**Guidelines to use Silicon Lab’s Simplicity Studio**

Simplicity Studio simplifies the Embedded development process with one-click access to everything developers need to complete their projects using an integrated development environment (IDE) based on Eclipse 4.5.

**Steps to create project and program compilation:**

**Steps:**

* Open Simplicity Studio.
* Click File > **Import…**, to import the project.
* Click General > Existing Projects into Workspace and click Next.
* First Select Root Directory of your Eclipse Template for EPB\_F340. Then select “Copy projects into workspace” check box and Click Finish.
* Every time you import a project make sure to rename it. So **Right Click Project > Rename** or press **F2** while selecting project to rename it.
* To write source files, right click on Sources, go to New > Source File and you will see New Source File Wizard. Enter Source File name (For example **main.c**) then Click Finish.
* To write header files, right click on Includes, go to New > Header File and you will see New Header File Wizard. Enter Header File name (For example **main.h**) then Click Finish.
* Write your code and then save your Files.

**OR**

* Copy necessary .c and .h files to your local Sources and Includes project folders respectively. They will be added to your project in Simplicity Studio IDE.
* Click on Build Project in **Project>Build Project** to compile and build the project.
* After successful building the project the .hex file will be generated in the project folder in the workspace.

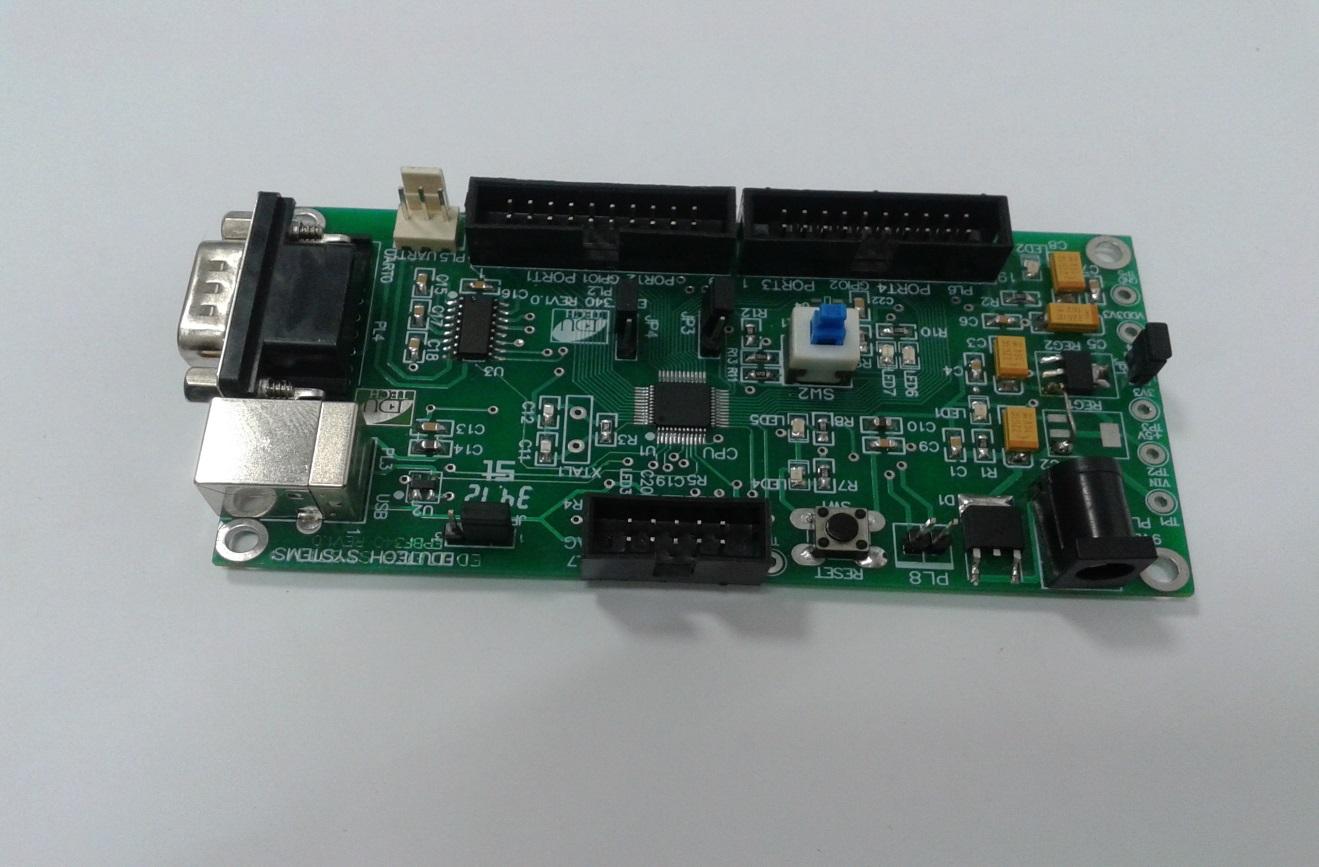
**Steps to use hardware:**

* Connect 9V DC Power supply to the educational practice board for F340
* Connect USB cable between PL4 connector of EPBF340 board and PC.
* Using the RUN/PROGRAM mode selection switch set the board in the program mode by pressing switch SW2 and press reset.
* Using download tool (USB Bootloader) download the .HEX file to the target board.
* Wait for the “Successfully loaded image” message in display.
* Connect flat cable between ASK25 and EPBF340 boards according to each experiment’s requirement.
* Using the RUN/PROGRAM mode selection switch, set the board in the run mode by releasing Switch SW2. Apply reset to execute the program.
* Observe the expected output of the experiment.

