Student Minor Research Project

PC BASED OSCILLOSCOPE USING ARDUINO

Under RUSA 2.0 Scheme

(Through Ch.S.D.St.Theresa's College for Women (Autonomous), Eluru, AP)

Submitted by

Ms K Vasanthi , III B.Sc. MECs (Reg.No.11705046)

Ms N Prathiba, III B.Sc. MECs (Reg.No.11705047)

Ms G Hema, III B.Sc. MECs (Reg.No.11705048)

Under the guidance of

Dr Ch Kanaka Rao

Lecturer & Project Advisor

Department Of Electronics

SRI Y N COLLEGE

(AUTONOMOUS)

Thrice Accredited by NAAC at 'A' Grade

Recognized by UGC as "College with Potential for Excellence"

Narsapur-534275, AP, India

December-2019

Department Of Electronics

SRI Y N COLLEGE

(AUTONOMOUS)

Thrice Accredited by NAAC at 'A' Grade

Recognized by UGC as "College with Potential for Excellence"

Narsapur-534275, AP, India

CERTIFICATE

This is to certify that the project work entitled " **PC Based Oscilloscope using Arduino** *" is bonafied work carried out by Ms K Vasanthi(Reg.No: 11705046), Ms N Prathiba (Reg.No: 11705047), Ms G Hema (Reg.No: 11705048), submitted in Third Year of the degree B.Sc. in Electronics during the year 2019-20 is an authentic work under my supervision and guidance.*

To the best of my knowledge, the matter embodied in the project work has not been submitted to any other College/Institute.

Date: 29-12-2019

Dr Ch Kanaka Rao
Project Advisor
Department of Electronics

ACKNOWLEDGEMENT

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our Project advisor, ***Dr Ch Kanaka Rao,*** *Lecturer, Department of Electronics,* ***Sri Y N College (Autonomous), Narsapur*** *in bringing this report to a successful completion.*

We are grateful to ***Mr K Vinaya Phaneendhra****, Lecturer, Department of Electronics for permitting us to make use of the facilities available in the department to carry out the project successfully. Last but not the least we express our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.*

Finally we extend our gratefulness to one and all who are directly or indirectly involved in the successful completion of this project work.

Ms. K Vasanthi	**Ms. N Prathiba**	**Ms. G Hema**
III.B.Sc.MECs	III.B.Sc.MECs	III.B.Sc.MECs
Reg. No 11705046	Reg. No.11705047	Reg.No.11705048

DECLARATION

We, the undersigned, declare that the project entitled "**PC Based Oscilloscope using Arduino**", being submitted in Third Year of Bachelor of Science in Electronics, Sri Y N College (Autonomous), is the work carried out by us.

Ms. K Vasanthi	**Ms. N Prathiba**	**Ms. G Hema**
III.B.Sc.MECs	III.B.Sc.MECs	III.B.Sc.MECs
Reg. No 11705046	Reg. No.11705047	Reg.No.11705048

Contents **Page No.**

List of Figures & Tables Page No.

1. ABSTRACT

The main aim of this project is to develop the PC Based Oscilloscope with the Arduino interfaced with PC or Laptop and an Arduino board as the hardware for signal acquisition. Now a days PC based measurements have become more affordable and easy to use, thus opening the door for “virtual instrumentation”.

In this project we use an Arduino board which is used to capture multiple input values and pass them via USB serial port to a PC or Laptop that takes the values, decode it and display them on the screen. This system will operate up to 5 kHz frequency and it operated on 230 A.C. The Arduino board, the heart of the oscilloscope, reads the values from its inbuilt analogue-to-digital converter (ADC) and pushes these to the PC via USB port. This application works as the front-end to plot input signals as waveforms on your computer screen.

2. INTRODUCTION TO OSCILLOSCOPE

Oscilloscope is enlightened as one of the most advantageous instrument available for testing the circuits. It displays the signals at different points in the circuit. One way to examine an electronic system is to observe signals at the input and output and check the operating conditions of each system block. Traditionally, signals were analyzed using electromagnetic oscilloscope graph which later was substituted by Cathode Ray Oscilloscope for the purpose of getting higher bandwidth. Cathode ray oscilloscope is a hardware instrument which is based on Cathode Ray Tube (CRT), designed to display voltage variations (periodic or otherwise). But they are expensive, bulky, and have difficulties on displaying low frequency waveforms. Hence they are replaced by Digital Storage Oscilloscope (DSO) .

Digital Storage Oscilloscopes are widely used in recent market environment because of their higher bandwidth.The most attractive feature of this type of oscilloscopes is they are capable of evaluating both analog and digital signals. Another enticing feature of digital storage oscilloscopes is that the captured data can be adequately used for future analysis Inspite of having these additional features , these digital storage oscilloscopes are a bit more expensive .In today’s educational institutions ,especially in electronics field oscilloscope play an important role for analyzing the circuit performance. Due to the expensive costs of oscilloscope these are less accessible to the undergraduate novice students.

Moreover, digital storage oscilloscope cannot be carried out easily from one place to another as it was bulky in nature. To overcome all these disadvantages of digital storage oscilloscope , a portable PC based multichannel oscilloscope was developed. The main theme of this system is interfacing personal computer(PC) with a Aurdino board to develop a portable oscilloscope. This oscilloscope can be carried out with much ease and was affordable to the undergraduate students to evaluate the performance their circuits.

3. INTRODUCTION TO ARDUINO

What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.

Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

3.1 INTRODUCTION TO ARDUINO UNO

- **Arduino Uno** is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.
- First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.
- The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.
- It allows the designers to control and sense the external electronic devices in the real world.
- This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. IDE is equally compatible with Windows, MAC or Linux Systems, however, Windows is preferable to use. Programming languages like C and C++ are used in IDE.
- Apart from USB, battery or AC to DC adopter can also be used to power the board.
- Arduino Uno boards are quite similar to other boards in Arduino family in terms of use and functionality, however, Uno boards don’t come with FTDI USB to Serial driver chip.
- There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno are the most official versions that come with Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.
- When nature and functionality of the task go complex, Mirco SD card can be added in the boards to make them store more information.

Features of Arduino Uno:

- Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer.

- Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

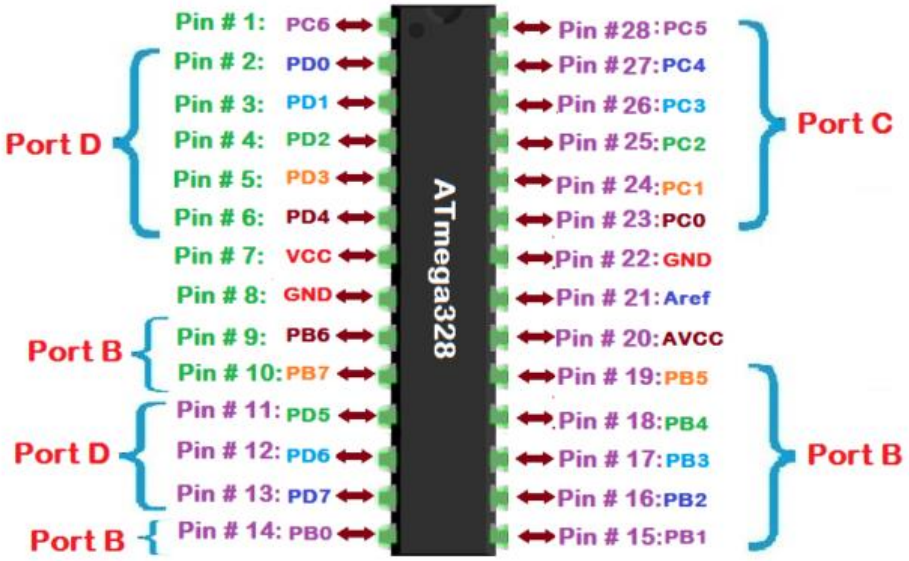


Fig 3.1 (a) Pin Diagram of ATmega328

- It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
- This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
- Reset pin is added in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
- There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.

- ❖ The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analogReference() function and AREF pin.
- ❖ 13KB of flash memory is used to store the number of instructions in the form of code.
- ❖ Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

Arduino Uno Pinout:

Arduino Uno is based on AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. Following figure shows the pinout of the Arduino Uno Board.

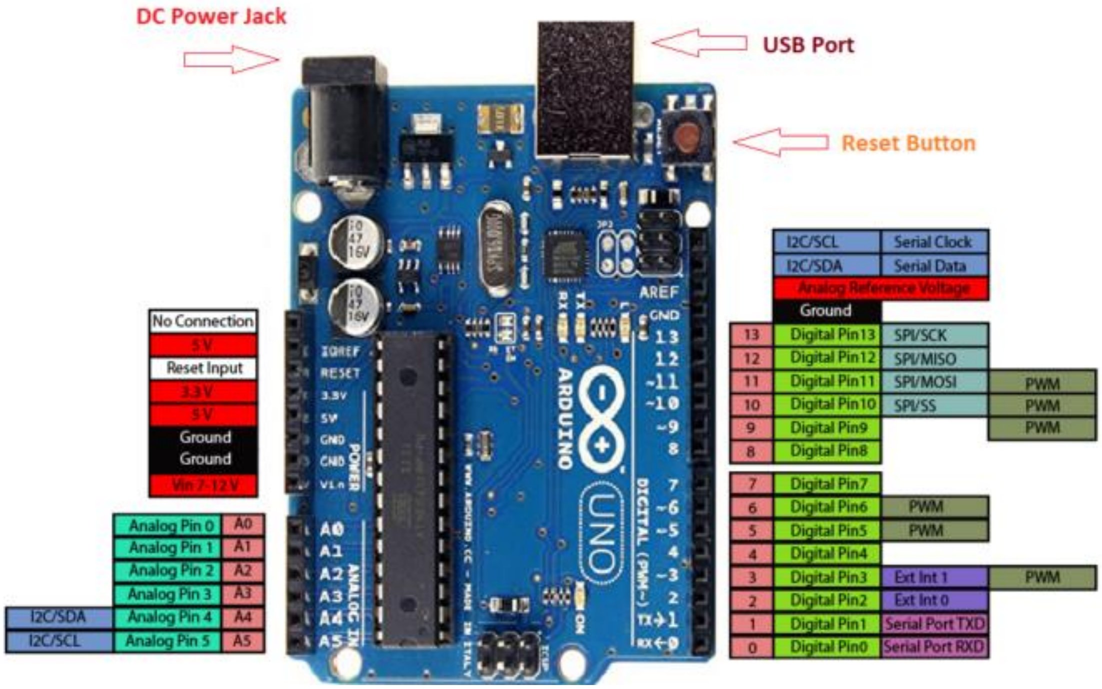


Fig 3.1 (b) Pinout of Arduino Uno

Pin Description:

There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resisters useless and damages the device.

LED: Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

Vin: It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

5V: This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack.

USB supports voltage around 5V while Vin and Power Jack support a voltage ranges between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins, they result in bypassing the voltage regulation that can damage the board if voltage surpasses from its limit.

GND: These are ground pins. More than one ground pins are provided on the board which can be used as per requirement.

Reset: This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

IOREF: This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.

PWM: PWM is provided by 3,5,6,9,10, 11pins. These pins are configured to provided 8-bit output PWM.

SPI: It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.

AREF: It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

TWI: It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

Serial Communication: Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx pin is used to receive data while Tx pin is used to transmit data.

External Interrupts: Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

Communication and Programming:

Arduino Uno comes with an ability of interfacing with other other Arduino boards, microcontrollers and computer. The Atmega328 placed on the board provides serial communication using pins like Rx and Tx.

The Atmega16U2 incorporated on the board provides a pathway for serial communication using USB com drivers. Serial monitor is provided on the IDE software which is used to send or receive text data from the board. If LEDs placed on the Rx and Tx pins will flash, they indicate the transmission of data.

Arduino Uno is programmed using Arduino Software which is a cross-platform application called IDE written in Java. The AVR microcontroller Atmega328 laid out on the base comes with builtin bootloader that sets you free from using a separate burner to upload the program on the board.

