AEMBEDDED SYSTEM DESIGN
PROJECT 1: SMART LIGHT (V1.2)

OVERVIEW

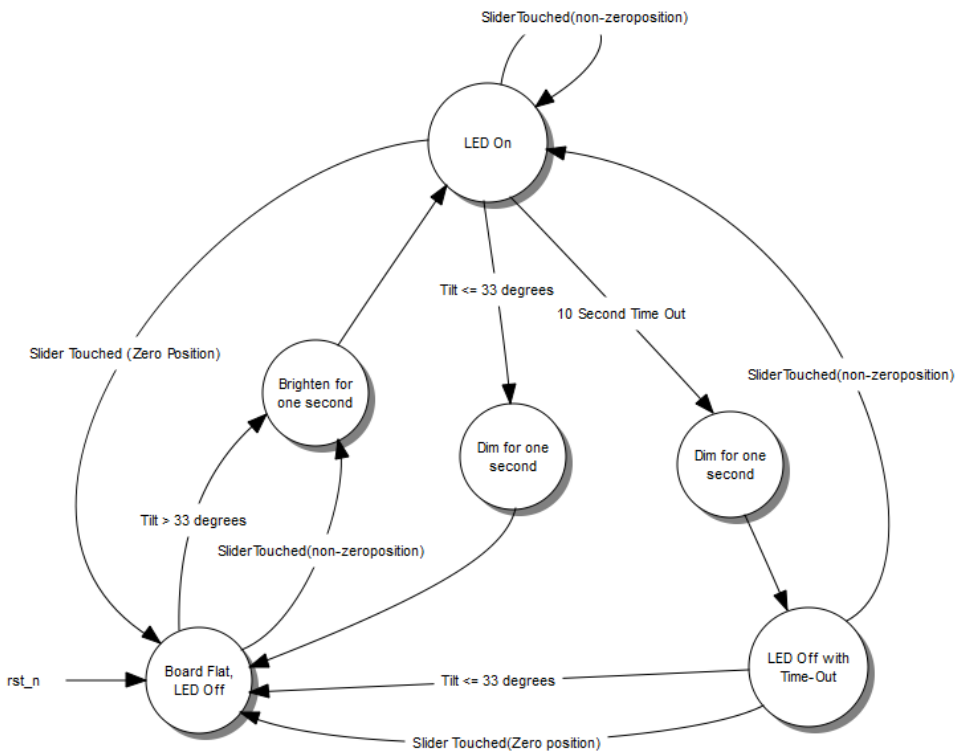
For this project you will program your Freedom board to behave as a smart light with these features:

- Triggers
  - Board orientation
  - Touch sense interface to set brightness
- Fade in, fade out
- Low battery warning (561)

You will modify existing demonstration code from Freescale to complete this project.

DETAILS

REQUIREMENTS



1. Operational states:
    a. ON: White light is to be on:
        i. When board is more than 33 degrees from horizontal (unless modified by other requirements (e.g. time-out)),
        ii. When set by the touch sense slider.
    b. OFF: White light is off otherwise.
2. Time-out: Light turns off after being on for 10 seconds, regardless of board angle.
3. Fader: When switching between ON and OFF states, the white light will gradually brighten or dim over a period of 1 second.
4. Low battery warning: If the voltage on the P3V3 supply rail is below 3.2 V, use the red LED to indicate this (ECE 561 only)
    a. OFF state: flash the red LED for 100 ms every 5 seconds (no fading).
    b. ON state: LED is white for 2 seconds, red for 250 ms (no fading).

PERIPHERALS USED

THREE-AXIS ACCELEROMETER

Use the Freedom board's accelerometer to determine board orientation. For details, see the introductory course module on Serial Communication (Presentation, I2C and Accelerometer Demonstration Code) and the FRDM-KL25Z User's Manual.

RGB LEDS AND TIMER/PWM MODULES

Use the RGB LEDs on the Freedom board to create the white or red light. Drive them with pulse-width modulated signals from the timer/PWM modules (NOT the GPIO modules) to control the brightness and color of the LED.

