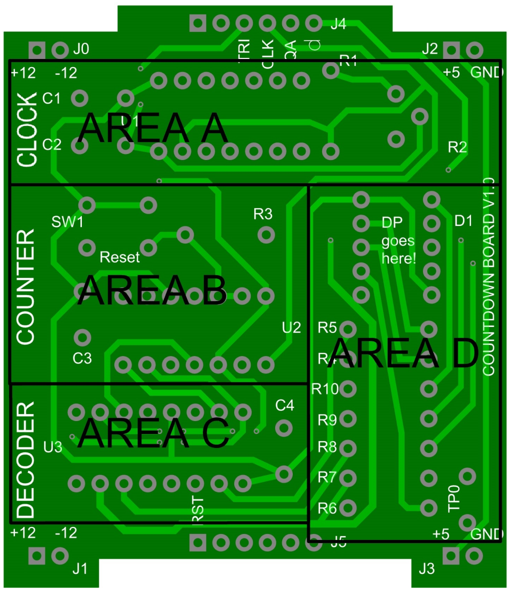
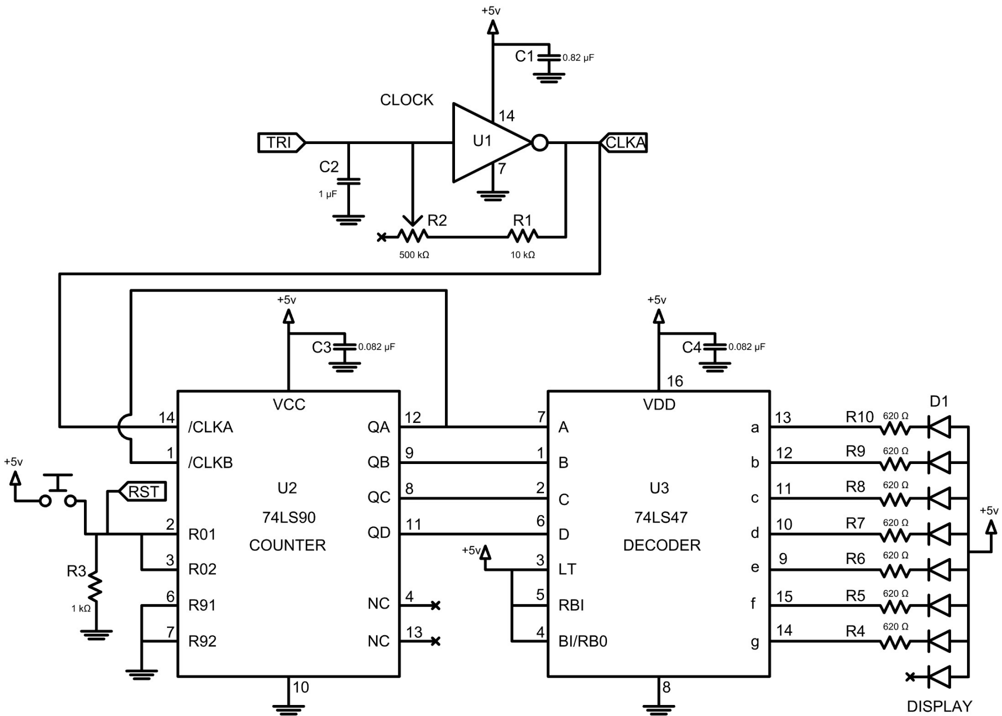
Massachusetts Institute of Technology
Electromechanical Systems Group
Electronics First: COUNTERS

Board and **Circuit Schematic:**





Materials:

