  
**SHRI VILE PARLE KELAVANI MANDAL’S**

**SHRI BHAGUBHAI MAFATLAL POLYTECHNIC**

IRLA, N.R.G. MARG, VILE PARLE (WEST), MUMBAI-400056.

**DEPARTMENT OF INFORMATION TECHNOLOGY – VI SEMESTER**

**TERM – DECEMBER ’15 TO APRIL ‘16**

]

**A PROJECT REPORT ON**

**VISI LIGHT**

**SUBMITTED BY**

**[Group 1]**

Jaimish Ashar (1398001)

Manthan Mehta (1398018)

Bhavya Shah (1398033)

Marmik Thakkar (1398040)

**GUIDED BY**

Mr. Abhijit Dongaonkar

**TECHNICIAN**

Mrs.Kimaya Maikar

**HEAD OF DEPARTMENT**

Mrs. Neeta G. Kadukar

INDEX

VISI LIGHT

1. [OVERVIEW 1](#_Toc447910686)
   1. [INTRODUCTION 2](#_Toc447910687)
   2. [LITERATURE STUDY 3](#_Toc447910688)
   3. [ADVANTAGES 5](#_Toc447910689)
   4. [PROBLEM DEFINITION 6](#_Toc447910690)
2. [ANALYSIS 7](#_Toc447910691)
   1. [TIME MANAGEMENT 8](#_Toc447910692)
   2. [LIMITATIONS 9](#_Toc447910693)
3. [TECHNOLOGY USED 10](#_Toc447910694)
   1. [REQUIREMENTS 11](#_Toc447910695)
   2. [SOFTWARES 12](#_Toc447910696)
   3. [COMPONENTS 17](#_Toc447910697)
   4. [REQUIREMENT SHEET 33](#_Toc447910698)
4. [DESIGN 36](#_Toc447910699)
   1. [GENERAL DIAGRAMS 37](#_Toc447910700)
   2. [CIRCUIT DIAGRAMS 40](#_Toc447910701)
   3. [LOGIC DIAGRAMS 43](#_Toc447910702)
5. [SOURCE CODE 47](#_Toc447910703)
6. [SCREENSHOTS 82](#_Toc447910704)
7. [CONCLUSION 86](#_Toc447910705)
   1. [FUTURE SCOPE 87](#_Toc447910706)
8. [REFERENCES 88](#_Toc447910707)

# OVERVIEW

## INTRODUCTION

Visible Light Communication is the communication which uses the visible light (which is between 400 to 800 THz). This technology is the subset of Optical Wireless Communication Technologies. [1]

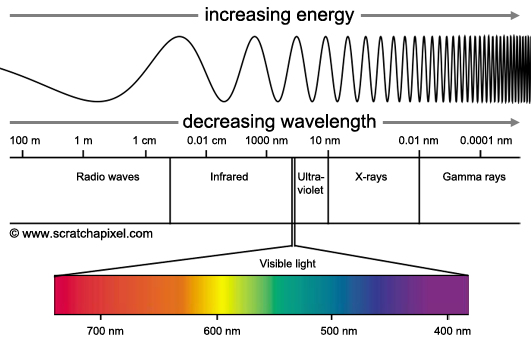
With the radio frequency spectrum becoming crowded, an alternative means to wireless communication is necessary to accommodate the exponentially increasing wireless traffic demand. Visible light communication systems provide an alternative to the current standards of wireless transfer of information, using light from LEDs as the communication medium. In these systems, light-emitting diodes blink at a rapid rate such that the human eye will not notice the change in light intensity, but a sensitive photodiode can detect the on-off behaviour and decode the information embedded within it. [1]

White LEDs (Light Emitting Diodes) in Visible Light Communication (VLC) is an emerging technology that is being researched so it can eventually be used for common communications systems. We proposed an optical indoor wireless communication system that used white LEDs like plug-in devices. We developed a practical implementation of VLC and demonstrated it experimentally. In particular, we focused on designing a prototype of VLC that can be used for transmission using LED and USB.

## LITERATURE STUDY

Visible light communications (VLC) can provide cable free communication at very high bit rates as high as 224Gbps. In addition, it has a major advantage that it causes no interference to RF-based devices. This made wireless communication possible in RF hazardous areas such as hospitals and space station. In addition to these two key advantages, safety, simple installation procedures and band licensing-free characteristic also helped to increase VLC’s potential to be developed as an alternative, or even a new standard to the wireless communication scheme. VLC uses white Light Emitting Diodes (LED), which send data by flashing light at speed. VLC uses white Light Emitting Diodes (LED), which send data by flashing light at speeds undetectable to the human eye. [1]

The idea of using visible light for data transmission is not entirely new. Using smoke signals to transfer messages goes back several thousand years and was used by many different cultures. The idea of using light as a communication medium was implemented by Alexander Graham Bell in 1880 with his invention of the Photo phone, a device that transmitted a voice signal on a beam of light. Bell focused sunlight with a mirror and then talked into a mechanism that vibrated the mirror.



## ADVANTAGES

The communication done through this concept transmits data through sockets of existing light, no additional medium is required because we can make use of the visible light spectrum frequencies easily.

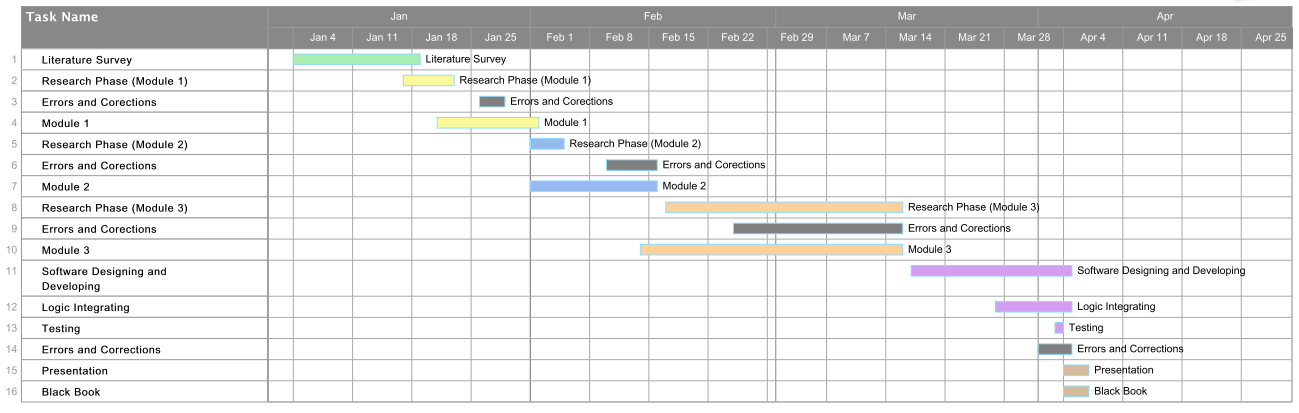
* This technology alleviates the most common and important problems which we face in communication through radio frequencies which are being used in today’s world.
* VLC is very secure and alleviates the problems pertaining to intruders or sniffers because it’s visible light, meaning only data will be transmitted only in near field thereby needing the malicious person to also be within that range. Eg: - VLC cannot travel through walls so even if you in living room a person sitting in another room cannot sniff data transmitted in the living room.
* Most importantly this technology is harmless to human body, it’s simply light and cannot harm anyone in any possible way when compared to RF which is being used in today’s world.
* The new transmission technology is suitable for hospitals, for example: because radio transmissions are not allowed there. Despite this fact, high data rates must be transmitted without losses and unzipped, according to the experts. If part of the communication occurs via the light in the surgical room, this would make it possible to control wireless surgical robots or transmit x-ray images

## PROBLEM DEFINITION

* First when the user opens the software for transmission of data he will see a basic GUI with buttons stating all the basic functions such as sending text, sending files and configuring the settings.
* After going through the GUI the user can click on any of the buttons to perform the particular task.
* If the user selects the sending text button, then he will be able to connect to the COM Port and send text through the circuit with the help of light.
* There will be 2 option in this
  + Transmission of text
  + Transmission of Images
* The user who wants to implement this circuit will have to first select anyone of the options from all the options listed above listed above.
* After selecting anyone option the user will be directed to the interface for transmission and connected to the COM Port
* After feeding in the suitable input the user can click the send button for transmission.
* Then the transceiver circuits will activate and send the data through LED which will be detected on the other side by a Photodetector
* If the transmission is successfully the data will appear on the system connected to the receiving circuit in the particular case.

# ANALYSIS

## TIME MANAGEMENT



## LIMITATIONS

* With just one LED, the range of communication is limited. The range can be increased by using a LED panel, or multiple LED panels. In module 2, 1 LED panel will support a maximum distance of 30cm. By cascading 4 panels, a distance of 120cm can be achieved.
* Communication is not over any network protocol. If implemented over TCP/IP protocols, communication can be done at higher ranges, and can pose as a real-world solution and can give speed up to 224Gbps which was achieved by Massachusetts Institute of Technology in their
* The intensity of the light has to always be high. If the intensity drops, then the connection speeds will drop, thereby defeating the purpose of the technology itself.
* As the system needs some source of light at all times, it consumes a lot of power.
* You cannot use VLC while the lights are off, unless you use only IR, which restricts bandwidth.
* Li-Fi has no standard yet, so equipment from one vendor won't work with another.
* VLC doesn't pass through walls and if you cover up the receiver it can't work.

# TECHNOLOGY USED

## REQUIREMENTS

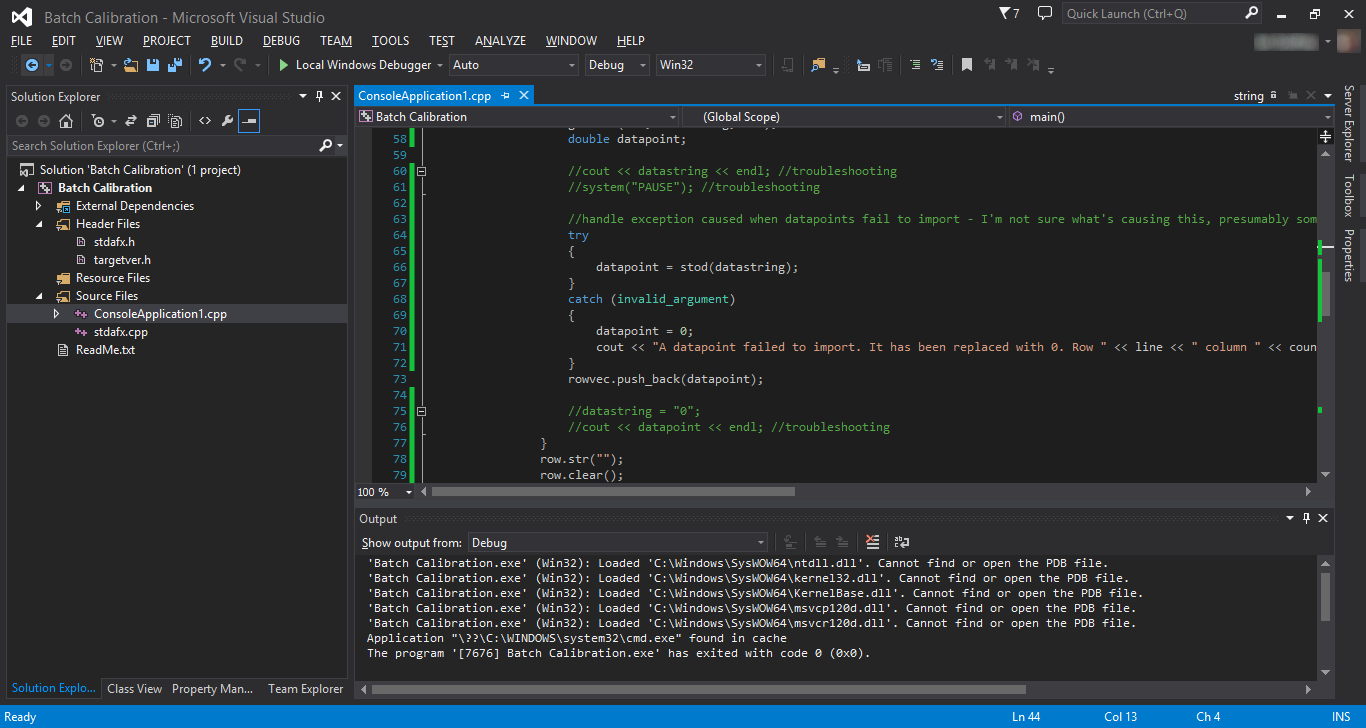
Software

* Visual Studio 2013 Community Edition. (Free)
* Hyper Terminal (Free Trial).
* Arduino IDE
* Circuit Diagram
* ExpressPCB
* PhototShop CS6
* MS Office

