Emulator Developer Guide

UNSW DESN200 23T2 (COMP)

Group: "Group H"

Members:

Siyu Qiu (z5348946)
Muzhi Wang (z5340914)
Hanyang Xu (z5254111)
Shaoqiu Lyu (z5325799)

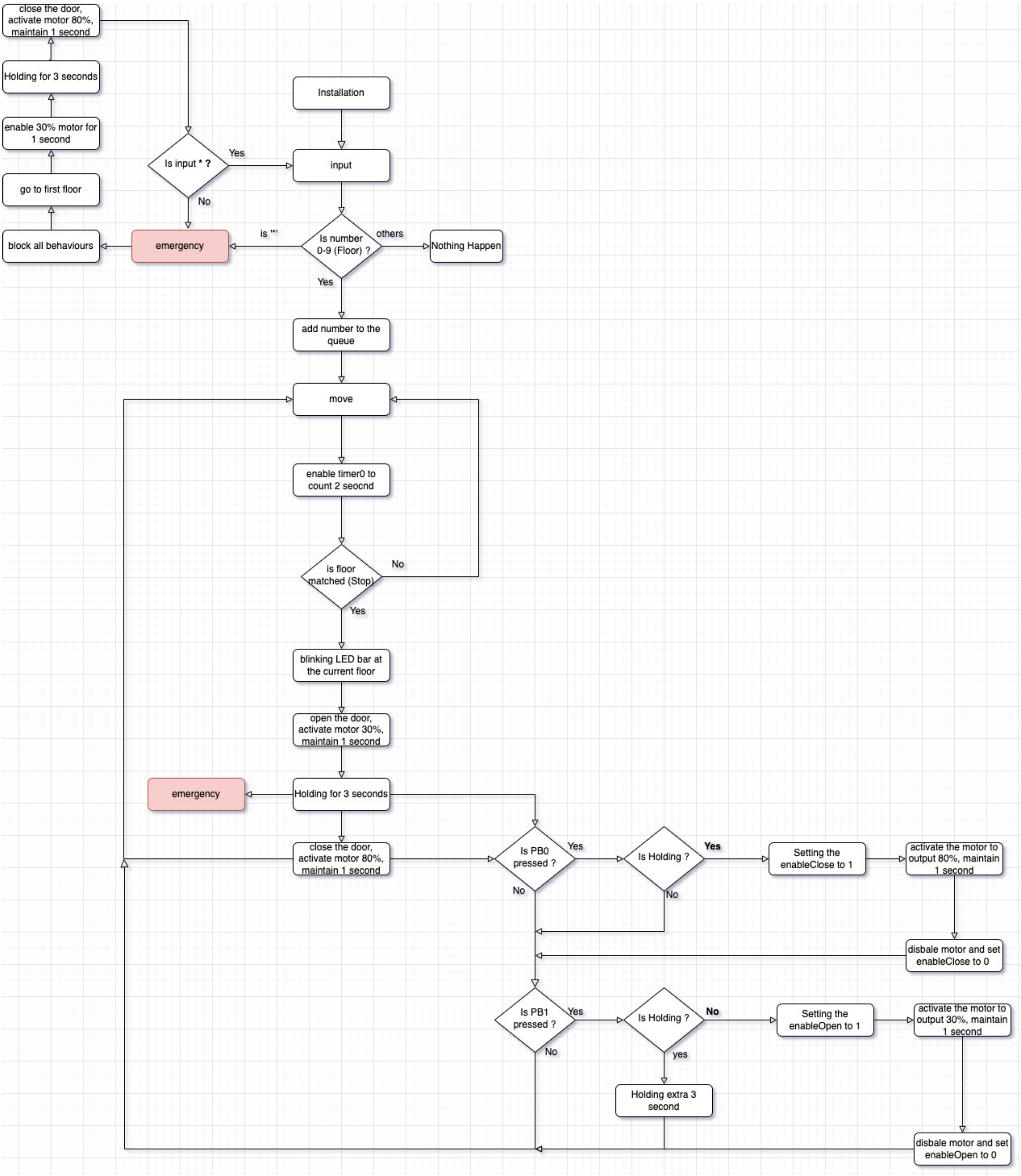
Table of Contents

Introduction

This is the DESN2000 Lift Emulator system which is developed based on board Arduino Mega. The microcontroller in this board uses Microchip Atmega2560 with 8-bit AVR chip.