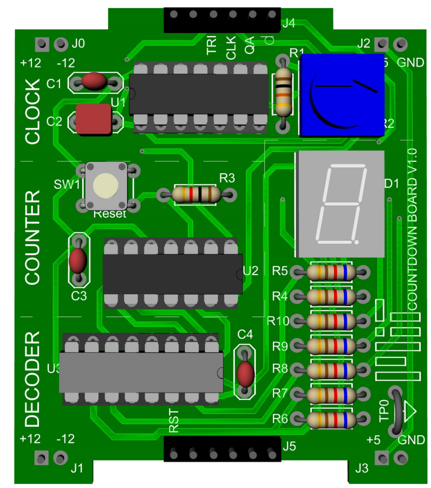
Part	Quantity	Part Number	Vendor	Part	Quantity	Part Number	Vendor
Breadboard	1	$377-2646-ND$	DigiKey	$74LS47$ Decoder	1	$296-3712-5-ND$	DigiKey
Breadboard Power Supply Module	1	-	-	7-Segment LED Display	1	$160-1525-5-ND$	Amazon
Multimeter	1	$MN35-ND$	DigiKey	C_1, C_3, C_4 0.082 μF Capacitor	3	$399-13945-ND$	DigiKey
6-pin Header Pins	2	$609-3263-ND$	DigiKey	C_2 0.082 μF Capacitor	3	$445-175543-ND$	DigiKey
2-pin Rail Connector Pins	4	$952-2262-ND$	DigiKey	R_1 100 kΩ Resistor	1	$A105979CT-ND$	DigiKey
Micro USB Power Cable	1	$102-4123-ND$	DigiKey	R_2 1 MΩ Resistor	1	$3386P-1-105TLF-ND$	DigiKey
Soldering Iron and Solder Wire	1 ea	$T0052918199N-ND$	DigiKey	R_3 10 kΩ Resistor	1	$CF14JT10K0CT-ND$	DigiKey
SPST NO Switch	1	$679-2428-ND$	DigiKey	R_4-R_{10} 620 Ω Resistor	1	$CF14JT620RCT-ND$	DigiKey
$74HC14$ Hex Inverter	1	$296-1577-5-ND$	DigiKey	14-Pin DIP Socket	2	$ED3045-5-ND$	DigiKey
$74LS90$ Counter	1	$296-34031-5-ND$	DigiKey	16-Pin DIP Socket	1	$3ED3046-5-ND$	DigiKey

The Build:

Counters are digital circuits that track how many times a specific event or process has occurred. Counters track and totalize signal pulses. For example, an LED and a photoresistor could create a detectable beam of light across a manufacturing line. As products like bottles pass between the LED and the photoresistor, they “break” the beam and create a pulse at the photoresistor circuit indicating that a bottle has passed. A counter might add up the number of bottles that have passed for packaging. In this lab, you will build a counter circuit, complete with a clock to produce signal pulses and a decoder and LED display to show the count. This circuit is essentially an “egg timer” that counts out an interval of time. The display could serve as a kitchen timer, or count out a model rocket launch sequence, or serve for many other timing applications. As the counter detects each clock pulse, the number on the display increases until it reaches its maximum value, at which point the count resets and starts again from zero.

Counters quantify our experience of time and space. The swing of a pendulum, the interval between drops of water falling from a small hole in a filled bucket, and the interval between the ebb and flow of the ocean tide are all events that occur over roughly fixed time periods. Counting these events creates a clock, with varying accuracy depending on the source of the events. Similarly, counting can totalize distance. The full rotations of a wheel of known circumference can be counted and totalized to measure the distance between two points. *Counter* hardware totalizes events like pendulum swings or wheel rotations.

Pendulum Clock and Measuring Wheel



Finished product of this build

Electronic counters add up the number of oscillations of an electronic signal. With appropriate additional circuitry, electronic counters can form very accurate timers and other associated measuring instruments. For example, an electronic counter can count the number of fixed periods of a fast electronic oscillator that occur during one period of a slower (less frequently oscillating or lower frequency) audio signal as measured by a microphone. The total number of fixed oscillator periods that occur within one period of the measured audio signal indicates the time period of the audio signal. The reciprocal of the time *period* is the audio *frequency*. An instrument for measuring frequency, sometimes called a frequency counter, can help tune musical instruments or enable many other scientific applications. In applications where the oscillations of a faster clock are used to time slower events, there is a clear value in having a fast reference clock that is not only high frequency but also very accurate and precise. The challenge of creating high quality oscillators and clocks has occupied designers for centuries. Solutions to timing problems have resulted in some of the most sophisticated and precise machines known.

