Design manual

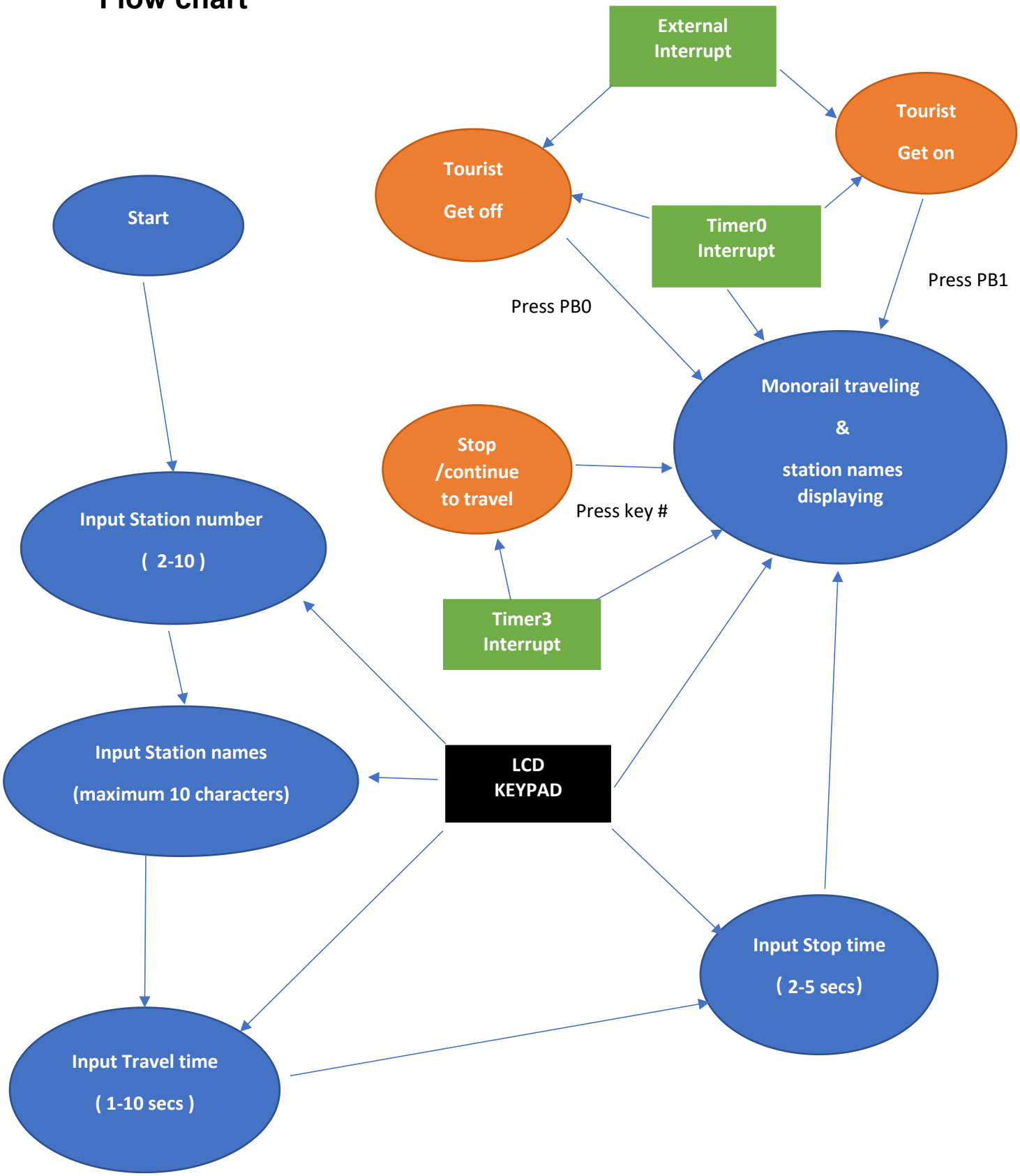
COMP2121 Project

A Monorail Emulator

z5109027 Jingcheng LI

z5028039 Yuanyuan Kong

Flow chart


External Interrupt
Tourist Get on
Tourist Get off
Start
Timer0 Interrupt
Press PB1
Press PB0
Monorail traveling & station names displaying
Stop /continue to travel
Press key #
Input Station number (2-10)
Timer3 Interrupt
Input Station names (maximum 10 characters)
LCD KEYPAD
Input Stop time (2-5 secs)
Input Travel time (1-10 secs)

