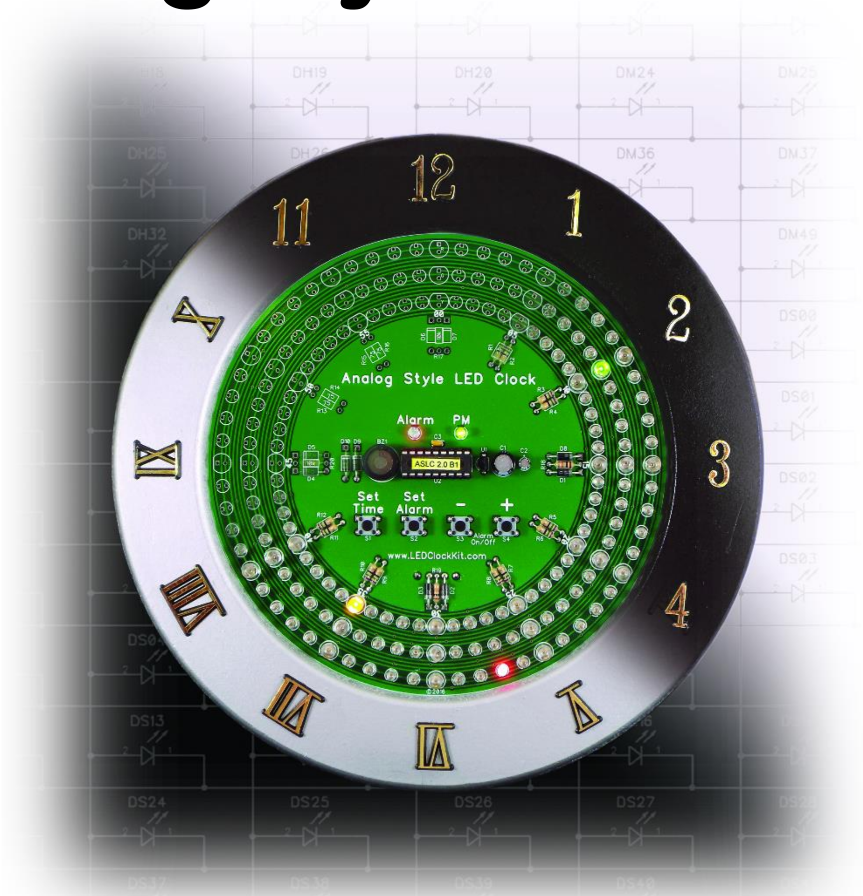
Analog Style LED Clock



Operation and Assembly Manual

For use with PCB Rev 2.1

Manual version 2.1d, for use with PCB revision 2.1, Software version 2.0.0.
This manual can be found at www.LEDClockKit.com

Contents

Introduction

Thank you for purchasing the Analog Style LED Clock kit! Although there are a lot of parts, they are all through-hole components – easy to work with and solder. With a little patience, you should have no problem making this clock, which you can proudly hang on your wall and enjoy for years to come!

Figure 1 – Completed Clock

This clock uses LEDs to simulate the hands of an analog clock. The second hand, minute hand and hour hand are each represented as 60 LEDs for each hand, arranged in concentric circles. The clock features an alarm, a PM indicator and keeps time using the 50 or 60Hz frequency from the electric utility. Four pushbuttons are used to operate the clock.

This clock was originally my Electrical Engineering Senior Project. A design goal was to use the fewest number of components and a microcontroller with the smallest pin count as possible. The clock was designed so than an 18-pin microcontroller could run the entire clock.

My original Senior Project proposal did not include the alarm nor did it include the PM and Alarm LEDs. It also used two pushbuttons instead of four. My instructor accepted my proposal, but on the condition that I add an alarm function. And I figured out a way to add the alarm, two more LEDs, and two more buttons *without adding to the microcontroller pin count!*

A total of 182 LEDs, a buzzer, four pushbuttons and the monitoring of electric utility frequency all interface to an 18-pin IC with no additional logic circuits or drivers. This seemingly impossible feat is explained in detail in the Theory of Operation section.

Operation

Plugging in the Clock

When the clock is first plugged in, the software version is displayed by lighting three LEDs. See the Specifications section for how to read the software version. After displaying the software version for two seconds, the clock starts keeping time starting at 12:00 midnight. The alarm is also set to 12:00 midnight upon power up, but the alarm is not turned on. Whenever power is interrupted, the clock always returns to this state.

Reading the Time

This clock is read like a traditional analog clock. The outer ring of Red LEDs represents Seconds, the middle ring of Green LEDs represents Minutes, and the inner ring of Yellow LEDs represents Hours. Only one LED in each ring will be lit at one time. When the Second LED reaches the 12 o'clock position, the Minute LED will advance one LED. Every time the Minute LED advances twelve times, the Hour LED advances once. This way, the Hour LED will advance five times during the course of an hour. Because the lit Hour LED can be between numbers, just like the hour hand on an analog clock can be between numbers, to read the hour you look at the number immediately counterclockwise from that LED.

Setting the Time

To set the time, press and hold the [Set Time] button. The clock continues to run until the [+] or [-] button is pressed, therefore if the [Set Time] button is released before pressing either of the other buttons, nothing changes. While holding the [Set Time] button down, press [+] to go forward one minute or [-] to go back one minute. At this time the Second LED disappears and the clock stops running until the [Set Time] button is released. If the [+] or [-] button is held down, the time will continue to change, slowly at first, but picking up speed every couple of seconds. At top speed, it will cycle through an entire 24 hours in less than a minute. When the clock gets close to the desired time, release the [+] or [-] button while continuing to hold the [Set Time] button. Then use the [+] and [-] buttons to get to the exact time. The clock will not start running until the [Set Time] button is released, at which time the Second LED comes back on starting at 00. This allows the clock to be synchronized exactly to another clock.

Setting and Using the Alarm

Setting the alarm works the same way as setting the time, except that the [Set Alarm] button is held down. The [+] and [-] buttons will go forward or back one minute for each press, or will accelerate to the desired time if held down, exactly like it does for setting the time. Once you have set the desired alarm time, release the [Set Alarm] button. Setting the alarm does not turn it on. The alarm is turned on by momentarily pressing the [+] and [-] buttons together at which time the Alarm LED will light. Pressing and releasing the [+] and [-] buttons again will turn off the alarm and the Alarm LED will go off.

If the alarm is turned on, when the time reaches the alarm time, the buzzer will sound and the Alarm LED will flash. Pressing any button will snooze the alarm: the alarm will quit sounding for 10 minutes and the Alarm LED will flicker, indicating that the alarm has been snoozed. After 10 minutes the alarm will sound again. To silence the alarm, press the [+] and [-] buttons together to turn the alarm off. Also, setting the alarm to a different time will silence the alarm until the newly set time arrives. If the alarm is not snoozed or turned off, it will silence after one hour, however, this does not turn off the alarm; it will sound again when the time displayed again reaches the alarm time.

Assembly Tips and Guidelines

There are a lot of parts in this kit. The components are all through-hole, so they are fairly easy to place and solder. If you are careful to insert the parts in the correct orientation and exercise good soldering techniques, you should have no problem building this clock. It should take about three or four hours to install and solder all the parts to the PC Board and about another hour to paint and apply the numbers to the frame of the clock, not including the drying time of the paint.

General Soldering Tips

There are over 200 parts to be soldered to the PC Board. It is important that each and every solder joint make a good connection. A single bad connection can cause your clock to not work as expected. In fact, if your clock does not work correctly, the first thing to do is look at the solder connections in the affected circuit.

The solder included in the kit is RoHS compliant, that is, it does not contain lead. The RoHS directive, enforced in the European Union and some other regions in the world, restricts the use of certain hazardous substances in electronic equipment. If you don't live in such a region, you can use your own lead-based solder; some people prefer working with it over RoHS solder.

It is beyond the scope of this manual to teach you how to solder. However, here are some tips for making a good solder connection:

- Use an appropriately sized tip. A small to medium chisel shaped tip should work well. If it is too large, you will tend to bridge solder across joints, creating shorts. If it is too small, it will be harder to get the solder joint sufficiently heated.
- Make sure there is a little solder on the tip of your iron. This helps transfer the heat from your iron to the joint.
- Do your soldering on the opposite side of the board from the side that you installed the component.
- Make sure the soldering iron is in contact with both the pad on the PC Board and the lead that is being soldered. Feed some solder into the spot where the iron, pad and lead come together.
- Do not press with the soldering iron. It can cause damage to the board or circuit traces. This can create problems that are particularly challenging to troubleshoot.
- Do not heat the joint for more than a few seconds. Heating the joint for an extended amount of time can damage the board or the component being soldered.
- Add enough solder so it wicks down into the hole. You should be able to see the solder in the hole on the front side of the board.
- Do not use too much solder. This is to prevent bridging solder to a nearby connection. Also, it is more difficult to determine if there is a good connection if there is too much solder.
- Do not allow the parts to move until the solder has cooled and become solid.
- A good solder connection will be smooth and have a concave look to it. It should be evident that the solder is sticking to both the pad and the lead. If you use too much solder, the joint will look convex and it will be harder to determine if the solder connection is good.

Some solder wick has been provided to aid in the removal of excess solder. It is also helpful in clearing holes of solder if a part has to be replaced. To use, place some wick over the solder to be removed and place the tip of the soldering iron on top of the wick. Avoid the temptation to apply pressure with the soldering iron – let the heat do the work.

Component Orientation

Most of the parts in this kit are polarized – that is, they must be installed in the correct direction. Failing to do so can cause unexpected results – most certainly, the clock will not work as intended. In addition, it can damage the components.

The resistors are not polarized. However, it is recommended that you install them in the same orientation around the clock, with the same colored stripe toward the outside. This can aid in spotting an incorrect resistor value, and since the components of the clock are visible, it adds to the symmetry of the design.

Required Tools

See Figure 2 for the tools that are used to build and test the PC Board.

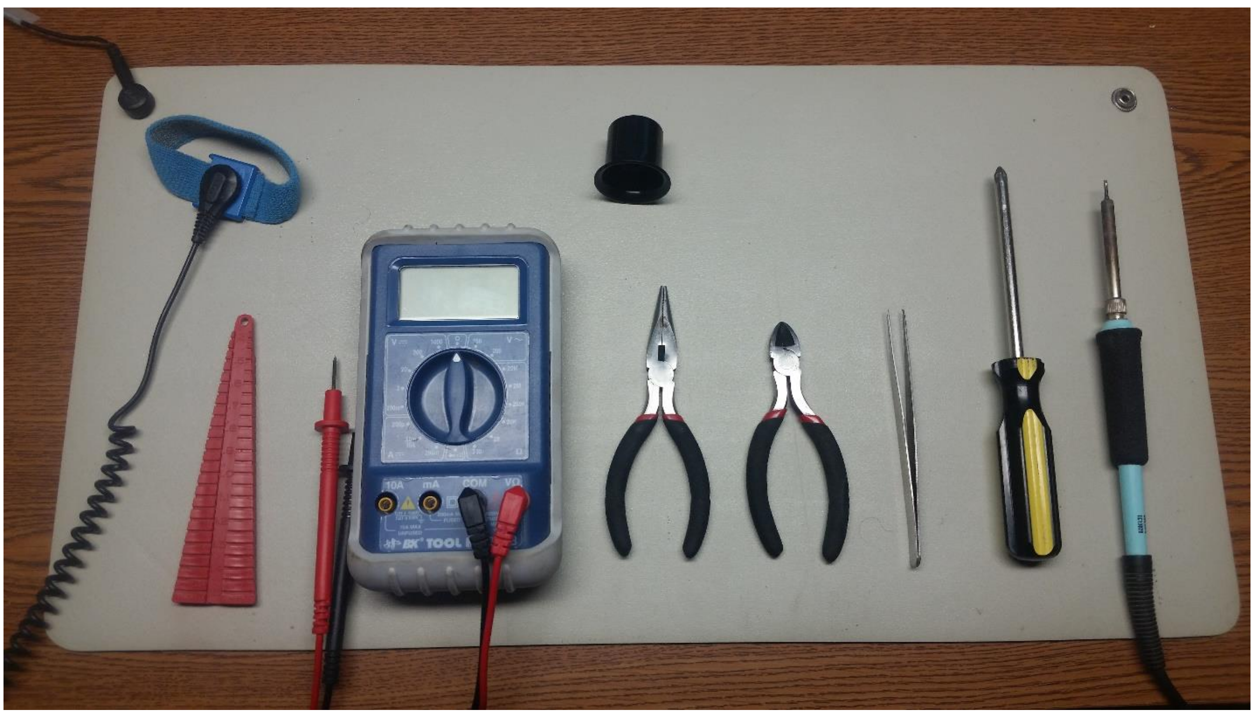


Figure 2 – Tools used to build and test the PC Board

- ESD Mat – Though not strictly necessary, assembling the kit on an ESD Mat will help prevent possible damage to components due to electrostatic discharge. On the mat from left to right:
- ESD Wrist Strap – This will prevent your body from building up a charge that can damage components. If you do not have these first two items, you can reduce the risk of component damage by touching something that is grounded prior to handling parts; minimizing your movement; avoiding the handling of components by the leads, especially the microcontroller; and handling the PC Board only by the edges.
- Axial Lead Bender – (optional) Useful for bending the leads of resistors and diodes.