An important design consideration when making or using a counter is *how* to count! That may seem silly or surprising, as we count every day, but there are actually many possibilities, each with different design implications for hardware. People typically count using a base-10, or *decimal* system. The base-10 system takes advantage of our natural counting hardware, hands with fingers, and represents numbers using successive powers of ten. As we count, each time we use up all our fingers, we "rollover," starting another count of 10 while making a rollover mark or record indicating that we have already recorded a full count of 10 and are ready to move on to 11, 12, and so forth. You could fold your toes, for example: one toe indicates a completed count of ten; two toes indicates twenty, three toes thirty, and so forth. In general, we represent numbers in base-10 with successive powers of 10. Consider, for example, the base-10 number 419. To represent 419 in powers of 10, we need nine "one counts" plus one "10 count" plus four "100 counts":

$$419 = (4 * 10^2) + (1 * 10^1) + (9 * 10^0)$$

Digital electronic systems represent information with switches that are either "on" or "off". It can therefore be convenient to design digital counters to count using *binary numbers*. A base-2 or *binary* system represents numbers with *bits* that are either 0 or 1. A decimal number "translated" into the binary system is represented by successive powers of 2. In binary, each successive bit to the left represents an increasing power of 2. For example, the decimal number 419*d* expressed in base-2 or as a binary number looks like this: 110100011*b*, i.e.,

Voltage	Binary	Decimal
LLLL	0000	0
LLLH	0001	1
LLHL	0010	2
LLHH	0011	3
LHLL	0100	4
LHLH	0101	5
LHHL	0110	6
LHHH	0111	7
HLLL	1000	8
HLLH	1001	9
HLHL	1010	10
HLHH	1011	11
HHLL	1100	12
HHLH	1101	13
HHHL	1110	14
HHHH	1111	15

Table of 4-Bit Binary Numbers

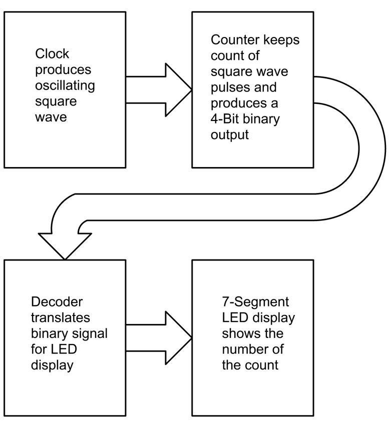
$$110100011b = (1*2^8) + (1*2^7) + (0*2^6) + (1*2^5) + (0*2^4) + (0*2^3) + (0*2^2) + (1*2^1) + (1*2^0)$$

or

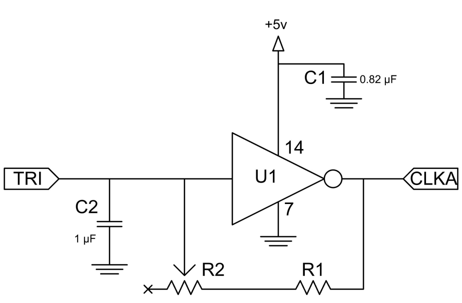
$$110100011b : 256 + 128 + 0 + 32 + 0 + 0 + 0 + 2 + 1 = 419d.$$

The *d* and *b* symbols are sometimes deployed to clarify whether a number is decimal or binary. Notice, for instance, that 101*b* (which is equivalent to 5*d*) is **not** the same number as 101*d*!

Using transistor hardware to represent binary numbers, any voltage falling within the "high" (H) voltage range represents a logical "1" or "on" for a bit, and any voltage in the "low" (L) range represents a logical "0" or "off" for the bit. More bits can be used to represent a wider range of numbers. For example, four bits taken together, each bit potentially implemented physically with one or more transistors, could represent any decimal number between 0 and 15. A collection of four bits is sometimes called a "nibble." A collection of eight bits is sometimes called a "byte." The counter you will use in this build is a four-bit or nibble-wide counter.



