

MIC5 Week 3 – lab exercise

Introduction

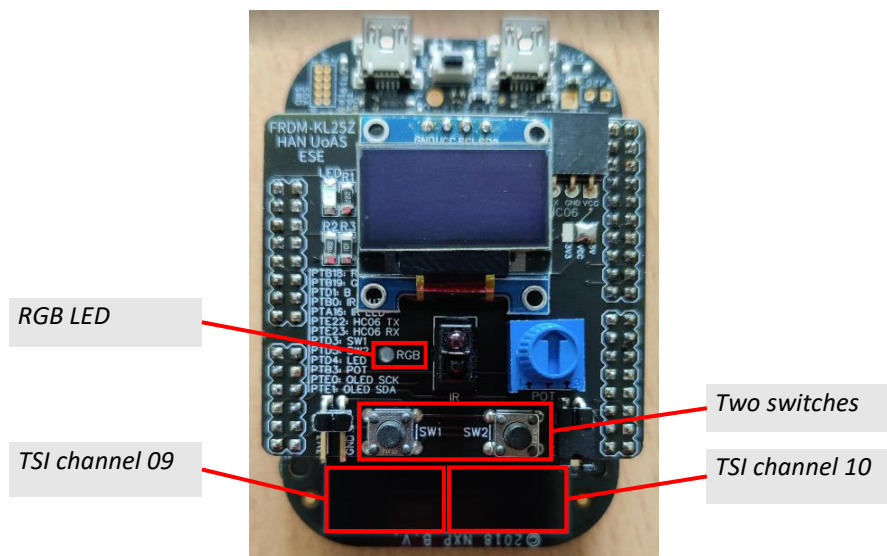
In this lab exercise the switches and Touch Sensing Input (TSI) module are both used for generating input commands. These commands are received by a command handler task that turns on LEDs depending on the command. All LEDs are turned off after a timeout. This behaviour will be implemented by using several tasks, a software timer and a queue.

Hardware

The hardware required for this project is outlined in the following table.

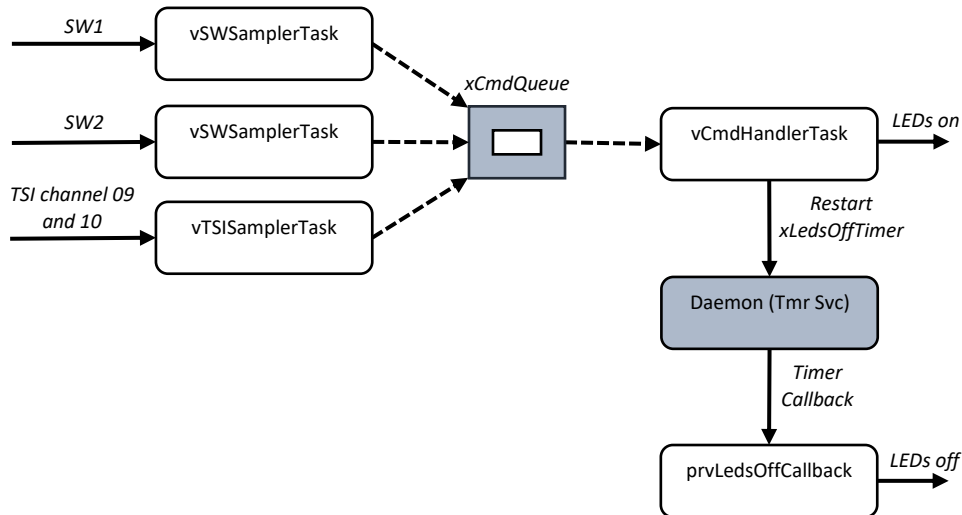
| Description | MKL25Z128VLK4 pins | Notes |
|-------------------------|--|---|
| RGB LED | PTB18 (red) PTB19 (green) PTD1 (blue) | - |
| Two switches | PTD3 (SW1) PTD5 (SW2) | Remember to enable the internal pullup resistors. |
| Capacitive Touch inputs | PTB16 (TSI0 Channel 09) PTB17 (TSI0 Channel 10) | - |

This hardware is available on the FRDM-KL25Z board and the oled shield.



Software

The software design is depicted in the following image. A description of the design elements follows next.



- The *vSWSamplerTask* tasks each sample one switch with a frequency of 5 Hz. If at the time of sampling the switch is pressed, the following commands are sent to the queue:
 - *SW1* sends DOWN commands
 - *SW2* sends UP commands
- The *vTSSamplerTask* task first samples TSI channel 09 and next 10 with a frequency of 5 Hz. If a touch was detected for one or both channels, the following commands are sent to the queue:
 - Channel 09 sends DOWN commands
 - Channel 10 sends UP commands
- The *xCmdQueue* queue can contain one data item. Such an item is of the following type:

```
typedef enum
{
    UP,
    DOWN,
}command_t;
```

- The *vCmdHandler* task waits for commands. It has a higher priority than the *vSWSamplerTask* and *vTSSamplerTask* tasks, so it will move immediately into Running state when commands are available in the queue.

As soon as a command is available, the *vCmdHandler* task will do the following:

- DOWN command: red LED on
- UP command: green LED on

After turning on an LED, the *xLedsOffTimer* is reset and the *vCmdHandler* task goes into Blocking state waiting for a new command.

