DRISHTI

A Revolutionary Concept

MAKERSPACE 5.0

FIRE ALARM SYSTEM

Team Members :

◆ Aniket Rana
◆ Ashish Kumar Karn
◆ Tirth Jain
◆ Ratnadeep Patra
◆ Ankit Raj
◆ Harshrajsinh gohil

Mentors :

◆ Naveen Rawat
◆ Nisha Prakash
◆ Deepam Sinha
◆ Dravi Makwana
◆ Harsh Kakadiya

Components & Softwares :

◆ **Atmega32**
An AVR 8-bit microcontroller,can attain 1 MHz of frequency.
◆ **Tmp36**
Temperature sensor
◆ **Atmel Studio (to write the code)**
AVR Code compiler and IDE
◆ **Proteus**
Software for simulation of project
◆ **MQ2 gas-sensor**
Gas sensor
◆ **SIM900D**
GSM module

TABLE OF CONTENTS

Problem Statement

To design a fire alarm device that can deliver a warning SMS directly to your cell phones.

INTRODUCTION

Fire alarm system detects increase in temperature, and presence of smoke to detect the presence of fire and warn by buzzers and LEDs (also by sending messages - by using GSM module). These sensors are fixed in different zones and all are wired to the Main Control Unit (MCU), such that we can pinpoint the zone which has caught fire.

PROTEUS
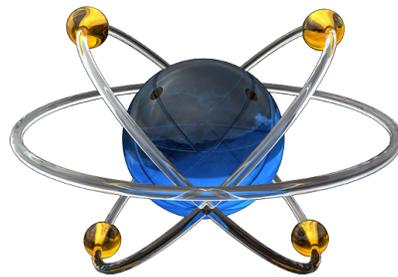
➔ Research on proteus software and to watch some tutorials of it.

https://en.wikipedia.org/wiki/Proteus_Design_Suite

Download/Install Proteus :
https://getintopc.com/softwares/3d-cad/proteus-professional-2020-free-download/

Research on ATMega32 :

Data Sheet :

https://ww1.microchip.com/downloads/en/DeviceDoc/doc2503.pdf
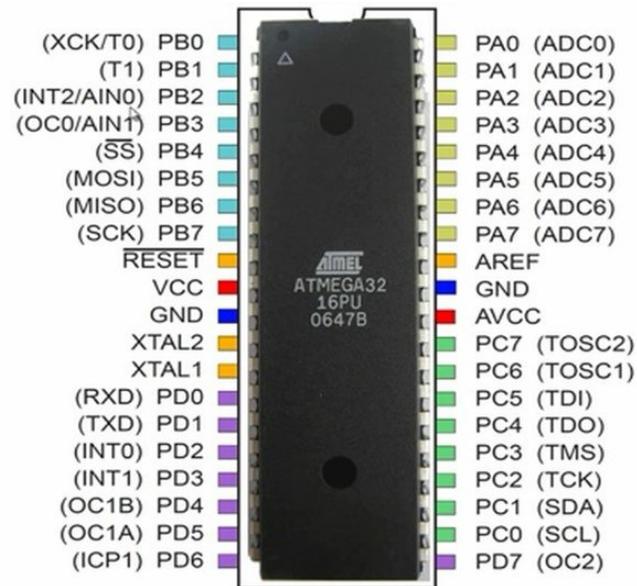
➜ How do components of circuits communicate (GSM module, Smoke detector, temperature sensor, etc)

ATMEL STUDIO :

➜ Research on proteus software and to watch some tutorials of it.

Download/Install Atmel Studio :

https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices?gclid=CjwKCAjwgZuDBhBTEiwAXNofRKQYXhRdMOMb7N04lRadZYnwg6MscA6rhtlPPotOwyyX4R-71mm36BoCE7sQAvD_BwE

➔ Learn till loops in C and Embedded C.

We learned about the basics of coding. We learnt if, while and all other functions, all those basic functions to write logic required in our code to carry out our project.

➔ Components needed for this build.

We will be using two types of sensor, temperature and smoke sensor to detect the presence of smoke or fire. We also require a microcontroller that will receive all the data from the sensors and perform several logical operations, which in turn sends output. We will require a component for sending msgs to the phone (GSM Module).

Date : 30/03/21

➔ Difference between BMP180 and IR Sensor (LM35). Which one is better ?

In BMP180, we can measure both barometric pressure and temperature. LM35 detects temperature only. We preferred BMP180, because it is easy to code and simulate. Moreover LM35 is very short distance range.

(But later we decided to go with TMP36, because debugging the data received from BMP180 was very much time consuming.)

➔ LM7805 Voltage regulator

We are not using it, as we are simulating in proteus, we can use a 5V supply. While making the circuit physically we would be requiring this voltage regulator as we would be using a battery (like a 9V one), so voltage is required to reduce to 5V.

➔ We would also be using diodes at the starting of the circuit to prevent damage to the components if by mistake we connect the battery with reverse polarity.

➔ Components :

(1) MQ2 (Gas Sensor)

It has 4 pins - VCC, Ground, Analog output, Digital output. VCC is connected to 5V, and Ground to ground. Thus, we can either directly use the digital output or we can take the analog output and using an analog to digital converter, we first convert it to digital output and then use it as input of the processing unit. As the amount of smoke increases, the voltage increases. In the simulation, we are using logic toggler, such that if the

logic toggler is 0, MQ2 gives 0V as output and if logic toggler is 1, MQ2 gives 5V.

