fi integrato nella scheda ESP32;
* ≥ 2 canali di output: buzzer, applicazione, display, led, ledRGB;
* ≥ 1 task per comunicazione molti-a-uno: TaskMain che permette la comunicazione tra il sistema di allarme e l’applicazione.

## additional requirements

Tra i requisiti addizionali:

* Git versioning: utilizzo di git per la condivisione di codice;
* Periodic execution (timer-based tasks): esecuzione periodica dei task
* Task WCET analysis measurement-based: analisi del Worst-Case Execution Time tramite la funzione vTaskGetTickCount() di FreeRTOS;
* MISRA C compliance: verifica dell’adempimento alle linee guida di sviluppo software nel contesto dei sistemi embedded tramite l’utilizzo del tool di analisi *cppcheck*.