CSSE3010 Stage 1

General Purpose Input/Output and Signal Synchronisation

(Closed Shoes MUST BE Worn in the labs)

1 Assessment

- Git due 4pm Monday in week 3 .
- Demo and Workook due in the first hour of your lab session in week 3 .
- Course Marks: 10%
- Electronic Course Profile Pass Hurdle: Must be submitted.

2 Resources

- Nucleo-F429ZI platform
- ITA1000G LED Light Bar
- Xenon Monitor

3 Structure

Your final stage demo code must be titled `main.c` and saved in your `stage1` folder.
PATH: `csse3010/stages/stage1`

Listed below are the required mylib libraries, that you will need to develop for this stage. They should be saved in your `mylib` folder, in your repository.
PATH: `csse3010/mylib`.

- LTA1000G LEDBar HAL Library
- Input Signal Synchroniser (ISS) HAL Library
- Pushbutton HAL Library

4 Introduction

This stage will introduce the General Purpose Input/Output (GPIO) and signal synchronising Hardware Abstract Layer (HAL) libraries for controlling the input/output pins of the Nucleo Development Board. You will use the LTA1000G LED light-bar, the Nucleo's onboard pushbutton and the xenon monitor. The `debug_printf` library will also be introduced.

4.1 STM32F4 Hardware Abstract Layer (HAL) Driver Library

The HAL driver library is a collection of C library files for accessing and controlling the various peripheral features of the STM32F429ZI processor. The STM32F429ZI is a Cortex M4F 32-bit processor with 192Kbytes RAM and 1Mbyte of Flash memory. The STM32F429ZI peripherals include GPIO, Timers, I2C, SPI, USART, ADC, USB, DMA and Ethernet peripherals. The HAL files are located in the `sourcelib/components` folder. Example programs for using each peripheral are located in the examples folder.

5 PART A Tasks - LedBar

5.1 Workbook Tasks

Follow the workbook guidelines on Blackboard. You need to use the hardware and flow chart templates. State diagrams are not used for this stage. Workbook tasks are due in your week 3 stage session.

NOTE: If your design is modified, all relevant workbook diagrams should be also be updated to reflect the changes. You are also required to draw conclusions of the result of your stage implementation for your week 3 session.

5.2 Worksheet Tasks

The worksheet contains helpful questions related to this stage. You must complete worksheet 1 before attempting this part of the stage. It must be printed out and pasted into your workbook. The worksheet task is designed to help you complete stage 1 .

5.3 Design Tasks

Design tasks are due in the first hour of your week 3 stage session.

5.3.1 mylib Setup

You MUST FOLLOW the Template Code given in the `sourcelib/examples/templates/mylib` folder. Your mylib code must meet the guidelines specified in the mylib and platform build guides, on Blackboard. You MUST create the right file structure in the `mylib` git folder.

You will create mylib HAL library files for controlling the LED bar and the synchronising the outputs from the xenon monitor periodic signal and pushbutton. Refer to the LTA1000G LED bar, ISS and pushbutton mylib specifications, on Blackboard.

5.3.2 Design Task 1A: Moving Pattern Display (MPD)

You MUST copy and modify the template code given in the `sourcelib/examples/template/hal` folder. You MUST NOT modify any of the other examples. You MUST create the right file structure in the `stages/stage1` git folder.

a) Moving Pattern

Create a Moving Pattern Display (MPD). The pattern 0x03 should be displayed as moving from left to right, as seen in Figures 1 and 2. The MPD should keep repeating, once the pattern has reached the end of the bar. The display should update each time a rising edge event is detected on the periodic signal output from the xenon monitor (see Part B tasks and use the event counter to determine if a rising edge event has occured). Use the bit shift operators (right shift: '≫' and left shift: '≪') and HAL delay function. DO NOT use any of the in-built HAL timer modules. *Refer to `examples/getting started/blink`.*

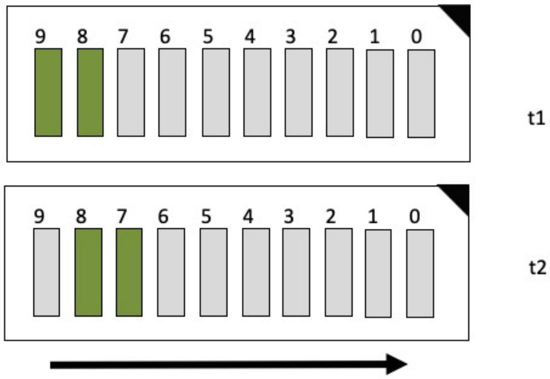


Figure 1: MPD moving from left to right

b) Console Display

Display the moving pattern value in hex each time it updates using the serial console (e.g. 0x0003, 0x0006, etc) (*hint*: use `debug_printf()`).