(2) SIM900D

SIM 900D receives and transmits data by UART protocol. Using the AT commands, we can send messages to the module to perform specific functions (like sending sms, calling someone, etc.). Various AT commands that we are going to use in our project are :

AT - Replies OK for acknowledgement
AT + CMGF = 1 - Send SMS in text mode only
AT + CMGS = "+91 XXXXXXXXXX" > "Msg" <Ctrl + Z> - Send "Msg" to the given phone no

(3) TMP36 (Temperature sensor)

TMP36 has 3 pins: voltage pin, adc pin and gnd pin.voltage pin is connected to output dc voltage and ground to ground.the adc pin is used to convert analog signal to digital signal and it is connected to pinA0 of atmega32.

Code for blinking of LED.
1].
https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/LED-blink/LED-blink-1.c

2].
https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/LED-blink/LED-blink-2.c

Write a code in which 0-7, 1-6, 2-5, … LED blinks. (to do)
1].
https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/LED-blink/LED-blink(8-2at%20a%20time).c

➔ In the above program we've written a code which blinks 8 LEDs, 2 at a time and runs in an infinite loop.

CIRCUIT DIAGRAM BELOW
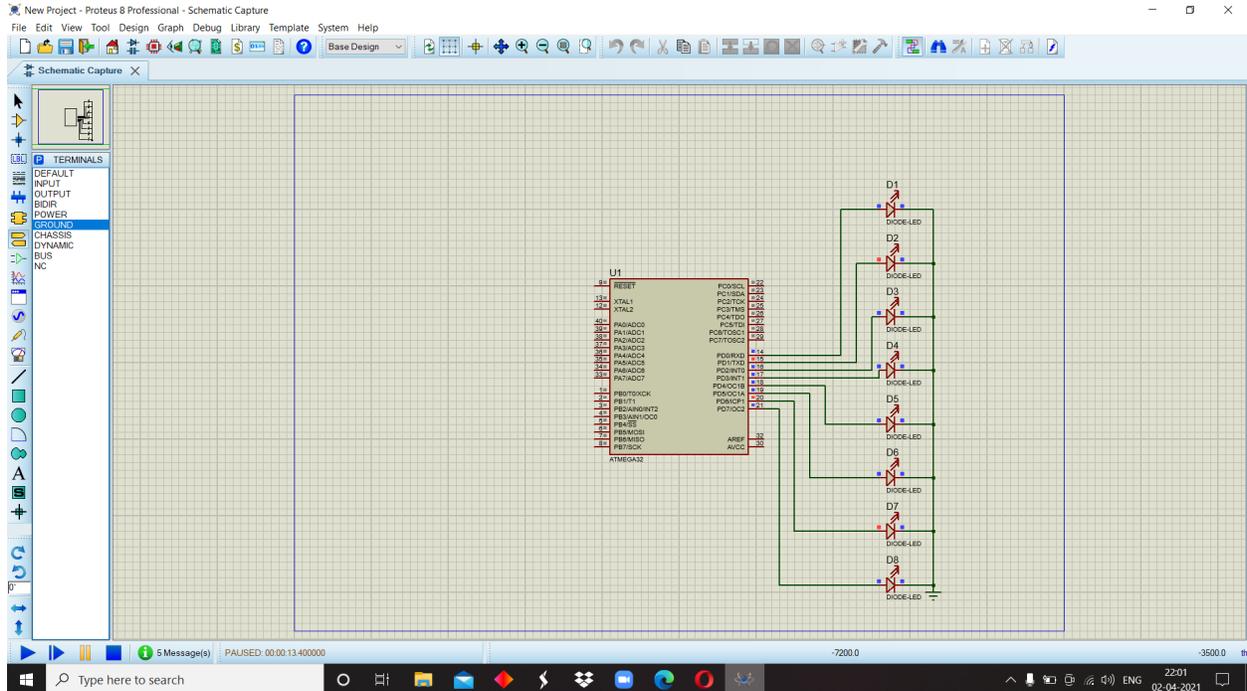
➔ What are the logical operations? (Bitwise operators)
We learnt about different gates - AND gate, OR gate. We also learnt about invertor.

➔ Watch newbie hack yt playlist (Especially vid no. 10 and 11)
https://youtube.com/playlist?list=PLE72E4CFE73BD1DE1

➔ https://www.microchip.com/content/dam/mchp/documents/parked-documents/as-installer-7.0.2542-web.exe (for atmel studio download)

➔ https://drive.google.com/file/d/1bKfTUkJstJ5l3h16DkSMrEglp_xVoz-0/view?usp=sharing (for proteus download)

➜ Logical operators {And (&), Or (|), Xor (^) and inverter (~)} and shift operator (<<,>>)

➜ Optimisation of previous code

➜ Different protocols
In our project we are using UART and I2C protocol only for communication between ATMEGA32, sensors, and GSM Module (SIM900D).

We are communicating with GSM Module via UART Communication.

BMP 180 is communicating via I2C communication protocol.

➜ Download GSM Module Library in proteus and error we faced
We encountered some issues with this library(1) while writing AT commands so decided to go with the other link(2)

    (1) http://www.theengineeringprojects.com/2016/03/gsm-library-proteus.html

    (2) https://www.electronicslovers.com/2016/10/how-to-add-gsm-library-in-to-proteus-7-8.html

➜ Download MQ2 Sensor Library in proteus
(We decided to use the MQ2 sensor because it detects the presence of Methane, Butane, LPG, Smoke.)
https://youtube.com/watch?v=UVac3Hq3LTs&feature=share

