

PCB DESIGN

Submitted in partial fulfillment of the requirements for the

Professional Training –I of

Bachelor of Engineering Degree in

Electrical and Electronics Engineering

BY

MD DILSHAD(39140031)



DEPARTMENT OF ELECTRICAL AND ELECTRONICS

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119. TAMILNADU.

OCTOBER 2021

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. Nandhita., Dean, School of Electrical and Electronics Engineering** and **Dr. V. Sivachidambaranathan, Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our year coordinator **Dr. M. KAVITHA** for her valuable guidance and constant encouragement paved the way for the successful completion of our work.

INTERNSHALA TRAININGS

A little progress each day adds up to big results!
I am glad to share that I have successfully completed PCB Design training from Internshala Trainings. Check out my certificate here!



ABSTRACT

The printed circuit board is the platform upon which microelectronic components such as semiconductor chips and capacitors are mounted. It provides the electrical interconnections between components. Virtually every electronic product is constructed with one or more printed-circuit boards (PCBs). The PCBs hold the ICs and other components and implement the interconnections between them. PCBs are created in abundance for portable electronics, computers, and entertainment equipment. They are also made for test equipment, manufacturing, and spacecraft. Eventually, almost every EEE must design a PCB, which is not something that is taught in school. Yet engineers, technicians, and even novice PCB designers can create high-quality PCBs for any and every purpose with confidence that the outcome will meet or exceed the objective. Also, these designs can be completed on schedule and within budget while meeting the design requirements. Designers just need to mind the essential documentation, design steps and strategies, and final checks.

INTRODUCTION OF AUTODESK EAGLE

EAGLE is electronic design automation (EDA) software that lets printed circuit board (PCB) designers seamlessly connect schematic diagrams, component placement, PCB routing and comprehensive library content.

EAGLE stands for Easily Applicable Graphical Layout Editor and is developed by [CadSoft Computer GmbH](#). The company was acquired by [Autodesk Inc.](#) in 2016.

EAGLE contains a [schematic editor](#), for designing [circuit diagrams](#). Schematics are stored in files with .SCH extension, parts are defined in device libraries with .LBR extension. Parts can be placed on many sheets and connected together through ports.

The [PCB](#) layout editor stores board files with the extension .BRD. It allows [back-annotation](#) to the schematic and auto-routing to automatically connect traces based on the connections defined in the schematic.

EAGLE saves [Gerber](#) and [PostScript](#) layout files as well as [Excellon](#) and [Sieb & Meyer](#) drill files. These are standard file formats accepted by [PCB fabrication](#) companies, but given EAGLE's typical user base of small design firms and hobbyists, many PCB fabricators and [assembly shops](#) also accept EAGLE board files (with extension .BRD) directly to export optimized production files and [pick-and-place](#) data themselves.

EAGLE provides a multi-window graphical user interface and menu system for editing, project management and to customize the interface and design parameters. The system can be controlled via mouse, keyboard hotkeys or by entering specific commands at an embedded [command line](#). Keyboard hotkeys can be user defined. Multiple repeating commands can be combined into [script files](#) (with file extension .SCR). It is also possible to explore design files utilizing an EAGLE-specific object-oriented programming language (with extension .ULP).

PROJECT 1:- ASTABLE 555 TIMER

The 555 timer IC can be used with a few simple components to build an astable circuit which produces a 'square wave'. This is a digital waveform with sharp transitions between low (0V) and high (+Vs), the durations of the low and high states may be different. The circuit is called an *astable* because it is not stable in any state: the output is continually changing between 'low' and 'high'.

The frequency is the number of pulses per second. The formula to calculate the frequency of the output voltage is:

$$f = \frac{1.44}{(R_1 + 2R_2)C}$$

The period is the time covered for one pulse. This is just the reciprocal of the frequency:

$$T = \frac{1}{f} = 0.694(R_1 + 2R_2)C$$

The high time (T_1) and low time (T_0) can be calculated using the formulas below. Note that the period is the sum of the high time and the low time.

$$T_1 = 0.694(R_1 + R_2)C$$

$$T_0 = 0.694R_2C$$

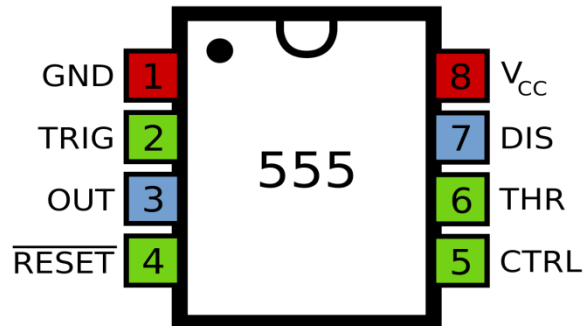
The mark space ratio is the ratio between the high time and the low time or:

$$\text{Mark Space Ratio} = \frac{T_1}{T_0}$$

The duty cycle is more commonly used than the mark space ratio. The formula for the duty cycle is:

$$\text{Duty Cycle} = \frac{T_1}{T} \times 100$$

PIN CONFIGURATION



Pin 1 (Ground):

Connects to the 0v power supply.

