DRISHTI

A Revolutionary Concept

MAKERSPACE 6.0

FUNCTION GENERATOR

PROBLEM STATEMENT :

To design a circuit which can take amplitude and frequency at input and create an appropriate waveform output.

KEY OUTCOMES :

To output waveforms of different frequencies and types.

TEAM MEMBERS :

SRUSHTI BENDKE	(U21EC020)
ADITYA	(U21EC015)
AKSHAT SONI	(U21EC091)
SHUBH KAWA	(U21EC011)
AKANKSHA THIGLE	(U21EC058)
KRISHNA PATHAK	(U21ME076)
MANISH KUMAR SINGH	(U21EC123)
KRIYA JAIN	(U21EE106)
NEEL ALPESH MISTRY	(U21EE049)

MENTORED BY :

DHRUV SINGHI

DHRUVI TALSANIYA

AISHWARY MEHTA

INDEX :

INTRODUCTION :

Function generator is a device that generates different types of signal of different amplitude and frequency. When we say Different types of signal it includes sine wave, square wave, pulse, sawtooth wave, triangular wave.

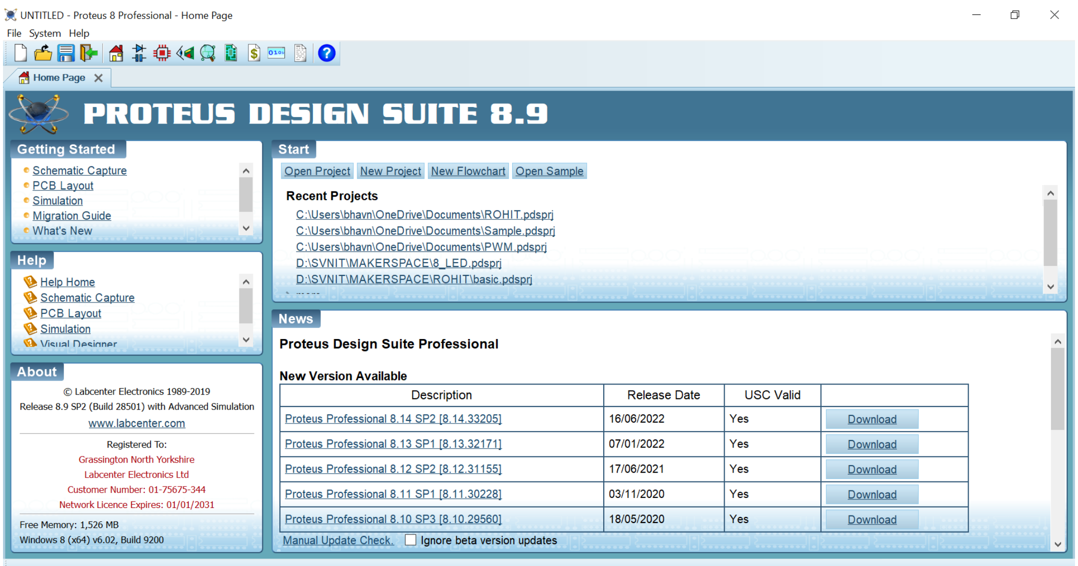
COMPONENTS AND SOFTWARE USED :

1) PROTEUS :

As we are going to deal with circuits in this project we would require a simulator which would help us create our own circuit and simulate it. So for this purpose we are using Proteus.

Here is the link for downloading Proteus on desktop :

https://drive.google.com/file/d/1bKfTUkJstJ5I3h16DkSMrEglp_xVoz-0/view?usp=sharing



2) MICROCHIP/ATMEL STUDIO :

Also we want to make a circuit which might be used by us for the long run. So we are gonna make a device that performs a particular function when feeded with required inputs. In our case we can say that we are providing Amplitude and Frequency of the type of wave we want to generate. So we would require a microcontroller which helps us perform this. So the Microcontroller we are choosing is ATmega 32. And also we would require a simulator to program the microcontroller. So for this purpose we are using Microchip Studio / Atmel Studio.

Here is the link for downloading Microchip Studio on desktop :

https://drive.google.com/file/d/1-ccbPNV-yz3hyys_LvNBvExkVJSHF_pO/view?usp=sharing

3) MICROCONTROLLER :

So there are many operations a microcontroller can perform. For knowing the insights of the same it is advisable to read the Datasheet of Microcontroller we are using. Also it is advisable to read the Datasheet of any component we use so as to get some prerequisite knowledge about operating points of the component.

So here is the link for Datasheet of ATmega 32 :

https://www.alldatasheet.com/datasheet-pdf/pdf/77378/ATMEL/ATMEGA32.html

Now let's understand the name ATmega.

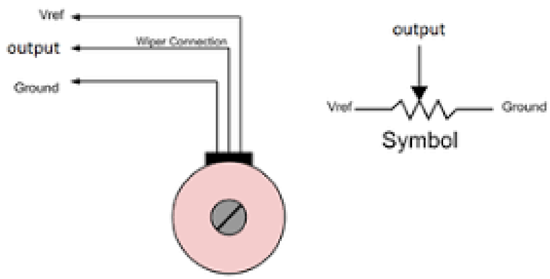
ATmega XX : Over here AT stands for Atmel, mega stands for mega AVR, XX are numbers and denote the XXkb of flash memory. For example if we have ATmega 32 then it means we have 32kb of flash memory.

4) BIPOLAR JUNCTION TRANSISTOR :

BJTs are the most commonly used basic amplifier. They are made up of semiconductor materials. Mainly they are of two types : NPN transistor and PNP transistor. We commonly use the NPN transistor. In these projects we have used the Voltage Divider bias configuration of BJT which provides us with amplification of the input wave provided to it. Voltage Divider bias offers high Stabilisation and hence Q point does not change. The Emitter Resistor has the control over the voltage and current that region receives. So dependence on Beta value of transistor decreases.

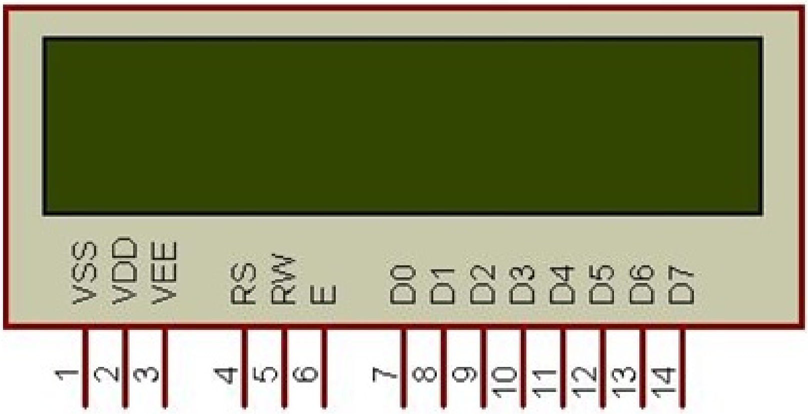
5) POTENTIOMETER:

The potentiometer is an instrument used for measuring the unknown voltage by comparing it with the known voltage. It can be used to determine the emf and internal resistance of the given cell and also used to compare the emf of different cells.



6) LCD:

LCD stands for Liquid Crystal Display. We can use the I2C protocol to use it/control it with the help of ATmega32. I2C stands for Inter-Integrated Circuit. It is a bus interface connection protocol incorporated into devices for serial communication. It was originally designed by Philips Semiconductor in 1982. Recently, it is a widely used protocol for short-distance communication. It is also known as Two Wired Interface(TWI). We have usedLCD 16x2 (LM016L).

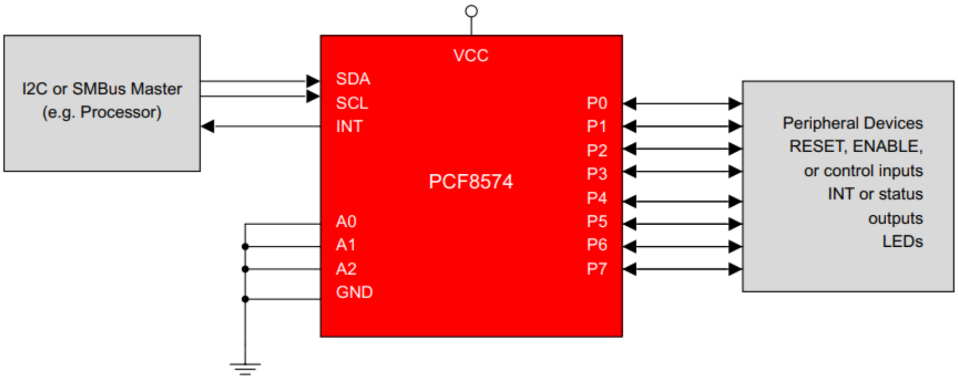


7) PORT EXPANDER :

We have used PCF8574 Remote 8-Bit I/O Expander for I 2C Bus.

Features:

- Low Standby-Current Consumption of 10 µA Max
- 2C to Parallel-Port Expander
- Open-Drain Interrupt Output
- Compatible With Most Microcontrollers
- Latched Outputs With High-Current Drive
- Capability for Directly Driving LEDs
- Latch-Up Performance Exceeds 100 mA.



