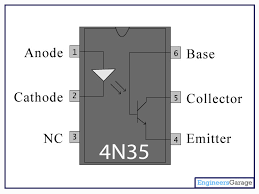
***DAY 1***

* This is Optocoupler, Transistor output,1 channel,DIP,6 pins,50 mA,5 kV.
* Manufacturer : VISHAY
* Cost : 12.09 per piece2
* Meaning of optocoupler
* It is also known as opto-isolator
* This type of components are used to transmit signal or data from one part (sub-system) of system to any other part( sub-system) of system.
* These types of components are used when source and destination are at very different level voltage.
* e.g. microprocessor is operate on 5 V but being used to control a triac which is switched at 240V AC.
* Each of 6 optocoupler consists og gallium arsenide infrares LED and silicon NPN phototransistor.
* This optocoupler is widely used at
* AC mains detection
* Reed relay driving
* Switch mode power supply
* Logic ground isolation

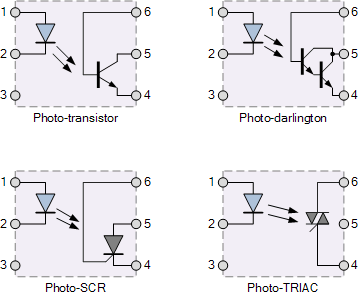
***DAY 2***

* 4N35 is an optocoupler integrated circuit in which an infrared emitter diode drives a phototransistor.
* It is used to on and off another circuit using first circuit without electrical connection.
* The first circuit is connected to IR diode and another circuit is connected to phototransistor.
* Working of both relay and optocoupler is same, the difference is that in relay both circuit is connected by magnetically means and in optocoupler
* both circuit is connected with optical means.
* Optocoupler is smaller and faster than relay.

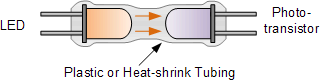
***DAY 3***



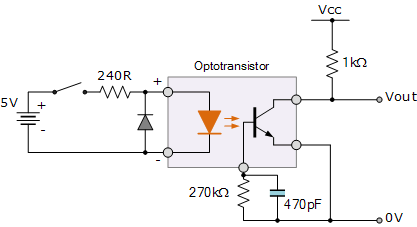
* Current from the source signal passes through the input LED which emits an infra-red light whose intensity is proportional to the electrical signal.
* This emitted light falls upon the base of the photo-transistor, causing it to switch-ON and conduct in a similar way to a normal bipolar transistor.
* The base connection of the photo-transistor can be left open (unconnected) for maximum sensitivity to the LEDs infrared light energy or connected to ground via a suitable external high value resistor to control the switching sensitivity making it more stable and resistant to false triggering by external electrical noise or voltage transients.
* When the current flowing through the LED is interrupted, the infrared emitted light is cut-off, causing the photo-transistor to cease conducting.
* Optocoupler Type



* The photo-transistor and photo-darlington devices are mainly for use in DC circuits while the photo-SCR and photo-triac allow AC powered circuits to be controlled.
* Homemade optocoupler:
* Simple homemade optocouplers can be constructed by using individual components. A Led and a photo-transistor are inserted into a rigid plastic tube or encased in heat-shrinkable tubing as shown.
* Tubing with a reflective inner would be more efficient than dark black tubing.
* Common Application:
* Microprocessor input and output switching
* Power supply regulation



* An Optotransistor DC Switch



* As well as detecting DC signals and data, Opto-triac isolators are also available which allow AC powered equipment and mains lamps to be controlled. Opto-coupled triacs such as the MOC 3020, have voltage ratings of about 400 volts making them ideal for direct mains connection and a maximum current of about 100mA. For higher powered loads, the opto-triac may be used to provide the gate pulse to another larger triac via a current limiting resistor as shown.

### Triac Optocoupler Application

### opto6.gif

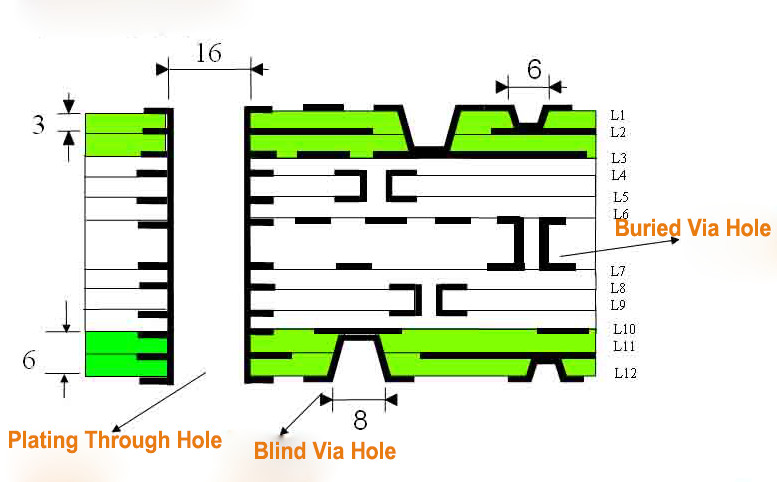
* This type of optocoupler configuration forms the basis of a very simple solid state relay application which can be used to control any AC mains powered load such as lamps and motors. Also unlike a thyristor (SCR), a triac is capable of conducting in both halves of the mains AC cycle with zero-crossing detection allowing the load to receive full power without the heavy inrush currents when switching inductive loads.
* **Optocouplers** and **Opto-isolators** are great electronic devices that allow devices such as power transistors and triacs to be controlled from a PC’s output port, digital switch or from a low voltage data signal such as that from a logic gate. The main advantage of opto-couplers is their high electrical isolation between the input and output terminals allowing relatively small digital signals to control much large AC voltages, currents and power.
* An optocoupler can be used with both DC and AC signals with optocouplers utilizing a SCR (thyristor) or triac as the photo-detecting device are primarily designed for AC power-control applications. The main advantage of photo-SCRs and photo-triacs is the complete isolation from any noise or voltage spikes present on the AC power supply line as well as zero-crossing detection of the sinusoidal waveform which reduces switching and inrush currents protecting any power semiconductors used from thermal stress and shock.
* Using 4N25: <https://www.instructables.com/id/How-to-Use-an-Optocoupler-to-Pass-Signals-Between-/>
* <http://www.arunet.co.uk/tkboyd/ec/ec1optoiso.htm>
* <https://www.renesas.com/en-in/products/optoelectronics/technology/usage.html>

***DAY 4***

* A **printed circuit board** (**PCB**) mechanically supports and electrically connects [electronic components](https://en.wikipedia.org/wiki/Electronic_components) or [electrical](https://en.wikipedia.org/wiki/Electrical) components using [conductive](https://en.wikipedia.org/wiki/Electrical_conductor) tracks, pads and other features [etched](https://en.wikipedia.org/wiki/Industrial_etching) from one or more sheet layers of copper [laminated](https://en.wikipedia.org/wiki/Laminated) onto and/or between sheet layers of a [non-conductive](https://en.wikipedia.org/wiki/Insulator_(electricity)) [substrate](https://en.wikipedia.org/wiki/Substrate_(electronics)).
* PCBs require additional design effort to lay out the circuit, but manufacturing and assembly can be automated.
* Specialized CAD software is available to do much of the work of layout. Like DipTrace, Eagle.
* Why PCBs
* Mass-producing circuits with PCBs is cheaper and faster than with other wiring methods, as components are mounted and wired in one operation.
* Large numbers of PCBs can be fabricated at the same time, and the layout only has to be done once.
* PCBs can also be made manually in small quantities, with reduced benefits.

***DAY 5***

* PCB Design Course
* Common Tips:
* Keep font in True Type.
* 1 inch=2.54 cm
* We can align components in rows or in column using Align Objects in right click submenu.



