

---

# ECE 375

Final Project Report – Morse code

Lab Time: Thursday, 12, 10

*Longjie Guan*

## ABSTRACT

The introduction part will introduce people the basic thing this project do, the program overview show people the idea and design of this project. The Main routine parts will show people the details of all functions in this program. In conclusion part, all thing will be sum up.

## INTRODUCTION

For this project, it needs student to change the letter in the LCD in 2 ways which is right-order way and inverse-order way. So, the right-order way means people can select the letter they want from A to Z and reverse-order means people can select the letter they want from Z to A. People can select multiple characters by increasing the position in the LCD display in the second line. And then, it is necessary to show people the morse code in the method of LED flashing. After the completion of the transmission, the LCD should back to initialization which is "A" needs to be display in the second line first character of the LCD display. And then, it is enables for people to choose second many characters and transmit in morse code again.

## PROGRAM OVERVIEW

The first thing need to be design in this program is initialization part, so in initialization part, there are also many things that student need to initialize for better usage in the following programs.

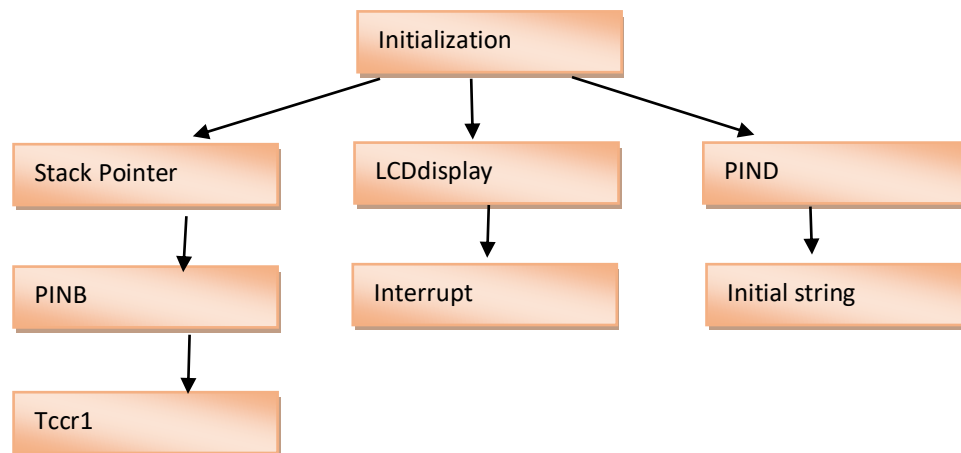


Figure 1: Block Diagram (1) - Initialization

From figure 1, people can see the basic part of initialization module, it covers stack pointer, LCDdisplay, PIND, PINB, interrupt, Initial string, and Time/cunter1. The stack pointer is the very basic set in this course and people need to use LCD to display, so it is necessary to include that. Also, PIND is used for controlling the input part, it used in polling way in this project. The interrupt is also used to control the first bit in PIND. The PINB charges on showing the morse code and time/cunter 1 used for setting the time delay. People need to store the initial string in the program memory and apply it in the first display.

## MAIN ROUTINES

There are basic 4 main functions control the running of this program. However, it is complex, they are connecting together in some parts. It means they are all related and if people can't deal with any function very well, it will cause "bug".

### MAIN ROUTINE

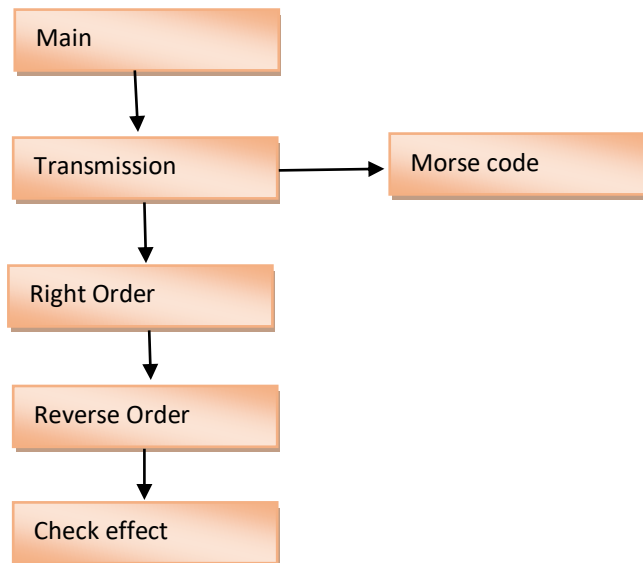


Figure 2: Block Diagram (2) – Main routine

In the main routine, it has 4 functions which has their unique working ways.

