PRACTICAL NO: 1

Date : 8/1/25

TITLE : Interfacing LEDs with PIC 16F887 micro-controller

AIM / OBJECTIVE: To write an embedded program for PIC 16F887
To Blink different combination of LEDs

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

MPLAB IDE
Keil Software
Boot loader

CONCEPT / THEORY OF EXPERIMENT: Ports in a micro-controller are used to connect the micro-controller to the peripheral devices and to communicate with them. PIC 16F887 has 5 Ports labelled from A to E. Ports have 8 pins each except for port E. All ports are bidirectional in nature.

TRIS Register:- Used to configure port as I/P or O/P is 0x00 or 0xFF

Port Register:- Used to set the output level High / low

ADCON :- Disables all the analog inputs

PROCEDURE :

1) Open MPLAB IDE and create new project
2) Select the Mid Range 8bit MCUs
3) Select PIC 16F887 micro-controller
4) Select the XC8 compiler
5) Generate a New C file in the Source files
6) Write the code
7) Build Project (Generate hex files)
8) Open the boot loader AN1310.exe
9) Select the device and write device file

CALCULATIONS :

Configuration for Different LED combinations (8 no's leds)

1) All LEDs ON :

Hex Pattern : 0xFF: 1 1 1 1 1 1 1 1 (1 = High)

2) All LEDs OFF : 0x00; 0 0 0 0 0 0 0 0 (0 : Low)

3) Alternate LEDs : 0xAA 1 0 1 0 1 0 1 0

4) Turn on 2 LEDs : 0xCC 1 1 0 0 1 1 0 0
and Skip 2 LEDs

RESULTS :

CONCLUSION :

Successfully executed the program and verified its output on the micro-controller kit.

PRACTICAL NO: 2

Date : 15/11/25

TITLE : Display LED Blinking Patterns

AIM / OBJECTIVE : To Display various LED blinking Patterns (4)
To write an Embedded program for PIC16F887

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

MPLAB IDE
Bootloader

CONCEPT / THEORY OF EXPERIMENT: Ports in a micro-controller are used to connect the micro-controller to the peripheral devices and to communicate with them. PIC16F887 has 5 ports labelled from A to E. Port have 8 pins each except for Port E. All ports are bidirectional in Nature.

Tris Register :- Used to configure ports as IP/OP using 0x00 or 0xFF

Port Register :- Used to set the output level High / Low

ADCON :- Disable all the analog Input

PROCEDURE :

1) Open MPLAB IDE and create new project
2) Select the mid Range 8 bit MCU
3) Select PIC16F887 micro-controller
4) Select the XC8 Compiler
5) Generate a new C file in source file
6) Write the code
7) Build Project (Generate hex file)
8) Open the Bootloader AN1310.exe
9) Select the file and write the Device.

CALCULATIONS :

1) For displaying alternate LEDs

Pattern : 1010 1010 = AA

Hex Pattern : 0x AA

2) Turning on 2 LEDs and skipping 2

Pattern : 1100 1100 : CC

Hex Pattern : 0xCC

3) Turning 1 LEDs at a time from right side

0000 0001 → 0x01

0000 0010 → 0x01<<1

⋮

0000 0000 → 0xd <<7

RESULTS : 0x01 << i, where i is $(n-1)^{th}$ LED to be turned on

CONCLUSION :

Learned how to display different patterns using PIC 16F877 micro-controller

PRACTICAL NO: 3

Date : 5/2/25

TITLE : Interfacing DC motor with PIC16F887 micro-controller

AIM / OBJECTIVE: Understand the L293d motor Driver IC
Understand the Embedded program to interface DC motor with PIC16F887.

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

MPLAB IDE
Bootloader
DC 8o motor

CONCEPT / THEORY OF EXPERIMENT:

A DC motor converts Electrical energy into mechanical motion using Electromagnetic interaction. Its direction depends on the polarity of voltage applied to its terminals. A H bridge motor driver (L293D) allows switching this polarity to achieve forward and reverse rotation. L293D can drive two motors simultaneously and provides built in protection against back emf

PROCEDURE :

1) Open MPLAB IDE and create new project
2) Select the mid Range 8bit MCU
3) Select PIC16F887 micro-controller
4) Select the XC8 compiler
5) Generate a new c file in source files
6) Write the code
7) Build Project (Generate hex file)
8) 8o connect the motor
9) Open the Bootloader ANBicon
10) Select the file and write Device.