* DipTrace Schematic:
* Net is group of wires whose names are same.
* In component library all standard libraries are available.
* In User library we have to add library only for one time which we used repeatedly.
* We can also add library in user component list which are not available in component library using “Add from File”.
* In Project library all library used in current project are saw.
* We can also add new grid size using customized grid.
* We can also search component based on name, RefDef , value, pattern, manufacturer, datasheet, etc.
* To set one type of component in bulk quantity then we can use “Edit\Copy Matrix” function.
* The tool to move RefDef, pin number etc. things is in “View\Part Marking\Move Tool”.
* We can use Undo and Redo. DipTrace saves up to 50 steps.
* We can Find ohm, lemda like symbol in “Character Map”.
* To identify IC’s pin no. 1, all pins are denoted by circle except pin no. 1. Pin no. 1 is denoted by square symbol.
* Like this battery’s positive leg is denoted by square and negative leg is denoted by circle.
* After making Schematic and before going to PCB Layout, we must check that every component has Pattern, if not then select Attached pattern from right click submenu.
* To printout Schematic we have drag and drop our whole circuit in printing area (a table).
* We can use PDF printer (downloaded) for printing purpose.
* Use “File\Preview” option to save schematic as jpg or bmp format.
* To connect two things without wire then select anyone thing and right click to select “Add to Net”.
* DipTrace PCB Layout:
* Start PCB Layout work with Route menu, after that also pour copper layer.
* Then gone through menu “Verification”.
* In the PCB Layout ratlines are used to identify which pin of component is connected to which pin of another component.
* We can use auto arrangement and auto routing in DipTrace in Placement toolbar.
* The tool to move RefDef, pin number etc. things is in “View\Component Marking\Move Tool”.
* To put ratline: Object/ratlines
* To delete ratline: right click and delete from net or delete net
* If we change anything in Schematic then we can reflect its effect in PCB Layout using File toolbar.
* There is some part on PCB where we do not have to pour copper then we can set it using Route Keepout.
* Traces are drawn usually on bottom layer.
* We can change width and clearance of trace and diameter of vias using “Route/Route Setup”.
* All signals are divided into two basic type : Signal and Non-signal layer
* Top/Bottom Silk are silkscreen layers, all texts and graphical information are automatically added there
* Top/Bottom Mask and Paste layers carry information about solder mask and solder paste application zones.
* Traces and copper pours can be created only on signal layers.
* There are two types of signal layer: Signal and Plane
* Signal layer contains traces and copper pours
* Plane layers are inner layer, which contain one or several copper pours.
* We can also mirror the circuit using “View\mirror”. However this is not necessary, because Gerber export automatically creates the correct copper layout on the bottom layer.
* At higher level PCB traces have resistance, inductance, and capacitance, just like your circuit does.
* We give Gerber file to manufacturer with N/C Drill file.
* Another way to this is using Titles and Sheet Setup, Preview.
* We can also make list of component in Excel using “Object\Creat Bill of Material”
* Via is something which connects layers of PCB.
* There are three types of vias,
* Through Hole
* Blind via Hole
* Buried via Hole
* In Through Hole types of vias are made by using drill to make full borehole.
* If we want to select particular thinga then you can do it using “Edit\Edit Selected”.
* We can add logo in BMP, DXF, or JPEG etc. formats.
* Red circle means clearance error and magenta circle means size error.
* DipTrace Pattern Editor:
* If you know that the pattern is already available in the standard pattern library then you can directly start from component Editor to draw the schematic symbol.
* DipTrace Component Editor:
* In DipTrace there is three “roop” of any component,
* Schematic symbol is made by Component Editor and used by Schematic Capture.
* Pattern drawing (footprint) is made by Pattern Editor and used by PCB Layout.
* We cannot make 3D model, and we can use 3D model for 3D visualisation.
* <https://diptrace.com/diptrace-software/guided-tour/>

***DAY 6***

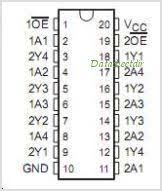
* Description : Buffers and line Drivers tri-state octal
* Max. Supply voltage : 6 V
* Min. Supply voltage : 2 V
* Typical operating supply voltage : 5V
* Mounting style :Through Hole
* Output states : Logic 1

: Logic 0

: High impedance

* This SN74HC244N is designed to improve both the performance and density of 3-state memory driver, clock drivers and bus-oriented receivers and transmitters.
* In this there are 2 active low output enable and 0 input enable.
* Application
* Clock & timing
* Signal Processing





***DAY 7***

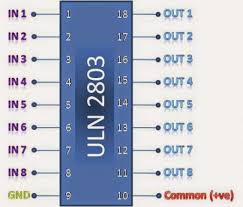


* Brand name: Delson Electronics
* Uses of pull-up resistor are to no and off the switch.
* Part number: r237b\_48
* It is 1000 ohm resistors.
* Each resistor array has a dot or block at one end to inform that that lid is common ground.
* A resistor array is convenient when we need several resistors which will all have one end connected to the same point in a circuit.
* A resistor array is especially convenient to replace all of the resistors for a DIP switch.
* We use A102J in place of individual array due to its less weight and size.

***DAY 8***

* The ULN2803APG/AFWG series are high voltage and high current darlington drivers comprised with 8 NPN darlington transistor pair.
* Application:
* Relay
* Hammer
* Display(LED) driver
* Lamp Drivers
* Line Drivers
* Logic Buffers
* Stepper Motors
* IP Camera
* HVAC Valve and LED Dot Matrix
* Features:
* Output voltage : 50V max
* Input Voltage : 30V max
* Output current : 500 mA





***DAY 9***

******

* Description: Power supply control with built in Watchdog timer.
* What is Packaging?
* In electronics manufacturing, IC packaging is the final stage of semiconductor device fabrication, in which the tiny block of semiconducting material is encapsulated in a supporting case that prevents physical demage and corrosion.
* What is Watchdog timer?
* A watchdog timer is an electronic timer that is used to detect and recover from computer malfunction (this is state of something in which that thing working wrongly or does not work at all).
* During normal operation, the computer regularly reset the watchdog timer to prevent it from elapsing or “time out”.
* If due to hardware failure or program error, the computer fails to reset the watchdog time, the timer elapsing and generate a timeout signal
* This timeout signal is used to initiate actions.
* These initiate actions typically include placing the computer system in safe state and restoring normal system operation.
* But why Watchdog timer, we can’t do it manually?
* These timers are used where human cannot easily access the equipment or would be unable to react to fault in timely manner.
* The system in which there cannot be any manual monitoring such as control panel at terrace, the computer cannot depend on a human to invoke a reboot, if it hangs; it must be self-reliant.
* In1232N is designed to monitor power supply within the system of reset signal generation for microprocessors.
* The IN1232N micromonitor chip monitors three vital conditions for microprocessor.

1. Precision temperature-compensated reference and comparator circuit monitors the status of VCC.

* When out-of-tolerance condition occurs, an internal power failure signal is generated which forces reset to the active state.
* When VCC falls below a present level as defined by TOL, the VCC comparator output the signal reset.TOL is connected to VCC. And as VCC falls below 4.5V reset signal is activated.
* When VCC returns to an in-tolerance state, the reset signals are kept in the active state of a minimum of 250 ms to allow the power supply and processor to stabilize.

1. It also performs push button reset control.

* The IN1232 provide an input direct pin for direct connection to the push button.
* When PBRST pin goes low reset signal is generated.
* And internally this input is de-bounced and timed such that RST signal of at least 250 ms minimum is generated.

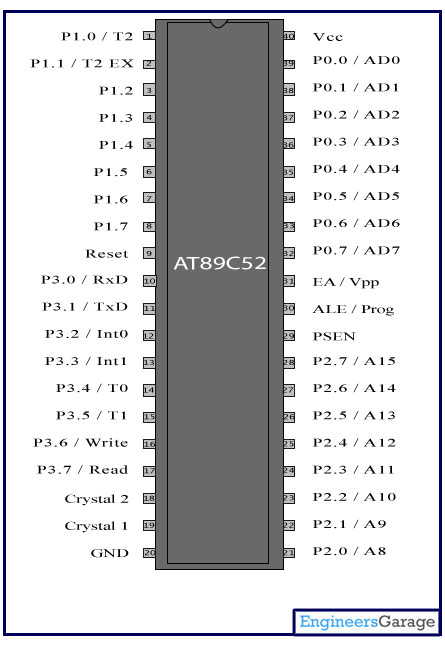
1. The third function is Watchdog timer.

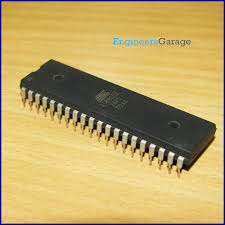
***DAY 10***

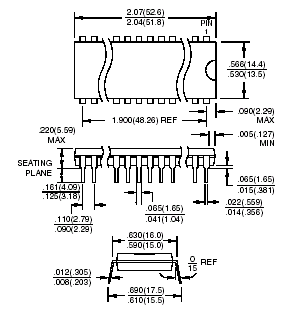
* The IN1232 has internal timer that forces the reset signals to the active state if the strobe input is not driven low prior to time-out.
* The time delay is set by the TD input to be typically
* 150 ms with TD connected to GND
* 600 ms with TD left unconnected
* 1.2 s with TD connected to VCC
* If high to low transition occurs on the ST pin prior to time-out, then the Watchdog timer is reset and begins from zero.
* If the watchdog timer is allowed to time-out, then the RST and RST signals are driven to the active state for 250 ms minimum.
* The ST input can be derived from microprocessor address signals, data signals, and/or control signals.
* Features
* Halts and restarts an out-of-control microprocessor
* Automatically restart microprocessor after power failure
* Accurate 5% or 10% microprocessor power supply monitoring
* Elimination the need of discrete components
* Supply voltage: 5V