LED	Port Connection	Timer/PWM Module
Red	Port B 18	TPM2 Channel 0
Green	Port B 19	TPM2 Channel 1
Blue	Port D 1	TPM0 Channel 1

For details, see the introductory course module on Timers (Presentation, Timer Demo Code) and the FRDM-KL25Z User's Manual.

TOUCH SENSE SLIDER

Use the TSI slider to detect user input. Freescale provides the Touch-Sensing Software Suite (TSS) driver software which you can start with, or integrate into an existing project. Download the latest version of TSS from the download tab on http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TSS and run it to install TSS. On the Select Features window, ensure that these items are checked: FRDMKLXX Examples, Documentation, PC Host Tools. Read the Touch Sense Software User Guide (TSSUG.pdf). The code and documentation will be installed at C:\Freescale\Freescale_TSS_3_1.

INSTALLING AND CONFIGURING THE TSS DEMO

You will start with the TSS demonstration code for the Freedom KL25Z board. The project directory is C:\Freescale\Freescale_TSS_3_1\examples\FRDMKLXX_DEMO. Open the project FRDMKLXX_DEMO\build\uv4\FRDMKL25Z_DEMO\FRDMKL25Z_DEMO.uvproj with uVision (MDK).

- Configure the project as follows:
    - In TSS_SystemSetupGlobal.h, change the #define of TSS_USE_FREEMASTER_GUI to 0. If this file is not visible in MDK's Project View window, build the project to make it visible, or else add the file manually to the project.
    - In board.h, change the #define of LOW_POWER_TSI to 0
- Build the project using Project->Rebuild all target files.
- Configure the debugger to change its settings to work with the Freedom KL25Z. Select Project->Options for Target "FRDMKL25Z_DEMO".
    - Select the Debug tab. Make sure your PC is connected via USB to your Freedom board's mini-USB port labeled "SDA".
        - On the top right side of the window, in the Use box select CMSIS-DAP Debugger
        - Click Settings.
            - In the Cortex-M Target Driver Setup window (Debug tab), on the left in the CMSIS-DAP – JTAG/SW Adapter section, for the Port select SW. If your Freedom board is connected (and has the proper CMSIS DAP driver), the SW Device section should show a device called ARM CoreSight SW-DP.
            - Click OK.
    - Select the Utilities tab.
        - In the Configure Flash Menu Command section, select Use Target Driver for Flash Programming and check the Use Debug Driver box.
        - Click Settings.
            - In the Cortex-M Target Driver Setup window (Flash Download tab), in the Programming Algorithm section, there should be one entry and it should be MKXX 48Mhz 128kB Prog Flash 128k On-chip Flash. If it does not match that, then select that entry, click Remove, then click Add.
                - In the Add Flash Programming Algorithm window, select MKXX 48Mhz 128kB Prog Flash / 128k / On-chip Flash (it may be the first entry) and click Add.
            - Still in the Cortex-M Target Driver Setup window (Flash Download tab), in the RAM for Algorithm section the Start address should be 0x1FFFF000 and the Size should be 0x4000. Correct these values if they are wrong.
        - Click OK to exit the Cortex-M Target Driver Setup window.
    - Click OK to exit the Options for Target 'FRDMKL25Z_DEMO' window.
- Now you can download the project to your Freedom board, press the reset button and then experiment with the slider.
- Note that the demo code some undesirable features which you will need to correct:
    - Turning off the LEDs after a few seconds, then flashing the green LED.
    - White LED is too dim.

UNDERSTANDING THE TSS DEMO APPLICATION STRUCTURE

The TSS demonstration code is large and complex, with many configuration options (e.g. for different types of MCU) and multiple activities operating concurrently. Refer to the TSS documentation, located at C:\Freescale\Freescale_TSS_3_1\doc.

