ECE 4010 Embedded Systems

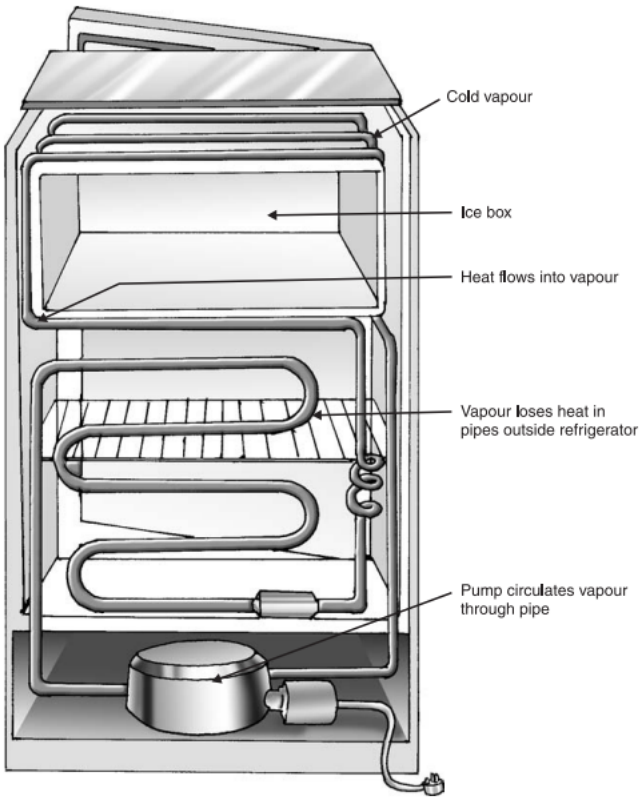
Tutorial – 1

Batch – B1 Weightage – 5 Marks

The refrigerator is an essential part of almost every household for preserving food and thereby reducing wastage. The primary function of a refrigerator or freezer is to provide food storage space maintained at a low temperature for the preservation of food. Its essential secondary function is the formation of ice cubes for domestic consumption.

A storage temperature of 0° to 4°C (32° to 39°F) is satisfactory for the preservation of most of the fresh foods. For the short-term storage of frozen foods, however, temperatures much below the freezing point are required. The evaporator in the domestic refrigerator, formed as a box, serves as a freezer for the storage of frozen food as well as for making ice cubes. It is mounted above the food storage space.

Pictorial representation of commonly used single door domestic refrigerator has been mentioned below:



As an embedded engineer, design an embedded system for refrigerator unit to meet the requirements of a domestic user. Provide the details of the designed embedded system in following format:

a. Detailed hardware block diagram for the embedded system. (2M)
b. Detailed software flow-chart for the operation of the embedded system. (2M)
c. Choice of an appropriate microcontroller with justification. (1M)

Name :- V. Surya Kumar

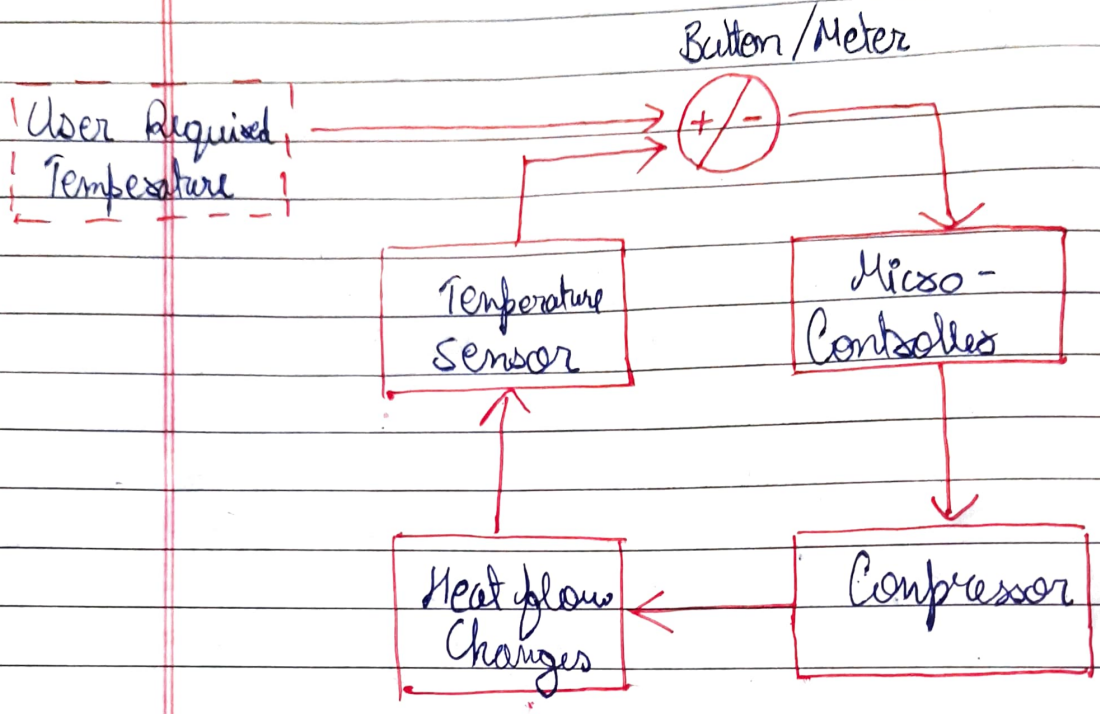
Reg no. :- 19BCE10286

PAGE NO.:

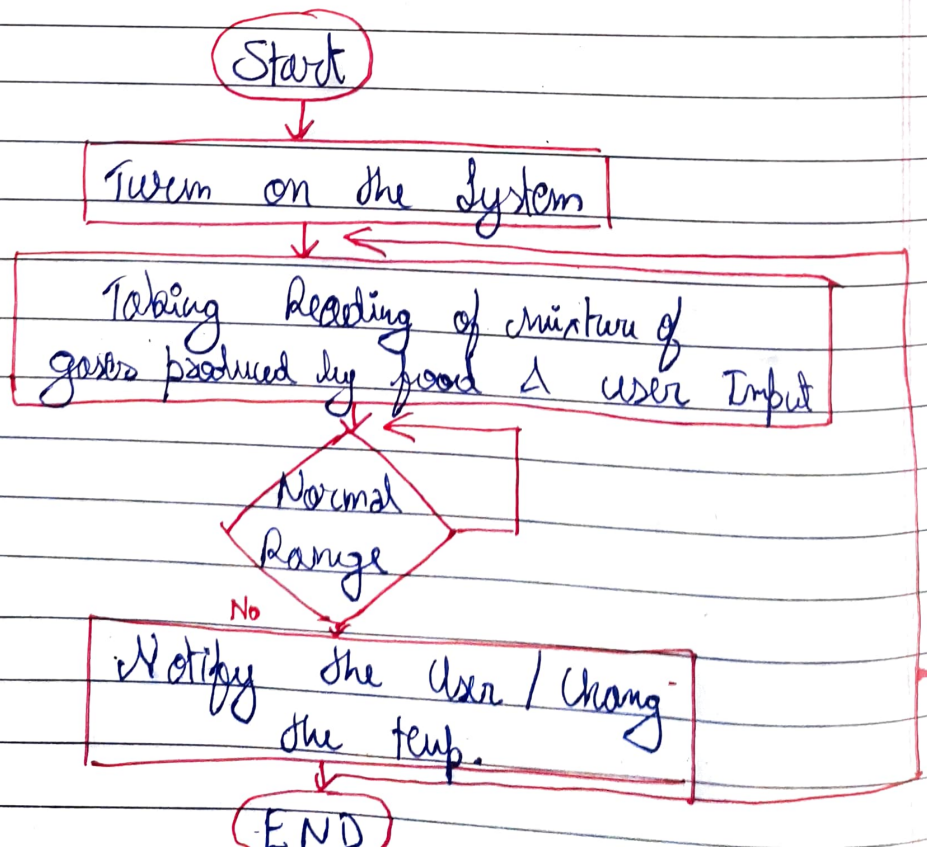
DATE: / /20

Tutorial - 1

ⓐ Hardware block diagram



ⓑ



(c) In Refrigerator, microcontroller will recieve input from button / meter (user temperature req.), temperature sensor, humidity sensors etc. through an external ADC.

Also, an external EEPROM to store data such as desired temperature setting and control functions including auto-restart, over heat protection, de-freeze support.

To handle all these tasks with efficiency & optimal power consumption as well as cost, we can use 8051 micro-controller & AT89C51 in refrigerator.

ECS 4010 Lab Task – 1

Batch – **B** Marks – **5**

Date: 11/02/2022 **Duration: 10:30-11:30 AM**

Delayed submissions and submissions through emails would not be evaluated.

Problem Statement:

As an embedded engineer, the use of the delay function is an essential requirement to implement the majority of real-world systems. The delay can be generated in two ways: 1. through Software loops 2. through Hardware peripherals

In lab task 1, you must implement a software delay for AT89C51 microcontroller unit.

The value of delay can be the **reverse order of last three digits** of your registration number in microseconds.

(Example: If your registration number is 19BCE10123, you must implement a delay of 321 microseconds.)

Assumptions:

1. Crystal frequency for AT89C51 is 11.0592 MHz
2. AT89C51 – 1 instruction cycle = 12 clock cycles

Simulate the delay with the help of the Keil IDE and Proteus tool.

Validate the delay on the Proteus simulation tool using Digital Oscilloscope and cursors.

Find out difference between actual delay and the simulated delay to comment upon the accuracy of software delays.