- A one-shot timer called *xLedsOffTimer* is used to turn off the LEDs after a delay of 1.5 seconds. The *Daemon* task runs at the lowest priority. The one-shot timer is reset by the *vCmdHandler* task if another command is received within 1.5 second.

A template project is provided:

Week3 – Lab.zip

Configuring FreeRTOS

Configure FreeRTOS in the file FreeRTOSConfig.h to make sure the following configurations are applicable:

1. Enable the use of software timers.
2. The timer queue length is set to five.
3. The timer service task (Tmr Svc) priority is one.
4. The stack depth of the timer service task (Tmr Svc) is equal to the minimal stack size.
5. Set the timer service task (Tmr Svc) name to “Daemon”.

Answer

Command queue

Implement the queue xCmdQueue. In the main() function, make sure that creating the queue succeeded.

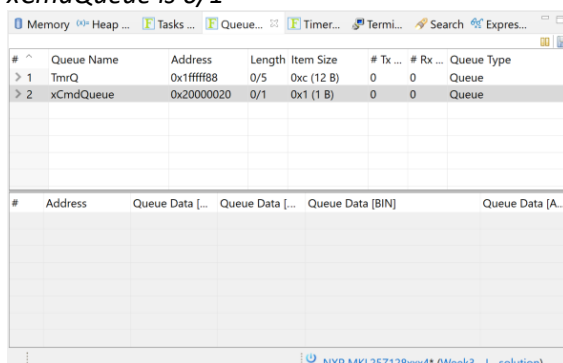
Answer

Switch sampler task

Implement the task vWSamplerTask. Create two instances of this one task in main(). The switch (SW1 or SW2) that must be sampled is passed as a parameter when the task is created.

Use the FreeRTOS kernel aware debugger in MCUXpresso to check and see if a command is sent to the queue as expected when one of the switches is pressed. To do this, set a breakpoint at the function xQueueSend() in the vWSamplerTask task, start the debugger, and press one of the switches. The program should halt at the breakpoint and the image on the left should be visible. Take one more step (F6) to send the data to the xCmdQueue. Now the image on the right should be visible.

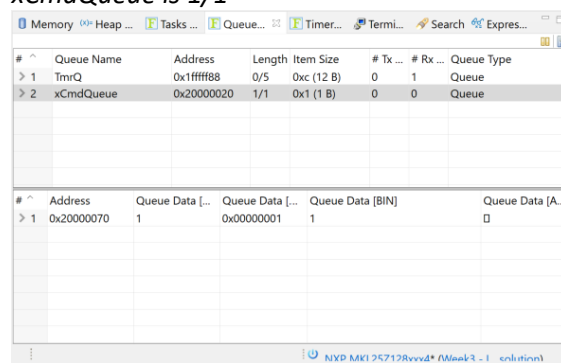
Before pressing SW1 the length of the xCmdQueue is 0/1



| # | Queue Name | Address | Length | Item Size | # Tx ... | # Rx ... | Queue Type |
|-----|------------|------------|--------|------------|----------|----------|------------|
| > 1 | TmrQ | 0x1fffff88 | 0/5 | 0xc (12 B) | 0 | 0 | Queue |
| > 2 | xCmdQueue | 0x20000020 | 0/1 | 0x1 (1 B) | 0 | 0 | Queue |

| # | Address | Queue Data [...] | Queue Data [...] | Queue Data [BIN] | Queue Data [A...] |
|-----|------------|------------------|------------------|------------------|-------------------|
| > 1 | 0x20000070 | 1 | 0x00000001 | 1 | |

After pressing SW1 the length of the xCmdQueue is 1/1



| # | Queue Name | Address | Length | Item Size | # Tx ... | # Rx ... | Queue Type |
|-----|------------|------------|--------|------------|----------|----------|------------|
| > 1 | TmrQ | 0x1fffff88 | 0/5 | 0xc (12 B) | 0 | 1 | Queue |
| > 2 | xCmdQueue | 0x20000020 | 1/1 | 0x1 (1 B) | 0 | 0 | Queue |

| # | Address | Queue Data [...] | Queue Data [...] | Queue Data [BIN] | Queue Data [A...] |
|-----|------------|------------------|------------------|------------------|-------------------|
| > 1 | 0x20000070 | 1 | 0x00000001 | 1 | |

Answers

TSI sampler task

The implementation of the vTSISamplerTask task is given. Carefully study this task. Two things are not yet implemented.

1. In main() the task must be created. Give this task the same priority that you gave to the vSWSamplerTask tasks.
2. Update the vTSISamplerTask so that when a touch is detected, the appropriate command is sent to the xCmdQueue queue.

Again, use the FreeRTOS Kernel Aware debugger to see if a command is sent to the queue when one of the capacitive pads is touched.

Answers

Command handler task

Implement the task vCmdHandler and create one instance of this task in main(). Make sure that the priority of this task is higher than the vSWSamplerTask and vTSISamplerTask tasks. The implementation can be verified by checking if the red LED turns on at a DOWN command and the green LED turns on at an UP command.

Answer

LEDs off timeout timer

Implement the one-shot software timer to turn off the LEDs after 1.5 seconds. Several things need to be done:

1. Create a local variable that holds the timer handle called xLedsOffTimer.
2. Create an instance of the timer in main(). Make sure to check if the creation succeeded.
3. Implement the callback function prvLedsOffCallback(). In this function, turn off both LEDs.
4. Reset the xLedsOffTimer in the task vCmdHandlerTask as soon as a command has been received.

Answers