Pin 2 (Trigger):

The o/p of the timer depends on the amplitude of the external trigger pulse applied to this pin.

Pin 3 (Output):

(Pins 3 and 7 are "in phase.") Goes HIGH (about 2v less than rail) and LOW (about 0.5v less than 0v) and will deliver up to 200mA.

Pin 4 (Reset):

The 555 timer can be reset (disabled) by applying a negative pulse to this pin. When the reset function is not in use, the reset terminal should be connected to +V_{CC} to avoid any false triggering.

Pin 5 (Control):

A voltage applied to this pin will vary the timing of the RC network (quite considerably).

Pin 6 (Threshold):

This is the non inverting input terminal of upper comparator which monitors the voltage across the external capacitor.

Pin 7 (Discharge):

This pin is connected internally to the collector of transistor Q1.

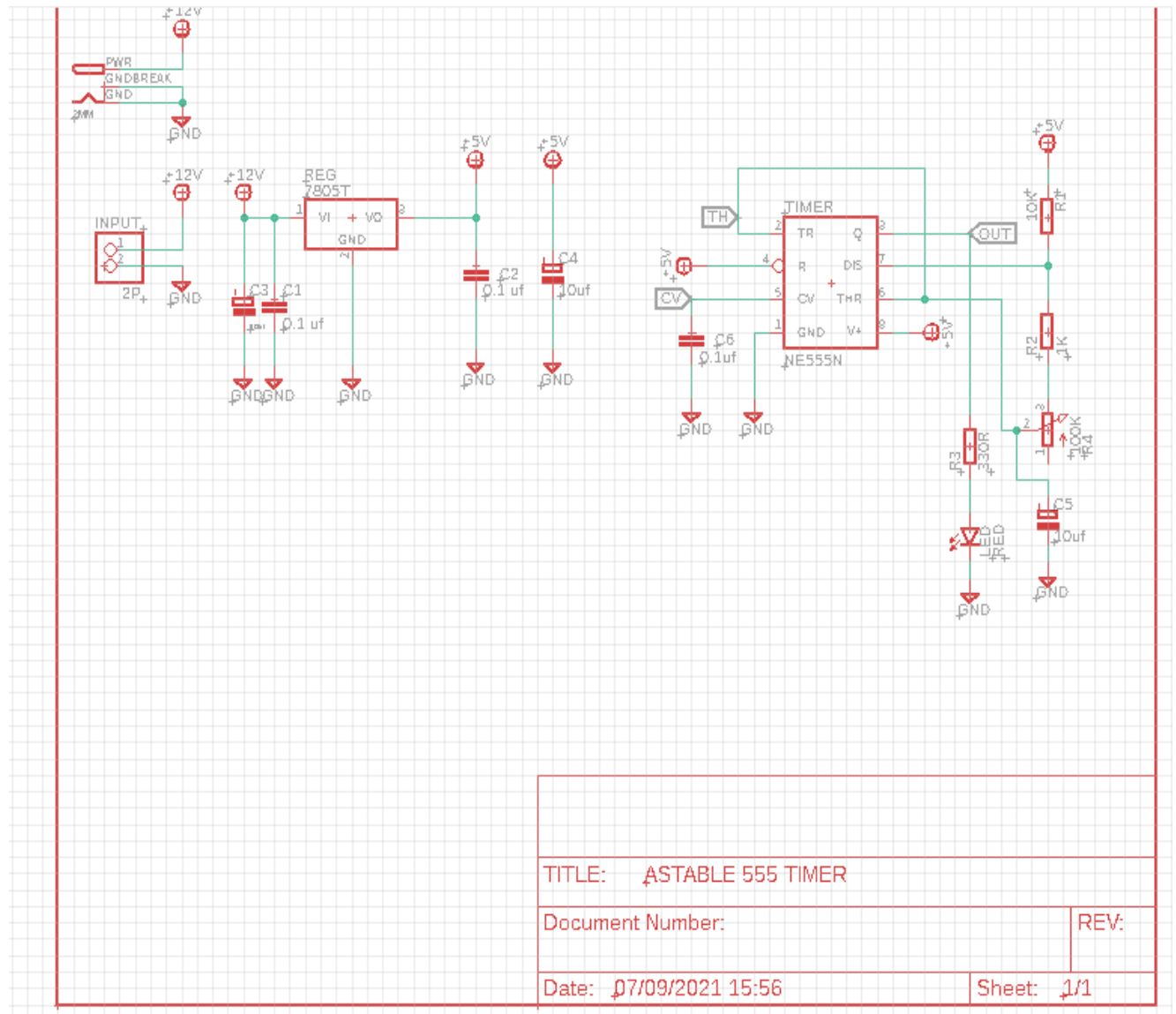
When the output is high Q1 is OFF.