Data structure

The registers are defined as below, using .def:

Name	Register	Function
to_stop	R2	Flag register to check if the monorail stops at the next station
rolling	R3	Flag register to check if the motor is running
speed	R4	To control the speed of the motor
to_stop2	R5	Flag register to check if the monorail stops at the next station
need_stop	R6	To check if motor should start running at beginning
index	R7	To count the characters in the station names
trans_time	R8	Temporary register that stores the travel time between two stations
stop_time	R9	Register that stores the stop time at any stations
keys_entered	R10	Calculate number of the keys entered, also works as a counter
stat_num	R11	Status register that stores the number of stations
stat_count	R12	Counter that helps to represent the current station
step	R13	To show the step the program reaches
lcd_temp	R14	Temporary register that stores the ASCII value of a number or character
flag	R15	Flag register to check if the value has been displayed
temp	R16	Temporary register
row	R17	The current row number
col	R18	The current column number
mask	R19	Mask for column/row during scanning
temp2	R20	Temporary register
stop_now	R21	Flag register to check if the monorail needs to stop immediately
Timertemp	R22	Temporary register for Timer0 interrupt
shift	R23	Status register to represent which character is needed
	R24	Lower byte of the counter used in Timer0 interrupt
	R25	Higher byte of the counter used in

		Timer0 interrupt character is needed
XL	R26	Pointer that used for the travel time between two stations
XH	R27	Pointer that used for the travel time between two stations
YL	R28	Pointer that used for the counter in Timer0 interrupt
YH	R29	Pointer that used for the counter in Timer0 interrupt
ZL	R30	Pointer that used for the names of the stations
ZH	R31	Pointer that used for the names of the stations

The addresses are defined as below, using .equ:

Name	Address	Function
PORTLDIR	0xF0	PL7-4 for the output, PL3-0 for the input
INITCOLMASK	0xEF	To scan from the leftmost column
INITROWMASK	0x01	To scan from the top row
ROWMASK	0x0F	To obtain input from Port L
LCD_RS	7	Different pins for the LCD
LCD_E	6	
LCD_RW	5	
LCD_BE	4	
HOB_LABEL	0b00100000	High order bit for symbols
HOB_NUM	0b00110000	High order bit for numbers
HOB_CHAR	0b01000000	High order bit for characters

The data segment id defined as below, using .byte:

Name	Bytes	function
name_list	100	To store the names of ten stations, 10 bytes each, with a maximum number of 10
time_list	10	To store the travel time between two stations, 1 byte each, with a maximum number of 10
TempCounter	2	The time counter in Timer0

Functions and Algorithms

Step 1: Enter the number of stations

This step is to get the number of stations. At first, we call the function “step0” to display the instruction on the LCD screen. Then the main function keeps scanning if there is a key gets pressed. When a keys is pressed, the program will check the status and handle it. Since the data type we need here is numeric value, pressing key * and key A, B, C and D will not work. After entering the value we want, we need to press key # to indicate the end of input, the program will automatically move to the next step.

Also it is important for us to handle the error input. As we know, the maximum number of stations is 10, so station numbers over 10 and numbers like 0,1 or press key # without any input should be handled. When key # is pressed, the function will check whether the value is correct first. If nothing is entered or the station number is 0, 1, maximum number 10 will be stored, if the value is larger than 10, only the last digit will be regarded as input(input will be 6 if 16 is entered), and error inputs will be handled in the same way if it is 0 or 1. The value will be stored into variable “stat_num”. Then the program will automatically move to the next step, and function “step1” will be called to dis play the instruction.