➔ Make a circuit diagram in the proteus consisting of all the circuits (excluding programming) - To do.
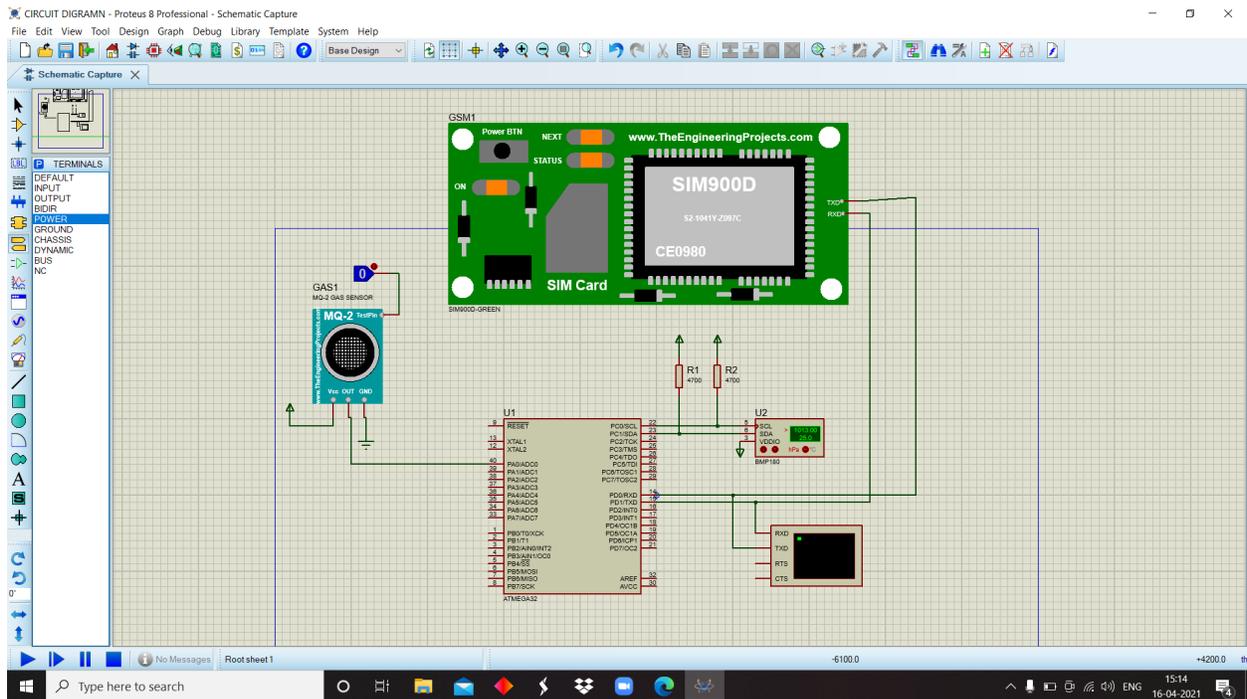
➔ Discussed about the previous problem of optimization of code.

USART, UART

FLOW CHART
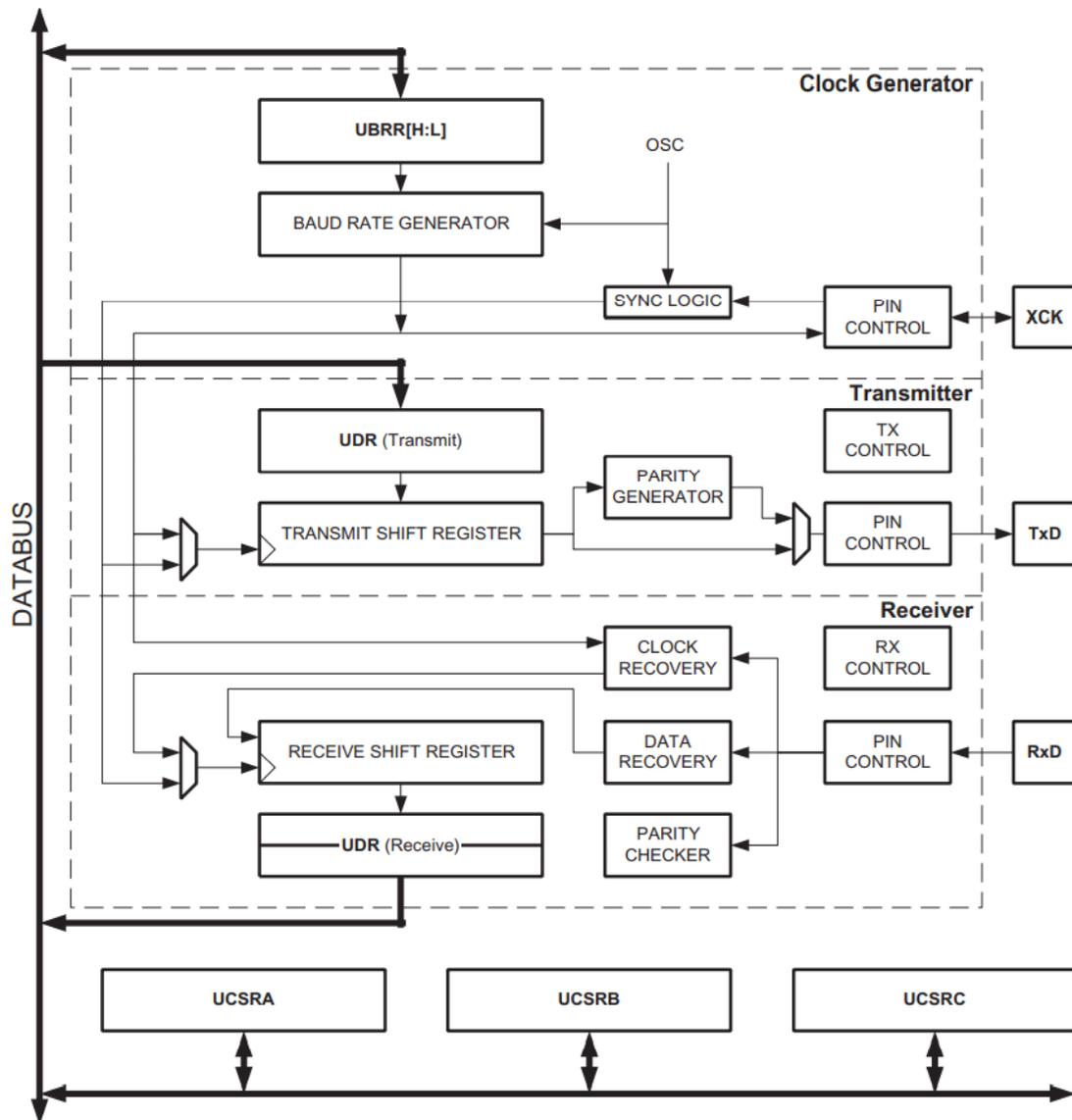
UART communicates using two wires. The RX pin of one device is connected to the TX pin of the other one and vice versa.
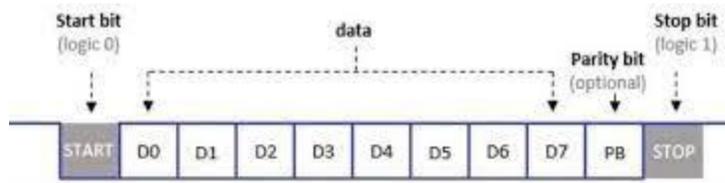
USART CONTROL AND STATUS REGISTER