When the output is low Q is (saturated) ON.

Pin 8 (Supply):

The supply voltage of +5V to +18V is applied to this pin with respect to ground.

SCHEMATIC DIAGRAM

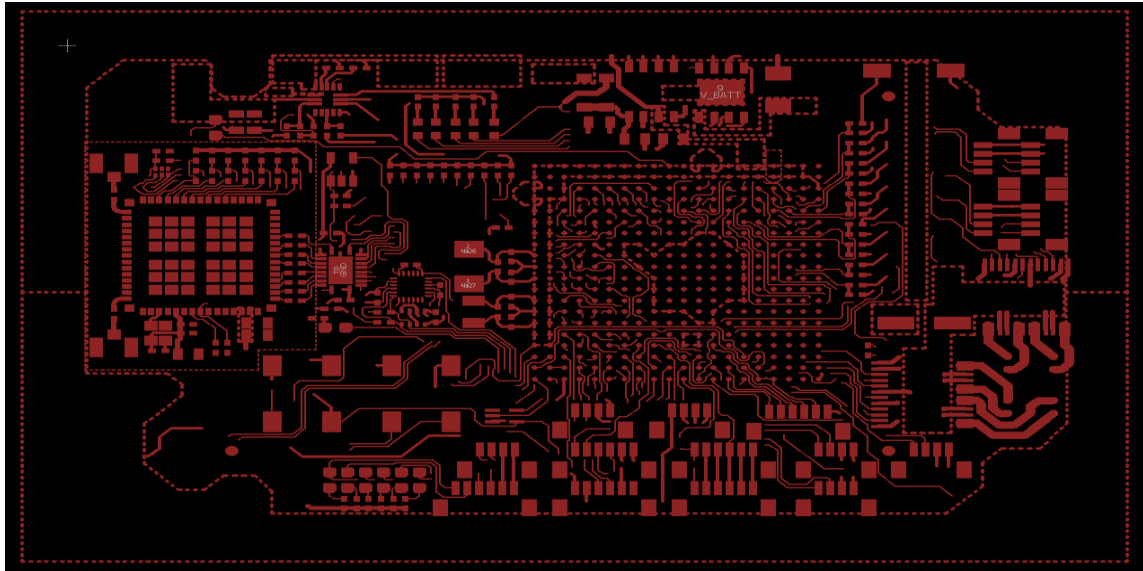


PCB LAYERS

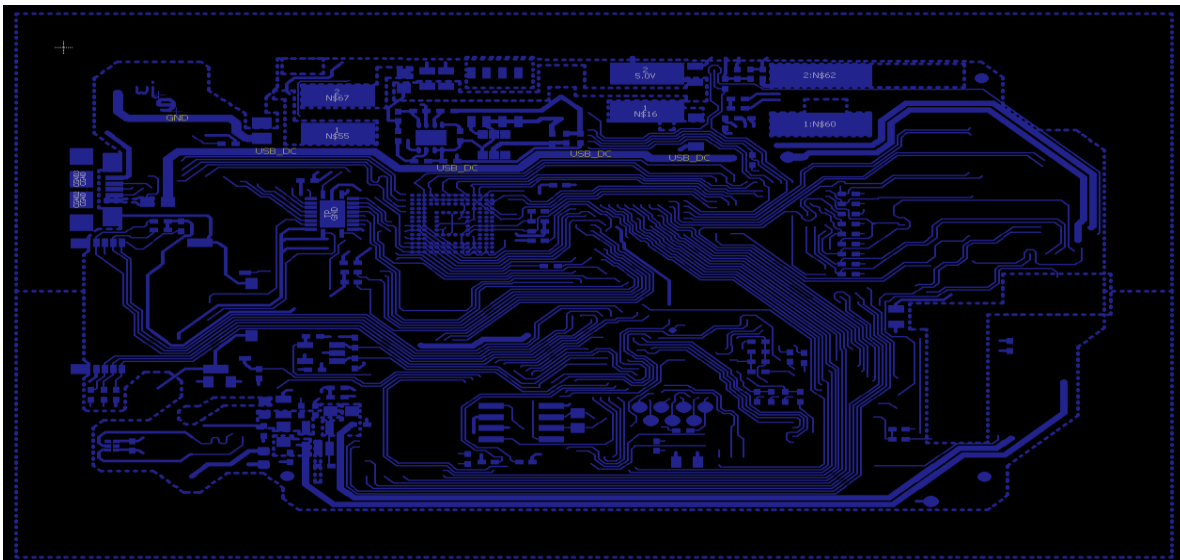
Printed circuit boards (PCBs) consist of anything from one layer to multiple layers of dielectric and conductive materials. When bonded into boards, these layers carry circuits that power a vast range of home electronics, such as alarm clocks, kitchen appliances, desk supplies, computers and mobile devices.

