Automatic System Adapter using Bidirectional Object Counting System

Mohammad Rashedul Alam
School of Electrical and
Computer Engineering
North South University
Dhaka,Bangladesh
Email:rashedul.alam@northsouth.edu

***Abstract*—Counting the number of objects inside a certain place is needed for a lot of system. The management for counting objects for a system is costly. Headcount of people is time consuming and not very accurate most of the time. The low cost of availability of sensors has made it possible for making low cost counting systems. The aim of this project is to introduce digital low cost counting system in different sectors.**

I. Introduction

The number of smart devices being used is increasing day by day. So to make thing more efficient new systems are required. Devices managing power according to demand is crucial. So many indoor Aircondition, Light, Fan etch appliance needs data of the density to act accordingly. Also Counting objects is one of the crucial thing for most of the factories.Factories use sophisticated systems to count their products. But a lot of the time we need counting solution for simple uses that is easy to manage and low costly. For such cases this project is suitable to implement as it requires low power and management. A simple Microcontroller hooked up with sensors is used. As nowadays the small packaging companies are increasing the demand for counting system is increasing. Also a lot of conference hall, auditorium and public places demand people count for their limited access. So a simple counting solution is very demanding for a lot of people. The purpose of this project is to make that counting system simple, low costly and accurate.

II. Materials And Methodology

The system mainly works on the principle that Infrared light falls on photodiode all the time and the voltage remains low on photodiode as current passes through photodiode. But when an obstacle passes through the path the infrared falling on photodiode gets blocked and the voltage across photodiode gets high. The voltage across the photodiode is measured through a PIC16F877A microcontroller. The PIC microcontroller controls the system and counts the people. Microcontroller also decides the direction of entering and exiting. An LCD is connected with the microcontroller and prints the status and total object present in the system. The device keeps track of the total objects even if the power gets disconnected. EEPROM stores the counter variable which is non volatile. Equipments used in the system are given details overview below:

1) IR Circuit: The sensor consist of IR transmitter and Photodiode.The photodiode reacts only to IR rays falling on it. So normal light doesnt affect the system. Photodiode responds to the change of IR light falling on it.

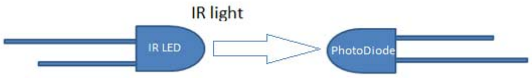

Fig. 1. IR Principle

2) PIC Microcontroller: The system uses PIC16F877A Microcontroller.This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC architecture into an 40 package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. The PIC16F877A features 256 bytes of EEPROM data memory, self programming, an ICD, 2 Comparators, 8 channels of 10-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI) or the 2-wire Inter-Integrated Circuit (IC) bus and a Universal Asynchronous Receiver Transmitter (USART).

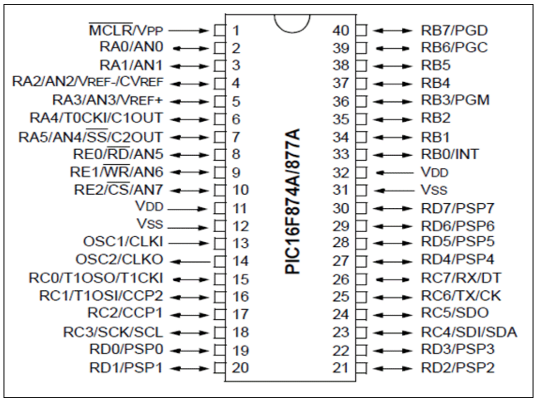

Fig. 2. PIC 16F877A

3) LCD Display: The Model JHD 162A Series LCD is the typical standard HD44780 type of LCD with 16 characters x 2 row LCD module. The LCD displays the state of objects. And indicates ingoing or outgoing direction.

4) Buzzer: The Buzzer beeps when an object passes through. Bidrectional objects triggers the buzzer.

5) Led: The LED's are used for indicating power status and Bidirectional indication. Red LED is used for outgoing objects. And Green LED is used for incoming objects.