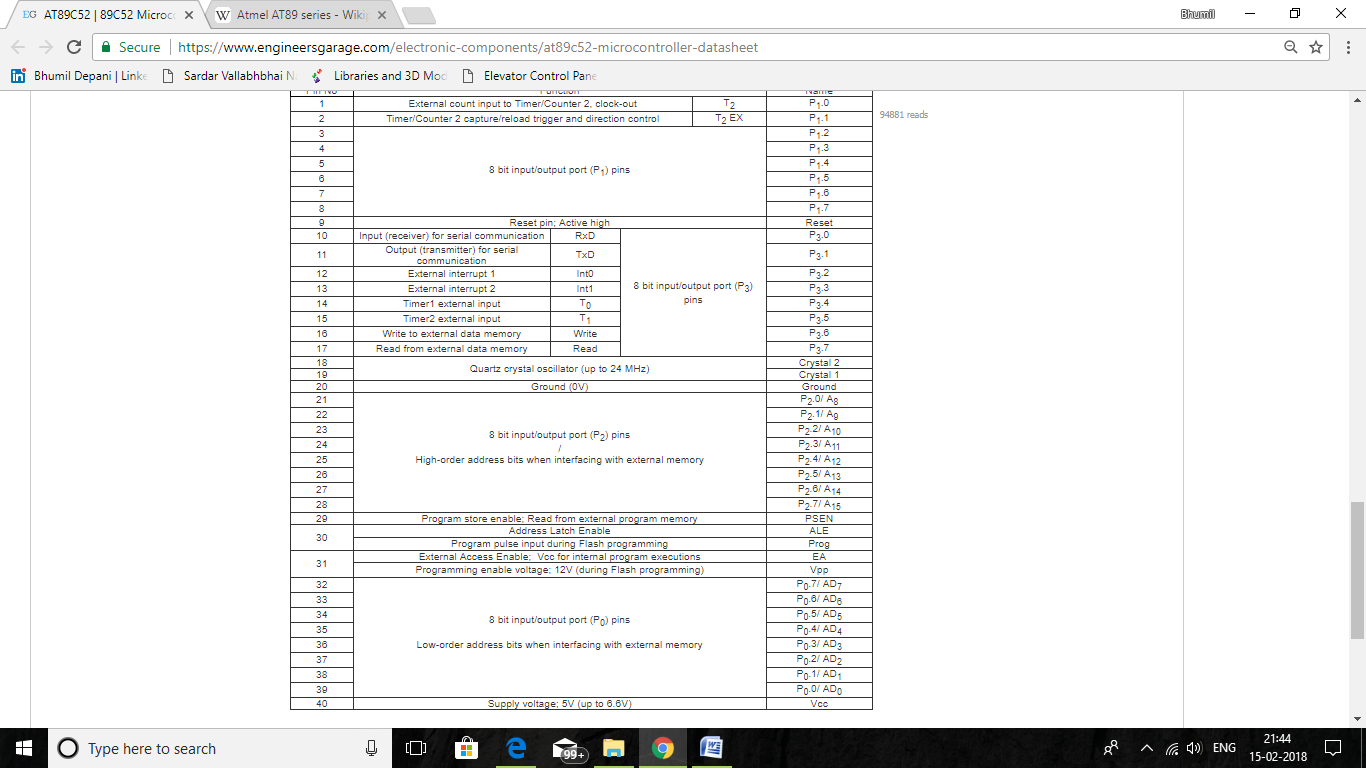
***DAY 11***

* **AT89C52**is an 8-bit [microcontroller](http://www.engineersgarage.com/microcontroller) and belongs to Atmel's [8051 family](http://www.engineersgarage.com/8051-microcontroller).
* The most commonly used set of microcontrollers belong to **8051** Family.
* **8051 Microcontrollers** continue to remain a preferred choice for a vast community of hobbyists and professionals.
* Intel fabricated the original **8051** which is known as MCS-51.
* The other two members of the 8051 family are:
  + 1. 8052
    2. 8031
* 8K Bytes of In-System Reprogrammable Flash Memory(this is not for AT89C51-52)
* Fully Static Operation: 0 Hz to 24 MHz
* Three-level Program Memory Lock
* 256 x 8-bit Internal RAM
* 32 Programmable I/O Lines
* Two 16-bit Timer/Counters
* Eight Interrupt Sources
* Programmable Serial Channel
* Low-power Idle and Power-down Modes
* 40-pin DIP



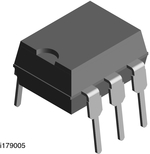






***DAY 12***

* 4N33
* Very high current transfer ratio, 500 % min
* High isolation resistance, 1011 Ω typical
* Standard plastic DIP package



***DAY 13***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | rate | Quantity | Amount |
| GCB | 50 | 1 | 50 |
| AT89C52 | 80 | 1 | 80 |
| Soldier Wire | 250 | 1 | 250 |
| 7805 | 15 | 3 | 45 |
| 4N35 | 15 | 5 | 75 |
| 4N33 | 15 | 5 | 75 |
| 74LS244N | 30 | 3 | 90 |
| ULN2803 | 25 | 2 | 50 |
| Other | 155 | 1 | 155 |
| Total: | 870 | Grand Total: | 870 |

***DAY 14***

* How to make PCB at home: <https://youtu.be/q6iu8Fe8UOY>

***DAY 15***

* Keil Software for coding was downloaded.
* Output of optocoupler is from 4 and 5 pin.
* Current will flow from 5 to 4.
* Input LED’s voltage drop is 0.8 to 1.5 V (1.3 V).
* Current at input side is 60 mA max.
* As instructable says , we set current 40 mA.
* From voltage and current we can find value of required resistor, which comes out to be 267.5Ω.
* As circuit says, if resistance is of value 1 kΩ then current is 10 mA.
* In PCB there is 32 4N35 so, there is also four A102J required.
* There are three sensors at every floor.
* One at 18 inch below level, which is called reed sensor, second is at 18 inch upper level, which also called reed sensor, other at levelled floor, which is called level sensor.
* When microprocessor card get reed signal it gives second speed as output to VFD, and getting this second speed VFD lows frequency from 50 Hz to 8 Hz.
* As floor coming nearer frequency output from VFD goes to 0 and lift stop.

***DAY 16***

* Both side of optocoupler has its own GND, so there are two GNDs in this situation.
* 4N35 IC work properly in negative logic.
* At input side of optocoupler we want 1 kΩ resistance, so we used a102J.
* We used four tri-state buffer IC, reason you know.
* Found a source from which we can learn embedded programming, link is
* <https://www.youtube.com/playlist?list=PLKbSRxrdxkT3sRzWE465KoxOH00BFbRs6>
* <https://www.youtube.com/watch?v=XP8KCgwBSkE&list=PLRuRKN7_FVgtdXWWM-uosWYfmOFDp63dd>

***DAY 17***

* <https://www.youtube.com/watch?v=JEMoXnuHaA0>

***DAY 18***

* Programming stuffs for AT89C52
* There are four types of ports there in PCs,

1. USB port: Short for UniversalSerialBus, an external bus standard that supports data transfer rates of 12 Mbps. A single USB port can be used to connect up to 127 peripheral devices, such as mice, modems, and keyboards. USB also supports Plug-and-Play installation and hot plugging. It is the standard interface for simplified connections between computers and external devices. It is used to connect USB devices to a host, and it has wide compatibility with most platforms, operating systems and external devices including printers, scanners, modems, cameras and portable storage units.

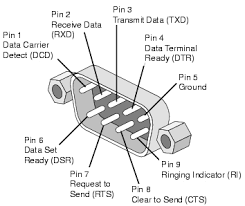


1. HDMI port: **High Definition Multimedia Interface**In the past decade, high-definition broadcasts became the new standard of what it means to be high quality. Unlike VGA and DVI, HDMI sends both video and audio signals together. The signals are digital only; thus, HDMI is only compatible with newer devices.



1. Serial port rs232 : RS-232  is a standard communication protocol for linking computer and its peripheral devices to allow **serial** data exchange. In simple terms RS 232 defines the voltage for the path used for data exchange between the devices. The main **difference between** a **serial port** and a parallel **port** is that a **serial port** transmits data one bit after another, while a parallel **port** transmits all 8 bits of a byte in parallel. Computers have both **serial** and parallel **ports** along with newer technology called a **USB** (Universal **Serial** Bus) **port**.\





1. VGA port: A [**Video Graphics Array**](https://en.wikipedia.org/wiki/Video_Graphics_Array) (**VGA**) **connector** is a three-row 15-pin [DE-15](https://en.wikipedia.org/wiki/D-subminiature) connector. The 15-pin VGA connector was provided on many video cards, computer monitors, laptop computers, projectors, and [high definition television](https://en.wikipedia.org/wiki/High_definition_television) sets. On laptop computers or other small devices, a [mini-VGA](https://en.wikipedia.org/wiki/Mini-VGA) port was sometimes used in place of the full-sized VGA connector. But now a days HDMI port is overcome by VGA port.