CONCEPT :

1) SPECIFICATION OF ATMEGA32 :

Let us discuss about some of the specification of ATmega 32:

ALU :

ALU stands for Arithmetic-Logic Unit. They play a very important role in performing all the operations of the ATmega32. They help in smooth working of all the 32 general purpose registers. Within a single clock cycle, many arithmetic operations are performed between the registers. ALU operations are divided into three sections : Arithmetic, Logical and Bit Functions.

Status Register:

The status register contains information about the last arithmetic operation executed. This information can be used in a programme to perform conditional operations which will result in faster execution and more compact code as compared to dedicated conditional operations.

It consists of 8 bits.

Pin Description :

1. BIT 0- Carry Flag(C): It indicates a carry in arithmetic operation.
2. BIT 1- Zero Flag(Z): It indicates if an arithmetic operation has zero result:
3. BIT 2- Negative Flag(N): It indicates if an arithmetic operation has a negative result.
4. BIT 3- Two's Complement Overflow Flag(V): It indicates if the result is too large to fit in a 7-bit number and two's complement arithmetic has to be used to show it.
5. BIT 4- Sign Bit(S): It indicates if the arithmetic operation has a negative or positive sign. It is an exclusive OR(XOR) of Negative Flag(N) and Two's Complement Flag(V) and will be set only if either of the two is set but not both.
6. BIT 5- Half Carry Flag(H): It indicates overflow between upper and lower nibbles of a register and is used when dealing with Binary Coded Digits(BCD).
7. BIT 6- Bit Copy Storage(T): It indicates success or failure of a custom subroutine.
8. BIT 7- Global Interrupt Enable(I): It is used to enable interrupts in the programme. An interrupt is a way for an external event to pause the current processor's activity, so that it can complete a brief task before resuming execution where it left off

Different type of clock sources in Atmega32:-

1.Low Power crystal Oscillator
2.Full swing Crystal oscillator
3.Low frequency crystal oscillator
4.Internal 128khz RC oscillator
5.Calibrated Internal RC oscillator
6.External clock

LOW POWER CRYSTAL OSCILLATOR:-

Pins XTAL 1 and XTAL2 are input and output pins.It supports a quartz crystal or a ceramic resonator.

This crystal oscillator is a low power oscillator,with a reduced voltage swing on the XTAL2 output.it is not capable of driving other clock inputs and may be more susceptible to noise in noisy environments.

FULL SWING CRYSTAL OSCILLATOR:-

Pins XTAL 1 and XTAL2 are input and output pins.

It is a full swing oscillator and gives a rail to rail swing on the output pin.this is useful in noisy environments.Power consumption is more than that of low power crystal oscillator.

LOW FREQUENCY CRYSTAL OSCILLATOR:-

It is optimized for use with a 32.768 watch crystal.

It has two capacitors connected to the output and input pin respectively.

INTERNAL 128KHZ RC OSCILLATOR:-

It is a low power oscillator providing a cloc of 128 khz.The clock may be selected as the system clock by programming the CKSEL fuses to 11.It is not designed for high accuracy.

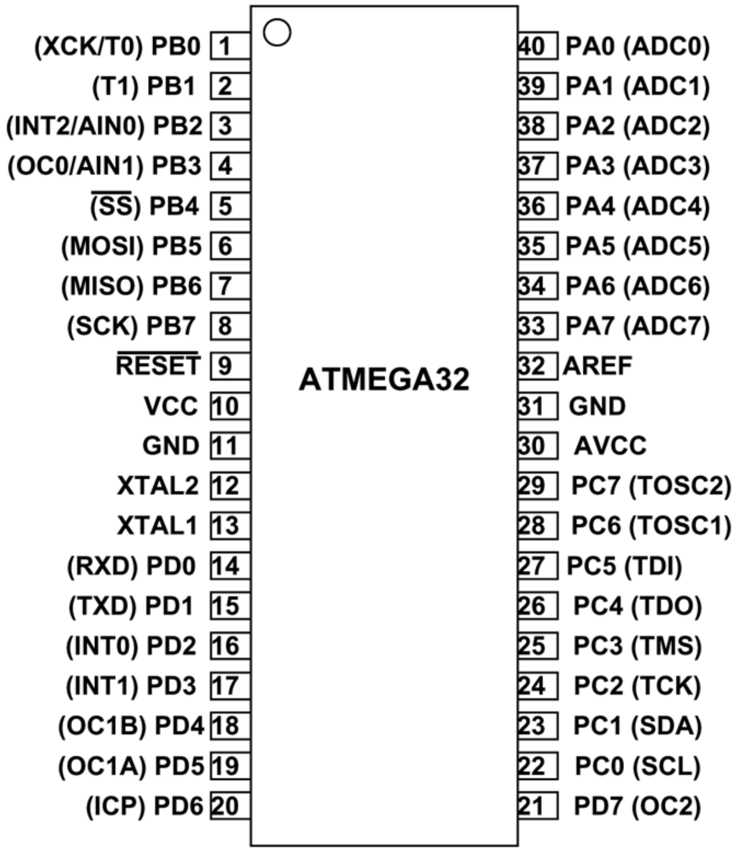
CALIBRATED INTERNAL RC OSCILLATOR:-

The internal RC oscillator provides an approximate 8MHx clock. It is voltage and temperature dependent,this clock can be very accurately calibrated by the user. If selected, it will operate with no external components.

EXTERNAL CLOCK:-

To drive the device from an external clock source the XTAL 1 pin is connected to the clock source and the XTAL2 pin can be used as GPIO PB7.

When applying an external clock,it is required to avoid sudden changes in the applied clock frequency to ensure the stable operation of the MCU.



2) PWM

Pulse-width modulation(PWM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. We have used it here to generate square waveforms.

Duty Cycle :

A duty cycle is the fraction of one period in which a signal or system is active. Here, to generate a brightness of 50% and 75%, we were required to set the duty cycle at 50% and 75% respectively.

3) FAST PWM MODE :

The fast Pulse Width Modulation or fast PWM mode (WGM01:0 = 3) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to MAX then restarts from BOTTOM.

Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase-correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

Period of the FAST PWM mode can be half of the phase correct PWM wave. Frequency of the fast PWM mode can be twice as high as the phase correct PWM wave.

FREQUENCY -

$$\boxed{f_{OCnPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot 256}}$$

DUTY CYCLE -

- **Duty cycle = ((OCR2+1) /256) X 100 (Non Inverting Mode)**
- **Duty cycle = 100 – (((OCR2+1) /256) X 100) (Inverting Mode)**

4) PHASE PWM MODE

The phase correct PWM mode (WGM01:0 = 1) provides a high resolution phase correct PWM waveform generation option. The phase-correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to MAX and then from MAX to BOTTOM. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

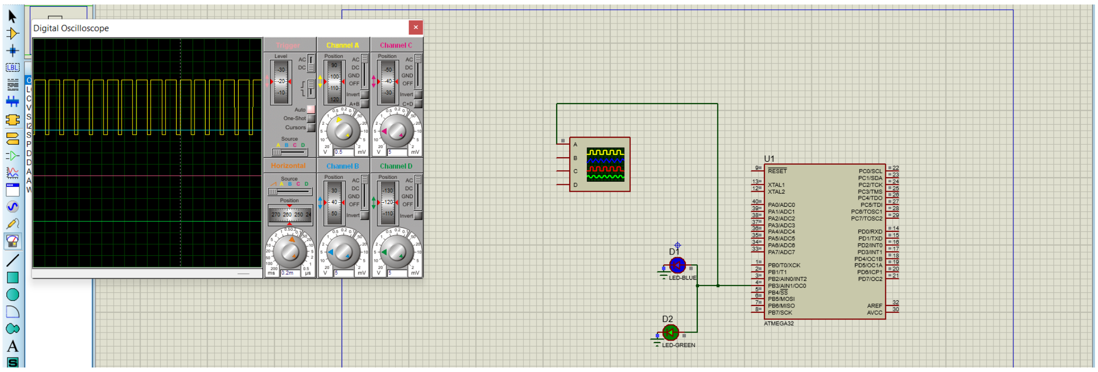
FREQUENCY -

$$\boxed{f_{OCnPCPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot 510}}$$

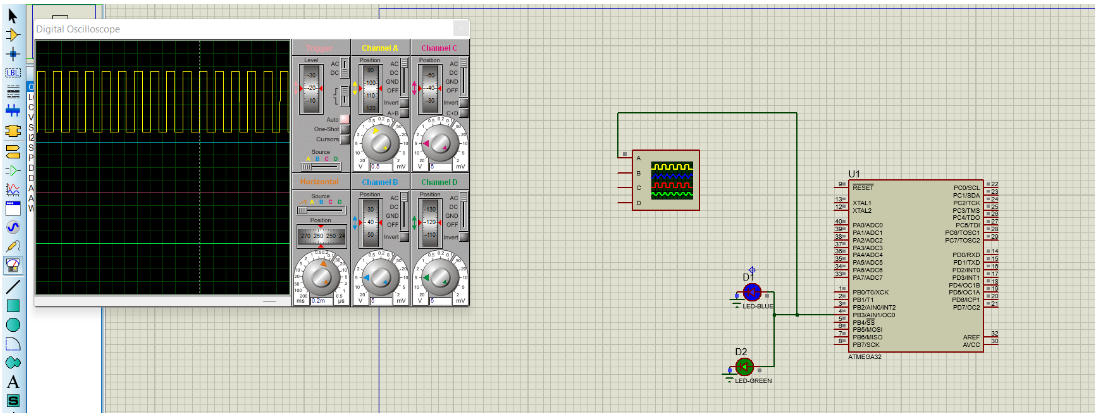
DUTY CYCLE -

- **Duty Cycle = (OCR2/255) X 100 (Non Inverting Mode)**
- **Duty Cycle = 100 – ((OCR2/255) X 100) (Inverting Mode)**

75% brightness…



50% brightness



5) I2C concept:-

I2C is a serial bus interface connection protocol. It is also called TWI(two wire interface) since it uses only two wires for communication, that is two wires called SDA and SCL. AVR based atmega 32 has a TWI module made up of several submodules as shown in the figure.