Overview of the Build Hardware



Clock Circuit

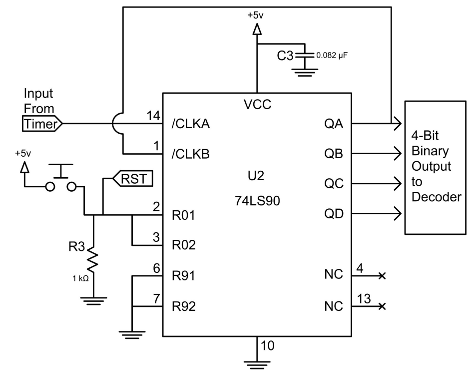
Theory of Operation and Predictions

THIS BUILD ASSEMBLES A CLOCK, A COUNTER, A DECODER, AND AN LED DISPLAY. The LED display is the "goal" of the build, as the display serves as our user interface or readout. We would like the single-digit LED display to show a count in sequence, beginning with a display of 0*d* and, if undisturbed, counting sequentially, 1*d*, 2*d*, and so forth until the display reads 9*d*, when it "rolls over" and starts counting up from 0*d* again. How does the build hardware achieve this functionality?

First, the *clock* produces an oscillating signal, a square wave, which serves as the "heartbeat" for the system: each period of the clock will cause the counter to advance one count. The clock for this build is an *RC Relaxation Oscillator* circuit constructed with a 74HC14 Hex Schmitt Trigger Inverter, as you may have already seen. Altering the setting of a potentiometer, which serves as variable resistor R_2 in the clock circuit, adjusts the rate of the count. The fixed resistor R_1 in series with the variable resistor sets a minimum resistance in the RC network, and therefore a maximum frequency.

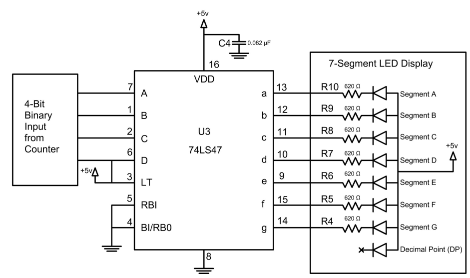
The *counter*, a 74LS90 integrated circuit (IC), offers four output bits, or one nibble. In principle, a binary nibble can range over any value from 0000*b* (0*d*) to 1111*b* (15*d*). However, for this build, we are only interested in counting from

0000b to 1001b, since the single-digit LED display will not display a 10d or 11d or higher decimal counts. The 74LS90 is a special counter that will count in "binary-coded decimal," or *BCD*. That is, the output of the counter is available on four IC pins as a binary nibble that starts at 0000b and counts 0001b, 0010b, ... 1001b, 0000b, and so on. In short, the BCD counter counts numbers the way we might in base-10, but uses a binary nibble to keep track of the count. A "rollover" resets the counting to start again at zero; however, our build also offers an alternative method for resetting the count. The push-button switch connected to pin 2 of the counter "pulls up" the $R01/R02$ pins to $5V$, resetting each of the 4 binary outputs to 0 when the user presses the button. While the button is depressed, the counting freezes and the display will lock to show 0d. Counting commences when the button is released, allowing the user to control the start of the timing.



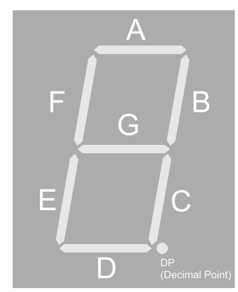
Counter Circuit

Finally, the *decoder* drives the *LED display* to produce a useful visual timing display for the user. Four simple LED's could display the status of each of the four bits of the counter directly. This display approach would show the pattern of the nibble corresponding to the current BCD count. For example, the "first light" or the light connected to the least-significant bit (LSB) would be "on" and all other lights would be off when the count reached 0001b. This display choice would require a user to "translate" the binary count to a decimal number in their mind, not necessarily the most convenient design.