Also display the event counter value for the ISS and pushbutton sources, when an event occurs (e.g. ISS:01 — PB:01). The event counter values should reset to 0, after 10 counts.

c) LED Board Display

The MPD output should be displayed using the LED light-bar (*hint*: use 'bit masking' to check if a bit is set, e.g. to check bit5 of 0x33, use "0x33 & 0x20"). Connect the LED light-bar to digital pins D16 – D25 on the NUCLEO.

Fill in the LTA1000G mylib functions (See LTA1000G mylib specification): `sxxxxx_hal_lta1000g_init`, `sxxxxxx_hal_lta1000g_write` and `lta1000g_seg_set`.

Refer to ***getting started/blink*** *and* ***gpio/gpio*** *examples.*

5.3.3 Design Task 2A: Moving Pattern Display Direction Control

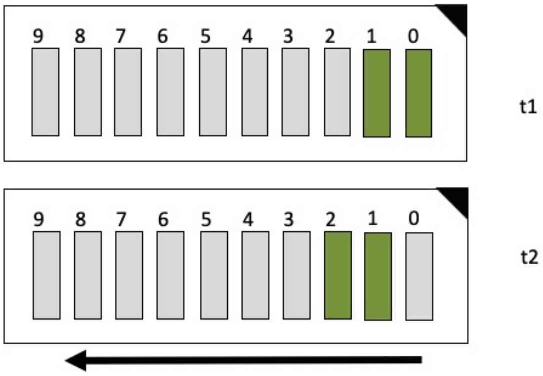


Figure 2: MPD moving from right to left

Control the direction of the MPD using the onboard pushbutton (See Part B tasks). For example, if the current display direction is from left to right, then when the onboard pushbutton is pressed, the direction should change from right to left (See Fig 2). You must use the pushbutton's ISS event counter to determine if a pushbutton event has occured.

Refer to ***getting started/blink****,* ***gpio/gpio*** *and* ***gpio/gpio_interrupt*** *examples.*

6 PART B Tasks - Input Signal Synchronisation

6.1 Workbook Tasks

Follow the workbook guidelines on Blackboard. You need to use the hardware and flow chart templates. State diagrams are not used for this stage. Workbook tasks are due in your week 3 stage session.

NOTE: If your design is modified, all relevant workbook diagrams should be also be updated to reflect the changes. You are also required to draw conclusions of the result of your stage implementation for your week 3 session.

6.2 Design Tasks

Design tasks are due in the first hour of your week 3 stage session. You should expand the code that you developed, previously in part A.

6.2.1 mylib Setup

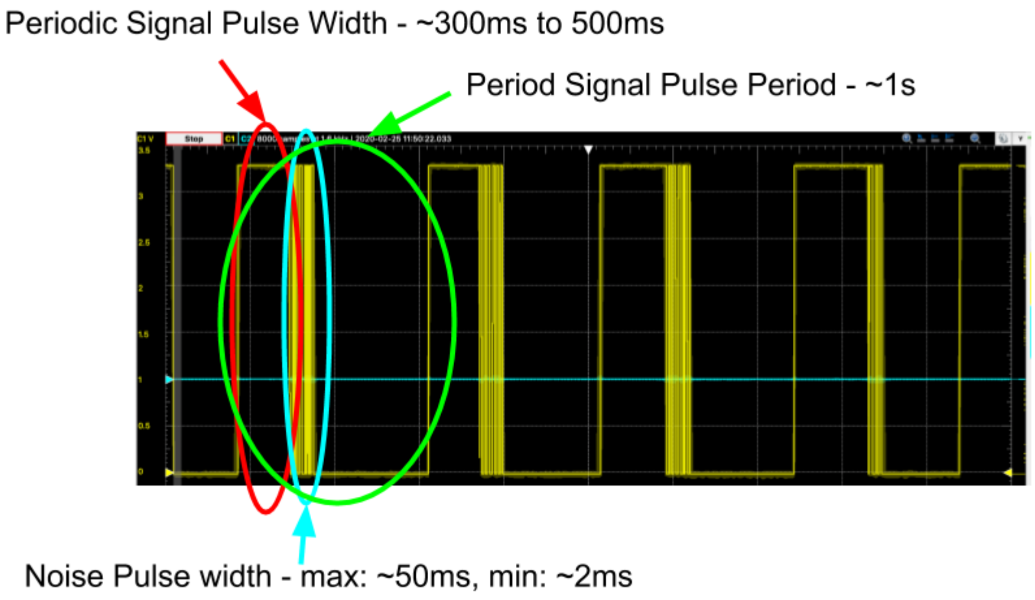
You MUST FOLLOW the Template Code given in the `sourcelib/examples/templates/mylib` folder. Your mylib code must meet the guidelines specified in the mylib and platform build guides, on Blackboard. You MUST create the right file structure in the `mylib` git folder.