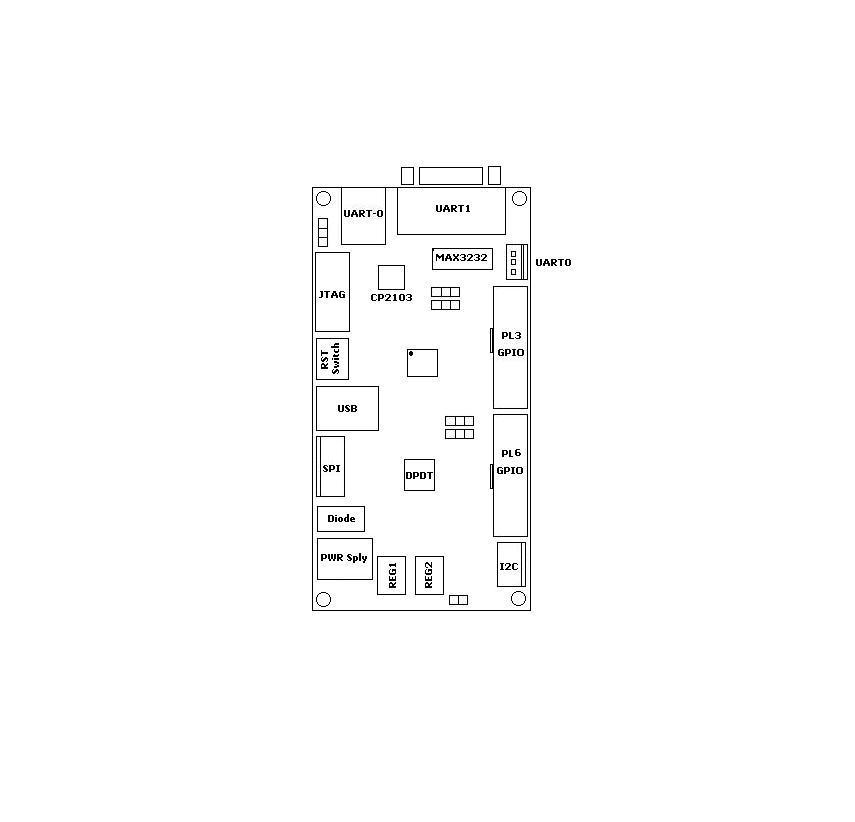
***NOTE:*** *Use this tool for Experiment No. 3 to 10. Draw interfacing diagram on separate sheet and attach the printout of the tested code.*

**F340 Hardware**

The EPB-F340Mini is a stand-alone card--allowing developers to evaluate the C8051F340 USB Flash MCU Family to determine if it meets their application requirements. Furthermore, the module is an excellent platform to develop and run software for the 8051 processor. The EPB-F340Mini is shipped with a C8051F340. The EPB-F340Mini allows full speed verification of 8051 code. In addition, an onboard C2 connector provides interface to emulators, with assembly language and ‘C’ high level language debug.



*Figure 3.1: The EPB-F340Mini development board*

******

*Figure 3.2 The EPB-F340Mini development board layout*

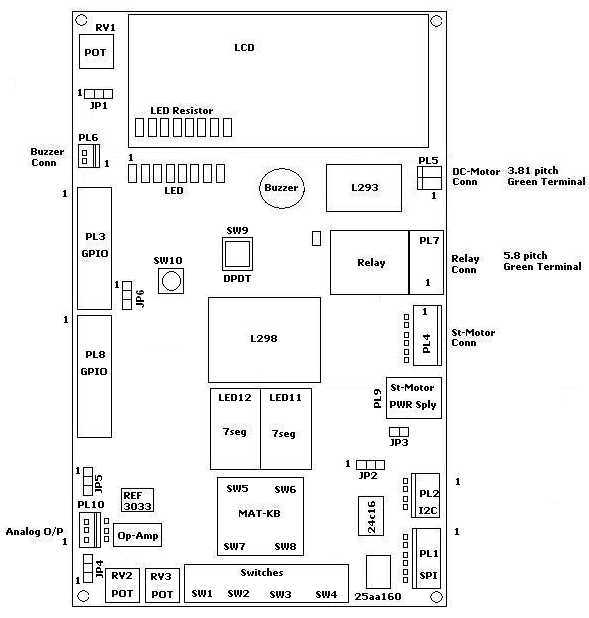
The EPB-F340Mini has 8 connectors. The function of each connector is shown in the table below:

Table 3.1 EPB-F340Mini connector description

|  |  |  |
| --- | --- | --- |
| **Unit** | **Reference** | **Description** |
| Power jack connector | PL1 | Power jack connector |
| I2C | PL2 | 4 Pins |
| I/O Port | PL3 | PORT1 and 2 GPIO |
| USB | PL4 | USB Connector |
| JTAG | PL5 | JTAG Interface |
| I/O Port | PL6 | PORT3 and 4 GPIO |
| SPI | PL7 | 6 Pins |
| UART0/Debug | PL8 | USB Connector |
| UART0 | PL9 | 3 Pins Connector |
| UART1 | PL10 | DB9 -M Connector |

**General Purpose I/O Interfacing Kit – ASK 25A**

The general purpose I/O interfacing Board is a generic board which focuses on the interfacing of different input and out devices to microcontroller. The board is populated with variety of devices like LED, Key, Relay, LCD, Signal conditioning circuit for ADC, DC motor interface, Stepper Motor interface, I2C EEROM, SPI EEPROM, 2x2 Matrix Key board etc.



*Figure 3.3 The ASK-25 board layout*

The ASK-25A has 10 connectors. The function of each connector is shown in the table below:

Table 3.2 ASK-25A connector description

|  |  |  |
| --- | --- | --- |
| **Unit** | **Reference** | **Description** |
| SPI | PL1 | 6 Pins |
| I2C | PL2 | 4 Pins |
| I/O Port | PL3 | GPIO |
| Stepper Motor Connector | PL4 | 6 Pins Connector |
| DC Motor Connector | PL5 | 2 Pin Green Terminal |
| Buzzer Connector | PL6 | 2 Pin Connector |
| Relay Connector | PL7 | 3 Pin Green Terminal |
| I/O Ports | PL8 | GPIO |
| Stepper Motor Power Supply | PL9 | Power Socket |
| Analog Output | PL10 | 3 Pin Connector |

**T. Y. B. Tech (Electrical and Computer Engineering)**

**Trimester: V Subject: Microcontroller and Applications**

**Name: Shreerang Mhatre Class: TY**

**Roll No: 52 Batch: A3**

**Experiment No: 03**

**Name of the Experiment:** Interfacing of LED, Buzzer, Relay and Switch with C8051F340



**Performed on: 03/10/2023**

**Submitted on: 07/10/2023**



**Aim:** Write C program for interfacing of LED, Buzzer, Relay and Switch with C8051F340 to turn it ON when key is pressed.