4. RESISTORS

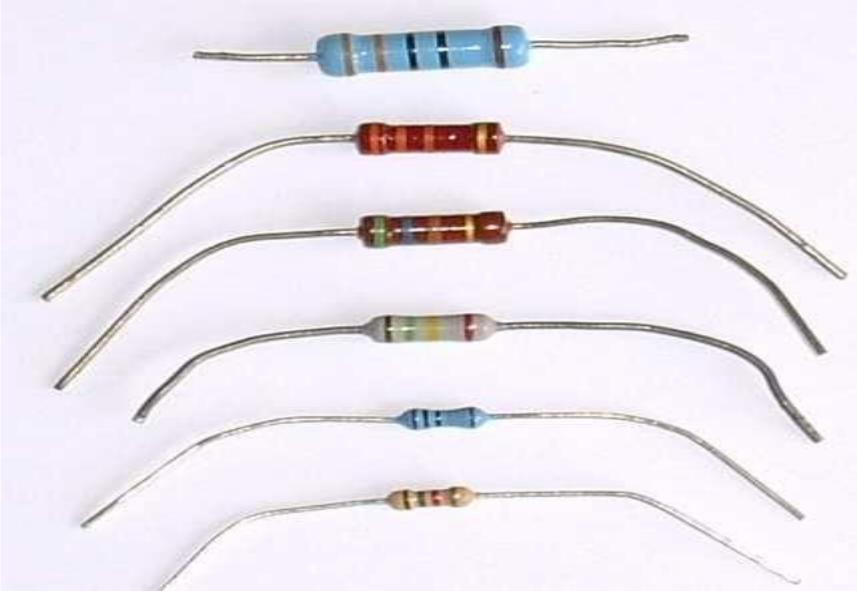
A resistor is a two-terminal electronic component designed to oppose an electric current by producing a voltage drop between its terminals in proportion to the current, that is, in accordance with Ohm's law:

$$V = IR$$

Resistors are used as part of electrical networks and electronic circuits. They are extremely commonplace in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel/chrome).

The primary characteristics of resistors are their resistance and the power they can dissipate. Other characteristics include temperature coefficient, noise, and inductance. Less well-known is critical resistance, the value below which power dissipation limits the maximum permitted current flow, and above which the limit is applied voltage. Critical resistance depends upon the materials constituting the resistor as well as its physical dimensions; it's determined by design.

Resistors can be integrated into hybrid and printed circuits, as well as integrated circuits. Size, and position of leads (or terminals) are relevant to equipment designers; resistors must be physically large enough not to overheat when dissipating their power.



A resistor is a two-terminalpassiveelectronic component which implements electrical resistance as a circuit element. When a voltage V is applied across the terminals of a resistor, a current I will flow through the resistor in direct proportion to that voltage. The reciprocal of the constant of proportionality is known as the resistance R, since, with a given voltage V, a larger value of R further "resists" the flow of current I as given by Ohm's law:

$$I = \frac{V}{R}$$

Resistors are common elements of electrical networks and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel-chrome). Resistors are also implemented within integrated circuits, particularly analog devices, and can also be integrated into hybrid and printed circuits.

The electrical functionality of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than 9 orders of magnitude. When specifying that resistance in an electronic design, the required precision of the resistance may require attention to the manufacturing tolerance of the chosen resistor, according to its specific application. The temperature coefficient of the resistance may also be of concern in some precision applications. Practical resistors are also specified as having a maximum power rating which must exceed the anticipated power dissipation of that resistor in a particular circuit: this is mainly of concern in power electronics applications. Resistors with higher power ratings are physically larger and may require heat sinking. In a high voltage circuit, attention must sometimes be paid to the rated maximum working voltage of the resistor.

The series inductance of a practical resistor causes its behavior to depart from ohms law; this specification can be important in some high-frequency applications for smaller values of resistance. In a low-noise amplifier or pre-amp the noise characteristics of a resistor may be an issue. The unwanted inductance, excess noise, and temperature coefficient are mainly dependent on the technology used in manufacturing the resistor. They are not normally specified individually for a particular family of resistors manufactured using a particular technology. A family of discrete resistors is also characterized according to its form factor, that is, the size of the device and position of its leads (or terminals) which is relevant in the practical manufacturing of circuits using them.

Units

The ohm (symbol: Ω) is the SI unit of electrical resistance, named after Georg Simon Ohm. An ohm is equivalent to a volt per ampere. Since resistors are specified and manufactured over a very large range of values, the derived units of milliohm (1 mΩ = 10^{-3} Ω), kilohm (1 kΩ = 10^{3} Ω), and megohm (1 MΩ = 10^{6} Ω) are also in common usage.

The reciprocal of resistance R is called conductance G = 1/R and is measured in Siemens (SI unit), sometimes referred to as a mho. Thus a Siemens is the reciprocal of an ohm: $S = \Omega^{-1}$. Although the concept of conductance is often used in circuit analysis, practical resistors are always specified in terms of their resistance (ohms) rather than conductance.

Theory of operation

Ohm's law

The behaviour of an ideal resistor is dictated by the relationship specified in Ohm's law:

$$V = I \cdot R$$

Ohm's law states that the voltage (V) across a resistor is proportional to the current (I) passing through it, where the constant of proportionality is the resistance (R).

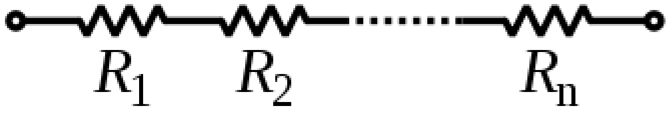
Equivalently, Ohm's law can be stated:

$$I = \frac{V}{R}$$

This formulation of Ohm's law states that, when a voltage (V) is present across a resistance (R), a current (I) will flow through the resistance. This is directly used in practical computations. For example, if a 300 ohm resistor is attached across the terminals of a 12 volt battery, then a current of 12 / 300 = 0.04 amperes (or 40 mill amperes) will flow through that resistor.

Series and parallel resistors

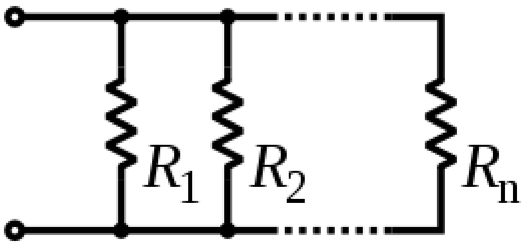
In a series configuration, the current through all of the resistors is the same, but the voltage across each resistor will be in proportion to its resistance. The potential difference (voltage) seen across the network is the sum of those voltages, thus the total resistance can be found as the sum of those resistances:



$$R_{\mathrm{eq}} = R_1 + R_2 + \cdots + R_n$$

As a special case, the resistance of N resistors connected in series, each of the same resistance R, is given by NR.

