"Heaven's light is our guide"

Rajshahi University of Engineering & Technology

Department of Electrical & Electronic Engineering

Course No : EEE 4210

Course Title : Embedded System Design Sessional

Experiment no : 01

Name of the Experiment: To get familiar with edsim51 software

Date of Experiment: October 04, 2023

Date of Submission: October 18, 2023

Submitted By

Md Maruf Hassan

Department : EEE

Roll : 1801105

Section : B

Submitted To

Md Mayenul Islam

Lecturer,

Department of EEE,

RUET

Experiment No.: 01

Name of the Experiment: To get familiar with edsim51 software

Objectives:

- To observe the interface of EdSim51 software
- To learn different section in edsim51 software

Introduction:

Edsim51 is a widely used simulation software for programming of 8051.The several parts of edsim51 software:

1. Registers and Internal RAM

2. Program section

3. Port connections

4. Sensors and Actuators (7 segment display, LCD, Converters, keyboard etc.)

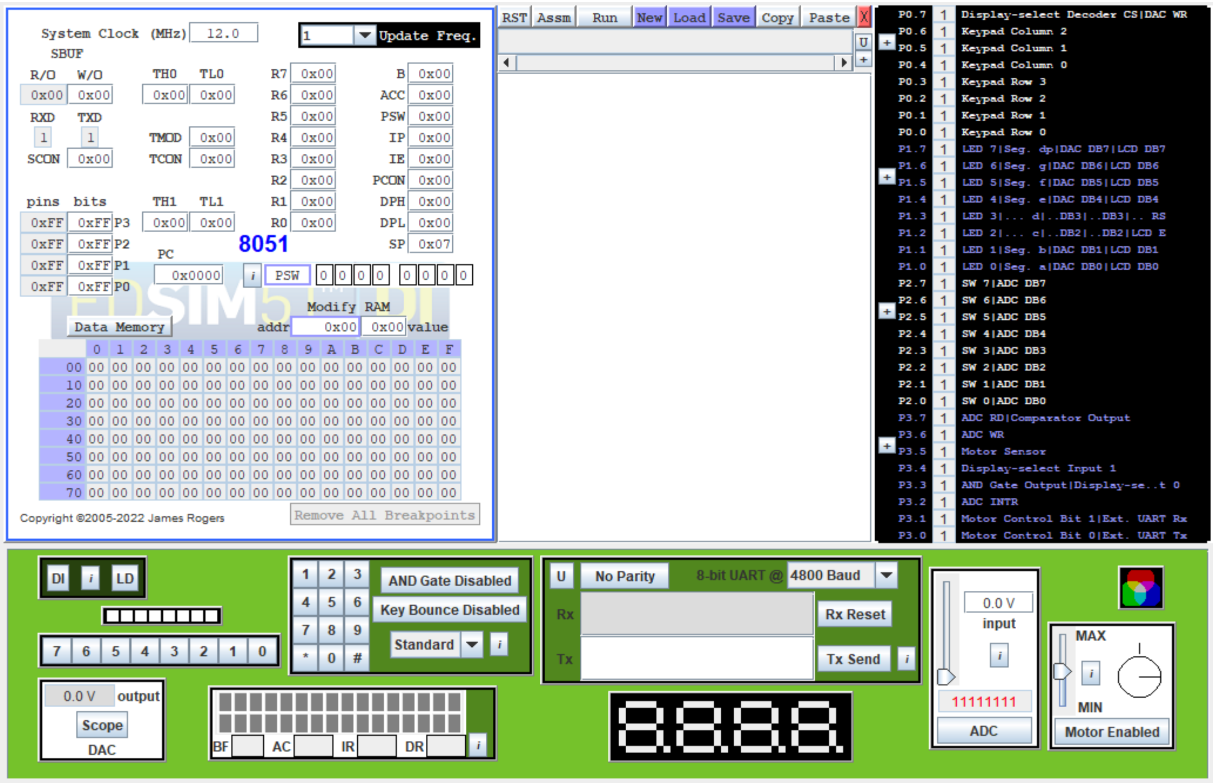


Fig. 1.1. Initial interface of edsim51 software

- The top left box gives the user access to all the 8051's registers, data memory and code memory.

