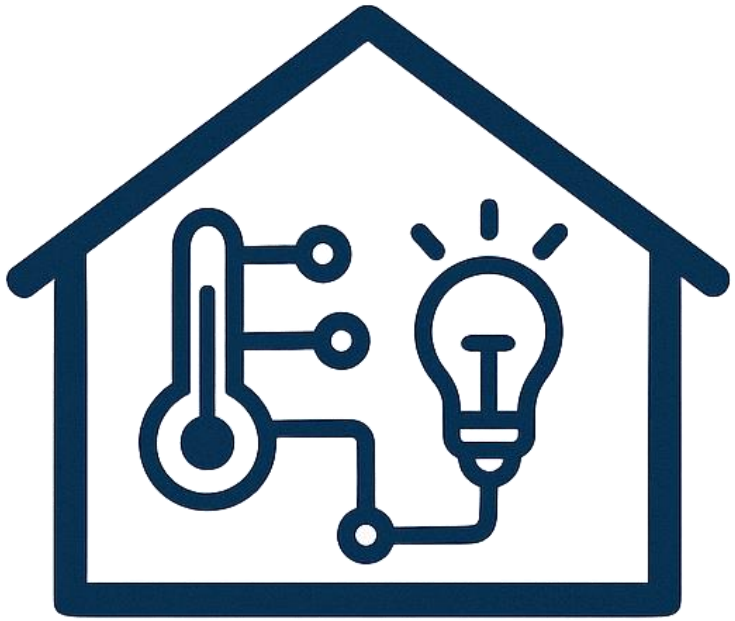
SMART ROOM MONITOR

your all-in-one environmental sidekick



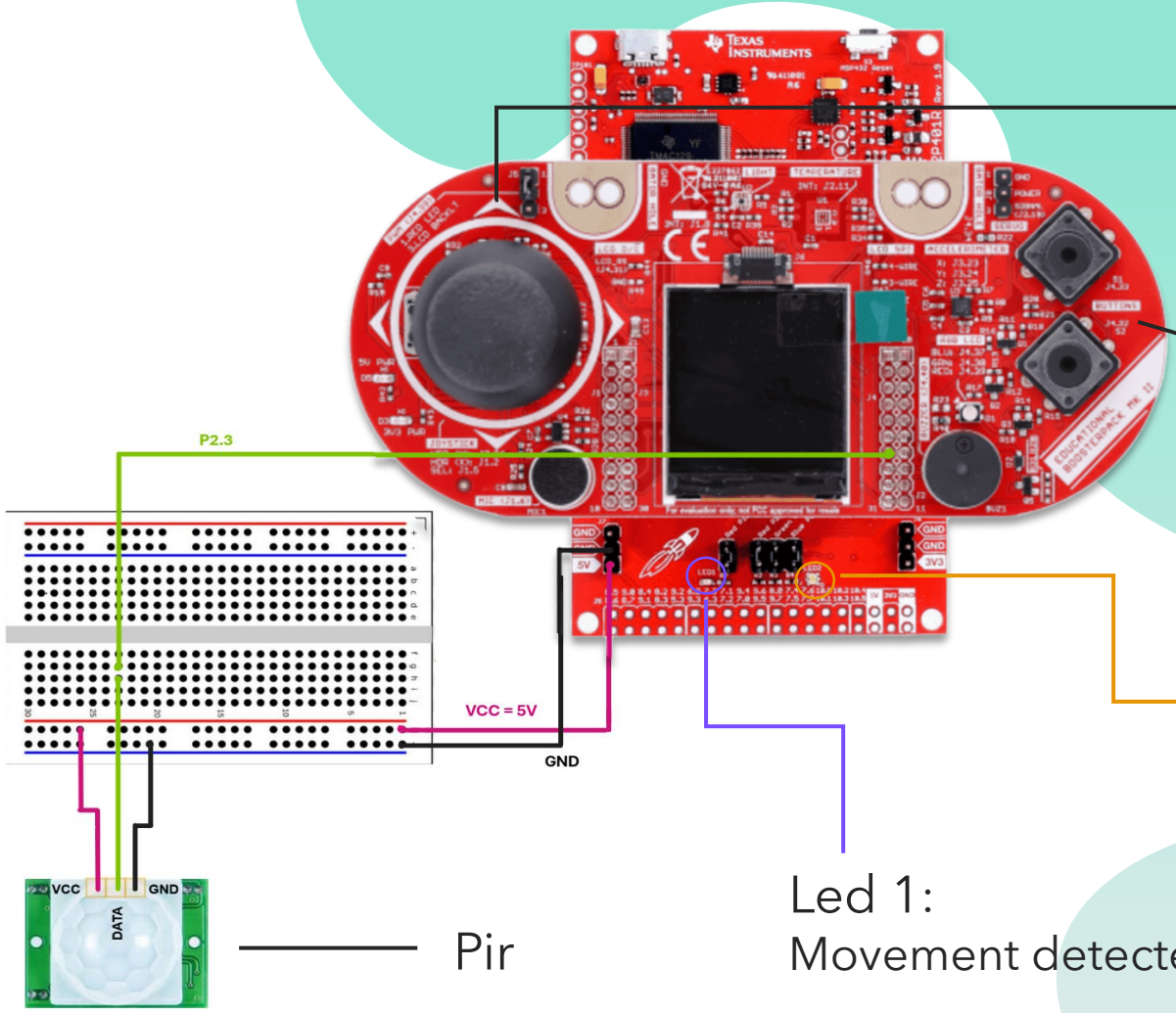
Stefano Corelli
Elisa Sterpu
Mattia Bernabè
Nicolas Venturi

