**CHAPTER 1**

**INTRODUCTION**

Eye protection is a critical part of any job. This is because your eyes are a crucial part of your body and are incredibly sensitive. Eye injuries cost companies millions of dollars each year, and the majority of these injuries could have been prevented.

The purpose of this project is to develop a simple and convenient eye protection during welding. Nearly three out of five injured welders were not wearing eye protection at the time of the accident or were wearing the wrong kind of eye protection for the job. Eye injuries alone cost more than $300 million per year in lost production time, medical expenses and worker compensation. Welding helmets are bulky and uncomfortable. So a large number of welders risk the health of their eyes while welding.

The smart welding glass with automatic shutter technology is a good solution to this problem. Welding light is detected by an LDR. When the welding light is present, the voltage variations cause shutter to turn on and the glass becomes dark. In the absence of light, which is detected by the LDR, the shutter turns off and the glass becomes transparent again. Style, comfort, and small size are all characteristics of the smart welding glass.

The smart welding glass also has a display unit which shows working time, welding time, number of welding counts, and efficiency of the welder on the worker’s android mobile phone.

**CHAPTER 2**

**BLOCK DIAGRAM AND EXPLANATION**

**2.1 BLOCK DIAGRAM**

POWER SUPPLY

LCD GLASS

WELDING LIGHT DETECTION UNIT

CONTROL AND CALCULATION UNIT

BLUETOOTH TRANSMITTER

ANDROID PHONE WITH BLUETOOTH ELECTRONIC APPLICATION

9V – 5V REGULATOR

Fig.1 Block diagram of smart welding glass

**2.2 BLOCK DIAGRAM EXPLANATION**

1. POWER SUPPLY

The circuit is driven by a 9v lithium battery.

1. WELDING LIGHT DETECTION UNIT

The presence of welding light is detected by LDR. The resistance of the LDR decreases when there is welding light.

1. LCD GLASS

The LCD glass is the unit worn by welder for eye protection. It resembles a normal sunglass.

1. CONTROL AND CALCULATION UNIT

This unit calculates welding time, working time, etc with the help of microcontroller.

1. BLUETOOTH TRANSMITTER

This unit transmits details about welding time, working time, and number of counts from glass unit to our mobile phones.

1. VOLTAGE REGULATOR

This unit is used to regulate the 9volt from battery into the required 5volt.

1. ANDROID PHONE WITH BLUETOOTH ELECTRONIC APPLICATION

This unit displays the information such as welding time, on time, number of counts, and efficiency.

**CHAPTER 3**

**CIRCUIT DIAGRAM AND EXPLANATION**

**3.1 CIRCUIT DIAGRAM**

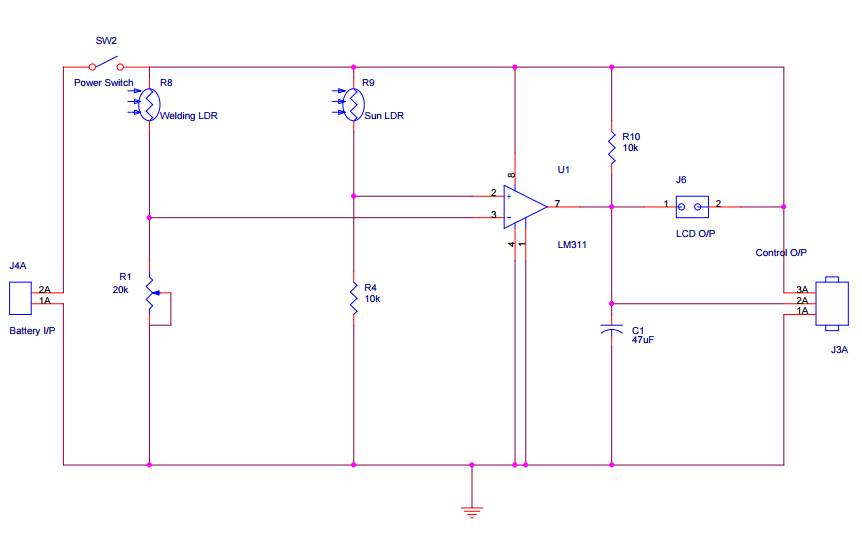


Fig.2 Circuit diagram of smart welding glass (GLASS END)