CALCULATIONS :

L293D specifications :

- 16 Pin IC
- Can Run upto 2 DC motors with Direction control
- Two H bridge circuit
- 2 output pins, 2 input pins and 1 Enable pin for each motor

PIC16F887 Specifications / features :

- 14-bit instruction set
- 368 bytes of RAM and 256 bytes of EEPROM
- 33 I/O pins and 5 I/O ports
- 3 timers : Timer0 (8 bit), Timer-1 (16-bit) and Timer-2 (8-bit)
- Watchdog timer
- 10-bit ADC
- Internal oscillator (4 MHz) and external oscillator up to 20 MHz
- Low power mode
- 8 bit DAC

RESULTS :

CONCLUSION :

Successfully Interfaced DC motor with PIC 16F887 Micro-controller and Verified its output

PRACTICAL NO: 4

Date : _____

TITLE : Interfacing LCD with PIC16F887 Micro-controller

AIM / OBJECTIVE: To configure the LCD Display
To Display the Required Data in LCD display / screen

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

MPlab IDE
Bootloader
LCD Display
Connecting wires

CONCEPT / THEORY OF EXPERIMENT:

An LCD screen is an electronic display module that uses liquid crystal to produce a visible image. The 16×2 displays 16 characters per line in 2 such lines. In this LCD each character is displayed in 5×7 pixel matrix.

2 modes:

8bit mode and 4bit mode:

PROCEDURE :

1) Open MplabIDE and create a new project
2) Select the mid Range 8bit MCU and then PIC16F887
3) Select the XC8 compiler
4) Generate a new c file
5) Write code:
   - Initialize the LCD
   - Write the function to display the data
6) Build the project (generate the hex file)
7) Connect the LCD to the required ports
8) Open the bootloader and write the device.

OBSERVATIONS

Pin no	Symbol	level	I/O	function
1	Vss	-	-	Power supply GND
2	Vcc	-	-	power supply Vcc (+5V)
3	Vee	-	-	contrast adjust
4	RS	0/1	I	0: Data mode, 1: command mode
5	Enable(E) R/W	0/1	I	0: Write, 1: Read
6	Enable (E)	0/1	I	Sends Data when high to low Pulse
7	DB0	0/1	I/O	Data Bus 1
8	DB1	0/1	I/O	Data Bus 2
9	DB2	0/1	I/O	Data Bus 3
10	DB3	0/1	I/O	Data Bus 4
11	DB4	0/1	I/O	Data Bus 5
12	DB5	0/1	I/O	Data Bus 6
13	DB6	0/1	I/O	Data Bus 7
14	DB7	0/1	I/O	Data Bus 8
15	VB1	1	-	Backlight supply
16	VB-	0	-	

Steps:

Step 1 Initialize LCD :-

- Write command display on, cursor off
- Increment shift cursor right
- Clear Display

Display 1 :- Write command 1st line (Set cursor to 1st line)
Load Ascii character from given string
Write data to LCD
Repeat above 2 steps 16 times

Display (2nd line): Set cursor to 2nd line
Load Ascii character from the string
Write data to LCD
repeat above 2 steps 16 times

CALCULATIONS :

Specifications :

- 16 characters x 2 lines
- operating voltage 4.7V to 5.3V
- logic input voltage 2.2V to 5V
- current : 1mA to 2mA
- Dot matrix : 5x8 or 5x7 pixels
- Can show custom characters (upto 8 chars via CGRAM)
- Adjustable contrast and Backlight
- 4bit mode or 8bit mode
(4bits at a time) (8 bits at a time)

Commands to set 4bit mode :

~~0x33 (8bit mode)~~
~~0x32 (4bit mode)~~
0x28 (4bit mode, 2-line, 5x8 font)
4bit mode