I2C works in two mode namely:-

1 .Master mode

2. Slave mode

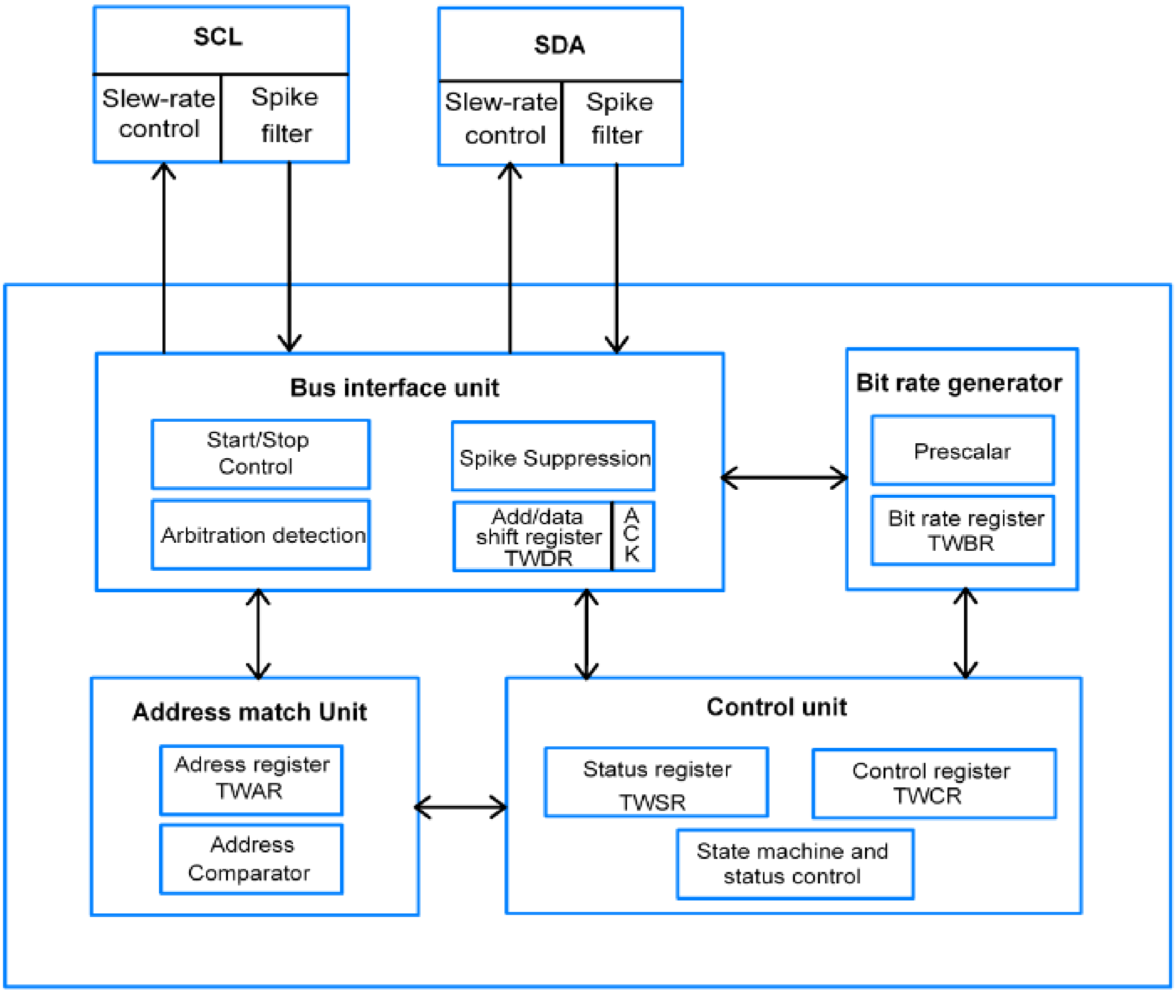
SDA & SCL pins:-

Used to interface the TWI based external peripherals and microcontroller

Bus Interface Unit:-

The bus interface unit containing Start/Stop control is responsible for the generation and detection of START,REPEATED START and STOP conditions.

ACK bit receives ack/nack in transmitter mode and it is generated through software in receiving mode.



Address match unit:-

In slave mode, the address match unit recieves an incoming 7 bit address and compares with the adress in TWAR register to check whether it matches or not and upon match occur it intimate to control unit to take necessary action.

BIT RATE GENERATOR UNIT:-

It controls the SCL period in the master mode to generate SCL frequency.

The formula is :- SCL frequency = (CPU CLK frequency) / (16+2(TWBR)*4^TWPS)

6) PORT EXPANDER:-

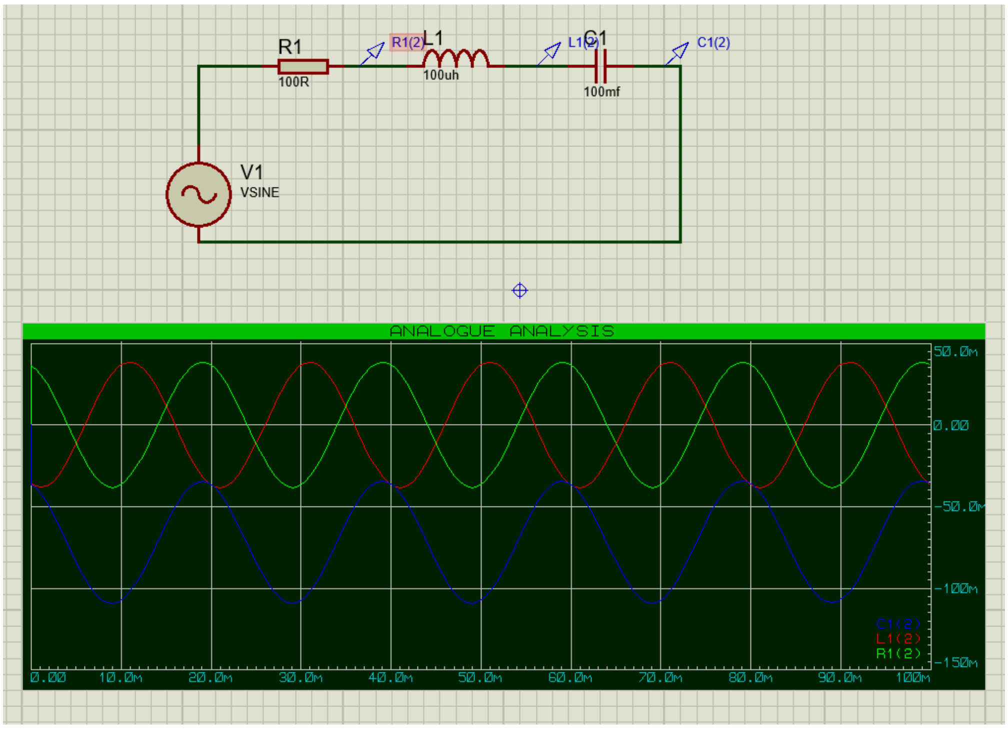
The PCF8574 provides general purpose remote I/O expansion via the two-wire bidirectional I2C-bus((serial clock SCL),serial data(SDA)).

The devices consist of eight quasi-bidirectional ports, 100kHz I2C-bus interface, three hardware address inputs and interrupt output operating between 2.5V and 6V. The quasi-directional port can be independently assigned as an input to monitor interrupt status or keypads or as an output to activate indicator devices such as LEDs. System master can read from the input port or write to the output port through a single register.