Resistors in a parallel configuration are each subject to the same potential difference (voltage), however the currents through them add. The conductance of the resistors then add to determine the conductance of the network. Thus the equivalent resistance (*R_{eq}*) of the network can be computed:



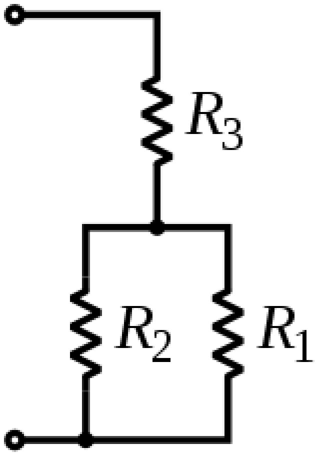
$$\frac{1}{R_{\mathrm{eq}}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}$$

The parallel equivalent resistance can be represented in equations by two vertical lines "||" (as in geometry) as a simplified notation. For the case of two resistors in parallel, this can be calculated using:

$$R_{\mathrm{eq}} = R_1 \| R_2 = \frac{R_1 R_2}{R_1 + R_2}$$

As a special case, the resistance of N resistors connected in parallel, each of the same resistance R, is given by R/N.

A resistor network that is a combination of parallel and series connections can be broken up into smaller parts that are either one or the other. For instance,



$$R_{\mathrm{eq}} = (R_1 \| R_2) + R_3 = \frac{R_1 R_2}{R_1 + R_2} + R_3$$

However, some complex networks of resistors cannot be resolved in this manner, requiring more sophisticated circuit analysis. For instance, consider a cube, each edge of which has been replaced by a resistor. What then is the resistance that would be measured between two opposite vertices? In the case of 12 equivalent resistors, it can be shown that the corner-to-corner resistance is $^5\!/_6$ of the individual resistance. More generally, the Y-Δ transform, or matrix methods can be used to solve such a problem. One practical application of these relationships is that a non-standard value of resistance can generally be synthesized by connecting a number of standard values in series and/or parallel. This can also be used to obtain a resistance with a higher power rating than that of the individual resistors used. In the special case of N identical resistors all connected in series or all connected in parallel, the power rating of the individual resistors is thereby multiplied by N.

5. PC BASED VIRTUAL IMPLEMENTATION

The instrumentation industry is heading steadily and rapidly towards the direction of virtual instrumentation. Virtual instruments are centered on a PC, embedded with specialized hardware to link it to the devices it must measure/control. This hardware typically includes plug-in boards for digitizing a signal directly or for controlling stand-alone instruments. Virtual instrumentation is known for its flexibility, interchangeability and low cost. Celma et al. conferred an idea behind the development of a PC- based spectrum analyzer suitable for use in undergraduate laboratories. They identified that the requirements of such a data acquisition system are not very high in view of the limited range of signals encountered in undergraduate laboratories. Chickamenahalli et al. developed an undergraduate research project that involved the interface of a HP digital oscilloscope to an IBM PC using National Instruments' General Purpose Interface Board. Smith etal. in their papers explained about setting up of a simple DSO integrated with a printer/plotter system for quick reproduction of the signal.

However, they were suspicious about the bench-space requirements for such a scheme involving the DSO, printer/plotter ,and the PC. This discovery led to the concept of integrated DSO-PC-based aurdino system described in the present project, one sufficient and affordable for most undergraduate-level laboratories

The present project mainly contributes in developing a fully functional, PC-based oscilloscope with associated modules of data acquisition hardware, software, interfacing, and graphical user interface (GUI). Processing 3 offers a versatile tool for developing a PC-based multichannel oscilloscope. This is an open source IDE which can be easily downloaded and explored with attractive designs,Hence this can be easily afforded by undergraduate students. With the help of this IDE, the proposed system is intended to provide a low-cost, simple, yet effective solution for integrated multichannel data acquisition, display, analysis, and printing operations in an undergraduate laboratory. It make use of the conventional PC parallel port, interfacing circuitry and commonly used GUI development tool and other programming languages, such as Java and Turbo C. The system is excellently used in undergraduate laboratories for real-time applications and educational/ instructional purposes as well.

6. PROJECT DESCRIPTION

In this project PC and an Arduino board as the hardware for signal acquisition. We can use this oscilloscope to capture frequency signals up to 5kHz. The Arduino board, the heart of the oscilloscope, reads the values from its inbuilt analogue-to-digital converter (ADC) and pushes these to the PC via USB port. This application works as the front-end to plot input signals as waveforms on your computer screen.

The Arduino board consists of Atmel’s AVR microcontroller, which can be 8-, 16- or 32-bit based on the type of the board. For this project, we can use any variant of the Arduino as hardware. The AVR microcontroller has an inbuilt ADC. In the project, we use pin A0 to capture the input signal. The captured input signal is fed to UART via UART-USB converter in the Arduino to the PC. A virtual COM port is created by Windows whenever the Arduino connects to the PC. A Windows-based application developed using NI LabWindows opens up the virtual COM port and starts plotting signals visually using Graph libraries.

The sampling speed of the oscilloscope is limited by the baud rate of the UART. The Arduino sketch is coded to read the ADC using ISR, and the UART baud rate is configured at 115200, which sends data at 85µs intervals. This gives an effective sampling rate of 12kSa/s.

Circuit and working of Project

The PC scope set-up is quite simple and straightforward as shown in Fig (6). The Arduino board connects to laptop or PC via the USB cable. Any external power supply for the board is not required as the board is powered by the USB only. Connect switching diodes (D1 and D2) as input protection circuit to pin A0 of the Arduino’s ADC. We need to develop Arduino sketch and PC software in order to use this circuit. We Install the developed PC Scope in Windows PC and open the application. Next, open the developed Arduino sketch from Arduino IDE and compile the sketch. Connect the Arduino board to the PC and flash the sketch into the microcontroller on the Arduino board.

The ADC of Arduino can measure voltages up to 5V. So it is advisable to add a small protection circuit to limit the input voltage to 5V and clamp the negative voltage. A low-power, fast-switching diode like 1N4148 can be used to protect the input pin. Connect a 10-kilo-ohm resistor in series with the input. It will work as a current limiter in case the input goes beyond 5V. Additional voltage dividers can be used in case you need to measure voltages higher than 5V.