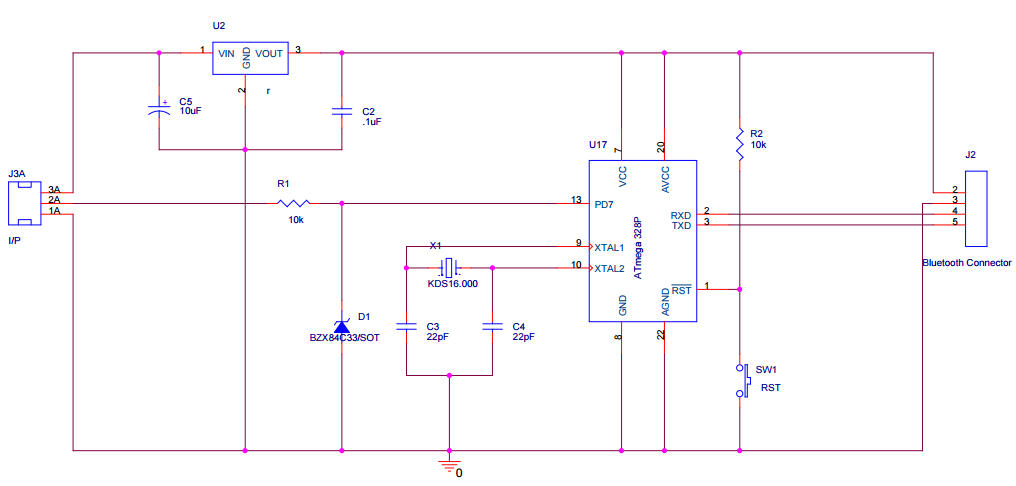


Fig.3 Circuit diagram of smart welding glass (CONTROLLER END)

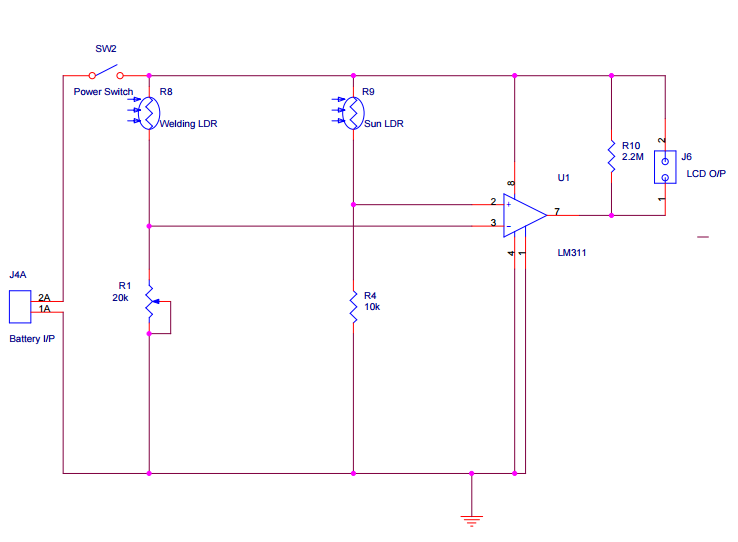


Fig. 4 Circuit diagram of smart welding glass (LIGHT MODE)

**3.2 CIRCUIT DIAGRAM EXPLANATION**

WORKING:

One of the LDRs is kept as reference, based on external light like sunlight, tube light, etc. This LDR is connected to the ‘+’ input of the comparator. Another LDR which detects the welding arc is connected to the ‘-’ input terminal of the comparator.

In the absence of welding light, the comparator output will be low. Low voltage given the display of welding glass makes it remain transparent. In the presence of welding light, the comparator will produce a high voltage output, which will turn on the display of welding glass, causing the screen to be dark.

Microcontroller will count the duration of working of welding glass, total number of welding, efficiency, etc. and send the data to the mobile phone which is connected through Bluetooth.