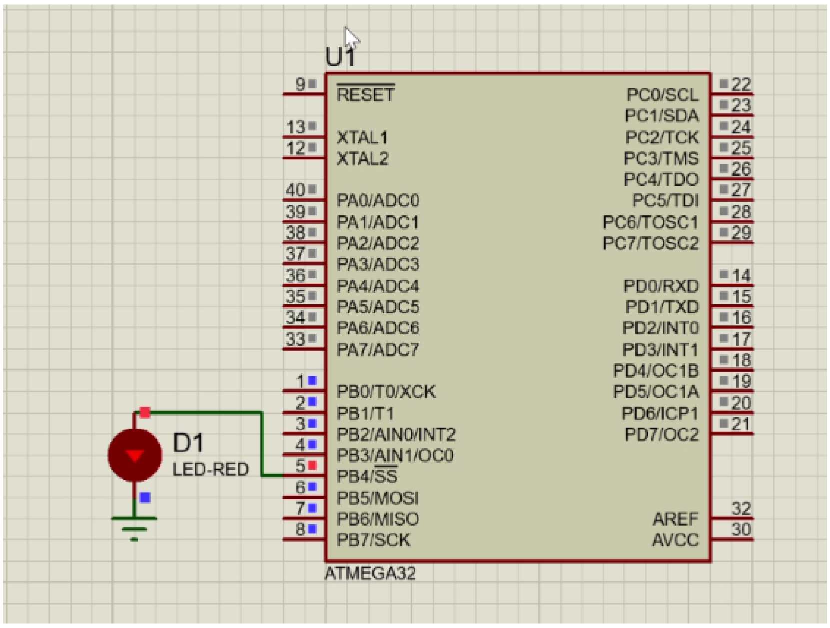
The low current consumption of 2.5 µA(typical,static) is great for mobile applications and latched output ports directly drive LEDs.

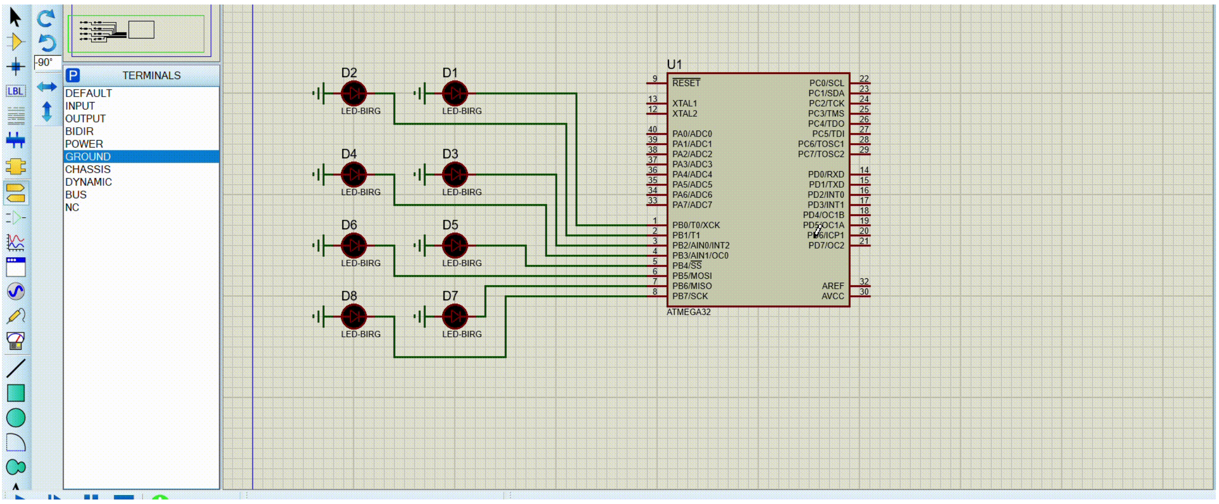
IMPLEMENTATION :

We started with a simple task of connecting an RLC Series and Parallel Circuit and obtaining it's waveform

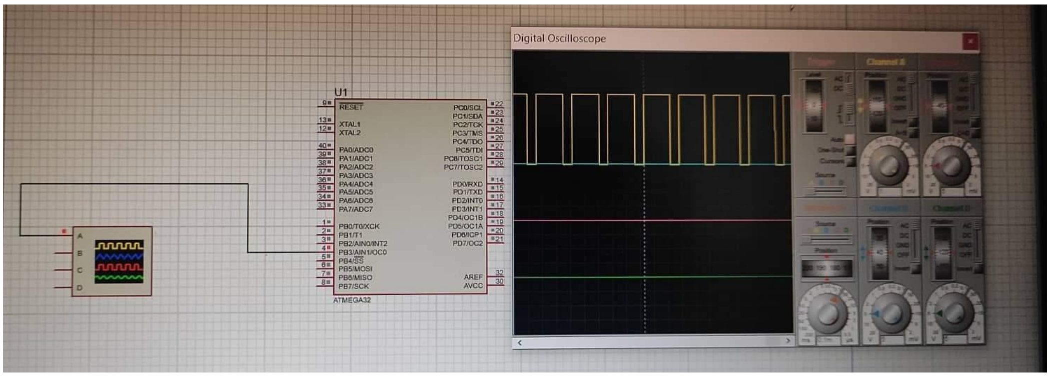


We started coding in ATMEL,and our first task was to blink an LED and

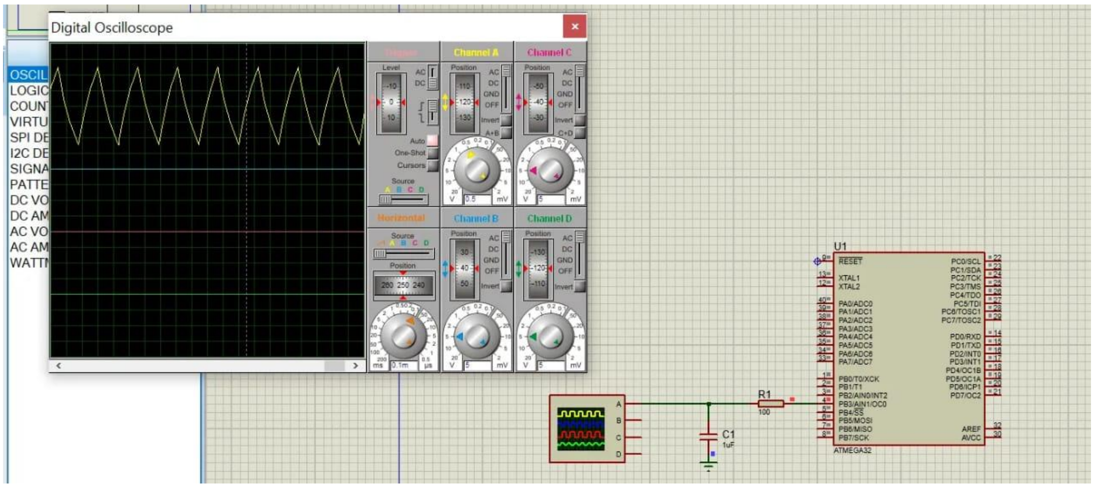




We were then taught about PWM and how to generate a PWM

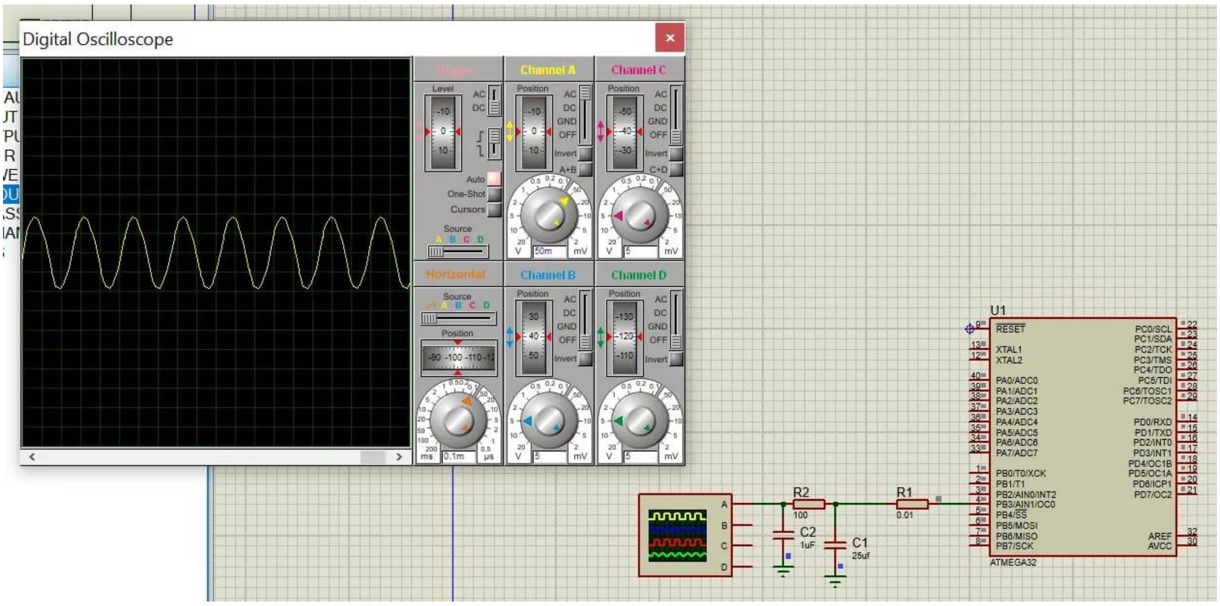


Being aware of the concept of low pass filter, we applied it in our existing circuit and generated different types of waves as shown in the images below



with 1 low pass filter we get a triangular wave.