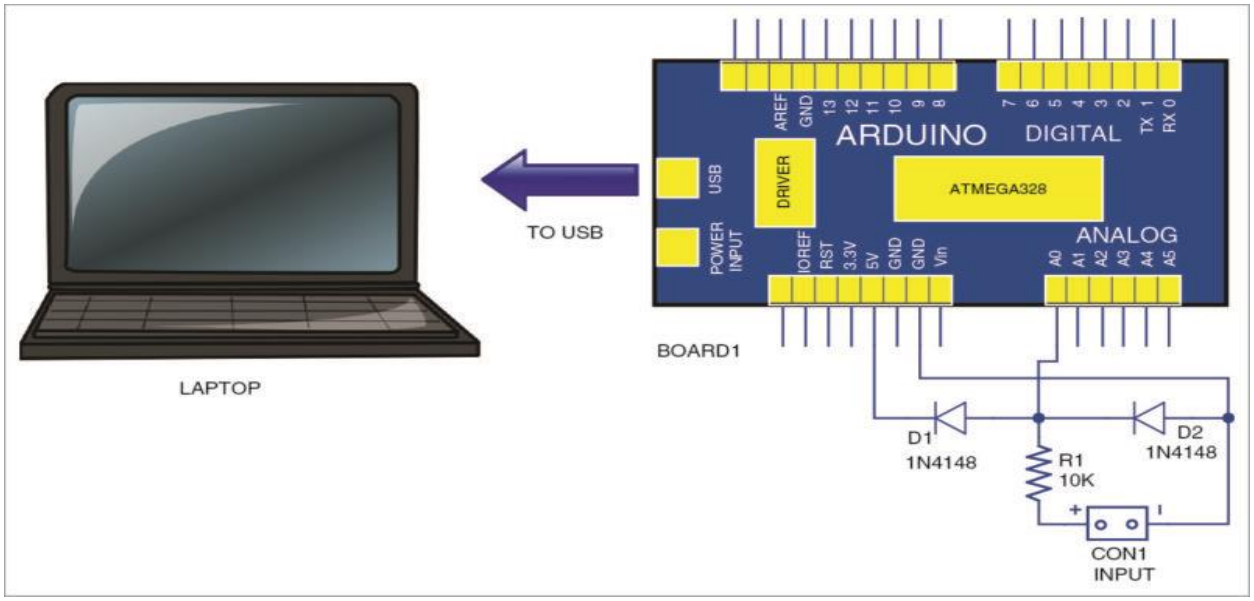


Fig (6) Circuit of the PC-based oscilloscope using Arduino

7. ARDUINO CODE

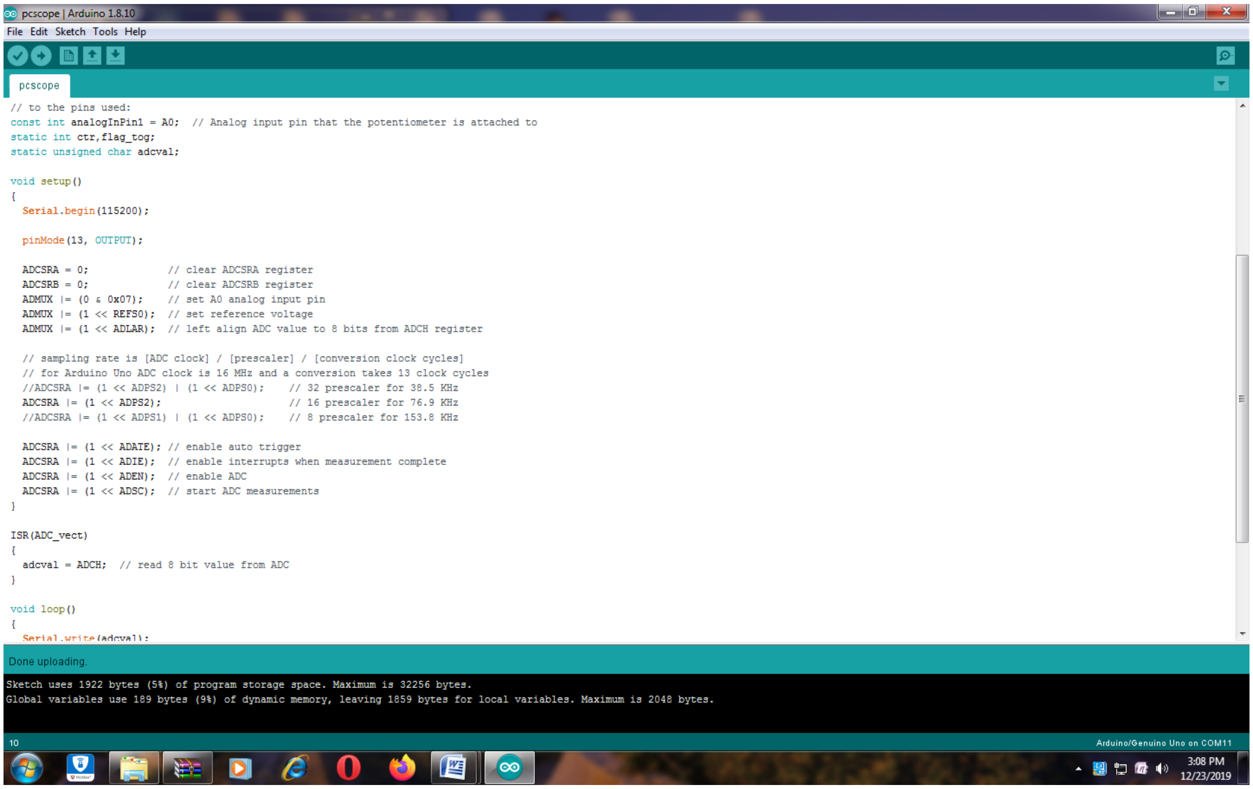
```
// These constants won't change.  They're used to give names
// to the pins used:
const int analogInPin1 = A0;  // Analog input pin that the potentiometer
is attached to
static int ctr,flag_tog;
static unsigned char adcval;

void setup()
{
  Serial.begin(115200);

  pinMode(13, OUTPUT);

  ADCSRA = 0;             // clear ADCSRA register
  ADCSRB = 0;             // clear ADCSRB register
  ADMUX |= (0 & 0x07);    // set A0 analog input pin
  ADMUX |= (1 << REFS0);  // set reference voltage
  ADMUX |= (1 << ADLAR);  // left align ADC value to 8 bits from ADCH
register

  // sampling rate is [ADC clock] / [prescaler] / [conversion clock
cycles]
  // for Arduino Uno ADC clock is 16 MHz and a conversion takes 13 clock
cycles
  //ADCSRA |= (1 << ADPS2) | (1 << ADPS0);    // 32 prescaler for 38.5 KHz
  ADCSRA |= (1 << ADPS2);                     // 16 prescaler for 76.9 KHz
  //ADCSRA |= (1 << ADPS1) | (1 << ADPS0);    // 8 prescaler for 153.8 KHz

  ADCSRA |= (1 << ADATE); // enable auto trigger
  ADCSRA |= (1 << ADIE);  // enable interrupts when measurement complete
  ADCSRA |= (1 << ADEN);  // enable ADC
  ADCSRA |= (1 << ADSC);  // start ADC measurements
}

ISR(ADC_vect)
{
  adcval = ADCH;  // read 8 bit value from ADC
}

void loop()
{
  Serial.write(adcval);
  // Following code to generate ref signal at pin 13 @ 50HZ. You can 
connect A0 to see the waveform in PCScope.exe
  ctr++;
  if(ctr>117) //117=10.03ms
  {
   ctr=0;
   flag_tog = !flag_tog;
   digitalWrite(13, flag_tog);
  }
}
```