Layers:		
Nr		Name
1		Top
16		Bottom
17		Pads
18		Vias
19		Unrouted
20		Dimension
21		tPlace
22		bPlace
23		tOrigins
24		bOrigins
25		tNames
26		bNames
27		tValues
28		bValues
29		tStop
30		bStop
31		tCream
32		bCream
33		tFinish
34		bFinish
35		tGlue
36		bGlue
37		tTest
38		bTest
39		tKeepout
40		bKeepout
41		tRestrict
42		bRestrict
43		vRestrict
44		Drills
45		Holes
46		Milling
47		Measures
48		Document
49		Reference
50		dxs
51		tDocu
52		bDocu

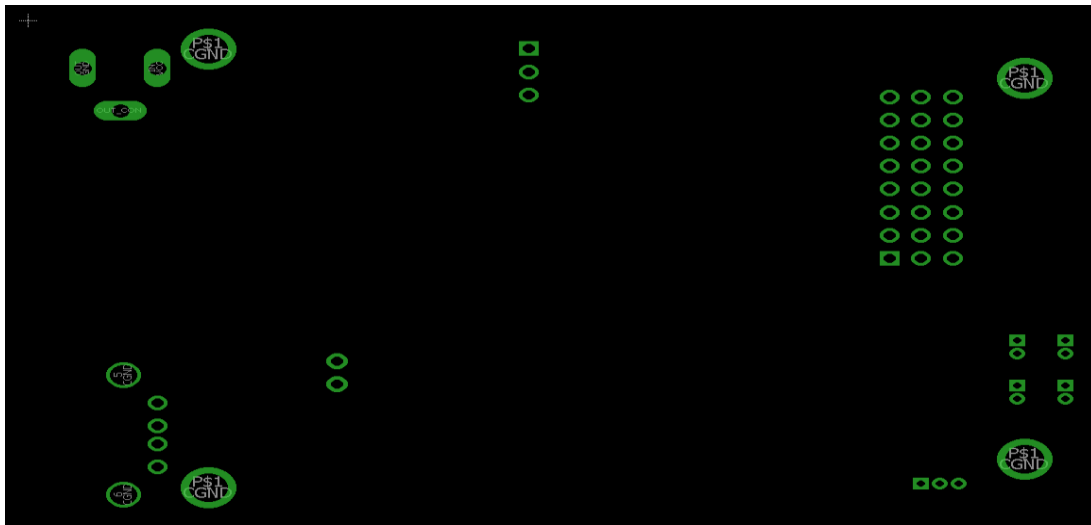
LAYER 1: TOP This first layer contains the copper on the top of your board, whether that's a polygon from a copper pour or individual copper traces. Using this layer to generate a pour will provide an accessible area of copper for all of your signals to ground. Also, when creating pads for surface mount components, Autodesk EAGLE will default to using Layer 1 for the pad's placement.



LAYER 16: BOTTOM Just like Layer 1, this layer contains the copper on the bottom of your board, whether that's from copper pours or individual copper traces. Components placed on the bottom of your board will also have their pads placed here.



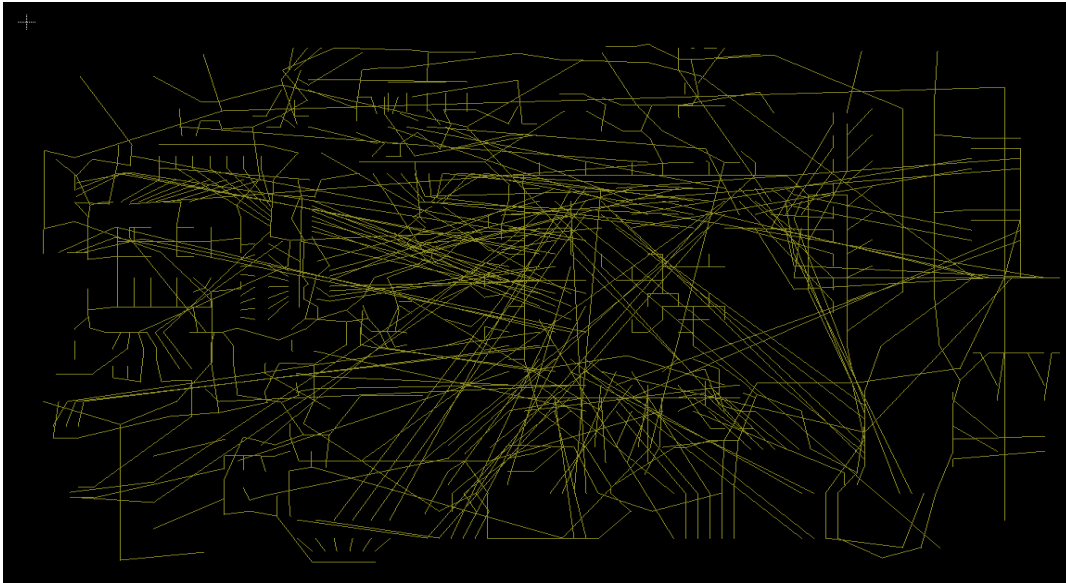
LAYER 17: PADS Here you'll find all of your through-hole pads, which includes both the hole on the copper surrounding it (Annular Ring). When placing a pad on this layer, it will place an annular ring on both the top and bottom layers of your board. Keep in mind that you'll rarely need to tinker with this layer, as through-hole pads are automatically generated once you place down a through-hole package.