6) Potentiometer: The variable resistor or potentiometer is used to adjust the contrast of the LCD

III. Implementation

The PIC Microcontroller is connected with 20MHz crystal oscillator. The Oscillator requires two 22pF capacitor to stabilize the clock speed. The code is written in Assembly language and uploaded to the PIC Microcontrollers Flash memory. The Photodiode sensor is connected to the A/D pin of the microcontroller. Microcontroller receives analog value and converts it to digital value and compares it with the threshold value set by user. The microcontroller decides and increase or decreases counter variable and displays it in the LCD display. The counter variable is stored in the EEPROM so that the counter will be safe in case of power loss.
The System is shown below:

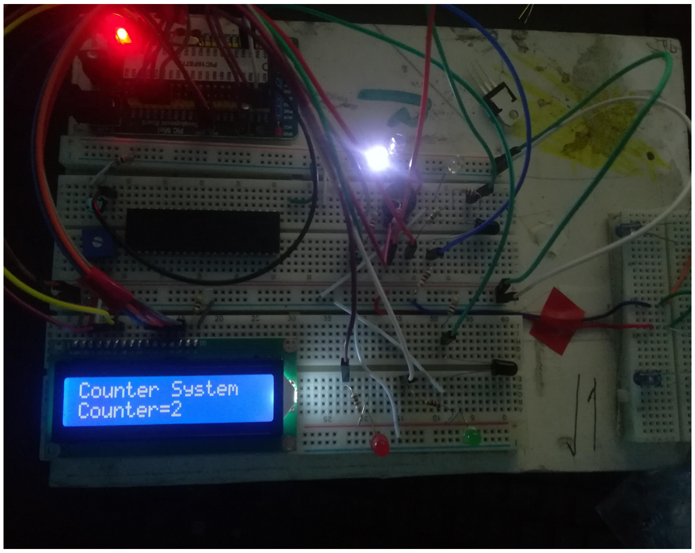

Fig. 3. System

A. Circuit And Figures

The Demo circuit is built in proteus. The proteus Diagram is given below.

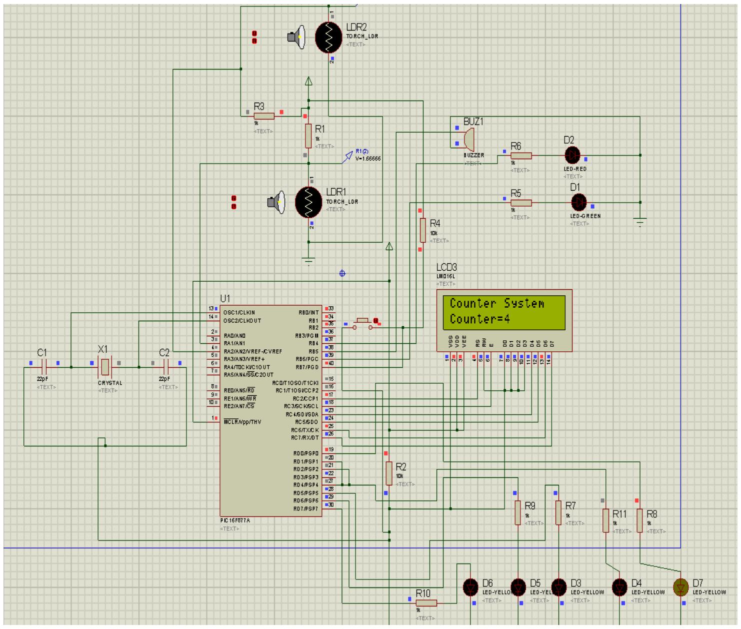

Fig. 4. Proteus Circuit

IV. Application

1) : Indoor Air conditioning can use this system and can adapt to the density of people present and auto adjust power.

2) : Amount of Light and Fan needed to run can be auto adjusted using this circuit.

3) : One example would be the sitting bus service of Dhaka city. They employ a lot of staff members for counting the passenger. As this counting process becomes annoying passengers also. This simple system could be installed in the door of the buses and the counting process becomes automatic.