8. SOFTWARE & TESTING

We used Arduino IDE to program the Arduino Uno., install the Arduino IDE With the help of this software we can easily upload the program to the programmer with the help of a inbuilt bootloader. In this project for hardware interfacing we used a compiler named “Processing”. Processing is a programming language, development environment and online community.

The sampling rate of this PC scope application is limited by the rate at which the data is sent to the PC. Baud rate of 115000 gives time interval of around 85 µs. It is important to get the ADC signals much before this time to get reliable data plotting. The sketch reads pin A0 of Board1 and sends to UART at 115200 baud rate. At this speed, bytes of the input are pushed at time intervals of around 85µs.

By default, the ADC configuration of the Arduino gives samples every 116µs. So here the ADC is configured with additional lines of code to get samples faster than 85µs by setting the prescaler to 16. With this, you get ADC conversion every 20µs, which is much faster than the UART data transfer rate.

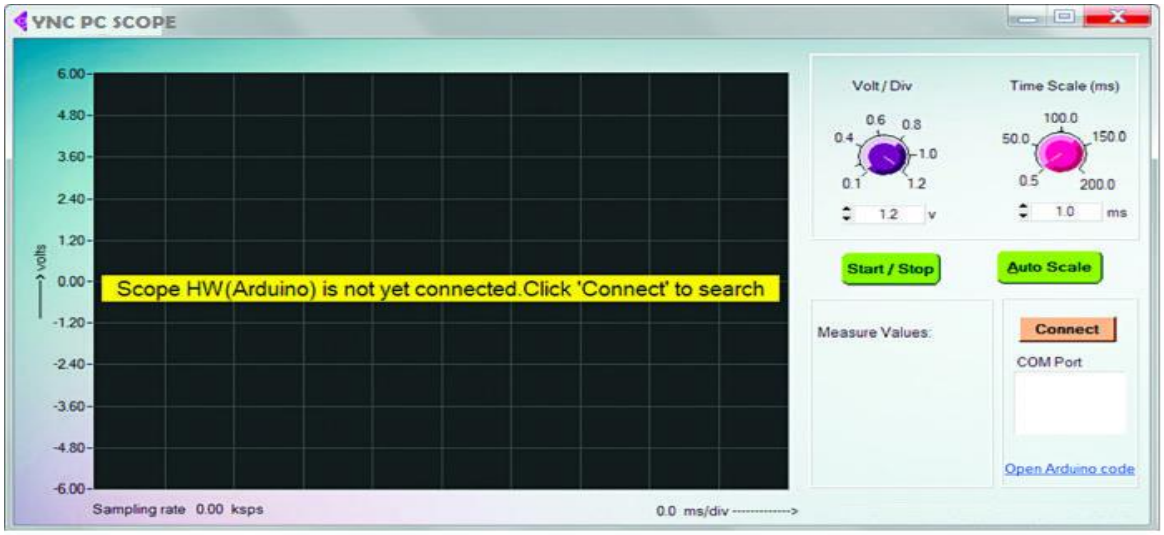


```
// to the pins used:
const int analogInPin1 = A0;  // Analog input pin that the potentiometer is attached to
static int ctr,flag_tog;
static unsigned char adcval;

void setup()
{
  Serial.begin(115200);

  pinMode(13, OUTPUT);

  ADCSRA = 0;             // clear ADCSRA register
  ADCSRB = 0;             // clear ADCSRB register
  ADMUX |= (0 & 0x07);    // set A0 analog input pin
  ADMUX |= (1 << REFS0);  // set reference voltage
  ADMUX |= (1 << ADLAR);  // left align ADC value to 8 bits from ADCH register

  // sampling rate is [ADC clock] / [prescaler] / [conversion clock cycles]
  // for Arduino Uno ADC clock is 16 MHz and a conversion takes 13 clock cycles
  //ADCSRA |= (1 << ADPS2) | (1 << ADPS0);    // 32 prescaler for 38.5 KHz
  ADCSRA |= (1 << ADPS2);                     // 16 prescaler for 76.9 KHz
  //ADCSRA |= (1 << ADPS1) | (1 << ADPS0);    // 8 prescaler for 153.8 KHz

  ADCSRA |= (1 << ADATE); // enable auto trigger
  ADCSRA |= (1 << ADIE);  // enable interrupts when measurement complete
  ADCSRA |= (1 << ADEN);  // enable ADC
  ADCSRA |= (1 << ADSC);  // start ADC measurements
}

ISR(ADC_vect)
{
  adcval = ADCH;  // read 8 bit value from ADC
}

void loop()
{
  Serial.write(adcval);
```

```
Sketch uses 1922 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 189 bytes (9%) of dynamic memory, leaving 1859 bytes for local variables. Maximum is 2048 bytes.
```

Fig 8(a) PC Based Oscilloscope Arduino Code window

PC software. As stated earlier, the front-end PC software for signal acquisition and processing is developed using NI Lab Windows. The serial port data is captured through Arduino at regular time intervals and plotted as a graph on the screen using the Plot function library. The display points along X-axis are calculated based on the user-defined time scale. The Y-axis range is set using the voltage selection control.