Submission:

1. A pdf file including screenshots of Keil code, Proteus Oscilloscope window and calculations for accuracy of software delay for your delay value.
2. A zip file containing Keil project

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Target 1

Project

- Project: lab-1
  - Target 1
    - Source Group 1
      - STARTUP.A51

labQues1.c

```c
// Name: V Surya Kumar
// Reg. No. 19BCE10286

#include <reg51.h>

sbit mybit = P2^4;

void MSDelay(unsigned int itime){
  unsigned int i, j;
  for(i=0; i<itime; i++){
    for(j=0; j<1000; j++);
  }
}

void main(){
  while(1){
    mybit = ~mybit;
    // Registration number = 268
    //      Reverse number = 862
    MSDelay(862);
  }
}
```

Build Output

```
    MODULE:  .\Objects\STARTUP.obj (?C_STARTUP)
*** WARNING L2: REFERENCE MADE TO UNRESOLVED EXTERNAL
    SYMBOL:  ?C_START
    MODULE:  .\Objects\STARTUP.obj (?C_STARTUP)
    ADDRESS: 080AH
Program Size: data=9.0 xdata=0 code=15
creating hex file from ".\Objects\lab-1"...
".\Objects\lab-1" - 0 Error(s), 2 Warning(s).
Build Time Elapsed:  00:00:01
```

Build target files | Simulation | L:1 C:1 | CAP NUM SCRL OVR R/W

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x082C
states	389
sec	0.0003890
psw	0x00

Project | Registers

Disassembly

```
15: void main(){
16:         while(1){
17:                 mybit = ~mybit;
18:                 // Registration number = 268
19:                 //                 Reverse number = 862
```

temp.c | STARTUP.A51

```
2  // Reg. No. 19BCE10286
3
4  #include <reg51.h>
5
6  sbit mybit = P2^4;
7
8  void MSDelay(unsigned int itime){
9    unsigned int i, j;
10   for(i=0; i<itime; i++){
11     for(j=0; j<1000; j++);
12   }
13 }
14
15 void main(){
16   while(1){
17     mybit = ~mybit;
18     // Registration number = 268
19     //      Reverse number = 862
20     MSDelay(862);
21   }
22 }
```

Parallel Port 2

Port 2

P2: 0xFF 7 Bits 0

Pins: 0xFF

Command

```
Running with Code Size Limit: 2K
Load "C:\\users\\crossover\\My Documents\\VIT Docs\\Sem 6\\Embedded\\Proj
```

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet

Call Stack + Locals

Name	Location/Value	Type
MAIN	C:0x082C	

Call Stack + Locals | Memory 1

Simulation | t1: 289.79155300 sec | L:17 C:1 | CAP NUM SCRL OVR R/W

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x082C
states	389
sec	0.0003890
psw	0x00

Project | Registers

Disassembly

```
15: void main(){
16:         while(1){
17:                 mybit = ~mybit;
18:                 // Registration number = 268
19:                 //                 Reverse number = 862
```

temp.c | STARTUP.A51

```
2  // Reg. No. 19BCE10286
3
4  #include <reg51.h>
5
6  sbit mybit = P2^4;
7
8  void MSDelay(unsigned int itime){
9    unsigned int i, j;
10   for(i=0; i<itime; i++){
11     for(j=0; j<1000; j++);
12   }
13 }
14
15 void main(){
16   while(1){
17     mybit = ~mybit;
18     // Registration number = 268
19     //      Reverse number = 862
20     MSDelay(862);
21   }
22 }
```

Parallel Port 2

Port 2

P2: 0xEF 7 Bits 0

Pins: 0xEF

Command

```
Running with Code Size Limit: 2K
Load "C:\\users\\crossover\\My Documents\\VIT Docs\\Sem 6\\Embedded\\Proj
```

>

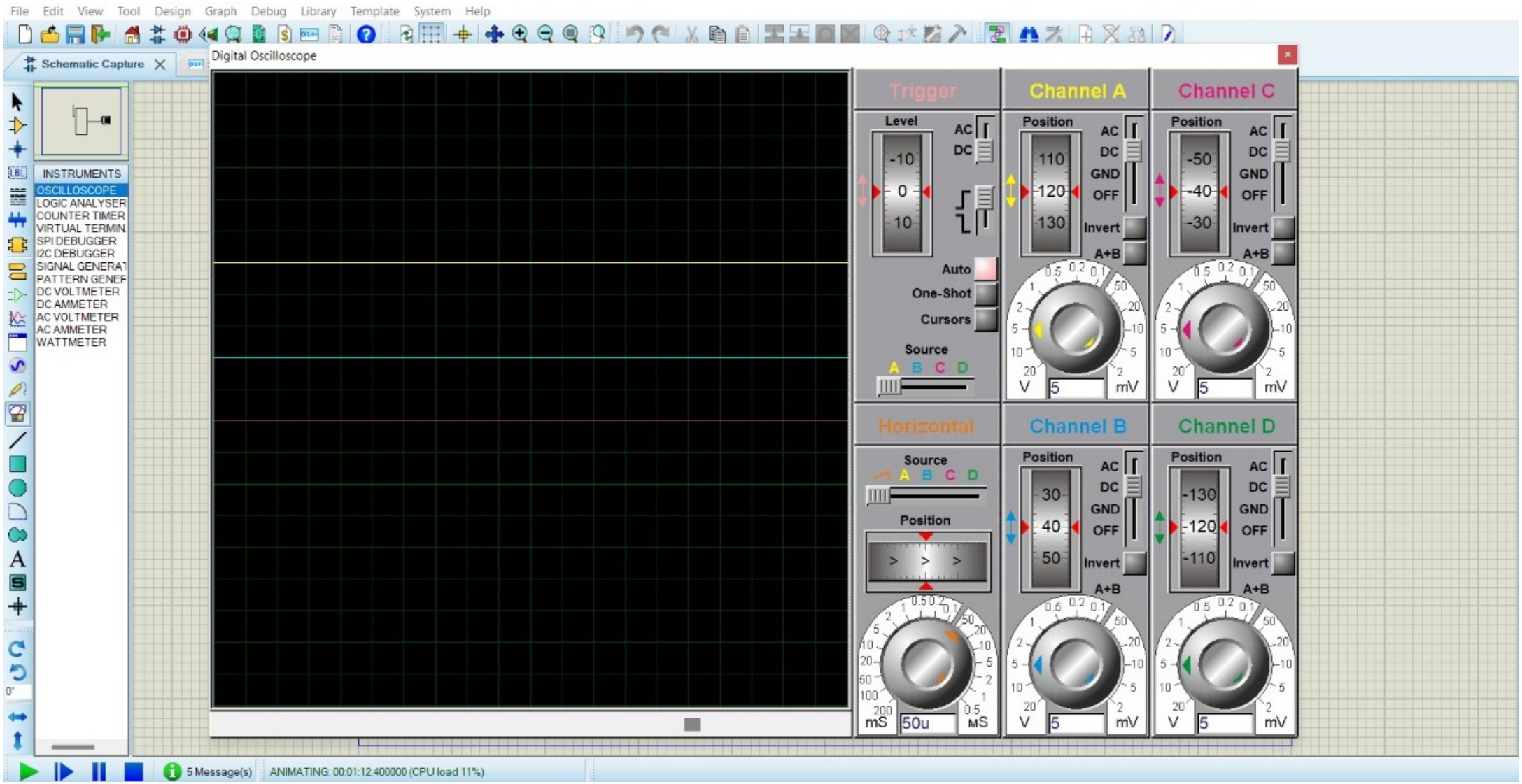
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet

Call Stack + Locals

Name	Location/Value	Type
MAIN	C:0x082C	

Call Stack + Locals | Memory 1

Simulation | t1: 415.99662900 sec | L:17 C:1 | CAP NUM SCRL OVR R/W


File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Digital Oscilloscope
INSTRUMENTS
OSCILLOSCOPE
LOGIC ANALYSER
COUNTER TIMER
VIRTUAL TERMIN
SPI DEBUGGER
I2C DEBUGGER
SIGNAL GENERAT
PATTERN GENER
DC VOLTMETER
DC AMMETER
AC VOLTMETER
AC AMMETER
WATTMETER
Trigger
Level
-10
0
10
AC
DC
Auto
One-Shot
Cursors
Source
A B C D
Channel A
Position
110
120
130
GND
OFF
Invert
A+B
V
5
mV
Channel C
Position
-50
-40
-30
Horizontal
Source
A B C D
Position
mS
50u
µS
Channel B
Position
30
40
50
Channel D
Position
-130
-120
-110
5 Message(s)
ANIMATING: 00:01:12.400000 (CPU load 11%)

ECS 4010 Lab Task – 2

Marks – **5**

Submission Post date: 18/02/2022 11:45 AM

Submission deadline: 21/02/2022 08:00 AM

Delayed submissions and submissions through emails will not be evaluated.

Problem Statement:

In Lab Task – 1, we got introduced to delays. Delays are essential building blocks of an embedded system that play a vital role in defining system operation. Input-output peripherals also serve as necessary building blocks of any embedded system. In this lab task, we would get introduced to interfacing input-output peripherals to the microcontroller.

In Lab Task – 2, interface an LCD to the microcontroller unit. The LCD is a 16x2 display.

The first row of the LCD should print your name, and the second row of the LCD should print your registration number.

Assumptions:

1. Crystal frequency for AT89C51 is 11.0592 MHz
2. AT89C51 – 1 instruction cycle = 12 clock cycles
3. LCD – 14 Pins

Simulate the task with the help of the Keil IDE (Embedded C) and Proteus tool.

Validate the details on the Proteus simulation tool.