**CHAPTER 4**

**HARDWARE DESCRIPTION**

**4.1 Atmega 328P microcontroller**

The high-performance Microchip Pico Power 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts.By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

**4.2 Power supply and regulator**

[Lithium](https://en.wikipedia.org/wiki/Lithium_battery) 9-volt batteries are consumer-replaceable, disposable high energy density batteries. In PP3 size they are typically rated at 0.8-1.2Ah (e.g., >1.2Ah @ 900 ohms to 5.4V @ 23 °C for one type), about twice the capacity of alkaline batteries. Manufacturers claim "High energy density, up to 5x more than alkaline". Lithium PP3 batteries have a long shelf life of up to 10 years.

78xx (sometimes L78xx, LM78xx, MC78xx...) is a family of self-contained fixed [linear voltage regulator](https://en.wikipedia.org/wiki/Linear_regulator) [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuits). The 78xx family is commonly used in electronic circuits requiring a regulated power supply due to their ease-of-use and low cost. There are common configurations for 78xx ICs, including 7805 (5 V), 7806 (6 V), 7808 (8 V), 7809 (9 V), 7810 (10 V), 7812 (12 V), 7815 (15 V), 7818 (18 V), and 7824 (24 V) versions. The 7805 is the most common, as its regulated 5-volt supply provides a convenient power source for most [TTL](https://en.wikipedia.org/wiki/Transistor-transistor_logic) components.

**4.3 Crystal oscillator KDS16.000MHz**

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a precise frequency. Crystal oscillator circuit works on the principle of the inverse piezoelectric effect, i.e., a mechanical deformation is produced by applying an electric field across certain materials. Thus, it utilizes the vibrating crystal’s mechanical resonance which is made of a piezoelectric material for generating an [electrical signal](http://www.edgefx.in/solar-charge-controller-circuit-using-microcontroller/) of a specific frequency. These quartz crystal oscillators are highly stable, consists of good quality factor, they are small in size, and are very economical. Hence, quartz crystal oscillator circuits are superior compared to other resonators such as LC circuits, turning forks, and so on. For example, 16MHz crystal oscillator (KDS16.000MHz) is used in [microcontroller](http://www.edgefx.in/difference-between-microprocessor-and-microcontroller/) ATmega 328P.

**4.4 Bluetooth Connector**

Bluetooth is a telecommunications industry specification that describes how mobile devices, computers and other devices can easily communicate with each other using a short-range [wireless](http://searchmobilecomputing.techtarget.com/definition/wireless) connection.

**4.5 Zener diode (BZX84C33/SOT)**

A Zener diode is a particular type of [diode](https://en.wikipedia.org/wiki/Diode) that, unlike a normal one, allows current to flow not only from its anode to its cathode, but also in the reverse direction, when the so-called "Zener voltage" is reached. Zener diodes have a highly doped [p-n junction](https://en.wikipedia.org/wiki/P-n_junction). Normal diodes will also [break down](https://en.wikipedia.org/wiki/Breakdown_voltage) with a reverse voltage but the voltage and sharpness of the knee are not as well defined as for a Zener diode. Also normal diodes are not designed to operate in the breakdown region, but Zener diodes can reliably operate in this region. ITT BZX84C33 SOT-23 33V 0.35W Zener diode is used in the circuit. The BZX84C Series is 350 mW 16 V 250 mA Surface Mount Zener Diode - SOT-23-3 package. It has an Operating Junction Temperature of -55 to +150 °C  
Power Dissipation is 350 mW.

**4.6 Resistors**

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. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, [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.

**4.7 Capacitors**

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) that stores electrical [energy](https://en.wikipedia.org/wiki/Energy) in an [electric field](https://en.wikipedia.org/wiki/Electric_field). The effect of a capacitor is known as [capacitance](https://en.wikipedia.org/wiki/Capacitance). While capacitance exists between any two electrical conductors of a circuit in sufficiently close proximity, a capacitor is specifically designed to provide and enhance this effect for a variety of practical applications by consideration of size, shape, and positioning of closely spaced conductors, and the intervening [dielectric](https://en.wikipedia.org/wiki/Dielectric) material.

**4.8 Transistor BC547**

BC547 is an NPN bi-polar junction transistor. A transistor, stands for transfer of resistance, is commonly used to amplify current. A small current at its base controls a larger current at collector & emitter terminals.