Embedded Software for the Internet of Things

Introduction

- Four easy-to-navigate modes
- Ambient heat & clock
- Light intensity & illumination mode
- Safety alerts

Software-Hardware Interaction



Select Button

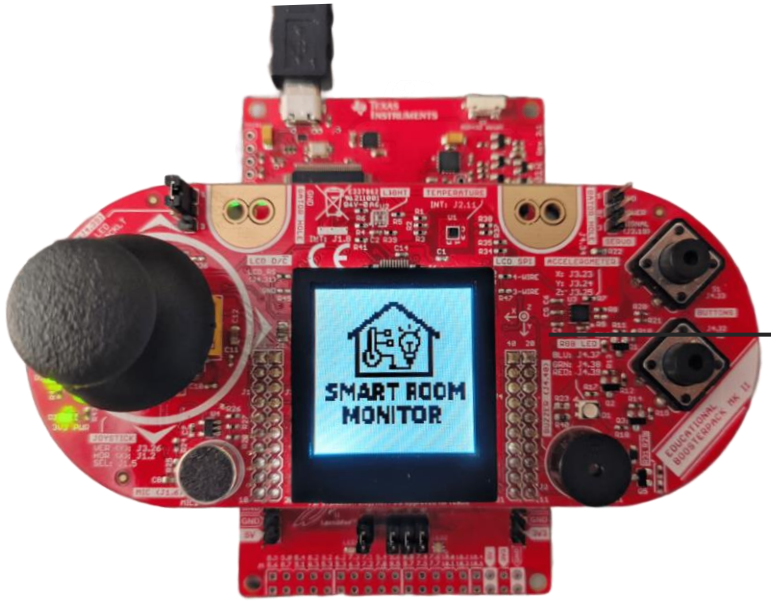
Interactions buttons

Led 2:
Yellow – earthquake
Red – high temperature
Blu – low temperature

Led 1:
Movement detected

Pir

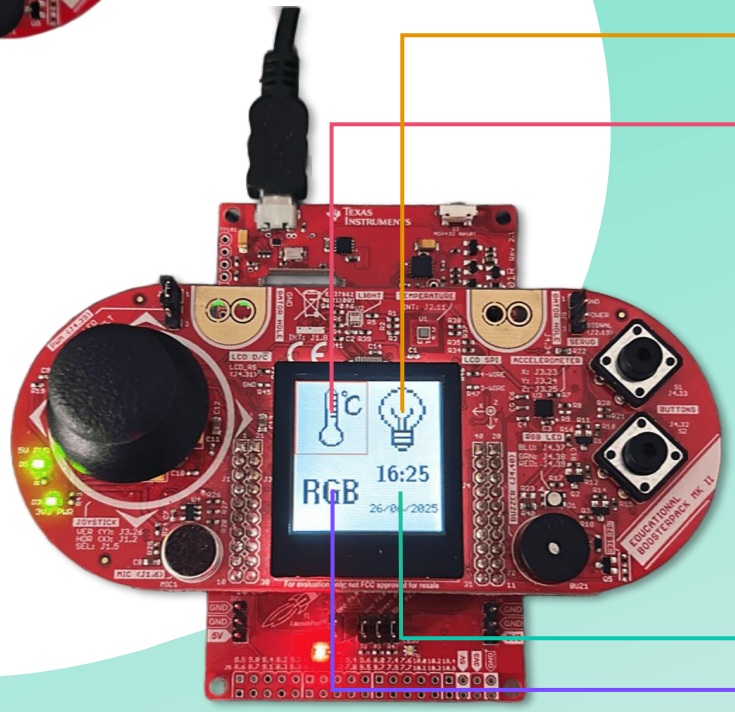
Software Architecture



Project Logo:

~3 sec then 4Q menu

After 3 sec



4Q menu

Thermometer:

in Celsius

Light Sensor:

Light intesity

RGB modifier:

select led color

Clock:

time modifier

Every team member worked together and is responsible for the whole project

Code Snippets

```
void drawTemperatureScreen_Alert(bool sameDataDisplay) {
    temp = getTemperature();
    if(!sameDataDisplay && temp >40) {
        Graphics_clearDisplay(&g_sContext);
        // Red background for the alarm
        Graphics_setForegroundColor(&g_sContext, GRAPHICS_COLOR_RED);
        Graphics_fillRect(&g_sContext, 0, 0, 128, 128);
    }else if(!sameDataDisplay && temp < 15){
        Graphics_clearDisplay(&g_sContext);
        // Blue background for the alarm
        Graphics_setForegroundColor(&g_sContext, GRAPHICS_COLOR_BLUE);
        Graphics_fillRect(&g_sContext, 0, 0, 128, 128);
    }


    Graphics_setForegroundColor(&g_sContext, GRAPHICS_COLOR_BLACK);

    // 1. String "HIGH TEMPERATURE" at the top
    Graphics_setFont(&g_sContext, &g_sFontCmss12);
    if(temp > 40)
        sprintf(string, "HIGH TEMPERATURE !");
    else if (temp < 15)
        sprintf(string, "LOW TEMPERATURE !");
    Graphics_drawStringCentered(&g_sContext, (int8_t *)string,
                                AUTO_STRING_LENGTH, 64, 20, OPAQUE_TEXT);
```

Alarm Screen

```
case LED_DIM:
    // Uses lux value read before
                if (lux >= 1000.0) {
                    red = green = blue = PWM_PERIOD;
                } else if (lux <= 10.0) {
                    red = PWM_PERIOD;
                    green = (uint16_t)(PWM_PERIOD * 0.6);
                    blue = (uint16_t)(PWM_PERIOD * 0.2);
                } else {
                    float t = (lux - 10.0) / (1000.0 - 10.0);
                    red = PWM_PERIOD;
                    green = (uint16_t)(PWM_PERIOD * (0.6 + 0.4 * t));
                    blue = (uint16_t)(PWM_PERIOD * t);
                }
                break;
```

Dimmerable light

```
void showSplashScreen(void) {
    GrImageDraw(&g_sContext, &dropLogoPalette, 0, 0);
    GrFlush(&g_sContext);
    __delay_cycles(48000000 * 1); // 5s delay
}
void drawQuadrantMenuWithSelection(uint8_t selected) {

    uint8_t quadWidth = 64;
    uint8_t quadHeight = 64;

    // Quadrant 0: Top Left (Thermometer)
    Graphics_drawImage(&g_sContext, &termometroImage, 0, 0);
    if (selected == 0) {
        Graphics_setForegroundColor(&g_sContext, ClrRed);
        Graphics_drawRectangle(&g_sContext, &(Graphics_Rectangle){0, 0, quadWidth-1, quadHeight-1});
    }

    // Quadrant 1: Top Right (Light bulb)
    Graphics_drawImage(&g_sContext, &dropLampPalette, quadWidth, 0);
    if (selected == 1) {
        Graphics_setForegroundColor(&g_sContext, ClrRed);
        Graphics_drawRectangle(&g_sContext, &(Graphics_Rectangle){quadWidth, 0, 2*quadWidth-1, quadHeight-1});
    }
```

4Q menu

Debug & Testing

Encountered problems:

Accelerometer:
Motion when stationary:
1. Insufficient calibration
2. Electronic noise
3. Inappropriate threshold

Buzzer:
Set on timer A0 (same as red and green led)

Pir:
First used P6.4 instead of P2.3. (P6.4 is for I2C)

Testing:

Black-Box:
Each team member tested the software parts as black-box to ensure the result's correctness.

Future Upgrades

- Onboard Battery Backup
- Configurable PIR Hold-Time in Firmware
- ESP-Powered Remote Connectivity