Scope HW(Arduino) is not yet connected.Click 'Connect' to search

Volt / Div: 1.2 v

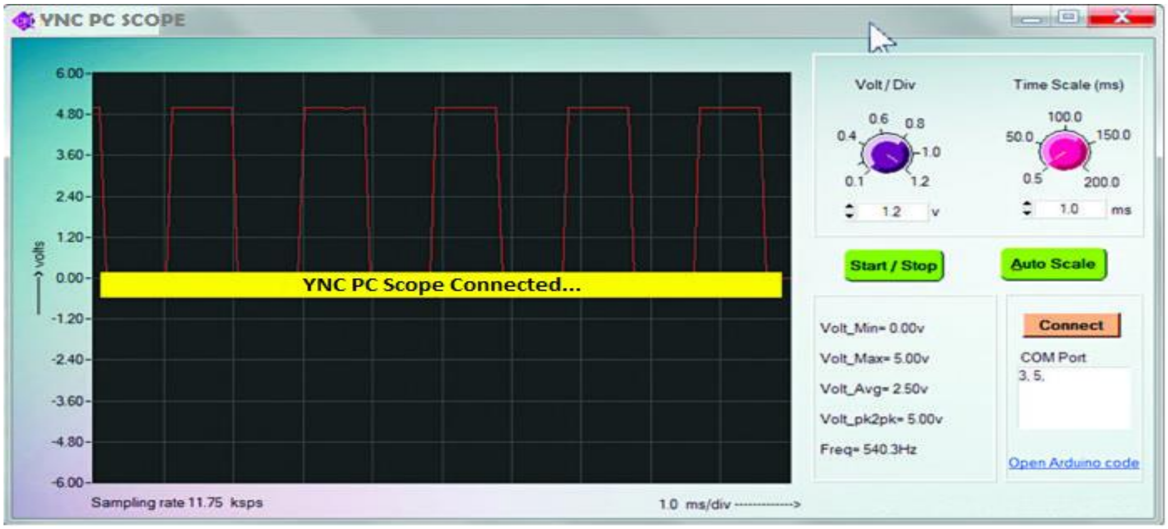
Time Scale (ms): 1.0 ms

Sampling rate 0.00 ksps

0.0 ms/div ------------>

Fig 8(b) Message on the screen when the PC-based Oscilloscope is run for the first time

After developing the PC scope application, click ‘Connect’ button on PC screen to connect to the Arduino board Fig 8(b). When the board gets connected to PC, we will get a confirmation message for three seconds as shown in Fig 8(c).



YNC PC Scope Connected...

Volt / Div: 1.2 v

Time Scale (ms): 1.0 ms

Volt_Min= 0.00v

Volt_Max= 5.00v

Volt_Avg= 2.50v

Volt_pk2pk= 5.00v

Freq= 540.3Hz

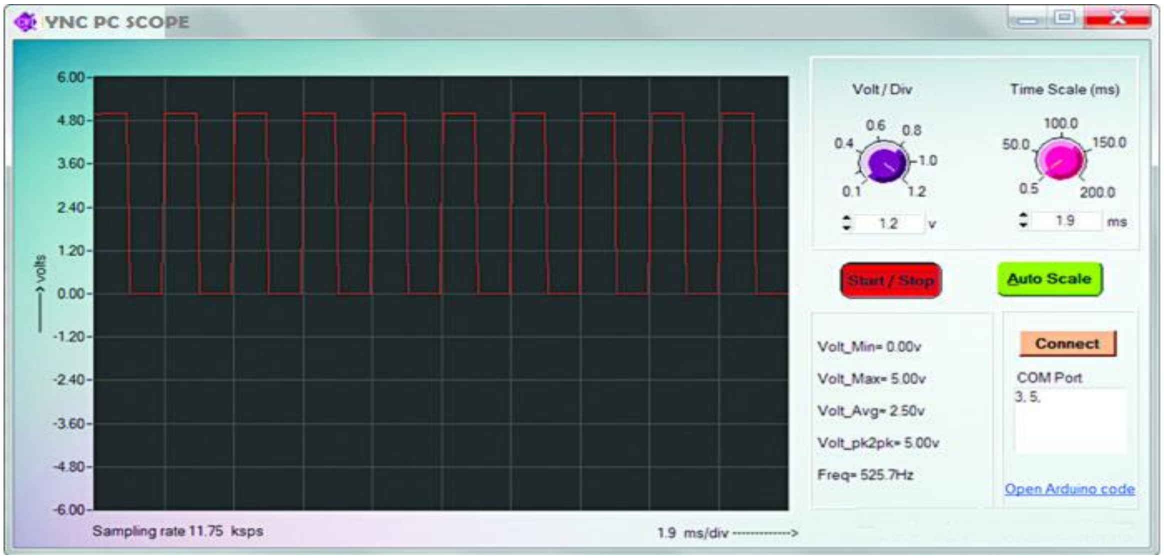
COM Port: 3, 5,

Sampling rate 11.75 ksps

1.0 ms/div ------------>

Fig 8(c) Message after the hardware successfully connects to the PC

Feed any squarewave input of up to 5kHz at CON1. The software must plot its output waveform on your PC. Square and triangular output waveforms of 525Hz and 530Hz captured on the screen during testing are shown in Figs 8(d) and 8(e), respectively. Similarly, we can feed rectangular or pulse inputs (but not sine waves) to get output waveforms.