BC547 is mainly used for amplification and switching purposes. It has a maximum current gain of 800. Its equivalent transistors are BC548 and BC549. The transistor terminals require a fixed DC voltage to operate in the desired region of its characteristic curves. This is known as the biasing. 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. BC547 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 off.

**4.9 Light Dependent Resistor (LDR)**

A Light Dependent Resistor (LDR) or a photo resistor is a device whose [resistivity](https://www.electrical4u.com/electrical-resistance-and-laws-of-resistance/) is a function of the incident electromagnetic radiation. Hence, they are light sensitive devices. They are also called as photo conductors, photo conductive cells or simply photocells. They are made up of [semiconductor](https://www.electrical4u.com/theory-of-semiconductor/) materials having high resistance. A light dependent resistor works on the principle of photo conductivity. Photo conductivity is an optical phenomenon in which the materials conductivity is increased when light is absorbed by the material. When light falls i.e. when the photons fall on the device, the electrons in the valence band of the semiconductor material are excited to the conduction band. These photons in the incident light should have energy greater than the band gap of the semiconductor material to make the electrons jump from the valence band to the conduction band. Hence when light having enough energy strikes on the device, more and more electrons are excited to the conduction band which results in large number of [charge carriers](https://www.electrical4u.com/mobility-of-charge-carrier/). The result of this process is more and more [current](https://www.electrical4u.com/electric-current-and-theory-of-electricity/) starts flowing through the device when the circuit is closed and hence it is said that the [resistance](https://www.electrical4u.com/electrical-resistance-and-laws-of-resistance/) of the device has been decreased.

**4.10 LM311**

The LM111, LM211, and LM311 devices are single high-speed voltage comparators. These devices are designed to operate from a wide range of power-supply voltages, including ±15-V supplies for operational amplifiers and 5-V supplies for logic systems. The output levels are compatible with most TTL and MOS circuits. These comparators are capable of driving lamps or relays and switching voltages up to 50 V at 50 mA. All inputs and outputs can be isolated from system ground. The outputs can drive loads referenced to ground, VCC+ or VCC−. Offset balancing and strobe capabilities are available, and the outputs can be wire-OR connected. If the strobe is low, the output is in the off state, regardless of the differential input.



Fig. 5 Model of smart welding glass

**CHAPTER 5**

**SOFTWARE DEVELOPMENT**

**5.1 EMBEDDED C PROGRAMMING LANGUAGE**

Embedded C is a set of language extensions for the [C programming language](https://en.wikipedia.org/wiki/C_(programming_language)) by the [C Standards Committee](https://en.wikipedia.org/wiki/ISO/IEC_JTC_1/SC_22) to address commonality issues that exist between C extensions for different [embedded systems](https://en.wikipedia.org/wiki/Embedded_system). Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as [fixed-point arithmetic](https://en.wikipedia.org/wiki/Fixed-point_arithmetic), multiple distinct [memory banks](https://en.wikipedia.org/wiki/Memory_bank), and basic [I/O](https://en.wikipedia.org/wiki/Input/output) operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

 Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. As it was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it the reputation as the language of choice for hackers too.

 As assembly language programs are specific to a processor, assembly language didn’t offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn’t find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C.

**EMBEDDED SYSTEMS PROGRAMMING**

Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:

·         Embedded devices have resource constraints(limited ROM, limited RAM, limited stack space, less processing power)

·         Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components. ·         Embedded systems are more tied to the hardware.

 Two salient features of Embedded Programming are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language.  Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

·           Machine Code

·           Low level language, i.e., assembly

·           High level language like C, C++, Java, Ada, etc.

·           Application level language like Visual Basic, scripts, Access, etc.

 Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

Use of C in embedded systems is driven by following advantages

·      It is small and reasonably simpler to learn, understand, program and debug.

·      C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.

·      Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.

·      As C combines functionality of assembly language and features of high level languages, C is treated as a ‘middle-level computer language’ or ‘high level assembly language’

·      It is fairly efficient

·      It supports access to I/O and provides ease of management of large embedded projects.

 Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high level languages.

 Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation.

 Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes). Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity. C is based on the philosophy ‘programmers know what they are doing’; only the intentions are to be stated explicitly. It is easier to write good code in C & convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

 Objected oriented language, C++ is not apt for developing efficient programs in resource constrained environments like embedded devices. Virtual functions & exception handling of C++ are some specific features that are not efficient in terms of space and speed in embedded systems. Sometimes C++ is used only with very few features, very much as C.

 Ada, also an object-oriented language, is different than C++. Originally designed by the U.S. DOD, it didn’t gain popularity despite being accepted as an international standard twice (Ada83 and Ada95). However, Ada language has many features that would simplify embedded software development.

 Java is another language used for embedded systems programming. It primarily finds usage in high-end mobile phones as it offers portability across systems and is also useful for browsing applications. Java programs require Java Virtual Machine (JVM), which consume lot of resources. Hence it is not used for smaller embedded devices.

 Dynamic C and B# are some proprietary languages which are also being used in embedded applications.

**5.2 PROGRAM CODE**

char BluetoothData;//  for reset switch communication

  int shutterr,shutter=7;    //shutter voltage give to this pin

  unsigned long realtime,activetimefirst,activetimeend,activetime=0;    //initialise realtime, active time as 0

  int STDcount=15, count=0,flag=0;//    No. of weldings

void setup() {

  pinMode(shutter,INPUT);//shutter detection pin is input

  Serial.begin(9600);//baudrate =9600

  }

void(\* resetFunc) (void) = 0;//initialize reset function

void loop() {

  if(Serial.available())

  {

  BluetoothData=Serial.read();

  if(BluetoothData=='R')

  resetFunc();//reset if reset butten is pressed

  }

  shutterr=digitalRead(shutter);//read shutter pin is weather high or low,ie. shutter is ON or OFF

    while(shutterr==HIGH)

    {

       if(Serial.available())

       {

         BluetoothData=Serial.read();

            if(BluetoothData=='R')

            resetFunc();

       }

      if(flag==0)

      {

        activetimefirst=millis();//counting first instant of welding time

        count++;//count no.of weldings

      }

      flag=1;

      activetimeend=millis();//check last welding time

      shutterr=digitalRead(shutter);//check shutter input continuously

    }

    flag=0;

    activetime=activetime+activetimeend-activetimefirst;//calculate total welding time

    activetimeend=0;

    activetimefirst=0;//reset times

    int ah,am,as;

    unsigned long aover;

    ah=int(activetime/3600000);//measuring working time

    aover=activetime%3600000;

    am=int(aover/60000);

    aover=aover%60000;

    as=int(aover/1000);

    if(ah<10&&am<10&&as<10)

    Serial.print("\*A0"+String(ah)+" : 0"+String(am)+" : 0"+String(as)+"\*");//convert to string to best display

     if(ah<10&&am<10&&as>=10)

    Serial.print("\*A0"+String(ah)+" : 0"+String(am)+" : "+String(as)+"\*");

     if(ah<10&&am>=10&&as<10)

    Serial.print("\*A0"+String(ah)+" : "+String(am)+" : 0"+String(as)+"\*");

     if(ah<10&&am>=10&&as>=10)

    Serial.print("\*A0"+String(ah)+" : "+String(am)+" : "+String(as)+"\*");

     if(ah>=10&&am<10&&as<10)

    Serial.print("\*A"+String(ah)+" : 0"+String(am)+" : 0"+String(as)+"\*");

     if(ah>=10&&am<10&&as>=10)

    Serial.print("\*A"+String(ah)+" : 0"+String(am)+" : "+String(as)+"\*");

     if(ah>=10&&am>=10&&as<10)

    Serial.print("\*A"+String(ah)+" : "+String(am)+" : 0"+String(as)+"\*");

     if(ah>=10&&am>=10&&as>=10)

    Serial.print("\*A"+String(ah)+" : "+String(am)+" : "+String(as)+"\*");

    realtime=millis();//measure total ON time

    int rh,rm,rs;

    unsigned long rover;

    rh=int(realtime/3600000);//measuring total time

    rover=realtime%3600000;

    rm=int(rover/60000);

    rover=rover%60000;

    rs=int(rover/1000);

   if(rh<10&&rm<10&&rs<10)