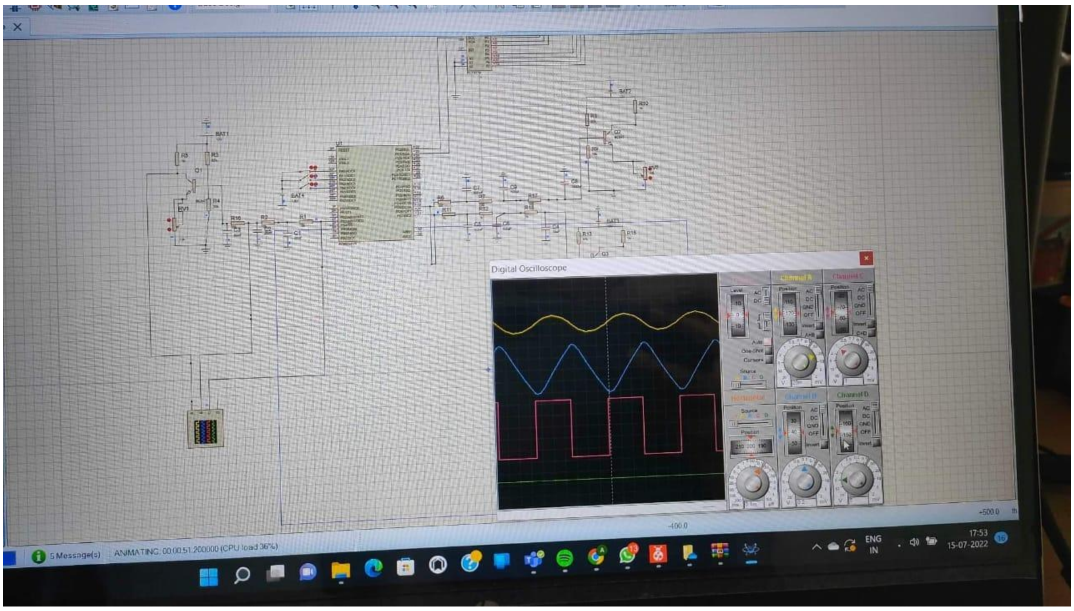
By increasing the number of low pass filters we can obtain a smooth sine wave.



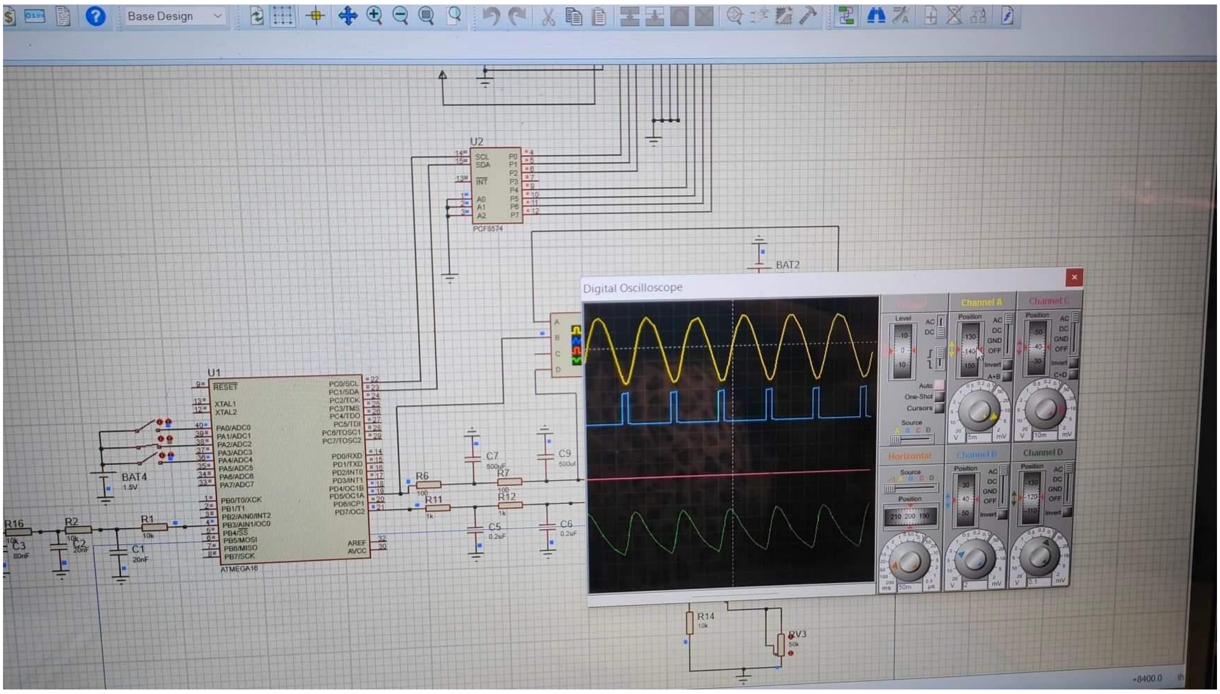
In the above circuit, we use two low pass filters to generate a smooth sine wave

Using different timers available on Atmega32 we are generating waves with different frequencies.

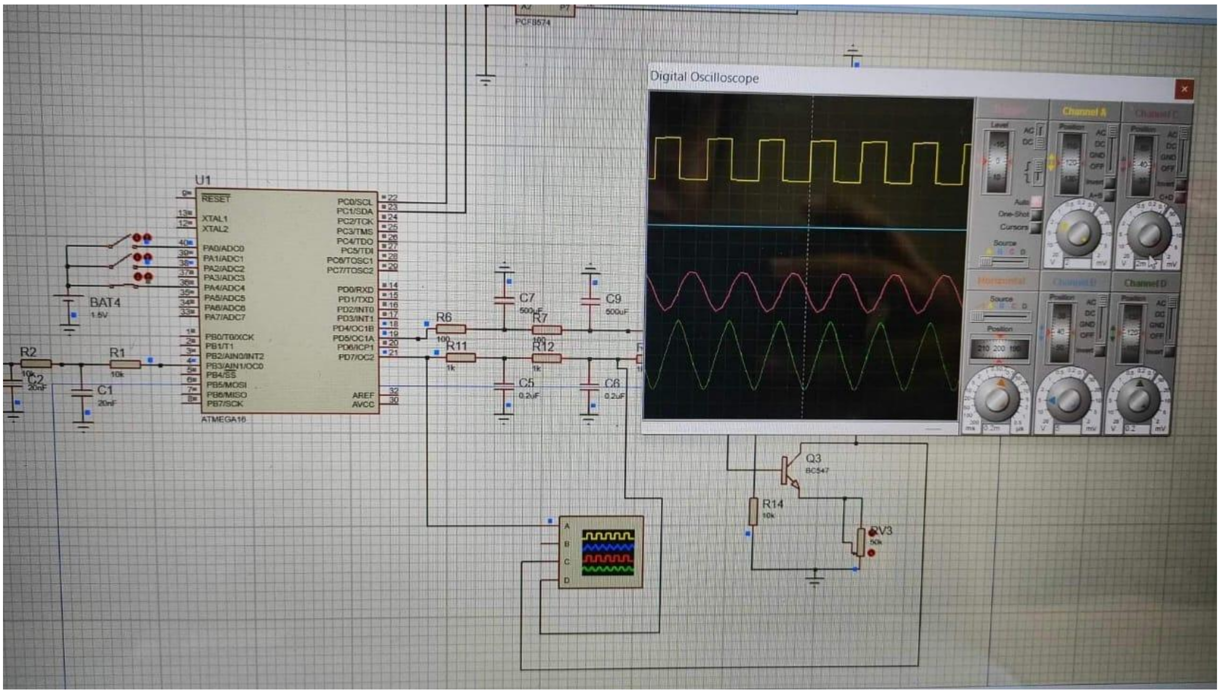
The waves being formed by using timer 0:-