* 8051 has a in-built UART(p) chip
* We can bifurcate a communication methods mainly in two groups,

1. Universal Asynchronous Receiver Transmitter(UART)(g)
2. Universal Synchronous Asynchronous Receiver Transmitter(USART)

* We can also bifurcate a communication methods mainly in two groups,

1. Serial communication
2. Parallel communication

* Being a group name, UART(g), is also a protocol for communication
* UART(p) is serial type communication
* In our AT89C52 we are going to use UART(p)
* This protocol is used in old microcontroller to communicate with PC, GSM(Global System for Mobile communication), XBEE, RFID Tag.
* Three method of UART(p) communication

1. Simplex
2. Half duplex
3. Full duplex

* Start bit is LOW and stop bit is HIGH
* UART(p) is responsible for transmitting and receiving data
* RS-232 is a set of rules that specifies voltage levels in serial communication
* In telecommunications, RS-232 is a standard for serial communication transmission of data. It formally defines the signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data circuit-terminating equipment, originally defined as data communication equipment), such as a modem. The RS-232 standard is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pinout of connectors. The current version of the standard is TIA-232-F Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange, issued in 1997. An RS-232 serial port was once a standard feature of a personal computer, used for connections to modems, printers, mice, data storage, uninterruptible power supplies, and other peripheral devices.
* However, RS-232 is hampered by low transmission speed, large voltage swing, and large standard connectors.
* In modern personal computers, USB has displaced RS-232 from most of its peripheral interface roles.
* Many computers do not come equipped with RS-232 ports and must use either an external USB-to-RS-232 converter or an internal expansion card with one or more serial ports to connect to RS-232 peripherals



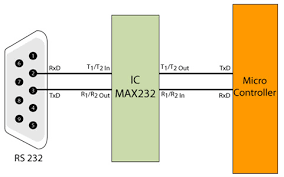
* It’s only work is to use USB as Serial Port

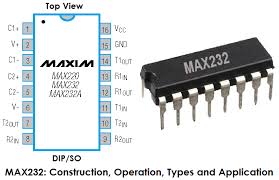


* <http://www.edgefxkits.com/blog/max232ic-and-interfacing-needs/>
* <https://www.quora.com/Microcontrollers-What-pins-are-used-to-upload-a-program-to-a-8051-IC>

***DAY19***

* RS-232 voltage levels
* For logic ‘0’..............High...................+25 to +3
* For logic ‘1’..............LOW...................-3 to -25
* This voltages come from PC
* But our microcontroller is works on 5V.
* So to converts the signals from the RS232 serial port to the proper signal which are used in the TTL compatible digital logic circuits we would use MAX-232 IC.
* The **MAX232** can convert the signals like RX, TX, CTS, and RTS and it is a dual driver/receiver.

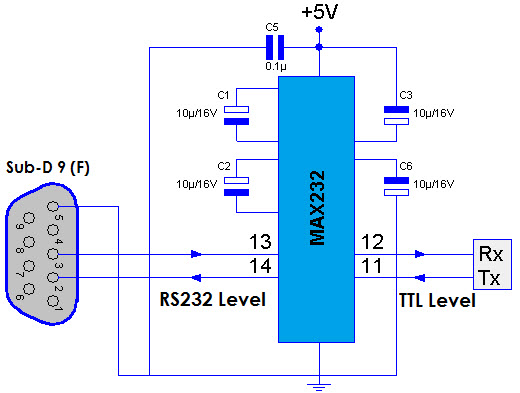




* The MAX-232 IC is an [integrated circuit](http://www.edgefxkits.com/blog/relay-driver-circuit-using-uln2003-ic/) which consists of 16 pins and it is a resourceful IC mostly used in the voltage level signal problems.
* Generally, the MAX-232 IC is used in the RS232 communication system for the conversion of voltage levels on TTL devices that are interfaced with the PC serial port and the Microcontroller.

|  |  |  |
| --- | --- | --- |
| S. No | Name | Function |
| 1 | C1 + | Positive Voltage Multiplier Unit of External Capacitor |
| 2 | C3 + | Positive Voltage Multiplier Unit of External Capacitor |
| 3 | C1- | Negative Voltage Multiplier unit of External capacitor |
| 4 | C2 + | Positive Voltage Multiplier Unit of External Capacitor |
| 5 | C2 – | Negative Voltage Multiplier unit of External capacitor |
| 6 | C4- | Negative Voltage Multiplier unit of External capacitor |
| 7 | T2 Out | Transmitter data Output from RS232 |
| 8 | R2 In | Receiver Data Input from RS232 |
| 9 | R2 Out | Receiver Data Output from TTL logic level |
| 10 | T2 In | Transmitter Data Input from TTL logic level |
| 11 | T1 In | Transmitter Data Input from TTL logic level |
| 12 | R1 Out | Output of Receiver Data from TTL logic level |
| 13 | R1 In | Input of Receiver Data from RS232 |
| 14 | T1 Out | Transmitter Output from RS232 |
| 15 | GND | Ground |
| 16 | Vcc | Power supply voltage |

* The following diagram gives the information about the PC serial port communication through the MAX232 interface.



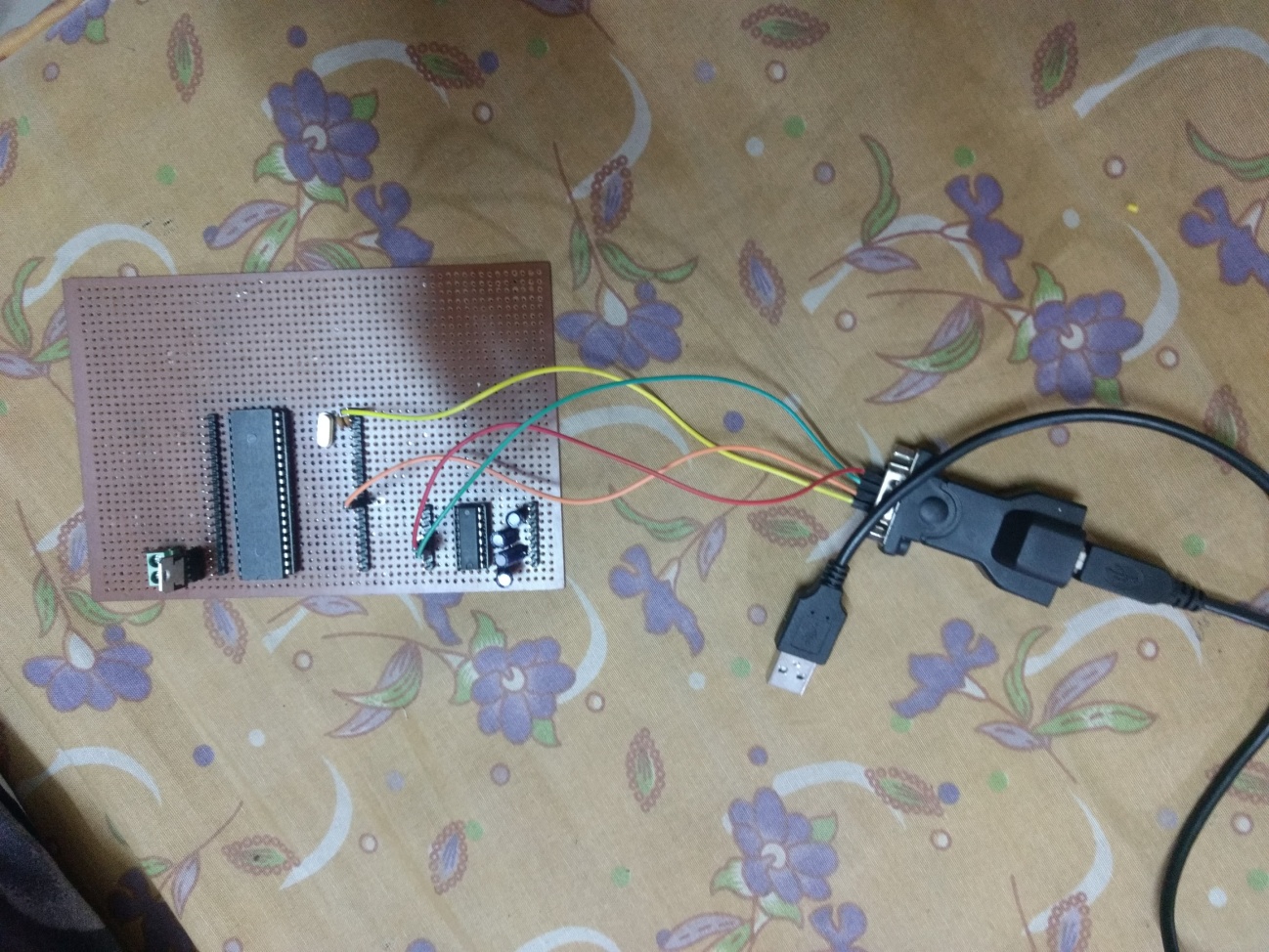
* We can do two such type of communication, because there are two T IN, T OUT, R IN, R OUT.
* From the DB9 package pins we can use only two pins which are transmitter (Tx) and receiver (Rx).

***DAY 20***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | rate | Quantity | Amount |
| USB To Serial | 200 | 1 | 200 |
| MAX 232 | 30 | 1 | 30 |
| Single stand wire | 10 | 3 | 30 |
| 4N35 | 15 | 5 | 75 |
| 4N35 stand | 3 | 8 | 25 |
| Total: | 360 | Grand Total: | 1230 |

***DAY 21***

* Made circuit to program AT89C52
* We used T1 and R1 set of MAX-232 because in this set Rx and Dx are as in microcontroller.



* Video-1 completed.

***DAY 22***

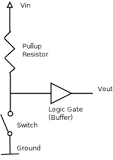
* Video-2 completed.
* Video-3 completed.
* Video-4 completed.

***DAY 23***

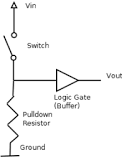
* There is a problem regarding programming microcontroller.
* Flash magic shown a message to reset microcontroller in ISR mode.