4bit mode :

LCD Data (D4 to D7) → Port-D (RD4 to RD7)
- LCD RS → RE0
LCD RW → RE1
LCD EN → RE2

RESULTS :

CONCLUSION :

Successfully configured the LCD display and displayed the message User

PRACTICAL NO: 5

Date : _____

TITLE : Controlling Individual LED with 8X8 RGB matrix led using arm cortex

AIM / OBJECTIVE: - To control individual LEDs on a 8X8 CC RGB Matrix using an arm cortex M3 controller

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

Universal Interface
Elprotronic experiment Board
Arm cortex m3 module
4 X (8X8 RGB led modules)

CONCEPT / THEORY OF EXPERIMENT:

In this experiment an 8x8 LED matrix is controlled by selecting individual LEDs based on binary inputs. Switches numbered from 0-9 are used to represent the binary values. By activating specific switches corresponding to the binary code, the desired LED could be switched on the matrix. This enables displaying particular LED one by one according to number.

PROCEDURE :

1) Open Keil uVision Software
2) Connect the RGB matrix with arm cortex and other connection
3) Connect the USB to arm cortex M3
4) Open Coocox IDE in tools
5) Open the project DMX Example
6) Build the project
7) Download code to flash
8) Change the position of Individual LEDs using DIP switches.

OBSERVATIONS

RGB LED module	Address	DIP switches (1-9)
Top left	0	000000000
Top Right	64	000000100
Bottom left	256	100000000
Bottom Right	320	101000000

The Dip switches are arranged in powers of 2

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9
1	2	4	8	16	32	64	128	256	512

- The led starts from Top left (0) bit and moves on till the bottom right (68) bit it counts from 1 to 64
- After 64 the array displayed starts from next panel and same format as above is used

CALCULATIONS :

Switching on the marked LED

ex bit 32:

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9
1	0	0	0	0	1	0	0	0	0

ie 1000 0010 0

The above order of switches will lit the 33rd led in the matrix

RESULTS :

Individual leds are controlled accurately using the Dip switches and arm coder

CONCLUSION :

Successfully controlled individual LEDs leds using the arm cortex M3 micro-controller.

PRACTICAL NO: 6

Date : _____

TITLE : Displaying Images using 8×8 RGB matrix by using arm cortex M3 module

AIM / OBJECTIVE:

To Display simple images on an 8×8 led matrix using arm micro-controller

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

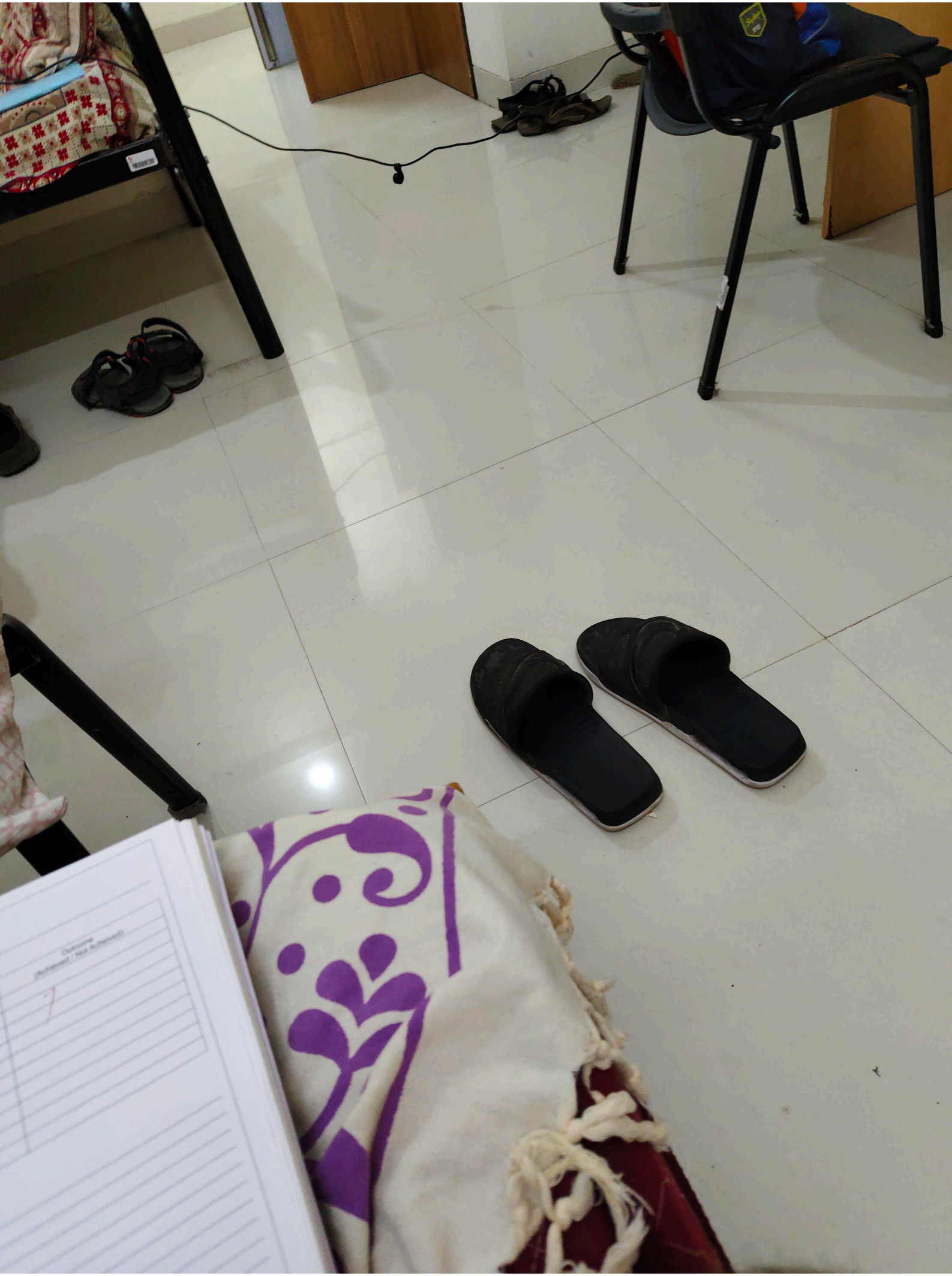
Unitrain Interface
Elotrain Experiment Board
Arm cortex M3 module
4× (8×8 RGB led module)

CONCEPT / THEORY OF EXPERIMENT:

In this experiment the custom images are programmed using by changing the Dmx array in user_data.h file. There are 512 bytes corresponding to Eight (8×8 RGB matrix). By chaing changing the pattern in one of 8×8 bytes, we can display seperate images in each 8×8 modules.

PROCEDURE :

1) Open Keil uVision software
2) Connect the RGB matrix with controller then with arm cortex
3) Connect the USB to arm cortex M3
4) Open code composer IDE in tools
5) Open project Dmx examples
6) In user_data.h change the code as per the required led pattern
7) Build the project
8) Download code to flash



PRACTICAL NO: 7

Date : _____

TITLE : Displaying moving images on 8x8 RGB matrix using Arm cortex M3 micro-controller

AIM / OBJECTIVE:

To display moving images on a 8x8 RGB LED matrix using arm cortex M3 micro-controller

APPARATUS / TOOLS / EQUIPMENT / RESOURCES USED:

Ultrain Interface
Ultrain Experimental Board
Arm Cortex M3 micro-controller
4x (8x8 LED matrix)

CONCEPT / THEORY OF EXPERIMENT:

In this experiment the switches are used to select till which pattern the RGB array is traversed. Each pattern is spaced at fixed interval with new pattern 'number' coming after 64. If the Dip switches are set to 128 then the array is traversed till 128 bytes, thus displaying 3 frames as number.

PROCEDURE :

1) Open Keil uVision software
2) Connect the RGB matrix with controller and arm cortex
3) Connect the USB to arm cortex
4) Open Coocox IDE
5) Open project DMX Excample 3 or 4
6) Build the project
7) Download code to flash.