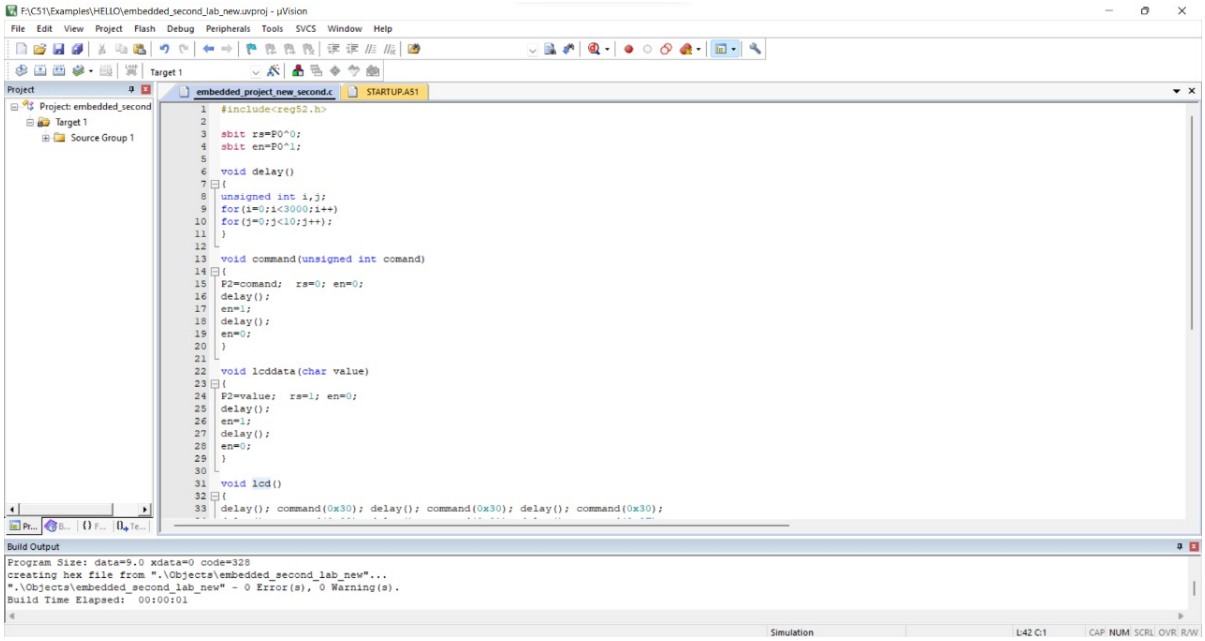
Submission:

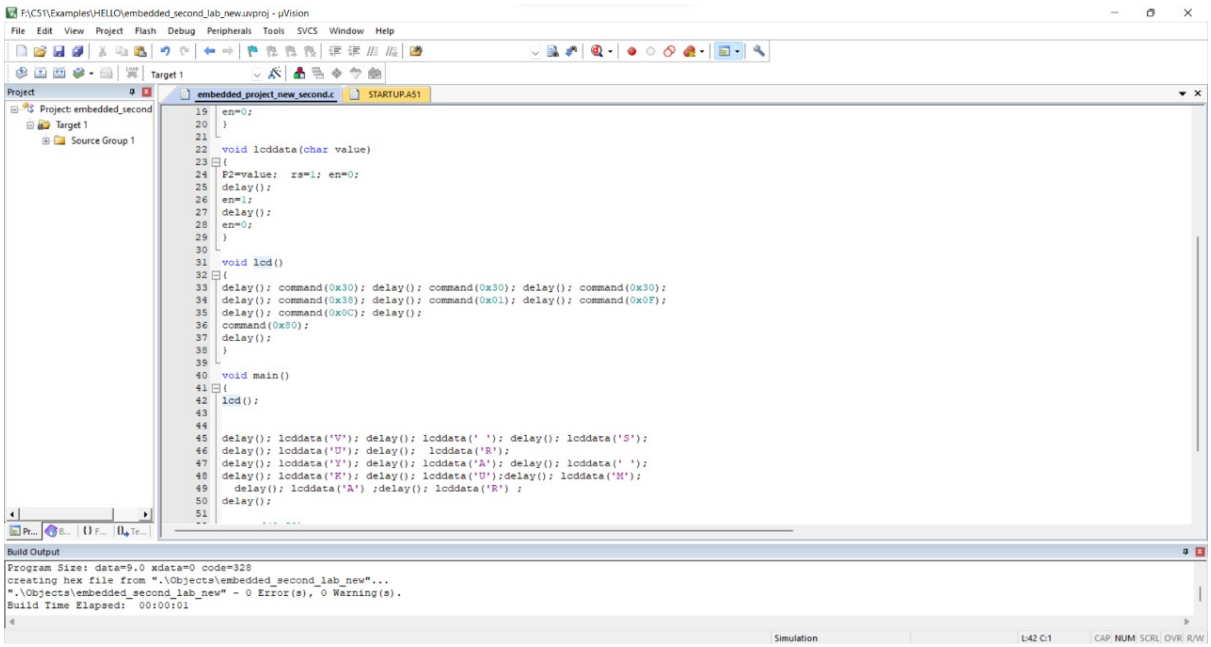
1. A pdf file including screenshots of Keil code, Proteus window which shows LCD screen including your name and registration number
2. A zip file containing Keil project