Decoder Circuit

Instead, this build uses a 74LS47 decoder. This integrated circuit was designed to work with a widely-available 7-segment LED display containing 7 individual LED's arranged as linear segments (labeled A-G). Some displays also include a circular decimal point (DP), not used here. The seven segment LED's in the display should be arranged in a "common anode" configuration for the decoder IC to work correctly. This means that the "high side" of each LED is connected to a common point where a positive voltage is always applied. The outputs of the 74LS74 decoder IC are *active low*. When a segment is supposed to be "on," or asserted, that particular decoder IC output will go low, passing current through the LED in the segment and a series current limiting resistor.



LED Display with Segment Labels

The decoder IC takes the binary nibble from the counter and creates an output with the necessary signals to activate the individual LED segments on the display. When properly connected, the decoder will drive the segments of the LED display to create a glowing number from 0 to 9 according to the input nibble. Imagine, for example, that the counter output was 0011b, corresponding to the decimal number 3d. The properly connected decoder IC would activate segments A, B, C, D and G, while leaving segments F and E off.

Let's make some predictions:

1.) Assume the potentiometer knob starts in the middle of its travel when power is applied to the circuit. If you turned that knob all the way clockwise for maximum resistance for R_2 will the LED display count faster or slower?

Faster Slower

(Circle one)

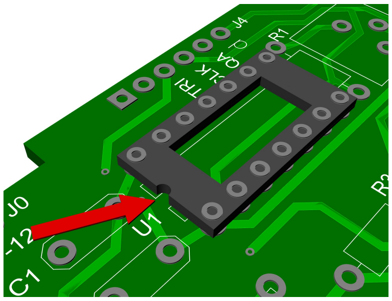
Prediction 1

2.) Now assume you turned the turned that knob all the way counterclockwise; would you expect the LED display to count faster or slower?

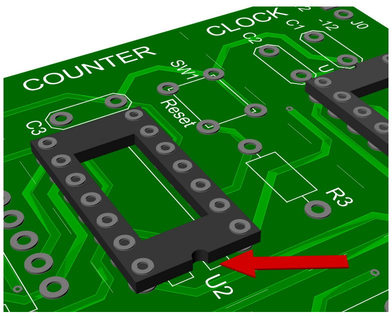
Faster Slower

(Circle one)

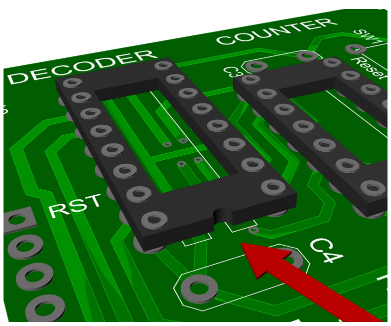
Prediction 2



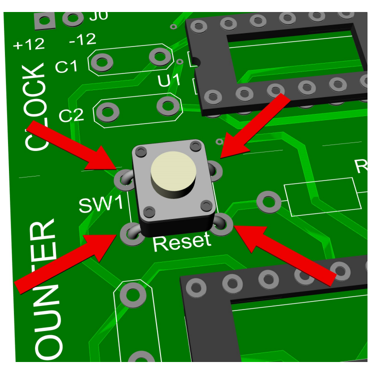
Clock DIP Socket Location



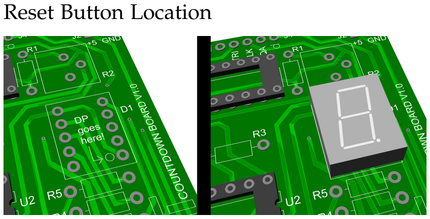
Counter DIP Socket Location



Decoder DIP Socket Location



Reset Button Location



7-Segment LED Display Location

Assembly

THIS BOARD USES SOCKETS FOR THREE INTEGRATED CIRCUITS: A CLOCK, A COUNTER, AND A DECODER. The user controls the board function using a push-button and a potentiometer. The board has been divided into separate areas shown on the cover page.

DIP Socket Installations

Install sockets but do NOT insert the integrated circuits yet.