- In the center, there is a textbox where the user either loads an assembly program or writes the code directly. Shown above is a program being single-stepped (execution is currently at location 0034H in code memory - hence that line is highlighted).

- On the right is a list of the 32 port pins and what each one is connected to. The current value of the port pin is displayed here.
- The bottom panel shows all the peripherals that are connected to the 8051.

Description of Different Panels of the Interface:

1. The Microcontroller Panel:

This gives the user access to all the 8051's registers and data memory. Boxes that are white can be edited directly. Those that are grey cannot. For example, the port latch bits can be edited directly by the user, but the port pins are controlled by the external peripherals and the port latches and cannot be edited.

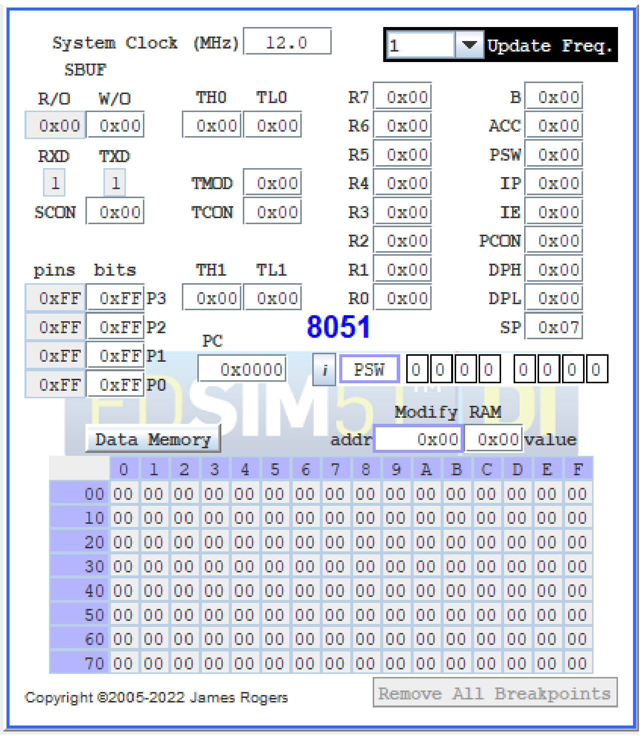


Fig. 1.2. The Microcontroller Panel

a) The Bitfield:

In the above image, the individual bits for the accumulator are shown (ACC). The user can enter any address or SFR name in the blue box (replacing ACC) and the bits for that given address will then be displayed.

b) Data and Code Memory:

By default, data memory is displayed. Any address in RAM (00H to 7FH) can be altered by entering the address in the blue box (labelled **addr**) and then entering the desired value in the box to the right (labelled **value**).

2. The Assembly Code Panel:

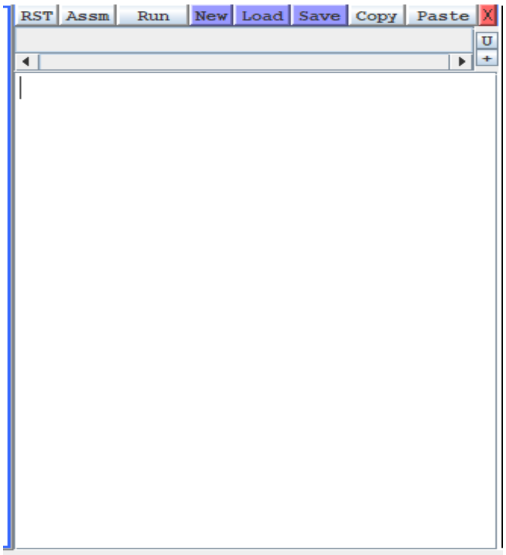


Fig. 1.3. The Assembly Code Panel

When the background of the assembly code text area is white is it editable. The programmer can write code directly here, or can load a program from file using the **Load** button (dealt with in the next section).

