

CSSE3010 Stage 3

Hamming Encoder/Decoder with Keypad Finite State Machine Controller

(Closed Shoes MUST BE Worn in the labs)

1 Assessment

- Git due 4pm Monday in week 7 .
- There is no demo. Do not attend a lab session. All lab sessions are cancelled
- No worksheet or workbook is due.
- Course Marks: 10%
- Electronic Course Profile Pass Hurdle: Must be submitted..

2 Resources

- Nucleo-F429ZI platform
- PMOD Keypad

3 Structure

Your final stage demo code must be titled `main.c` and saved in your `stage3` folder.

PATH: `csse3010/stage/stage3`

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

PATH: `csse3010/mylib/mylib_library_folder`.

- Alive Timer HAL Library (optional)
- Keypad HAL Library
- Hamming LIB Library

4 Introduction

Finite State Machines (FSM) controllers are a fundamental part of digital design. FSMs are used for a variety of applications that require operations to be controlled in a sequential manner. Complex designs are often partitioned into several FSMs. The stage will introduce you to using FSM controllers as a cyclic executive.

5 PART A Tasks - Keypad Finite State Machine Controller

5.1 Workbook Tasks - Not Assessed

The workbook tasks are optional and are not assessed. It is common and good practice to keep a workbook, with a schematic and firmware notes, for future reference.

5.2 Design Tasks - Do Not Attend Labs

The code required for the design tasks must be uploaded to your git repository by the specified due date in week 7. The lab sessions are cancelled. Alternate online sessions will be organised on BlackBoard.

5.2.1 Git Requirements

You MUST have your stage and mylib code in git, in the correct folders and it must be able to be compiled by the markers, in order to be assessed for this stage. If your code only compiles on your computer and cannot be compiled by the markers, then it is not acceptable for marking. Please ensure that you do the following:

- Folder names - Your stage and mylib code MUST be in the correct folder, as specified. Folder name must be alphanumeric characters and contain NO spaces e.g. stage1 NOT stage 1 or stage-1.
- Header file includes - MUST NOT have a file path. e.g. `include "sxxxxxxx_hal_joystick.h"` is correct BUT NOT:
`include "/home/csse3010/mylib/joystick/sxxxxxxx_hal_joystick.h"`. To include a file path for header files, modify the CFLAGS in filelist.mk.
- filelist.mk - your code MUST compile with a filelist.mk file. Do not rename filelist.mk.
- Code not compiled or not used - do not push to git, any files containing code that is not used or that will not compile. If you want to push code that is unfinished, then use compiler directive `#IFDEF _DONOTBUILD`, to not compile unused code.
- Gitlab - Check your code is correctly in your git repository by logging into gitlab: <https://csse3010.uqcloud.net/csse3010/>. Check the stages and mylib folders. If your code is not in the correct place, then it will not be compiled correctly.

- **.gitignore** - **you MUST HAVE A .gitignore file** in the stages, mylib and project folders. Do not put a .gitignore file into a subfolder e.g. stage1 - See https://csse3010.uqcloud.net/csse3010/docs/setup/git/git/#task_5_creating_a_gitignore_file
- **gitg** - use gitg (type gitg in terminal) to check which files are staged and can be pushed to git.

For more git details, see the user guide - <https://csse3010.uqcloud.net/csse3010/docs/setup/git/git/>

5.2.2 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 keypad FSM controller. Refer to the keypad HAL specifications, on Blackboard.

5.2.3 Design Task 1A: Keypad FSM Controller

Design a keypad FSM controller for interfacing and decoding keys pressed using the keypad from the PMOD Keypad. Your keypad FSM controller must use row scanning. No loops (e.g. `for`, `while`, etc) must be used, in the mylib driver. Note the cyclic executive in the `main.c` file should still be implemented with a while loop. A partially completed keypad controller state diagram can be seen in Figure 1. You will need to complete and implement it. You only have to detect a single key press on a column and not multiple key presses on a column.