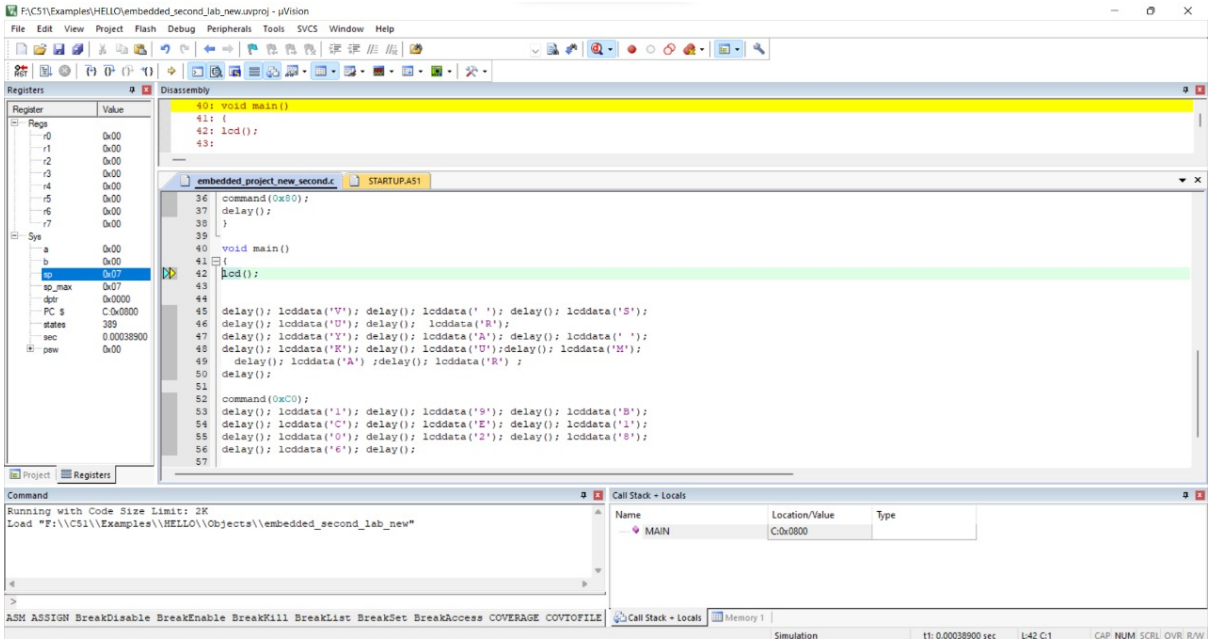
Name: V Surya Kumar
Reg. No. 19BCE10286

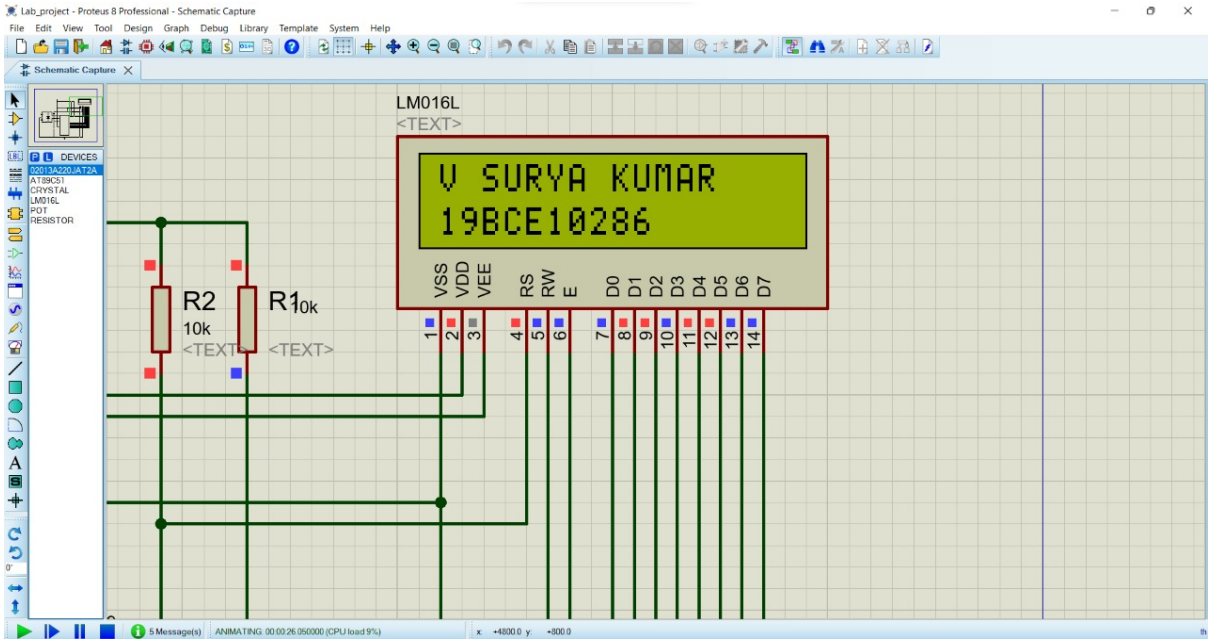
LAB TASK – 2

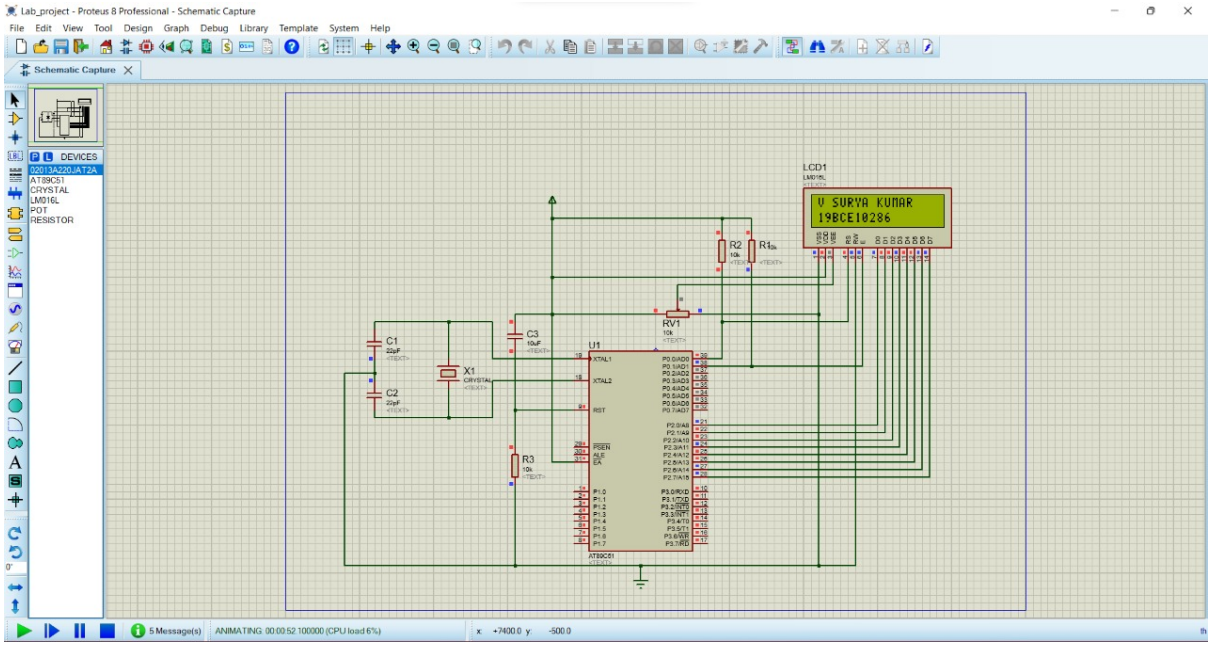
CODE:






F:\C51\Examples\HELLO\embedded_second_lab_new.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
40: void main()
41: {
42: lcd();
43:
embedded_project_new_second.c
STARTUP.A51
36 command(0x80);
37 delay();
38 }
39
40 void main()
41 {
42 lcd();
43
44
45 delay(); lcddata('V'); delay(); lcddata(' '); delay(); lcddata('S');
46 delay(); lcddata('U'); delay(); lcddata('R');
47 delay(); lcddata('Y'); delay(); lcddata('A'); delay(); lcddata(' ');
48 delay(); lcddata('K'); delay(); lcddata('U');delay(); lcddata('M');
49 delay(); lcddata('A') ;delay(); lcddata('R') ;
50 delay();
51
52 command(0xC0);
53 delay(); lcddata('1'); delay(); lcddata('9'); delay(); lcddata('B');
54 delay(); lcddata('C'); delay(); lcddata('E'); delay(); lcddata('1');
55 delay(); lcddata('0'); delay(); lcddata('2'); delay(); lcddata('8');
56 delay(); lcddata('6'); delay();
57
Command
Running with Code Size Limit: 2K
Load "F:\\C51\\Examples\\HELLO\\Objects\\embedded_second_lab_new"
Call Stack + Locals
Name
Location/Value
Type
MAIN
C:0x0800
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE


Lab_project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
02013A220JAT2A
AT89C51
CRYSTAL
LM016L
POT
RESISTOR
LM016L
<TEXT>
V SURYA KUMAR
19BCE10286
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
R2
10k
R1
10k
5 Message(s)
ANIMATING: 00:00:26.050000 (CPU load 9%)


Lab_project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
02013A220JAT2A
AT89C51
CRYSTAL
LM016L
POT
RESISTOR
LCD1
V SURYA KUMAR
19BCE10286
R2
R1
RV1
C1
C2
C3
X1
U1
R3
5 Message(s)
ANIMATING: 00:00:52.100000 (CPU load 6%)
x: +7400.0 y: -500.0

ECE 4010 Embedded Systems

Tutorial – 2,3

(Module 2, Module 3)

Batch – B1 Weightage – 10 Marks

Q1. A 4-bit DAC has been interfaced to AT89C51 for converting the digital signal to its analogue equivalent. (Module 2)

Considering the binary weighted DAC with value of digital signal = 1111_b.

Find the analogue equivalent value of the signal. V_{dc} = 5V and R_f = R ohms. [5]

Q2. An asynchronous serial communication needs to be established between two embedded systems which have low data-rate requirement. The protocol also should have basic error detection mechanism available. (Module 3)

a. Choose an appropriate protocol to meet requirement of the networked embedded system and justify your choice. [2]

b. Also, consider an 8-bit data needs to be serially transferred between embedded systems. The value of the 8-bit data being 45_h.

   What would be the value of the parity bit if the embedded systems are operating on even parity logic. Provide justification. [3]

Name: V. SURYA KUMAR

Reg No: 19BCE10286

PAGE NO.:

DATE: / /20

Digital Signal = 1111_b , $V_{dc} = 5V$

Q1

$$V_o = -R_f \sum_{i=1}^{n} \frac{B(n-i) \cdot V}{2^{i-1} R}$$

$(n=4)$

$B(4-1) = B3 = 1$

$B(4-2) = B2 = 1$

$B(4-3) = B1 = 1$

$B(4-4) = B0 = 1$

$$V_o = -R \left[\frac{B3 \cdot V}{2^{1-1} R} + \frac{B2 \cdot V}{2^{2-1} R} + \frac{B1 \cdot V}{2^{3-1} R} + \frac{B0 \cdot V}{2^{4-1} R} \right]$$

$$V_o = -\frac{\cancel{R}}{\cancel{R}} \left[\frac{1 \cdot V}{2^0} + \frac{1 \cdot V}{2^1} + \frac{1 \cdot V}{2^2} \cdot \frac{1 \cdot V}{2^3} \right]$$

$$V_o = -\left[\frac{5}{1} + \frac{5}{2} + \frac{5}{4} + \frac{5}{8} \right]$$

$$= \underline{\underline{-9.375}}$$

Q2. a) The protocol we will be using is RS232 interface.

Because it meets the two specified conditions in the problem, i.e,

- Have basic error detection mechanism of checking parity bit.

And also because the complexity is low

b) The parity bit would be '1'.

As in $(45)_H = (01000101)_2$

Because in even parity bit scheme, Parity bit is 1, if there are odd no. of 1's in data stream.

ECE 4010 Embedded Systems

Tutorial – 1

Batch – E1 Weightage – 5 Marks

A microwave oven (commonly referred to as a microwave) is an electric oven that heats and cooks food by exposing it to electromagnetic radiation in the microwave frequency range. This induces polar molecules in the food to rotate and produce thermal energy in a process known as dielectric heating. Microwave ovens heat foods quickly and efficiently for a homogeneous, high water content food item.

The major components of any microwave oven include but not limited to:

- **High Voltage Transformer:**
  Microwaves require a high voltage of electricity to effectively function and heat food, usually at a minimum level of between 3000 or 4000 volts. Standard power outlets in India provide just 230 volts of electricity and therefore require a transformer to alter the power level and provide an adequate voltage to the appliance. This is where the high voltage transformer comes in, as it can convert the voltage to a much higher level. To do this, it uses a range of capacitors that loop electrical currents to multiply their power, and thereby supplying the microwave with the power it needs to run.

- **Magnetron:**
  This part of a microwave receives the high voltage of electricity supplied by the transformer and converts it into microwave energy to heat the food. The magnetron does this with the use of a specially created diode, which controls electrons using magnetic fields. There are two ring-shaped magnets that deflect the electrons, causing them to move in a circular direction and prevent them from shooting off

- **Cooling Fan or Stirrer:**
  Just like other heating appliances, a microwave oven is at risk of overheating and therefore contains a cooling fan to prevent this from happening. The cooling fan in a microwave works to dissipate the heat produced by the oven, ensuring it remains at a safe level. Older microwaves did not have cooling fans but instead featured a stirrer fan which was designed to do the same thing.

- **Cooking Cavity and Case:**
  The cooking cavity is the central hollow part of a microwave where food is placed to be cooked, warmed up, or defrosted. The cooking cavity tends to be quite minor compared to the size of the entire microwave, which might seem unusual compared to other cooking appliances, where the size of cooking space is similar to the overall size of the appliance. Microwaves produced by microwave ovens are harmful to humans, and so it is important to keep these contained and away from people. This is why the cavity on a microwave is

so deep, with thick walls to provide optimum safety for the user. When the user shuts the door of a microwave, it forms a sealed container through which no microwaves are able to escape.

- **Turntable:**
  The turntable is a glass plate inside the microwave cavity, which is usually removable to make for easy cleaning. The purpose of the turntable is to rotate the food around so that it cooks more evenly, avoiding hot or cold spots. The turntable slots onto the bottom of the microwave cavity using ridges, allowing it to easily lock into place.

- **Door:**
  The door of a microwave is there to allow the user easy access to the inner cavity for placing or removing plates of food. However, it has a much greater and more important function, which is to seal the microwaves inside the cooking cavity to prevent injury to the user. Microwave doors have seals along the edge to ensure a secure compartment is created when the door is closed. This prevents any microwaves from leaking out of the edges and causing burns.

- **Control Panel:**
  The control panel is the part of the microwave oven, which allows the user to dictate what action they want the appliance to perform. Depending on the type of microwave, the control panel might be operated with a knob or with buttons. Using this, you can set the length of time and intensity of the microwave's cooking and select different operations such as defrost mode or grill mode. The control panel on a modern microwave will typically feature a digital display that counts down the amount of cooking time the microwave has left.