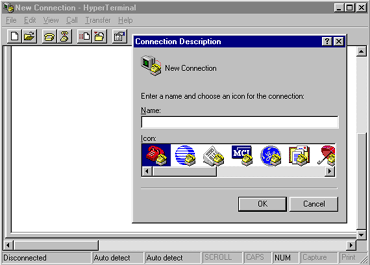
Hardware

* Using Pre-Existing Module
* Need to program the ATMega328 micro controller which is used in the Arduino.
* Using reference from the Internet
* Battery 9V
* Photo Transistor
* IC 7414 Inverter Buffer
* Resistor , LED and Capacitors
* Transistor BC-548
* IC Max 232
* DB9 Connector for PC communication

## SOFTWARES

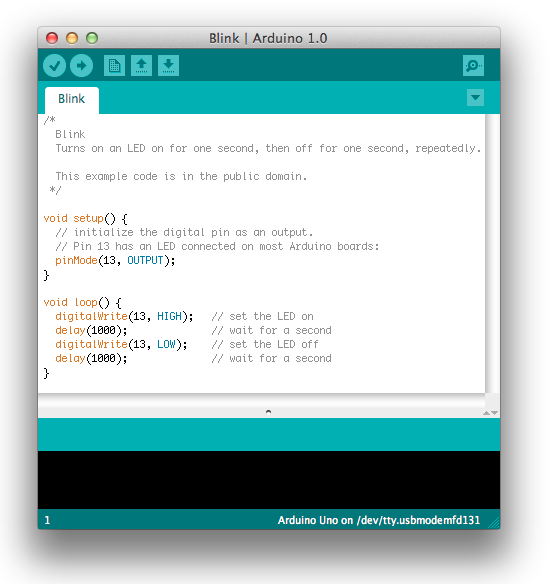
**Visual Studio 2013 Community Edition:**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. [4]

**Hyper Terminal:**

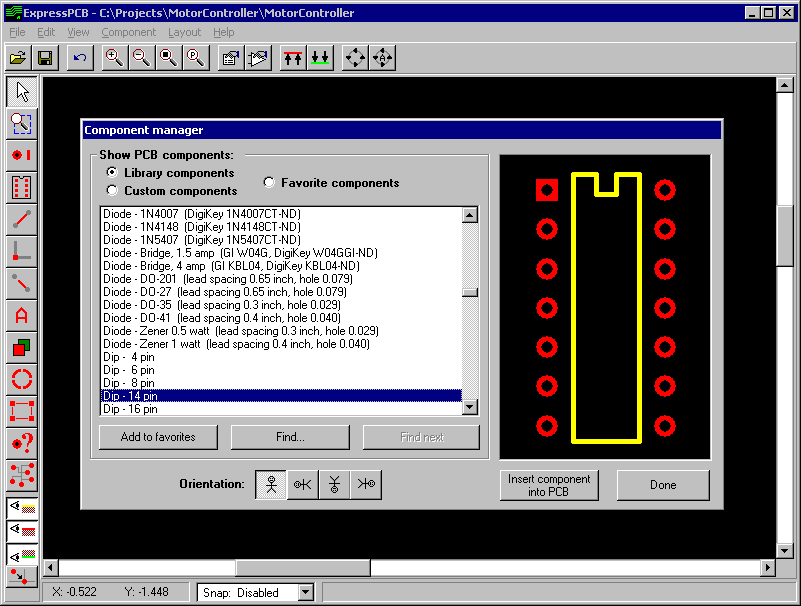
HyperTerminal Private Edition is an award winning terminal emulation program capable of connecting to systems through TCP/IP Networks, Dial-Up Modems, and COM ports. If you need HyperTerminal for Windows 7, 8, 10, or Vista. [5]

**Arduino IDE:**



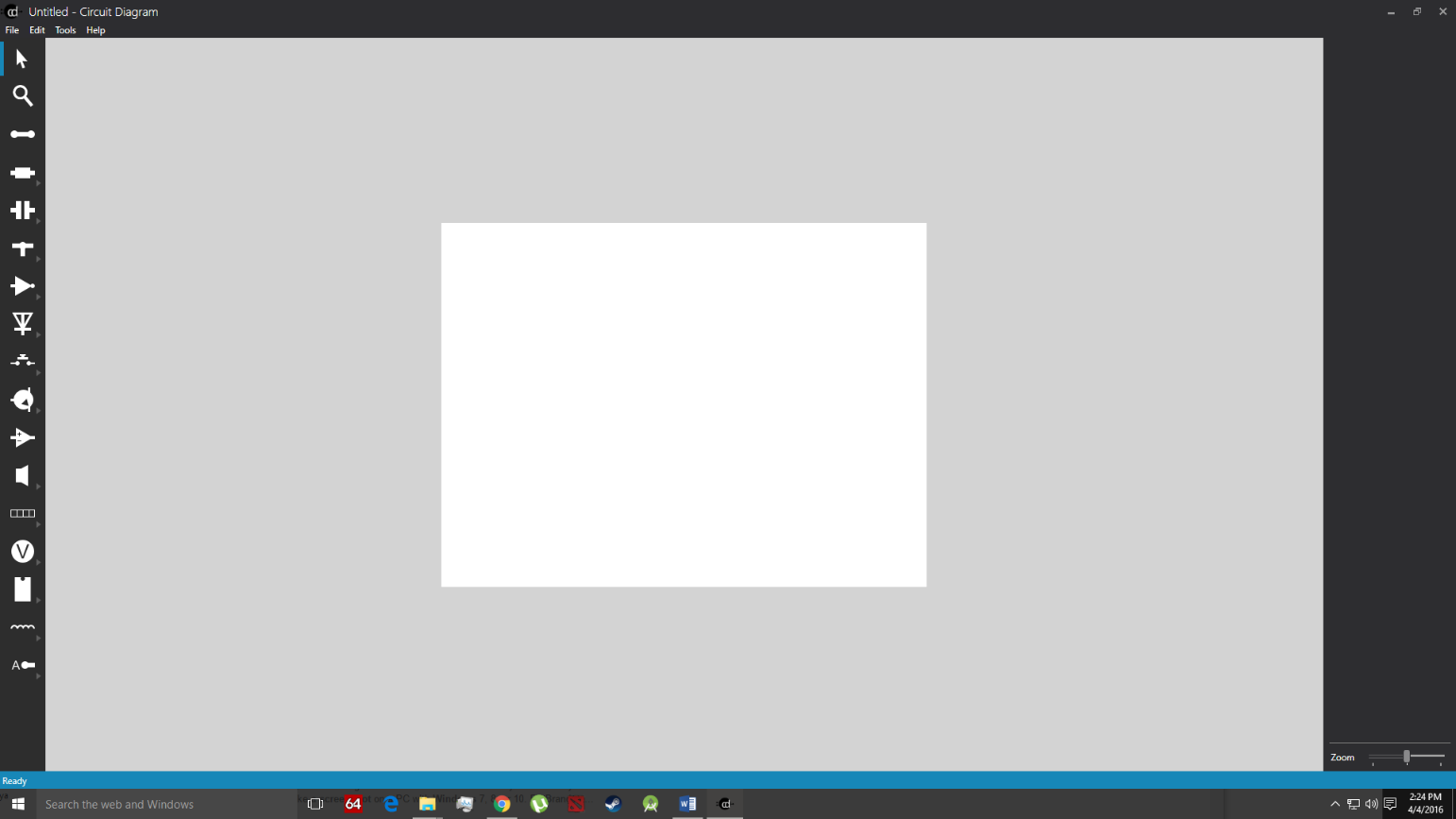
The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

**Express PCB:**



Designing 2 or 4 layer boards using the ExpressPCB  program is very simple. Start by inserting the component footprints, then drag them into position. Next, connect the pins by drawing the traces.

If you link your schematic file to the PCB, the ExpressPCB program will highlight the pins that should be wired together in blue.

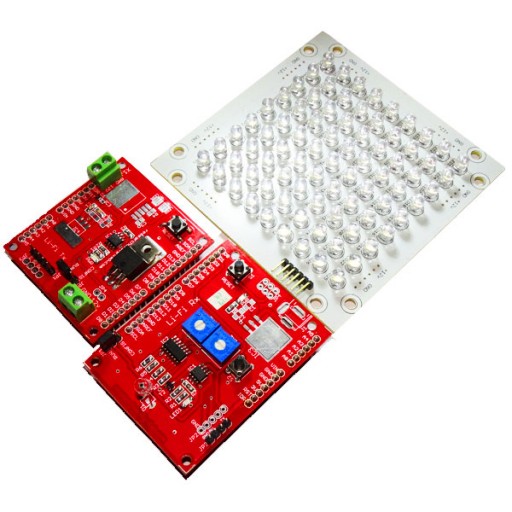
**Circuit Diagram**:

A **circuit diagram** (**electrical diagram**, **elementary diagram**, **electronic schematic**) is a graphical representation of an [electrical circuit](https://en.wikipedia.org/wiki/Electrical_network). A [pictorial](https://en.wikipedia.org/wiki/Pictorial) circuit diagram uses simple images of components, while a [schematic diagram](https://en.wikipedia.org/wiki/Schematic_diagram) shows the components and interconnections of the circuit using standardized symbolic representations. The presentation of the interconnections between circuit components in the schematic diagram does not necessarily correspond to the physical arrangements in the finished device.[[1]](https://en.wikipedia.org/wiki/Circuit_diagram#cite_note-1)

Unlike a [block diagram](https://en.wikipedia.org/wiki/Block_diagram) or [layout diagram](https://en.wikipedia.org/wiki/Integrated_circuit_layout), a circuit diagram shows the actual [electrical connections](https://en.wikipedia.org/wiki/Electrical_connection). A drawing meant to depict the physical arrangement of the wires and the components they connect

Circuit diagrams are used for the design ([circuit design](https://en.wikipedia.org/wiki/Circuit_design)), construction (such as [PCB](https://en.wikipedia.org/wiki/Printed_circuit_board) layout), and maintenance of electrical and electronic equipment.

## COMPONENTS

**Pre-Existing Module:**

The TX Side of the module will transmit the data. It’s connected to the array of LED’s through which the data is transferred. The data will receive at the Rx Side of the module.