When the program is ready for testing, the user can either click on the **Assm** button to execute instructions one at a time, or on the **Run** button to run the program continuously. Either way, the program will first be assembled. If an error in the code is discovered, a message is displayed in the message box above the assembly code (with a red background) and the line with the error is highlighted within the code in red.

If the code assembles without errors, **Assm** is replaced by **Step**, the text area's background changes to light grey. The code cannot be edited at this point.

3. The Peripherals Panel:

This panel consists of the following segments –

- ADC and Switch Bank
- DAC (output displayed on oscilloscope)
- LED Bank
- Four 7-segment LED Displays
- LCD Module
- Bi-directional Motor
- Comparator
- UART
- Keypad

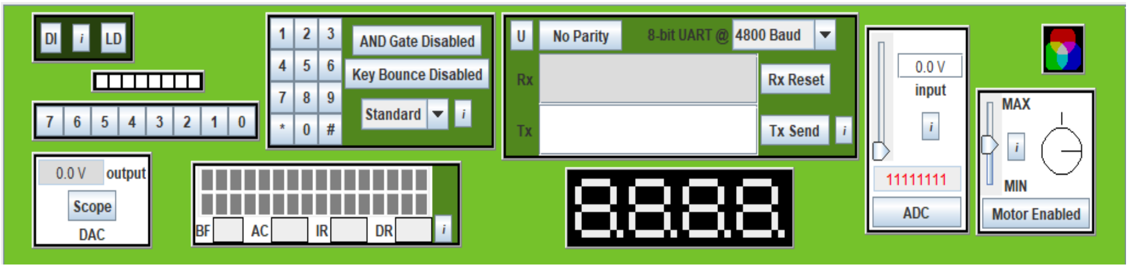


Fig. 1.4. The Peripherals Panel

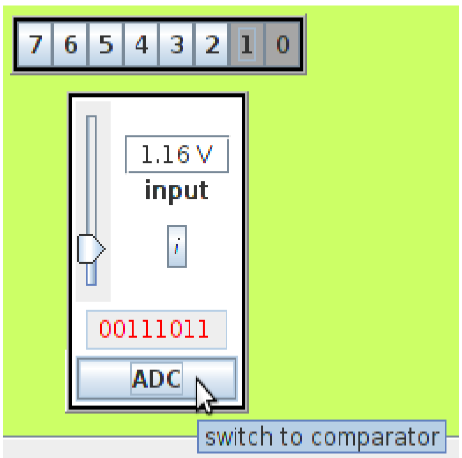


Fig. 1.4.1. Switch Bank and ADC Panel

The outputs of the ADC are tri-state: the RD line, which is connected to P3.7, must be low for the ADC reading to appear on the outputs. The WR line (connected to P3.6) is used to initiate a conversion. As it is positive edge triggered, it must be taken low and then high to start a conversion.

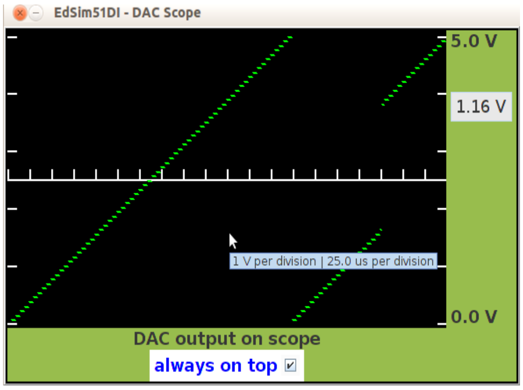


Fig. 1.4.2. DAC

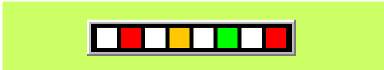


Fig. 1.4.3. LED Bank

Fig. 1.4.4. 7-Segment Display

The LED bank, the DAC inputs and the 7-segment display data lines all share port 1. The selection of which of the four displays is enabled is done via P3.3 and P3.4. These port pins are applied to a 2-to-4 line decoder, the outputs of which are applied to the base of transistors that enable/disable the displays.