**Apparatus:** EPBF340 Board, ASK25 board, Connectors

**Theory:**

**Ports in C8051F340:**

Digital and analog resources are available through 40 I/O pins. These pins are available on ports. C8051F340 has 5 ports, each port has eight pins. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Registers XBR0, XBR1, and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins.

**Port I/O Initialization**

Port I/O initialization consists of the following steps:

Step 1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).

Step 2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).

Step 3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).

Step 4. Assign Port pins to desired peripherals (XBR0, XBR1) as shown in figure 2.3.

Step 5. Enable the Crossbar (XBARE = ‘1’).

**SFR definitions:**

1. **PnMDIN: Portn Input Mode**

0: Corresponding P0.n pin is configured as an analog input.

1: Corresponding P0.n pin is not configured as an analog input.

1. **PnMDOUT: Portn Output Mode**

0: Corresponding P0.n Output is open-drain.

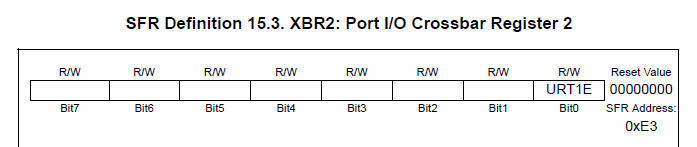
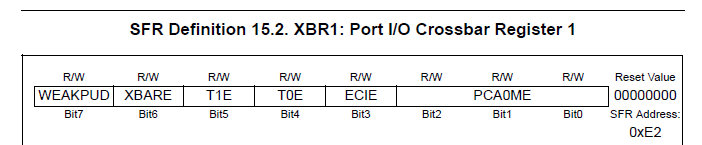
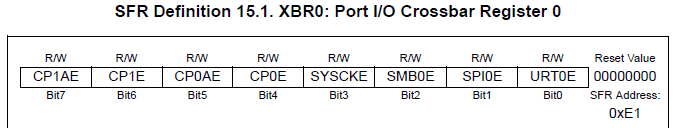
1: Corresponding P0.n Output is push-pull.

1. **PnSKIP: Portn Skip**

0: Corresponding P0.n pin is not skipped by the Crossbar.

1: Corresponding P0.n pin is skipped by the Crossbar.

1. **XBR0, XBR1, and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins**



*Figure 3.3 Crossbar registers*

**LED, Buzzer, Relay and Switch:**

LED: LED is an output device. 8 LED’s are connected to 8 port pins. The LED’s are connected in common anode configuration. Thus, to turn ON the LED logic ‘0’ must be given and to turn OFF the led logic ‘1’ must be given.

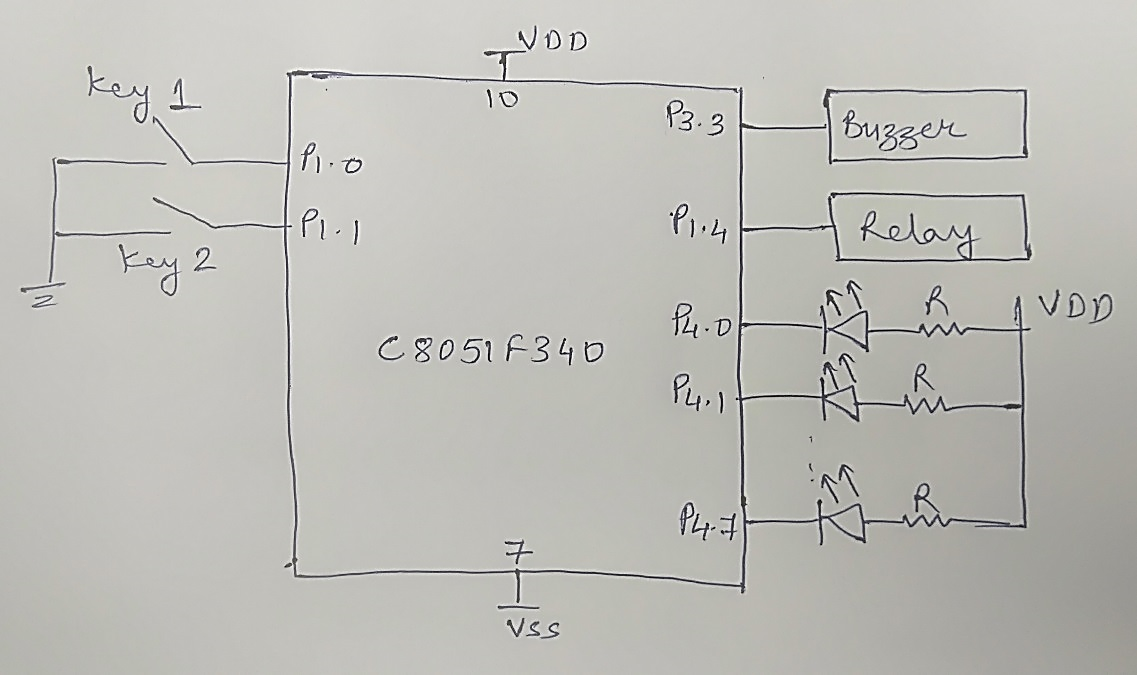
Buzzer: Buzzer is an output device. The buzzer is turned ON when logic ‘1’ is applied to the port pin and turned OFF when logic ‘0’ is applied to port pin.

Relay: Relay is an output device. The relay is turned ON when logic ‘0’ is applied to port pin and turned OFF when logic ‘1’ is applied to the port pin.

Switch/button: Switch is an input device. Push button switches are used. When the switch is released the port pin has logic ‘1’ and when the switch is pushed the port pin has logic ‘0’.

**Algorithm:**

**Interfacing Diagram:**

****

*Figure 3.4 LED, Buzzer, Relay, and Switch Interfacing Diagram*

**Hardware Connections:**

Connect flat cable between PL8 connector of ASK25 and PL6 connector of EPBF340 board. Connect other flat cable between PL3 connector of ASK25 and PL3 connector of EPBF340 board.