- Multi-meter – Used to verify the power supply operation prior to inserting the microcontroller. It is also indispensable for troubleshooting, and can be used to safely light an LED to determine its color.
- Eye Loupe – Great for inspecting for bad solder joints and solder bridges.
- Needle Nose Pliers
- Wire Cutters
- Tweezers – Used for handling parts and bending leads.
- #2 Phillips Screwdriver – Used to mount the PC Board to frame.
- Soldering Iron – I find that a small to medium chisel tip works best for soldering through-hole components.

These are the basic tools. You may find other tools and supplies to be helpful. Some examples include IC puller, solder sucker, wire strippers, acid brush, alcohol, oscilloscope, and logic probe.

Bill of Materials and Parts Identification

Before building the clock, now would be a good time to go over the bill of materials and identify the parts that are in the kit (Figure 3). You'll find that some extra parts have been included in case of loss or damage. Please note that this list is for the full kit. Partial "component" or "barebones" kits will not contain all the parts listed. See the product description of the kit purchased for more details.

Figure 3 –The kit

Compare what you have with Table 1. It is important that you do not open any of the bags at this time. Wait until you are ready to use the parts. This is particularly true concerning the LEDs. Unlit, they all look the same – the only way to tell what color light they produce, is to apply power and light them up. Fortunately this can be easily and safely accomplished using a multi-meter. See the Assembling the PC Board section, Preparing to install the LEDs subsection, for details.

If you did not buy a kit and would like to source your own parts, you can use Table 1 as your shopping list. It may be tempting to use different LEDs than what are listed, especially if you want to experiment with different colors. However, for the LEDs to work properly in the circuit, it is important that a proper match of LED characteristics is maintained. Of particular importance are the forward voltage drop and the luminance of the LEDs. Keep in mind that the LEDs listed have been tested to work in the LED matrix. If you want to experiment with other LEDs, you assume the risk that they may not work as expected.

Qty	Reference	Source	Part Number	Manufacturer	Description
1		Jameco	100061	Jameco Reliapro	AC/AC Wall Transformer, 9 VAC
1	J1	Jameco	2114600	Jameco Valuepro	Cable Assembly, 2.1mm x 5.5mm
1	BZ1	Digi-Key	668-1465-ND	PUI Audio	Transducer, 2.048 kHz
1	C1	Digi-Key	493-14503-ND	Nichicon	Electrolytic Capacitor, 100µF
1	C2	Digi-Key	493-5954-ND	Nichicon	Electrolytic Capacitor, 1µF
1	C3	Digi-Key	478-4852-ND	AVX	Ceramic Capacitor, 0.01µF
10	D1-10	Digi-Key	1N4001-TPMSCT-ND	Micro Commercial Co	Diode, 1N4001
12	DHxx	Digi-Key	160-1687-ND*	Lite-On	LED, Yellow, 5mm
49	DHxx, DA0x	Digi-Key	160-1664-ND*	Lite-On	LED, Yellow, 3mm
12	DMxx	Digi-Key	160-1947-ND*	Lite-On	LED, Green, 5mm
48	DMxx	Digi-Key	160-1659-ND*	Lite-On	LED, Green, 3mm
12	DSxx	Digi-Key	160-1682-ND*	Lite-On	LED, Red, 5mm
49	DSxx, DA0x	Digi-Key	160-1661-ND*	Lite-On	LED, Red, 3mm
16	R1-16	Digi-Key	10KQBK-ND	Yageo	Carbon Film Resistor, 75 Ohms
4	R17-20	Digi-Key	75QBK-ND	Yageo	Carbon Film Resistor, 10k Ohms
4	S1-4	Digi-Key	450-1650-ND	TE Connectivity	Pushbuttons
1	U1	Digi-Key	497-16173-1-ND	STMicroelectronics	Voltage Regulator, TO-92, +5V
1	U2 Socket	Digi-Key	ED3047-5-ND	On Shore Technology	IC DIP Socket
1	U2	Digi-Key	PIC16F628A-I/P-ND*	Microchip	8-bit Microcontroller – program yourself (or programmed part from LED Clock Kit)
1	U2	LED Clock Kit	ASLC-U2	LED Clock Kit	Programmed microcontroller (or unprogrammed part from Digi-Key)
1		LED Clock Kit	ASLC-PCB	LED Clock Kit	Bare PCB
1		LED Clock Kit	ASLC-FRAME	LED Clock Kit	Frame (includes 5/8" Arabic & Roman numeral sets and mounting hardware)
1		LED Clock Kit	ASLC-KIT	LED Clock Kit	Complete clock kit (Includes all parts listed above)

* Component has critical specs – do not substitute

Table 1 – Bill of Materials

Assembly Techniques

I'll outline the technique that I used in placing and soldering the parts, which worked really well. First, install the machine screws and nuts that are included in the kit onto the PCB as shown in Figure 4. You only need to snug them up finger tight. This allows the leads to dangle as the parts are stuffed into the board.

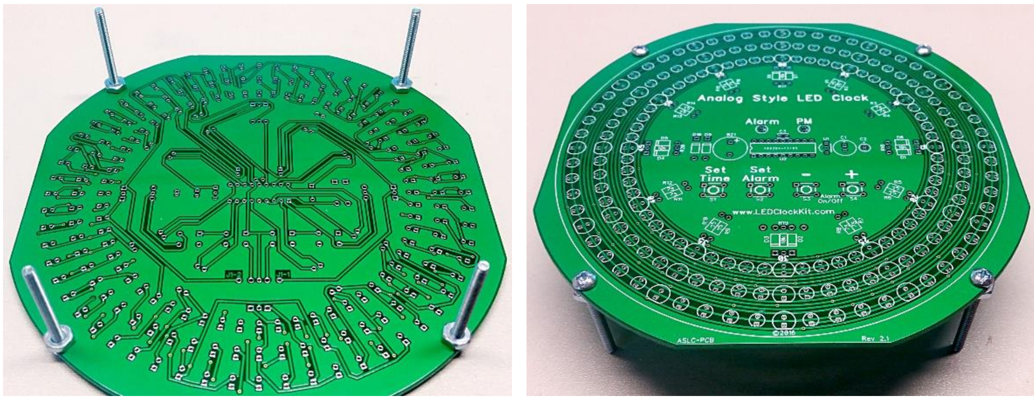


Figure 4 – Installing the machine screws and nuts

Components are stuffed and soldered in the order of component height. For this reason, it is important to install the parts in the order that is outlined in this manual. After inserting the components, a piece of cardboard, or something similar, is placed on top of the components. To solder the parts, the cardboard and PCB are turned over together, preserving the placement of the components. See Figure 5.

An alternate technique to using a piece of cardboard would be to bend the leads a little so the part cannot fall out when the board is flipped over. Just be careful that the bent lead does not touch an adjacent pad or via. Also, keep in mind that it is harder to remove a component that has had its lead bent before soldering.

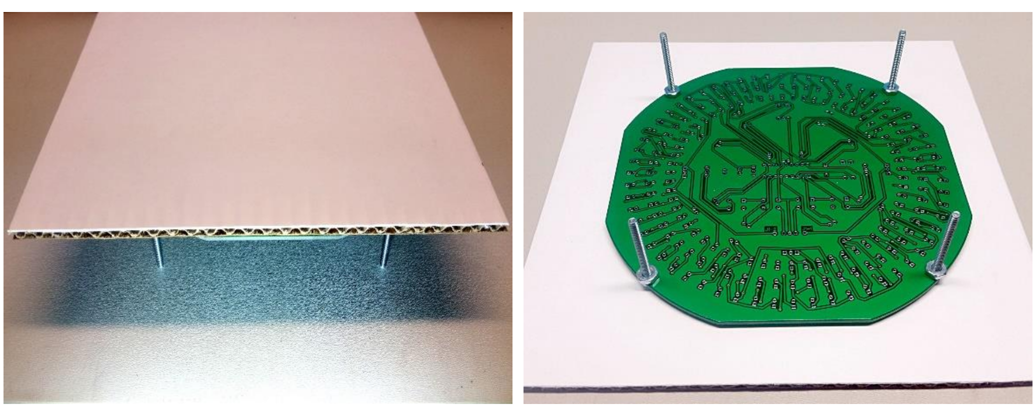


Figure 5 – Using cardboard to keep parts in place

While soldering, hold the solder between your index finger and thumb, while pressing down on the board with your remaining fingers, as shown in Figure 6. This will insure that the parts remain seated against the PCB. Avoid the temptation to press down on the board with the soldering iron.

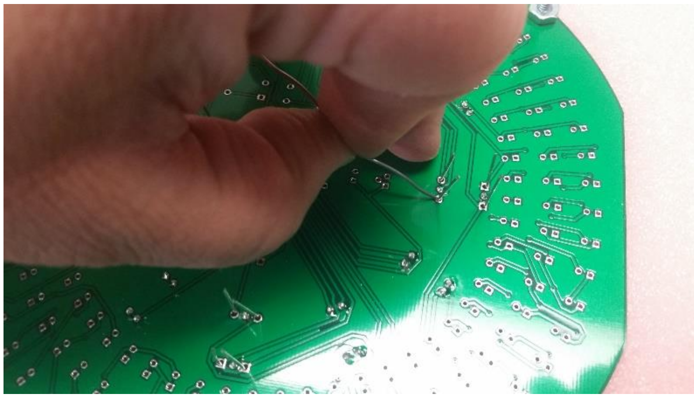


Figure 6
Press down on PCB to keep parts against board

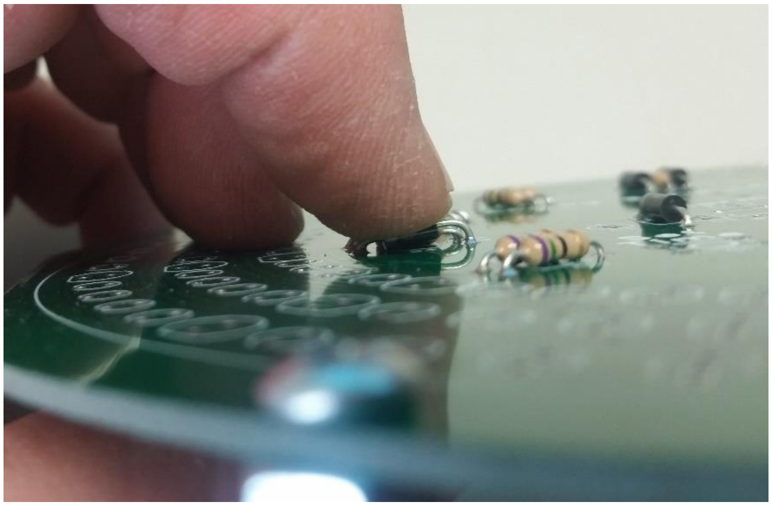


Figure 7 – Fixing parts that are not fully seated to PCB

After soldering each set of components, check that they are seated against the board. You may have to fix parts that are not fully seated by pressing down on them while reheating the solder on the other side. If the part gets too hot, you may need to use something besides your finger. A pencil eraser works well for this purpose.