UCSRB :
TXEN is set to 1
UCSRC:
URSEL,USBS,UCSZ0,UCSZ1 are set to 1 and UMSEL is set to 0.

LINKS for understanding UART Communication :

https://www.electronicwings.com/avr-atmega/atmega1632-usart

https://www.electronicshub.org/basics-uart-communication/

https://youtu.be/sTHckUyxwp8

Date 05/04/21

I^2C

LINKS for understanding I^2C Communication :
https://www.electronicwings.com/avr-atmega/atmega1632-i2c?fbclid=IwAR0nM8c6bYwZCFsIDVDwQyyLGNNNNJ_FEStp_8D6As4-NDcM6x64HfGe8

https://youtu.be/HGX457RA4IU

TWSR:TWI status register
➔ bits(7:3): TWS- TWI status:
➔ This five bit reflects the status of TWI logic and the two-wire serial Bus.
➔ Bit 2: reserved bit
➔ This bit always read as 0.
➔ Bit (1:0): TWPS - TWI prescaler bits:
➔ It can be read and written and controls bit rate prescaler.

TWCR: TWI Control register

➔ TWINT - interrupt pin
➔ TWEN - enable pin
➔ TWSTA - start pin
➔ TWSTO - stop pin
➔ TWEA - enable ack bit
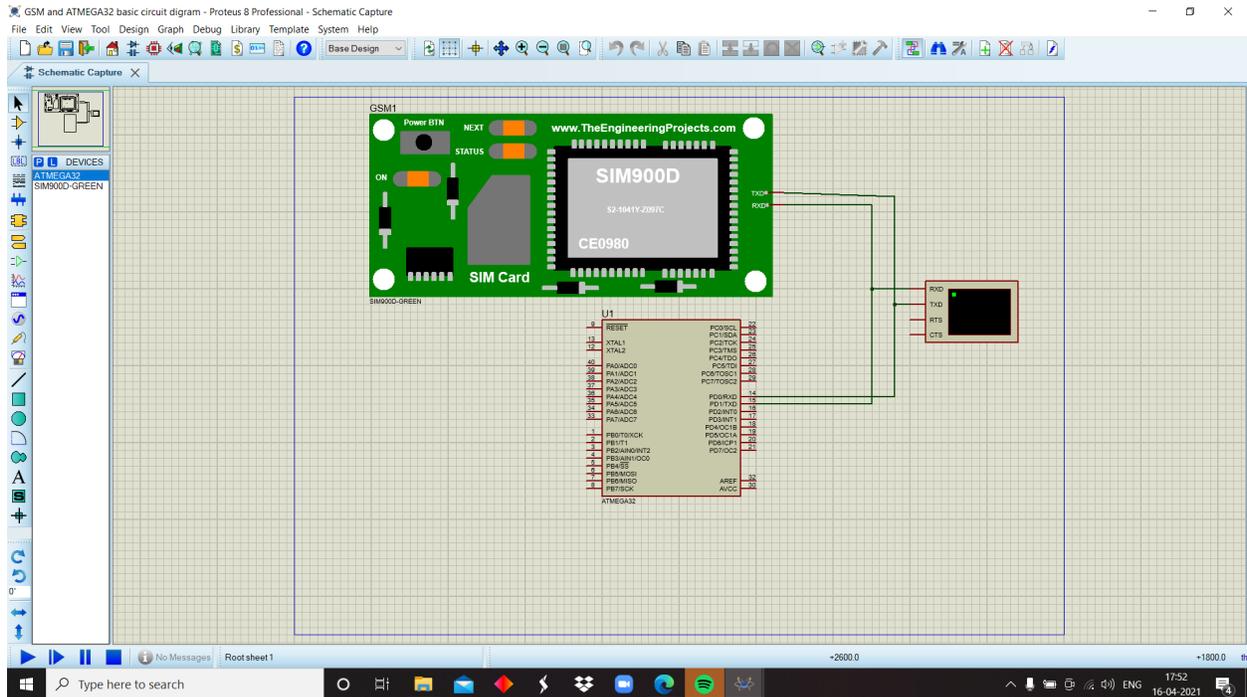
➔ UART Initialisation Code.

   (We added this code in GSM module code
   https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/GSM
   %2BAtmega32-(test-1).c)

➔ Read about different registers from the data sheet.

➔ Write UART initialization code.

➜ Continued on UART programming.

(All the progress were added here
https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/GSM
%2BAtmega32-(test-1).c)

➜ Programming for sending strings by UART protocol.

➜ We are sending msgs in the form of string, sending one character at
a time.