4) : Factories can install this system and can keep track of their inventory. Product entered and left the facility can be tracked.

5) : Large scale Auditorium, Movie theatre, Conference halls can install this system and keep count of the people available. Also Mall and Offices can use this system.

V. Limitation

This system is low range. Large range implementation of this system will result in error as the IR sensor photodiode gets weak. This limitation can be overcome using Lasers.

VI. Conclusion

The system is overall simplified counting solution for most use cases. This system can ultimately help reduce the overall cost of any counting mechanism service and be easy to adapt. The system saves data in EEPROM and thus avoid the loss of data due to power loss.The system has scope for introducing new functionality and can be easily upgraded. A Lot of small companies and business can greatly benefit themselves by installing this system.

VII. CODE:

Assembly Code of the System :

```
ORG 0x0000 ;

; _LightSet:
;
;       MOVLW   128
;       MOVWF   R0+0
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet45
;       MOVF    FARG_LightSet_countD+0, 0
;       SUBLW   0
; L__LightSet45:
;       BTFSS   STATUS+0, 0
;       GOTO    L_LightSet0
;       CLRF    PORTD+0
;       GOTO    L_LightSet1
; L_LightSet0:
;       MOVLW   128
;       MOVWF   R0+0
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet46
;       MOVF    FARG_LightSet_countD+0, 0
;       SUBLW   0
; L__LightSet46:
;       BTFSC   STATUS+0, 0
;       GOTO    L_LightSet4
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       MOVWF   R0+0
;       MOVLW   128
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet47
;       MOVLW   3
;       SUBWF   FARG_LightSet_countD+0, 0
; L__LightSet47:
;       BTFSC   STATUS+0, 0
;       GOTO    L_LightSet4
; L__LightSet43:
;       MOVLW   1
;       MOVWF   PORTD+0
;       GOTO    L_LightSet5
; L_LightSet4:
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       MOVWF   R0+0
;       MOVLW   128
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet48
;       MOVLW   3
;       SUBWF   FARG_LightSet_countD+0, 0
; L__LightSet48:
;       BTFSS   STATUS+0, 0
;       GOTO    L_LightSet8
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       MOVWF   R0+0
;       MOVLW   128
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet49
;       MOVLW   5
;       SUBWF   FARG_LightSet_countD+0, 0
; L__LightSet49:
;       BTFSC   STATUS+0, 0
;       GOTO    L_LightSet8
; L__LightSet42:
;       MOVLW   7
;       MOVWF   PORTD+0
;       GOTO    L_LightSet9
; L_LightSet8:
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       MOVWF   R0+0
;       MOVLW   128
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet50
;       MOVLW   5
;       SUBWF   FARG_LightSet_countD+0, 0
; L__LightSet50:
;       BTFSS   STATUS+0, 0
;       GOTO    L_LightSet12
;       MOVLW   128
;       XORWF   FARG_LightSet_countD+1, 0
;       MOVWF   R0+0
;       MOVLW   128
;       SUBWF   R0+0, 0
;       BTFSS   STATUS+0, 2
;       GOTO    L__LightSet51
;       MOVLW   10
;       SUBWF   FARG_LightSet_countD+0, 0
; L__LightSet51:
;       BTFSC   STATUS+0, 0
;       GOTO    L_LightSet12
; L__LightSet41:
;       MOVLW   63
;       MOVWF   PORTD+0
;       GOTO    L_LightSet13
; L_LightSet12:
;       MOVLW   255
;       MOVWF   PORTD+0
; L_LightSet13:
; L_LightSet9:
; L_LightSet5:
; L_LightSet1:
; L_end_LightSet:
;       RETURN
;
; _main:
;
;       MOVLW   128
;       MOVWF   ADCON1+0
;       MOVLW   255
;       MOVWF   TRISA+0
;       MOVLW   63
;       MOVWF   TRISC+0
;       MOVLW   128
;       MOVWF   TRISB+0
;       BSF     TRISB+0, 7
;       CLRF    TRISD+0
;       CLRF    PORTD+0
;       MOVLW   2
;       MOVWF   FARG_EEPROM_Write_Address+0
;       MOVLW   5
;       MOVWF   FARG_EEPROM_Write_data_+0
;       CALL    _EEPROM_Write+0
;       MOVLW   3
;       MOVWF   R11+0
;       MOVLW   138
;       MOVWF   R12+0
;       MOVLW   85
;       MOVWF   R13+0
; L_main14:
;       DECFSZ  R13+0, 1
;       GOTO    L_main14
;       DECFSZ  R12+0, 1
;       GOTO    L_main14
;       DECFSZ  R11+0, 1
;       GOTO    L_main14
;       NOP
;       NOP
;       MOVLW   2
;       MOVWF   FARG_EEPROM_Read_Address+0
;       CALL    _EEPROM_Read+0
;       MOVF    R0+0, 0
;       MOVWF   _count+0
;       CALL    _Lcd_Init+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   12
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   17
;       MOVWF   FARG_EEPROM_Read_Address+0
;       CALL    _EEPROM_Read+0
;       MOVF    R0+0, 0
;       MOVWF   _read+0
;       MOVF    R0+0, 0
;       SUBLW   0
;       BTFSC   STATUS+0, 0
;       GOTO    L_main15
;       MOVF    _read+0, 0
;       MOVWF   _count+0
;       MOVF    _read+0, 0
;       MOVWF   _countD+0
;       CLRF    _countD+1
;       GOTO    L_main16
; L_main15:
;       CLRF    _count+0
;       CLRF    PORTD+0
;       CLRF    _countD+0
;       CLRF    _countD+1
; L_main16:
; L_main17:
;       MOVLW   1
;       MOVWF   FARG_ADC_Read_channel+0
;       CALL    _ADC_Read+0
;       MOVF    R0+0, 0
;       MOVWF   _adc+0
;       MOVF    R0+1, 0
;       MOVWF   _adc+1
;       MOVLW   2
;       MOVWF   FARG_ADC_Read_channel+0
;       CALL    _ADC_Read+0
;       MOVF    R0+0, 0
;       MOVWF   _adc2+0
;       MOVF    R0+1, 0
;       MOVWF   _adc2+1
;       MOVF    _adc+1, 0
```

```
;       SUBLW   1
;       BTFSS   STATUS+0, 2
;       GOTO    L__main53
;       MOVF    _adc+0, 0
;       SUBLW   144
; L__main53:
;       BTFSC   STATUS+0, 0
;       GOTO    L_main20
;       INCF    _count+0, 1
;       MOVLW   1
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_row+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_column+0
;       MOVLW   ?lstr1_MyProject+0
;       MOVWF   FARG_Lcd_Out_text+0
;       CALL    _Lcd_Out+0
;       MOVLW   32
;       MOVWF   PORTB+0
;       MOVLW   2
;       MOVWF   R11+0
;       MOVLW   69
;       MOVWF   R12+0
;       MOVLW   169
;       MOVWF   R13+0
; L_main21:
;       DECFSZ  R13+0, 1
;       GOTO    L_main21
;       DECFSZ  R12+0, 1
;       GOTO    L_main21
;       DECFSZ  R11+0, 1
;       GOTO    L_main21
;       NOP
;       NOP
;       MOVLW   64
;       MOVWF   PORTB+0
;       INCF    _countD+0, 1
;       BTFSC   STATUS+0, 2
;       INCF    _countD+1, 1
;       CLRF    _i+0