    Serial.print("\*R0"+String(rh)+" : 0"+String(rm)+" : 0"+String(rs)+"\*");//total ON time convert to string for best disply

     if(rh<10&&rm<10&&rs>=10)

    Serial.print("\*R0"+String(rh)+" : 0"+String(rm)+" : "+String(rs)+"\*");

     if(rh<10&&rm>=10&&rs<10)

    Serial.print("\*R0"+String(rh)+" : "+String(rm)+" : 0"+String(rs)+"\*");

     if(rh<10&&rm>=10&&rs>=10)

    Serial.print("\*R0"+String(rh)+" : "+String(rm)+" : "+String(rs)+"\*");

     if(rh>=10&&rm<10&&rs<10)

    Serial.print("\*R"+String(rh)+" : 0"+String(rm)+" : 0"+String(rs)+"\*");

     if(rh>=10&&rm<10&&rs>=10)

    Serial.print("\*R"+String(rh)+" : 0"+String(rm)+" : "+String(rs)+"\*");

     if(rh>=10&&rm>=10&&rs<10)

    Serial.print("\*R"+String(rh)+" : "+String(rm)+" : 0"+String(rs)+"\*");

     if(rh>=10&&rm>=10&&rs>=10)

    Serial.print("\*R"+String(rh)+" : "+String(rm)+" : "+String(rs)+"\*");

    int efficiency=(count\*activetime\*100)/(STDcount\*realtime);//calculate efficiency

    Serial.print("\*E"+String(efficiency)+"\*");//efficiency convert to string

    if(count<10)

    Serial.print("\*C00"+String(count)+"\*");//count convert to string to best Display

     if(count>=10&&count<100)

    Serial.print("\*C0"+String(count)+"\*");

     if(count>=100)

    Serial.print("\*C"+String(count)+"\*");