LAYER 18: VIAS This is the layer for all of your vias, which provide a quick path of connectivity for signals between layers on your PCB. Note that both vias and through-hole pads will look nearly identical, so it's always useful to be able to hide/show just Layer 18 or 17 to understand what specific object you're viewing.



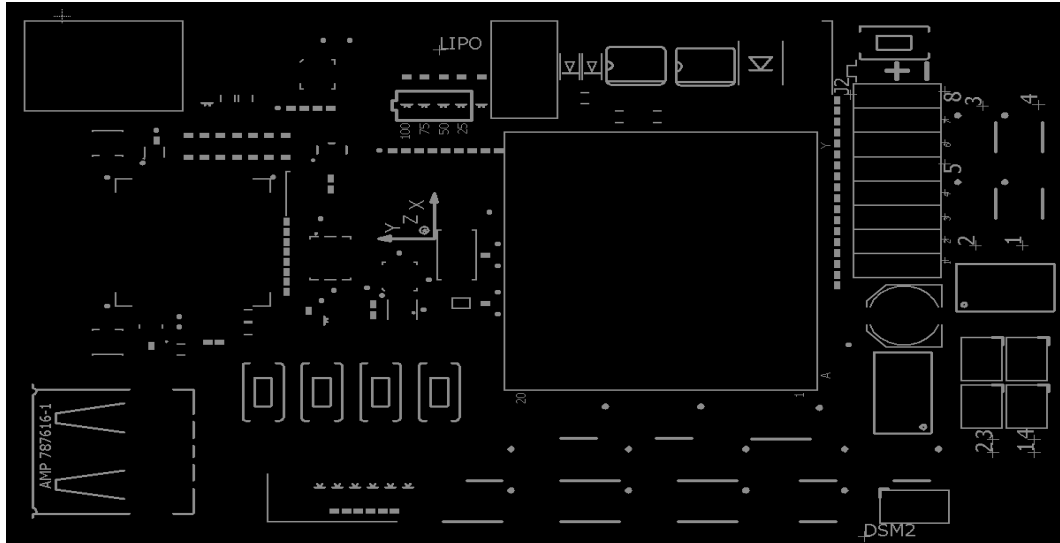
LAYER 19: UNROUTED When you first start a PCB layout, all of your components will be connected with a set of airwires, also called a ratsnest. These lines specify connectivity between all of the pins on your components, and these lines live on the Unrouted layer until they get connected. By the time your PCB has been fully routed, you should no longer have any visible Unrouted wires.



LAYER 20: DIMENSION The Dimension layer has several purposes, the first of which is to specify the outline of your board. Secondly, you can also use this layer in your design rules to keep copper pours away from the edge of your PCB.



LAYER 21-22: TPLACE/BPLACE These two layers contain both the top and bottom silkscreen on your PCB, and also component outlines to show the positioning of parts. You need to be careful when using this layer not to place silkscreen on any soldered areas. Otherwise, you could risk creating a short on your board or creating an unsolderable pad. However, if you want to add any kind of additional artwork outside of regular silkscreen, like text or logos, then Layer 21-22 is the place to do it.



LAYER 25-26: T NAMES/BNAMES As the name suggests, these two layers contain both the top and bottom print for your component names. Every component on your PCB has a unique name, also called a reference designator, and will look something like R1, C1, D1, etc.