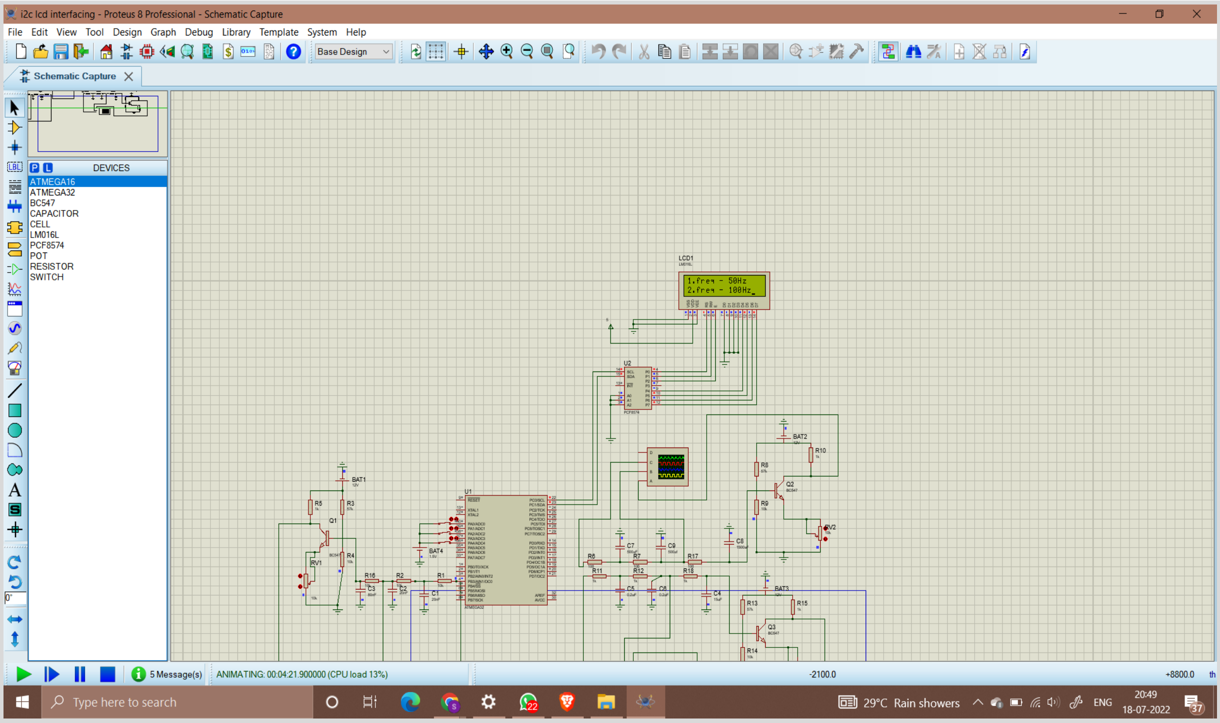


Now lets see the waves formed by timer 1:-



Timer 2:-

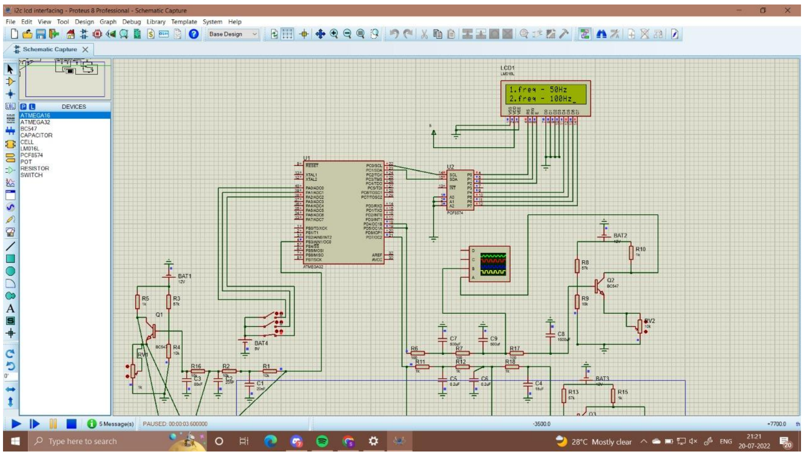




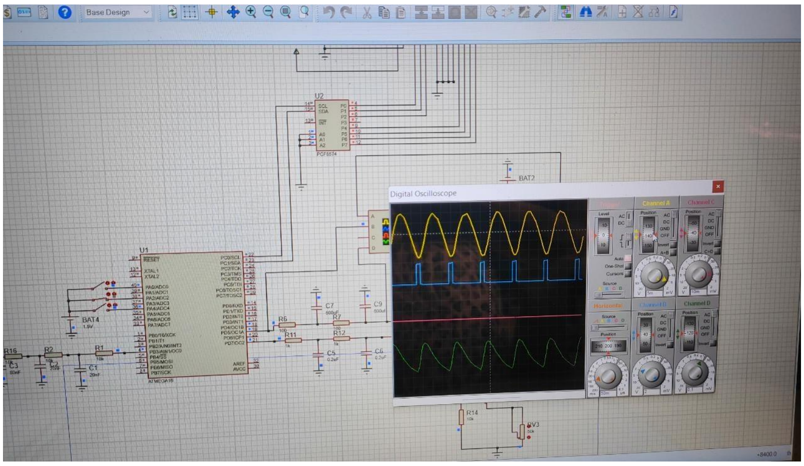
The above image shows the display on LCD

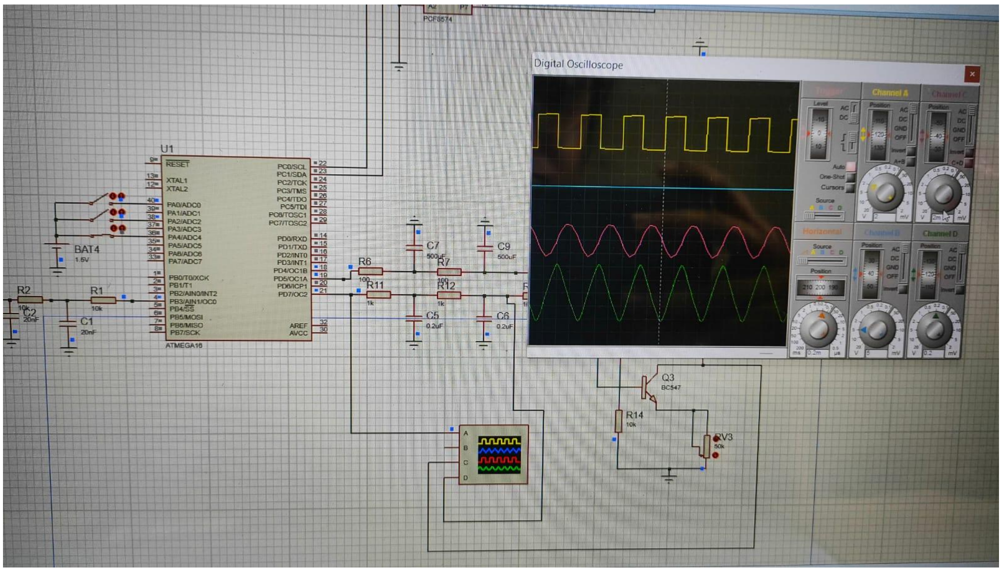
FINAL PROJECT OUTLOOK :

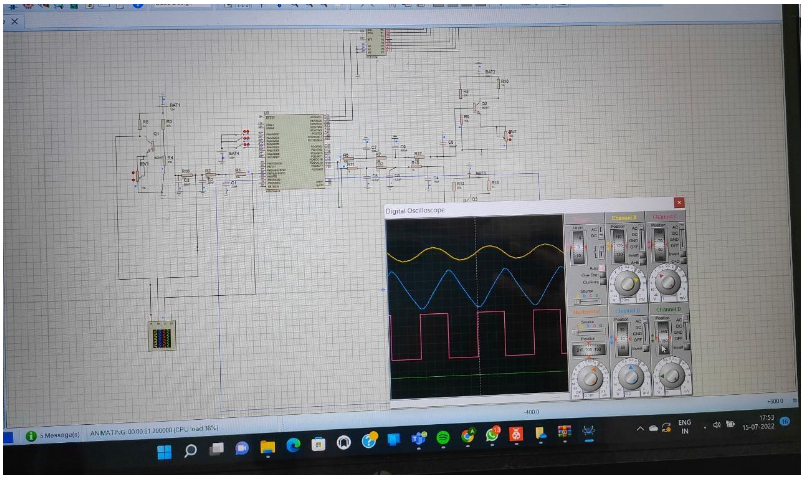
Final circuit :

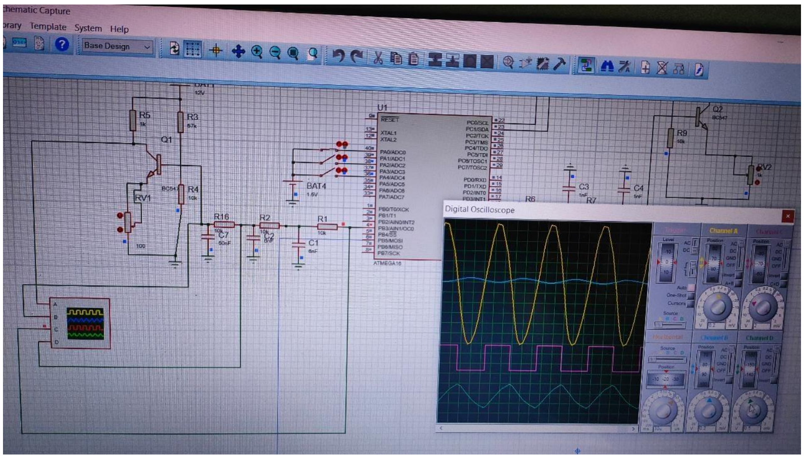


LCD interfacing waves :









TIMELINE :-

1st week:-

1. Objective of function generator
2. Learn about atmega 32 from the datasheet
3. Installation of ATMEL studios and Proteus
4. Understood about bitwise operators such as bitwise OR, bitwise AND,etc.
5. Made series and parallel RLC circuit and obtained its waveform on oscilloscope
6. Implemented code of blinking LEDs
7. We were taught about PWM and duty cycle
8. Implemented code to blink LED at 50% and 75% brightness

2nd week:-

1. Discussed and cleared doubts in the code of PWM.
2. Searched for ways to display an output on LCD.
3. Understood the Fourier synthesis.
4. Started learning about the I2C protocol.
5. Discussed about integrator and DAC.
6. Learned about I2C modes and its 4 units.
7. Discussed about low pass filter and sine wave generation by DAC.
8. Introduction to UART.
9. Learned about amplification of sine waves using BJT through voltage divider biased circuit.
10. Get to know about look up table method and frequency modulation methods
11. Discussion about how to generate signals of different frequencies using the different timers available in ATmega.