}

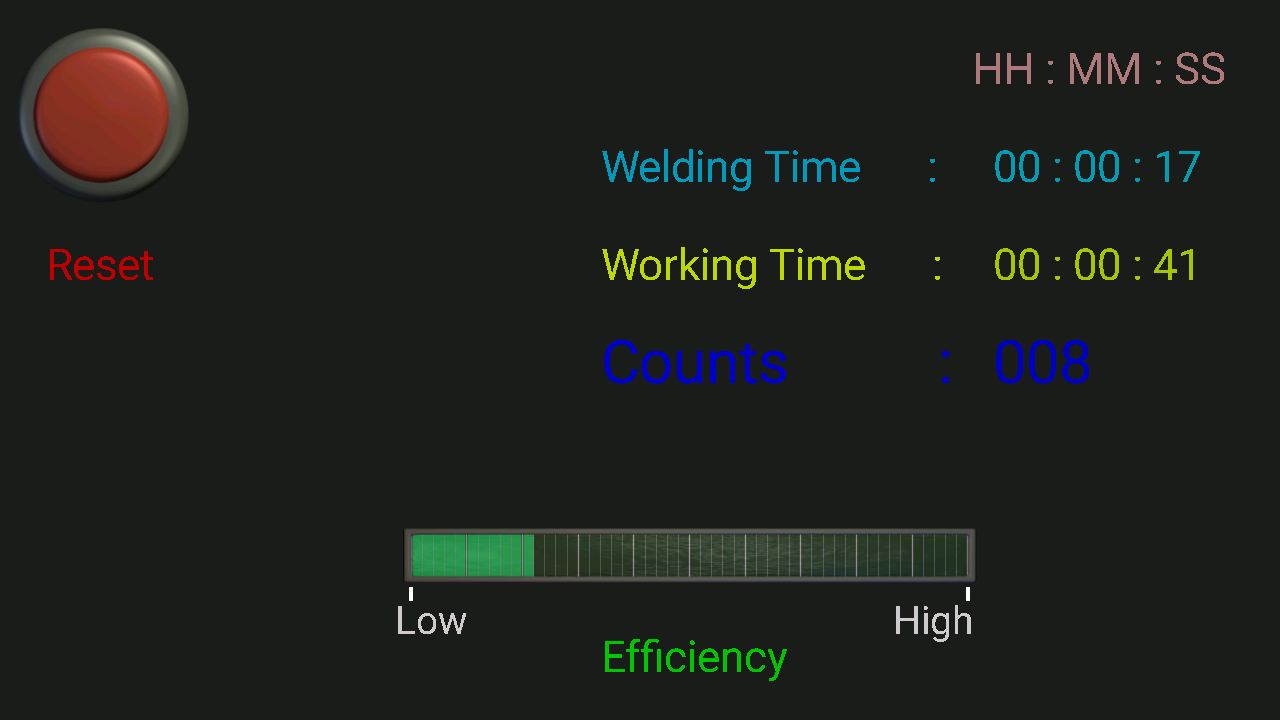


Fig.6 Display on android phone

**CHAPTER 6**

**PCB FABRICATION**

**6.1 PCB DESIGN**

It consists of two steps:

* The first step is to prepare schematic in any of PCB design software, such as Proteus, Diptrace, Escad, Smart PCB, etc. We made our schematic in Proteus. Proteus schematic capture includes many libraries with thousands of component symbol. We can select the required symbol from library and place it on the schematic page. After placing the component symbol, the inter connection is completing using bus tool.
* The second step is to prepare PCB layout either by auto routing or manual routing. We used both manual routing and auto routing. Routing is the interconnection of component using upper tracks of required width. The number of layers used and enabling the artwork depends upon the complexity of the circuit, and fabrication technology available. If the board is single sided, enable only bottom or solder side layer, so that track will come only on one side of the PCB If the circuit is much more complex the enable the required number of inner layer consider the fabrication technique and cost. In manual routing, the PCB designer has to manually connect each track. This is time consuming process, but is required some cases.

The layout is designed in such a way as to accommodate the whole circuit in minimum space, avoiding use of jumpers as far as possible. Besides the complete outlines and interconnections, the layout should include information on:

o Component hole diameter

o Conductor width

o Minimum spacing to be provided between the tracks

Finally take the print out of the pattern /layout on a normal A4 size paper. Also make sure that it is the correct size. Check the layout carefully. This printout is called as positive film. A problem we faced during the design was that, we had to design certain components for which there is no existing library. This was done in Diptrace Component editor and pattern editor.

**6.2 PCB MANUFACTURE**

PCB etching was done with the help of ferric chloride. The inherent property of

Ferric chloride to dissolve copper is utilized here.

* The layouts mirror view negative is printed and a butter paper and using this UV Ray is allowed to fall on the copper clad.
* Developer develops the image on the clad. This was then finely dyed and now all the traces are finely masked.
* Clad is now placed in ferric chloride solution, Now the unmasked copper is dissolved in the solution producing copper chloride and iron.
* Clad is washed well and holes are drilled.

The PCB (Printed Circuit Board) for the circuit is tow double - sided boards: one for display board and another is the main board. We manufactured the PCB using glass epoxies, become of its good electrical and mechanical properties. A few PCB manufacturing processes are described below.

**6.2.1 CLEANING**

The board is cleaned well with the help of steel wool. It helps to stick the photosensitive material to the board well.

**6.2.2 LAMINATION**

The copper clad lamination has to be converted with photosensitive film photo resist transferring the image on the PCB.

**6.2.3 EXPOSE**

Expose the photosensitive film resist to the UV radiation through the positive film master. So the areas where patterns are present are unexposed & the remaining photosensitive film photo resist is exposed.

**6.2.4 DEVELOPING**

Dip the PCB in 2% sodium carbonate solution for a minute. As a result the areas will not be stripped off but the remaining area, i.e. the patterns get stripped off.

**6.2.5 TIN PLATING**

Plate the tin or lead over the electroplated copper & so the deposition prevents the etching of copper.

Stopping of dry film photo resist: Immerse the development and electroplated board into a solution containing 20% NaOH. The dry film photo resist will be removed & so the tin- lead is coated on the pattern area & the unwanted copper will be visible.

**6.2.6 ETCHING**

For etching the copper, which is present in the unwanted area, we need to use cuprous chloride as etchant so the copper present in the unwanted area are removed.