After soldering the leads, be sure to clip them before moving on to the next set of parts. Clip the leads as short as possible without damaging the solder connection. The leads on the pushbuttons and on the socket for U2 do not require clipping.

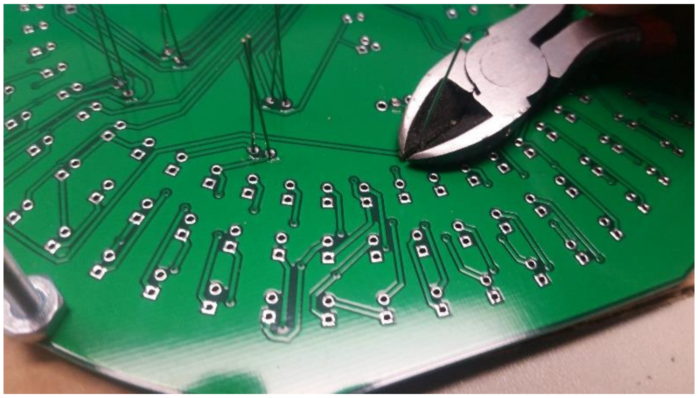


Figure 8 – Clip the leads as you go

If you have a lead bender, it makes forming the leads on the resistors and diodes easier. Place the component in the slot labeled .4 and bend the leads down as shown in Figure 9. Make sure the component is centered in the tool first.

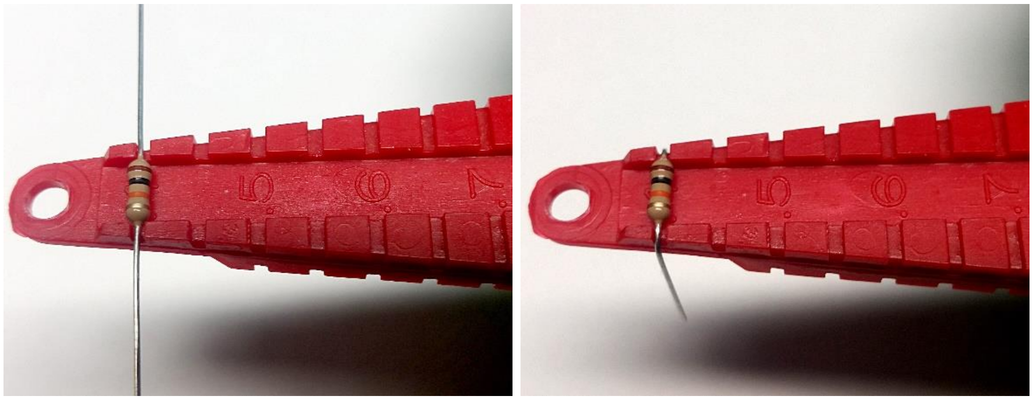


Figure 9 – Using a lead bender

If you don't have a lead bender, you can still get your leads bent perfectly using the technique shown in Figure 10. Place the resistor on the board with the body centered between its two pads. Grasp the lead with your tweezers so that the edge of the tweezers that are away from the component body is at the edge of the hole. Pull the part away from the board and bend the lead down at a 90° angle against the tweezers. Now, place the bent lead in the hole and grab the other lead in the same way, with the edge of the tweezers that is away from the component body placed at the edge of the hole. Carefully pull the component out of the hole and away from the board and bend the lead against the tweezers in the same direction as the other lead. The result is a part that fits perfectly on the board.

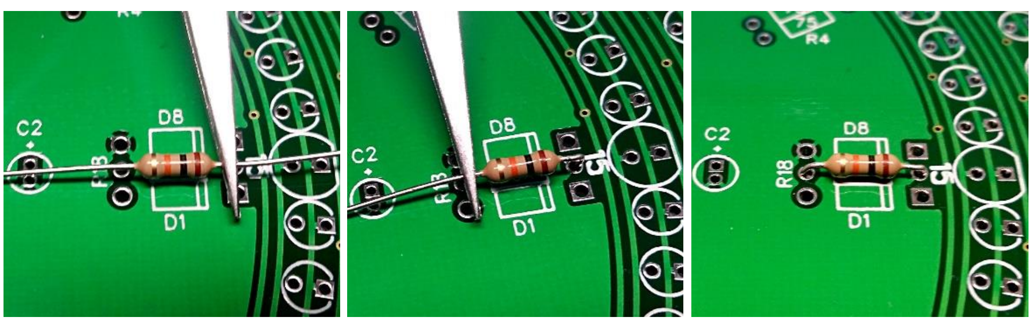


Figure 10 – Perfectly bent leads using tweezers

Assembling the PC Board

Now we are ready to build the clock! Remember to only open the bags as you are ready to use the parts. Some bags contain several different parts. For some parts, there are extra parts included in case of loss or damage. As you remove parts from these bags, reseal the bags with tape to keep from losing the remaining parts. Components are installed in the order of component height. For this reason, it is important to install the parts in the order that is outlined below.

☑ Tip: Check the boxes as you complete each step.

☐ Step 1 – R17-R20, 10kΩ Resistors

Use the lead bender – or the tweezers as demonstrated in the previous section – to bend the leads of R17, R18, R19 and R20, and place them into the PCB. They are located at the 12, 3, 6 and 9 o'clock positions on the board. These resistors are 10kΩ and have the color code Brown, Black, Orange. Although these parts are not polarized (they can be installed either way), I recommend installing them so the Brown stripe points toward the edge of the board. Flip the board over and solder and clip the leads.

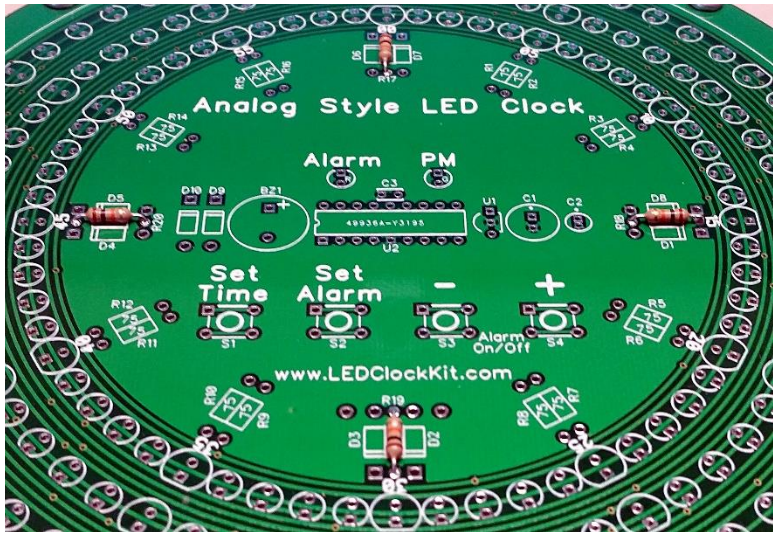


Figure 11 – Installing R17-R20

☐ Step 2 – R1-R16, 75Ω Resistors

Now place R1 through R16 into the PCB. They are located in pairs at the 1, 2, 4, 5, 7, 8, 10 and 11 o'clock positions. These resistors are 75Ω and have the color code Violet, Green, Black. Install them so the Violet stripe points toward the edge of the board. Flip the board over, solder and clip the leads.

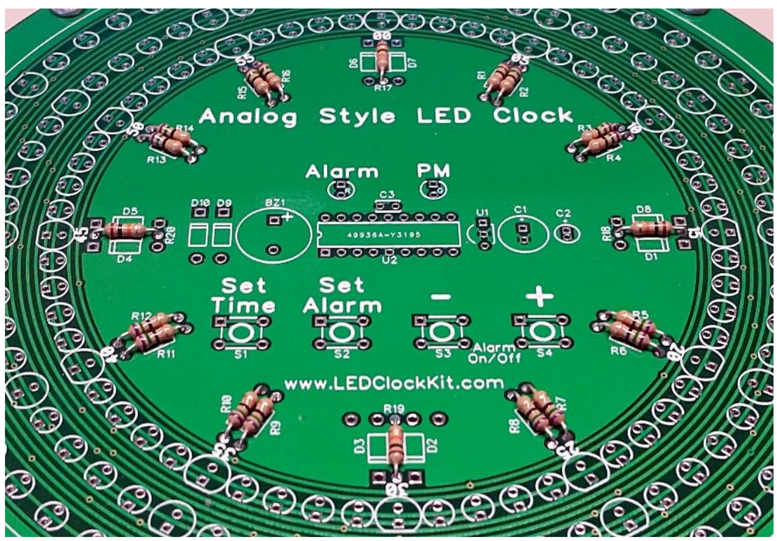


Figure 12 – Installing R1-R16

☐ Step 3 – D1-D10, Diodes

The ten diodes are now going to be installed. They are located at the 12, 3, 6 and 9 o'clock positions in pairs plus two more to the left of the center of the board. These parts are polarized; it is critical that they are installed in the correct direction. Match the stripe on the device with the stripe on the silkscreen outline on the PCB. D1 through D8 are installed with the stripe toward the edge of the board, D9 and D10 are installed with the stripe toward the top or 12 o'clock position. Install the diodes into the PCB. Take a minute to double check that the parts are installed in the correct orientation, then turn the board over, solder and clip the leads.

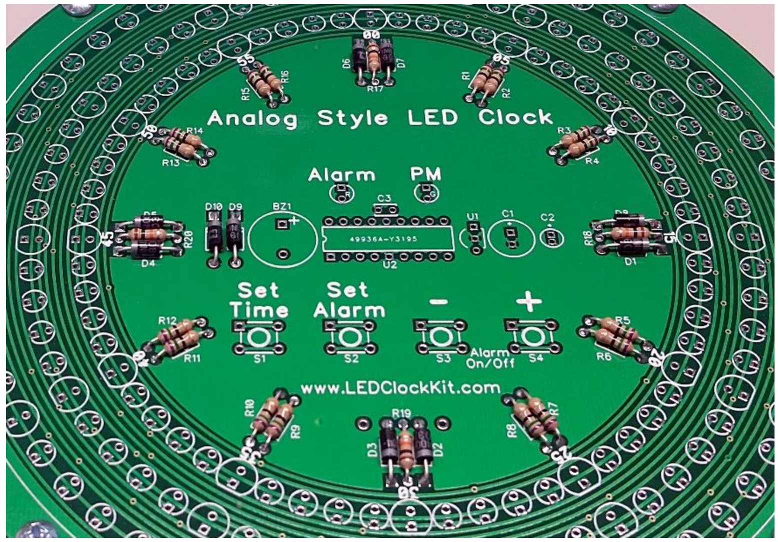


Figure 13 – Installing D1-D10

☐ Step 4 – Socket for U2

Install the socket for U2 into the PCB, making sure the notch in the socket lines up with the notch silkscreened on the board. Do not insert U2 into the socket at this time. We will do that after we have verified the proper operation of the power supply circuit. Solder two leads on opposite corners of the socket, then make sure the socket is flat against the board before soldering the rest of the leads. The leads of the socket do not need to be trimmed after soldering.