### TRANSMISSION ROUTINE

The transmission function is used for transmit the morse code just like it names. However, it uses the special way to transmit that which is transmit the data that people store in the data memory. After this function get these data, it will check the condition to transmit it as morse code. Such as the transmission fetch the number in the first address of data memory is 65. Then, the morse code function will check that if the value is 65, it will show the morse code of A. By repeating this way, the transmission function can transmit all characters and show them.

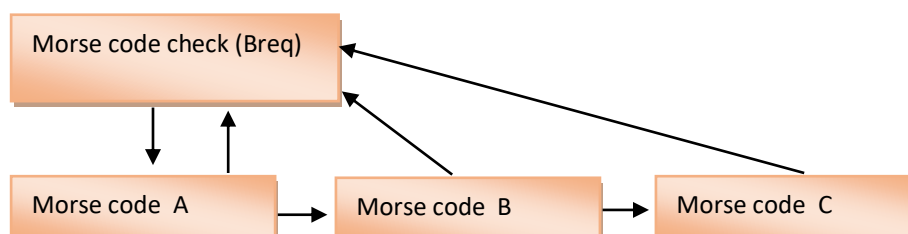


Figure 3: Morse code fetch and execute cycle

## RIGHT ORDER ROUTINE

In this function, people can increase the number shown in the LCD panel. So, it means it also need to have a function which is change the value in the data memory. For locating the exact data memory. The code people need to use is **ADD YL, INDEX**. By the application of this instruction, people can locate the position of pointer what ever it changes.

## REVERSE ORDER ROUTINE

In this function, people can decrease the number shown in the LCD panel. So, it means it also need to have a function which is change the value in the data memory. For locating the exact data memory. The code people need to use is **ADD YL, INDEX**. By the application of this instruction, people can locate the position of pointer what ever it changes.

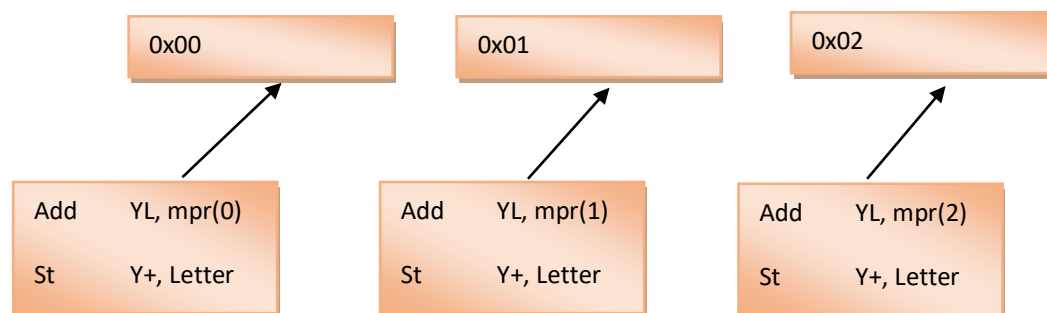


Figure 4: Manipulate data (persudo code)

According the figure showed upward, it teaches people the way to manipulate data in this program. The index is stored in the map because the add instruction only supported for 2 registers. So, it means increase the address of YL. By increase that, we can see the second part of chart shows, the data memory just directly improving to 0x01 and then people can store the value just in this data memory. The advantage of this way is it can avoid overwritten. If people increase pointer by one times, it way overwritten the first pointer.

## CHECK EFFECT ROUTINE

On check effect routine, people need to check is that has effect on the working of other pins. Such as people will press check effect and right order, if check effect has no effect on the working of right order. It should be good.

## CONFIRM ROUTINE

The confirm routine doesn't put in the main routine, however, it doesn't mean this project doesn't need this routine. Actually, the confirm routine is controlled by interrupt 0 and it also be initialized in the initialization part. The usage of interrupt is because it is straight forward and easy. Confirm function has several roles.