The RX Side of the module will be receiving the data form the TX Side of the module. This side can be directly connected to the Hyper Terminal software serially to the PC and can also be programmed to receive the data.

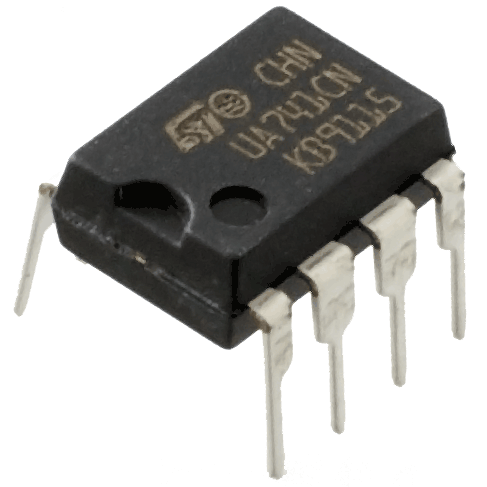
Transmits data serially at 38400 Baud rate, with a distance of 1-2 feet. Less energy required and is highly secured when compared to the Wi-Fi

**IC 7414: -**

IC7414 is a decision-making circuit. It is used to convert a slowly varying analogue signal voltage into one of two possible binary states, depending on whether the analogue voltage is above or below a pre-set threshold value. A comparator can do much the same job.

It’s also used as an Inverter also known as NOT gate is a fundamental block in Logic Design for Digital Circuits. Its purpose is to invert the signal. So, if input is Low (Logic 0), then the output will be High (Logic 1). If the input is High (Logic 1), then the output will be Low (Logic 0)

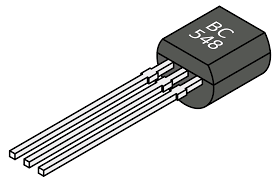
**IC 741:**



The 741 Operational Amplifier is probably the most versatile Integrated Circuit available. It is very cheap especially keeping in mind the fact that it contains several hundred components. The most common Op-Amp is the 741 and it is used in many circuits.

The OP AMP is a ‘Linear Amplifier’ with an amazing variety of uses. Its main purpose is to amplify (increase) a weak signal - a little like a Darlington Pair.

**Transistor BC 548: -**

****

BC548 is general purpose silicon, NPN, bipolar junction transistor. It is used for amplification and switching purposes. The maximum DC current gain is 800.

The transistor terminals require a fixed DC voltage to operate in the desired region of its characteristic curves. For amplification applications, the transistor is biased such that it is partly on for all input conditions. The input signal at base is amplified and taken at the emitter. BC548 is used in common emitter configuration for amplifiers. The voltage divider is the commonly used biasing mode. For switching applications, transistor is biased so that it remains fully on if there is a signal at its base. In the absence of base signal, it gets completely of

**IC Max 232:**

The MAX232 IC is used to convert the TTL/CMOS logic levels to RS232 logic levels during serial communication of microcontrollers with PC. The controller operates at TTL logic level (0-5V) whereas the serial communication in PC works on RS232 standards (-25 V to + 25V). This makes it difficult to establish a direct link between them to communicate with each other. [10]

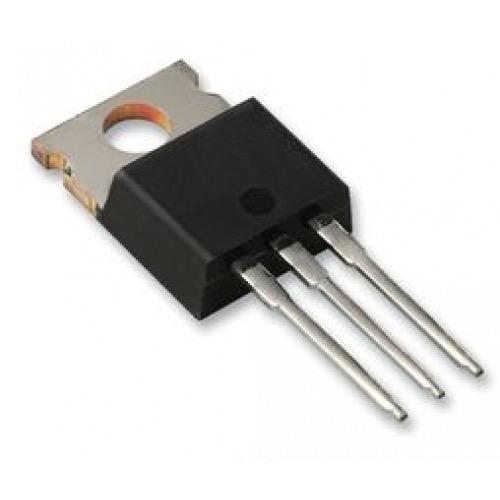
**Capacitor (.1uF):**

A Capacitor is a [passive](https://en.wikipedia.org/wiki/Passivity_(engineering)) [two-terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [electrical component](https://en.wikipedia.org/wiki/Electronic_component) used to store electrical [energy](https://en.wikipedia.org/wiki/Energy) temporarily in an [electric field](https://en.wikipedia.org/wiki/Electric_field). The forms of practical capacitors vary widely, but all contain at least two [electrical conductors](https://en.wikipedia.org/wiki/Electrical_conductor) (plates) separated by a [dielectric](https://en.wikipedia.org/wiki/Dielectric) (i.e. an [insulator](https://en.wikipedia.org/wiki/Insulator_(electricity)) that can store energy by becoming [polarized](https://en.wikipedia.org/wiki/Dipolar_polarization)). The conductors can be thin films, foils or sintered beads of metal or conductive electrolyte, etc. The non-conducting dielectric acts to increase the capacitor's charge capacity. Materials commonly used as dielectrics include [glass](https://en.wikipedia.org/wiki/Glass), [ceramic](https://en.wikipedia.org/wiki/Ceramic), [plastic film](https://en.wikipedia.org/wiki/Plastic_film), [air](https://en.wikipedia.org/wiki/Air), [vacuum](https://en.wikipedia.org/wiki/Vacuum), [paper](https://en.wikipedia.org/wiki/Paper), [mica](https://en.wikipedia.org/wiki/Mica), and [oxide layers](https://en.wikipedia.org/wiki/Oxide). Capacitors are widely used as parts of [electrical circuits](https://en.wikipedia.org/wiki/Electrical_circuit) in many common electrical devices.

**Resistor:**

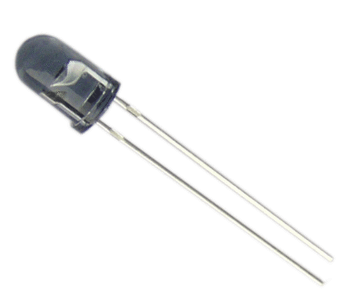
A resistor is a [passive](https://en.wikipedia.org/wiki/Passivity_(engineering)) [two-terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [electrical component](https://en.wikipedia.org/wiki/Electronic_component) that implements [electrical resistance](https://en.wikipedia.org/wiki/Electrical_resistance) as a circuit element. Resistors act to reduce current flow, and, at the same time, act to lower voltage levels within circuits. In electronic circuits, resistors are used to limit current flow, to adjust signal levels, [bias](https://en.wikipedia.org/wiki/Biasing) active elements, and terminate [transmission lines](https://en.wikipedia.org/wiki/Transmission_line) among other uses. High-power resistors, that can dissipate many [watts](https://en.wikipedia.org/wiki/Watt) of electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for [generators](https://en.wikipedia.org/wiki/Electric_generator). Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

Resistors are common elements of [electrical networks](https://en.wikipedia.org/wiki/Electrical_network) and [electronic circuits](https://en.wikipedia.org/wiki/Electronic_circuit) and are ubiquitous in [electronic equipment](https://en.wikipedia.org/wiki/Electronics). Practical resistors as discrete components can be composed of various compounds and forms. Resistors are also implemented within [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuits).

**Voltage Regulator:**

A voltage regulator is designed to automatically maintain a [constant voltage](https://en.wikipedia.org/wiki/Voltage_source) level. A voltage regulator may be a simple "[feed-forward](https://en.wikipedia.org/wiki/Feed_forward_(control))" design or may include feedback control. It may use an electromechanical [mechanism](https://en.wikipedia.org/wiki/Mechanism_(technology)), or electronic components. Depending on the design, it may be used to regulate one or more [AC](https://en.wikipedia.org/wiki/Alternating_current) or [DC](https://en.wikipedia.org/wiki/Direct_current) voltages.

Electronic voltage regulators are found in devices such as computer [power supplies](https://en.wikipedia.org/wiki/Power_supply) where they stabilise the DC voltages used by the processor and other elements. In automobile [alternators](https://en.wikipedia.org/wiki/Alternator) and central [power station](https://en.wikipedia.org/wiki/Power_station) generator plants, voltage regulators control the output of the plant. In an [electric power distribution](https://en.wikipedia.org/wiki/Electric_power_distribution) system, voltage regulators may be installed at a substation or along distribution lines so that all customers receive steady voltage independent of how much power is drawn from the line.

**Photo Diode:**

A photodiode is a semiconductor device that converts [light](https://en.wikipedia.org/wiki/Light) into [current](https://en.wikipedia.org/wiki/Electric_current). The current is generated when photons are absorbed in the photodiode. A small amount of current is also produced when no light is present. Photodiodes may contain [optical filters](https://en.wikipedia.org/wiki/Optical_filter), built-in lenses, and may have large or small surface areas. Photodiodes usually have a slower response time as their surface area increases. The common, traditional [solar cell](https://en.wikipedia.org/wiki/Solar_cell) used to generate electric [solar power](https://en.wikipedia.org/wiki/Solar_power) is a large area photodiode.

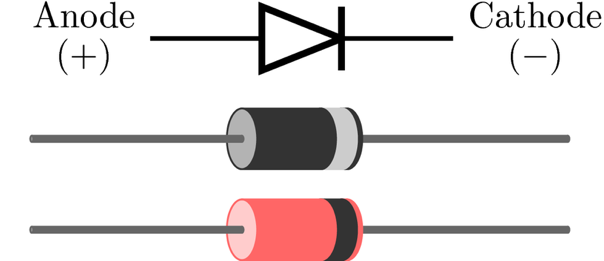
Photodiodes are similar to regular semiconductor diodes except that they may be either exposed (to detect [vacuum UV](https://en.wikipedia.org/wiki/Vacuum_UV) or [X-rays](https://en.wikipedia.org/wiki/X-rays)) or packaged with a window or [optical fibre](https://en.wikipedia.org/wiki/Optical_fiber) connection to allow light to reach the sensitive part of the device. Many diodes designed for use specifically as a photodiode use a [PIN junction](https://en.wikipedia.org/wiki/PIN_diode) rather than a [p–n junction](https://en.wikipedia.org/wiki/P%E2%80%93n_junction), to increase the speed of response. A photodiode is designed to operate in [reverse bias](https://en.wikipedia.org/wiki/Reverse_bias).

**Light Emitting Diode (LED):**

A light-emitting diode (LED) is a two-[lead](https://en.wikipedia.org/wiki/Lead_(electronics)) [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) [light source](https://en.wikipedia.org/wiki/Light_source). It is a [p–n junction](https://en.wikipedia.org/wiki/P%E2%80%93n_junction) [diode](https://en.wikipedia.org/wiki/Diode), which emits light when activated. When a suitable [voltage](https://en.wikipedia.org/wiki/Voltage) is applied to the leads, [électrons](https://en.wikipedia.org/wiki/Electrons) are able to recombine with [electron holes](https://en.wikipedia.org/wiki/Electron_holes) within the device, releasing energy in the form of [photons](https://en.wikipedia.org/wiki/Photon). This effect is called [electroluminescence](https://en.wikipedia.org/wiki/Electroluminescence), and the color of the light (corresponding to the energy of the photon) is determined by the energy [band gap](https://en.wikipedia.org/wiki/Band_gap) of the semiconductor.

**DB9 Connector:**

DB9 connectors were once very common on PCs and servers.  DB9 connectors are designed to work with the EIA/TIA 232 serial interface standard, which determined the function of all nine pins as a standard, so that multiple companies could design them into their products.  DB9 connectors were commonly used for serial peripheral devices like keyboards, mice, joysticks, etc. Also they are used on DB9 cable assemblies for data connectivity.