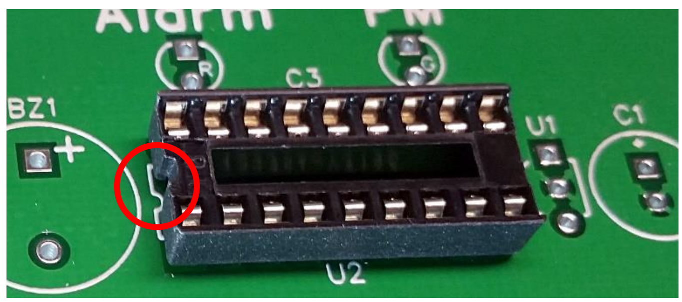


Figure 14 – Match notch on socket with notch on board

☐ Step 5 – C3, 0.01µF Capacitor

Install C3. It is not polarized, so it can be installed either way. It should be installed so it is about the same height as the socket for U2. Make sure that some of the bare lead is showing on the top side of the board as shown in Figure 15.

Figure 15 – C3 Installed

☐ Step 6 – S1-S4, Pushbuttons

Install the four pushbuttons. They are not polarized, so they can be installed either way. You do not need to clip their leads; just solder them in.

Figure 16 - Pushbuttons installed

☐ Step 7 – BZ1, Buzzer and C2, 1.0µF Capacitor

Install BZ1 onto the PCB. Note that this part is polarized. The (+) on the bottom of BZ1 must be installed in the hole with the square pad that is marked (+) on the PCB. See Figure 17. The buzzer also has a (+) on its top side that should be toward the 12 o'clock position on the board as seen in Figure 19.

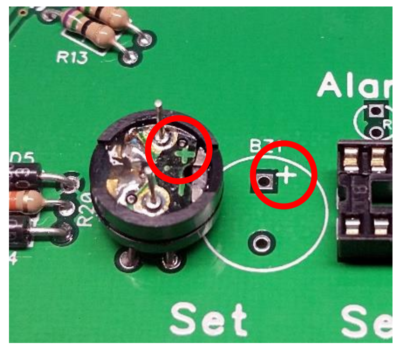


Figure 17 – BZ1 orientation [match the (+) marks]

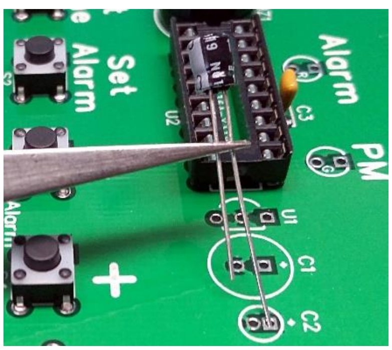


Figure 18 – C2 orientation

Now install C2. The long lead on C2 is the positive lead and must be inserted into the hole with the square pad that is marked (+) on the PCB. See Figure 18. The negative side of C2 is marked with a white stripe and should be toward the 6 o'clock position on the board. Be extra careful when soldering C2 – the leads are very close together and it is easy to create a short with too much solder.

After soldering both components and clipping their leads, we will be ready to start installing the smaller 3mm LEDs.

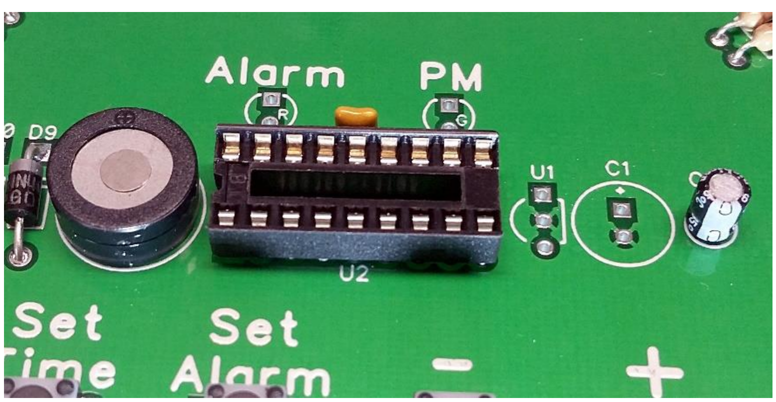


Figure 19 – BZ1 and C2 installed – note their orientation

Preparing to install the LEDs

The outer ring of LEDs is for displaying the Seconds, the next ring is for Minutes, and the innermost ring is for Hours, which corresponds to the relative length of the hands on a traditional analog clock. The LEDs representing Seconds are Red, Minutes are Green, and Hours are Yellow.

Keep in mind that the lenses of the LEDs are all clear. There is no way, looking at an unlit LED, to determine what color it is. For that reason, take precautions to not mix up the LEDs. Always work with one color of LED at a time and always solder all the LEDs of one color before installing LEDs of a different color. Replace extra LEDs into their bag and seal it with a piece of tape before opening a bag of different colored LEDs.

If you happen to get the LEDs mixed up, or come across a stray LED on your bench and want to know what color it is, you can use a multi-meter to safely light it up. Set the meter to diode test mode. It may say "Diode" or have the schematic symbol for a diode, that looks like this: ▷|. Connect the Black lead to Common on the meter, and the other end to the short lead of the LED. Connect the Red lead between Ohms (or Ω) on the meter and the long lead of the LED. The LED should light up – it may be faint, but you'll be able to tell what color it is. If it doesn't light up, it may be that you have the leads backwards on the LED – try reversing them.

Figure 20
Using a meter to light an LED

LEDs are also polarized; it is critical that you install every one of them in the correct orientation. You won't actually hurt anything if you happen to install one backwards, but because of how they are connected in a matrix, it can cause multiple, seemingly random LEDs to light at the wrong time. More information on troubleshooting an issue like this is covered in the Troubleshooting section.

Please note the absence of reference designators on the PCB for the LEDs – it would have cluttered the appearance of the board. From the outside to the inside, the LEDs are DSxx, DMxx, and DHxx, where xx is the number portion of the designation. The numbers start with 00 at 12 o'clock, and increment in the clockwise direction. Every 5th set of LEDs are numbered on the PCB – you can find these numbers just inside the innermost LED ring. The three LEDs at the 12 o'clock position are marked with a single letter (R, G, and Y) to indicate the color for each ring. The Alarm LED (DA00) has an R for red, and the PM LED (DA01) has a G for green.

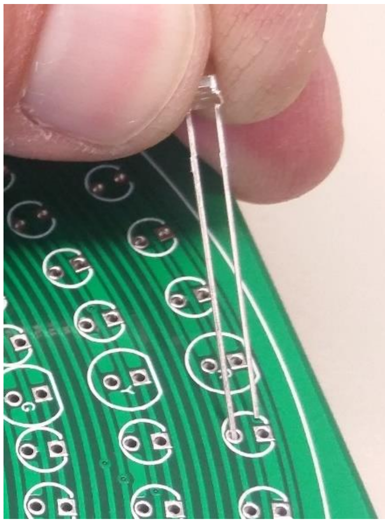


Figure 22
Proper LED orientation

The LEDs in this kit each have a long lead (anode) and a short lead (cathode). All of the LEDs are to be inserted into the board with the long lead going into the hole with the round pad, and the short lead going into the hole with the square pad. See Figure 21. For the LEDs that are arranged in rings around the board, the round pads are toward the center of the board, and the square pads are toward the edge of the board. For the PM and Alarm LEDs, the Round pads are toward the 6 o'clock position and the Square pads are toward 12 o'clock.

Figure 21 – Internal wires point the same way

You should always double, and even triple check, the orientation of the LEDs you have installed before soldering them. One way to tell, is to look through the top of the LEDs at the internal wires, and make sure they are they are all pointing the same way as shown in Figure 22. An eye loupe may be necessary to see the very thin wire. The wire points towards and is connected to the long (anode) lead.

Figure 23
5mm LED flat spot

The larger (5mm) LEDs each have a flat spot that, when inserted correctly, will align with the flat spot on the board silkscreen outline of the LED (Figure 23). Unfortunately the smaller (3mm) LEDs do not have this flat spot.

When you flip the board over, look at the leads. Make sure the long leads are toward the inside and the short leads are towards the outside. See Figure 24. Triple checking the orientation of the LEDs during assembly will be well worth the effort in the time it saves troubleshooting and reworking mistakes.

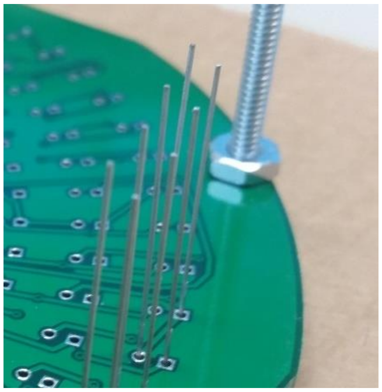


Figure 24
Long leads toward inside

☐ Step 8 – Small (3mm) Red LEDs

We will start with the Small Red LEDs. Insert them into the board with the long leads going into the holes with the round pads. The LEDs go in the outer ring in the positions marked with the outlines of small LEDs. Do not place any small LEDs in the 5 second positions, which have the outlines of large LEDs. You will place the large LEDs there later. You can place a few of the small LEDs at a time – for instance, four on each side – and solder and trim the leads, then place more. Or you can place all the small red LEDs, then do your soldering and trimming.

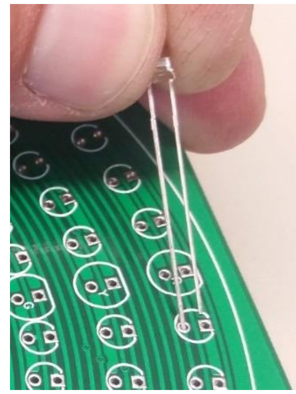


Figure 25
Proper LED orientation

It is easy to accidently start putting the LEDs in backwards as you work around the board. Be careful that by the time you get halfway around the board, you are not putting them in the wrong direction. I found that if I turned the board as I installed the LEDs, it made it easier to keep installing them in the same direction.

DA00 – the "Alarm" LED – also gets a small red LED. Be sure to double check and triple check the orientation of all the LEDs before soldering.

When you are all done with installing all the small red LEDs, you your clock should look like Figure 27. Be sure to put away any remaining LEDs into their bag and don't forget to solder and trim the leads before going on to the next color.

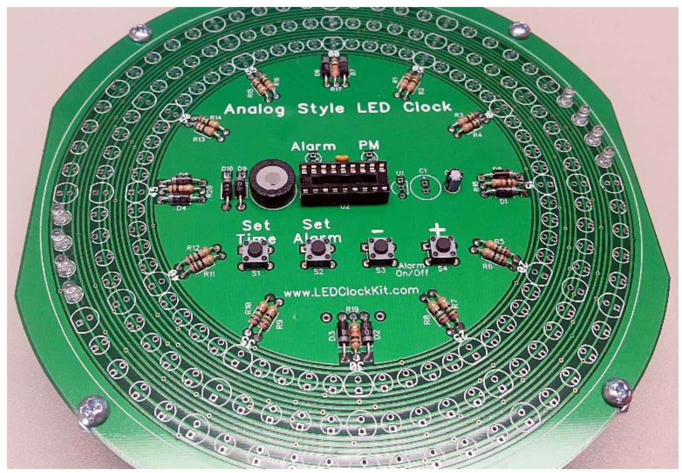


Figure 26 – Inserting LEDs eight at a time

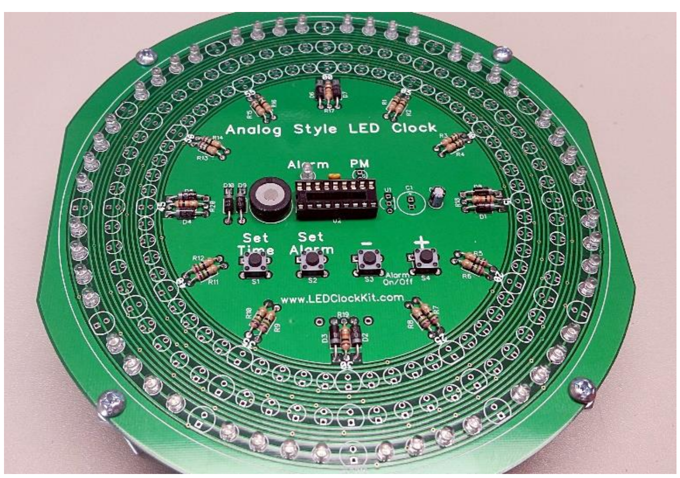


Figure 27 – All 3mm Red LEDs installed

☐ Step 9 – Small (3mm) Green LEDs

Now place the Small Green LEDs into the second ring, making sure they are all in the correct direction: longer leads in the holes with the round pads, toward the center of the board. Also place one in the spot for the "PM" LED. Don't forget to double check the orientation of the LEDs. Be sure to solder and trim all the leads, and put the extra Green LEDs back in their bag before moving on. Figure 28 shows what the clock should look like at this point.