Table 3.3 Hardware connections between EPBF340 and ASK25 board

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pin Connection** | **PL8 Connector of ASK25** | **PL6 Connector of EPBF340** | **PL3 Connector of ASK25** | **PL3 Connector of EPBF340** |
| 1 |  |  | SW1 | P1.0 |
| 2 |  |  | SW2 | P1.1 |
| 3 |  |  | SW3 | P1.2 |
| 4 | BUZZER | P3.3 | SW4 | P1.3 |
| 5 |  |  | RELAY | P1.4 |
| 10 | LED1 | P4.0 |  |  |
| 11 | LED2 | P4.1 |  |  |
| 12 | LED3 | P4.2 |  |  |
| 13 | LED4 | P4.3 |  |  |
| 14 | LED5 | P4.4 |  |  |
| 15 | LED6 | P4.5 |  |  |
| 16 | LED7 | P4.6 |  |  |
| 17 | LED8 | P4.7 |  |  |
| 18 |  | 3.3 V |  | 3.3 V |
| 19 | 5V | 5.0 V | 5V | 5.0 V |
| 20 | GROUND | GND | GROUND | GND |

**Program:** Attach printout of the tested code.

**Expected Result:**

LEDs, Buzzer and Relay should turn on when respective key is pressed.

**Conclusion:**

**---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

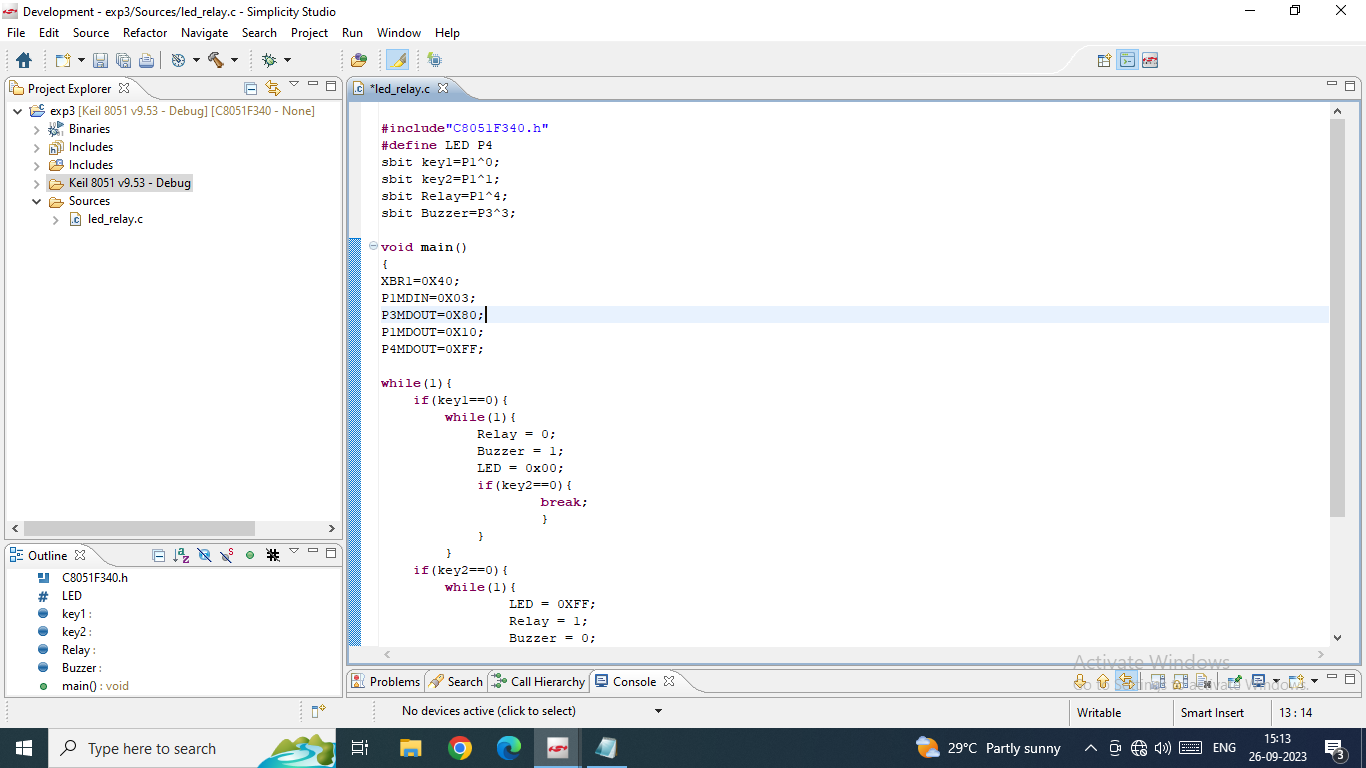
**Study Question:**

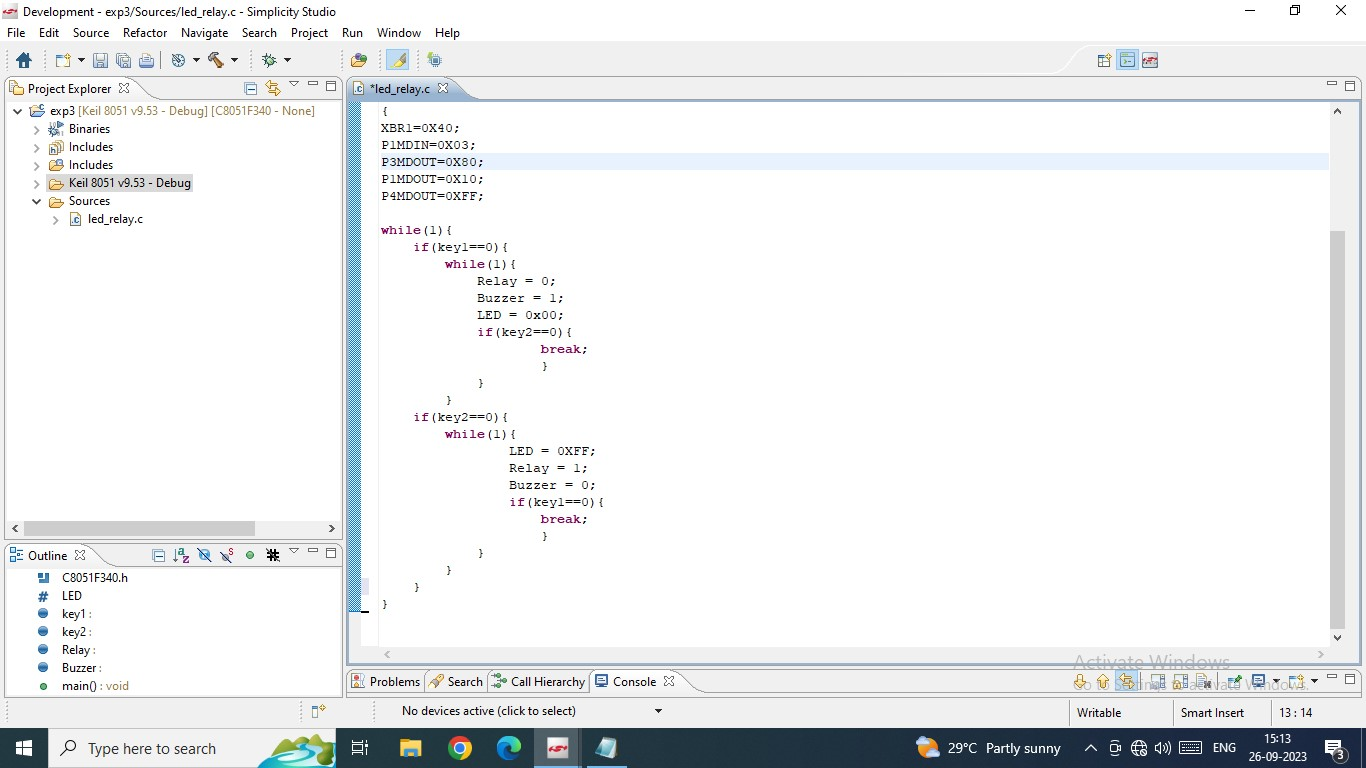
1. Explain the common anode and common cathode configuration of LED.
2. How many ports are available in C8051F340?
3. How to configure port as an input/output?
4. Explain Priority Crossbar Decoder.

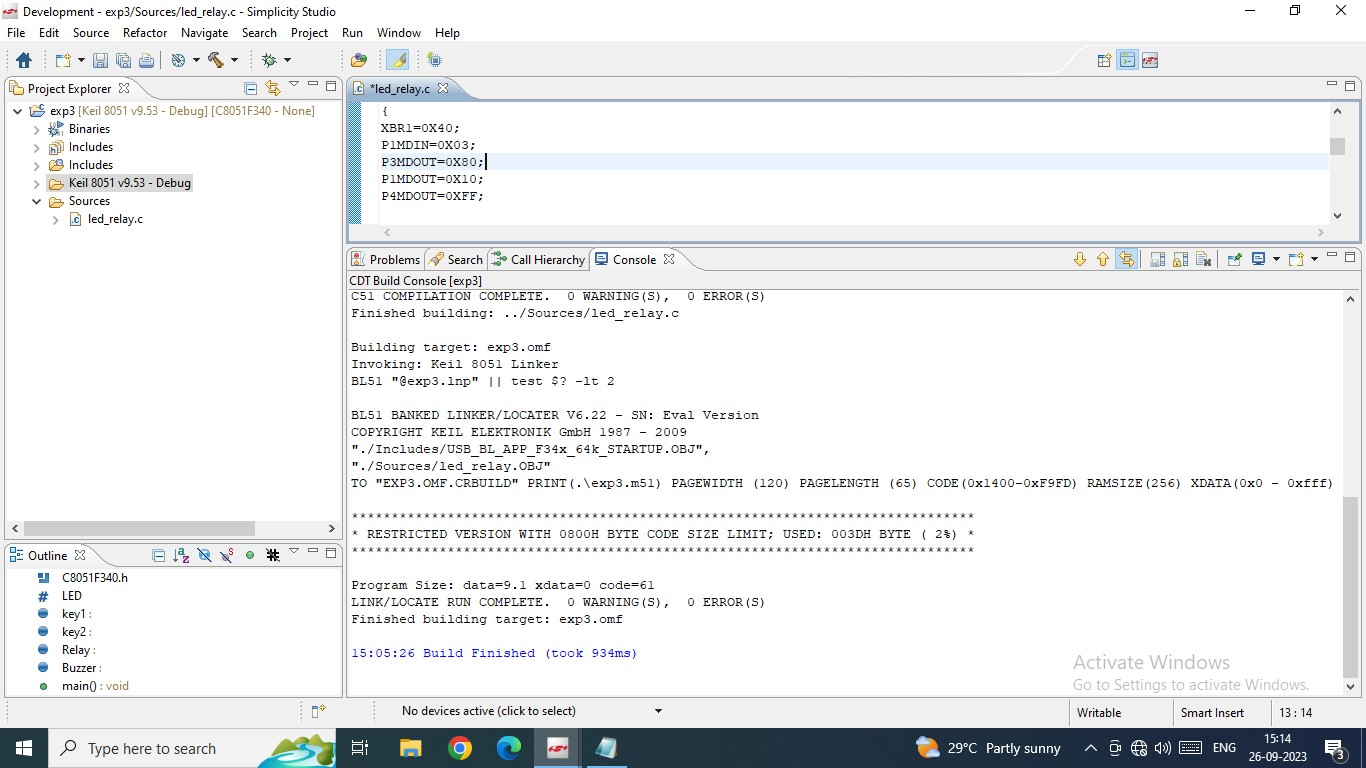
**Additional links:**

1. [*https://www.silabs.com/documents/public/data-sheets/C8051F34x.pdf*](https://www.silabs.com/documents/public/data-sheets/C8051F34x.pdf)
2. *https://aticleworld.com/interfacing-of-switch-and-led-using-the-8051/*