3rd week:-

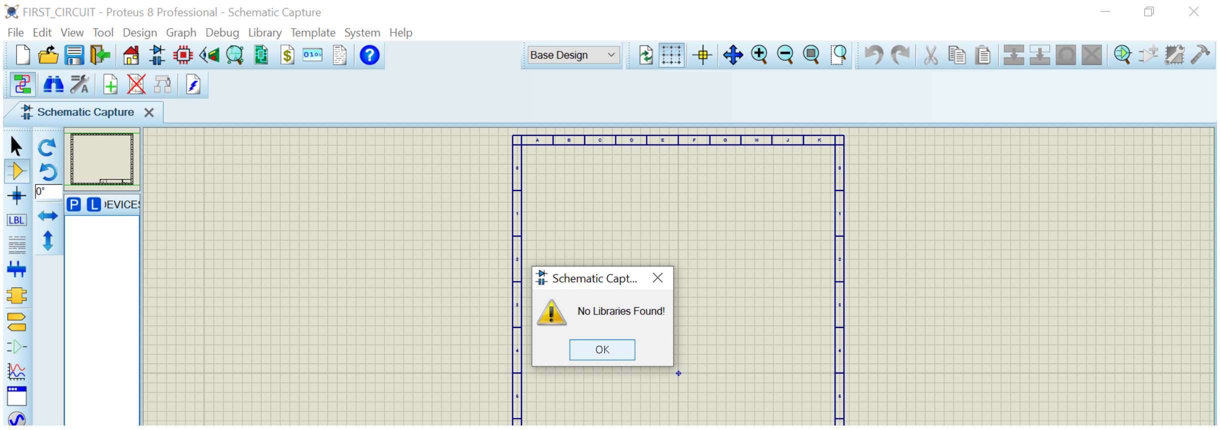
1. Discussed about switch application and pins of LCD.
2. Learned about PWM mode configuration of Timer 1.
3. Discussed about connections in I2C.
4. learned about code of switching applying if else statements.
5. Get to know about the code to I2C.
6. Debugging of the project.

4th week:-

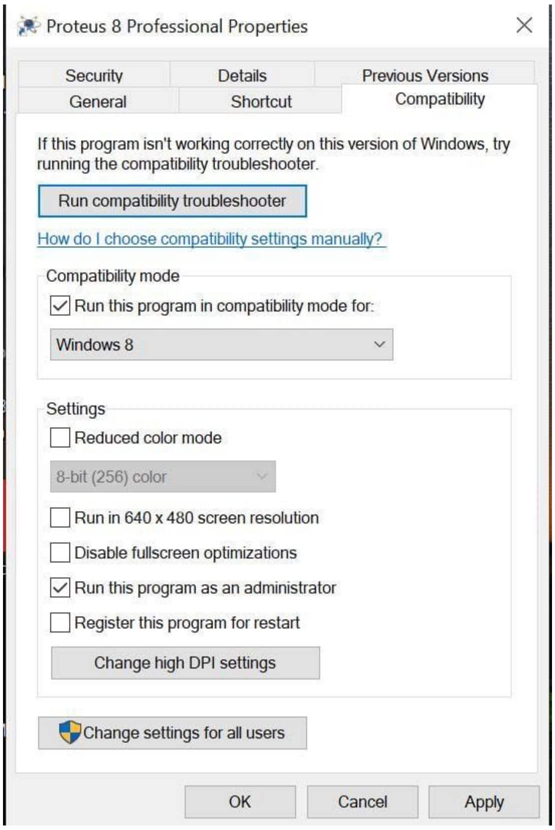
1. Final Documentation modification.
2. Presentation Preparation.

ERRORS FACED AND THEIR SOLUTION :

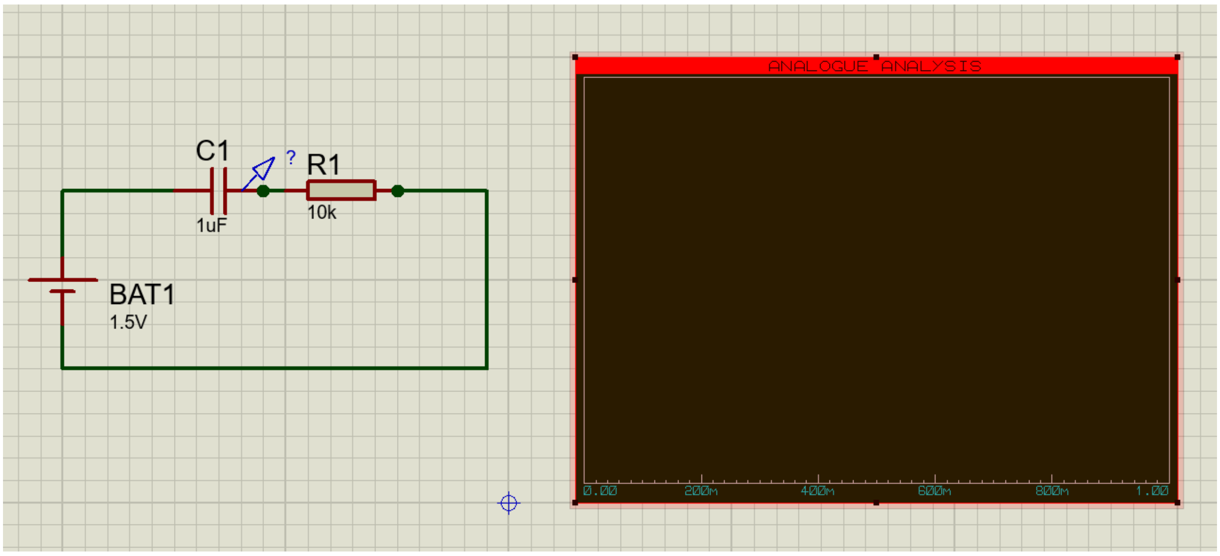
While running Proteus for first time :



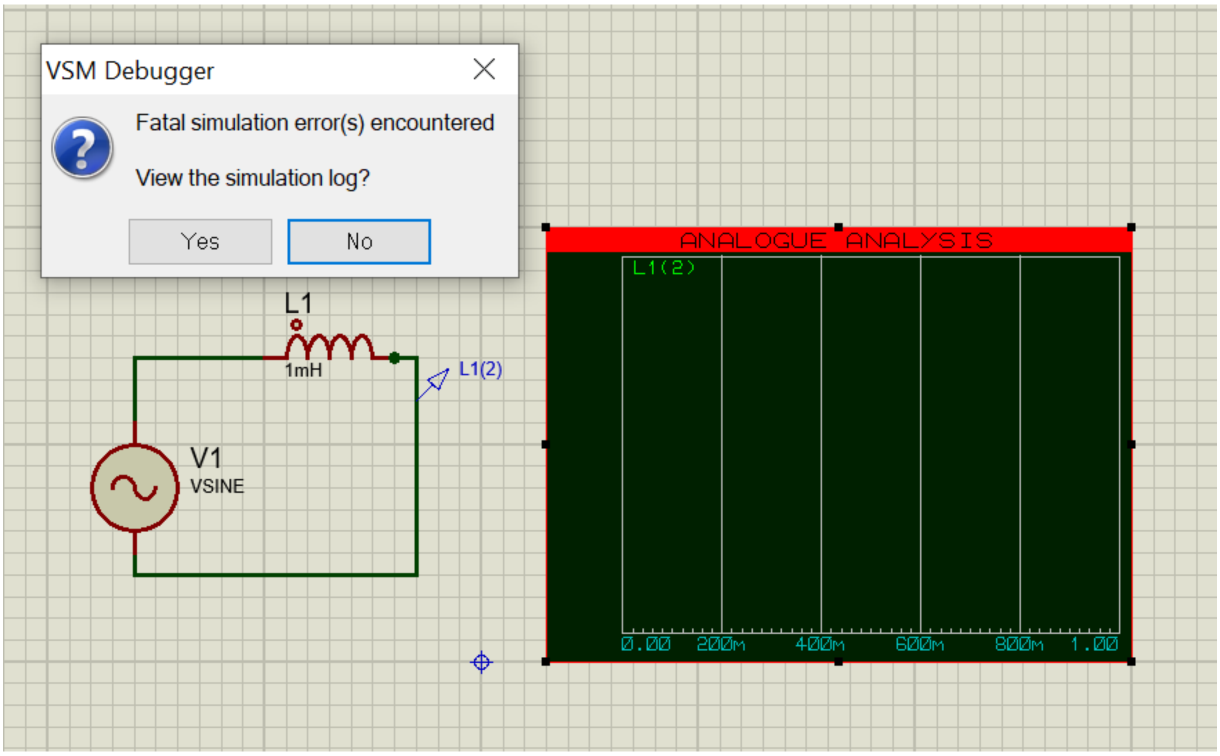
For anyone facing these error make the following changes in properties of Proteus 8 :