As an embedded engineer, design an embedded system for microwave oven to meet the requirements mentioned above. Provide the details of the designed embedded system in following format:

a. Detailed hardware block diagram for the embedded system. (2M)
b. Detailed software flow-chart for the operation of the embedded system. (2M)
c. Choice of an appropriate microcontroller with justification. (1M)

ECE 4010 Embedded Systems: Tutorial – 1

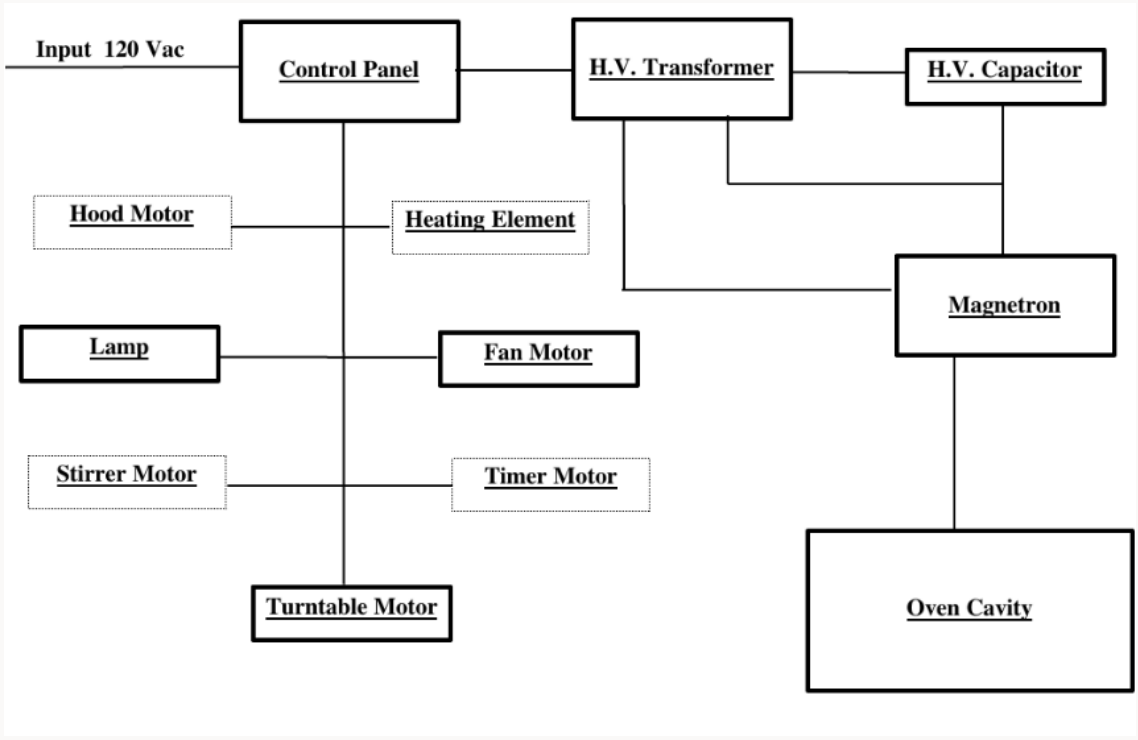
Name: Abhishek Srivastava
Registration Number: 19BCE10071

An embedded system for the microwave oven to meet the requirements:
Provide the details of the designed embedded system in the following format:

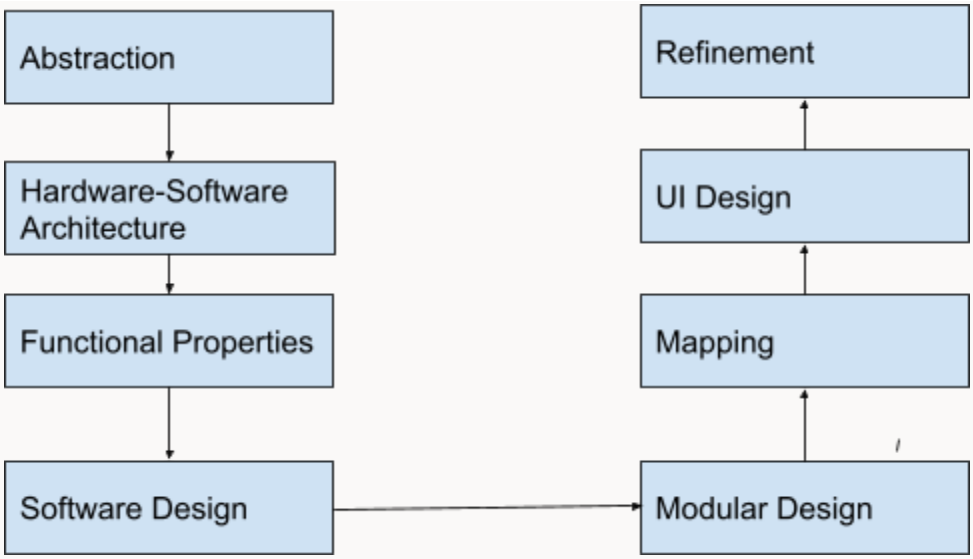
Detailed hardware block diagram for the embedded system.

Components required:

- DoorSensor
- WeightSensor
- Keypad
- Timer
- Heating Element
- Lamp
- Turntable
- Oven Display
- Sleeper
- Software System



Detailed software flow -chart for the operation of the embedded system.



Choice of an appropriate microcontroller with justification.

Depending on the environment, we tend to achieve temperature tolerance. We want devices that withstand extreme temperatures and consume less power at the same time.

Some temperature-tolerant microcontrollers include STMicroelectronics' STM32F103 series, NXP's Kinetis EA series, Renesas' RX24T and RX24U, Infineon's XMC series and AURIX™ series, Microchip's: PIC and AVR microcontrollers, and Texas Instruments' MSP430F2619S-HT

Among the above-mentioned microcontrollers, I would choose **STMicroelectronics' STM32F103 series**

Reasons for choosing **STMicroelectronics' STM32F103 series:**

- ARM 32-bit Cortex - M3 CPU Core
- Clock, reset and supply management
- Up to 80 fast I/O ports
- Up to 9 communication interfaces
- 7 timers
- VBAT supply for RTC and backup registers
- VBAT supply for RTC and backup registers
- It consumes less power.

ECE 4010 Embedded Systems

Tutorial – 2,3

(Module 2, Module 3)

Batch – E1 Weightage – 10 Marks

Q1. An 8-bit ADC needs to be interfaced been interfaced to AT89C51 for converting the analogue signal to its digital equivalent. (Module 2)

Provide a detailed connection diagram between ADC and microcontroller and elaborate functioning of the individual pin in the connection diagram. [5]

Q2. An embedded system comprises a microcontroller unit and a hardware peripheral unit. The microcontroller unit has been interfaced with the hardware peripheral unit using Serial Peripheral Interface Protocol. In this system, the microcontroller reads the data from the hardware peripheral unit device periodically. Generate timing diagram for the communication with the help of CS, SCLK, MISO and MOSI signals if the hardware peripheral unit wants to send the data 42h to the microcontroller. The transfer should occur at the falling edge of the clock, and the internal register value of the hardware peripheral unit is 24h. (Module 3) [5]