PCB WARNINGS & ERRORS

ELECTRIC RULE CHECKING (ERC):-

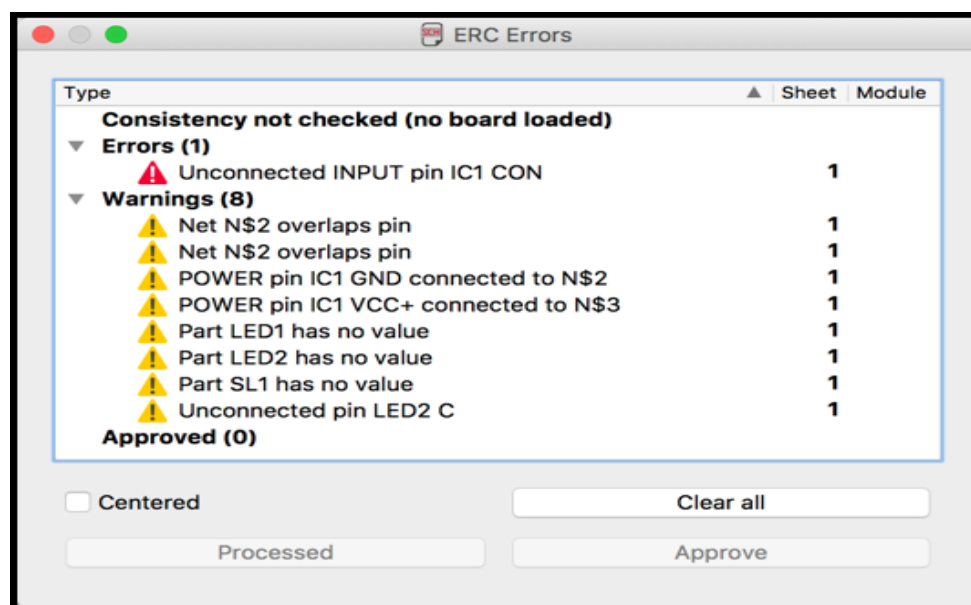
Electrical rule checking (ERC) is a methodology used to check the robustness of a design both at schematic and layout levels against various “electronic design rules”. These design rules are often project-specific and developed based on knowledge from previous tapeouts or in anticipation of potential new failures.

TYPES OF ERC ERRORS:-

Consistency Errors: ⚠ This error means that something isn't right between your schematic and PCB layout, and your changes likely aren't in sync between both files. If you get one of these errors, consider it a major red flag to compare both versions of your design before moving ahead.

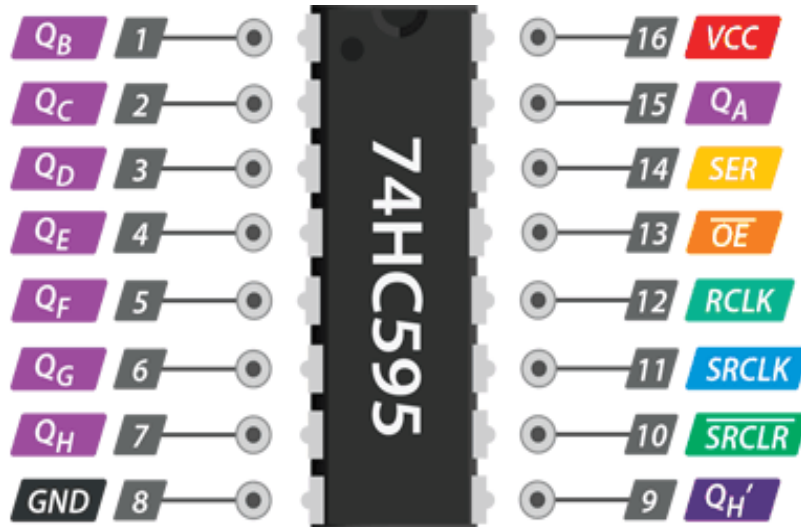
Errors: ⚠ Errors are the red stoplights. You need to step on the breaks and check these out before proceeding. Errors can be pretty significant, like having an unconnected pin that will throw your whole circuit out of whack.

Warnings: ⚠ Warnings are the yellow stoplights. While these types of errors might not mess up your design if you move forward without fixing them, it's good practice to slow down and review them one by one. These guys are minor compared to Errors, and include many non-design breaking issues like missing net values.



PROJECT 1:- SHIFT REGISTER 74595

The SNx4HC595 devices contain an 8-bit, serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. The storage register has parallel 3-state outputs. Separate clocks are provided for both the shift and storage register. The shift register has a direct overriding clear (SRCLR) input, serial (SER) input, and serial outputs for cascading. When the output-enable (OE) input is high, the outputs are in the high-impedance state.



GND should be connected to the ground of Arduino.

VCC is the power supply for 74HC595 shift register which we connect the 5V pin on the Arduino.

SER (Serial Input) pin is used to feed data into the shift register a bit at a time.

SRCLK (Shift Register Clock) is the clock for the shift register. The 595 is clock-driven on the rising edge. This means that in order to shift bits into the shift register, the clock must be HIGH. And bits are transferred in on the rising edge of the clock.

RCLK (Register Clock / Latch) is a very important pin. When driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register; which ultimately shows up at the output. So the latch pin can be seen as like the final step in the process to seeing our results at the output, which in this case are LEDs.

SRCLR (Shift Register Clear) pin allows us to reset the entire Shift Register, making all its bits 0, at once. This is a negative logic pin, so to perform this reset; we need to set the SRCLR pin LOW. When no reset is required, this pin should be HIGH.

OE (Output Enable) is negative logic too: When the voltage on it is HIGH, the output pins are disabled/set to high impedance state and don't allow current to flow. When OE gets low voltage, the output pins work normally.

QA–QH (Output Enable) are the output pins and should be connected to some type of output like LEDs, 7 Segments etc.

QH' Pin outputs bit 7 of the ShiftRegister. It is there so that we may daisychain 595s: if you connect this QH' to the SER pin of another 595, and give both ICs the same clock signal, they will behave like a single IC with 16 outputs. Of course, this technique is not limited to two ICs – you can daisychain as many as you like, if you have enough power for all of them.