➜ Learned about AT commands for GSM Module.
https://en.wikipedia.org/wiki/GSM_modem#:~:text=A%20GSM%20mo
dem%20or%20GSM,identify%20themselves%20to%20the%20networ
k.

We decided to go with **SIM900D** which is easy to code, which we are
using to send msgs.

CODE for GSM and ATMEGA32

https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/GSM%2B
Atmega32-(test-1).c

CODE for MQ2 and ATMEGA32

https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/MQ2+Atmega32-(test-2).c

Here we are using LED so as to confirm that our program is working or not.
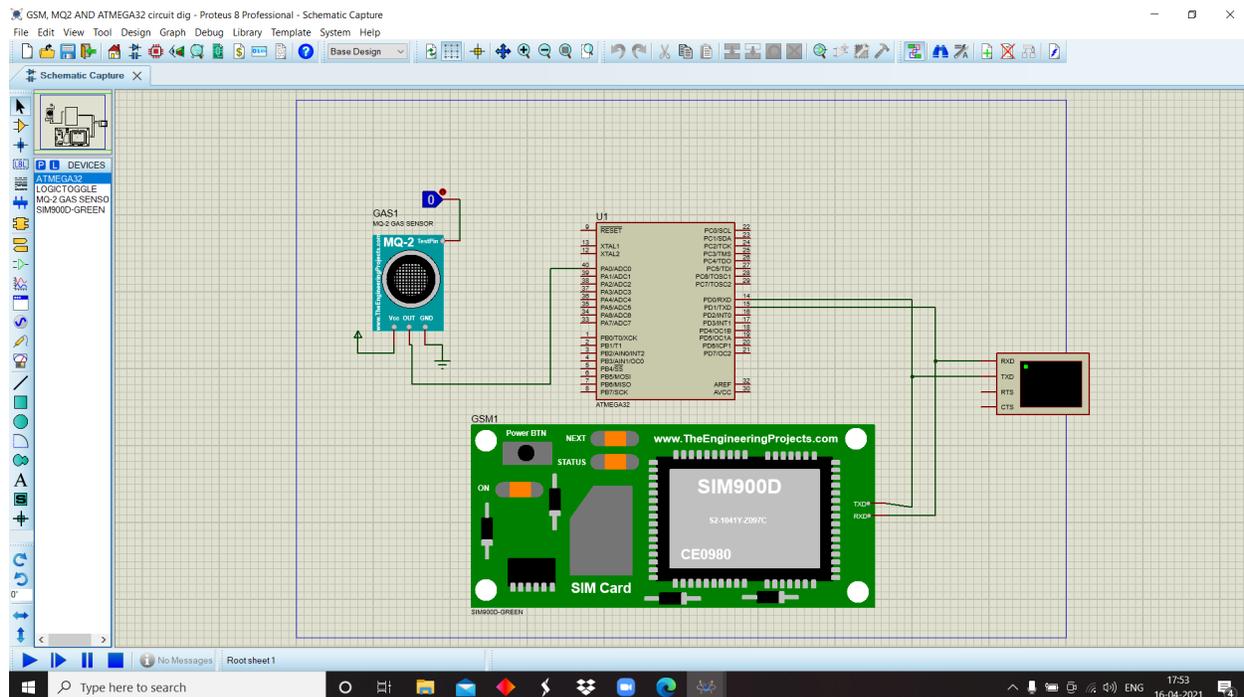
CODE for GSM, MQ2 and ATMEGA32

https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/GSM%2BMQ2%2BAtmega32(test-3).c

CODE we try to make without using library

https://github.com/Aniumbott/Fire-Alarm-System/blob/main/tests/I2C(without%20using%20library)(test-4).c

CODE made with help of library

https://github.com/Aniumbott/Fire-Alarm-System/tree/main/bmp180

→ But this wasn't seems working so we drop the idea of using bmp180
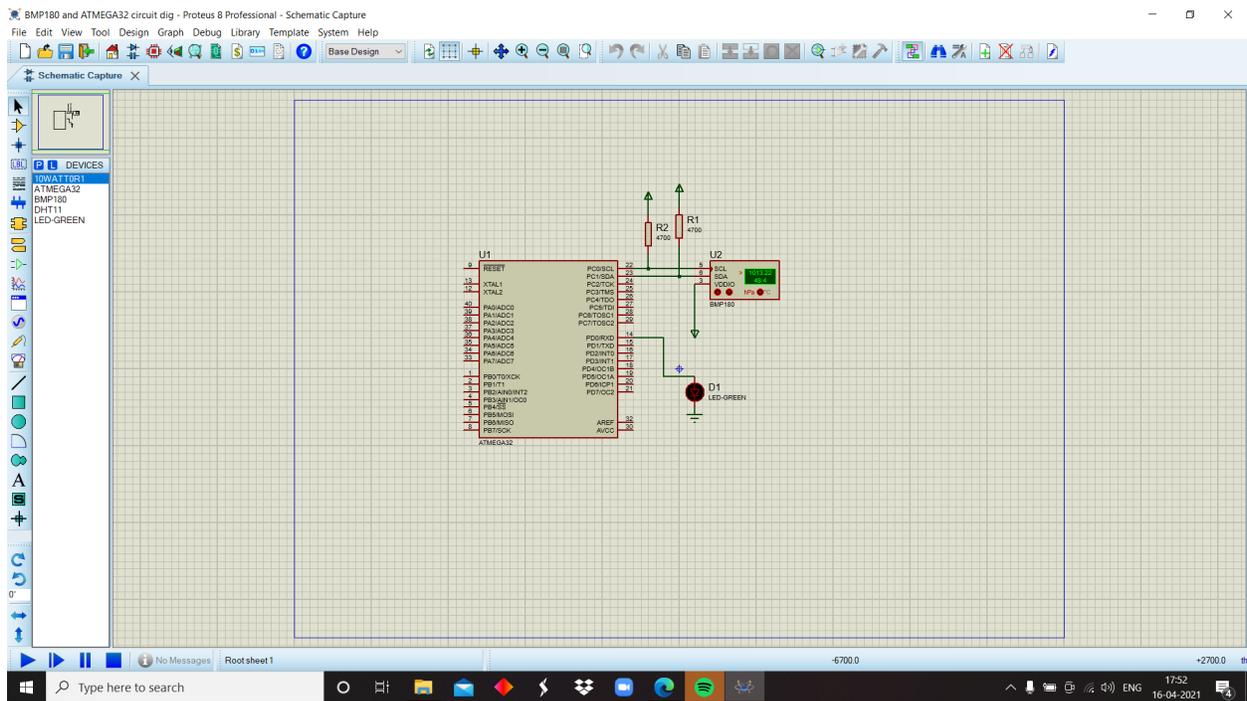
➔ We tried using another sensor temperature Tmp36.
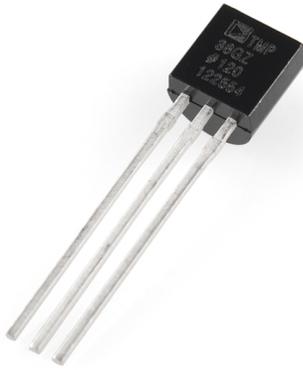
➔ which gives the voltage output in the analog signal we code for an ADC to convert it into a digital signal.