Embedded System

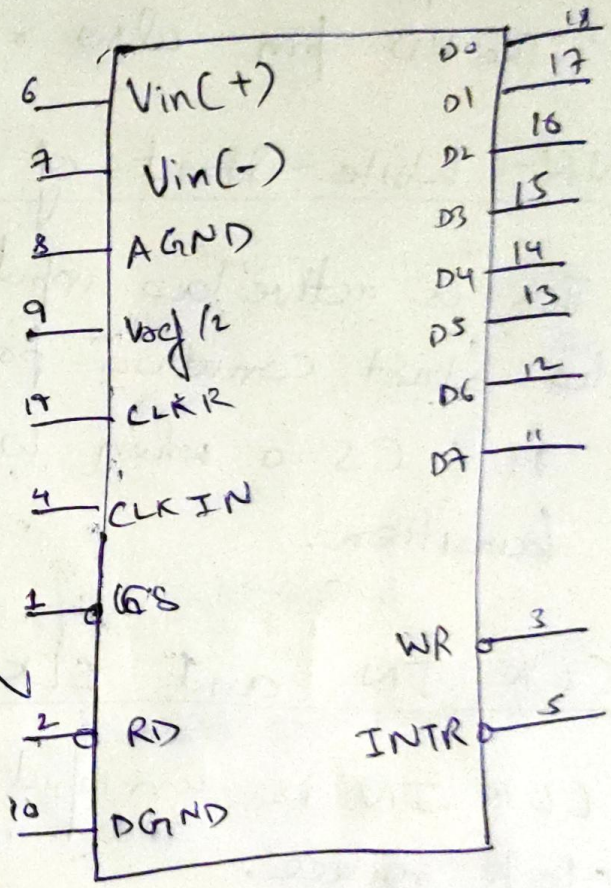
Tutorial 2,3

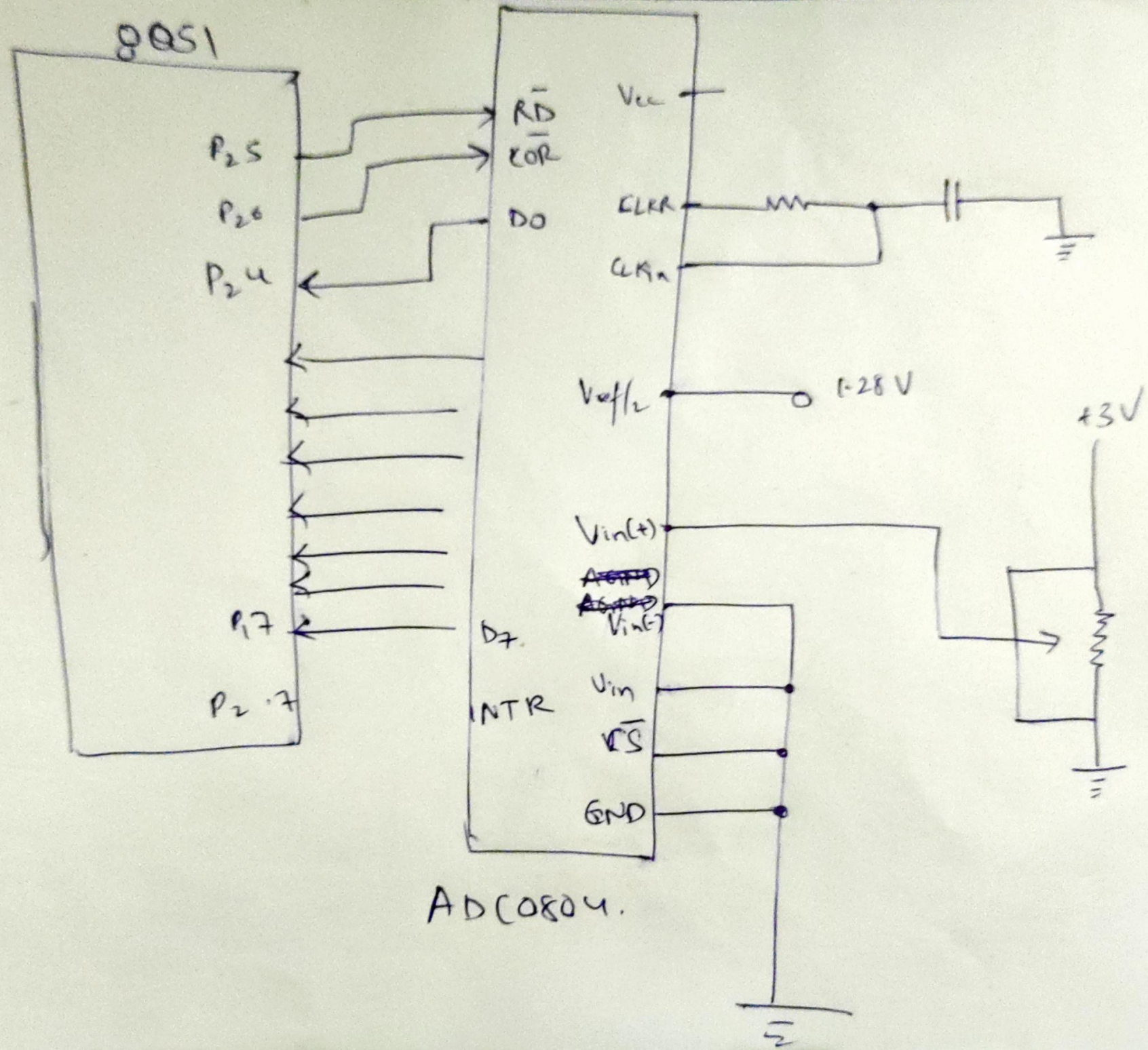
19BCE10071
Abhishek Srivastava.

Question-1

ADC -0804

- The ADC-0804 IC is an 8-bit parallel ADC in the family of the ADC0800 series from National Semiconductor.
- It works with +5V and has resolution of 8 bits.
- The conversion time varies depending on the clocking signals applied to th CLK IN pin.
- But it cannot be faster than 110 µs.




8051
P2.5
P2.6
P2.4
P1.7
P2.7
RD
WR
DO
D7
INTR
Vcc
CLKR
CLKin
Vref/2
1.28 V
+3V
Vin(+)
Vin(-)
Vin
CS
GND
ADC0804.

CS - Chip Select

It is an active low input used to activate the ADC0804 chip.

RD - Read

- It input signal in active low
- When CS=0, if a high-to-low pulse is applied to the RD pin.
- The RD pin also reffered as Output Enable (OE.

WR - Write - Start of Conversation.

- This is active low input, used to inform ADC0804 to start conversion process.
- If CS=0 when WR make a ~~high~~ low-to-high transition.

CLK IN and CLK R

- CLK IN is an input pin connected to an external clock source.
- To use internal clock generator, the CLK IN and CLK R are connected to the capacitor and a resistor.
- $f = \frac{1}{1.1\,RC}$.

INTR - Interrupt - End on Conversation

- This is an o/p pin and is in active Low.
- After INTR goes low, we make CS = 0

Vin(+) and Vin(-)

- These are differential analog inputs.
- Vin = Vin(+) + Vin(-)

VCC

- This is +5V power supply.

Vref/2

- It is an input voltage used for reference voltage.
- If this pin is open, the analog input voltage for the ADC0804 is in the range of 0 to 5V.

D0 - D7

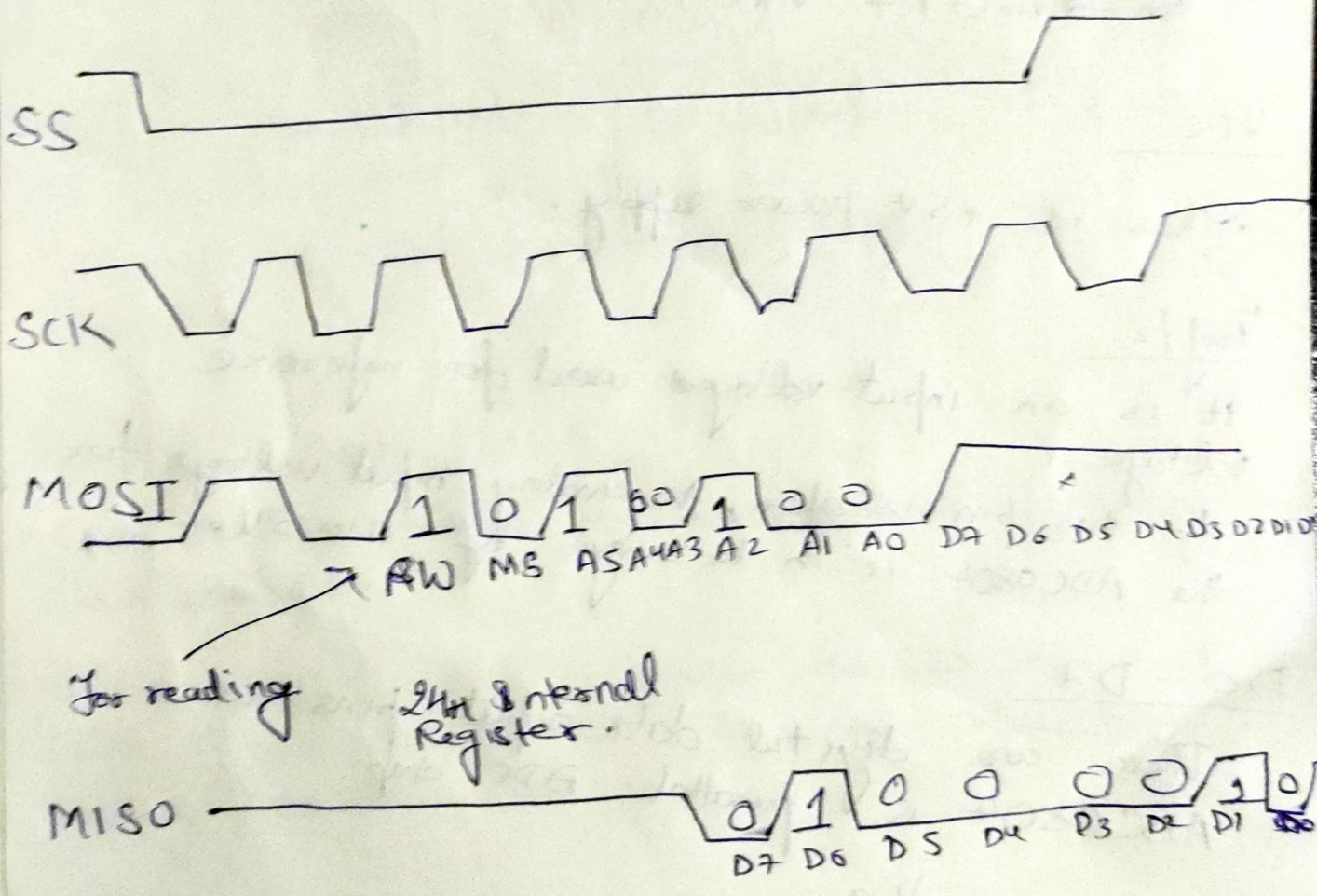
- These are digital data output pins since ADC0804 is parallel ADC chip.

→ $D_{out} = \frac{V_{in}}{\text{Step Size}}$

A Ground, D Ground

- There are 2 pins providing the ground for both the analog signals and the digital signals
- ~~the~~ We have two grounded pins in order to isolate the analog Vin signal.

Question 2



ECS 4010 Lab Task – 1

Batch – **E** Marks – **5**

Date: 10/02/2022 **Duration: 10:30-11:30 AM**

Delayed submissions and submissions through emails would not be evaluated.

Problem Statement:

As an embedded engineer, the use of the delay function is an essential requirement to implement the majority of real-world systems. The delay can be generated in two ways: 1. through Software loops 2. through Hardware peripherals

In lab task 1, you must implement a delay using hardware timer unit 0 available with the AT89C51 microcontroller unit.

The value of delay can be the **last three digits** of your registration number in microseconds.

(Example: If your registration number is 19BCE10234, you must implement a delay of 234 microseconds.)

Assumptions:

1. Crystal frequency for AT89C51 is 12 MHz
2. AT89C51 – 1 instruction cycle = 12 clock cycles

Simulate the delay with the help of the Keil IDE and Proteus tool.

Validate the delay on the Proteus simulation tool using Digital Oscilloscope and cursors.