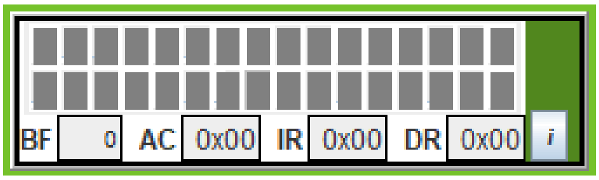


Fig. 1.4.5. The LCD Module

The LCD Module also shares port 1 with the LEDs and DAC.

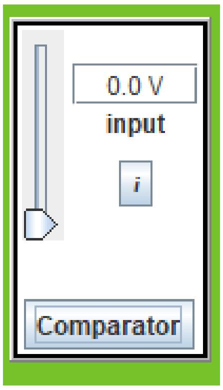


Fig. 1.4.6. Comparator Panel

The analogue input that is applied to the ADC is also applied to the non-inverting pin of the comparator, as can be seen above (in this extract, since the ADC is disabled, the analogue voltage connection to the ADC is omitted). When the comparator is enabled, the button's label says Comparator, as can be seen in the image opposite. Hovering over the button displays a tip - click the button to disable the comparator and enable the ADC. The button corresponds to the switch at the ADC chip select and the switch between the comparator output and P3.7 in the logic diagram extract above. You can see the switch between the comparator output and P3.7 is closed while at the same time the ADC CS line is switched to +V, disabling the ADC.

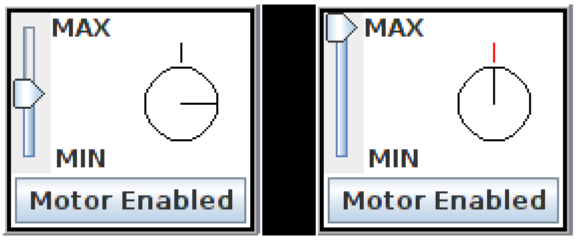


Fig. 1.4.7. Motor Panel

The motor sensor, which is applied to P3.5, goes low once every revolution (in the simulator, whenever the motor shaft lines up with the sensor, the sensor changes from black to red and P3.5 goes to logic 0). P3.5 is the external clock source for timer 1. Therefore, code can be written that, using timer 1, counts the motor's revolutions. The speed of the motor can be varied manually (using the slider to the right of the motor take a look at the hardware screenshots above). This will make the rev. counting programs more interesting.

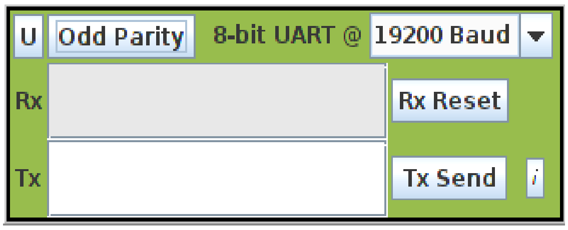


Fig. 1.4.8. External UART Panel

As stated above, the motor control lines share the same port pins as the 8051-serial port RXD and TXD. An external UART is connected to P3.0 and P3.1.Data received from the 8051's serial port appears in the Rx window. The data in this window can be cleared at any time by clicking the Rx Reset button.

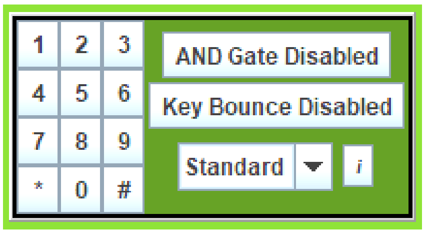


Fig. 1.4.9. The Keypad

The 4 X 3 keypad is interfaced in a standard format, as can be seen in the logic diagram extract above. All of port 0's pins, except pin 7, are used by the keypad.

