

COM2121 Project Document

Liangde Li (z5077896)

User Manual:

1. Wiring up the AVR board

- a. Connect LED pin 0-7 to Port C pin 0-7
- b. Connect LED pin 8-9 to Port B pin 0-1
- c. Connect C pin 0-3 to Port L pin 3-0
- d. Connect R pin 0-3 to Port L pin 7-4
- e. Connect OpE to one of +5v pin
- f. Connect Port E pin 2(PE2) to JP91
- g. Connect Mot to Pot
- h. Connect PB(Push Button) pin 0-1 to RDX 4-3
- i. Connect BE-RS to Port A pin 4-7
- j. Connect D pin 0-7 to Port F pin 0-7

2. Configuring system

- a. Load the program to the board.
- b. Now the LCD screen will display "Max stations:", you can press keypad number buttons to input a number you want, from 2 to 10, as the number of stations in this monorail system. And press 'D' to confirm. If you give any invalid input, the system will display "XMax stations:", you just need to input a valid number again.
- c. Now the system will ask you to give name to each station, display "Name1:", the number will vary from 1 to the max number of stations you input in step b. The name must be formed by 1 to 10 characters including upper case letters and space. For the keys from 2 to 9, each has three letters on it, to input a letter, press one of these keys means you want to select from these three letters, and then press 'A', 'B' or 'C' which means you choose the first, second or third letter. Then the letter you chose will display on screen. If you want input a space, press '0'. At the end of inputting one name, press 'D' to confirm. At the process of inputting name, if any invalid input you gave, the screen will display "XName1:", you just need to input this name again.
- d. Now the system will ask you to set the time to travelling between any neighbour stations, like 1->2, 2->3.....last->1. At beginning, the screen will display "Time1to2:". Please give a

number from 1 to 10 as described in step b. Any invalid input gotten, screen will display "XTime1to2:", just input a number again. For each time, press 'D' to confirm it.

e. Now the system will ask you to decide the amount of time to stop at each station. Screen will display "Stop time:". Give a number from 2 to 5 as input, any invalid input will be displayed as "XStop time:", select the input again. Press 'D' to confirm input.

f. Now the screen will display "Wait 5 seconds", and the emulation will start running in 5 seconds.

g. When motor is running (60 rps), the screen will display the name of next station.

f. When Push Button 0 is pressed, means tourist wants to get off at next station, the motor will stop at next station. LED 0-3 will light up.

g. When Push Button 1 is pressed, means tourist wants to get on at next station, the motor will stop at next station. LED 4-7 will light up.

h. When motor is running, pressing '#' means halt right now, the motor will start running if '#' is pressed again.

i. Whenever motor is stop or halt, LED8-9 will blink at frequency of 3Hz, they will be off when the motor is running.

Design Manual:

1. Module specification

a. RESET module(interrupt handler)

RESET module will be executed when a RESET interrupt occur or at the beginning of program starts. RESET module set the stack pointer, set the data direction of each port, enable the external interrupt, set the PWM, and initialise the beginning state of LCD screen.

b. Main module

Main module is the most important part of this program. It process all the information from other modules, and control the flow among different configuration steps. At the end of the main module, a "mainloop" is used to simulate the system and will run forever until a RESET interrupt occurs.

c. Interrupt zero module(interrupt handler)

Interrupt zero will occur when user pressed Push Button 0. Interrupt zero module will modify the value of register leds, to make sure the bit 4 to bit 7 are 1s.

d. Interrupt one module(interrupt handler)

Interrupt one will occur when user pressed Push Button 1. Interrupt one module will modify the value of register leds, to make sure the bit 0 to bit 3 are 1s.

d. GetKeypadNumInput module(function)

GetKeypadNumInput module basically is a loop and will run until user press the keypad. The information will be stored in r16(temp). 0-9 means the number keys, 10 means A, 11 means B, 12 means C, 13 means D, 14 means *, 15 means #.

e. display_station_number module(function)

display_station_number module gets the input from r16(temp), and display this number on the LCD screen. The number could be from 1 to 10.

f. display_station_name module(function)

display_station_name module gets the number of current station from register station, it will display the name of next station on the LCD screen. To get the name, it will make z points to the right place of address StationNames, get letters byte by byte.

g. sleep_1ms, sleep_5ms, sleep_10ms, sleep_50ms, sleep_100ms, sleep_500ms, sleep_1s(macros)

Sleep certain amount of time as shown in names.

2. Algorithms

Display "Max stations:"

Get input NUM

for(i=1; i<=NUM; i++){

 display "Namei: "

 Get input NAMEi

 Store in StationNames

}

for(i=1; i<=NUM; i++){

 Display "Time i to i+1:"

 Get input TIMEi

}

Display "Stop time: "

Get input STOPTIME

Display "Wait 5 seconds"

To emulate the behaviour of monorail, there is a mainloop basically will run every 100ms.

Assume currently at station m, total station n

Mainloop:

 Sleep 100ms

 If is in stop state{

```

        If has gone through enough time
            Reverse the state of LED8-9
    }
    If is in running state{
        Make sure LED8-9 is off
        If n==m{
            Get time t, from station m to 1
        }Else {
            Get time t, from station n to n+1
        }
        If counter == 10*t {
            Clear counter
            Change current station number
            Make sure LED0-7 is off
            If people wanted on or off {
                Change state to stop
            }
            Continue
        }
    }
    If is in stop state {
        Get stop time T
        If T==10*t{
            Clear counter
            Change state to running
            Continue
        }
    }
    Display next station name

```

```

    If is running{
        Make sure motor is running
    } else{
        Make sure motor is off
    }
    If is running{
        Check if # is pressed
        If # is pressed{
            Stop motor
            Loop keep checking if # is pressed again, break if is pressed
        }
    }
    Counter++;
}

```

3. Data Structure

Most of the data will be stored in the following part:

```

.dseg
.org 0x200
SecondCounter: .byte 2
TempCounter: .byte 2
MaxStations: .byte 1
StationNames: .byte 100
TimeToNext: .byte 10
StopTime: .byte 1

```

The SecondCounter and TempCounter are two byte long to store a 32-bits unsigned numbers.

MaxStations is just one unsigned byte.

StationNames is 100 bytes large, to store 10 names at maximum, each name can be formed by 10 char.

TimeToNext is 10 byte large, to store 10 8-bits unsigned numbers

StopTime is used to store one 8-bits unsigned number.

4. System control flow