Use the GPIO pins in Table 5.2.3

GPIO	Keypad Signal
D2	Column 4
D3	Column 3
D4	Column 2
D5	Column 1
D7	Row 4
D8	Row 3
D9	Row 2
A4	Row 1

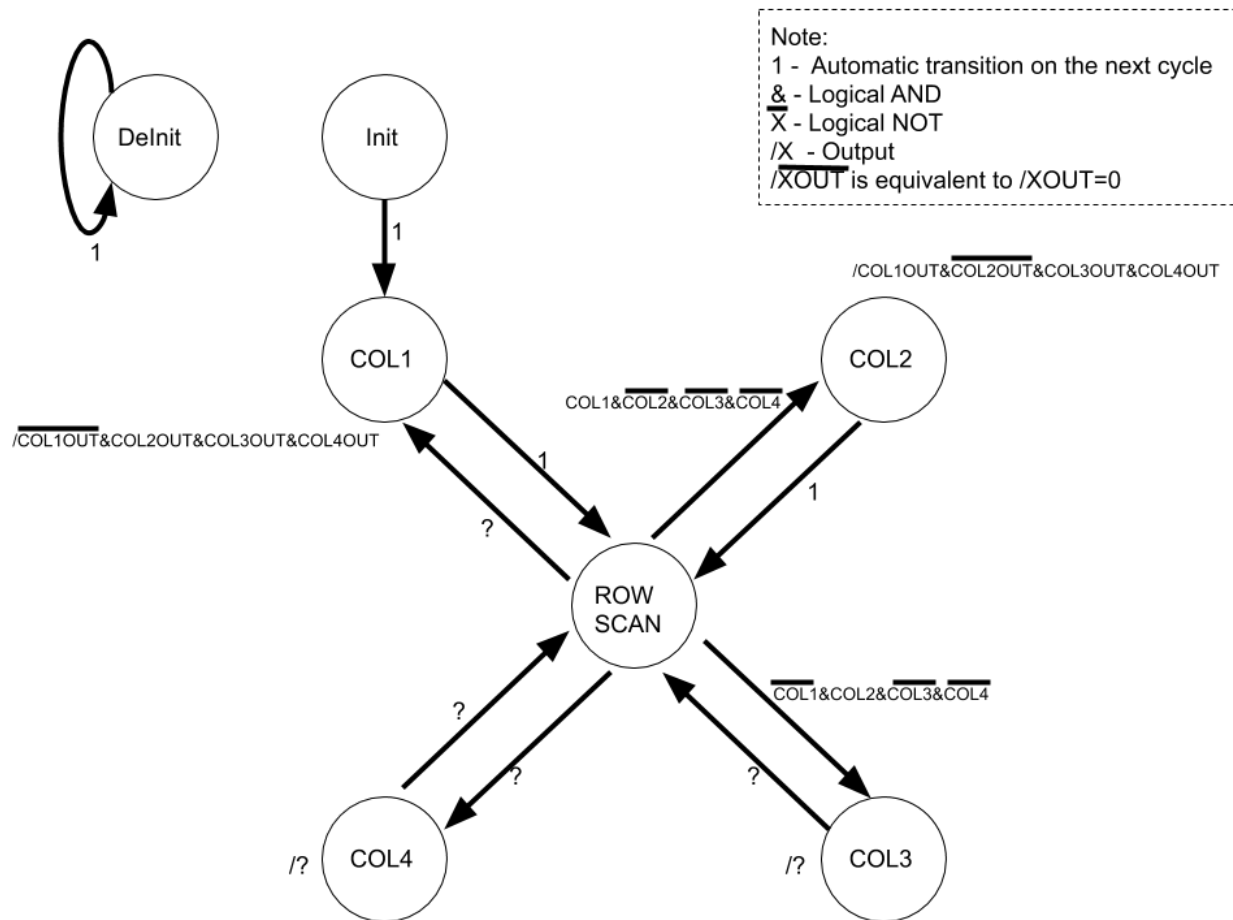


Figure 1: Keypad Row Scanning FSM (partially completed)

5.2.4 Design Task 2A: Console Display

The keypad FSM processing function should be periodically called by your main.c's cyclic executive. Use and configure either the atimer or HAL_GetTick() to ensure that keypad FSM processing function is called, every 50ms. No HAL_Delay or any delay functions are to be used. Once a key has been detected, it should be displayed to the console as the following:

<ASCII value of First key Pressed><ASCII value of Second Key Pressed>

6 PART B Tasks - Hamming Encoder and Decoder

6.1 Worksheet Tasks - Not Assessed

The worksheet contains helpful questions related to this stage. Worksheet is not assessed and is not compulsory for you to attempt. The worksheet task is designed to help you complete stage 3 .