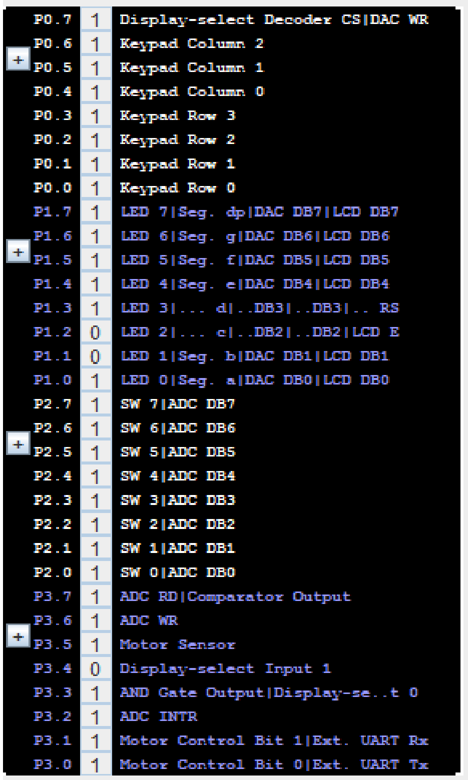


Fig. 1.5. 32 Port Pins

Discussion:

In this experiment, the EdSim51 software's interface was launched after installation. The microcontroller panel, the assembly code panel, and the peripherals panel were the interface's three primary panels. There were various sub-panels in each panel. All of the sub-panels' functions were explained previously, and they were all learned. So, it can be concluded that this experiment was successful in introducing participants to the EdSim51 software.

“Heaven’s light is our guide”

Rajshahi University of Engineering & Technology

Department of Electrical & Electronic Engineering

Course No : EEE 4210

Course Title : Embedded System Design Sessional

Experiment no : 02

Name of the Experiment: Observation of 7-segment Display Interfacing of 8051 Microcontroller Using EdSim51 Software

Date of Experiment: October 04, 2023

Date of Submission: October 18, 2023

Submitted By

Md Maruf Hassan

Department : EEE

Roll : 1801105

Section : B

Submitted To

Md Mayenul Islam

Lecturer,

Department of EEE,

RUET

Experiment No.: 02

Name of the Experiment: Observation of 7-segment Display Interfacing of 8051 Microcontroller Using EdSim51 Software

Theory:

The interfacing connection between the 8051 Microcontroller and the 7-segment displays is according to the following figure.