- User Guide. TSSUG.pdf. Chapter 3 is especially useful. You can ignore Chapters 6 and 7.
- API Reference Manual. TSSAPIRM.pdf
- Release Notes. TSS_Release_Notes.pdf

Additional tips

- Find the system's interrupt service routines and exception handlers by examining isr.h, startup_MKL25Z4.s (with Text Editor view, not Configuration Wizard),

FOR REFERENCE ONLY: ADDING TSS TO AN EXISTING PROJECT

Adding TSS to your existing project requires adding some source files for configuration and also linking with the TSS library. To do this, follow the procedure in section 2.3 (MDK-ARM uVision for ARM), with these comments:

- Figure 2-8 shows an example project in the screenshot. Don't try to find that particular project; use your own project instead.
- You will use uVision 5, though the user guide refers to uVision 4.
- You may wish to copy the file TSS_SystemSetup.h from the FRDMKL25Z project above (at C:\Freescale\Freescale_TSS_3_1\examples\FRDMKLXX_DEMO\build\uv4\FRDMKL25Z_DEMO)

ANALOG TO DIGITAL CONVERTER FOR VOLTAGE MEASUREMENT

Use the analog to digital converter to determine the voltage on the P3V3 supply rail.

CIRCUIT MODIFICATIONS

Although you could use a variable voltage power supply to power the Freedom board for testing, you can switch the P3V3 rail between two levels instead. This involves selecting whether to use a protection diode which lowers the voltage provided to P3V3. When the diode is shorted, P3V3 will be at about 3.3 V. When the diode is not shorted, P3V3 will be at about 3.1 V.

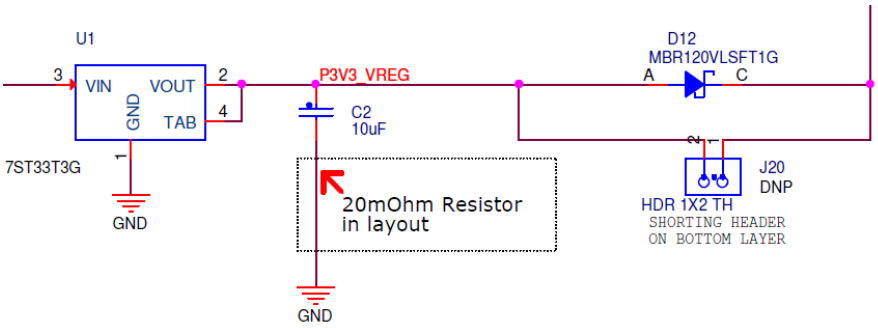


Figure 1. Power supply output circuit

Figure 1 shows the power supply circuit of the Freedom KL25Z board. Linear voltage regulator U1 drops the input voltage (on pin 3) to 3.3 V (on pins 2 and 4, and net P3V3_VREG). Diode D12 exists to protect the voltage regulator. If a higher voltage is applied to the P3V3 net, then D12 will be reverse-biased and will not conduct, isolating U1. However, shorting header J20 has been added in parallel with D12. The PCB actually has a trace which shorts out J20, as shown in Figure 2. As a result the diode never conducts and never drops the voltage.