➔ And merge that with the MQ2+GSM+Atmega32 in such a way that a message will be sent as soon as temperature rises from 55 $^{\circ}$C code and finally the code is working perfectly.

Final CODE of our project

https://github.com/Aniumbott/Fire-Alarm-System/blob/main/final.c

FLOWCHART OF OUR PROGRAM
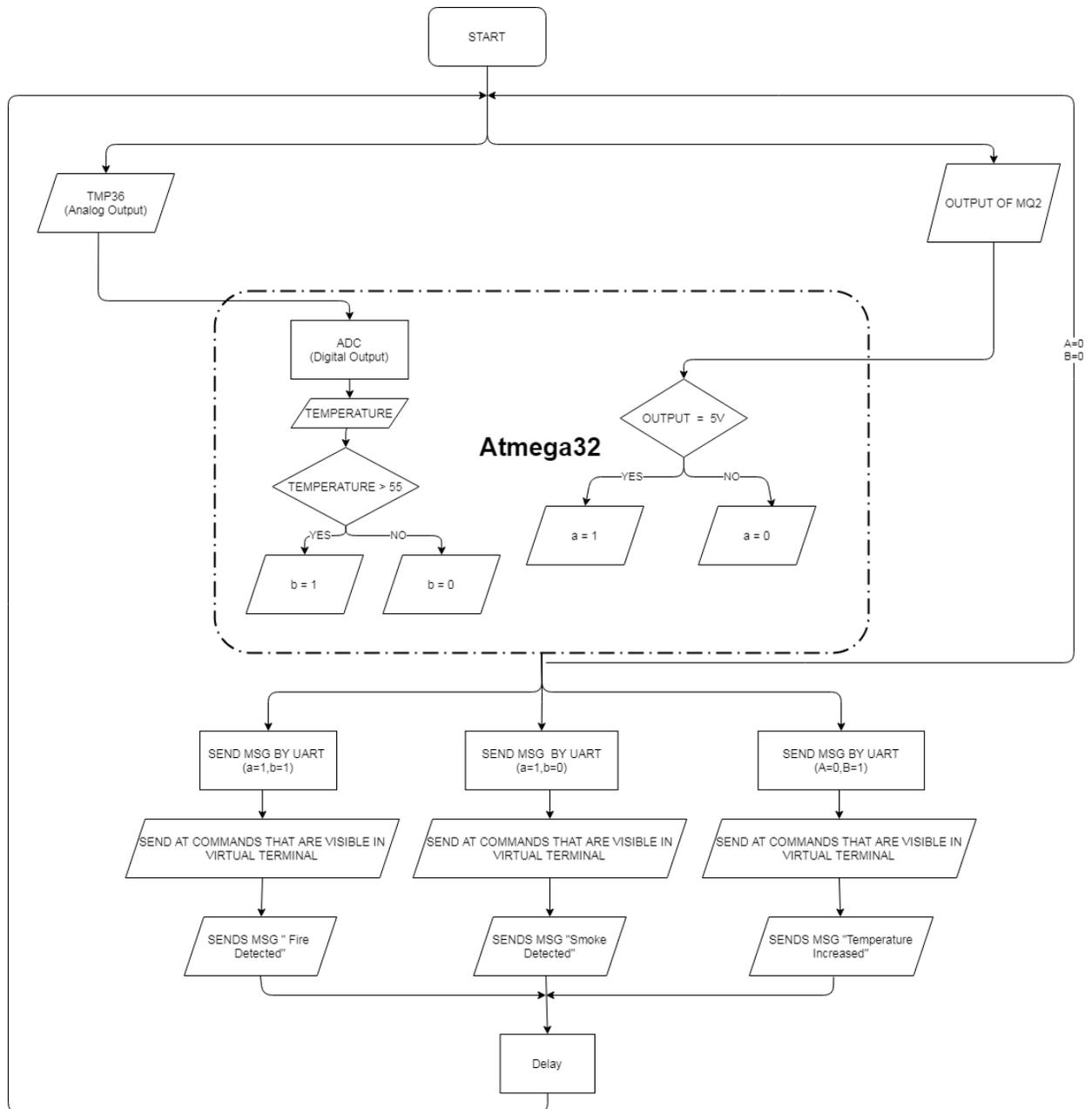
ERRORS AND SOLUTION

1].

