Report

Programming Language: C Programming (bare-metal embedded C)
Hardware: NXP FRDM-KL25Z (ARM Cortex-M0+) - Freedom Development Board
Software Tool: NXP MCUXpresso IDE

Objective:
This code was developed as part of the course project at the University of Colorado Boulder (MS in EE – Embedded Systems). The aim of this project (Plant Health Monitoring System) was to design and develop a prototype that could be used to determine the health/condition of a plant based on the various environmental parameters by measuring the temperature, humidity, and light intensity in the surroundings.

Overview:

1. A development board by NXP was used to accomplish this project. This board was interfaced with the two sensor modules; SHT21 (sensors measure temperature and humidity) and TSL2561 (sensor measures light intensity in lux). The sensors were interfaced with the I2C communication protocol.

2. After fetching the values from the sensors, the parameters were used to control the RGB LED controlled using PWM. This LED was used to indicate different situations by glowing a particular color. For example – if the temperature, humidity, and light intensity values fall below a certain threshold*, this condition is indicated by glowing a red-colored LED, if they are between their two respective threshold values, the situation would be indicated by glowing a yellow-colored LED. If everything is fine, then the LED would be turned green in color.

   However, there should be several combinations such as if temperature and humidity fall below a threshold but not the luminosity, if humidity and luminosity go below a risky value, etc. All these situations could be indicated by glowing different colors. ***I considered the above-mentioned situations but did not implement them as this project was just a prototype.***

3. Moreover, instead of using the inbuilt *printf()* and *scanf()* functions, I defined functions to tie the UART code with the standard I/O functions. Functions such as ***__sys_read*** and ***__sys_write()*** were implemented to link the system I/O. This UART implementation was used to display the sensor values on a serial terminal of a PC with a baud rate of 115200 while simultaneously glowing the RGB LED.

4. This mechanism of fetching values from the sensors using I2C was polling-based (not interrupt-based) as there was no power limitation. However, the systick timer and UART implementation were interrupt-driven.

Code Description:

1. The program begins by calling the initialization functions (peripheral configurations) for the system clock, systick timer, UART, PWM, I2C0 and I2C1, and TSL2561 sensor module.
2. Several peripheral configurations were done in their respective source files (.c files) with the header files (.h files).
3. In order to test the UART implementation, the circular buffer is tested before running the program by calling the *Test_Cbuff()* function.
4. *while(1)* loop is used to fetch the value from the sensors and use those values to glow the RGB LED and display the parameters on a serial terminal along with the current situation message (Danger, Warning, and Safe) if the log statements are enabled. The program will exclude the logging statements if the LOG macro is disabled in ***common.h*** file.
5. The values are fetched from the sensors every two seconds using a systick_timer. Hence, this was a polling mechanism and an interrupt-based mechanism to get the sensor values as there was no power limitation on this application.
6. An embedded C coding standard is followed which was provided by the Department of Electrical Engineering to write clean, clear, well-read, and efficient C programs. However, several other coding standards are widely followed in the embedded software domain and vary from company to company.

** threshold: the fixed value for temperature, humidity, light intensity was set by the me (programmer) to demonstrate the project by changing the color of LED.*