; L_main22:
;       MOVLW   4
;       SUBWF   _i+0, 0
;       BTFSC   STATUS+0, 0
;       GOTO    L_main23
;       MOVLW   28
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   13
;       MOVWF   R11+0
;       MOVLW   175
;       MOVWF   R12+0
;       MOVLW   182
;       MOVWF   R13+0
; L_main25:
;       DECFSZ  R13+0, 1
;       GOTO    L_main25
;       DECFSZ  R12+0, 1
;       GOTO    L_main25
;       DECFSZ  R11+0, 1
;       GOTO    L_main25
;       NOP
;       INCF    _i+0, 1
;       GOTO    L_main22
; L_main23:
```

```
;       MOVLW   1
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       CLRF    PORTB+0
;       MOVF    _countD+0, 0
;       MOVWF   FARG_LightSet_countD+0
;       MOVF    _countD+1, 0
;       MOVWF   FARG_LightSet_countD+1
;       CALL    _LightSet+0
;       GOTO    L_main26
; L_main20:
;       MOVF    _adc2+1, 0
;       SUBLW   1
;       BTFSS   STATUS+0, 2
;       GOTO    L__main54
;       MOVF    _adc2+0, 0
;       SUBLW   144
; L__main54:
;       BTFSC   STATUS+0, 0
;       GOTO    L_main27
;       MOVF    _count+0, 0
;       SUBLW   0
;       BTFSS   STATUS+0, 0
;       GOTO    L_main28
;       CLRF    _count+0
;       MOVLW   21
;       MOVWF   R11+0
;       MOVLW   75
;       MOVWF   R12+0
;       MOVLW   190
;       MOVWF   R13+0
; L_main29:
;       DECFSZ  R13+0, 1
;       GOTO    L_main29
;       DECFSZ  R12+0, 1
;       GOTO    L_main29
;       DECFSZ  R11+0, 1
;       GOTO    L_main29
;       NOP
;       CLRF    PORTD+0
;       CLRF    _countD+0
;       CLRF    _countD+1
;       CLRF    FARG_LightSet_countD+0
;       CLRF    FARG_LightSet_countD+1
;       CALL    _LightSet+0
;       GOTO    L_main30
; L_main28:
;       MOVLW   32
;       MOVWF   PORTB+0
;       MOVLW   2
;       MOVWF   R11+0
;       MOVLW   69
;       MOVWF   R12+0
;       MOVLW   169
;       MOVWF   R13+0
; L_main31:
;       DECFSZ  R13+0, 1
;       GOTO    L_main31
;       DECFSZ  R12+0, 1
;       GOTO    L_main31
;       DECFSZ  R11+0, 1
;       GOTO    L_main31
;       NOP
;       NOP
;       MOVLW   16
;       MOVWF   PORTB+0
;       DECF    _count+0, 1
```

```
;       MOVLW   1
;       SUBWF   _countD+0, 1
;       BTFSS   STATUS+0, 0
;       DECF    _countD+1, 1
;       MOVLW   1
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_row+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_column+0
;       MOVLW   ?lstr2_MyProject+0
;       MOVWF   FARG_Lcd_Out_text+0
;       CALL    _Lcd_Out+0
;       CLRF    _i+0
; L_main32:
;       MOVLW   4
;       SUBWF   _i+0, 0
;       BTFSC   STATUS+0, 0
;       GOTO    L_main33
;       MOVLW   28
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   13
;       MOVWF   R11+0
;       MOVLW   175
;       MOVWF   R12+0
;       MOVLW   182
;       MOVWF   R13+0
; L_main35:
;       DECFSZ  R13+0, 1
;       GOTO    L_main35
;       DECFSZ  R12+0, 1
;       GOTO    L_main35
;       DECFSZ  R11+0, 1
;       GOTO    L_main35
;       NOP
;       INCF    _i+0, 1
;       GOTO    L_main32
; L_main33:
; L_main30:
;       MOVLW   1
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       CLRF    PORTB+0
;       MOVF    _countD+0, 0
;       MOVWF   FARG_LightSet_countD+0