Stripping of the fin coating: For stripping the tin lead, which is presenting the unwanted area, we need to use a solution containing acetic acid and hydrogen peroxide. If we leave the tin – lead as it during the wave soldering operation, due to high temperature tin lead will meet and so and the soldering mask will have full finish. Immerse the panel in to a tank coating tin lead stripper until the tin or tin lead is completely dissolved take out and rinse with water.

**6.2.7 DRILLING**

Drilling is carried on using CNC drilling machine or manually. Here manual drilling is used.

**6.2.8 SOLDERING**

A variety of soldering techniques such as pip soldering, wave soldering, reflow soldering, manual soldering, etc. are used in an industry. Here manual soldering is used. There are two main classifications for the methods of soldering in use today: mechanical or non-electrical and electrical. Soldering is a method of making a permanent electrical and mechanical connection between metals. There are many different processes utilized in soldering, virtually all of them involve four basic elements: base metals, flux, solder, and heat. A base material is any metal that contacts the solder and forms an intermediate alloy. Flux is often applied as a liquid to the surface of the base metals prior to soldering. The first and primary purpose of flux is to stop the base metals from oxidizing while they are being heated to the soldering temperature. The flux covers the surface to be soldered, shielding it from oxygen and thereby preventing oxidization during heating. Solder is a metal or metallic alloy used, when melted, to join metallic surfaces together. The most common alloy is some combination of tin and lead. In any soldering, some form of flux is necessary to cause the solder to flow into the cleaned joint. Ant flux is necessary if it is important to keep the solder out of some region closely adjacent. Wave soldering and reflow soldering are the two major types of assembly process. Wave soldering is often preferred when through hole and surface mounted components are mixed on one printed circuit board. However, wave soldering is not always suitable for surface mount ICs, or for printed circuits with high population densities. In these situation reflow soldering is often used.

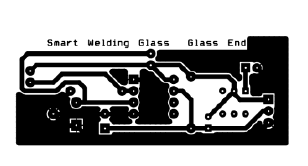


Fig.7 Glass end layout

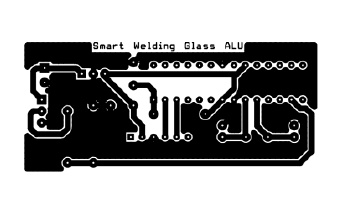


Fig.8 Controller end layout

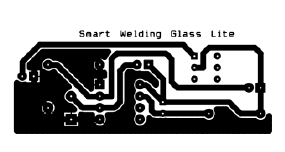


Fig.9 Light mode layout

**CHAPTER 7**

**ADVANTAGES AND LIMITATIONS**

* Smart welding glass can be used to protect the eyes from flash burn, [ultraviolet light](https://en.wikipedia.org/wiki/Ultraviolet_light), [sparks](https://en.wikipedia.org/wiki/Spark_(fire)), [infrared light](https://en.wikipedia.org/wiki/Infrared_light), and heat.
* They are necessary to prevent [arc eye](https://en.wikipedia.org/wiki/Arc_eye), a painful condition where the [cornea](https://en.wikipedia.org/wiki/Cornea) is inflamed.
* Smart welding glass can also prevent [retina](https://en.wikipedia.org/wiki/Retina) burns, which can lead to a loss of vision. Both conditions are caused by unprotected exposure to the highly concentrated ultraviolet and infrared rays emitted by the welding arc.
* However, this smart welding glass only provides protection to the eyes during welding. Skin on the face and hands are also in danger while welding and must be given protection.
* The display unit will not be useful in case the welder does not own an android mobile phone.

**CHAPTER 8**

**CONCLUSION AND FUTURE SCOPE**

A smart welding glass, along with calculation and display unit, was made using LDRs, comparator IC LM311, microprocessor ATmega328P, and Bluetooth module.

This glass can be further modified to operate in several modes, such as welding mode\sunglasses mode.

**CHAPTER 9**

**REFERENCES**

[www.wikepedia.com](http://www.wikepedia.com)

[www.keuwl.com](http://www.keuwl.com)

bluetoothelectronics application

Arduino IDE application

**CHAPTER 10**

**APPENDIX**