***DAY 24***

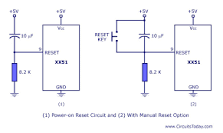
* Now there is reset pin in hardware kit of microcontroller, but instead of buying that whole hardware kit we had decided to make it ourselves.
* So now we have to make reset button in our kit.
* Concept of pull-up and pull-down resistor:
* When we connect microcontroller’s pin direct to GND through switch, then what happened?
* If switch is off then pin is in undetermined state, due to electromagnetic radiation. To solve this problem we used pull-up or pull-down resistor.
* If switch is on then pin is in low state. So there is no problem with this.
* Pull-up Resistor:



* Here in place of diode we can connect microcontroller’s pin.
* In this circuit if switch is off, then current will not flow through switch and pin is in HIGH state.
* And when switch is on, then current will flow through switch and pin is in LOW state.
* So when switch is off, it pull pin to high(up) so it is called pull-up resistor.
* Pull-down Resistor:



* Here in place of diode we can connect microcontroller’s pin.
* In this circuit if switch is off, then current will not flow through switch and pin is in LOW state.
* And when switch is on, then current will flow through switch and pin is in HIGH state.
* So when switch is off, it pull pin to low (down) so it is called pull-down resistor.
* Now the question is how we used this concept in our circuit.
* From datasheet we find that to make microcontroller reset we have to make RESET pin high for two machine-cycle.
* So first you think that, it is very simple, we have to connect PESET pin to only VCC through switch.
* When switch is off then pin is in undetermined state, which is undesirable.
* When switch is on then pin is HIGH, and it resets microcontroller.
* Due to undetermined status we can’t use this circuit.
* We can use pull-up or pull-down resistor circuit.
* Now we want that when we put switch in on state, pin should be goes into HIGH state.
* This requirement is fulfilled by pull-down resistor circuit.
* Now the circuit is



* Capacitor is extra added to make sure that device is not reset due to EMI.
* There are three series of microcontroller
* https://www.youtube.com/watch?v=773jNNjMIyE

1. Intel’s 8051(Harvard architecture, CICS instruction set)
2. ATmel’s AVR(modified Harvard architecture, RISC instruction set)
3. Microchips’s PIC(something different than Harvard architecture, RISC instruction set)

* Communication Protocols:
* <https://www.youtube.com/watch?v=PIQBwYwqP2g>
* In our projects we are going to use UART protocol.
* KEIL tools by ARM:
* The Keil μVision Integrated Development Environment (μVision IDE) supports

three major microcontroller architectures and sustains the development of a wide range of application.

1. **8-bit (classic and extended 8051)** devices include an efficient interrupt

system designed for real-time performance and are found in more than 65%

of all 8-bit applications. Over 1000 variants are available, with peripherals

that include analog I/O, timer/counters, PWM, serial interfaces like UART,

I2C, LIN, SPI, USB, CAN, and on-chip RF transmitter supporting low-power

wireless applications.

1. **16-bit (Infineon C166**, **XE166**, **XC2000)** devices are tuned for optimum

real-time and interrupt performance and provide a rich set of on-chip

peripherals closely coupled with the microcontroller core.

1. **32-bit (ARM7 and ARM9 based)** devices support complex applications,

which require greater processing power. These cores provide high-speed 32-

bit arithmetic within a 4GB address space. The RISC instruction set has

been extended with a Thumb mode for high code density.

* Page 28 of Book
* Video-5 completed.
* Video-6 completed.
* Video-7 completed.

***DAY 25***

* There is little bit changes in circuit.
* Continuing of DAY19,
* Brief procedure of UART communication:

1. Configure timer 1 in auto reload mode.
2. Load TH1 with value as per required baud rate e.g. for 9600 baud rate load 0xFD. (-3 in decimal)
3. Load SCON with serial mode and control bits. e.g. for mode 1 and enable reception, load 0x50.
4. Start timer1 by setting TR1 bit to 1.
5. Load transmitting data in SBUF register.
6. Wait until loaded data is completely transmitted by polling TI flag.
7. When TI flag is set, clear it and repeat from step 5 to transmit more data.

#### Timer 1 as a clock generator

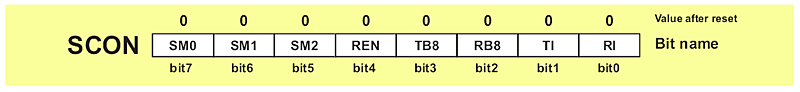
Timer 1 is usually used as a clock generator as it enables various baud rates to be easily set. The whole procedure is simple and is as follows:

* First, enable Timer 1 overflow interrupt.
* Configure Timer T1 to operate in auto-reload mode.
* Depending on needs, select one of the standard values from the table and write it to the TH1 register. That's all.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BAUD RATE** | **FOSC. (MHZ)** | | | | | **BIT SMOD** |
| 11.0592 | 12 | 14.7456 | 16 | 20 |
| 150 | 40 h | 30 h | 00 h |  |  | 0 |
| 300 | A0 h | 98 h | 80 h | 75 h | 52 h | 0 |
| 600 | D0 h | CC h | C0 h | BB h | A9 h | 0 |
| 1200 | E8 h | E6 h | E0 h | DE h | D5 h | 0 |
| 2400 | F4 h | F3 h | F0 h | EF h | EA h | 0 |
| 4800 |  | F3 h | EF h | EF h |  | 1 |
| 4800 | FA h |  | F8 h |  | F5 h | 0 |
| 9600 | FD h |  | FC h |  |  | 0 |
| 9600 |  |  |  |  | F5 h | 1 |
| 19200 | FD h |  | FC h |  |  | 1 |
| 38400 |  |  | FE h |  |  | 1 |
| 76800 |  |  | FF h |  |  | 1 |

1. SELECT COMMUNICATION MODE AND BAUD RATE

#### Serial Port Control (SCON) Register



* **SM0** (Serial port mode bit 0): It is used for serial port mode selection.
* **SM1** (Serial port mode bit 1): It is used for serial port mode selection.
* **SM2** (Serial port mode 2 bit): It also known as multiprocessor communication enable bit.

When set, it enables multiprocessor communication in mode 2 and 3, and eventually mode 1.

It should be cleared in mode 0.

* **REN** (Reception Enable bit): It enables serial reception when set.

When cleared, serial reception is disabled.

* **TB8** (Transmitter bit 8): Since all registers are 8-bit wide, this bit solves the problem of transmiting the 9th bit in modes 2 and 3.P bit of PSW resister is come into bit in order to be a part of the message.
* **RB8** (Receiver bit 8): It received the 9th bit in modes 2 and 3.

Cleared by hardware if 9th bit received is a logic 0.

Set by hardware if 9th bit received is a logic 1.

* **TI** (Transmit Interrupt flag): It is automatically set at the moment the last bit of one byte is sent. It's a signal to the processor that the line is available for a new byte transmite.

**It must be cleared from within the software.**

* **RI** (Receive Interrupt flag): It is automatically set upon one byte receive. It signals that byte is received and should be read quickly prior to being replaced by a new data.

**This bit is also cleared from within the software.**

* As seen, serial port mode is selected by combining the SM0 and SM1 bits:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SM0** | **SM1** | **MODE** | **DESCRIPTION** | **BAUD RATE** |
| 0 | 0 | 0 | 8-bit Shift Register | 1/12 the quartz frequency |
| 0 | 1 | 1 | 8-bit UART | Determined by the timer 1 |
| 1 | 0 | 2 | 9-bit UART | 1/32 the quartz frequency (1/64 the quartz frequency) |
| 1 | 1 | 3 | 9-bit UART | Determined by the timer 1 |

* MODE 0: Since there are no START and STOP bits or any other bit except data sent from the SBUF register in the pulse sequence, this mode is mainly used when the distance between devices is short, noise is minimized and operating speed is of importance. A typical example is I/O port expansion by adding a cheap IC (shift registers 74HC595).
* MODE 1: In mode 1, 10 bits are there: a START bit (always 0), 8 data bits (LSB first) and a STOP bit (always 1). The START bit is only used to initiate data receive, while the STOP bit is automatically written to the RB8 bit.
* MODE 2: In mode 2, 11 bits are there: a START bit (always 0), 8 data bits (LSB first), a programmable 9th data bit and a STOP bit (always 1). On transmit, the 9th data bit is actually the TB8 bit. This bit usually has a function of parity bit. On receive, the 9th data bit goes into the RB8 bit of the same register (SCON).The baud rate is either 1/32 or 1/64 the oscillator frequency.9th bit is used as parity chacker.
* MODE 3: Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable

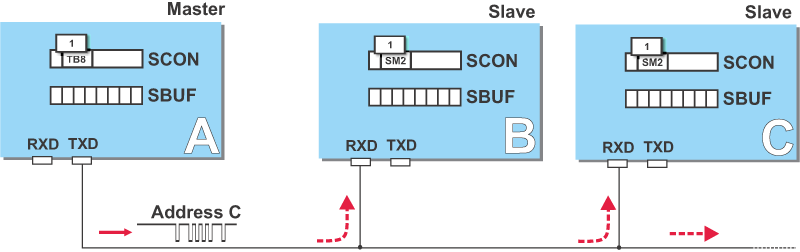
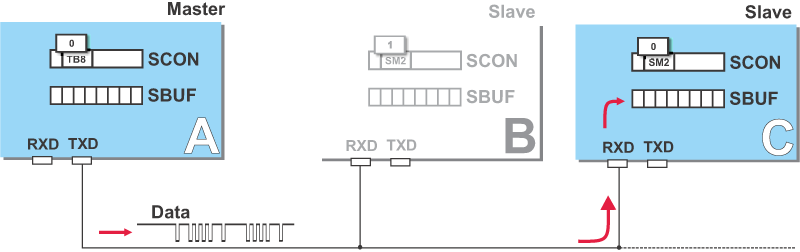
1. TRASMISSION

* All mode have same procedure for transmission
* Data transmit is initiated by writing data to the SBUF register.
* LSB transmitter first
* When all 8 bits have been sent, the TI bit of the SCON register is automatically set.