1. Overwritten the line 1 to ""Enter a world".
2. Set the first letter is "A" when it was pressed.
3. Move to next letter in LCD part
4. Increase the pointer in data memory part.

In confirm routine, each time people press that, people can see there is an A was added in the next address and the people can change that by right order and reverse order, so it is really a base function of this project. If the value in confirm routine is changed and people press it again can have a new A in next address.

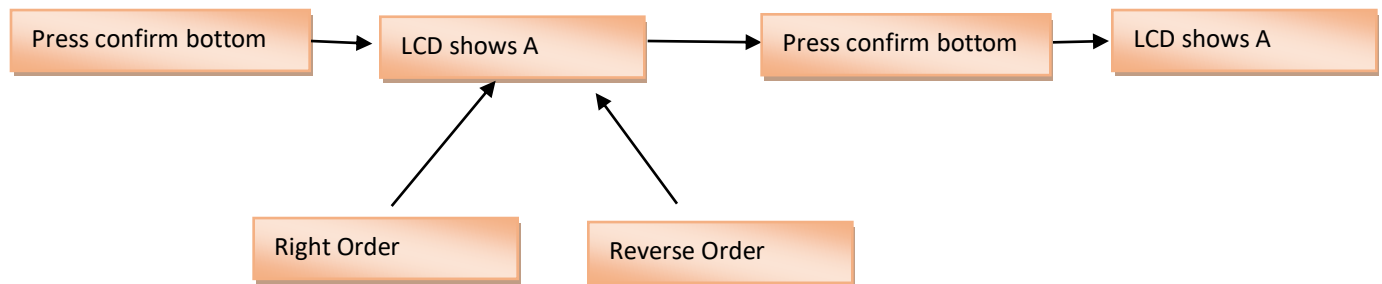


Figure 5: Block diagram (3) – Confirm stage

As the diagram shown, if the bottom of confirm was pressed, LCD shows A and then right order and reverse order both can change the value of A. However, if the bottom press again, it will show A again.

## USAGE OF TIME/COUNTER 1

The time/counter for this project to use is time/counter 1 and the mode is normal mode. The prescaler be set is 8 in this program for better working on both 200ms delay and 1s delay

Function to calculate that for normal mode:

$$\text{Delay} = [(MAX + 1 - VALUE) * PPRESALE] / CLK$$

### 200MS DELAY

$$\text{Delay} = [(65536 - 100) * 8] / 16\text{Mhz} = 0.0325\text{s}$$

$$\text{Count} = 6$$

$$\text{TotDelay} = 0.0325\text{s} * \text{Count} = 0.195\text{s}$$

### 1S DELAY

$$\text{Delay} = [(65536 - 100) * 8] / 16\text{Mhz} = 0.0325\text{s}$$

$$\text{Count} = 31$$

$$\text{TotDelay} = 0.0325\text{s} * \text{Count} = 1.0075\text{s}$$

So, by calculation, the delay for 200ms is 195ms and delay for 1s is 1.0075s. Those value are very close, people can't get the exact value, they can try their best to make sure the answer is close. They also can change the value and the prescale.

## CHALLENGE

For this part, I have many changes because it has very complex logic. The first one is set up the input because I don't know why the interrupt can't used for PD4-7. And then, the time/counter part also challenge me because I am not familiar with time/counter 1 and it is the new set. And then, it is really hard for some main role function to cause bugs. Usually, if one function has problem, I need to find several functions because they are really connected and the process is long. So, it is a good way to put them in a test program.

## IMPROVEMENT

If I have second time, I believe I can improve that because I think I have a right track on the initialization part and it can save me too much time because I work hard on that before. And also, I find another way to do in this program is may be store A-Z to program memory and it may be more easy and don't use ASCII to translate that. In addition, some parts I can use a loop to do that. The logic for my project is not perfect clear, so if I have second time to design that, I think I will have a more clear logic and it is very important to design that.

## CONCLUSION

In this project, student use two different kinds of input which are interrupt and polling. People can't used interrupt for PD4-7 because the ATmega128 doesn't support that. So, people can use polling for those bits. The benefit of polling is easy to debug because it has multiple rcall functions in main function. The interrupt way is hard to debug but really straight forward. Each functions has their unique rules. By the exercise of this project, student can have a better coding skills for assembly.