**CODE:**

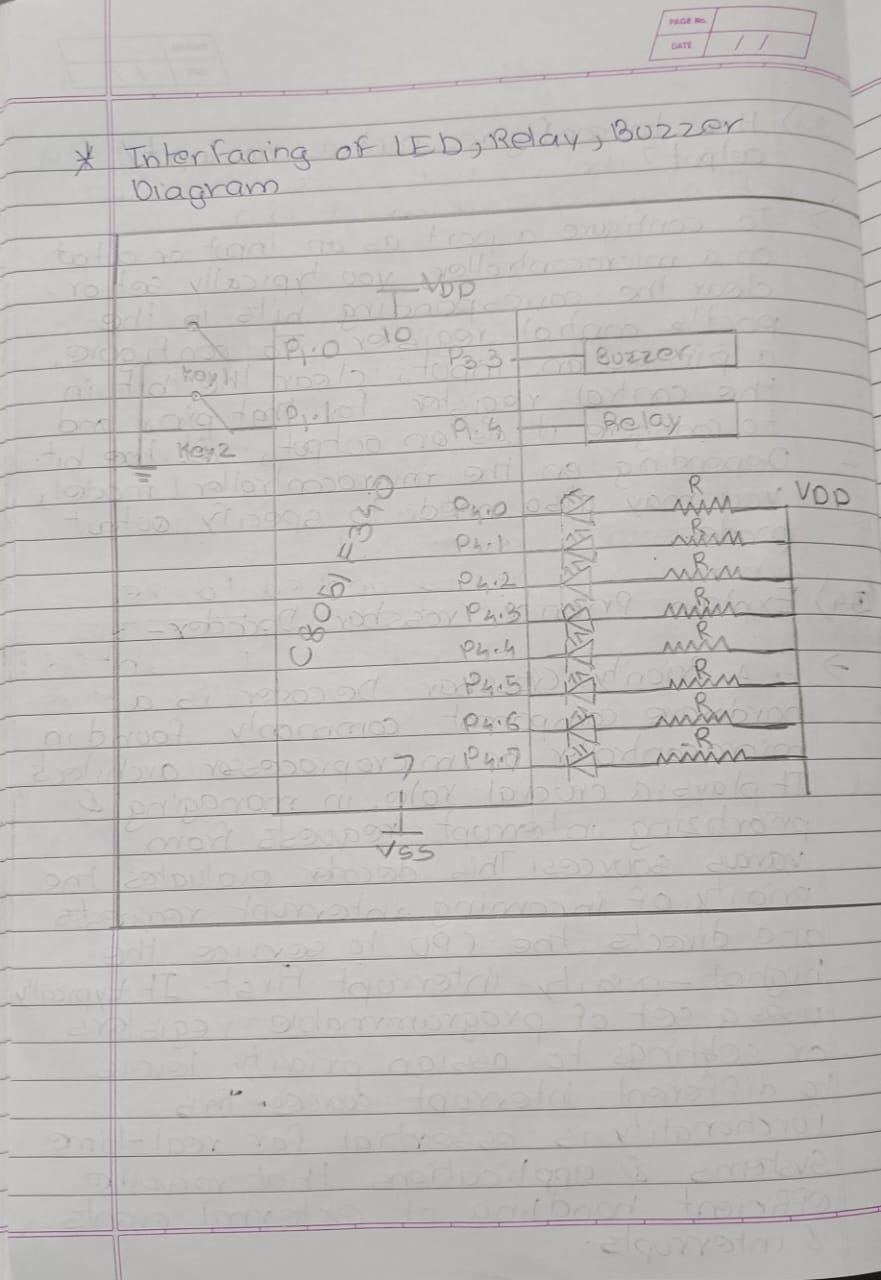


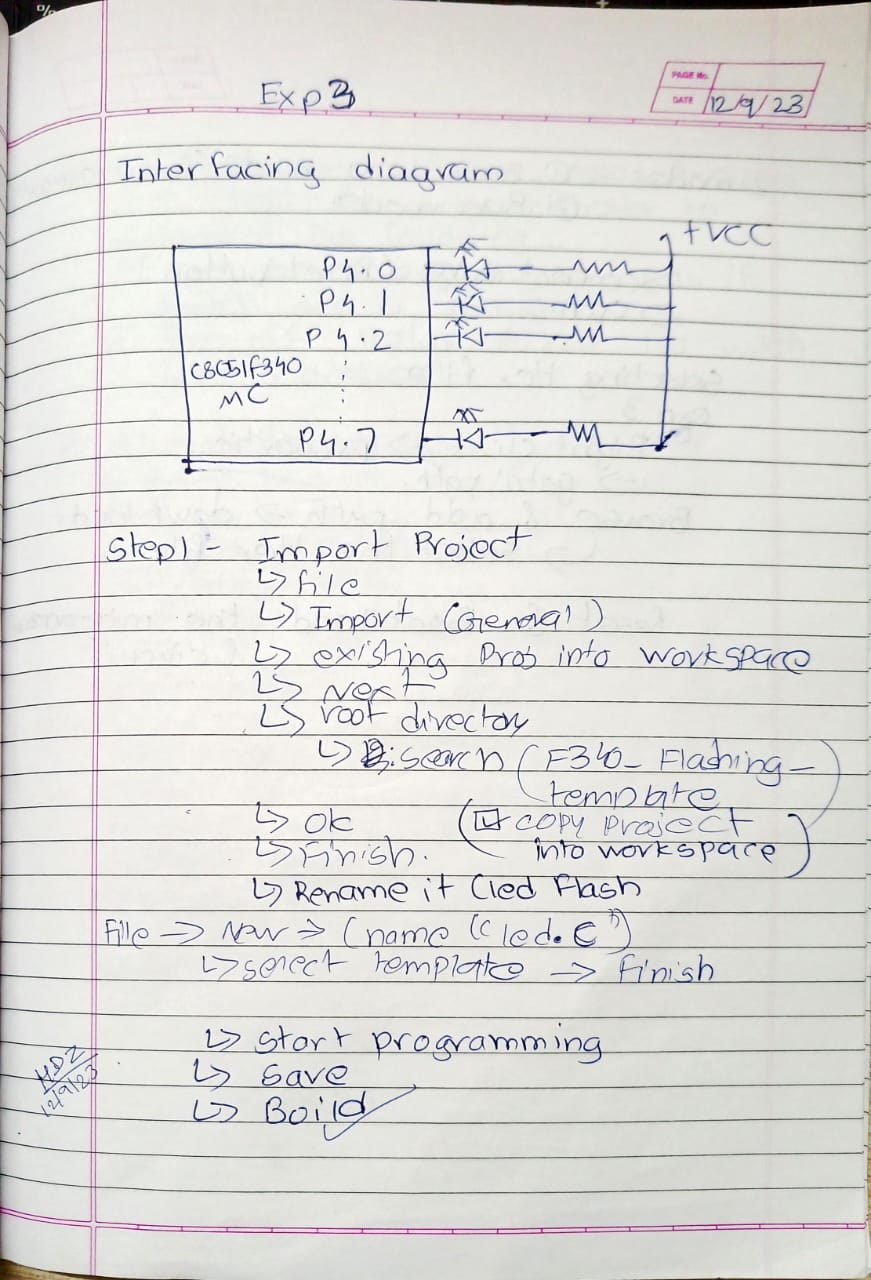