Install the 14-pin DIP socket for the $74HC14$ hex inverter in **AREA A**. Orientation matters - ensure that the semi-circular notch at the top of the socket lines up with the corresponding notch illustrated on **U1** of the board. The pads for **U1** are arranged as two rows of 7 contact points. Press the 14 socket pins through the corresponding pads on the board. Make good solder connections between each of the 14 socket pins and the metal board contacts located on the underside of the circuit board.

Repeat this process for the $74LS90$ counter 14-pin DIP socket in **AREA B**. Align, insert, and solder this socket into **U2**. Finally, insert the $74LS47$ decoder socket in **AREA C**. Note that this socket has 16 pins - install and solder it into **U3**.

Reset Button Installation

Insert the reset button in **AREA B**. There are 4 pads for the switch in **SW1**. The switch is not polarized, and the pins are in a rectangular layout; that is, with wider spacing between the pins on two of the four sides. Carefully fit the “wide sides” in the “widely spaced” holes so that the 4 metal legs align with the holes. Push the legs through and make soldered joints on the underside of the board.

LED Display Installation

Add the 7-segment LED display in **AREA D**. There are two vertical rows of 5 pads located in **D1**. Position the 10 pins of the LED display so that they align with these holes. *Orientation matters!* Ensure that the decimal point lines up with the "DP goes here" labelling on the board. Push the pins through the contacts and make soldered joints on the underside of the board.

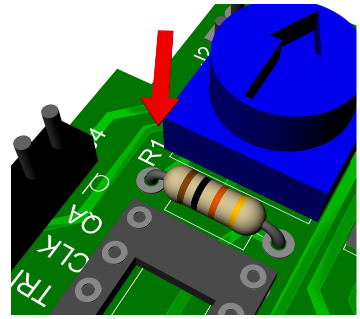
Resistor Installations

Add the 10 kΩ resistor in **AREA A**. This resistor forms a voltage divider with the variable resistance potentiometer, allowing you to adjust the clock output frequency. Using the resistor band color code, identify and install the R_1 resistor into the PCB location labeled for R_1. Make soldered connections at the proper locations on the opposite side of the board.

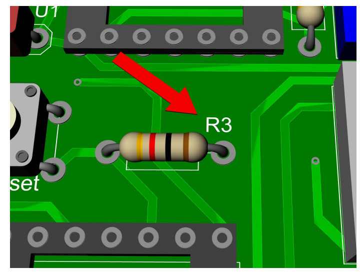
Add the 1 kΩ resistor in **AREA B**. This resistor is a "pull down" resistor that grounds pins 2 and 3 on the 74LS90 counter until the reset button SW1 is pressed. Install and solder the 1 kΩ resistor into the PCB slot labeled for R_3.

Install the seven 620 Ω resistors in **AREA D**. These current limiting resistors protect the LEDs in the display. Install and solder a 620 Ω resistor into each of the seven slots labeled R_4-R_{10}. Save your wire cuttings for the next step, installing the ground clip.

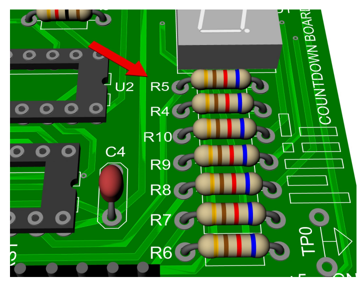
Add the potentiometer in **AREA A**. Install the blue, rotating knob potentiometer into the section labeled R_2. Ensure its three triangularly-positioned pins are aligned with the three corresponding points of contact. Solder the pins.