Initially we faced an error though our code was correct and in the virtual terminal it was showing signs of euros. (**on 11th**)
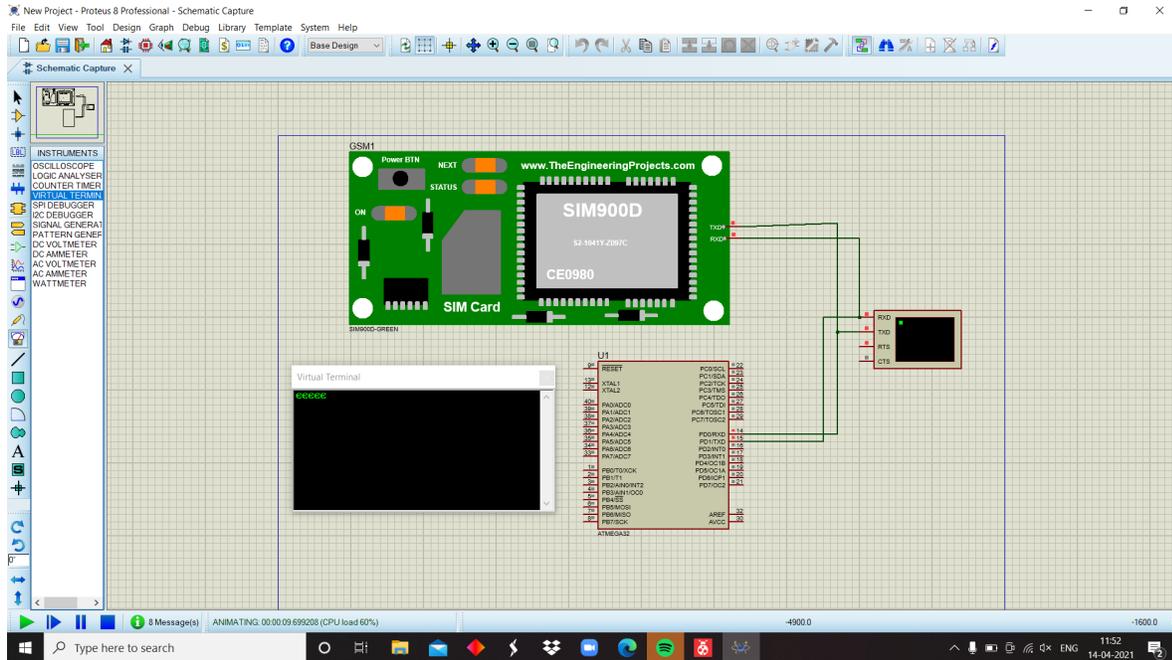
We solved it by changing the frequency of ATMEGA32 from 1MHz to 8MHz as shown in the photos below.
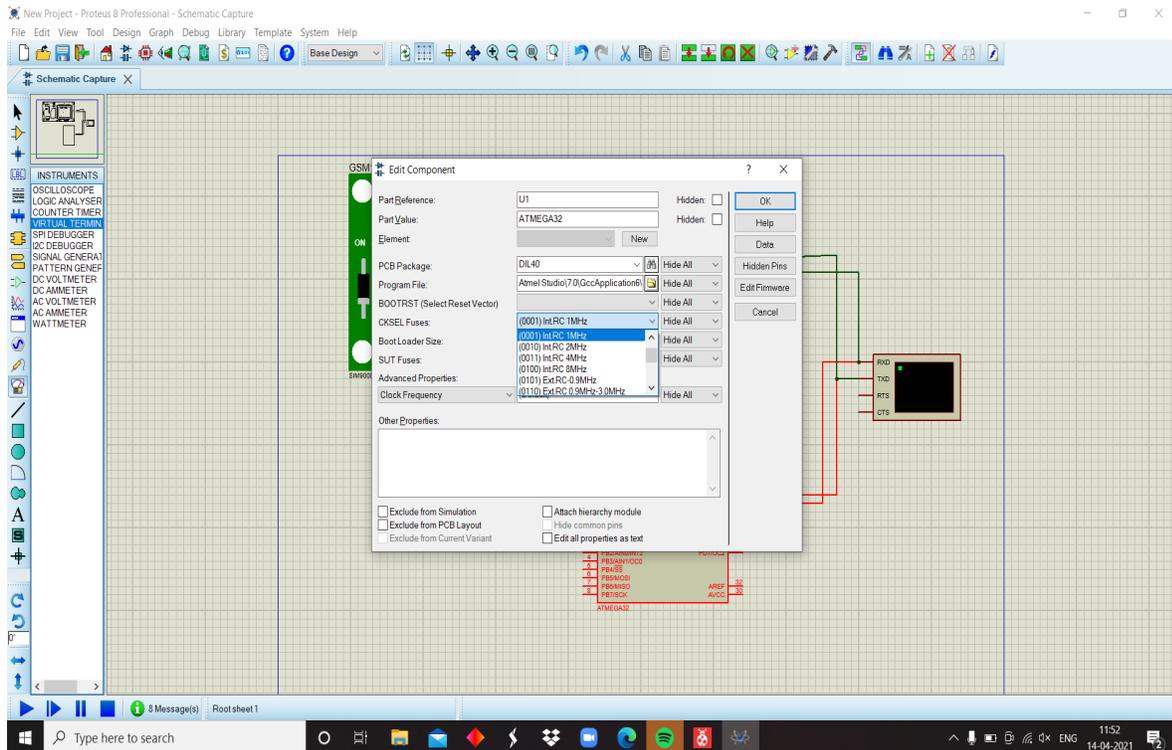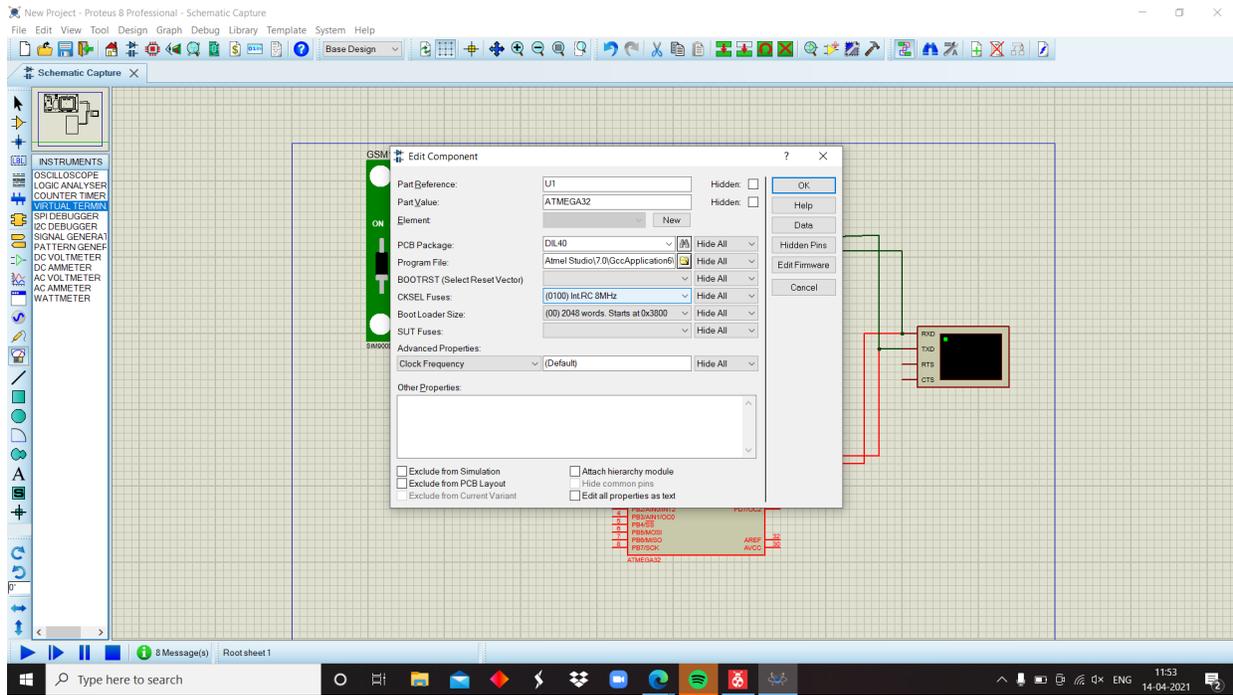