The maximum digit can be entered is 7, after the 7^{th} digit is entered, you can’t enter anything and what you need to do is to press key # to store your input and move to the next step.

Number of stations	
Button	Function
0-9	Enter number of stations
#	Finish inputting and store the number of stations
A	Will not work
B	Will not work
C	Will not work
D	Will not work
*	Will not work

Step 2: Enter the name of each station

This step is to get the names of all stations. As there are 3 characters on 6 keys and 4 on 2 keys, we need to use a “shift” to choose which typical character we need. In out solution, key A, B, C and D represent each character on a number key(D for the 4th character on key 7 and 9), after pressing one of those keys, then press a number key, and the character we want will be displayed on the LCD screen. or example, if you want to enter “D”, you need to press key A first, then press key 3, as a result, “D” will be displayed on the screen. Similarly, if you want “Z”, you need to press key D first, then press key 9. The main function keeps scanning if there is a key gets pressed. When a keys is pressed, the program will check the status and handle it. In this step, since the data type we need here is characteristic value, key D will only work for characters on key 7 and key 9, and pressing number keys 2-9 without pressing character keys first will not work. Also key 0 and key 1 will not work in this step. If we want the white space, we can just press key *. Every time we input a character, it will be stored into the variable namelist. After entering the station name we want, we need to press key # to indicate the end of input, then we can enter the name of the next station. After entering the name of the last station, the program will automatically move to the next step. Detailed algorithm description for character input will be introduced later.

The maximum number of characters of any name is 10. After entering the 10th character, you can’t enter anything and what you need to do is to press key #. Also, if the length of the name you entered is less than 10, after pressing key #, we will use white spaces to make up the rest characters. The function will increase the variable stat_count and call function “step1” to display instruction.

After storing the name of the last station, the program will move to the next step. Function “step2” will be called to display the instruction for the next step.

Names of stations	
Button	Function
2-9	Enter characters
#	Finish inputting and store the name of station
A	Represents the first character on the number key
B	Represents the second character on the number key

C	Represents the third character on the number key
D	Represents the 4th character on the number key
*	White space
0/1	Will not work

Step 3: Enter the travel time from one station to another

This step is to get the time for the monorail to travel from one station to the next station without stopping. The main function keeps scanning if there is a key gets pressed. When a keys is pressed, the program will check the status and handle it. Since the data type we need here is numeric value, pressing key * and key A, B, C and D will not work. After entering the value we want, we need to press key # to indicate the end of input.

Also it is important for us to handle the error input. As we know, the maximum number of travel time is 10, so station numbers over 10 and numbers like 0 or press key # without any input should be handled. When key # is pressed, the function will check whether the value is correct first. If nothing is entered or the station number is 0, maximum travel time 10 seconds will be stored, if the value is larger than 10, only the last digit will be regarded as input(input will be 6 if 16 is entered), and error inputs will be handled in the same way if it is 0. The value will be stored into variable time_list. Function “step2” will be called to display the instruction. And we then can enter the travel time for the next two stations.

The maximum digit can be entered is 7, after the 7th digit is entered, you can’t enter anything and what you need to do is to press key # to store your input and move to the next step. After storing the travel time form the last station to the first station, the program will move to the next step. Function “step3” will be called to display the instruction for the next step.

Travel time from one station to another	
Button	Function
0-9	Enter number of stations
#	Finish inputting and store the number of stations
A	Will not work

B	Will not work
C	Will not work
D	Will not work
*	Will not work

Step 4: Enter the stop time

This step is to get the stop time of the monorail at any station. The main function keeps scanning if there is a key gets pressed. When a keys is pressed, the program will check the status and handle it. Since the data type we need here is numeric value, pressing key * and key A, B, C and D will not work. After entering the value we want, we need to press key # to indicate the end of input, the program will automatically move to the next step.