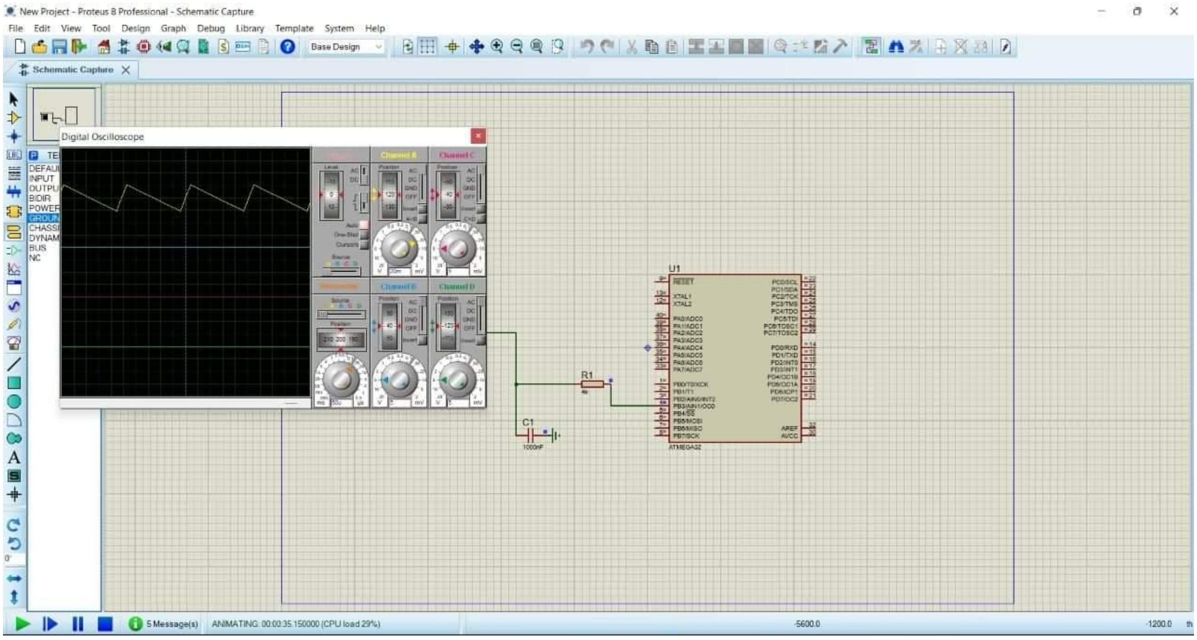
While making Circuit on Proteus 8 do not insert a probe on the component / device or at the end of component / device. It will result in an error.
It will not simulate the graph



Also We need to insert a resistor in series with an inductor if we are directly connecting it to ac supply.

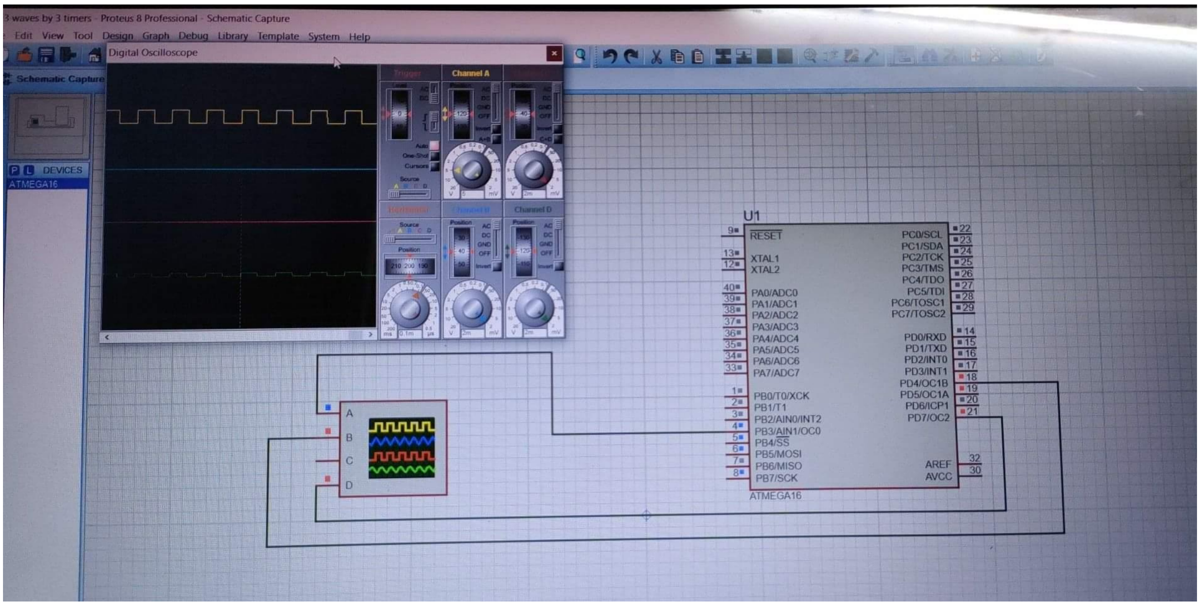


Error faced while generating sine wave using PWM



Here we need to do the calculations to get proper sine wave

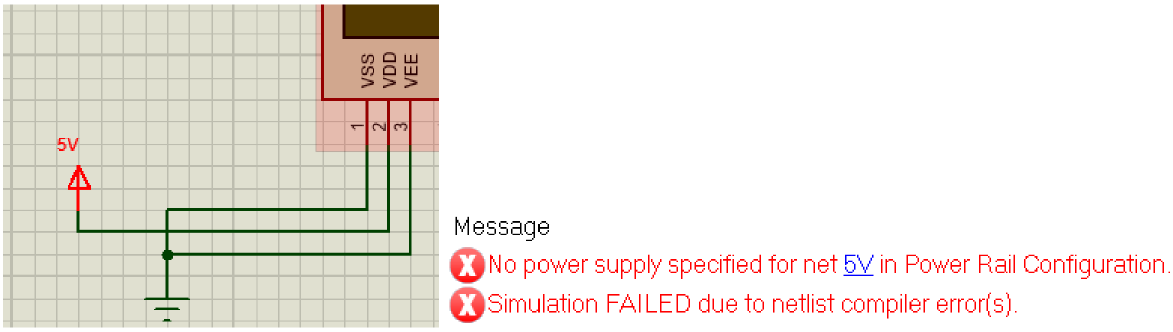
While generating PWM by 3 frequencies, the waveform of timer 1 was missing



Solution:

Use com1b1 as a tccrb register instead of tccra

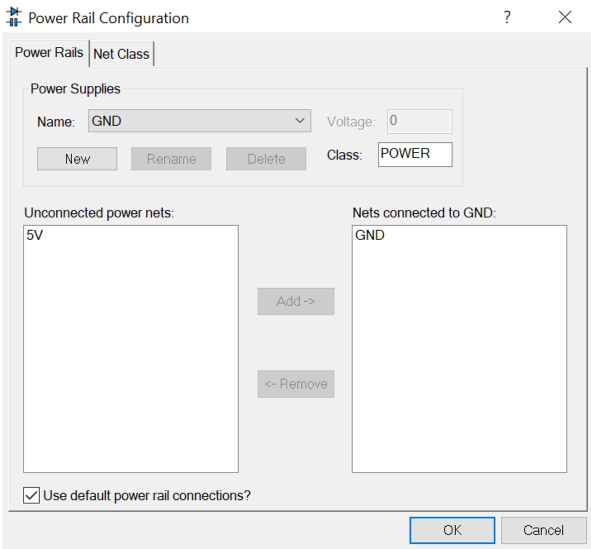
If we are not configuring the Power Rail we might face the following error.

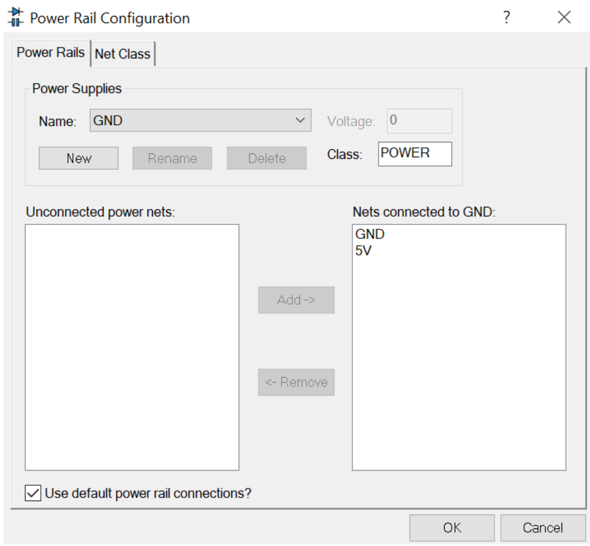


To resolve it Open the Configure Power Rail option in Design.

Add the power rail from Unconnected Power Rail to Nets connected to Rail.

Before : After :





Atmega16

Once Check the device you are using before you import code in it / while making connections so as to avoid unnecessary errors while running the simulation.

Multiple definition

Timer1 and timer2

References:

- How to build a function generator? https://www.allaboutcircuits.com/projects/how-to-DIY-waveform-generator-analog-devices-ad9833-ATmega328p/
- Generation of sine waves using atmega: https://www.youtube.com/watch?v=W2viTr7ZDwk
- I2C in atmega 32: https://www.electronicwings.com/avr-atmega/atmega1632-i2c
- About I2C : https://maxembedded.com/2014/02/inter-integrated-circuits-i2c-basics/
- About LCD: https://embeddedcenter.wordpress.com/ece-study-centre/display-module/lcd-16x2-lm016l/
- Port Expander : https://www.ti.com/lit/ds/symlink/pcf8574.pdf