**Diode:**

In [electronics](https://en.wikipedia.org/wiki/Electronics), a diode is a two-terminal electronic that conducts primarily in one direction (asymmetric [conductance](https://en.wikipedia.org/wiki/Electrical_conductance)); it has low (ideally zero) [resistance](https://en.wikipedia.org/wiki/Electrical_resistance_and_conductance) to the flow of [current](https://en.wikipedia.org/wiki/Electric_current) in one direction, and high (ideally [infinite](https://en.wikipedia.org/wiki/Infinity)) resistance in the other. A semiconductor diode, the most common type today, is a [crystalline](https://en.wikipedia.org/wiki/Crystallinity) piece of [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) material with a [p–n junction](https://en.wikipedia.org/wiki/P%E2%80%93n_junction) connected to two electrical terminals

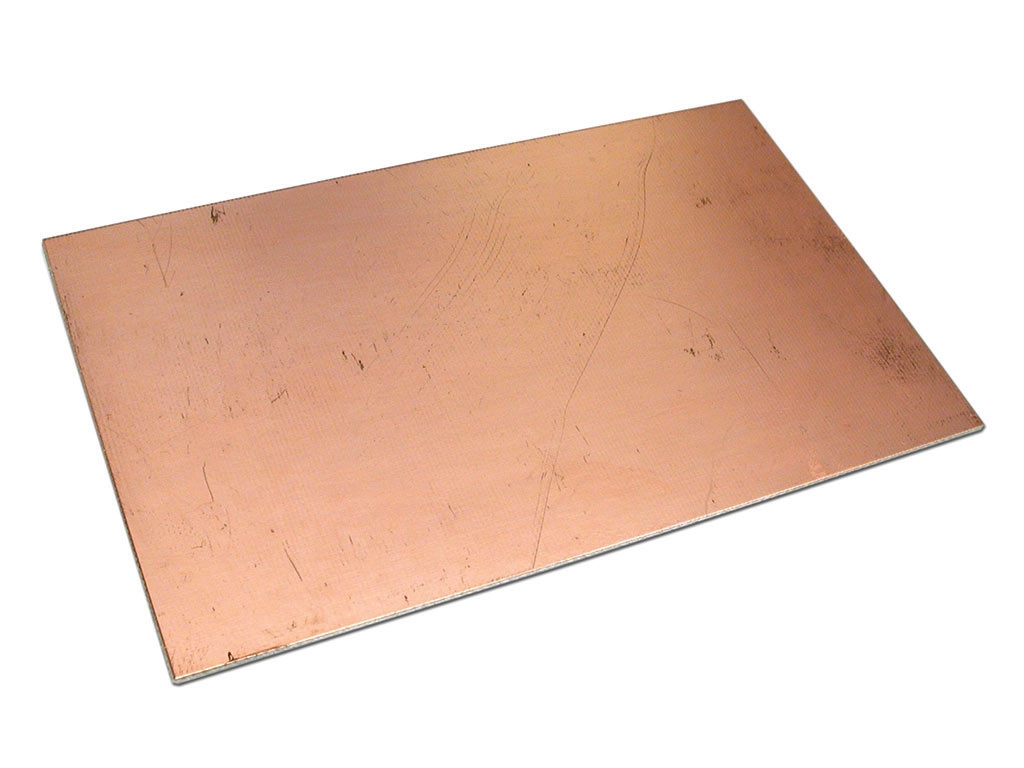
**Soldering Iron:**



A soldering iron is a hand tool used in soldering. There are many soldering irons available on the market. They come in a variety of sizes and shapes. Which soldering iron to choose for yourself depends on the soldering projects you are planning to do, as well as how often you are planning on using it. [12]

It is composed of a heated metal tip and an insulated handle. Heating is often achieved electrically, by passing an electric current (supplied through an electrical cord or battery cables) through a resistive heating element. Cordless irons can be heated by combustion of gas stored in a small tank, often using a catalytic heater rather than a flame. Simple irons less commonly used than in the past were simply a large copper bit on a handle, heated in a flame. [11]

**Copper Clad:**



Inexpensive copper-clad board material with thicknesses of 1/16" or 1/32" in sizes from 3"x5" up to 12"x12". They are available of both single- and double-sided copper-clad boards.

These are used to make your own **PCB** (Printed Circuit Board), it electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer (outer and inner layers). Multi-layer PCBs allow for much higher component density. Conductors on different layers are connected with plated-through holes called via Advanced PCBs may contain components - capacitors, resistors or active devices - embedded in the substrate.

**Hand Drill:**



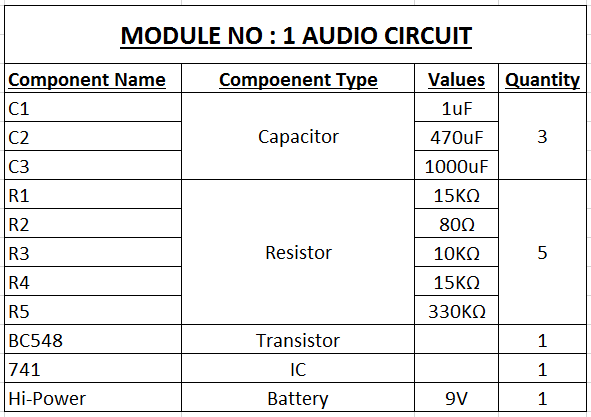
These are used for drilling a range of sizes of hole and they are very useful especially if machine drills are not available. The hand drill generally holds drill sizes from 1mm to 9mm whilst the brace will hold larger drill bits called 'forstner bits' and 'auger bits'.

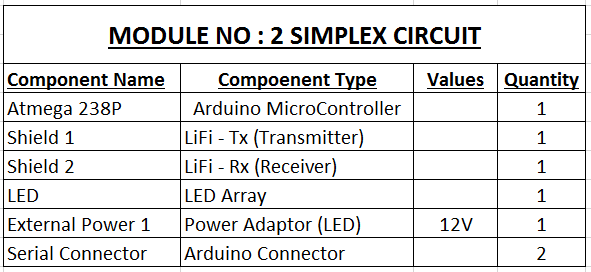
**Ferric Chloride:**

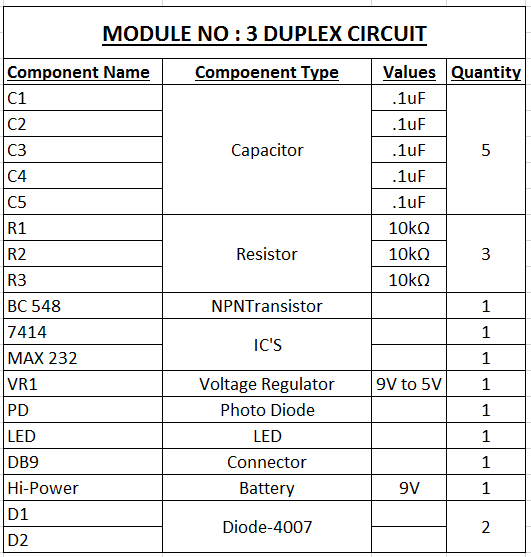


Ferric chloride, is an industrial scale commodity chemical compound, when dissolved in water, ferric chloride undergoes hydrolysis and gives off heat in an exothermic reaction. The resulting brown, acidic, and corrosive solution is used as a flocculant in sewage treatment and drinking water production, and as an etchant for copper-based metals in printed circuit boards. Anhydrous ferric chloride is a fairly strong Lewis acid, and it is used as a catalyst in organic synthesis.

## REQUIREMENT SHEET



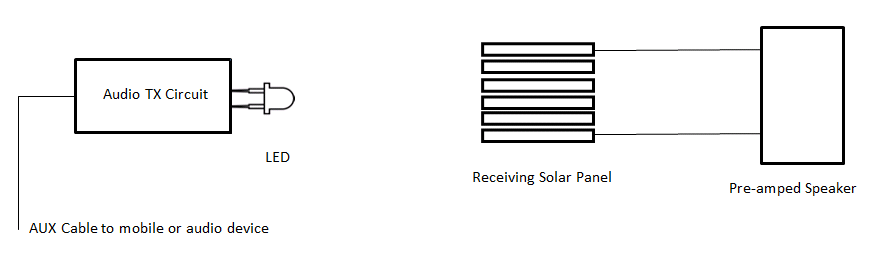




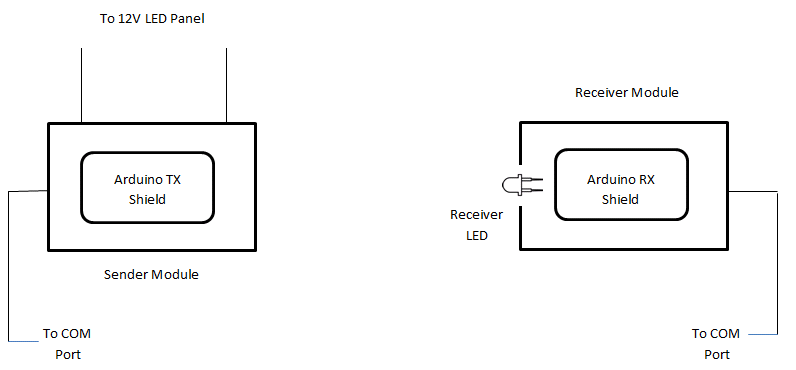
# DESIGN

## GENERAL DIAGRAMS

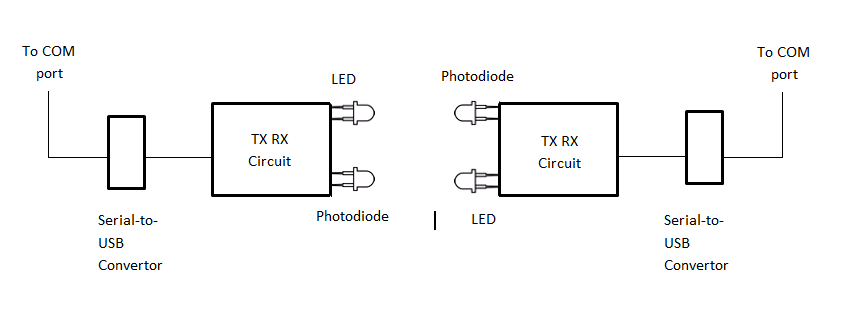
Module 1:



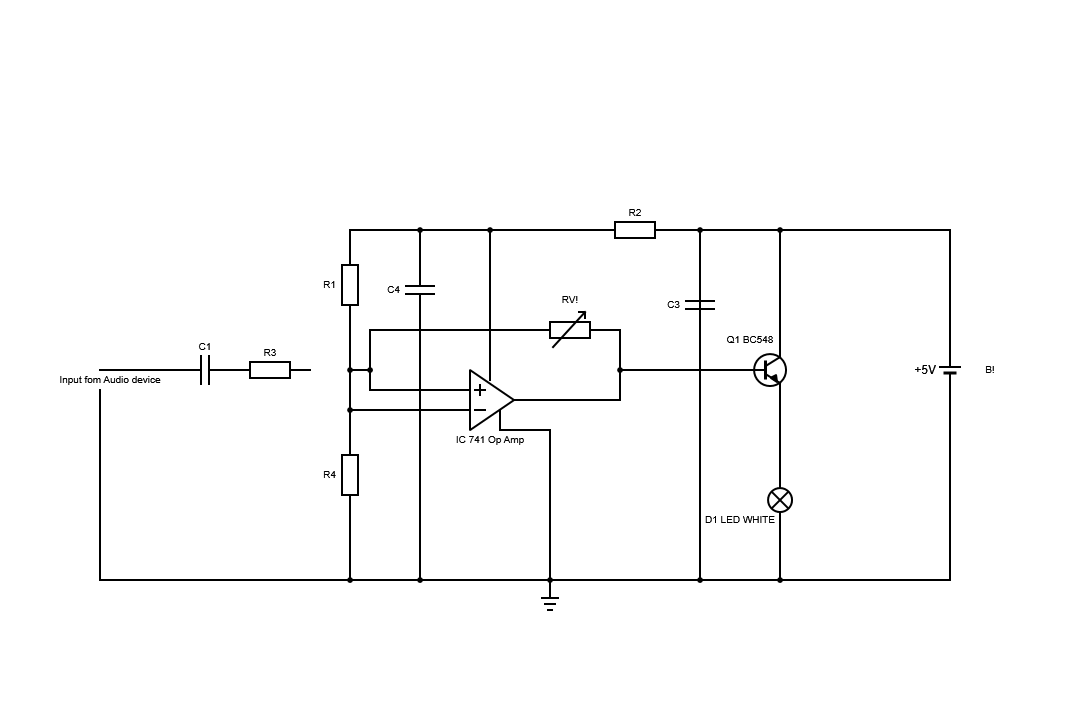
Module 2:



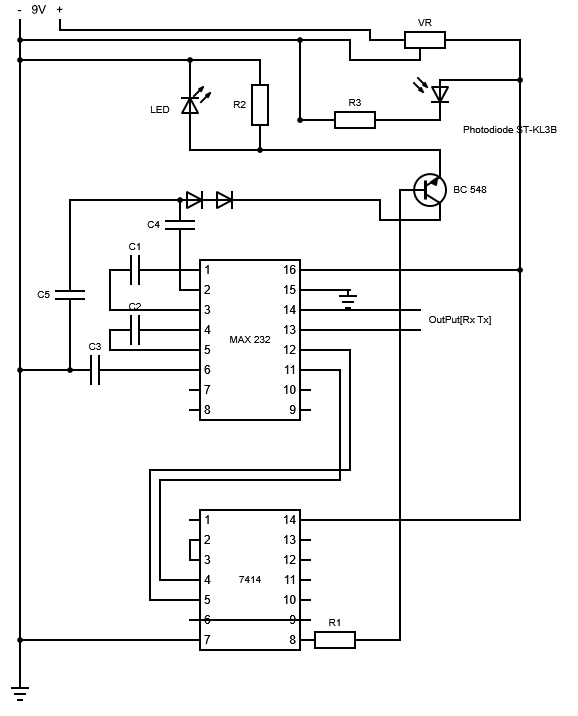
Module 3:

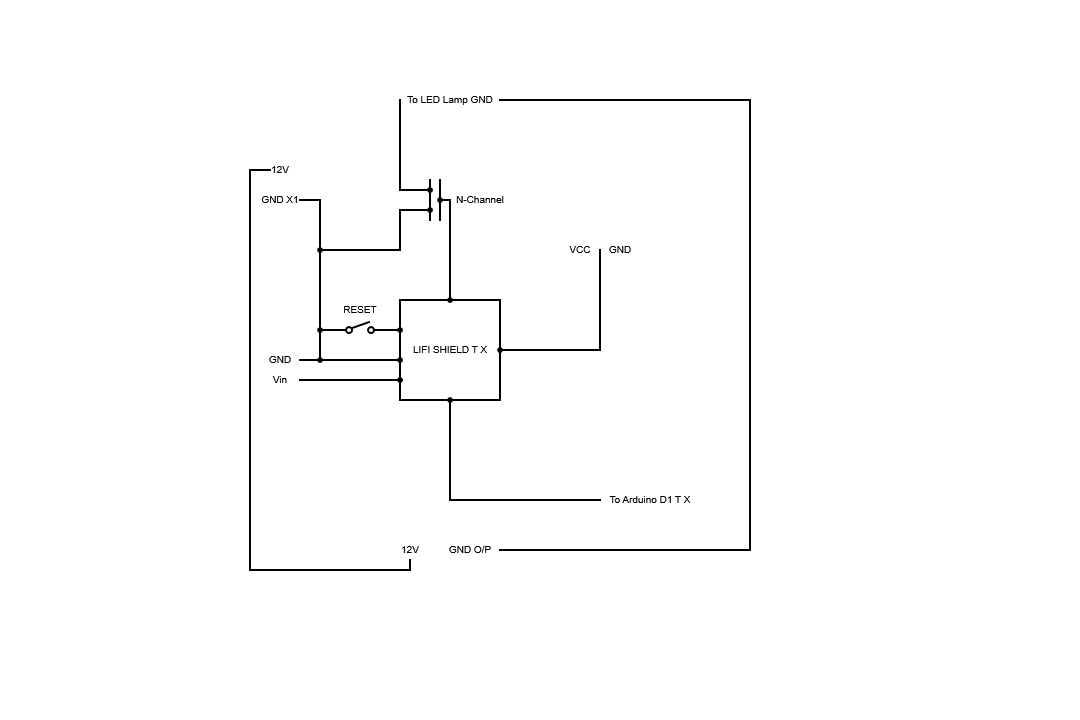


## CIRCUIT DIAGRAMS

**Module 1: Arduino Transmission**

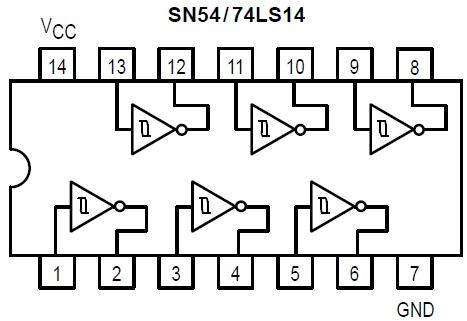
**Module 2: Duplex Circuit**



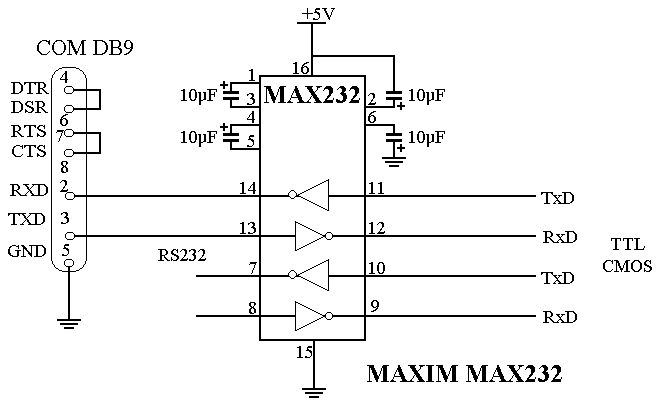
**Module 3: Arduino Tx Rx**

## LOGIC DIAGRAMS

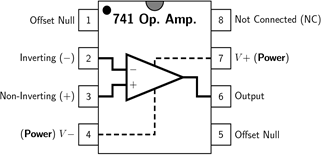
IC7414:



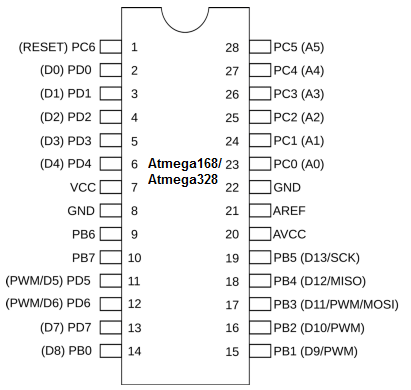
MAX 232:



IC 741:



ATMEGA 328:



# 

# SOURCE CODE

**Manifest.xml**

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5"/>

</startup>

</configuration>

**HomeFormDesigner.vb**

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()>

Partial Class homeForm

Inherits System.Windows.Forms.Form

'Form overrides dispose to clean up the component list.

<System.Diagnostics.DebuggerNonUserCode()>

Protected Overrides Sub Dispose(ByVal disposing As Boolean)

Try

If disposing AndAlso components IsNot Nothing Then

components.Dispose()

End If

Finally

MyBase.Dispose(disposing)

End Try

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough()>

Private Sub InitializeComponent()

Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(homeForm))

Me.titleBox = New System.Windows.Forms.Label()

Me.descriptionBox = New System.Windows.Forms.Label()

Me.receiveFilesButton = New System.Windows.Forms.PictureBox()

Me.receiveTextButton = New System.Windows.Forms.PictureBox()

Me.settingsButton = New System.Windows.Forms.PictureBox()

Me.sendFileButton = New System.Windows.Forms.PictureBox()

Me.sendTextButton = New System.Windows.Forms.PictureBox()

Me.logoBox = New System.Windows.Forms.PictureBox()

CType(Me.receiveFilesButton, System.ComponentModel.ISupportInitialize).BeginInit()

CType(Me.receiveTextButton, System.ComponentModel.ISupportInitialize).BeginInit()

CType(Me.settingsButton, System.ComponentModel.ISupportInitialize).BeginInit()

CType(Me.sendFileButton, System.ComponentModel.ISupportInitialize).BeginInit()

CType(Me.sendTextButton, System.ComponentModel.ISupportInitialize).BeginInit()

CType(Me.logoBox, System.ComponentModel.ISupportInitialize).BeginInit()

Me.SuspendLayout()

'

'titleBox

'

Me.titleBox.AutoSize = True

Me.titleBox.BackColor = System.Drawing.Color.Transparent

Me.titleBox.Font = New System.Drawing.Font("Microsoft Sans Serif", 25.0!)

Me.titleBox.Location = New System.Drawing.Point(204, 57)

Me.titleBox.Name = "titleBox"

Me.titleBox.Size = New System.Drawing.Size(347, 39)

Me.titleBox.TabIndex = 1

Me.titleBox.Text = "Welcome To VisiLight"

'

'descriptionBox

'

Me.descriptionBox.AutoSize = True

Me.descriptionBox.BackColor = System.Drawing.Color.Transparent

Me.descriptionBox.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.0!)