Resistor R_1 and Potentiometer R_2



1 kΩ Resistor R_3 Location



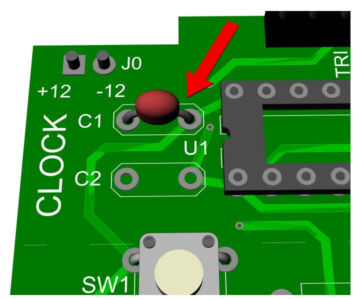
620 Ω Resistors R_4-R_{10}

Capacitor Installations

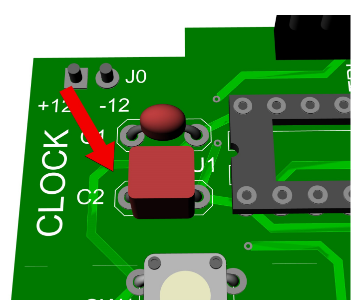
Add the 74HC14 Bypass Capacitor in **AREA A**. This capacitor effectively "smooths out" the power supply, ensuring stable operation of the 74HC14 IC. Properly install and solder the C_1 capacitor into the C_1 location on the board. Pull the leads through until the capacitor is on the surface of the board. Solder in the proper locations beneath the board and trim the leads.

Add the RC network timing capacitor in **AREA A**. Together with the variable resistor, this capacitor sets the frequency of the clock. Install and solder the C_2 capacitor into the C_2 location on the board.

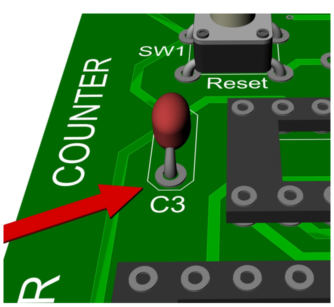
Add the 74LS90 counter Bypass Capacitor in **AREA B**. Again, this capacitor protects the operation of the 74LS90 IC from power supply fluctuations. Properly install and solder the C_3 capacitor into the C_3 board location. Also add the 74LS47 decoder bypass capacitor in **AREA C**. Install and solder the C_4 capacitor into the C_4 board location.



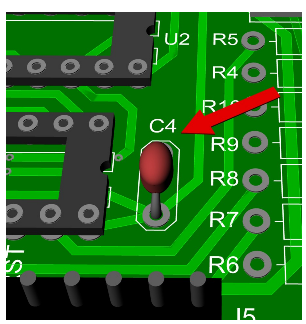
74HC14 Bypass Capacitor C_1 Location



Timing Capacitor C_2 Location



74LS90 Bypass Capacitor C_3 Location

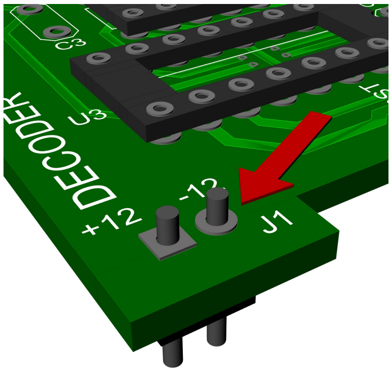


74LS47 Bypass Capacitor C_4 Location

Pin Installation

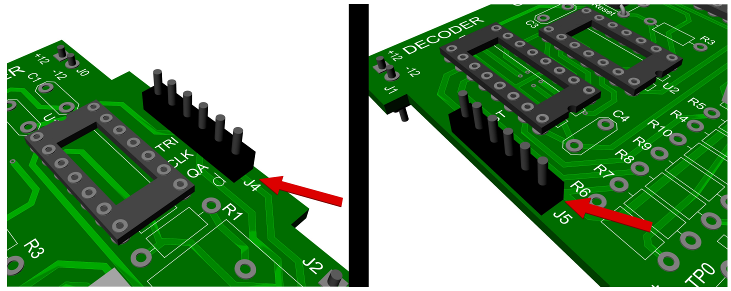
Install one 6-pin header into **J4** at the top of your board, and the other 6-pin header into **J5** at the bottom of the board. Insert the pins into the pads and make soldered joints on the underside of the board.

Next, install 2-pin rail connectors into **J0, J1, J2, and J3**. These are installed in the reverse direction, with the base of the pin on the underside of the board, and the location of the soldered joint on the top of the board. *BEFORE SOLDERING*,

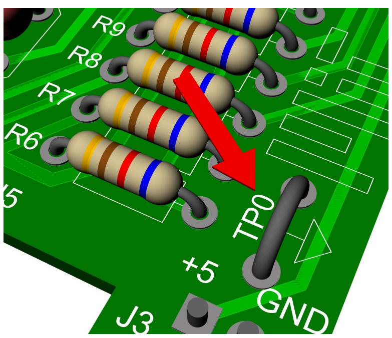


Rail Connector Pin Installation

insert the bottom of the pins into your breadboard, and *then* make your soldered connections on the top of the circuit board to ensure proper alignment.