After doing this, the result was like this
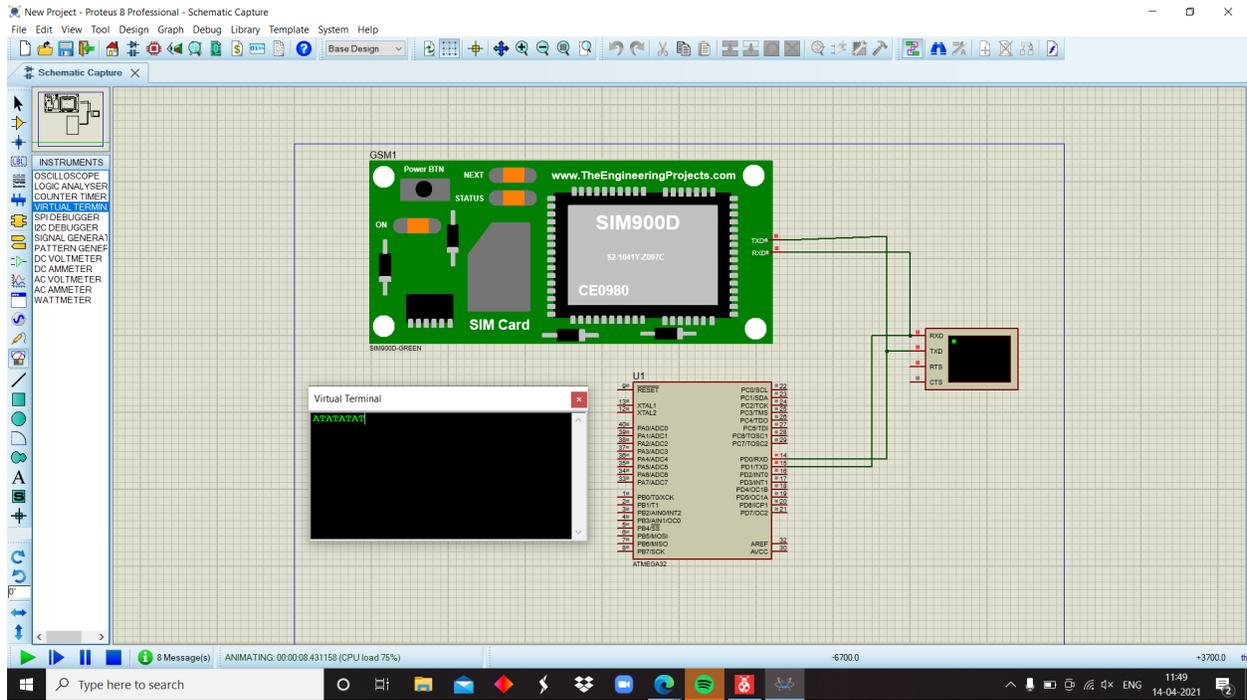
2].

If we open Proteus directly, it shows that no libraries are found.We have to run as administrator to access all the libraries.

3].

We were facing some problems with the GSM library, so replaced that downloaded library with a new one. Both links for the library are given above.

.

THE END