Volt / Div: 1.2 v

Time Scale (ms): 1.9 ms

Volt_Min= 0.00v

Volt_Max= 5.00v

Volt_Avg= 2.50v

Volt_pk2pk= 5.00v

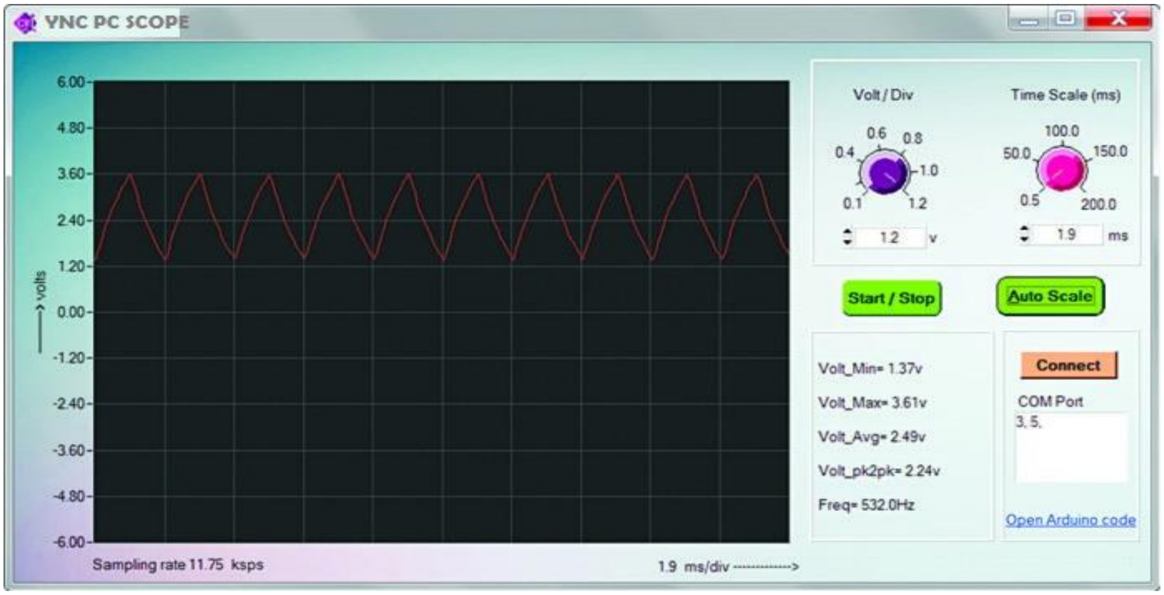
Freq= 525.7Hz

COM Port
3, 5,

Sampling rate 11.75 ksps

1.9 ms/div ------------>

Fig 8(d) Test signal of 525Hz square waveform captured on the screen



Volt / Div: 1.2 v

Time Scale (ms): 1.9 ms

Volt_Min= 1.37v

Volt_Max= 3.61v

Volt_Avg= 2.49v

Volt_pk2pk= 2.24v

Freq= 532.0Hz

COM Port
3, 5,

Sampling rate 11.75 ksps

1.9 ms/div ------------>

Fig 8(e) Test signal of 530Hz triangular waveform captured on the screen

9. RESULT

PC based oscilloscope was firstly simulated by using processing Debugger. A test signal generated from function generator is given as input signal to arduino which was then connected to PC through a USB connection, using the system hardware prototype.

The results are displayed in developed oscilloscope system is shown in Fig (9)

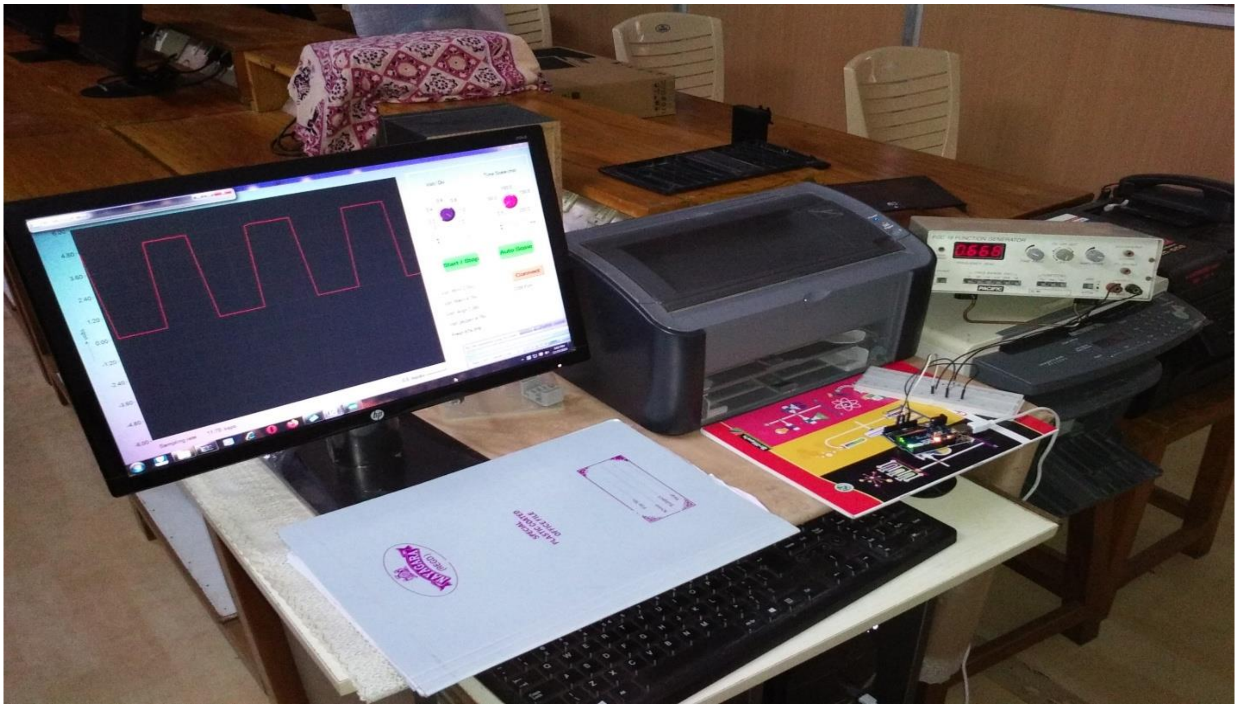


Fig (9) PC-Based Oscilloscope Project Prototype

10. CONCLUSION

A low cost, portable, plug-in type PC oscilloscope is implemented based on the communication features of the PC parallel port. Hereby it is concluded that Arduino can be used as an Oscilloscope for displaying and analyzing waveforms. Arduino Oscilloscope also has the added feature of performing various operations on the applied input signals such as addition, subtraction etc. This helps the undergraduate students to analyze in a simple way and overcomes the drawbacks of the conventional CRO's. the hardware interface circuit developed makes the functioning of a oscilloscope much easier with few affordable electronics components. so as for the conversion and processing of the analog signal into the digital form The Arduino software (IDE) used was easily available on the Arduino . The software developed was used to manage the acquisition, to represent the signals and post processing of the signal. The use of potential divider circuit protects the Arduino from over voltages or from getting damaged.

11. BIBLIOGRAPHY

REFERENCES:

1. Arduino Programming: The Ultimate Beginner's Guide to Learn Arduino Programming by Ryan Turner.
2. “The ATMEGA Microcontroller and Embedded systems” by Mazidi.
3. ATMEGA 328 Data Sheets.

WEBSITES:

- www.atmel.com
- www.arduino.cc
- www.wikipedia.org
- www.alldatasheets.com