1. RECEIVE

* All mode have same procedure for transmission
* The START bit (logic zero (0)) on the RXD pin initiates data receive
* The following two conditions must be met: bit REN=1 and bit RI=0.
* LSB received first
* The RI bit is automatically set upon data reception is complete.
* Serial data is transmitted and received through RDX in MODE 1
* TDX pin outputs clock

#### Multiprocessor Communication

* As you may know, additional 9th data bit is a part of message in mode 2 and 3. It can be used for checking data via parity bit. Another useful application of this bit is in communication between two or more microcontrollers, i.e. multiprocessor communication. This feature is enabled by setting the SM2 bit of the SCON register. As a result, after receiving the STOP bit, indicating end of the message, the serial port interrupt will be generated only if the bit RB8 = 1 (the 9th bit). This is how it looks like in practice: Suppose there are several microcontrollers sharing the same interface. Each of them has its own address. An address byte differs from a data byte because it has the 9th bit set (1), while this bit is cleared (0) in a data byte. When the microcontroller A (master) wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte will generate an interrupt in all slaves so that they can examine the received byte and check whether it matches their address.Of course, only one of them will match the address and immediately clear the SM2 bit of the SCON register and prepare to receive the data byte to come. Other slaves not being addressed leave their SM2 bit set ignoring the coming data bytes.
* programming in C: <http://what-when-how.com/8051-microcontroller/serial-port-programming-in-c/>

***DAY 26***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| Charger | 220 | 1 | 220 |
| Capacitor | 5 | 10 | 50 |
| Resistors | 2 | 10 | 20 |
| Transistor | 5 | 3 | 15 |
| Diode | 1 | 2 | 2 |
| Total: | 310 | Grand Total: | 1540 |

***DAY 27***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| GCB | 50 | 1 | 50 |
| AT89C52 | 70 | 1 | 70 |
| IC pad | 30 | 1 | 30 |
| MAX232 | 30 | 1 | 30 |
| Total: | 190 | Grand Total: | 1730 |

***DAY 28***

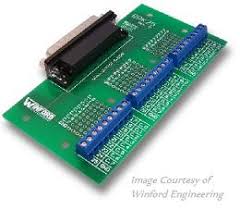
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |  |  |  |  |
| 8051 development board | 320 | 1 | 320 |  |  |  |  |
| AT89S52 | 70 | 1 | 70 |  |  |  |  |
| Total: | 390 | Grand Total: | 2120 |  |  |  |  |

* AT89C52 is not ISP programmable, so we have to use parallel programmer and can’t use 8051 development board.
* AT89S52 is pin-to-pin and code-to-code compatible to AT89C52, and in advance AT89S52 can be capable of in system program (ISP).
* Price of both microcontrollers are also same, but then why people are not using, that I don’t know.
* P89V51rd2 is microcontroller who is programmed just using development board, because it has in-built serial bootloader, so we can program it serially, but it is costly (around 220 INR).
* AT89S52 is ISP compatible device so we have to use USB ISP to convert USB signals into ISP form.
* The main difference between 89C51 and S51 lies in the following points:
* Pin function: Pins almost identical changes were, in the AT89S51 in P1.5, P1.6, P1.7 have a second function, that is, the second 3-pin serial ISP function formed programming interface.   
    
  Programmability: AT89C51 only supports parallel programming, But AT89S51 supports both parallel programming and ISP programming.   
  Voltage in the programming area, AT89C51 in addition to normal programming voltage of 5V, another 12v is needed for Vpp. But AT89S51 just need to be 4-5V.   
    
  Endurance: AT89C51 flash memory can be programmed upto 1000 times while the AT89S51 can be programmed from 1000-10000 times, so that the learner repeated the beginning of a more favorable programmer, reducing learning costs.   
    
  Frequency: AT89C51 limits the operating frequency is 24MHZ, and AT89S51 maximum operating frequency is 33MHZ, (AT89S51 chip in two models, to support the maximum operating frequency of 24MHZ and 33MHZ) in order to have a faster computing speed.   
    
  Power Supply: AT89S51 operating voltage range is up to 4-5.5V, while the AT89C51 at the end of the 4.8V and 5.3V.   
    
  ***Anti-jamming more: AT89S51 has integrated watchdog timer (Watchdog Timer), and AT89C51 needs an external watchdog timer circuit, or single-chip software constitute the internal watchdog timer to achieve anti-jamming software.***
* So we are changing our controller from AT89C52 to AT89S52.



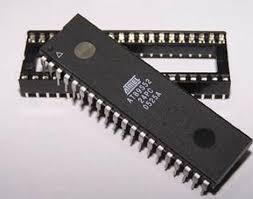
ZIF Socket

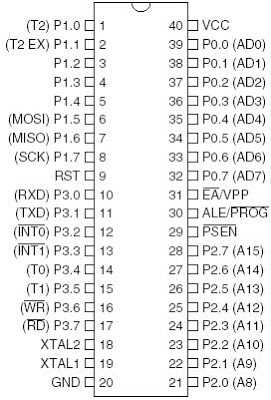
* Development is a PCB (GCB) board in which we can put our controller, oscillator, and other required component and we can connect peripherals with controller.
* Programmer is a board which contain another microcontroller to convert HEX file that out controller can understand.
* What is Breakout Board?
* Breakout boards are a common electrical components that take a bundled cable and “breaks out each conductor to a terminal that can easily accept a hook-up wire for distribution to another device. It breaks bundle of signals/wires to separate signal/wire.

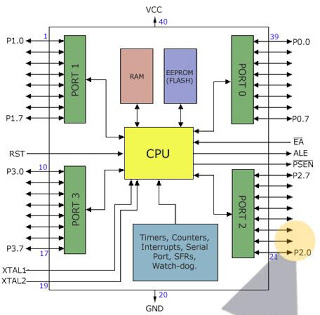


***DAY 29***

* So let’s start with aT89S52







* Program memory type - Flash ROM
* Program memory size – 8 KB
* One UART
* Operating voltage – 4 to 5.5 V
* Watchdog timer
* Two data pointer
* Three 16 bit timer/counter
* On-chip oscillator
* 10000 write/erase cycle
* Operating frequency – 0 to 33 MHz
* Three level program memory lock
* Two software selectable modes
* Pin description

1. VCC: power supply
2. GND: ground
3. PORT 0:

* Port 0 is an 8-bit open drain bidirectional I/O port.
* As an output port, each pin can sink eight TTL inputs.
* When used as an input it is in high impedance input state

1. PORT 1:

* Port 1 is an 8-bit bidirectional I/O port with internal pullups.
* The Port 1 output buffers can sink/source four TTL inputs.
* When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs.



1. PORT 2:

* Port 2 is an 8-bit bidirectional I/O port with internal pullups.
* The Port 2 output buffers can sink/source four TTL inputs.
* When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs.

1. PORT 3:

* Port 3 is an 8-bit bidirectional I/O port with internal pull-ups
* The Port 3 output buffers can sink/source four TTL inputs.
* When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs.



1. RST: this pin is works as an input to microcontroller.

* A high on this pin for two machine cycles, resets the device.
* This pin drives High for 96 oscillator periods after the Watchdog times out.
* The DISRTO bit in SFR AUXR (address 8EH) is used to control operation of RST.
* In its default state RESET high out feature is enabled
* To disable it the DISTRO bit in SFR AUXR register is used.

1. ALE/PROG:

* Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory.
* In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. To disable only this clock like operation we would set to ‘1’ to bit 0 of SFR register (8EH).
* This pin is also the program pulse input (PROG) during Flash programming.

1. PSEN :

* Program Store Enable (PSEN) is the read strobe to external program memory.

1. EA/VPP:

* External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH.
* EA should be strapped to VCC for internal program executions.
* This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

1. XTAL 1:

* For external oscillator

1. XTAL 2:

* For external oscillator
* The oscillator and clock
* Pins XTAL1 and XTAL2 is provided for connecting a resonant network to form an oscillator.
* Typically a quartz crystal and capacitors are employed. The crystal frequency is the basic internal clock frequency of the microcontroller.
* **Special Function registered memory:**
* Special function registers are the areas of memory that control specific functionality of the 8052 micro controller.

1. **Accumulator (0E0h)**

* As its name suggests, it is used to accumulate the results of large no of instructions. It can hold 8 bit values.