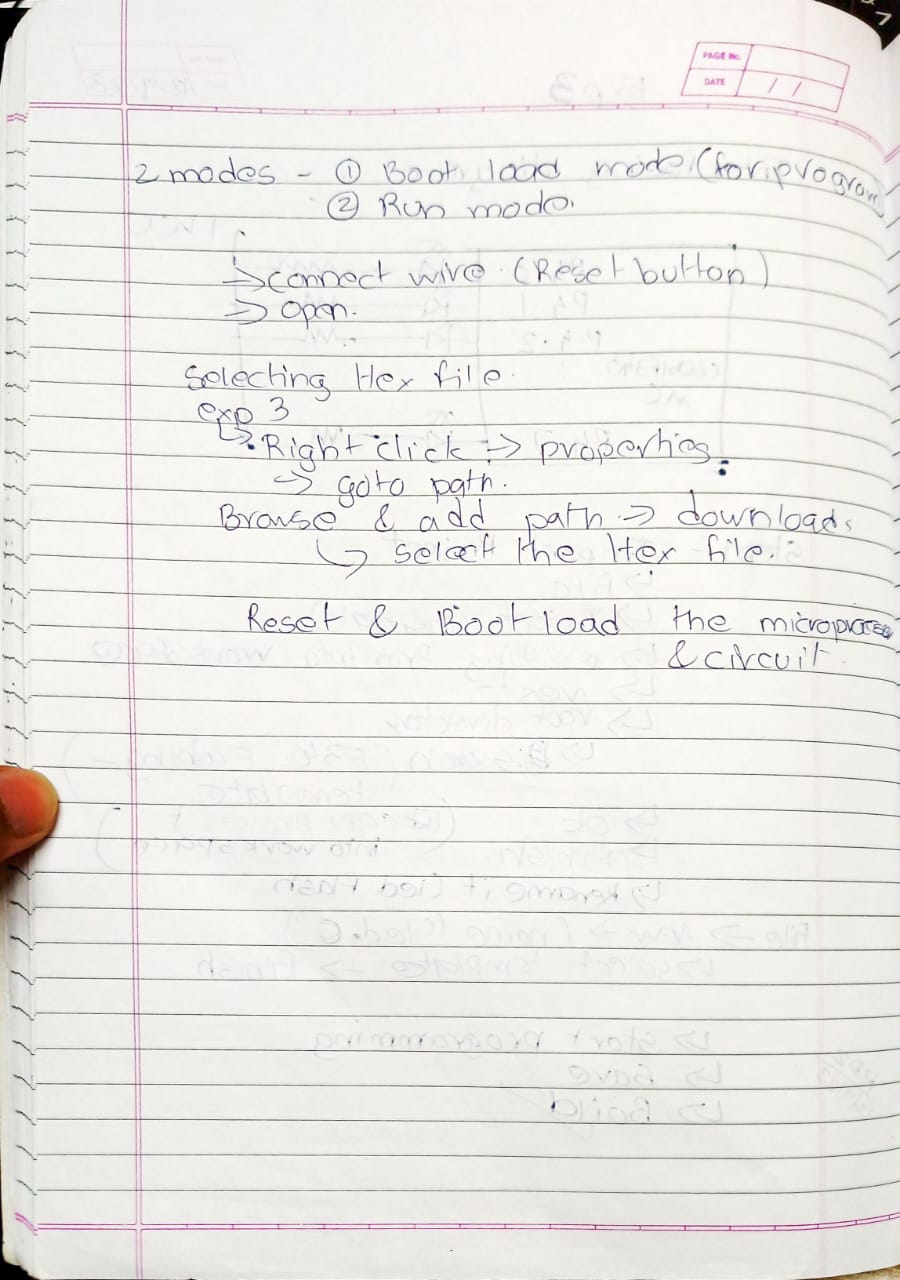


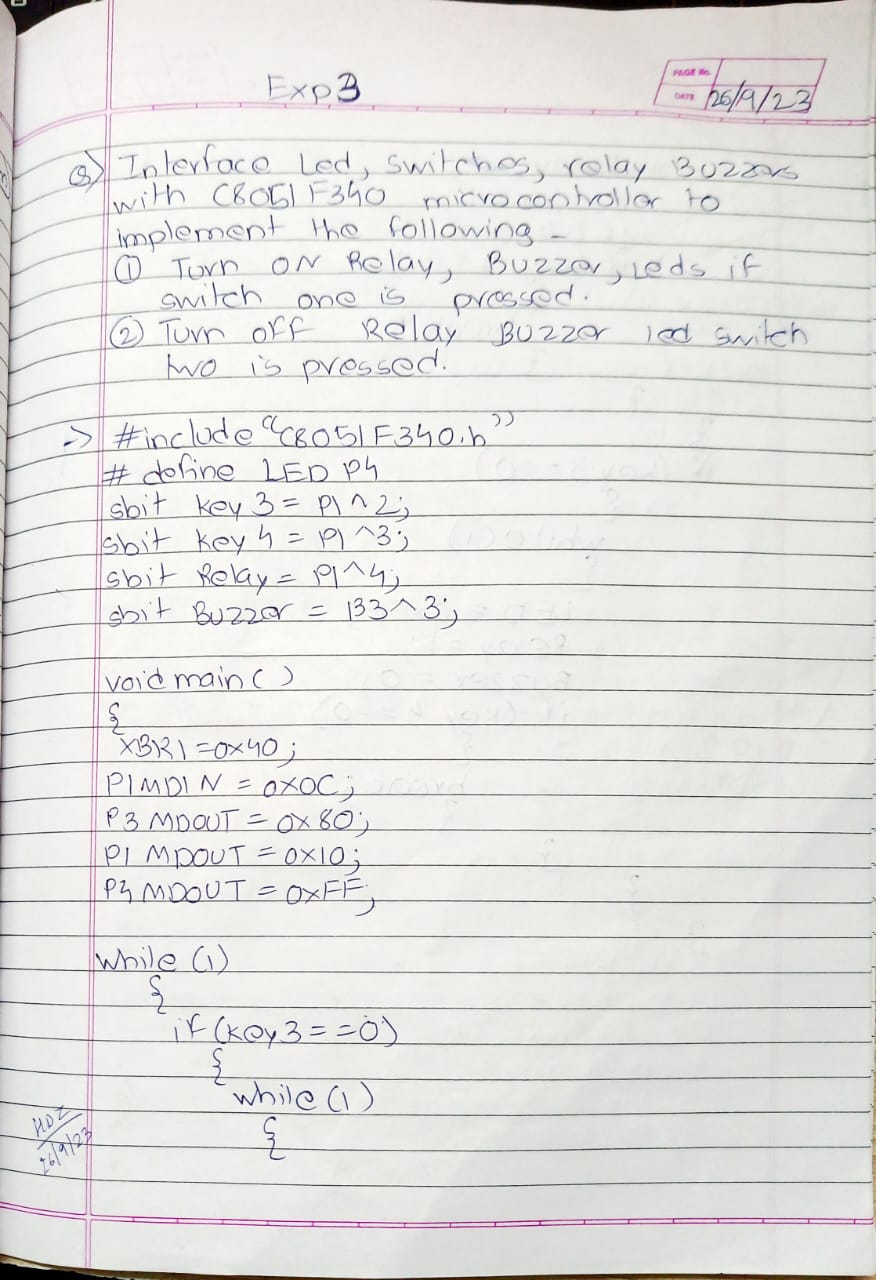