Me.descriptionBox.Location = New System.Drawing.Point(206, 106)

Me.descriptionBox.Name = "descriptionBox"

Me.descriptionBox.Size = New System.Drawing.Size(372, 25)

Me.descriptionBox.TabIndex = 2

Me.descriptionBox.Text = "A new way to send data over Visible Light"

'

'receiveFilesButton

'

Me.receiveFilesButton.BackColor = System.Drawing.Color.FromArgb(CType(CType(44, Byte), Integer), CType(CType(93, Byte), Integer), CType(CType(136, Byte), Integer))

Me.receiveFilesButton.BackgroundImage = Global.WindowsApplication3.My.Resources.Resources.receiveFiles

Me.receiveFilesButton.Cursor = System.Windows.Forms.Cursors.Hand

Me.receiveFilesButton.Location = New System.Drawing.Point(211, 400)

Me.receiveFilesButton.Name = "receiveFilesButton"

Me.receiveFilesButton.Size = New System.Drawing.Size(180, 150)

Me.receiveFilesButton.TabIndex = 7

Me.receiveFilesButton.TabStop = False

'

'receiveTextButton

'

Me.receiveTextButton.BackColor = System.Drawing.Color.FromArgb(CType(CType(44, Byte), Integer), CType(CType(93, Byte), Integer), CType(CType(136, Byte), Integer))

Me.receiveTextButton.BackgroundImage = Global.WindowsApplication3.My.Resources.Resources.receiveText

Me.receiveTextButton.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch

Me.receiveTextButton.Cursor = System.Windows.Forms.Cursors.Hand

Me.receiveTextButton.Location = New System.Drawing.Point(12, 400)

Me.receiveTextButton.Name = "receiveTextButton"

Me.receiveTextButton.Size = New System.Drawing.Size(180, 150)

Me.receiveTextButton.TabIndex = 6

Me.receiveTextButton.TabStop = False

'

'settingsButton

'

Me.settingsButton.BackColor = System.Drawing.Color.FromArgb(CType(CType(44, Byte), Integer), CType(CType(93, Byte), Integer), CType(CType(136, Byte), Integer))

Me.settingsButton.BackgroundImage = Global.WindowsApplication3.My.Resources.Resources.settings1

Me.settingsButton.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Center

Me.settingsButton.Cursor = System.Windows.Forms.Cursors.Hand

Me.settingsButton.Location = New System.Drawing.Point(413, 231)

Me.settingsButton.Name = "settingsButton"

Me.settingsButton.Size = New System.Drawing.Size(180, 319)

Me.settingsButton.TabIndex = 5

Me.settingsButton.TabStop = False

'

'sendFileButton

'

Me.sendFileButton.BackColor = System.Drawing.Color.FromArgb(CType(CType(44, Byte), Integer), CType(CType(93, Byte), Integer), CType(CType(136, Byte), Integer))

Me.sendFileButton.BackgroundImage = Global.WindowsApplication3.My.Resources.Resources.sendFiles1

Me.sendFileButton.Cursor = System.Windows.Forms.Cursors.Hand

Me.sendFileButton.Location = New System.Drawing.Point(211, 231)

Me.sendFileButton.Name = "sendFileButton"

Me.sendFileButton.Size = New System.Drawing.Size(180, 150)

Me.sendFileButton.TabIndex = 4

Me.sendFileButton.TabStop = False

'

'sendTextButton

'

Me.sendTextButton.BackColor = System.Drawing.Color.FromArgb(CType(CType(44, Byte), Integer), CType(CType(93, Byte), Integer), CType(CType(136, Byte), Integer))

Me.sendTextButton.BackgroundImage = Global.WindowsApplication3.My.Resources.Resources.sendText

Me.sendTextButton.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch

Me.sendTextButton.Cursor = System.Windows.Forms.Cursors.Hand

Me.sendTextButton.Location = New System.Drawing.Point(12, 231)

Me.sendTextButton.Name = "sendTextButton"

Me.sendTextButton.Size = New System.Drawing.Size(180, 150)

Me.sendTextButton.TabIndex = 3

Me.sendTextButton.TabStop = False

'

'logoBox

'

Me.logoBox.BackColor = System.Drawing.Color.Transparent

Me.logoBox.BackgroundImage = CType(resources.GetObject("logoBox.BackgroundImage"), System.Drawing.Image)

Me.logoBox.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch

Me.logoBox.Image = CType(resources.GetObject("logoBox.Image"), System.Drawing.Image)

Me.logoBox.Location = New System.Drawing.Point(30, 27)

Me.logoBox.Name = "logoBox"

Me.logoBox.Size = New System.Drawing.Size(138, 141)

Me.logoBox.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage

Me.logoBox.TabIndex = 0

Me.logoBox.TabStop = False

'

'homeForm

'

Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)

Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font

Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(59, Byte), Integer), CType(CType(157, Byte), Integer), CType(CType(243, Byte), Integer))

Me.ClientSize = New System.Drawing.Size(604, 576)

Me.Controls.Add(Me.receiveFilesButton)

Me.Controls.Add(Me.receiveTextButton)

Me.Controls.Add(Me.settingsButton)

Me.Controls.Add(Me.sendFileButton)

Me.Controls.Add(Me.sendTextButton)

Me.Controls.Add(Me.descriptionBox)

Me.Controls.Add(Me.titleBox)

Me.Controls.Add(Me.logoBox)

Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog

Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)

Me.Name = "homeForm"

Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen

Me.Text = "VisiLight"

CType(Me.receiveFilesButton, System.ComponentModel.ISupportInitialize).EndInit()

CType(Me.receiveTextButton, System.ComponentModel.ISupportInitialize).EndInit()

CType(Me.settingsButton, System.ComponentModel.ISupportInitialize).EndInit()

CType(Me.sendFileButton, System.ComponentModel.ISupportInitialize).EndInit()

CType(Me.sendTextButton, System.ComponentModel.ISupportInitialize).EndInit()

CType(Me.logoBox, System.ComponentModel.ISupportInitialize).EndInit()

Me.ResumeLayout(False)

Me.PerformLayout()

End Sub

Friend WithEvents sendTextButton As PictureBox

Friend WithEvents receiveTextButton As PictureBox

Friend WithEvents receiveFilesButton As PictureBox

End Class

**HomeForm.vb**

Public Class homeForm

Friend WithEvents logoBox As PictureBox

Friend WithEvents titleBox As Label

Friend WithEvents descriptionBox As Label

Friend WithEvents sendFileButton As PictureBox

Friend WithEvents settingsButton As PictureBox

Private Sub sendTextButton\_MouseEnter(sender As Object, e As EventArgs) Handles sendTextButton.MouseEnter

sendTextButton.BackColor = System.Drawing.Color.FromArgb(75, 115, 151)

End Sub

Private Sub sendTextButton\_MouseLeave(sender As Object, e As EventArgs) Handles sendTextButton.MouseLeave

sendTextButton.BackColor = System.Drawing.Color.FromArgb(44, 93, 136)

End Sub

Private Sub sendFileButton\_MouseEnter(sender As Object, e As EventArgs) Handles sendFileButton.MouseEnter

sendFileButton.BackColor = System.Drawing.Color.FromArgb(75, 115, 151)

End Sub

Private Sub sendFileButton\_MouseLeave(sender As Object, e As EventArgs) Handles sendFileButton.MouseLeave

sendFileButton.BackColor = System.Drawing.Color.FromArgb(44, 93, 136)

End Sub

Private Sub settingsButton\_MouseEnter(sender As Object, e As EventArgs) Handles settingsButton.MouseEnter

settingsButton.BackColor = System.Drawing.Color.FromArgb(75, 115, 151)

End Sub

Private Sub settingsButton\_MouseLeave(sender As Object, e As EventArgs) Handles settingsButton.MouseLeave

settingsButton.BackColor = System.Drawing.Color.FromArgb(44, 93, 136)

End Sub

Private Sub receiveTextButton\_MouseEnter(sender As Object, e As EventArgs) Handles receiveTextButton.MouseEnter

receiveTextButton.BackColor = System.Drawing.Color.FromArgb(75, 115, 151)

End Sub

Private Sub receiveTextButton\_MouseLeave(sender As Object, e As EventArgs) Handles receiveTextButton.MouseLeave

receiveTextButton.BackColor = System.Drawing.Color.FromArgb(44, 93, 136)

End Sub

Private Sub receiveFilesButton\_MouseEnter(sender As Object, e As EventArgs) Handles receiveFilesButton.MouseEnter

receiveFilesButton.BackColor = System.Drawing.Color.FromArgb(75, 115, 151)

End Sub

Private Sub receiveFilesButton\_MouseLeave(sender As Object, e As EventArgs) Handles receiveFilesButton.MouseLeave

receiveFilesButton.BackColor = System.Drawing.Color.FromArgb(44, 93, 136)

End Sub

Private Sub sendTextButton\_MouseClick(sender As Object, e As MouseEventArgs) Handles sendTextButton.MouseClick

Me.Hide()

sendTextForm.Show()

End Sub

Private Sub settingsButton\_Click(sender As Object, e As EventArgs) Handles settingsButton.Click

Me.Hide()

settingsForm.Show()

End Sub

Private Sub sendFileButton\_Click(sender As Object, e As EventArgs)

End Sub

Private Sub receiveTextButton\_Click(sender As Object, e As EventArgs) Handles receiveTextButton.Click

Me.Hide()

receiveTextForm.Show()

End Sub

Private Sub sendFileButton\_Click\_1(sender As Object, e As EventArgs) Handles sendFileButton.Click

Me.Hide()

sendFilesForm.Show()

End Sub

End Class

**ReceiveTextFormDesigner.vb**

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()>

Partial Class receiveTextForm

Inherits System.Windows.Forms.Form

'Form overrides dispose to clean up the component list.

<System.Diagnostics.DebuggerNonUserCode()>

Protected Overrides Sub Dispose(ByVal disposing As Boolean)

Try

If disposing AndAlso components IsNot Nothing Then

components.Dispose()

End If

Finally

MyBase.Dispose(disposing)

End Try

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough()>

Private Sub InitializeComponent()

Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(receiveTextForm))

Me.Panel1 = New System.Windows.Forms.Panel()

Me.formTitle = New System.Windows.Forms.Label()

Me.receiveTextBox = New System.Windows.Forms.RichTextBox()

Me.Button1 = New System.Windows.Forms.Button()

Me.Panel1.SuspendLayout()

Me.SuspendLayout()

'

'Panel1

'

Me.Panel1.BackColor = System.Drawing.Color.FromArgb(CType(CType(59, Byte), Integer), CType(CType(157, Byte), Integer), CType(CType(243, Byte), Integer))

Me.Panel1.Controls.Add(Me.formTitle)

Me.Panel1.Dock = System.Windows.Forms.DockStyle.Top

Me.Panel1.Location = New System.Drawing.Point(0, 0)

Me.Panel1.Name = "Panel1"

Me.Panel1.Size = New System.Drawing.Size(604, 70)

Me.Panel1.TabIndex = 0

'

'formTitle

'

Me.formTitle.AutoSize = True

Me.formTitle.Font = New System.Drawing.Font("Microsoft Sans Serif", 16.0!)

Me.formTitle.Location = New System.Drawing.Point(12, 23)

Me.formTitle.Margin = New System.Windows.Forms.Padding(3)

Me.formTitle.Name = "formTitle"

Me.formTitle.Size = New System.Drawing.Size(276, 26)

Me.formTitle.TabIndex = 2

Me.formTitle.Text = "Connected to Com Port XX"

'

'receiveTextBox

'

Me.receiveTextBox.BackColor = System.Drawing.Color.FromArgb(CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer))

Me.receiveTextBox.BorderStyle = System.Windows.Forms.BorderStyle.None

Me.receiveTextBox.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.0!)