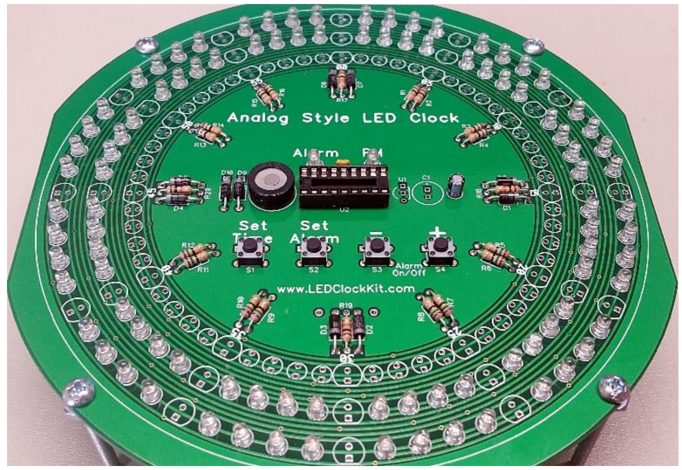


Figure 28 – All 3mm Green LEDs installed

☐ Step 10 – Small (3mm) Yellow LEDs

The Small Yellow LEDs can now be placed in the inner ring, again with the longer leads toward the center of the board, in the holes with the round pads. Double check the orientation of the LEDs before soldering and trimming the leads. At this point, all the small LEDs have been installed into the board as shown in Figure 29.

Now would be a good time to look over the board to make sure you didn't forget any LEDs and that you didn't inadvertently install one in a location reserved for a larger LEDs. Check that all the LEDs are snug against the board and make any corrections before continuing.

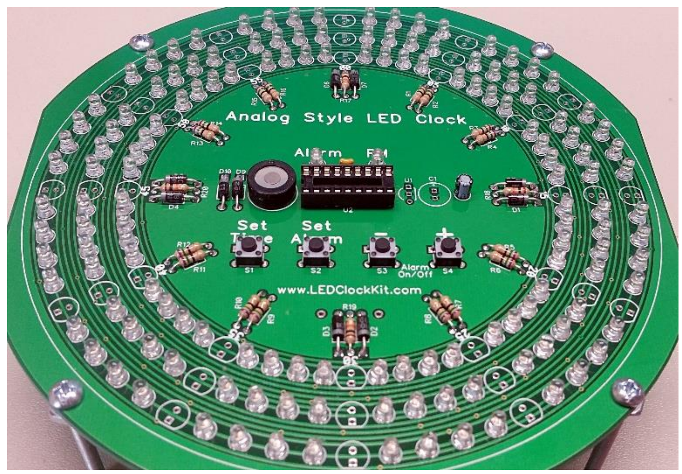


Figure 29 – All 3mm Yellow LEDs installed

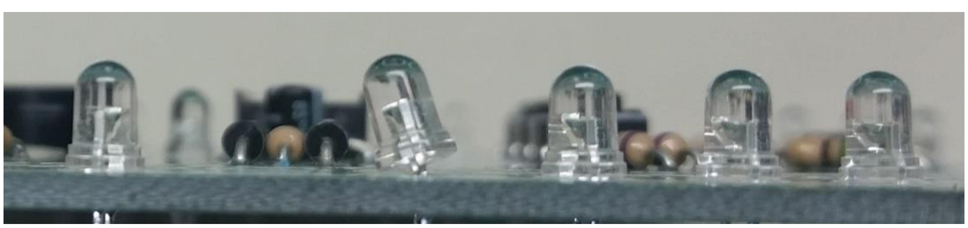


Figure 30 - LED not snug against board, needs rework

☐ Step 11 – C1, 100µF Capacitor

Before we install the large (5mm) LEDs, we'll install a couple of other parts. These parts are shorter than the 5mm LEDs. Install C1 – it is polarized and must be installed in the correct orientation. Its longer lead (positive) is to be installed in the hole with the square pad marked (+). The white stripe denotes its negative lead and should be toward the 6 o'clock position on the board.

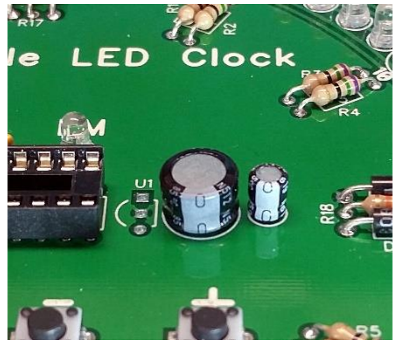


Figure 31 – C1 installed

☐ Step 12 – U1, +5V Regulator

Insert U1, the +5 Volt Regulator with its flat side toward C1. Solder and clip its leads. There should be some space between the bottom of U1 and the board as shown in Figure 32.


Figure 32 – U1 installed

☐ Step 13 – Large (5mm) Red LEDs

Now it is time to install the large LEDs. As with the small LEDs, work with one color at a time. Solder and trim the leads, and return any extra LEDs to its bag before going to the next color.

Install the Large Red LEDs into the remaining positions in the outer ring. There are twelve total – every fifth LED will be a large one. Watch polarity (long lead goes in hole with round pad toward center of board and the flat edge on LED matches the flat edge on silkscreen). Double check your work before soldering and trimming the leads.

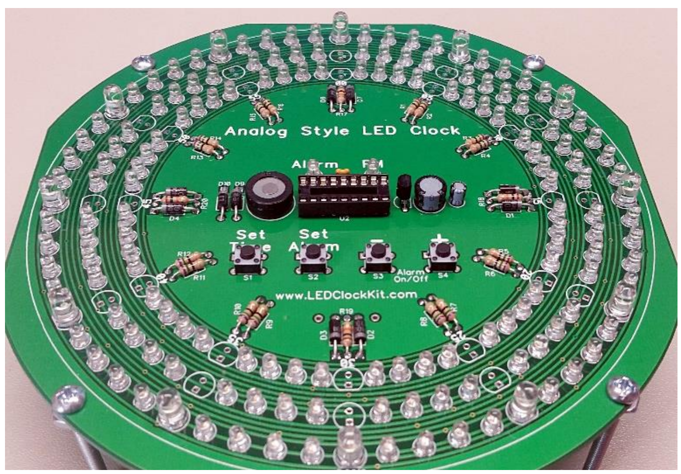


Figure 33 – All 5mm Red LEDs installed

☐ Step 14 – Large (5mm) Green LEDs

In the same way, install the twelve Large Green LEDs into the second ring. Double check polarity before soldering.

Figure 34 – All 5mm Green LEDs installed

☐ Step 15 – Large (5mm) Yellow LEDs

Finally, install the twelve Large Yellow LEDs into the innermost ring. Double check polarity before soldering. Whew! Finally done installing all the LEDs! Take moment to examine the large LEDs once again for polarity, and for being straight and snug against the board, making any corrections necessary.

You are now done with installing components onto the front side of the PCB. The screws and nuts can now be removed.

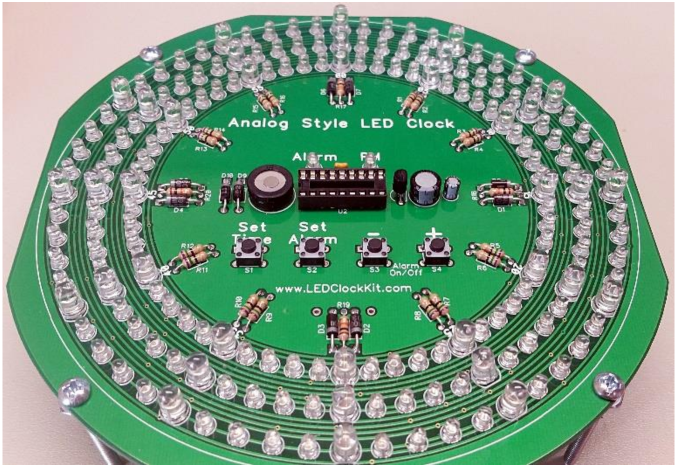


Figure 35 – All 5mm Yellow LEDs installed

☐ Step 16 – J1, Power Jack Cable

The leads of the power jack cable assembly get installed onto the back side of the board. The cable is pretty long. You may want to cut off some of the excess wire, so that it is one or two feet long. It does need to be at least five inches long, to bring the connector outside of the frame. You can also leave it full length; it is up to you. You may want get out the wall transformer and plug it into the jack and see how long they are together. If you're uncertain, leave it long – you can always shorten it later.

Strip back about ½ inch (13mm) of insulation and tin the leads. Insert the wires into the back

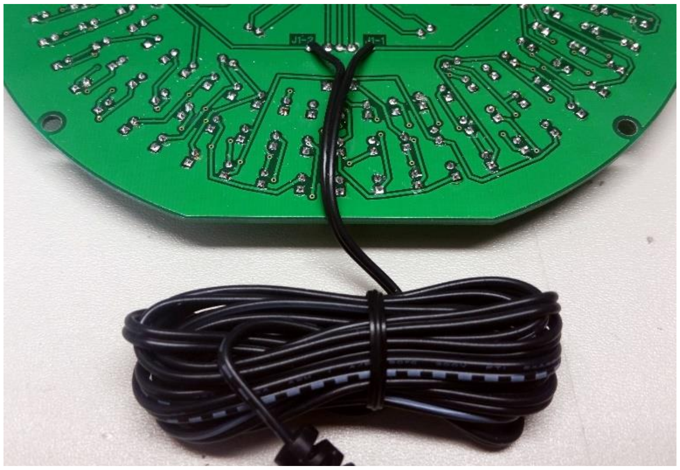


Figure 36 – J1 installed

of the board, into the holes labeled “J1-1” and “J1-2”. Since the wall transformer has an AC voltage output, the polarity of J1 doesn’t matter – you can install the leads either way. Turn the board over and solder and trim the excess leads.

Congratulations! You have completed assembling the PC Board. It should look like Figure 37. Do not plug in U2 or apply power to the board yet. We will be doing that in the next section.

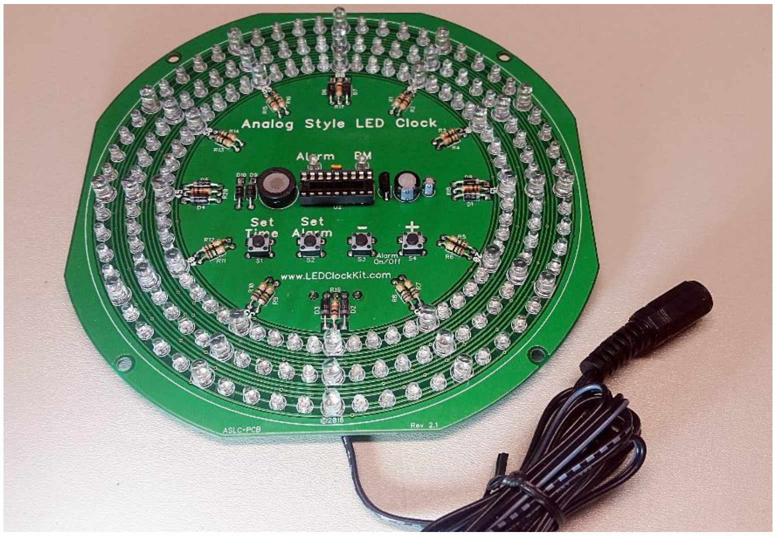


Figure 37 – Completed PCB

Testing

Double Check Your Work

Before going any further, double check the work you have done so far. Make sure the polarized components are in the correct orientation: Diodes, Capacitors C1 and C2, Buzzer BZ2, U1, and the Socket for U2. Compare the orientation of your parts to Figure 38. We won't worry about the LEDs at this point; hopefully you double and triple checked their orientation as you installed them. Make any corrections before continuing. If you did happen to install the socket for U2 in backwards, it is probably easier to leave it that way, and just make sure that when you install U2, to install it according to the board's silkscreen instead – remember that U2 is installed with the notch closest to the buzzer, BZ1.