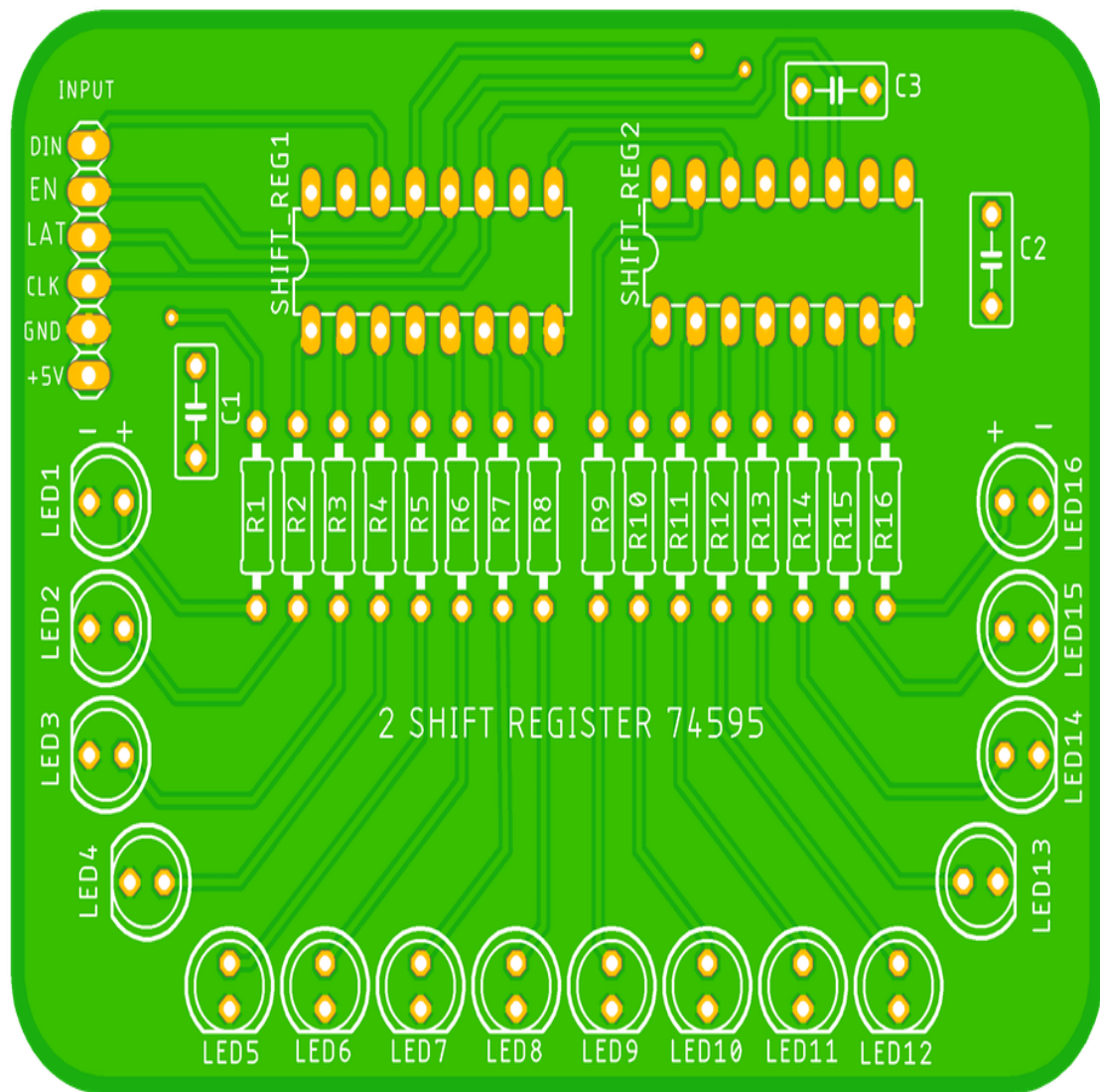
NAME	PIN		I/O ⁽¹⁾	DESCRIPTION
	SOIC, PDIP, SO, CDIP, SSOP, or TSSOP	LCCC		
GND	8	10	—	Ground Pin
OE	13	17	I	Output Enable
Q _A	15	19	O	Q _A Output
Q _B	1	2	O	Q _B Output
Q _C	2	3	O	Q _C Output
Q _D	3	4	O	Q _D Output
Q _E	4	5	O	Q _E Output
Q _F	5	7	O	Q _F Output
Q _G	6	8	O	Q _G Output
Q _H	7	9	O	Q _H Output
Q _{H'}	9	12	O	Q _{H'} Output
RCLK	12	14	I	RCLK Input
SER	14	18	I	SER Input
SRCLK	11	14	I	SRCLK Input
SRCLR	10	13	I	SRCLR Input
NC	—	1	—	No Connection
		16		
		11		
		16		
V _{CC}	—	20	—	Power Pin