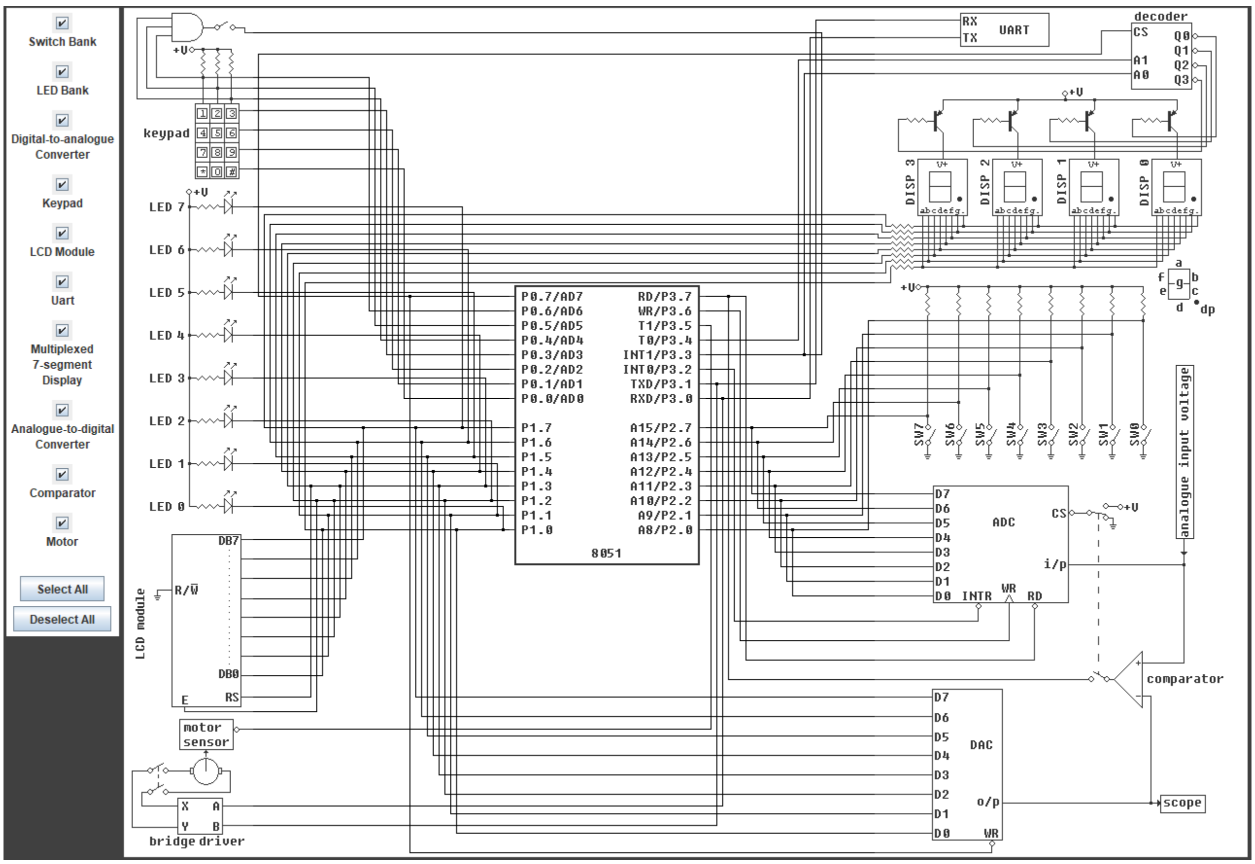


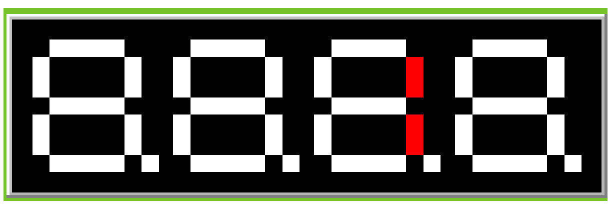
Fig. 01: 7-segment display interfacing with 8051 Microcontroller

There are 4 displays (DISP 0 – DISP 3) for 7-segment display. The segments of each display are connected to the LEDs. Thus, controlling the bits of port-1, the output of any 7-segment display can be controlled. To select a particular display among the 4 displays, the chip selects bit (CS) of the decoder, connected to P0.7, is always kept high and the other inputs of the decoder, A0 and A1 connected to P3.3 and P3.4 respectively, are to be designed accordingly.

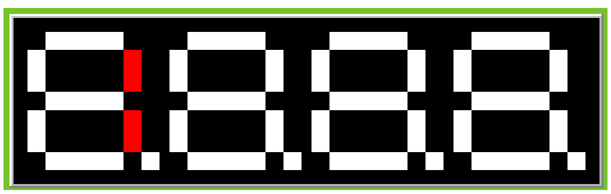
Code:

```
CLR P3.4
SETB P3.3

CLR P1.1
CLR P1.2
LOOP: MOV A,#0FFH
                  DEC A
JZ LOOP1

LOOP1:
SETB P3.4
SETB P3.3

CLR P1.1
CLR P1.2
```

Output:



(a)



(b)

Fig. 02: Display output showing last two digit of ID 1801105 i.e., 11

Discussion and Conclusion:

In this experiment, many activities involving the LEDs and the 7-segment display were carried out once the interface connection was observed. The preceding section displays the appropriate programs and outputs. The results were achieved in accordance with the expected outcomes. Therefore, it can be concluded that the observation of the 8051 microcontroller's interface with a 7-segment LED display using edsim51 software was successful.