6.2 Design Tasks - Do Not Attend Labs

The code required for the design tasks must be uploaded to your git repository by the specified due date in week 7 . The lab sessions are cancelled. Alternate online sessions will be organised on BlackBoard.

6.2.1 Git Requirements

You MUST have your stage and mylib code in git, in the correct folders and it must be able to be compiled by the markers, in order to be assessed for this stage. If your code only compiles on your computer and cannot be compiled by the markers, then it is not acceptable for marking. Please ensure that you follow the previously given git requirements.

6.2.2 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 LIB library files for Hamming encoding and decoding. Refer to the Hamming LIB specifications, on Blackboard.

6.2.3 Design Task 1B: Hamming Encode

Implement the hamming byte and half byte encode functions. Use the generator and parity matrix given in the `getting_started/hamming` example.

Refer to `getting_started/hamming` example.

6.2.4 Design Task 2B: Hamming Decode

Implement the hamming byte decode and error return functions.

6.2.5 Design Task 3B: Cyclic Executive FSM

Implement the FSM shown in Figure 2. Figure 2 is a partially completed FSM, which you will have to complete. Use an appropriate FSM structure and state definitions, in your cyclic executive. No `HAL_Delay` or any delay functions are to be used.

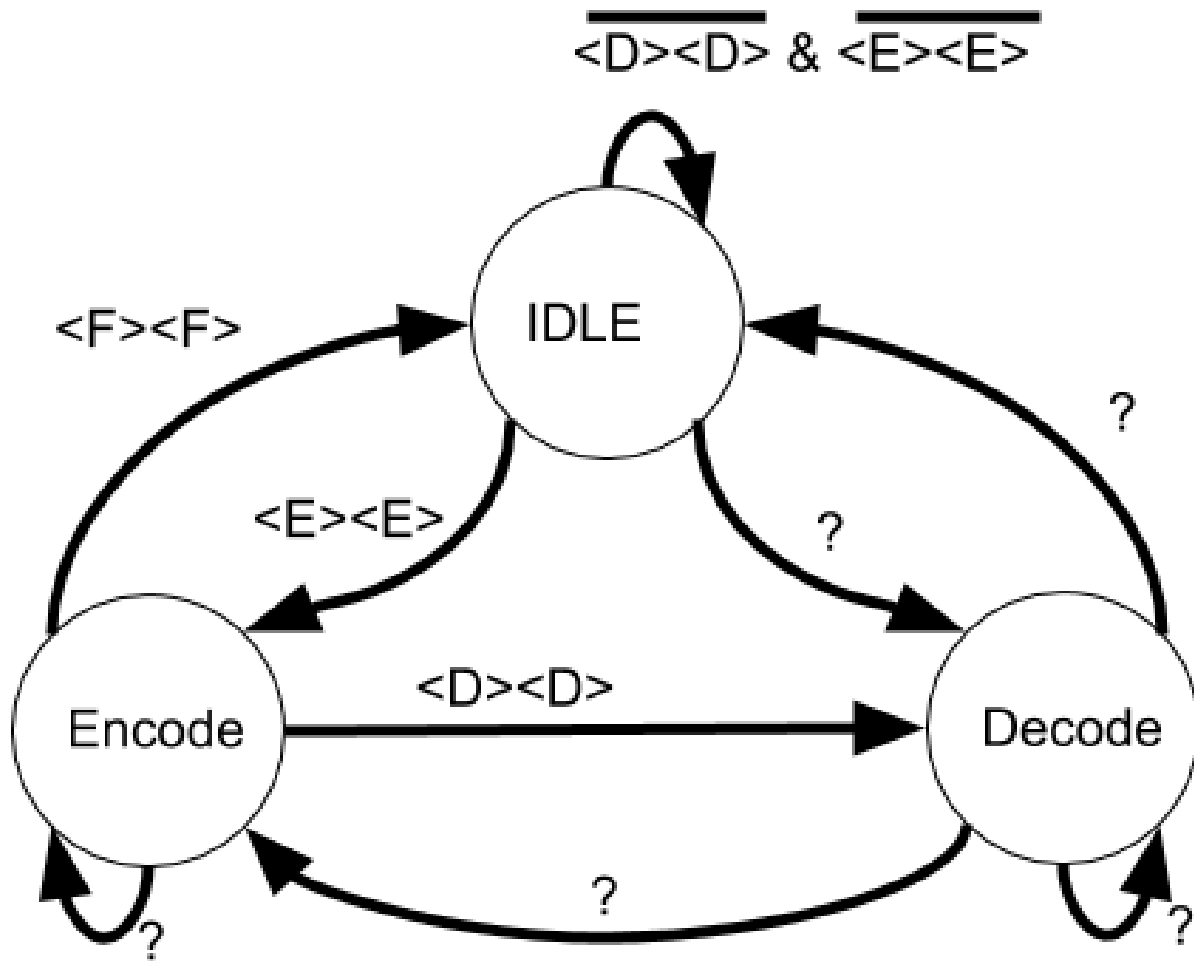


Figure 2: Cyclic Executive State Diagram (Partially Completed)

Example Cases:

- Encode - Press <E> <E> <4><5> will encode the value 0x45. Display the hex output on a new line: "HE 4x5x". Note: 'x' is the Hamming bits.
- Decode - Press <D> <D> <3><4> will decode the value 0x34. Display the hex output on a new line: "HD x Py". Where y is the parity error check (1 for error, 0 for no error).
- Return to Idle - Press <F> <F>.

7 Final Demo

Implement all design tasks as one demo.

8 Criterion

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

8.0.1 Pre Marking Checks

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

Check	P/F
Your latest stage and mylib code must be in git.	
Your git repository must be up to date with the latest version of sourcelib.	
Your git repository conforms to the git guidelines, provided.	
Your stage code must build without errors (including with the automated build process).	