;       MOVF    _countD+1, 0
;       MOVWF   FARG_LightSet_countD+1
;       CALL    _LightSet+0
; L_main27:
; L_main26:
;       MOVF    _countD+0, 0
;       MOVWF   FARG_LightSet_countD+0
;       MOVF    _countD+1, 0
;       MOVWF   FARG_LightSet_countD+1
;       CALL    _LightSet+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_row+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_column+0
;       MOVLW   ?lstr3_MyProject+0
;       MOVWF   FARG_Lcd_Out_text+0
;       CALL    _Lcd_Out+0
;       MOVLW   2
;       MOVWF   FARG_Lcd_Out_row+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_column+0
;       MOVLW   ?lstr4_MyProject+0
;       MOVWF   FARG_Lcd_Out_text+0
;       CALL    _Lcd_Out+0
;       MOVF    _count+0, 0
;       MOVWF   FARG_ShortToStr_input+0
;       MOVLW   _counterArray+0
;       MOVWF   FARG_ShortToStr_output+0
;       CALL    _ShortToStr+0
;       MOVLW   _counterArray+0
;       MOVWF   FARG_Ltrim_string+0
;       CALL    _Ltrim+0
;       MOVF    R0+0, 0
;       MOVWF   _t+0
;       MOVLW   2
;       MOVWF   FARG_Lcd_Out_row+0
;       MOVLW   9
;       MOVWF   FARG_Lcd_Out_column+0
;       MOVF    R0+0, 0
;       MOVWF   FARG_Lcd_Out_text+0
;       CALL    _Lcd_Out+0
;       BTFSC   PORTB+0, 7
;       GOTO    L_main36
;       MOVLW   3
;       MOVWF   R11+0
;       MOVLW   138
;       MOVWF   R12+0
;       MOVLW   85
;       MOVWF   R13+0
; L_main37:
;       DECFSZ  R13+0, 1
;       GOTO    L_main37
;       DECFSZ  R12+0, 1
;       GOTO    L_main37
;       DECFSZ  R11+0, 1
;       GOTO    L_main37
;       NOP
;       NOP
;       BTFSC   PORTB+0, 7
;       GOTO    L_main38
;       MOVLW   1
;       MOVWF   FARG_Lcd_Cmd_out_char+0
;       CALL    _Lcd_Cmd+0
;       MOVLW   3
;       MOVWF   R11+0
;       MOVLW   138
;       MOVWF   R12+0
;       MOVLW   85
;       MOVWF   R13+0
; L_main39:
;       DECFSZ  R13+0, 1
;       GOTO    L_main39
;       DECFSZ  R12+0, 1
;       GOTO    L_main39
;       DECFSZ  R11+0, 1
;       GOTO    L_main39
;       NOP
;       NOP
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_row+0
;       MOVLW   1
;       MOVWF   FARG_Lcd_Out_column+0
;       MOVLW   ?lstr5_MyProject+0
;       MOVWF   FARG_Lcd_Out_text+0
;       CALL    _Lcd_Out+0
;       CLRF    _count+0
;       CLRF    _countD+0
;       CLRF    _countD+1
; L_main38:
; L_main36:
;       MOVLW   6
;       MOVWF   R11+0
;       MOVLW   19
;       MOVWF   R12+0
;       MOVLW   173
;       MOVWF   R13+0
; L_main40:
;       DECFSZ  R13+0, 1
;       GOTO    L_main40
;       DECFSZ  R12+0, 1
;       GOTO    L_main40
;       DECFSZ  R11+0, 1
;       GOTO    L_main40
;       NOP
;       NOP
;       MOVLW   17
;       MOVWF   FARG_EEPROM_Write_Address+0
;       MOVF    _count+0, 0
;       MOVWF   FARG_EEPROM_Write_data_+0
;       CALL    _EEPROM_Write+0
;       GOTO    L_main17
; L_end_main:
;       GOTO    $+0

   END
```

ACKNOWLEDGMENT

The author would like to thank microchip for PIC tutorials.

REFERENCES

[1] Microchip Forum for PIC .

[2] Getting started with PIC Microcontroller Tutorial https://circuitdigest.com/microcontroller-projects/getting-started-with-pic-microcontroller