Header Pin Installation



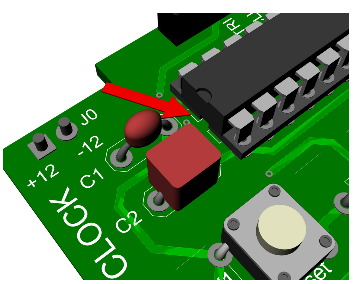
Ground Clip Location

Ground Clip Construction

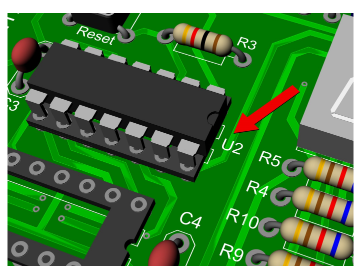
Construct the ground clip in **AREA D**. Take one of the wire cuttings you saved earlier and bend it into an arc, inserting each end into the contact points in **TP0**. Trim the length of the wire as necessary. Make soldered joints on the bottom of the board. This is a ground clip.

Integrated Circuit Installations

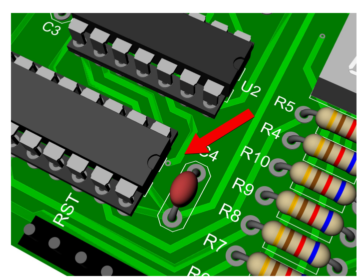
Once all of your soldered joints are completely cooled, carefully install the 74HC14 hex inverter, the 74LS90 counter, and the 74LS47 decoder into their respective sockets. Be sure to put the right IC in the right socket! *Orientation matters!* Ensure that the notch on the topside of each component body aligns with the notch on the corresponding DIP socket.



74*HC*14 Hex Inverter Orientation



74*LS*90 Counter Orientation



74*LS*47 Decoder Orientation

Testing

Carefully insert your counter board into the breadboard in line with the power supply module. Ensure that the pins of both boards are fully inserted into the breadboard contact holes. Ensure that your power cord is properly fitted into the power port on the power supply module. Check that the indicator lights are illuminated. If so, the LED display on the counter board should be counting.

Starting the Count

Supplying power to the board should start the counting process. You should see the LED display flashing the numbers 0-9 in sequence.
1.) Turn the knob of your potentiometer to the middle position and observe the count. Now turn it as far as it will go in the clockwise direction. Was the speed of the count faster or slower? Does this concur with your prediction?

Faster Slower
(Circle one)

Test 1 - Check against Prediction 1

2.) Now turn the knob of your potentiometer as far as it will go in the counter-clockwise direction. Did the count speed up or slow down? Does this concur with your prediction?

Faster Slower
(Circle one)

Test 2 - Check against Prediction 2

3.) Some questions to consider:

- Press and hold the reset button. Now release the reset button. Describe the behavior of the board when the button is pressed, and when released. Explain.
- Advanced questions: Can you think of a way to make the board count down from 9 instead of up from 0? If not, what 74XX part number or numbers could be used to make a new but similar circuit that could count either up or down?
- Very advanced question: Read about the 7400 quad-NAND gate IC. How could you use this IC to look at the counter output Q_a, Q_b, Q_c, and Q_d on 74LS90 pins 12, 9, 8, and 11, respective, and create a new signal that went "high" only when the count was $9d$ or $1001b$?
- Very advanced question: The 74LS90 is a "ripple" counter, its outputs are not synchronous. That is, all of the outputs are not guaranteed to change to the new count value at exactly the same instant in time; there could be a small but measureable delay before the four outputs stabilize. Could there be a problem with using the 7400 NAND gate as contemplated in the previous question?