```
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"
#include "portmacro.h"
#include <stdio.h>

#define FIRE_ALARM_PRIORITY      3
#define SECURITY_BREACH_PRIORITY 2
#define LOW_BATTERY_PRIORITY     1

#define FIRE_ALARM_PIN           0
#define SECURITY_BREACH_PIN      1
#define BATTERY_VOLTAGE_PIN      2

volatile BaseType_t fire_alarm_triggered = pdFALSE;
volatile BaseType_t security_breach_triggered = pdFALSE;
volatile BaseType_t low_battery_triggered = pdFALSE;

void fire_alarm_ISR(void) {
    fire_alarm_triggered = pdTRUE;
}

void security_breach_ISR(void) {
    security_breach_triggered = pdTRUE;
```

```
}

void battery_check_ISR(void) {
    if (/* Check battery level */) {
        low_battery_triggered = pdTRUE;
    }
}

void fire_alarm_task(void *pvParameters) {
    for(;;) {
        if (fire_alarm_triggered) {
            printf("Fire Alarm Triggered! Activating emergency response...\n");
            fire_alarm_triggered = pdFALSE;
        }
        vTaskDelay(10);
    }
}

void security_breach_task(void *pvParameters) {
    for(;;) {
        if (security_breach_triggered) {
            printf("Security Breach Detected! Triggering alarm...\n");
            security_breach_triggered = pdFALSE;
        }
        vTaskDelay(10);
    }
```

```
}

void low_battery_task(void *pvParameters) {
    for(;;) {
        if (low_battery_triggered) {
            printf("Low Battery Detected! Please charge the system.\n");
            low_battery_triggered = pdFALSE;
        }
        vTaskDelay(10);
    }
}

void init_hardware(void) {
    // Configure pins for external interrupts
    configure_external_interrupt(FIRE_ALARM_PIN, fire_alarm_ISR);
    configure_external_interrupt(SECURITY_BREACH_PIN, security_breach_ISR);
    configure_timer_interrupt(battery_check_ISR);
}

void configure_external_interrupt(int pin, void (*ISR_handler)(void)) {
    // Configure external interrupt
}

void configure_timer_interrupt(void (*ISR_handler)(void)) {
    // Configure timer interrupt for periodic battery check
}
```

```
int main(void) {
    init_hardware();

    xTaskCreate(fire_alarm_task, "FireAlarm", 100, NULL, FIRE_ALARM_PRIORITY, NULL);
    xTaskCreate(security_breach_task, "SecurityBreach", 100, NULL,
SECURITY_BREACH_PRIORITY, NULL);
    xTaskCreate(low_battery_task, "LowBattery", 100, NULL, LOW_BATTERY_PRIORITY, NULL);

    vTaskStartScheduler();

    while(1);
}
```