

# void LCD\_init (void);

## • Function Description:

- This function is used to initialize the LCD.
- Neglect the delay by using the timer with interrupt, and one interrupt equal to the delay that we want, so at this time we can write the command on LCD port to complete the LCD initialization. At the end of initialization, set the End Flag\_G (Static Global Flag) and call the LCD\_WriteChar function to check, if there are a data that wanted to print.
- In interrupt service routine(ISR), call the LCD\_init function.

This function will initialize the LCD.

```
start
Declare static flag=0

if "flag==0"
set timer & enable OVF interrupt
flag=1

else if "flag==1"
RS pin is low
Write on LCD port
EN pin is high
set timer & enable OVF interrupt
flag=2

else if "flag==2"
EN pin is low
RS pin is low
Write on LCD port
EN pin is high
set timer & enable OVF interrupt
flag=3

else if "flag==3"
EN pin is low
RS pin is low
Write on LCD port
EN pin is high
set timer & enable OVF interrupt
End_Flag_G=1
Call the LCD_WriteData function

End
```

```
void LCD_WriteData(u8 data);
```

- **Function Description:**

- This function is used to display any data on LCD
- The function store any data that send in the array of char (As a queue) to start displaying the data on LCD by using timer & interrupt to neglect the delay after the LCD initialization finish.
- In interrupt service routine(ISR), call the LCD\_WriteData function.

This function will display the data on LCD.

```
start
Declare static index=0,i=0

if "Data!=NULL"
    str[index]=data
    index++
    if "End_Flag_G==1 &&
str[i]!=NULL" >>line #
        RS pin is high
        Write on LCD port
        EN Pin is high
        indexx--
        i++
        set timer & enable OVF interrupt
    else
        End

    else
        go to line #

End
```

```
Keypad_Status_en KEYPAD_Init(PIn_name First_Output,PIn_name Firs_Input);
```

### Function Description:

- This function is used to initialize the Keypad.
- At first, the function check if argument pins is valid to be keypad row and colume, then return Done or Error based on last check.

**This function will initialize the Keypad.**

start

if"Row and column pins are  
valid"

initial the keypad pins

Key\_Flag\_G=1

return Done

else

return Error

End

```
void KEYPAD_GetKey(u8* key);
```

### Function Description:

- This function is check any key that user is pressed.
- It checks all buttons.If the button isn't pressed, increase the count .If the button is pressed and the count doesn't reach the limit, increase the count. If the button is pressed and the count reach the limit, return the key that is pressed and make the count=0.
- So, we overcome the bouncing whithout using delay or busy wait (Hold the CPU)

This function will check which  
button the user has pressed.

```
start
Declare Row=0,Col=0,limit=8,static
count=0
All row pins are high

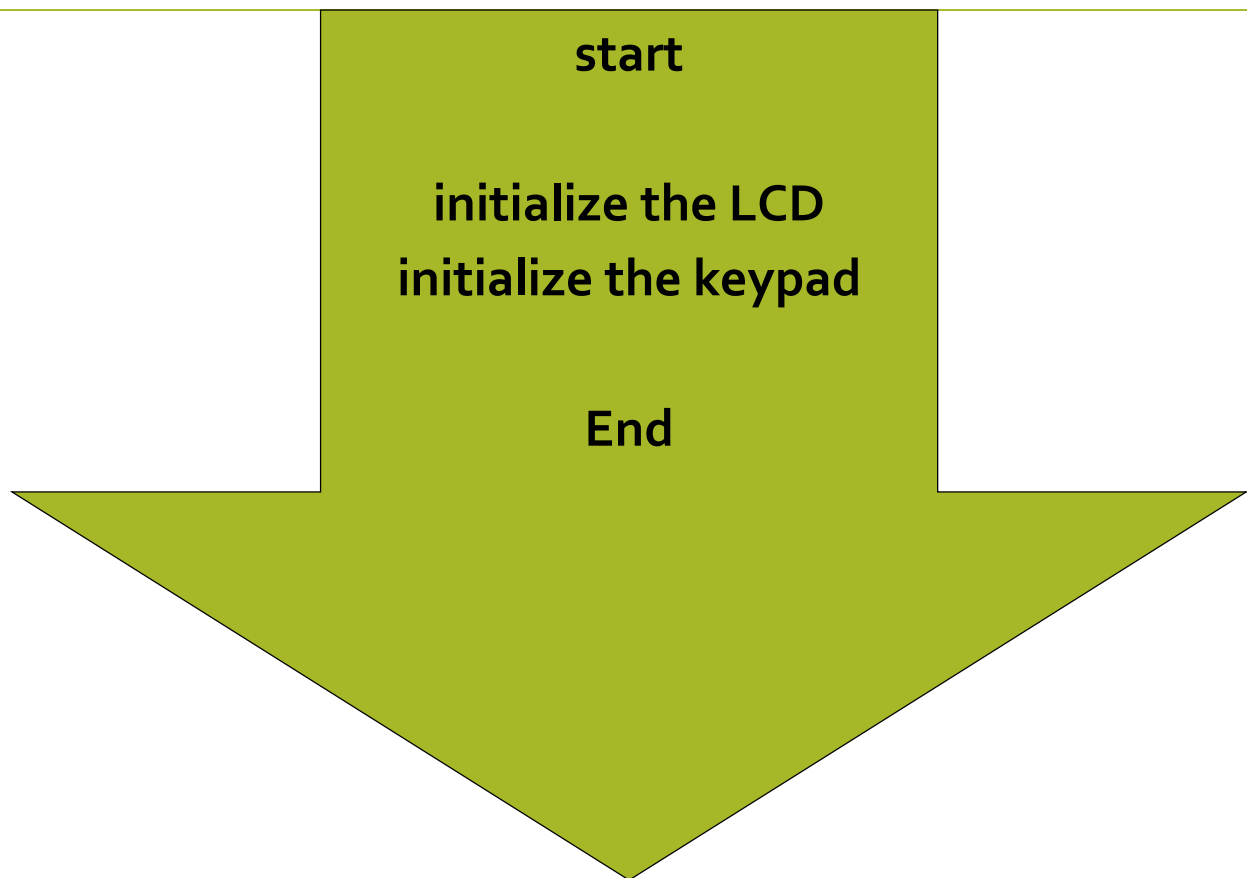
if "Row<Keypad_Row &&
Key_Flag_G==1" >>line #
  Keypad Row is high
  if "Col<Keypad_Col"
    if "The button is pressed &&
count>=limit"
      count=0
      store the pressed button
    else
      count++
  else
    go to line #
```

End

```
void APP_Init(void);
```

**The application Description:**

1) Initialize the LCD & Keypad.



```
void APP_Start(void);
```

**The application Description:**

2)check the all keypad buttons and display the button that is pressed on LCD.

**start**

**Check any pressed  
button & store it  
Display the pressed  
button on LCD**

**End**