80.00

60.00

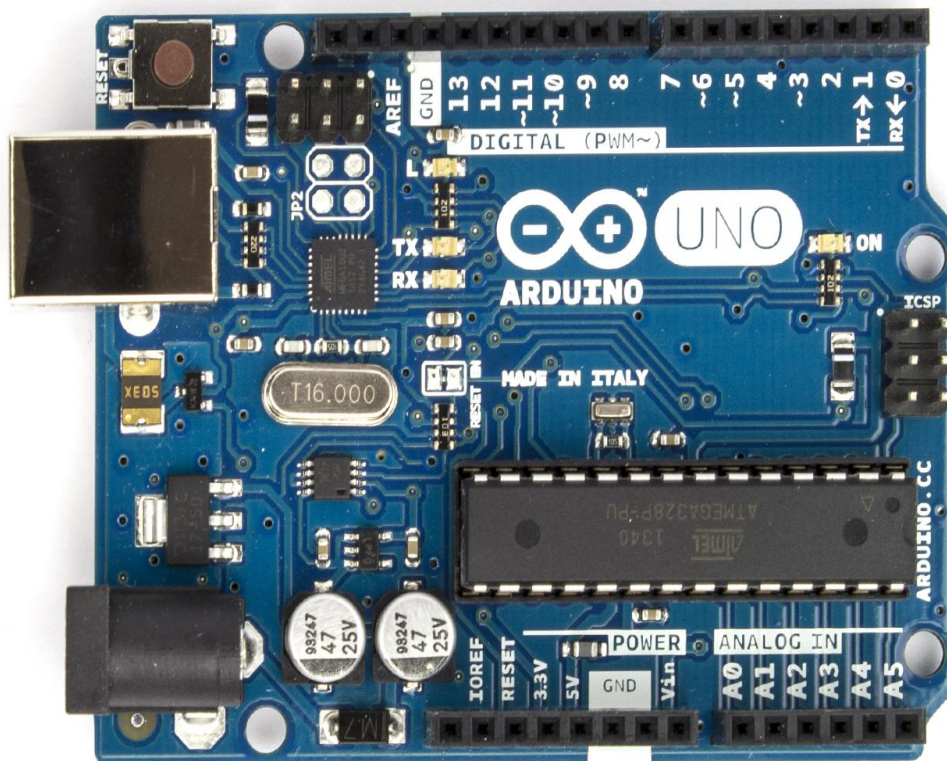
2.83

GREEN SOLDERMASK



ARDUINO

Arduino is an open source programmable circuit board that can be integrated into a wide variety of makerspace projects both simple and complex. This board contains a microcontroller which is able to be programmed to sense and control objects in the physical world. By responding to sensors and inputs, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays. Because of its flexibility and low cost, Arduino has become a very popular choice for makers and makerspaces looking to create interactive hardware projects. Arduino was introduced back in 2005 in Italy by Massimo Banzi as a way for nonengineers to have access to a low cost, simple tool for creating hardware projects. Since the board is open-source, it is released under a Creative Commons license which allows anyone to produce their own board.



BOARD BREAKDOWN :-

Here are the components that make up an Arduino board and what each of their functions are.

1. **Reset Button** – This will restart any code that is loaded to the Arduino board
2. **AREF** – Stands for “Analog Reference” and is used to set an external reference voltage
3. **Ground Pin** – There are a few ground pins on the Arduino and they all work the same
4. **Digital Input/Output** – Pins 0-13 can be used for digital input or output
5. **PWM** – The pins marked with the (~) symbol can simulate analog output
6. **USB Connection** – Used for powering up your Arduino and uploading sketches
7. **TX/RX** – Transmit and receive data indication LEDs
8. **ATmega Microcontroller** – This is the brains and is where the programs are stored
9. **Power LED Indicator** – This LED lights up anytime the board is plugged in a power source
10. **Voltage Regulator** – This controls the amount of voltage going into the Arduino board
11. **DC Power Barrel Jack** – This is used for powering your Arduino with a power supply
12. **3.3V Pin** – This pin supplies 3.3 volts of power to your projects
13. **5V Pin** – This pin supplies 5 volts of power to your projects
14. **Ground Pins** – There are a few ground pins on the Arduino and they all work the same
15. **Analog Pins** – These pins can read the signal from an analog sensor and convert it to digital

REFERENCE

<https://www.autodesk.com/products/eagle/blog/schematic-basics-part-1/>

<https://www.4pcb.com/media/presentation-how-to-build-pcb.pdf>

<https://www.autodesk.com/products/eagle/blog/every-layer-explained-autodesk-eagle/>

<https://www.build-electronic-circuits.com/smd-soldering/>

https://cei-lab.github.io/ece3400-2017/tutorials/Eagle/Eagle_Example.html

<https://cei-lab.github.io/ece3400-2017/tutorials/PCB>