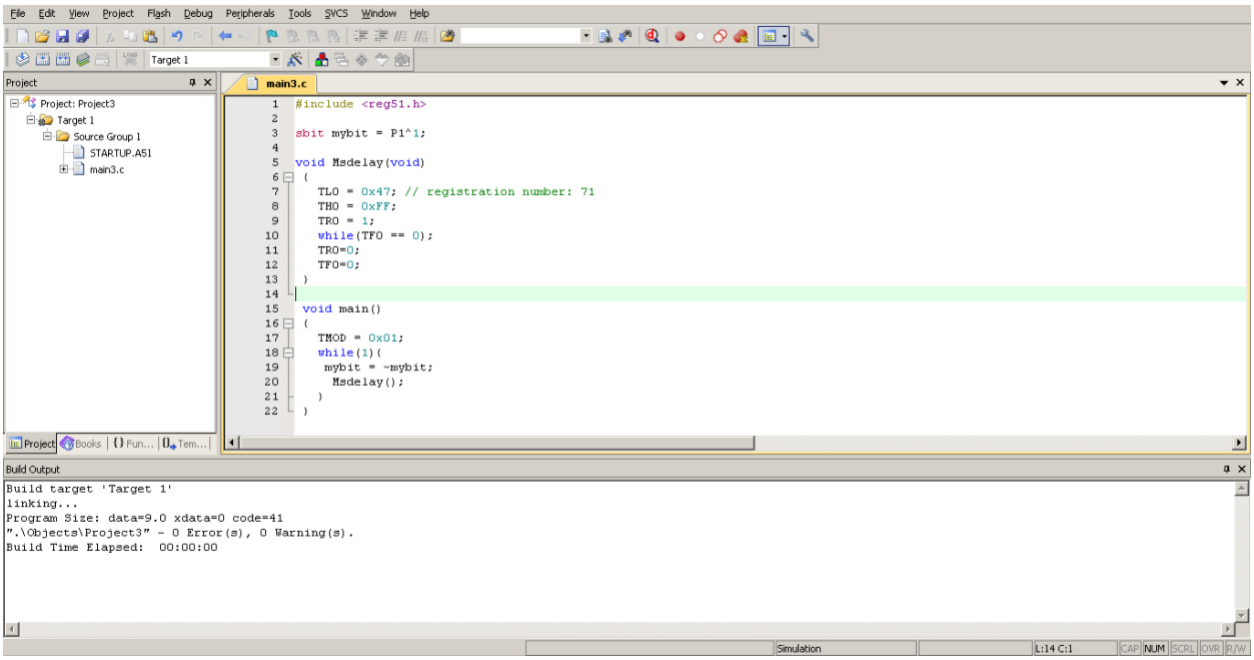
Submission:

1. A pdf file including screenshots of Keil code and Proteus Oscilloscope window
2. A zip file containing Keil project

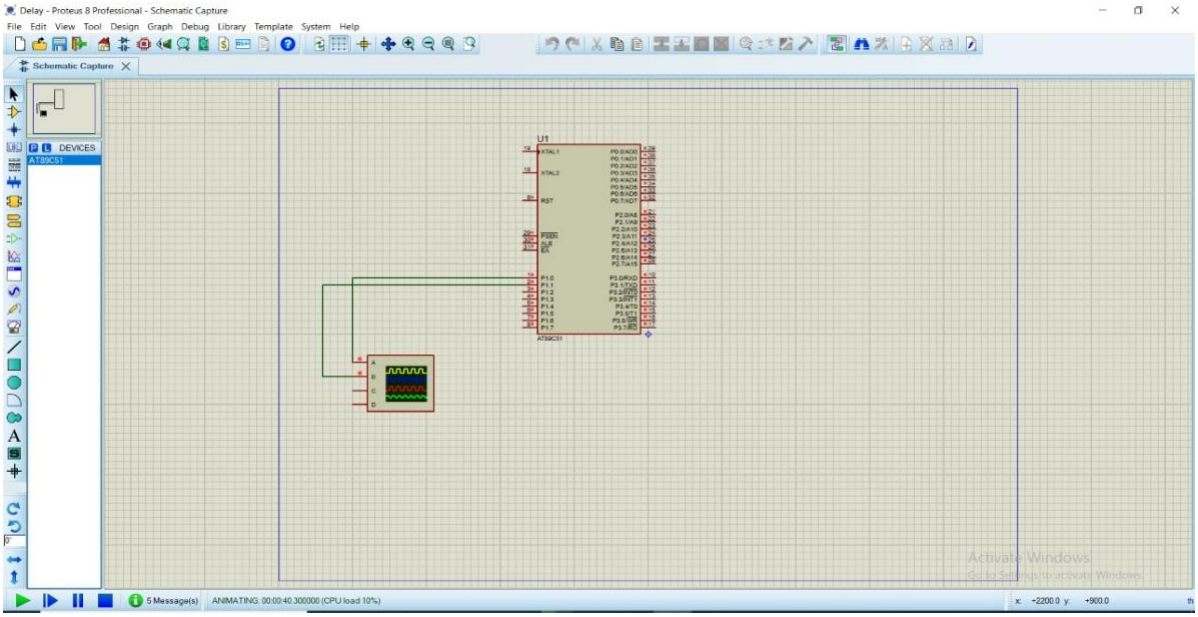
Abhishek Srivastava
19BCE10071

ECS 4010 Lab Task – 1

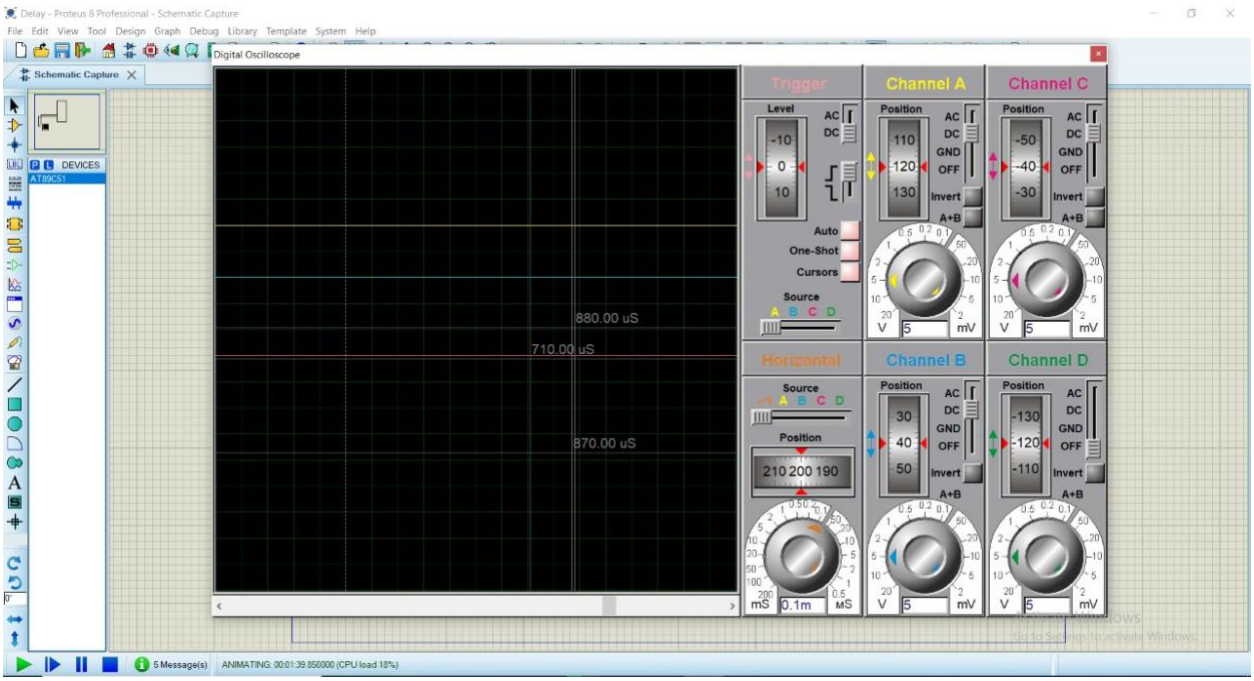
KEIL SCREENSHOT



PROTEUS SCREENSHOT



Abhishek Srivastava
19BCE10071



ECS 4010 Lab Task – 2

Marks – **5**

Submission Post date: 18/02/2022 11:45 AM

Submission deadline: 21/02/2022 08:00 AM

Delayed submissions and submissions through emails will not be evaluated.

Problem Statement:

In Lab Task – 1, we got introduced to delays. Delays are essential building blocks of an embedded system that play a vital role in defining system operation. Input-output peripherals also serve as necessary building blocks of any embedded system. In this lab task, we would get introduced to interfacing input-output peripherals to the microcontroller.

In Lab Task – 2, interface an LCD to the microcontroller unit. The LCD is a 16x2 display.

The first row of the LCD should print your name, and the second row of the LCD should print your registration number.

Assumptions:

1. Crystal frequency for AT89C51 is 11.0592 MHz
2. AT89C51 – 1 instruction cycle = 12 clock cycles
3. LCD – 14 Pins

Simulate the task with the help of the Keil IDE (Embedded C) and Proteus tool.

Validate the details on the Proteus simulation tool.