You will create mylib HAL library files for Input Signal Synchroniser (ISS) and the pushbutton. Refer to the ISS and pushbutton mylib HAL specifications on Blackboard. Assign the following source indexes as ISS sources. Table 1 lists the sources for the ISS.

Table 1: ISS Sources

ISS Index	Source
1	Nucleo A0 pin connected to Periodic Signal from Xenon Monitor
2	Nucleo Onboard pushbutton

6.2.2 Xenon Monitor



Periodic Signal Pulse Width - ~300ms to 500ms

Period Signal Pulse Period - ~1s

Noise Pulse width - max: ~50ms, min: ~2ms

Figure 3: Xenon Monitor Periodic Signal output

The Xenon Monitor will output a periodic signal with a time period between $800ms$ and $1s$. The Xenon Monitor periodic signal is affected by noise, as seen in Figure 3. You must use a synchronising algorithm to correctly sample the Xenon Monitor. Connect the ISS signal from pin A4 of the Xenon Monitor, to pin A0 on the Nucleo board.

Refer to the Xenon Monitor Guide on BlackBoard

7 Callback Graph

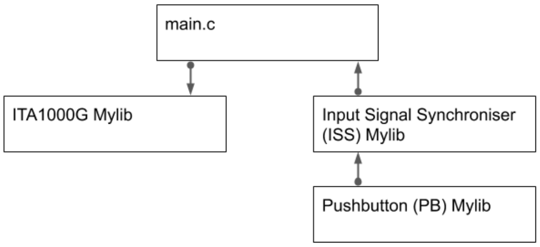


Figure 4: Callback Graph (NOT a flowchart)

Figure 4 is a callback graph for this stage. The callback graph shows how each file/function/library is called by other files/functions/libraries. It is not a flowchart and is only for your reference.

8 Criterion

The stage demonstrations are marked according to the criterion outlined in the table below. If you fail to demonstrate sufficient understanding and functionality in the specified marking time you will not be allowed to repeat the stage. You must pass the pre-demo checks before you are allowed to demo. **All code assessed for the stage must be your own work.**

8.0.1 Pre Demo Checks

The following criteria **must** be met **before** you are allowed to demo.

Check	**P/F**
Your latest stage and mylib code **must** be in git by Monday 4pm in Week 3 .	
Your git repository must be up to date with the latest version of sourcelib.	
Your stage code must build without errors.	
Your mylib and top comments are correctly filled out.	

Failure to meet pre-demo checks will mean that you are not allowed to demo the stage.

8.0.2 Demo Criteria

Note: You may be asked to make minor modifications to your code by the assessor, during your demo, which must be passed.

Design Task 1A: Moving Pattern Display (MPD)	
0	Moving pattern generator is not implemented.
1	The pattern is displayed by console **and** LED bar output, it may be incorrect but it updates by the xenon monitor periodic signal.
2	The pattern is correctly displayed by console **and** LED bar output, but looping is **not** used to set segment values, and it updates by the xenon monitor periodic signal.
3	The pattern is correctly displayed by console **and** LED bar output, and looping is used to set segment values and it updates by the xenon monitor periodic signal.
Design Task 2A: Moving Pattern Display Direction Control	
0	Nothing occurs when the pushbutton is pressed.
1	Pushbutton presses are registered with an interrupt and debounced (not with ISS or PB), and there is some functionality to demonstrate this.
2	Pushbutton press changes the direction but fails to meet one or more ISS or PB mylib specifications.
3	Pushbutton press changes the direction and meets all ISS and PB mylib specifications in section.
Design Task B: Input Signal Synchroniser	
0	Nothing occurs when the ISS xenon monitor periodic and pushbutton signals change.
1	The event counter for the ISS xenon monitor periodic and pushbutton signals changes but incorrectly, when an event occurs.
2	The event counter for the ISS xenon monitor periodic and pushbutton signals changes correctly but is unable to be reset.
3	The event counter for the ISS xenon monitor periodic and pushbutton signals changes correctly and meets all specifications for the ISS mylib driver.

Code Style	
0	One or more style errors found in one tutor selected stage file.
1	No style errors found in **one** tutor selected stage file.
Mylib	
0	Required Mylib files are not correctly implemented (functions).
1	Required Mylib files are only partially (correctly) implemented (functions) or have no top comment.
2	All Mylib files are correctly implemented (functions and top comment).
Workbook Tasks	
0	Workbook is missing sections that specified in the workbook guidelines.
1	Workbook is meets all specifications (all sections are present).
Hardware Schematic	
0	Hardware schematic diagram has not been completed.
1	Hardware schematic diagram is complete.
Flow Charts	
0	Flow Charts have not been attempted.
1	Flow Charts have been attempted but do not meet the required specifications or have errors.
2	Flow Charts meet the required specifications and have no errors.

Student Name:

Student Number	**Mark (/16)**	**Marker**	**Date**