

ELEC 5260/6260/6266 Basic Input/Output

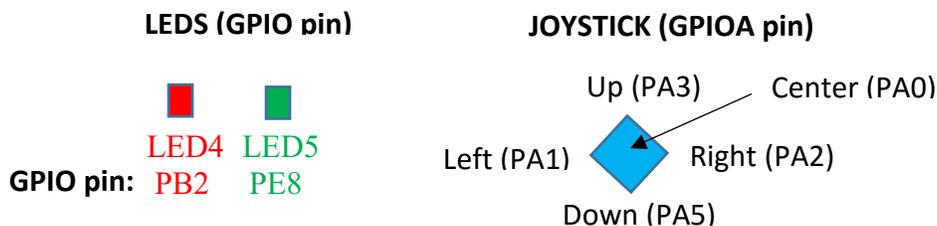
PART 1: Input/Output Concepts from Chapter 3

At the end of Chapter 3, answer/work the following, which deal with input/output operations:

- Q3-6 UML sequence diagram for a busy-wait device read
- Q3-8 UML sequence diagram for copying characters between devices
- Q3-13 UML sequence diagram for interrupt-driven device read
- Q3-17 UML sequence diagram for prioritized nested interrupts
- Q3-18 UML sequence diagram for prioritized nested interrupts

PART 2 – I/O Project (two programs)

To practice with program-controlled I/O, write two programs that create and display one of several patterns on the two User LEDs on the Discovery board, as selected by the Joystick button on the board.



The program is to operate as follows.

- Initially, all LEDs OFF.
- On the first press of a JOYSTICK button, the LEDs should turn ON and OFF with the indicated pattern, changing to the next step of the pattern every one-half second, with the pattern repeating continuously until the next button press.

JOYSTICK Up:

1. LED4 ON
2. LED4 OFF

JOYSTICK Down:

1. LED5 ON
2. LED5 OFF

JOYSTICK Left:

1. LED5 ON
2. LED4 LED5 ON
3. Both OFF

JOYSTICK Right:

1. LED4 ON
2. LED4 LED5 ON
3. Both OFF

JOYSTICK Center:

Both LEDs turn OFF and remain OFF.

PROGRAM 1

The program is to contain the following six C program modules:

1. An *output handler*, which writes patterns to the LEDs.
2. An *input handler*, which tests the JOYSTICK and returns some value that indicates which, if any, JOYSTICK button is pressed.
3. A *system tick timer interrupt handler*, which is activated every one-half second. This routine should call the output handler, if the LEDs are to be changed.
4. A *main program*, which executes in a continuous loop, calling the input handler every time through the loop.
5. The “startup” and “system initialization” files for the STM32L476G-Discovery board (inserted if you select “startup” under “device” in the “Manage Run-Time Environment” window when you create the project.)
6. The STM32L476xx microcontroller header file, `stm32l476xx.h`. (At top of main program, use `#include “stm32l476xx.h”`)

See program testing and submission below.

PROGRAM 2

Repeat PROGRAM 1, but instead of using your own output and input handlers, use JOYSTICK and LED functions from the Board Support Package (BSP) driver *stm32l476g_discovery.c*. This file can be added and/or copied to your project from your Keil installation directory (which is the following on my computer):

C:/Keil_v5/ARM/PACK/Keil/STM32L4xx_DFP/2.0.0/Drivers/BSP/STM32L476G-Discovery/

Functions in *stm32l476g_discovery.c* utilize Hardware Abstraction Layer (HAL) drivers for GPIO pins and related modules of the microcontroller. These need to be loaded, in addition to the startup file, when you create the project. If you do not add them at project creation time, you may do so from the “Manage Run-Time Environment” tool. HAL drivers for all peripheral modules are located under “Device > STM32Cube HAL”. Note that if you select the HAL driver for GPIO, one or more boxes will be colored orange and there will be messages at the bottom of the window telling you to add additional HAL drivers. Doing so will change the orange boxes to green.

Testing and Submitting the Programs

Each project is to be programmed into the flash memory of the board, so that the program can be demonstrated without being connected to the Keil debugger.

Print and submit the source program, *and also email it to me*.

Instead of printing debug windows, either create a narrated video demonstrating the operation of your program running on your board, or bring your programmed board to my office and demonstrate the program’s operation to me.