Me.receiveTextBox.Location = New System.Drawing.Point(16, 76)

Me.receiveTextBox.Name = "receiveTextBox"

Me.receiveTextBox.Size = New System.Drawing.Size(576, 302)

Me.receiveTextBox.TabIndex = 1

Me.receiveTextBox.Text = "Sample Text to show text box edit once you finish coding"

'

'Button1

'

Me.Button1.Location = New System.Drawing.Point(256, 384)

Me.Button1.Name = "Button1"

Me.Button1.Size = New System.Drawing.Size(75, 23)

Me.Button1.TabIndex = 2

Me.Button1.Text = "Button1"

Me.Button1.UseVisualStyleBackColor = True

'

'receiveTextForm

'

Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)

Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font

Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer))

Me.ClientSize = New System.Drawing.Size(604, 576)

Me.Controls.Add(Me.Button1)

Me.Controls.Add(Me.receiveTextBox)

Me.Controls.Add(Me.Panel1)

Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog

Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)

Me.Name = "receiveTextForm"

Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen

Me.Text = "VisiLight Receive Text"

Me.Panel1.ResumeLayout(False)

Me.Panel1.PerformLayout()

Me.ResumeLayout(False)

End Sub

Friend WithEvents Panel1 As Panel

Friend WithEvents formTitle As Label

Friend WithEvents receiveTextBox As RichTextBox

Friend WithEvents Button1 As Button

End Class

**ReceiveTextForm.vb**

Imports System.IO.Ports

Public Class receiveTextForm

Dim WithEvents mySerialPort As SerialPort = New System.IO.Ports.SerialPort

Private Sub receiveTextForm\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

homeForm.Show()

End Sub

Private Sub receiveTextForm\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

With mySerialPort

.PortName = "COM4"

.BaudRate = 38400

.Parity = Parity.None

.DataBits = 8

.StopBits = StopBits.One

End With

mySerialPort.ReadTimeout = 0

Try

mySerialPort.Open()

receiveTextBox.Text = ""

While (True)

Try

Dim msgr = mySerialPort.ReadLine()

receiveTextBox.Text = receiveTextBox.Text + msgr + Environment.NewLine() + vbCrLf

Catch generatedExceptionName As TimeoutException

End Try

End While

mySerialPort.Close()

Catch asd As System.IO.IOException

MsgBox("Serial Port Was Not detected, Please check your COM port number in the settings")

End Try

End Sub

Private Sub receiveTextBox\_TextChanged(sender As Object, e As EventArgs) Handles receiveTextBox.TextChanged

End Sub

End Class

**SendFileFormDesigner .vb**

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> \_

Partial Class sendFilesForm

Inherits System.Windows.Forms.Form

'Form overrides dispose to clean up the component list.

<System.Diagnostics.DebuggerNonUserCode()> \_

Protected Overrides Sub Dispose(ByVal disposing As Boolean)

Try

If disposing AndAlso components IsNot Nothing Then

components.Dispose()

End If

Finally

MyBase.Dispose(disposing)

End Try

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough()> \_

Private Sub InitializeComponent()

Me.OpenFileDialog1 = New System.Windows.Forms.OpenFileDialog()

Me.Button1 = New System.Windows.Forms.Button()

Me.PictureBox1 = New System.Windows.Forms.PictureBox()

Me.Button2 = New System.Windows.Forms.Button()

Me.RichTextBox1 = New System.Windows.Forms.RichTextBox()

CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()

Me.SuspendLayout()

'

'OpenFileDialog1

'

Me.OpenFileDialog1.FileName = "OpenFileDialog1"

'

'Button1

'

Me.Button1.Location = New System.Drawing.Point(12, 328)

Me.Button1.Name = "Button1"

Me.Button1.Size = New System.Drawing.Size(92, 32)

Me.Button1.TabIndex = 0

Me.Button1.Text = "Button1"

Me.Button1.UseVisualStyleBackColor = True

'

'PictureBox1

'

Me.PictureBox1.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch

Me.PictureBox1.Location = New System.Drawing.Point(12, 32)

Me.PictureBox1.Name = "PictureBox1"

Me.PictureBox1.Size = New System.Drawing.Size(101, 290)

Me.PictureBox1.TabIndex = 1

Me.PictureBox1.TabStop = False

'

'Button2

'

Me.Button2.Location = New System.Drawing.Point(110, 328)

Me.Button2.Name = "Button2"

Me.Button2.Size = New System.Drawing.Size(95, 32)

Me.Button2.TabIndex = 2

Me.Button2.Text = "Button2"

Me.Button2.UseVisualStyleBackColor = True

'

'RichTextBox1

'

Me.RichTextBox1.Location = New System.Drawing.Point(131, 32)

Me.RichTextBox1.Name = "RichTextBox1"

Me.RichTextBox1.Size = New System.Drawing.Size(838, 290)

Me.RichTextBox1.TabIndex = 3

Me.RichTextBox1.Text = ""

'

'sendFilesForm

Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)

Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font

Me.ClientSize = New System.Drawing.Size(1031, 451)

Me.Controls.Add(Me.RichTextBox1)

Me.Controls.Add(Me.Button2)

Me.Controls.Add(Me.PictureBox1)

Me.Controls.Add(Me.Button1)

Me.Name = "sendFilesForm"

Me.Text = "sendFilesForm"

CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()

Me.ResumeLayout(False)

End Sub

Friend WithEvents OpenFileDialog1 As System.Windows.Forms.OpenFileDialog

Friend WithEvents Button1 As System.Windows.Forms.Button

Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox

Friend WithEvents Button2 As System.Windows.Forms.Button

Friend WithEvents RichTextBox1 As System.Windows.Forms.RichTextBox

End Class

**SendFileForm.vb**

Imports System.IO

Public Class sendFilesForm

Public fileFullPath As String

Public dirPath As String

Public txtFilePath As String

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

If OpenFileDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then

PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)

fileFullPath = OpenFileDialog1.FileName

dirPath = Path.GetDirectoryName(fileFullPath)

txtFilePath = dirPath + "\temp.txt"

FileCopy(fileFullPath, txtFilePath)

End If

End Sub

Private Sub sendFilesForm\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

homeForm.Show()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Dim content As String

'Using textReader As New System.IO.StreamReader("C:\Users\sbmp\Desktop\temp.txt")

'content = TextReader.ReadToEnd

'End Using

content = File.ReadAllText(fileFullPath)

RichTextBox1.Text = content

End Sub

End Class

**SendTextFormDesigner.vb**

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()>

Partial Class sendTextForm

Inherits System.Windows.Forms.Form

'Form overrides dispose to clean up the component list.

<System.Diagnostics.DebuggerNonUserCode()>

Protected Overrides Sub Dispose(ByVal disposing As Boolean)

Try

If disposing AndAlso components IsNot Nothing Then

components.Dispose()

End If

Finally

MyBase.Dispose(disposing)

End Try

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough()>

Private Sub InitializeComponent()

Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(sendTextForm))

Me.topPanel = New System.Windows.Forms.Panel()

Me.formTitle = New System.Windows.Forms.Label()

Me.sendTextBox = New System.Windows.Forms.RichTextBox()

Me.sendButton = New System.Windows.Forms.Button()

Me.topPanel.SuspendLayout()

Me.SuspendLayout()

'

'topPanel

'

Me.topPanel.BackColor = System.Drawing.Color.FromArgb(CType(CType(59, Byte), Integer), CType(CType(157, Byte), Integer), CType(CType(243, Byte), Integer))

Me.topPanel.Controls.Add(Me.formTitle)

Me.topPanel.Dock = System.Windows.Forms.DockStyle.Top

Me.topPanel.Location = New System.Drawing.Point(0, 0)

Me.topPanel.Name = "topPanel"

Me.topPanel.Size = New System.Drawing.Size(604, 70)

Me.topPanel.TabIndex = 0

'

'formTitle

'

Me.formTitle.AutoSize = True

Me.formTitle.Font = New System.Drawing.Font("Microsoft Sans Serif", 16.0!)

Me.formTitle.Location = New System.Drawing.Point(12, 23)

Me.formTitle.Name = "formTitle"

Me.formTitle.Size = New System.Drawing.Size(276, 26)

Me.formTitle.TabIndex = 1

Me.formTitle.Text = "Connected to Com Port XX"

'

'sendTextBox

'

Me.sendTextBox.AcceptsTab = True

Me.sendTextBox.BackColor = System.Drawing.Color.FromArgb(CType(CType(80, Byte), Integer), CType(CType(80, Byte), Integer), CType(CType(80, Byte), Integer))

Me.sendTextBox.BorderStyle = System.Windows.Forms.BorderStyle.None

Me.sendTextBox.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.0!)

Me.sendTextBox.Location = New System.Drawing.Point(17, 494)

Me.sendTextBox.Name = "sendTextBox"

Me.sendTextBox.Size = New System.Drawing.Size(483, 58)

Me.sendTextBox.TabIndex = 3

Me.sendTextBox.Text = "Enter You Text Here"

'

'sendButton

'

Me.sendButton.BackColor = System.Drawing.Color.Transparent

Me.sendButton.BackgroundImage = Global.WindowsApplication3.My.Resources.Resources.sendButton

Me.sendButton.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch

Me.sendButton.Cursor = System.Windows.Forms.Cursors.Hand

Me.sendButton.Location = New System.Drawing.Point(523, 494)

Me.sendButton.Name = "sendButton"

Me.sendButton.Size = New System.Drawing.Size(58, 58)

Me.sendButton.TabIndex = 4

Me.sendButton.UseVisualStyleBackColor = False

'

'sendTextForm

'

Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)

Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font

Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer))

Me.ClientSize = New System.Drawing.Size(604, 576)

Me.Controls.Add(Me.sendButton)

Me.Controls.Add(Me.sendTextBox)

Me.Controls.Add(Me.topPanel)

Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog

Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)

Me.Location = New System.Drawing.Point(200, 200)

Me.Name = "sendTextForm"

Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen

Me.Text = "VisiLight Send Text"

Me.topPanel.ResumeLayout(False)

Me.topPanel.PerformLayout()

Me.ResumeLayout(False)

End Sub

Friend WithEvents sendButton As Button

End Class

**SendTextForm.vb**

Imports System.IO.Ports

Public Class sendTextForm

Dim WithEvents mySerialPort As SerialPort = New System.IO.Ports.SerialPort

Friend WithEvents topPanel As Panel

Friend WithEvents formTitle As Label

Private Sub PictureBox1\_MouseHover(sender As Object, e As EventArgs)

sendButton.Image = Image.FromFile("C:\Users\marmi\Documents\Visual Studio 2015\Projects\WindowsApplication3\WindowsApplication3\VisiLight Res\sendButtonHover.png")

End Sub

Private Sub sendButton\_MouseLeave(sender As Object, e As EventArgs)

sendButton.Image = Image.FromFile("C:\Users\marmi\Documents\Visual Studio 2015\Projects\WindowsApplication3\WindowsApplication3\VisiLight Res\sendButton.png")

End Sub

Friend WithEvents sendTextBox As RichTextBox

Private Sub RichTextBox1\_GotFocus(sender As Object, e As EventArgs) Handles sendTextBox.GotFocus

If sendTextBox.Text = "Enter You Text Here" Then

sendTextBox.Text = ""

End If

End Sub

Private Sub RichTextBox1\_LostFocus(sender As Object, e As EventArgs) Handles sendTextBox.LostFocus

If sendTextBox.Text = "" Then

sendTextBox.Text = "Enter You Text Here"

End If

End Sub

Private Sub Form2\_Activated(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Activated

sendButton.Focus()

End Sub

Private Sub Form2\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

homeForm.Show()

End Sub

Private Sub sendButton\_Click(sender As Object, e As EventArgs) Handles sendButton.Click

With mySerialPort

.PortName = "COM4"

.BaudRate = 38400

.Parity = Parity.None

.DataBits = 8

.StopBits = StopBits.One

End With

Try

mySerialPort.Open()

Dim msg As String = sendTextBox.Text

mySerialPort.WriteLine(msg)

mySerialPort.Close()

Catch asd As System.IO.IOException

MsgBox("Serial Port Was not detected, Please check your COM port number in the settings")

End Try

End Sub

Private Sub sendTextForm\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

**SettingsFormDesigner.vb**

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> \_

Partial Class settingsForm

Inherits System.Windows.Forms.Form

'Form overrides dispose to clean up the component list.