Submission:

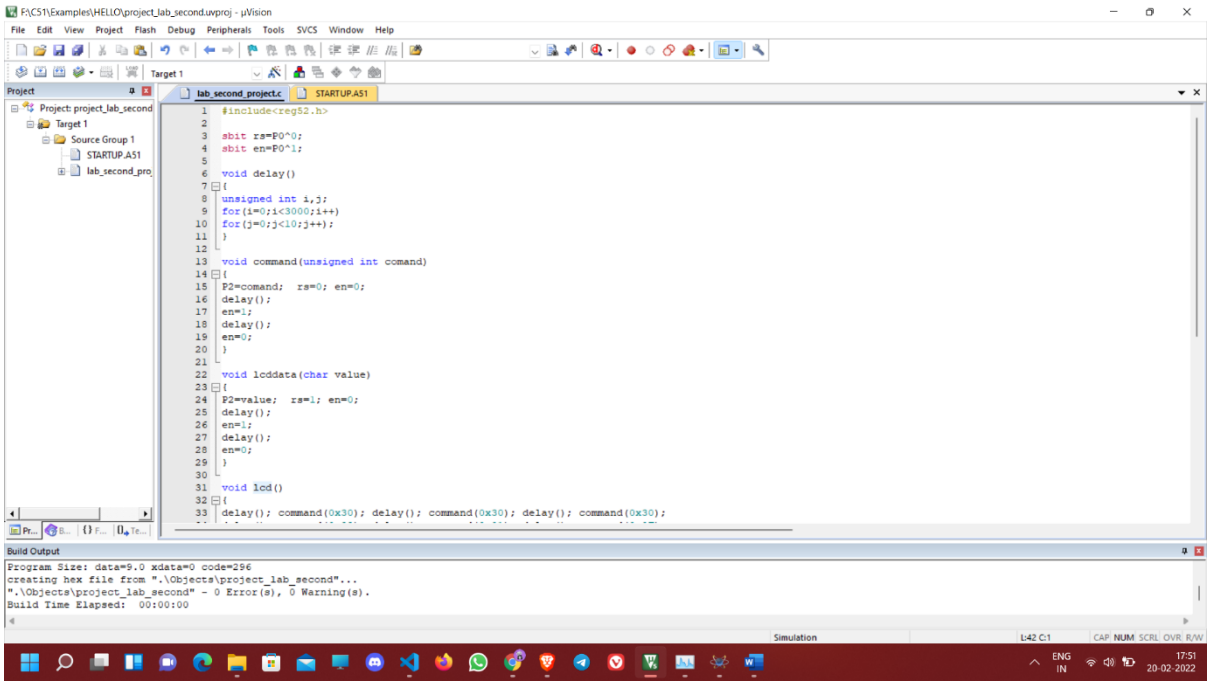
1. A pdf file including screenshots of Keil code, Proteus window which shows LCD screen including your name and registration number
2. A zip file containing Keil project

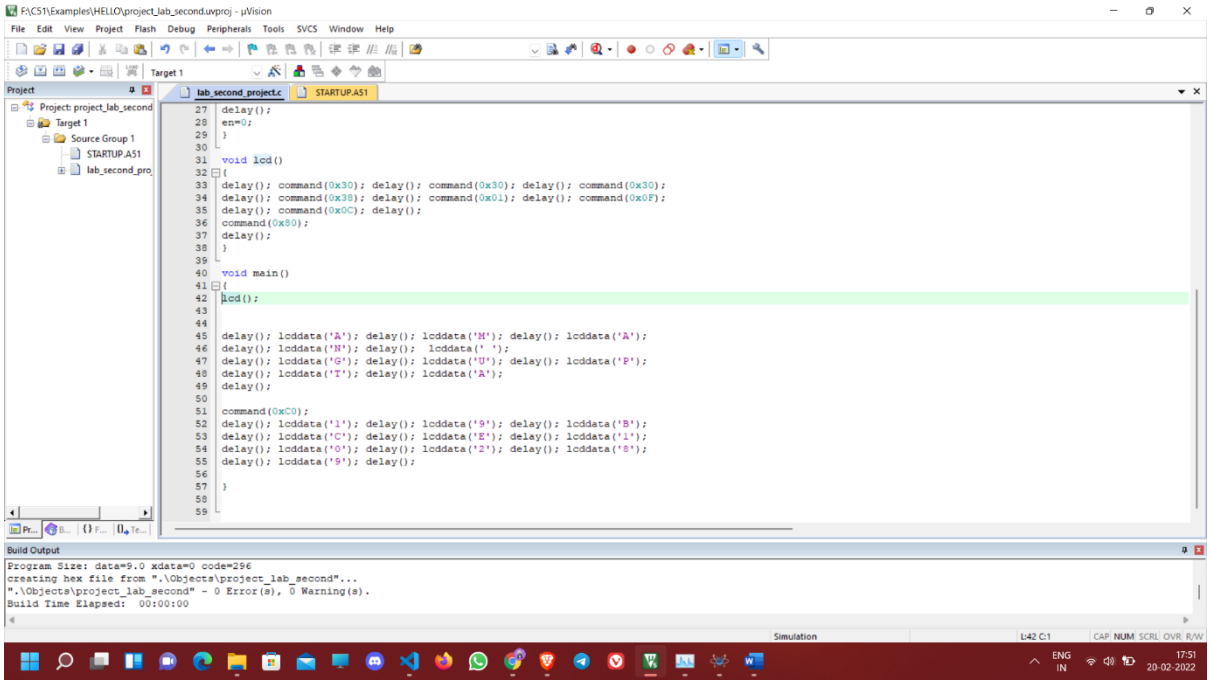
Name – Aman Gupta

Reg. No. – 19BCE10289

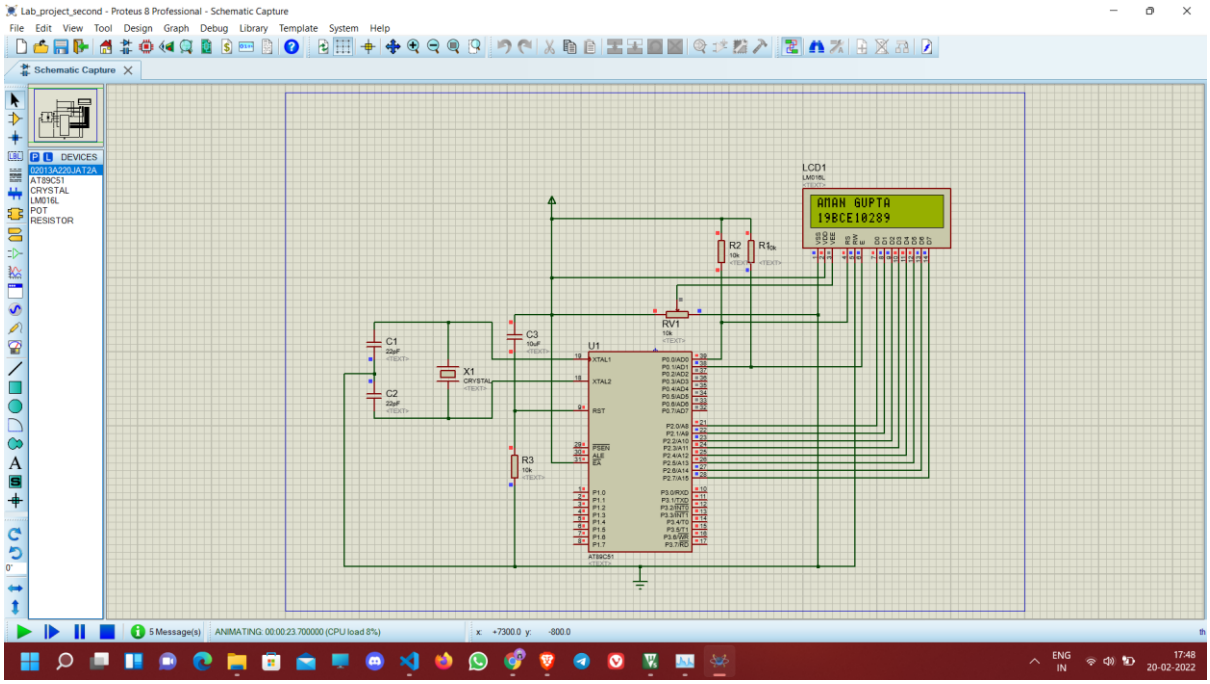
Lab Assignment – 2

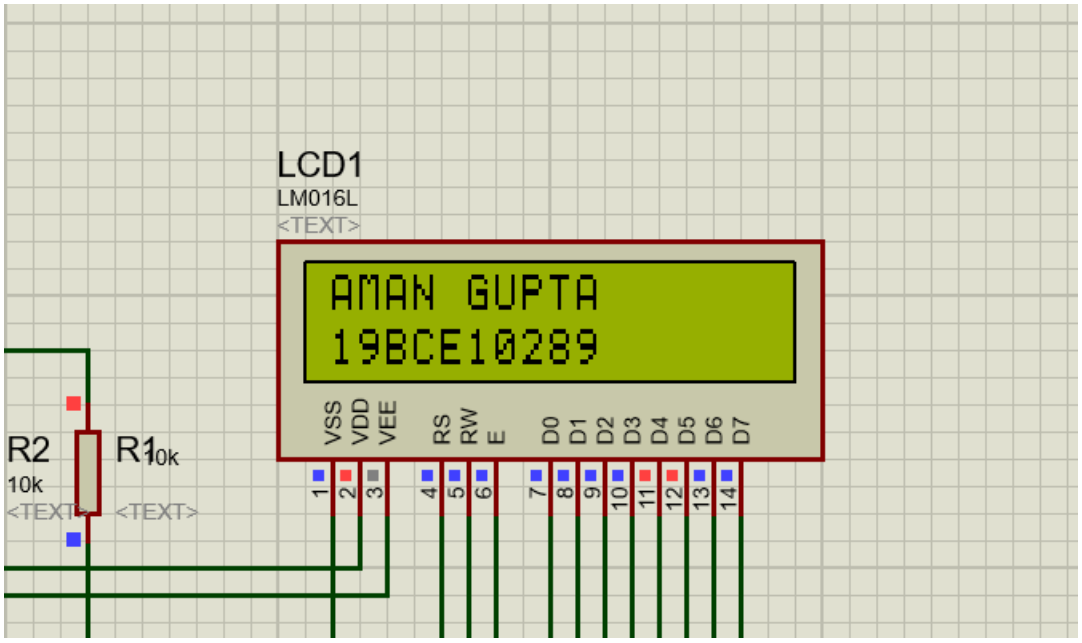
Keil Screenshots: -





Proteus Screenshots: -




LCD1
LM016L
<TEXT>
AMAN GUPTA
19BCE10289
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
R2
10k
<TEXT>
R1
10k
<TEXT>