Code Structure

1. The flow chat for our basic project (bonus part not included) as below:


close the door, activate motor 80%, maintain 1 second
Holding for 3 seconds
enable 30% motor for 1 second
go to first floor
block all behaviours
Installation
Is input * ?
Yes
No
input
emergency
is '*'
Is number 0-9 (Floor) ?
others
Nothing Happen
Yes
add number to the queue
move
enable timer0 to count 2 seocnd
is floor matched (Stop)
No
Yes
blinking LED bar at the current floor
open the door, activate motor 30%, maintain 1 second
emergency
Holding for 3 seconds
close the door, activate motor 80%, maintain 1 second
Is PB0 pressed ?
Yes
No
Is Holding ?
Yes
No
Setting the enableClose to 1
activate the motor to output 80%, maintain 1 second
disbale motor and set enableClose to 0
Is PB1 pressed ?
Yes
No
Is Holding ?
No
yes
Setting the enableOpen to 1
activate the motor to output 30%, maintain 1 second
Holding extra 3 second
disbale motor and set enableOpen to 0

2. Module Information

Module	Function	Input/Output
Timer Interrupt 0	is used to calculate the 2 seconds	count the time that the elevator to move from the current floor to the next floor
Push Button Interrupt 0 (INT0)	is used to close the door	Press PB0 to start interrupt and set closeenable to 1 Start the motor output 80% PWM to close the elevator door

Module	Function	Input/Output
Push Button Interrupt 1 (INT1)	is used to open the door	Press PB1 to start interrupt and set openenable to 1 Start the motor output 30% PWM to open the elevator door
LCD	LCD were used to display and call functions that would convert and display data onto the LCD screen	Register or Characters. The output value displayed on the LCD
LED	LED will display the current floor, the direction of the movement and the stop stage (by blinking)	the currentFloor after pattern changed displayed on the LED
Motor	motor will move in different speed to indicate opening / closing the door	will be activied in different speed depends on different push buttons
Keypad	read the input from keypad	0 - 9, A - D, *, #
Strobe light	will blink when the emergency mode activated	Blink

3. design states:

   Here are several state for our project:

- Moving : indicate whether the elevator is moving.
- Open: indicate whether the motor is activated to open the door.
- Holding: indicate the whether is waiting (when elevator stops at a floor the 3 seconds consumes between open the door and close the door)
- Close: indicate whether the motor is activated to close the door.
- emergency: indicate whether the emergency state is activated.
- long-time holding: indicate whether the long-time holding state is activated.

Component Mapping

1. The address are defined as below:

Name	Address	useful
PORTLDIR	0xF0	PL0-PL3 for input, PL4-PL7 for output
INITCOLMASK	0xEF	To go through from the left column
INITROWMASK	0x01	To go through from the top row
ROWMASK	0x0F	Get input from Port L
LCD_RS	7	Different pins for LCD
LCD_E	6	Different pins for LCD
LCD_RW	5	Different pins for LCD

Name	Address	useful
LCD_BE	4	Different pins for LCD

2. The data segments are defined as below:

Name	Bytes	useful
SecondCounter	1	indicate if the elevator moves for 2 seconds
TempCounter	2	indicate the prescaler inside the timer0
UpdateInd	1	indicate is there a new input from the keypad
closeEnable	1	closeEnable equal to 1 means enable close, otherwise 0 is disable
openEnable	1	openEnable equal to 1 means enable open, otherwise 0 is disable
specialState	1	indicate to the keyboard input "*" -> emergency state
holdingState	1	indicate to the keypad input "#" -> long time holding
emergencyLight	1	indicate the light should on/off when the elevator in the emergency state
RequestQueue	10	the maximum number is 10 in the queue

3. The registers are defined as below, and shown the useful for registers as well(most of registers are different meaning in different functions):

For the whole project:

Register	Name	useful
R16	currentFloor	Displays the number of floors where the elevator currently stops
R17	row	the row number for keypad
R18	col	the column number for keypad
R19	direction	display the lift is currently going up or down. (1 for up, 0 for down)
R20	temp	Temporary register
R21	haveUpdate	
R22	mask	
R23	temp2	Temporary register
R30	ZL	Pointer that used for low 8 bits of the address of the request queue
R31	ZH	Pointer that used for high 8 bits of the address of the request queue

For pattern_shift:

Register	Name	useful

Register	Name	useful
R16	iterator	check how many tims for loop (jump out if equal to R18)
R17	pattern	pattern that need to be shifted
R18	times	time need to be shifted
R24	returnValue	The value we want to return(receive the paattern to display how many bars on the LED)

For pattern_display:

Register	Name	useful
R17	pattern	pattern that need to be shifted
R18	times	time need to be shifted
R24	returnValue	The value we want to return(receive the paattern to display how many bars on the LED)

For match_modify_queue:

Register	Name	useful
R17	loopCounter	compare go through this loop 9 times or not
R18		Temporary register
R28	YL	Pointer that used for read the next address of the request queue
R29	YH	Pointer that used for read the next address of the request queue
R30	ZL	Pointer that used for store of the request queue
R31	ZH	Pointer that used for store of the request queue

Key Functions

Here are some important functions for our project:

Function name	useful
direction_add_regulate	function to add the floor with the direction and regulate the floor number and will change the direction if edge cases reaches
direction_add	will be called in function `direction_add_regulate` to add the floor with the direction: 1 for + (up), 0 for - (down)
pattern_shift	will be called in function `pattern_display` for shift the given number for given times
pattern_display	function is for displaying current floor with the given number on the LED (need to call `pattern_shift`)

