

```

1  #include
   <main.h>
                                   //CCS
2
3  #define LCD_ENABLE_PIN
   PIN_B2
                                   //CCS
4  #define LCD_RS_PIN
   PIN_B0
                                   //CCS
5  #define LCD_RW_PIN
   PIN_B1
                                   //CCS
6  #define LCD_DATA4
   PIN_B4
                                   //CCS
7  #define LCD_DATA5
   PIN_B5
                                   //CCS
8  #define LCD_DATA6
   PIN_B6
                                   //CCS
9  #define LCD_DATA7
   PIN_B7
                                   //CCS
10
11 #define HS
   PIN_B3
                                   //Handshaking between PICs
12
13 #include
   <lcd.c>
                                   //CCS
14
15 unsigned int Input_Buttons(void);
                                   //Function for Inputs
16 unsigned int Read_Memory(unsigned int);
                                   //Function for Reading Memeory
17 void Write_Memory(unsigned int,unsigned int);
                                   //Function for Writing to Memory
18 void Clear_Memory(unsigned int);
                                   //Function for Clearing Memory
19
20
21 void main()
                                   //Main Program Below
22 {
23     unsigned int Data,Value,Local=0;;
                                   //Function for Data, Value and Local
24     port_b_pullups(0x08);
                                   //Set Handshaking on Port B with Pull Up
25
26     lcd_init();
                                   //LCD Programming Below
27
28     while(TRUE)
                                   //Start of LCD Loop
29     {
30         Value=0;

```

```

31         //Set Value as 0
        Value=Input_Buttons();

32        //Inputs are the Value in Loop
        switch(value)

33            //Start of LCD Program
            {
34                case 0x01:        if(!(Local & 0x01))Local+=1;
                                    //If Off turn On
35                                    else Local=Local & 0xfe;
                                    //If On turn Off
36                                    Write_Memory(0x00,Local);
                                    //Write to Memory
37                                    break;
                                    //End of Case

38                case 0x02:        if(!(Local & 0x02))Local+=2;
                                    //If Off turn On
39                                    else Local=Local & 0xfd;
                                    //If On turn Off
40                                    Write_Memory(0x00,Local);
                                    //Write to Memory
41                                    break;
                                    //End of Case

42                case 0x04:        if(!(Local & 0x04))Local+=4;
                                    //If Off turn On
43                                    else Local=Local & 0xfb;
                                    //If On turn Off
44                                    Write_Memory(0x00,Local);
                                    //Write to Memory
45                                    break;
                                    //End of Case

46                case 0x08:        if(!(Local& 0x08))Local+=8;
                                    //If Off turn On
47                                    else Local=Local & 0xf7;
                                    //If On turn Off
48                                    Write_Memory(0x00,Local);
                                    //Write to Memory
49                                    break;
                                    //End of Case

50                case 0x10:        if(!(Local & 0x10))Local+=0x10;
                                    //If Off turn On
51                                    else Local=Local & 0xef;
                                    //If On turn Off
52                                    Write_Memory(0x00,Local);
                                    //Write to Memory
53                                    break;
                                    //End of Case

54                case 0x20:        if(!(Local & 0x20))Local+=0x20;
                                    //If Off turn On
55                                    else Local=Local & 0xdf;
                                    //If On turn Off
56                                    Write_Memory(0x00,Local);
                                    //Write to Memory
57                                    break;
                                    //End of Case

58            }

59            if(Value)

60                //If Value
61                {
62                    Data=Read_Memory(0x00);
63
64                }
65            }
66        }
67    }
68
69

```

```

//Data
in Memory is Data for Loop
70         printf(lcd_putc, "\f Value = %x", Data);
//Print on LCD Value
followed by the Stored Data
71         lcd_gotoxy(3,3);

//Positioning of Digits onto LCD
72         printf(lcd_putc, " Memory %x, %x, %x", Read_Memory(0x00), Read_Memory(0x05),
Read_Memory(0x09)); //Print Data from Specific Memory Locations onto LCD
73     }
74 }
75
76 }
77
78 unsigned int Read_Memory(unsigned int Address) //Function
for Reading Memeory
79 {
80     unsigned int val;

//Start
81     while(!input(HS));

//If Handshaking Line is High
82     output_low(HS);

//Set Handshaking Line Low
83     i2c_start();

//I2C Start Pulse
84     i2c_write(0xA0);

//Set PIC Type and Number
85     i2c_write(Address);

//Set Address
86     delay_ms(10);

//Delay for PIC
87     i2c_start();

//I2C Start Pulse
88     delay_ms(10);

//Delay for PIC
89     i2c_write(0xA1);

//Set PIC Type and Number
90     val=i2c_read();

//Read from PIC
91     delay_ms(10);

//Delay for PIC
92     i2c_stop();

//I2C Stop Pulse
93     delay_ms(10);

//Delay for PIC
94     input(HS);

//Set Handshaking Line High
95     return(val);

//End
96 }
97
98 void Write_Memory(unsigned int Address, unsigned int Data)

```

```

99         Writing to Memeory
100         {
101             while(!input(HS));

102             //If Handshaking Line is High
103             output_low(HS);

104             //Set Handshaking Line Low
105             i2c_start();

106             //I2C Start Pulse
107             i2c_write(0xA0);

108             //Set PIC Type and Number
109             i2c_write(Address);

110             //Set Address
111             i2c_write(Data);

112             //Write Data to Address
113             delay_ms(10);

114             //Delay for PIC
115             i2c_stop();

116             //I2C Stop Pulse
117             delay_ms(10);

118             //Delay for PIC
119             input(HS);

120             //Set Handshaking Line High
121         }

122 void Clear_Memory(unsigned int Data, unsigned int Clear=0)
123                                     //Function for
124                                     Clearing Memeory
125         {
126             i2c_start();

127             //I2C Start
128             Pulse
129             i2c_write(0xA0);

130             //Set PIC Type and Number
131             i2c_write(Clear);

132             //Set Clear
133             i2c_write(Data);

134             //Write Data to Address
135             i2c_stop();

136             //I2C Stop Pulse
137         }

138 unsigned int Input_Buttons(void)
139                                     //Function for
140                                     Inputs
141         {
142             unsigned int value=0;

143             //Read Values
144             value=input_a();

145             //Values are derived from Terminals A
146             return(value);

```

```
//End of Function  
}
```