Also it is important for us to handle the error input. As we know, the minimum stop time is 2 seconds while the maximum stop time is 5 seconds, so stop time over 5 and numbers like 0,1 or press key # without any input should be handled. When key # is pressed, the function will check whether the value is correct first. If nothing is entered or the station number is 0, 1, minimum stop time 2 seconds will be stored, if the value is between 6 to 9, maximum stop time 5 seconds will be stored. If the value is larger than 9, only the last digit will be regarded as input(input will be 6 if 16 is entered), and error inputs will be handled in the same way if it is 0, 1 or 6 to 9. The value will be stored into variable “stop_time”. Then the program will automatically move to the next step, and function “step4” will be called to display the instruction.

The maximum digit can be entered is 6, after the 6^{th} digit is entered, you can’t enter anything and what you need to do is to press key # to store your input and move to the next step.

Stop time at any stations	
Button	Function
0-9	Enter number of stations
#	Finish inputting and store the number of stations
A	Will not work

B	Will not work
C	Will not work
D	Will not work
*	Will not work

Step 5: Monorail emulation

Then the monorail emulation will start. First the initialization for Timer0 interrupt will start and some variables will also be reset. After waiting for 5 seconds, the motor will start spinning with a speed of 60 round per second which represents that the monorail start to travel. During this step, the screen will always display the name of the next station and, with that certain travel time.

PB0 and PB1 are to simulate if a tourist wants to get off or get on at the next station. If you press one of them, the monorail will stop at the next station with the stop time you entered.

Key # is to simulate if the monorail stops half way between two stations. When you press key #, the monorail will stop or it will start travel again.

Moreover, whenever the monorail stops, the motor will also stop and two LEDs will blink at a frequency of 3 Hz. When the monorail is traveling, the two LEDS will be off.

Some important algorithms

Character generating:

Keypad scanning

Press key *: store white space into namelist and display it on the LCD screen

Press character keys: change value of variable shift(A:1, B:2, C:3, D:4)

Press number keys: shift = 4 => more_key

shift = 0 => skip_key

0, 1 pressed => skip_key

8 pressed => more_key

Other => normal_key

Normal/more key: convert the column and row value into ASCII code of the character

End_key: store the character into variable namelist, increase the counter

Convert_end: display the character on the screen

Emulating the behaviour of the monorail

This is basically implemented in the Timer0OVF.

Start of Timer0OVF

Load and increase the temporary counter

Check if need to stop immediately

Check if the motor is spinning

If the motor is not spinning, check if 1/3 second has passed, and 2 LEDs will blink at a frequency of 3 Hz

Check if 1 second has passed

Check if need to display “wait” and wait for 5 seconds, also do initialization for monorail emulation

Check if the monorail emulation need to start

Print the name of the next station, counting travel time， go to the next station if it has reach the saved travel time. Check if need to stop at next station, the monorail will stop for the saved stop time if needed.

Reset the pointer of namelist and timelist and some other variables if the monorail has traveled from the last station to the first station.

End of Timer0OVF

 # operations

Check which step the program has reached

If step 0: handle error input

store station number

call function “step1” to display the instruction

increase the step

If step 1: check if the length of name is less than 10, will use white spaces to make up the characters if it is

call function “step1” to display the instruction

check if all station names are entered

call function “step2” to display the instruction

increase the step

If step 2: handle error input

call function “step2” to display the instruction

check if all travel time is entered

call function “step2” to display the instruction

increase the step

If step 3: handle error input

store stop time

call function “step4” to display the instruction

increase the step

If step 5: change flag of stopping immediately

EXT_INT0:

Get status from SREG

Change flag variable of stopping at the next station

Write status back to SREG

EXT_INT1:

Get status from SREG

Change flag variable of stopping at the next station

Write status back to SREG

Macros

We used macros for LCD commands and clear 2 byte variables.

Interrupts

Interrupts we need to use is External Interrupt, Timer0 Interrupt and Timer3 Interrupt.

External Interrupt: Used for PB0 and PB1, changes the flag to control the monorail to stop at the next station

Timer0 Interrupt: Used for time counting, calculates the time need for monorail emulation

Timer3 Interrupt: Used for motor controlling, change the speed of motor.

Reference and other features:

I used some words in the project specification. Also I want to clarify about how I handle the error input, I emailed Hui before about if my solution is OK and he said that was fine.