1. **B registers (0F0h)**

* The B register is very similar to accumulator. It may hold 8-bit value. The b register is only used by MUL AB and DIV AB instructions. In MUL AB the higher byte of the product gets stored in B register. In div AB the quotient gets stored in B with the remainder in A.

1. **Stack pointer (81h)**

* The stack pointer holds 8-bit value. This is used to indicate where the

next value to be removed from the stack should be taken from.

* When a value is to be pushed onto the stack, the 8052 first increase the value of SP and then store the value at the resulting memory location. When a value is to be popped from the stack, the 8052 returns the value from the memory location indicated by SP and then decrements the value of SP.

1. **Data pointer**

* The SFRs DPL and DPH work together work together to represent a 16-bit value called the data pointer.
* To facilitate accessing both internal and external data there are two registers each of 16 bit is provided, named DP0 at 82-3H and DP1 at 84-5
* In AUXR1 if DPS is 0 DP0 is selected
* In AUXR1 if DPS is 1 DP1 is selected
* So user should always initialize DPS before using data pointer

1. **Program counter**

* The program counter is a 16 bit register, which contains the 2 byte address, which tells the 8052 where the next instruction to execute to be found in memory. When the 8052 is initialized PC starts at 0000h. And is incremented each time an instruction is executes. It is not addressable SFR.

1. **PCON (power control, 87h)**

* The power control SFR is used to control the 8051’s power control modes. Certain

operation modes of the 8051 allow the 8051 to go into a type of “sleep mode”

which consumes much less power.



1. **TCON (timer control, 88h)**

* The timer control SFR is used to configure and modify the way in which the 8051’s two timers operate.
* This SFR controls whether each of the two timers is running or stopped and contains a flag to indicate that each timer has overflowed. Additionally, some non-timer related bits are located in TCON SFR. These bits are used to configure the way in which the external interrupt flags are activated, which are set when an external interrupt occurs.



1. **TMOD (Timer Mode, 89h)**

* The timer mode SFR is used to configure the mode of operation of each of the two

timers.

* Using this SFR your program may configure each timer to be a 16-bit timer, or 13 bit timer, 8-bit auto reload timer, or two separate timers. Additionally you may configure the timers

to only count when an external pin is activated or to count “events” that are indicated on an

external pin.



1. **TO (Timer 0 low/high, address 8A/8C h)**

* These two SFRs taken together represent timer 0. Their exact behavior depends on how

the timer is configured in the TMOD SFR; however, these timers always count up. What is

configurable is how and when they increment in value.

1. **T1 (Timer 1 Low/High, address 8B/ 8D h)**

* These two SFRs, taken together, represent timer 1. Their exact behavior depends on how the timer is configured in the TMOD SFR; however, these timers always count up.

1. **T2 (Timer 2)**

* T2CON is control register for timer 2.
* T2MOD is status register for timer 2.
* The register pair RCAP2H and RCAP2L is the capture/reload register for timer 2 in 16-bit capture mode or 16-bit reload mode.

1. **P0 (Port 0, address 90h, bit addressable)**

* This is port 0 latch. Each bit of this SFR corresponds to one of the pins on a micro

controller. Any data to be outputted to port 0 is first written on P0 register. For e.g., bit 0 of

port 0 is pin P0.0, bit 7 is pin p0.7. Writing a value of 1 to a bit of this SFR will send a high

level on the corresponding I/O pin whereas a value of 0 will bring it to low level.

1. **P1 (port 1, address 90h, bit addressable)**

* This is port latch1. Each bit of this SFR corresponds to one of the pins on a micro

controller. Any data to be outputted to port 0 is first written on P0 register. For e.g., bit 0 of

port 0 is pin P1.0, bit 7 is pin P1.7. Writing a value of 1 to a bit of this SFR will send a high

level on the corresponding I/O pin whereas a value of 0 will bring it to low level.

1. **P2 (port 2, address 0A0h, bit addressable):**

* This is a port latch2. Each bit of this SFR corresponds to one of the pins on a microcontroller. Any data to be outputted to port 0 is first written on P0 register. For e.g., bit 0 of port 0 is pin P2.0, bit 7 is pin P2.7. Writing a value of 1 to a bit of this SFR will send a high level on the corresponding I/O pin whereas a value of 0 will bring it to low level.

1. **P3 (port 3, address B0h, bit addressable) :**

* This is a port latch3. Each bit of this SFR corresponds to one of the pins on a microcontroller. Any data to be outputted to port 0 is first written on P0 register. For e.g., bit 0 of port 0 is pin P3.0, bit 7 is pin P3.7. Writing a value of 1 to a bit of this SFR will send a high level on the corresponding I/O pin whereas a value of 0 will bring it to low level.

1. **IE (interrupt enable, 0A8h):**

* The Interrupt Enable SFR is used to enable and disable specific interrupts. The low 7 bits of the SFR are used to enable/disable the specific interrupts, where the MSB bit is used to enable or disable all the interrupts. Thus, if the high bit of IE is 0 all interrupts are disabled regardless of whether an individual interrupt is enabled by setting a lower bit.



1. **PSW (Program Status Word, 0D0h)**

* The program Status Word is used to store a number of important bits that are set and cleared by 8052 instructions. The PSW SFR contains the carry flag, the auxiliary carry flag, the parity flag and the overflow flag. Additionally, it also contains the register bank select flags, which are used to select, which of the “R” register banks currently in use.



1. **SBUF (Serial Buffer, 99h)**

SBUF is used to hold data in serial communication. It is physically two registers. One is writing only and is used to hold data to be transmitted out of 8052 via TXD. The other is read only and holds received data from external sources via RXD. Both mutually exclusive registers use address 99h.

***DAY 30***

* Memory Organization:
* Microcontroller can maximum of 64KB of memory for each program and data memory can be address.
* Program memory
* When EA is 0 all program is fetched by external memory
* When EA is 1 all program is fetched by internal and external memory both
* Internal from 0000H to 1FFFH
* External from 2000H to FFFFH
* Data memory
* 256 bits of on-chip-RAM

***DAY 31***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| 40 pin IC pad | 15 | 2 | 30 |
| Oscillator | 10 | 2 | 20 |
| Counter part of USBISP | 5 | 2 | 10 |
| Total: | 60 | Grand Total: | 2180 |

***DAY 32***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| Jumper wire | 5 | 30 | 150 |
| A 102j | 5 | 10 | 50 |
| Led | 1 | 20 | 20 |
| Push button | 2 | 10 | 20 |
| Total: | 240 | Grand Total: | 2420 |

* We successfully programmed our AT89S52 using USBISP programmer and microcontroller PCB.
* Now we are trying to program our controller using only USBISP and crystal oscillator.
* We are getting “flash verify error ant 0h” error.
* But at last we solved error.

***DAY 33***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| GCB | 50 | 2 | 100 |
| Male header | 10 | 2 | 20 |
| Total: | 120 | Grand Total: | 2540 |

* Video-8 completed.
* Video-9 completed.
* Video-10 completed.
* Video-11 completed.
* Video-12 completed.
* Video-13 completed.
* Video-14 completed.
* Video-15 completed.
* Video-16 completed.
* Video-17 completed.
* Video-18 completed.
* Tried to sort problem of tri-state buffer.
* Then I realized that, when we will connect buffer’s output with MC then it may work correctly.
* So decided to leave it.
* Make a power supply with two 7805, whose output is 12 V as well as 5 V.
* Make MC circuit to program it and connect it with input and output using male-female header.
* Make an input circuit no. 1 using switch as an input.
* Once again read about IN1232 watchdog timer IC.
* Analyzed output circuit of MC card.
* Started reading about MC from starting to write just one line of code.

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| 4N35 | 15 | 10 | 150 |
| connector | 10 | 5 | 50 |
| 6 pin IC pad | 1 | 2 | 2 |
| 74HC244n | 2 | 25 | 50 |
| 20 pin IC pad | 3 | 2 | 6 |
| Red led | 1 | 10 | 10 |
| 7 segment LED CC | 15 | 2 | 30 |
| 4 bit DIP switch | 20 | 2 | 40 |
| Total: | 340 | Grand Total: | 2760 |

***DAY 34***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| GCB | 50 | 2 | 100 |
| 1 k resistor | 1 | 20 | 20 |
| Oscillator 16 | 10 | 2 | 20 |
| Total: | 140 | Grand Total: | 2900 |