Figure 38 – Check the orientation of the parts

Once you have determined that everything looks good on the top side of the board, turn the board over and look at the back side. Using an eye loupe or magnifying glass, take a close look at each of the solder connections. Do they look like good solder connections as described in the Soldering Tips section? Make sure that every lead has been soldered. Check that the solder has stuck to both the pad and the component lead. Make sure there are no solder bridges – that is, solder that has bridged the gap between pads. Don't move on to the next section until you are fully satisfied with the quality of the solder connections.

Testing the Power Supply Circuit

Before installing U2, let's check out the power supply circuit. Plug J1 into the wall transformer and plug the transformer into the wall. Note that the wall transformer is made for the United States electrical system. If you are in another country, you will need to obtain the appropriate adaptor. As an alternative, you could use a wall transformer that is made for your country. It needs to have between 9 and 12 volts AC output (not DC) and should be capable of supplying at least 100mA (or 0.1A) of current. If it is able to supply more current, that's ok, just as long as it is at least 100mA.

Use your multi-meter to check the voltage being supplied to U2. Set the meter for DC Volts on a scale that can measure at least 5 volts, like the 20 volt scale. Place the positive lead on pin 14 and the negative lead on pin 5 as shown in Figure 39. The pins of U2 are numbered starting with the lower left corner and going counterclockwise. The meter should read close to 5 volts. It must be within 0.25 volts. If you get a different reading, unplug the clock and check the orientation and solder connections of these components: D5, D6, D7, D8, C1, C2, C3, U1, and U2 Socket. See the Troubleshooting section for more information.

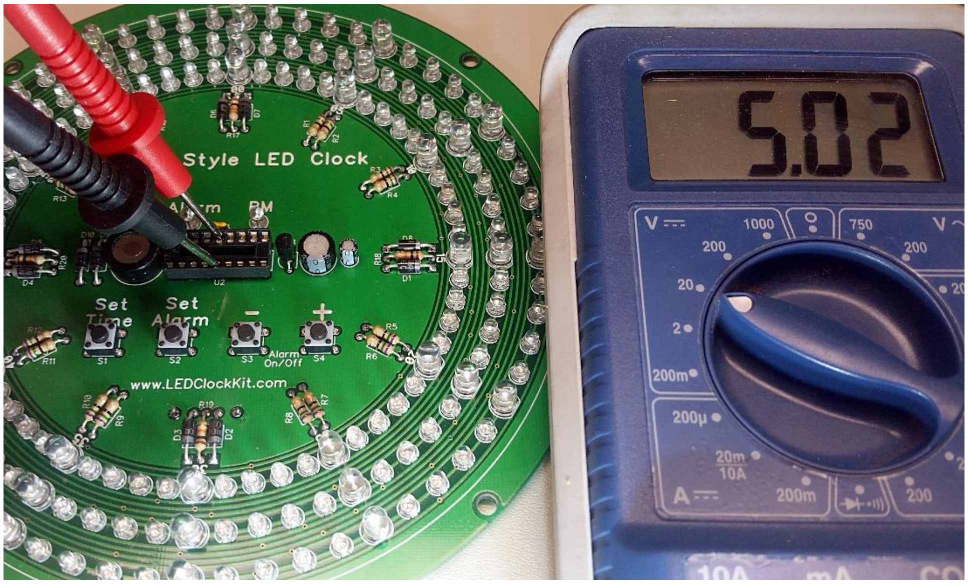


Figure 39 – Checking the power supply to U2

Testing the Clock

Unplug the power from the clock and carefully install U2 into its socket, in the correct orientation, with the notch closest to the buzzer. Insert it slowly, checking that every lead is going into the socket correctly. It is easy to accidently bend a pin under the IC.

Now we are going to plug in the clock make sure it operates correctly. If any of the LEDs or buttons do not operate as they should, or if the buzzer doesn't sound, refer to the Troubleshooting section for how to find and fix the problem. It would be helpful at this point to review the Operation section at the beginning of this manual before continuing.

After plugging the clock in, three LEDs should light up for about two seconds to indicate the version of the software. During these two seconds, press and hold the [Set Time] and [Set Alarm] buttons down. The clock should go into the LED Test Mode, where it lights each LED, one at a time, in sequence. Once the LED Test Mode starts, you can release the buttons. If you miss pressing these buttons during the first two seconds, the clock will display the time, starting at midnight; you can unplug the clock and try getting into LED Test Mode again. Pressing any button exits this test mode and starts displaying the time. If you cannot get the clock to go into LED Test Mode, you may have a problem with a button. You can use the

Operation section at the beginning of this manual to attempt to operate the clock. If you find that a button is not responding, check out the Troubleshooting section to isolate and fix the problem.

When the clock is in the LED Test Mode, it will sequentially light each LED. Check that each LED lights. It will run through all the Second LEDs, then the Minute LEDs, then the Hour LEDs. Then the Alarm LED will light followed by the PM LED, then back to the Second LEDs. The sequence keeps repeating until you press a button. If any of the LEDs don't light, or LEDs light that are not supposed to, see the Troubleshooting section before continuing. Otherwise, if everything looks good, press any button to exit the test mode.

An alternate way to check the LEDs is to attempt to operate the clock using the Operation section at the beginning of this manual. Allow the Seconds to advance through the whole minute, to see that all the Second LEDs work. Then set the time, advancing through all the Minute LEDs and Hour LEDs, looking for any LED that won't light.

After exiting the LED Test Mode, the clock should start keeping time starting at 12:00 midnight. The Second (red) LEDs should advance once per second. See the Troubleshooting section if the LEDs do not advance.

Now, let's check the Pushbuttons. Press and hold the [Set Time] button and press and release the [+] button. The Second LED should go out and the Minute LED should advance one. Press and hold the [+] button. The Minute LED should begin advancing, slowly at first, then picking up speed the longer it is held down. Release the [+] button. While still pressing the [Set Time] button, press and release the [-] button. The Minute LED should go back one. Press and hold the [-] button. The Minute LED should be moving backwards, slowly at first, then picking up speed the longer it is held down. Release the [-] button, then release the [Set Time] button. When you release the [Set Time] button, the Second LED should light at 00 seconds and begin advancing once per second. For any issues, see the Troubleshooting section before continuing.

Press and hold the [Set Alarm] button and then press and hold the [+] button. The Minute LED will start advancing, slowly at first, but quickly picking up speed. Release the [+] button and check that the [-] button makes it go backwards. Release the [-] button, then release the [Set Alarm] button and the clock should go back to displaying the time. Refer to the Troubleshooting section for any issues.

At this point, you have verified the operation of all the LEDs and all of the pushbuttons. Now, let's test the buzzer. Referring to the Operation section, set the alarm to any time you like, and then set the time to one minute prior to the time you set for the alarm. Make sure that the PM light matches for both the alarm and the time – both times need to be set to AM or both need to be set to PM. Press and release the [+] and [-] buttons simultaneously. The Alarm LED should light. Wait for the time to advance to the alarm time, and the buzzer should sound – beep, beep, pause, beep, beep, pause, beep, beep, pause, etc. Press and release the [+] and [-] buttons simultaneously to turn the alarm off. If the Alarm LED doesn't light, or the buzzer seems muffled or doesn't sound at all, see the Troubleshooting section.

If you have made it this far, Congratulations! You have successfully tested all the hardware functions of your new clock. Now proceed to the section on Finishing the Frame.

Troubleshooting

Use the Testing section to check the operation of the clock. If you come across a problem, you should stop and consult this section before continuing. Look at each subsection below and start troubleshooting with the first subsection that describes your problem. Reading through the Theory of Operation section can also aid in troubleshooting problems.

If you are not able to resolve the problem with your clock after working through this section of the manual, or you are in need of additional parts, please contact us through our website: www.LEDClockKit.com.

Power Supply is Producing Less Than 4.75 Volts

If the voltage on the socket for U2 across pin 14 and pin 5 is below 4.75 volts, check the voltage across C1. Depending on the output of your wall transformer – which should be between 9V AC and 12V AC – the voltage at C1 should be somewhere between 10 volts and 25 volts DC. If it is less than 7 volts, then check for bad solder connections at J1, J2, D5, D6, D7, D8, and C1. Also check for the proper orientation of D5, D6, D7, and D8. If you find a bad connection, or a diode installed backwards, remove power, fix the problem, then apply power and check the voltage across C1 again.

If it is still under 7 volts, check for shorts (solder bridges or touching leads) at C1, C2, C3, U1, and the socket for U2. Check D5, D6, D7, and D8 with the diode check setting on your meter. Again, if you find a problem, remove power, fix the problem, reapply power, and check the voltage at C1 again.

If the voltage at C1 is at least 7 volts and the voltage at U2's socket is still low, the problem will likely be with C2, C3, U1 or the socket for U2. Check for bad solder connections and shorts.

If the voltage is low only when U2 is plugged in, and you cannot locate any other problem, it may be that U2 is bad.

Power Supply is Producing More Than 5.25 Volts

Hopefully you have determined this before installing U2. If U2 has been installed, unplug the power cord and remove U2 before proceeding. Plug the power cord back in and check for voltage on the socket for U2 across pin 14 and pin 5. If it is between 4.75 and 5.25 volts, then you should be good to go.

If the voltage is slightly too high, it is likely a damaged U1 that is operating, but out of spec. Check the voltage across C1. Depending on the output of your wall transformer – which should be between 9V AC and 12V AC – the voltage at C1 should be somewhere between 10 volts and 25 volts DC. If the voltage at the socket of U2 is about the same as at C1, then you either have a short (look for a bridged solder connection or touching leads), likely at U1 or a bad U1.

If the voltage across C1 is greater than 30 volts, then you very likely have a damaged U1 as well. The voltage at C1 shouldn't be more than 25 volts. If it is, then you are using the wrong wall transformer or plugging it into a 220V AC outlet. The transformer is designed for a 110V AC outlet as is used in the US. If you are outside the US, you need to either get an adaptor that converts your country's voltage and plug style to the US voltage and plug style, or get an AC wall transformer made for your country that has an output between 9 and 12 volts AC (not DC). It also needs to be capable of supplying at least 100mA (or 0.1A) of current. If it is able to supply more current, that's ok, just as long as it is at least 100mA.

If at any time the voltage to U2 exceeded 6.5 volts while U2 was in its socket, it is possible that U2 was damaged. If you fix the overvoltage problem and get the correct voltage at U2, and the clock still fails to work, it is likely you will need to replace U2.

No LEDs are Coming On

At this point, it is assumed that you have determined that your power supply is working correctly and you have installed U2. If no LEDs come on when you plug in your clock, first check that you have installed U2 correctly. The notch should be toward the buzzer, BZ1. Also, make sure that you didn't install U2 one pin off to the right or left. If that's the problem, immediately unplug the clock. Remove U2 and install it correctly and try plugging the clock back in. Another thing to check for are bent pins on U2 that failed to mate with the socket.