Your mylib and top comments are correctly filled out.	

Failure to meet pre-marking checks will mean that your stage is not marked.

8.0.2 Marking Criterion

You must be able to combine all design tasks, into the same file and demo all design tasks, without reprogramming your Nucleo. Note: Your code must pass the Pre-marking checks or it will not be marked.

Design Task A: Keypad FSM	
0	No output from the keypad is observed.
1	Some output from the keypad is observed but is incorrect or the output is viewed from 2 or less rows or 2 or less columns.
2	Correct output observed but from 3 or less rows or 3 or less columns.
3	Correct output observed from all keypad rows/columns but occasional errors occur.
4	Correct output from the keypad is observed but with no errors.
Design Task 1B & 2B: Hamming Encode/Decode	
0	Hamming encode or decode is not implemented or does not function.
1	Hamming encoder or decoder is implemented but does not work correctly).
2	Hamming encoder works and Hamming decoder will decode but not correct 1bit errors.
3	Hamming encoder works and Hamming decoder will decode, corrects 1bit errors in data bits only.
4	Hamming encoder works and Hamming decoder will decode, correct 1bit errors in data and hamming parity bits.
Design Task 3B: Cyclic Executive FSM	
0	Cyclic Executive for keypad and hamming is not correctly implemented with an FSM or HAL_delays are used.
1	Cyclic Executive for keypad and hamming is only partially implemented with an FSM or function with errors.
2	Cyclic Executive for keypad and hamming is implemented correctly with an FSM and functions correctly.

Mylib Keypad HAL Implementation	
0	Keypad HAL is not implemented at all or does not function or is not implemented with an FSM structure or uses loops (e.g. while or for).
1	Keypad HAL is only partially implemented with an FSM structure.
2	Keypad HAL is fully implemented with an FSM and functions correctly.
Mylib Hamming LIB Implementation	
0	Hamming encode or decode functions are not implemented.
1	Both Hamming encode and decode functions are implemented but do not function correctly.
2	Both Hamming encode and decode functions are implemented and function correctly.
Mylib Configuration	
0	No configuration is not implemented (no sxxxxxx_mylib_configure.h).
1	Configuration is implemented correctly.

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.

Student Name:

Student Number	Mark (/16)	Marker	Date