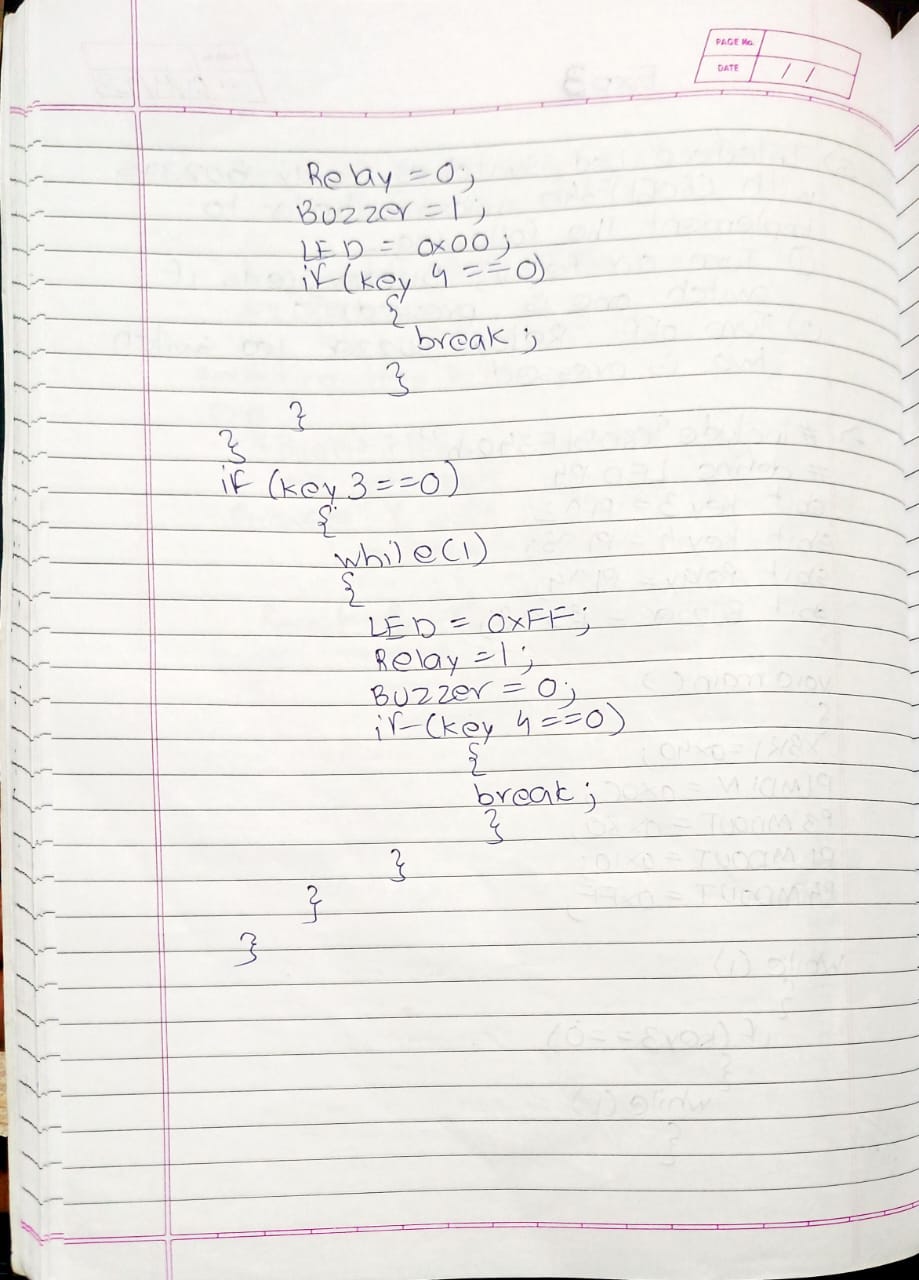












**POST LAB QUESTIONS:**