With U2 installed correctly, check the voltage at U2 across pin 14 and pin 5. If it is below 4.75 volts – but it was in spec before you installed U2 – look at the subsection Power Supply is Producing Less Than 4.75 Volts for troubleshooting. Since the undervoltage issue only shows up when U2 is installed, you'll need make the voltage measurements outlined in that section with U2 installed.

If you have determined that U2 is installed correctly and the voltage at U2 is correct, and yet no LEDs come on, it is most likely that U2 is defective or has not been programmed correctly.

Clock is Stuck at 12 O'clock

Check the solder connections on R20 and U2 pin 4. Also look for solder bridges or shorted leads in that area. Make sure that R20 is 10kΩ (Brown, Black, Orange). Check that you are using a wall transformer that puts out AC voltage. If you use one that has DC output, the clock will light up, but it will not run.

Another possibility is that one or more pushbuttons have failed. See the subsections Single Pushbutton Failure or All Pushbuttons Fail below.

An LED Fails to Light

If an LED fails to light, it is typically accompanied by several other LEDs that light when they are not supposed to. The problem is with the LED that does not light, not with the other LEDs. The LED that does not light is either installed backwards, has a bad solder connection or solder bridge, or the LED itself has gone bad. If it is not a solder issue, try replacing the LED with a new one, making sure that it is installed the correct direction.

If several LEDs are failing to light, check R1 through R14 for bad connections and the proper value (75Ω, Violet, Green, Black). Also check the Socket for U2 for bad solder connections and check that all the pins of U2 are seated in the socket correctly.

Wrong LEDs Lighting

If several LEDs are lit that are not supposed to be, that is typically due to an LED that is supposed to be lit but does not. Watch the clock during its initial test mode, looking for an LED that is not lighting when it is supposed to. Then follow the troubleshooting outlined in the previous subsection, An LED Fails to Light.

Single Pushbutton Failure

The buttons are often used in pairs, so you may need to try different combinations to determine which button is at fault. For example, if you can set the alarm, but not the time, the problem is with the [Set Time] button. If you can advance the time, but not set it backwards, then the problem is with the [-] button.

Referring to the schematic in Figure 50 on page 38, determine which diode is associated with the failed button. For example, if button S2 doesn't work, you will also check out D3. Look at the button and diode for bad solder connections or shorts.

All Pushbuttons Fail

If all the buttons are failing to respond, or troubleshooting per the Single Pushbutton Failure subsection fails to reveal the problem, check for bad solder connections or shorts at R2, R3, R4, R5, R12, R17, R18, and R19.

Make sure the resistors are the correct values. R2, R3, R4, R5, and R12 should be 75Ω (Violet, Green, Black) and R17, R18, and R19 should be 10kΩ (Brown, Black, Orange).

The Alarm is Muffled or Sounds Distorted

Check that BZ1 is install in the correct direction. The (+) on the top of the buzzer should be oriented toward the 12 o'clock position. Also, make sure that R15 and R16 are the correct value, 75Ω. The colors should be Violet, Green, Black.

The Alarm Does Not Sound

Be sure that when you are testing the alarm, that you are setting both the time and the alarm to AM or both to PM, and that you are turning on the alarm. The Alarm LED should be lit solid when the alarm is turned on and will flash when the alarm sounds. It will flash once a second, and will be on slightly longer than it is off.

If the Alarm LED is flashing in this pattern and the alarm is still not sounding, check the solder connections on BZ1, R15, R16, D9, and U2 pin 3. Make sure that R15 and R16 are the correct value, 75Ω. The colors should be Violet, Green, Black. Also, make sure that D9 is installed in the correct direction.

The problem could also be that a pushbutton is stuck, which will immediately snooze the alarm and prevent it from sounding. The Alarm LED flashes while the alarm is snoozed as well, but the difference is that the LED is on much longer than it is off – it very briefly turns off once a second. If that's the case, troubleshoot per the subsections Single Pushbutton Failure or All Pushbuttons Fail above.

Finishing the Frame

The frame has been made out of Medium-Density Fiberboard (MDF). I chose MDF because it is durable and easy to mill. It also paints well. A can of spray paint is all you need to give your clock frame a nice finish. I painted my original clock black and my next clock silver. I did not use primer on either one and they came out looking great.

If you want a higher gloss finish, you should use a couple of coats of solvent-based primer on the edges and one on the face. Before applying the primer and after each coat dries completely, lightly sand the edges with 100-grit sand paper or, better yet, foam-backed sanding pads. Try to avoid sanding the face of the frame – any scratches there will stand out, especially after painting with a high-gloss paint.

Here is a technique that worked well for me. First, place the frame face up and with the notch for the power cord towards you. See Figure 40. As you paint, concentrate on the edges as they need more paint than the face. The face of the frame usually gets enough overspray to cover it. Make two or three passes across the inside top edge and two or three passes across the outside bottom edge. Then turn the frame 45° and repeat the process. Keep rotating and painting the frame until the power cord notch is facing you again – then you know you are done. Finally, make a quick pass across any area on the face that may have gotten missed.

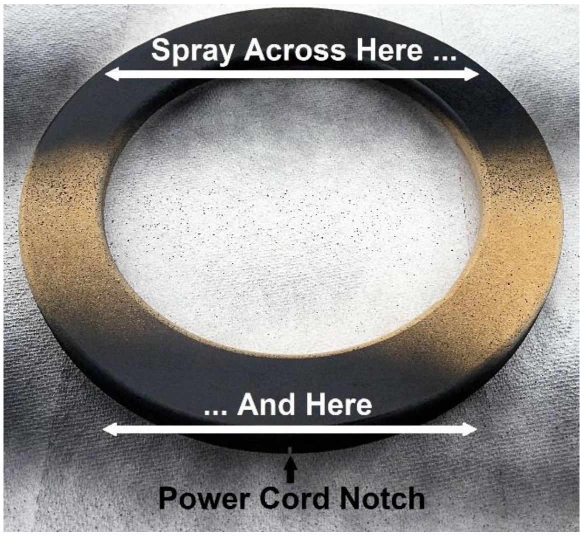


Figure 40 – Painting the frame

It is better to err on the side of not enough paint, than to get too much and have it run. Apply two or three light coats of paint, following the instructions on the spray paint can concerning drying time between coats. Allow the final coat of paint to dry fully, per the paint's instructions before proceeding.

Before applying the numbers, mount the completed PC Board in the frame. Make sure that the top of the PCB (12 o'clock) is lined up with the top of the frame, at the keyhole hanger. Use the four nylon washers and four wood screws to attach the PCB to the frame. Be careful not to over torque the screws – just snug them up so the PCB is not loose. Route the power cord through the slot at the bottom of the frame leaving some slack between where the wire comes out of the notch and where it attaches to the PCB. If you find the slot is too tight, or paint has gotten into the slot, you can enlarge it a little using a nail file, an emery board, or some folded up sandpaper.

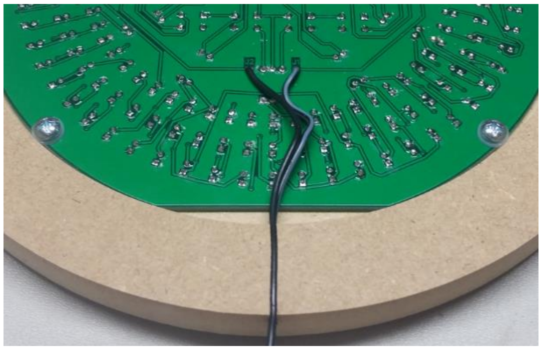


Figure 41 – Routing the power cord

Finally, apply the numbers using the large (5mm) LEDs as a guide. A set of Arabic numerals (1, 2, 3, etc.) and a set of Roman numerals (I, II, III, etc.) are provided in the kit.

Figure 42 – Original clock for Senior Project, painted black with Arabic numerals

These are some general instructions for finishing the frame. You can be as artistic as you like. You can brush on paint, drizzle or sponge paint to create unique designs, or use something besides paint, like contact paper. You can use the numerals supplied in the kit, or use something else – stickers, stamps or hand written numerals. You can orient the numerals as you like. You don't even have to use the supplied frame. Whatever you use, and however you decorate it, just make sure it doesn't interfere with the operation of the clock. In general, the opening should not be smaller than the large white ring that is silkscreened on the PCB, just beyond the outer ring of LEDs. The PCB should mount into a recessed area of the frame so that the component leads on the back side of the board do not make contact with the wall.

Check out the photo gallery at our website, www.LEDClockKit.com, for some other ideas of how you can finish your frame. Send us a picture of your clock and we might add it to the gallery too!

Figure 43 – This version of the clock, painted silver with Roman numerals

Theory of Operation

As you read through this section, please refer to the appropriate schematic in the Schematics subsection, Figure 47 through Figure 50 (pages 36 - 38).

Power Supply

The power supply is shown in Figure 47. The clock gets its power from an AC to AC wall transformer. The AC output from the transformer connects to the board at J1-1 and J1-2. The AC voltage is converted to DC by a bridge rectifier, comprised of D5-D8, and filtered by a 100µF capacitor, C1. The resulting voltage should be in the range of 10 to 25 volts DC across C1, depending on the actual output of the wall transformer.

This DC voltage is regulated using a +5V regulator, U1. A 1µF bypass capacitor, C2, improves transient response.

Microcontroller

The microcontroller, U2, has a total of 16 I/O pins (see schematic in Figure 48). One I/O pin, RA5, is an input only pin and is dedicated to monitoring the 60Hz AC input. One I/O pin, RA4, can sink but not source current. This pin would not be suitable to drive the LEDs, therefore it is dedicated to drive the buzzer (BZ1). The remaining 14 I/O pins are capable of sourcing and sinking current and can also be placed in a high-impedance input mode. These tristate I/O pins are ideal for driving the LEDs in this project. They are connected to the LED matrix through 75Ω current limiting resistors, R1-R14. A comparator and programmable voltage reference, which are built into the microcontroller, are used for reading the pushbuttons – more about that in the Detecting Button Status subsection.

Using the Electric Utility Frequency for Timing

R20 limits the current of the 50 or 60Hz AC signal as it is fed to the RA5 input on the microcontroller as seen in Figure 48. The RA5 input pin doubles as the programming enable pin for the device. Since a high voltage is used to force the microcontroller into programming mode, D9 is used to clamp the AC signal to +5V to prevent the microcontroller from entering that mode during operation. The microcontroller has a built in diode clamping the signal to ground. These clamping diodes add 0.7V each, resulting in a 6.4V peak-to-peak signal. The actual signal can be seen in Figure 44. The built-in Schmitt trigger input on this pin nicely squares up this signal for microcontroller.

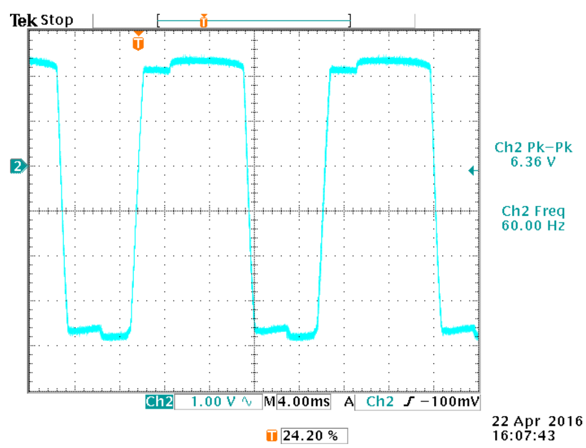


Figure 44 – Clamped 60Hz AC waveform

Driving the Buzzer

Still referring to the schematic in Figure 48, the RA4 I/O port on the microcontroller is used to drive the buzzer, BZ1. This I/O port is not able to source current as would be needed for the LED matrix, however all that is needed to drive the buzzer is a current sinking output. According to the buzzer's datasheet it can be driven by up to a 3V peak-to-peak square wave. A resistor is needed in series to drop the 5V power supply voltage to under 3 volts.