<System.Diagnostics.DebuggerNonUserCode()> \_

Protected Overrides Sub Dispose(ByVal disposing As Boolean)

Try

If disposing AndAlso components IsNot Nothing Then

components.Dispose()

End If

Finally

MyBase.Dispose(disposing)

End Try

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough()> \_

Private Sub InitializeComponent()

Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(settingsForm))

Me.topPanel = New System.Windows.Forms.Panel()

Me.formTitle = New System.Windows.Forms.Label()

Me.baudrateSelect = New System.Windows.Forms.ComboBox()

Me.serialPortSelect = New System.Windows.Forms.ComboBox()

Me.baudRateLabel = New System.Windows.Forms.Label()

Me.comPortLabel = New System.Windows.Forms.Label()

Me.setButton = New System.Windows.Forms.Button()

Me.topPanel.SuspendLayout()

Me.SuspendLayout()

'

'topPanel

'

Me.topPanel.BackColor = System.Drawing.Color.FromArgb(CType(CType(59, Byte), Integer), CType(CType(157, Byte), Integer), CType(CType(243, Byte), Integer))

Me.topPanel.Controls.Add(Me.formTitle)

Me.topPanel.Dock = System.Windows.Forms.DockStyle.Top

Me.topPanel.Location = New System.Drawing.Point(0, 0)

Me.topPanel.Name = "topPanel"

Me.topPanel.Size = New System.Drawing.Size(604, 70)

Me.topPanel.TabIndex = 0

'

'formTitle

'

Me.formTitle.AutoSize = True

Me.formTitle.Font = New System.Drawing.Font("Microsoft Sans Serif", 16.0!)

Me.formTitle.Location = New System.Drawing.Point(12, 23)

Me.formTitle.Name = "formTitle"

Me.formTitle.Size = New System.Drawing.Size(91, 26)

Me.formTitle.TabIndex = 0

Me.formTitle.Text = "Settings"

'

'baudrateSelect

'

Me.baudrateSelect.Font = New System.Drawing.Font("Microsoft Sans Serif", 11.0!)

Me.baudrateSelect.FormattingEnabled = True

Me.baudrateSelect.Items.AddRange(New Object() {"9600", "19200", "38400"})

Me.baudrateSelect.Location = New System.Drawing.Point(210, 194)

Me.baudrateSelect.Name = "baudrateSelect"

Me.baudrateSelect.Size = New System.Drawing.Size(158, 26)

Me.baudrateSelect.TabIndex = 1

'

'serialPortSelect

'

Me.serialPortSelect.Font = New System.Drawing.Font("Microsoft Sans Serif", 11.0!)

Me.serialPortSelect.FormattingEnabled = True

Me.serialPortSelect.Items.AddRange(New Object() {"COM1", "COM2", "COM3", "COM4", "COM5", "COM6", "COM7", "COM8", "COM9", "COM10"})

Me.serialPortSelect.Location = New System.Drawing.Point(211, 284)

Me.serialPortSelect.Name = "serialPortSelect"

Me.serialPortSelect.Size = New System.Drawing.Size(157, 26)

Me.serialPortSelect.TabIndex = 2

'

'baudRateLabel

'

Me.baudRateLabel.AutoSize = True

Me.baudRateLabel.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!, System.Drawing.FontStyle.Bold)

Me.baudRateLabel.Location = New System.Drawing.Point(207, 171)

Me.baudRateLabel.Name = "baudRateLabel"

Me.baudRateLabel.Size = New System.Drawing.Size(95, 20)

Me.baudRateLabel.TabIndex = 3

Me.baudRateLabel.Text = "Baud Rate"

'

'comPortLabel

'

Me.comPortLabel.AutoSize = True

Me.comPortLabel.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!, System.Drawing.FontStyle.Bold)

Me.comPortLabel.Location = New System.Drawing.Point(206, 261)

Me.comPortLabel.Name = "comPortLabel"

Me.comPortLabel.Size = New System.Drawing.Size(100, 20)

Me.comPortLabel.TabIndex = 4

Me.comPortLabel.Text = "COM PORT"

'

'setButton

'

Me.setButton.Location = New System.Drawing.Point(211, 373)

Me.setButton.Name = "setButton"

Me.setButton.Size = New System.Drawing.Size(157, 41)

Me.setButton.TabIndex = 5

Me.setButton.Text = "Set"

Me.setButton.UseVisualStyleBackColor = True

'

'settingsForm

'

Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)

Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font

Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer), CType(CType(120, Byte), Integer))

Me.ClientSize = New System.Drawing.Size(604, 576)

Me.Controls.Add(Me.setButton)

Me.Controls.Add(Me.comPortLabel)

Me.Controls.Add(Me.baudRateLabel)

Me.Controls.Add(Me.serialPortSelect)

Me.Controls.Add(Me.baudrateSelect)

Me.Controls.Add(Me.topPanel)

Me.Font = New System.Drawing.Font("Microsoft Sans Serif", 8.25!)

Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog

Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)

Me.Location = New System.Drawing.Point(200, 200)

Me.Name = "settingsForm"

Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen

Me.Text = "VisiLight Settings"

Me.topPanel.ResumeLayout(False)

Me.topPanel.PerformLayout()

Me.ResumeLayout(False)

Me.PerformLayout()

End Sub

End Class

**SettingsForm.vb**

Public Class settingsForm

Friend WithEvents topPanel As Panel

Friend WithEvents formTitle As Label

Friend WithEvents baudrateSelect As ComboBox

Private Sub Form3\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

homeForm.Show()

End Sub

Friend WithEvents serialPortSelect As ComboBox

Friend WithEvents baudRateLabel As Label

Friend WithEvents comPortLabel As Label

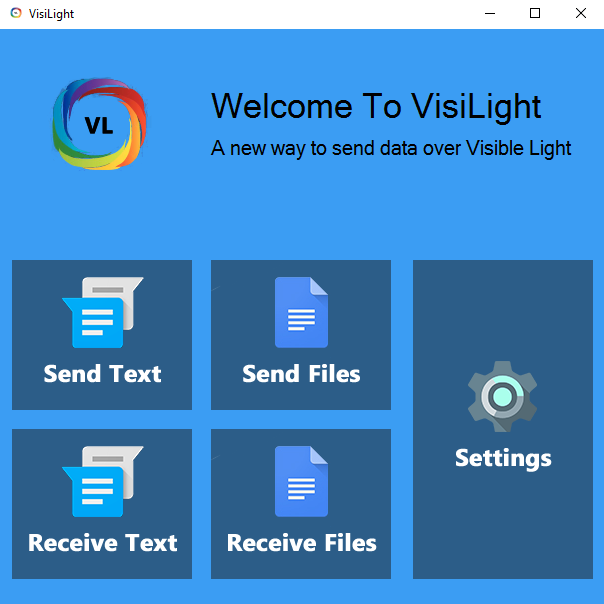
Friend WithEvents setButton As Button

End Class

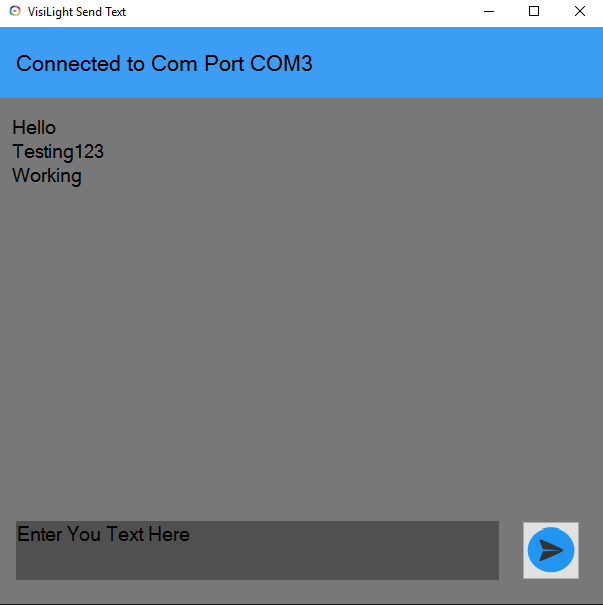
# 

# SCREENSHOTS

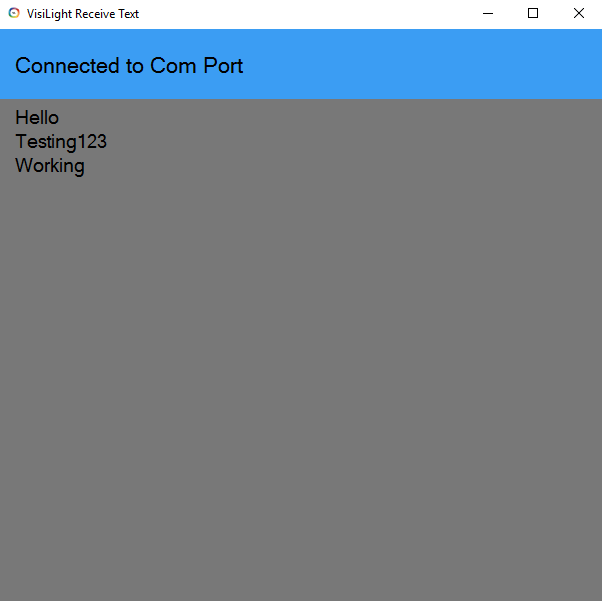
**Home Screen:**



**Send Text:**



**Receive Text:**



# CONCLUSION

## FUTURE SCOPE

Regular LEDs can be turned into optical WLAN with only a few additional components thanks to visible light communication (in short, VLC). The lights are then not just lighting up; they also transfer data. They send films in HD quality to your iPhone or laptop, with no loss in quality, quickly and safely. For a long time, this was just a vision for the future. However, the scientists were able to present the results of the project in Rennes, France. They were able to transfer data at a rate of 100 megabits per second (Mbit/s) without any losses, using LEDs in the ceiling that light up more than ten square meters (90 square feet). The receiver can be placed anywhere within this radius, which is currently the maximum range. One advantage is that it takes only a few components to prepare the LEDs so that they function as transfer media. The scientists emphasize that VLC is not intended to replace regular WLAN or UMTS. It is best suited as an additional option for data transfer where radio transmission networks are not desired or not possible? Without needing new cables or equipment in the house. In airplanes, each passenger could view his own entertainment program on a display, saving aircraft manufacturing miles of cables.

# 

# REFERENCES

1. en.wikipedia.org/wiki/Visible\_light\_communication
2. [www.wpi.edu/Pubs/E-project/Available/E-project-032814-001416/unrestricted/](http://www.wpi.edu/Pubs/E-project/Available/E-project-032814-001416/unrestricted/)
3. [www.ijeit.com/Vol%203/Issue%2012/IJEIT1412201406\_07.pdf](http://www.ijeit.com/Vol%203/Issue%2012/IJEIT1412201406_07.pdf)
4. [www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx](http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx)
5. [www.hilgraeve.com/hyperterminal/](http://www.hilgraeve.com/hyperterminal/)
6. [www.slideshare.net/sagineshmk/visible-light-communication-13982334](http://www.slideshare.net/sagineshmk/visible-light-communication-13982334)
7. [www.tomsguide.com/us/LED-Lighting-Wifi-Networking,news-12061.html](http://www.tomsguide.com/us/LED-Lighting-Wifi-Networking,news-12061.html)
8. [www.tomsguide.com/us/LED-Lighting-Wifi-Networking,news-12061.html](http://www.tomsguide.com/us/LED-Lighting-Wifi-Networking,news-12061.html)
9. [researchdesignlab.com/li-fi-rx-visible-light-communication.html](http://researchdesignlab.com/li-fi-rx-visible-light-communication.html)
10. en.wikipedia.org/wiki/MAX232
11. www.instructables.com/id/How-to-Pick-the-Best-Soldering-Iron/
12. en.wikipedia.org/wiki/Soldering\_iron