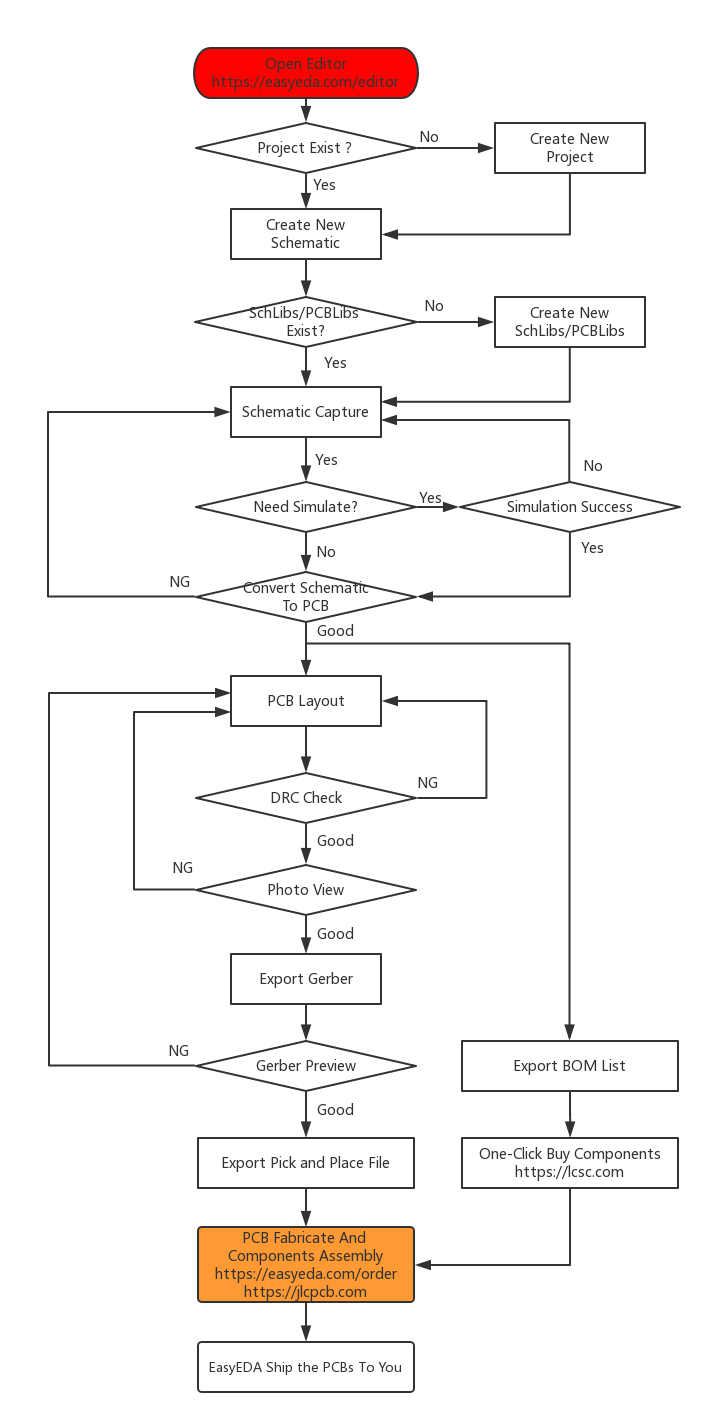
***DAY 35***

* Through Hole Technology: For years, through-hole technology was utilized in the construction of nearly all printed circuit boards (PCBs).
* This particular mounting scheme involves the use of leads on electrical components, which are then inserted into holes that were drilled on the PCB and soldered to pads situated on the opposite side.
* Through-hole mounting is extremely reliable, as it provides strong mechanical bonds.
* Surface Mount Technology: SMT enables electrical components to be mounted, or directly placed, onto the surface of a PCB.
* Because the PCB of a surface-mount devices (SMDs) does not require as many drilled holes, and the components are more compact, higher circuit densities are possible on smaller boards.
* Major differences:

|  |  |
| --- | --- |
| Through Hole Technology | Surface Mount Technology |
| It has space problem | SMT resolves the space problems |
| through-hole components require lead wires that pass through drilled holes. | In SMT, components do not have leeds and are directly mounted to the PCB |
| The pin count is lower in THT | The pin count is higher in SMT |
| It has lower density, due to large component size | Because components are more compact, the packing density achieved through SMT is much higher than in through-hole mounting. |
| It has lower capital requirements | Although SMT is typically cheaper on the production side, the capital required for investing in machinery is higher |
| It is slower than SMT | SMT makes it easier to acquire higher circuit speeds because of its reduced size. |
| It required less advance technology than SMT | [SMT demands is quite advanced as compared to through-hole technology](http://info.zentech.com/blog/bid/246434/SMT-Production-Adds-Efficiency-To-Any-Process). |
| Through-hole mounting is typically more desirable than SMT in terms of large, bulky components, components that are subject to frequent mechanical stress, or for high-power and high-voltage parts. | It could be used in passive, low powered components. Like transistor, resistor, led, etc. |

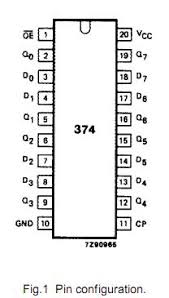
***DAY 36***

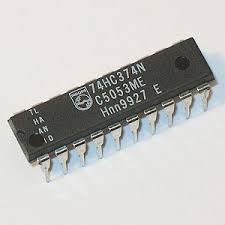
* <https://youtu.be/35YuILUlfGs>



***DAY 37***

* 3-state non-inverting outputs
* 8-bit positive, edge-triggered register
* Common 3-state output enable input
* Output capability: bus driver
* The 74HC/HCT374 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL.
* They are specified in compliance with JEDEC standard no. 7A.
* The 74HC/HCT374 are octal D-type flip-flops featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications.
* A clock (CP) and an output enable (OE) input are common to all flip-flops.
* The 8 flip-flops will store the state of their individual D-inputs that meet the set-up and hold times requirements on the LOW-to-HIGH CP transition.
* When OE is LOW, the contents of the 8 flip-flops are available at the outputs.
* When OE is HIGH, the outputs go to the high impedance OFF-state.
* Operation of the OE input does not affect the state of the flip-flops.
* The “374” is functionally identical to the “534”, but has non-inverting outputs.





|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| Male-female jumper | 5 | 20 | 100 |
| 2n 2222 transistor | 5 | 10 | 50 |
| 10 uf capacitor | 2 | 10 | 20 |
| 10 k ohm resistor | 1 | 10 | 10 |
| 10 k ohm resistor array | 5 | 2 | 10 |
| Red led 3 mm | 1 | 10 | 10 |
| 1232 watchdog timer | 25 | 2 | 50 |
| Copper plate | 120 | 1 | 120 |
| FeCl3 | 140 | 1 | 140 |
| Total: | 500 | Grand Total: | 3400 |

* Diameter of typical dip packages hole is 1 mm.
* Diameter of pin of green connector is also 1 mm.
* Distance between two holes in GCB is 100 mil.

***DAY 38***

* To set 1 second time delay use number 32768 to load in variable and count.

***DAY 39***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| Copper plate | 120 | 1 | 120 |
| PCB hand drill | 160 | 1 | 160 |
| Drill panu 1.5 mm | 30 | 1 | 30 |
| D-flip flop | 30 | 4 | 120 |
| Total: | 430 | Grand Total: | 3330 |

***DAY 40***

* PCB schematic:
* 3 mm LED
* 4N35: LCSC part no. C141786
* SN74HC244N: LCSC part no. C2912
* AT89S52: LCSC part no. C9441
* IN1232 or DS1232LP: LCSC part no. C35525
* 4 BIT DIP SWITCH: LCSC part no. C99418
* SN74HC374N OR D-flip flop: LCSC part no. C5237
* 2N2222: LCSC part no. C118536
* ULN2803A: C73936

***DAY 41***

* To give PCB in manufacture we have to give mentioned gerber files:
* TOP
* BOTTOM
* TOP MASK
* BOTTOM MASK
* TOP SILK
* BOTTOM SILK
* DRILL FILE

***DAY 42***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| 4N35 | 12 | 24 | 288 |
| 20 pin IC pad | 5 | 8 | 40 |
| Led | 2 | 46 | 92 |
| 2k resistor | 1 | 35 | 35 |
| 2 pin connector | 5 | 25 | 125 |
| 1 k resistor | 1 | 20 | 20 |
| 6 pin IC pad | 3 | 60 | 180 |
| 3 pin connector | 7 | 2 | 14 |
| SN74HC244N | 25 | 5 | 125 |
| 0.1 microfarad | 1 | 15 | 15 |
| 16 MHz oscillator | 10 | 1 | 10 |
| AT89S52 | 1 | 70 | 70 |
| ULN2803 | 25 | 1 | 25 |
| D flip flop | 30 | 3 | 90 |
| other | 1 | 110 | 110 |
| Total: | 1245 | Grand Total: | 4575 |

***DAY 43***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| 4N35 | 10 | 35 | 350 |
| Single stand wire | 10 | 2 | 20 |
| Tamiya wire | 10 | 2 | 20 |
| 3 mm LED | 2 | 15 | 30 |
| 1k resistor array | 5 | 20 | 100 |
| Total: | 525 | Grand Total: | 5100 |

***DAY 44***

* 170 lines of code have been written.

***DAY 45***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| Various IC pads | 650 | 1 | 650 |
| AT89S52 | 90 | 1 | 90 |
| Total: | 740 | Grand Total: | 5840 |

***Day 46***

|  |  |  |  |
| --- | --- | --- | --- |
| Component name | Rate | quantity | Amount |
| AT89S52 | 90 | 1 | 90 |
| Red black wire | 40 | 1 | 40 |
| IC pad | 5 | 5 | 25 |
| DMM wire | 20 | 1 | 20 |
| Total: | 175 | Grand Total: | 6015 |

And Happy Coding