Function name	useful
OneSec	function is for busy wait for a second(still need to check if there is input for keypad during this one second or not)
compare_floor_stop	function is for checking the current level in the list for stopping or not. Return R24 to check the current floor is in the RequestQueue or not -> 1 for match, 0 for not
match_modify_queue	function is for remove the first entry from the queue and up others by 1, filled by 0 at the last
initial_request_queue	function is for initialize the request queue by setting the 10 entries to 0
keypad_main	function is for push the used registers into the stack, and the input from the keypad will be returned in R24
keypad_input_to_request	function is for adding the given input number into the queue
insert_order_queue	This function read the input from the keypad and check validation then insert to the request queue
LCD_display	this function print the ordinary information on the LCD
emergency_lcd	this function print the emergency information on the LCD

Interrupts

Interrupt name	useful
INT0	This interrupt use to set the state closeEnable to 1 (activate close door) when the PB0 is pressed
INT1	This interrupt use to set the state openEnable to 1 (activate open door) when the PB1 is pressed
Timer0	This interrupt use to count 2 seocnds for elevator moving between floors

Extending the Code

If users want to add a new features or functionality, please write the function firstly and then check where the function can be added in main function to implement related feature and prevent other functions to be affected. For example: if users want to add the feature for long time holding, firstly write a function which can actually implement long time holding feature. Then think about where need to add this function in main function. Because long time holding is only check when the door is open, so the new function can be added in openDoorAgain after `rcall OneSec`, writing a if statement like checking code to determine whether the long time holding function is activated. If yes, `rcall` the new function, otherwise, continue to execute the code in the main code.

Optimizing Performance

1. Elevators are always moving and it can be improved that only can move when request happen to reduce energy consumption.
2. Using an insertion algorithm can simplify the code structure of insert stopped floor to queue and reduce time complex.
3. We put all code in a single file currently, and it can be improved by dispersing entire code into different files (modularity) for use.
4. Currently the code is linear, thus it may cause cumbersome to change or add extra functionalities.
5. The use of long time holding is not humanized enough. Lack of valid prompts displayed on LCD to give user feedback.
6. Long time holding can be added an extra limitation: only one elevator can on the holding state at the same time to prevent congestion.
7. The emegency function design logic can be optimized that the elevaor can move downward directly even the elevator in the moving state(current situation is that the lift in the moving state will move to the nearest floor and then enter the emergency mode).

Bug Fixes

If find some errors from the code, please check the configuration or some more informations for the Board and coding, please check ATmega2560datasheet, AVR-Instruction-Set and AVR-Assembler-Guide. At the same time, please according to the comment and check where need to fix. For example, is the error for LCD result, please check the LCD functions to debug.

1. When we have simulater, we can set a breakpoint and check each value for the code.

2. When we want to check the value on the board is correct or not when have not simulater.

- Connect the wire to the LED Bar to show the value through the LED Bar. For example: 1011 is shown as the light is on-off-on-on.

- Or we can connect the line to LCD and print out the number on LCD to check is correct or not.

Assumptions

1. If press open when in the holding state, will hold another 3 seconds again.
2. When door is closing and press open, start a new 5 seconds from open state.
3. When door is opening and press close, skips the holding 3 seconds part and executes the close state.
4. When the elevator is in the holding state(i.e in the 3 seoncds) and press the close door button, the elevator will skip rest holding time and close the door directly.
5. When the elevator is in the holding state(i.e in the 3 seoncds) and press the open door button, the elevator will hold another 3 seconds again.
6. The floor movement direction is displayed on the second line of LCD (i.e. the U represents the direction Up and the D represents the direction Down).
7. The floor movement direction is displayed on the LED as well. When moving upwards, blink the most upper LED bar. When mobing downwards, blink the most lower LED bar.
8. When in holding state, the LED Bar will blinking together from the first floor to current floor.
9. When in open state and press emergency, the door will close immediately after open.(which means no holding)

10. The Strobe LED will blink with same frequency as elevator moving down before reaching the first floor during the emergency state. as the elevator arrives the first floor, when the door opening/closing and the lift holding, the strobe LED blink once per second. After closing the Strobe LED will flash rapidly.
11. As the emergency state activated, when the elevator is moving up, the elevator will go up to the nearest floor first then do not open the door and down to the first floor. If direction is down, elevator will move to the first floor directly. For example, if the emergency state is activated when the elevator is moving from fifth floor to sixth floor, the elevator will arrive the sixth floor first then go to the first floor directly.
12. ABCD on keypad is not used for the whole project. Therefore, pressing A, B, C, and D will cause nothing change in the elevator system.
13. The elevator will maintain moving state up and down between floors 1 to 10, even if no body requires for the elevator, it will keep moving from 1 to 10 and then 10 to 1.
14. Long time holding button can only be used in the holding state.
15. If the close door button or emergency button are pressed, the long-time holding state will be exited.
16. We use some delay functions to perform the debounce. Therefore, LED bars, Strobe LED and motor may behave slow response for a short time when activate related functions.
17. Because of the unstable signal from the buttons, the action of the motor will not be 100% exactly as what we expected. (may not detect the signals)
18. When you holding the open button when the lift is holding, and you press the close button, the door will close immediately.