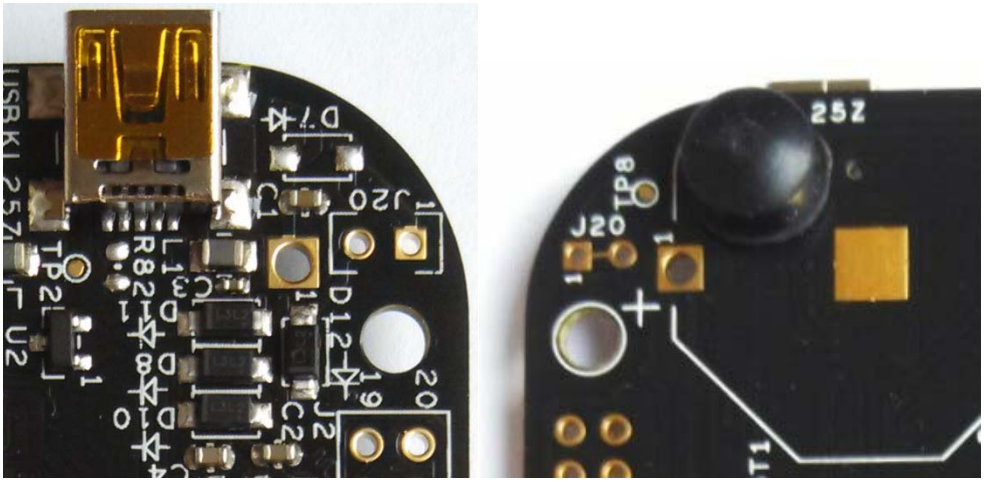


Figure 2. Detail of protection diode D12 and jumper J20 on PCB (left: PCB top, right: PCB bottom)

In order to allow the diode to drop the voltage, cut the PCB trace on the bottom of the PCB which shorts J20. To short out the diode again, place a small wire (e.g. paperclip) in the holes of J20.

MEASURING THE VOLTAGE

Measuring the P3V3 supply voltage with the ADC is a bit more complicated than it might seem initially. The ADC calculates a ratio (V_{in}/V_{ref}) between the input voltage V_{in} and a reference voltage V_{ref}.

$$ADC\_result = (V_{in}/V_{ref})*2^{ADC\_Resolution}$$

However, by default V_{ref} is connected to the P3V3 supply rail (actually P3V3_KL25Z, which is at the same voltage). If we connect V_{in} to P3V3 to measure it, the ADC will always provide a full-scale reading (e.g. 0xffff).

The MCU includes a fixed band-gap voltage reference (V_{BG} = 1.0 V ±0.03 V) which can be measured by the ADC by selecting input channel 27. Further details on the band-gap reference can be found in the KL25 Sub-family Data

Sheet (Section 5.2.2) and the KL25 Sub-family Reference Manual (Chapter 28, and Section 3.7.1.3.1). Measuring this input voltage will allow us to calculate the value of V_{ref}, which connected to the P3V3 supply rail.

$$V_{Ref} = (V_{BG} / ADC\_result) * 2^{ADC\_Resolution}$$

You can use the following code to configure the ADC and calculate the P3V3 rail voltage.

```
#define VBG_VALUE (1.00)

void Init_ADC(void) {

    SIM->SCGC6 |= (1UL << SIM_SCGC6_ADC0_SHIFT);
    ADC0->CFG1 = ADC_CFG1_ADLPC_MASK | ADC_CFG1_ADIV(0) | ADC_CFG1_ADICLK(0) |
        ADC_CFG1_ADLSMP_MASK | ADC_CFG1_MODE(3);
    ADC0->SC2 = ADC_SC2_REFSEL(0); // VREFHL selection, software trigger
    PMC->REGSC |= PMC_REGSC_BGBE_MASK;
}

float Measure_VRail(void) {
    float vrail;
    unsigned res=0;

    ADC0->SC1[0] = ADC_SC1_ADCH(27); // start conversion on channel 27 (Bandgap
reference)
    while (!(ADC0->SC1[0] & ADC_SC1_COCO_MASK))
        ;
    res = ADC0->R[0];
    vrail = (VBG_VALUE/res)*65536;
    return vrail;
}
```

DELIVERABLES

- Zipped archive of project
- Documentation, electronic submission. Can be hand-drawn and scanned, or created on computer
    - Architecture diagram showing software and hardware components and their communication paths
        - Original TSS application, as configured for the KL25Z
        - Your modified version for Project 1
    - Sequence diagrams showing examples of communications between HW and SW components. If you like, use this tool (http://sdedit.sourceforge.net/) to create your sequence diagrams
        - Original TSS application, as configured for the KL25Z
        - Your modified version for Project 1
    - Development effort (time) tracking
        - Estimated hours
        - Hours spent
- Demonstration