It turns out that 37.5Ω in series with BZ1, reduces the square wave to 2.7 volts in amplitude, well within the buzzers specs. To maintain symmetry in the printed circuit board design, two 75Ω resistors in parallel, R15-16, are used to create a resistance of 37.5Ω.

Since the microcontroller's RA4 I/O port does not have diode protection to V_{dd}, D10 was added to the circuit to clamp voltage spikes coming from the buzzer's coil.

Complementary LED Drive

Complementary LED drive, also known as Charlieplexing, allows a large number of LEDs to be controlled with a small number of I/O pins. Charlieplexing is named after Charlie Allen who used this technique to create LED driver ICs in the 1990's. The idea behind Charlieplexing is simple: given every possible combination of two I/O ports, connect two LEDs between them, with the two LEDs in parallel and in opposite directions. Each I/O port, of course, would also need a current limiting resistor.

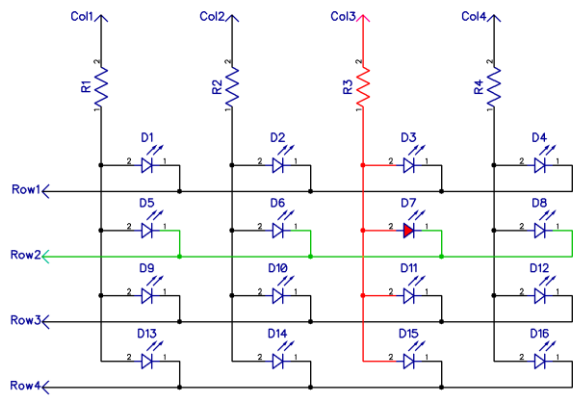


Figure 45 – Standard 4x4 LED matrix

Another way to look at Charlieplexing is to consider a standard 4x4 matrix of LEDs (see Figure 45). There are four rows and four columns, requiring a total of eight I/O lines to control 16 LEDs. Notice that each column has a current limiting resistor. To light D7, +5V is applied to Column 3 and Ground is applied to Row 2. The result is that D7 illuminates.

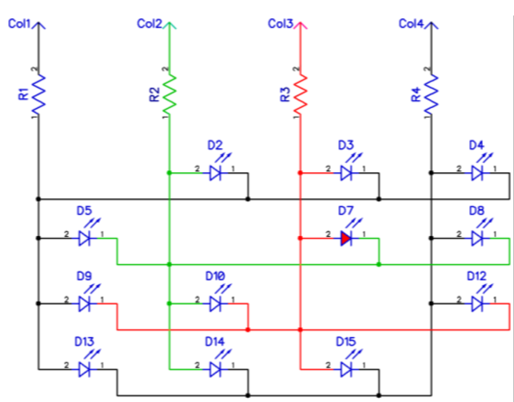


Figure 46 – Four line Charlieplexed matrix

If the diagonal LEDs are removed, and at these intersections, the rows are connected to the columns, the row inputs will no longer been needed and can be removed. The result is seen in Figure 46, where 12 LEDs are connected to four I/O lines. This is Charlieplexing.

To light D7, +5V is applied to Column 3 as before, but Ground is applied to Column 2. All other lines are placed in a high-impedance input state. The result is that only D7 lights. In this configuration, current flows through two resistors, R3 and R2. The LED will have about a 2.2V drop across it. Therefore, R2 and R3 will have a voltage drop of about 1.4V.

Although D10 has power and ground applied, it is reversed biased and will not light. There are additional forward-biased current paths, for example through D3 and D5. However, there are at least two LEDs in any such path. Since D7 maintains about a 2.2V drop, no other LED in the matrix will have more than about 1.1V across it. This voltage is insufficient to break through the LED's PN junction, therefore no current will flow and no other LED will light.

It is important to note here that if D7 fails to light, due to a faulty connection or bad LED, it will not maintain the 2.2V drop. The higher voltage will be enough to light LEDs on other paths, for example D3 and D5. For this reason, the key to troubleshooting problems with a Charlieplexed matrix is to look at the LED that fails to light, instead of the other LEDs that light when they are not supposed to.

With a given number of I/O lines (N), how can the number of LEDs that can be controlled be calculated? Figure 46, gives a visual representation of the answer. The number of LEDs that can be Charlieplexed are N (number of rows) times N (number of columns) minus N (number of diagonal LEDs removed). For this example, the number of LEDs four I/O pins can control can be calculated:

$$LEDs = (N \cdot N) - N = (4 \cdot 4) - 4 = 12\,.$$

If a matrix is constructed using 14 I/O pins, then we have:

$$LEDs = (N \cdot N) - N = (14 \cdot 14) - 14 = 182\,.$$

That happens to be exactly how many LEDs are used in the LED Clock. The schematic in Figure 48 shows each of the 14 I/O pins used for Charlieplexing (RA0-3, RA6-7, RB0-7) having a current-limiting resistor (R1-R14) between the microcontroller and the LED matrix (CP_Bus). The LED matrix is shown in Figure 49.

Only one LED can be lit at a time in a Charlieplex matrix. To achieve the illusion of multiple LEDs being on at the same time, the LEDs are lit one at a time in rapid sequence, fast enough that the human eye cannot distinguish any flickering.

Detecting Button Status

The four pushbuttons share five of the I/O lines with the LEDs. Incorporating the buttons into the Charlieplex circuit was done to avoid increasing the pin count on the microcontroller. To prevent inadvertently lighting LEDs that are not supposed to be on, the pushbutton circuit was carefully designed so as to not interfere with the operation of the LEDs (see schematic in Figure 50). Diodes D1-D4 keep CP02-CP05 lines isolated from each other when more than one button is pressed at the same time. Resistors R17 and R18 create a voltage divider that provides 2.5V to the buttons (S1-S4) and CP12 of the Charlieplexed LED circuit. R19 prevents this voltage from varying much when CP12 is used to source or sink current to turn on LEDs.

For the following discussion, recall that the voltage drop across the current limiting resistors connected to active outputs of the microcontroller is about 1.4V. While an LED is lit, the minimum voltage in the matrix, with respect to ground, would be about 1.4V, and the maximum voltage would be about 3.6V.

When CP12 is not driving an LED, it is kept in a high impedance state. Therefore, the R17, R18 voltage divider output of 2.5V appears on CP12. Considering the minimum and maximum voltages in the matrix, this would place no more than 1.1V across any affected LED in the matrix, which is not enough to turn on any extra LEDs.

When CP12 is used to sink current in the LED matrix, its voltage would be about 1.4V. When CP12 is sourcing current into the matrix, its voltage would be about 3.6V. The voltage at the node common to R17, R18 and R19 can be calculated to range between about 2.1 and 2.9V.

If a button, say S2, is being pressed at this time, current could flow through that button and D3, into the matrix at CP03. D3 prevents this voltage from introducing a current sink into the Charlieplex circuit. Due to the voltage drop across D3, no more than a 2.2V current source would be placed into the matrix. This would place no more than 0.8V across any LED, far below what is required to turn it on.

If S2 is pressed while CP03 is sinking current, the voltage at the node drops to no less than 2.1V, due to the 0.7V drop across D3. If CP12 is in high-impedance mode, this would introduce 2.1 volts into the Charlieplex circuit at that point. The sourcing Charlieplexed line could produce as much as a 1.5V drop across an LED (3.6V - 2.1V), while the sinking line could produce as much as a 0.7V drop (2.1V - 1.4V). Neither of these voltages are enough to turn on an LED. Therefore, no matter which LED is lit, and no matter what button is pressed, the button circuit cannot cause any other LED in the matrix to light.

To detect the state of a button, the I/O line to that button is brought low, while all remaining I/O lines in the LED matrix are left in the high-impedance input state. For example, to read S1, CP02 which is connected to S1 via D4, would be set to low output. If the button is not pressed, 2.5V from the voltage divider would be feed through R19 into CP12. However, the LED that is connected with its cathode on CP02 and its anode on CP12 will cause that voltage to drop to the operating voltage of that LED, about 2.2V. This voltage appears at the microcontroller at RA1 via R12. Since the current going through the LED is greatly limited by R17 and R19 and the reading of a button happens in a few microseconds, the LED will not appear to illuminate to the human eye.

If S1 is pressed, current through D4 will force the output of the voltage divider to drop to 0.7V. That voltage is passed through R19 and R12, appearing on the microcontroller's RA1 pin. This pin is internally connected to the inverting (-) input of the microcontroller's built-in comparator. The comparator is used to detect the small voltage change. The noninverting (+) input to the comparator is connected to the programmable internal voltage reference that is set to 1.042V. If the button is pressed, the voltage at the I/O pin is about 0.7V and the comparator's output goes high. If the button is not pressed, the input voltage is about 2.2V and the comparator's output goes low. The output of the comparator is read as a 1 or a 0 by the microcontroller.

Each button is read, one at a time, by bringing that button's line low while leaving the other Charlieplexed lines in their high-impedance state. Immediately after reading the button status, the button's I/O line is set back to high-impedance state to minimize any affect this has on the LED matrix.

Schematics

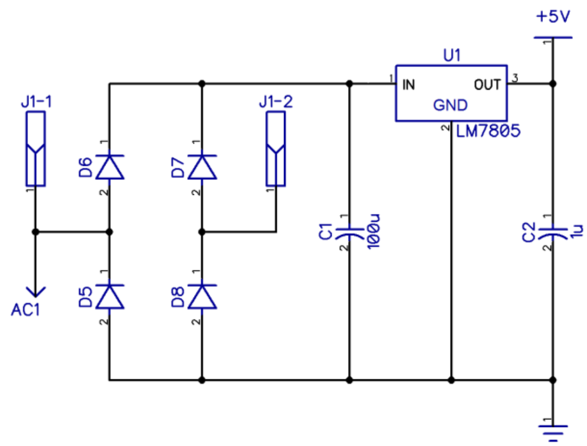


Figure 47 – Schematic, Power Supply

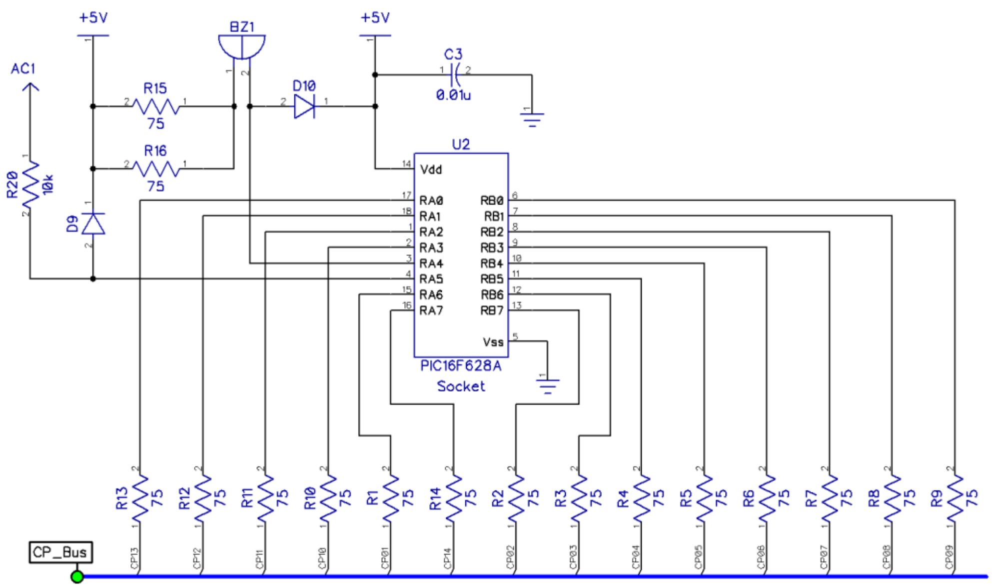


Figure 48 – Schematic, Microcontroller

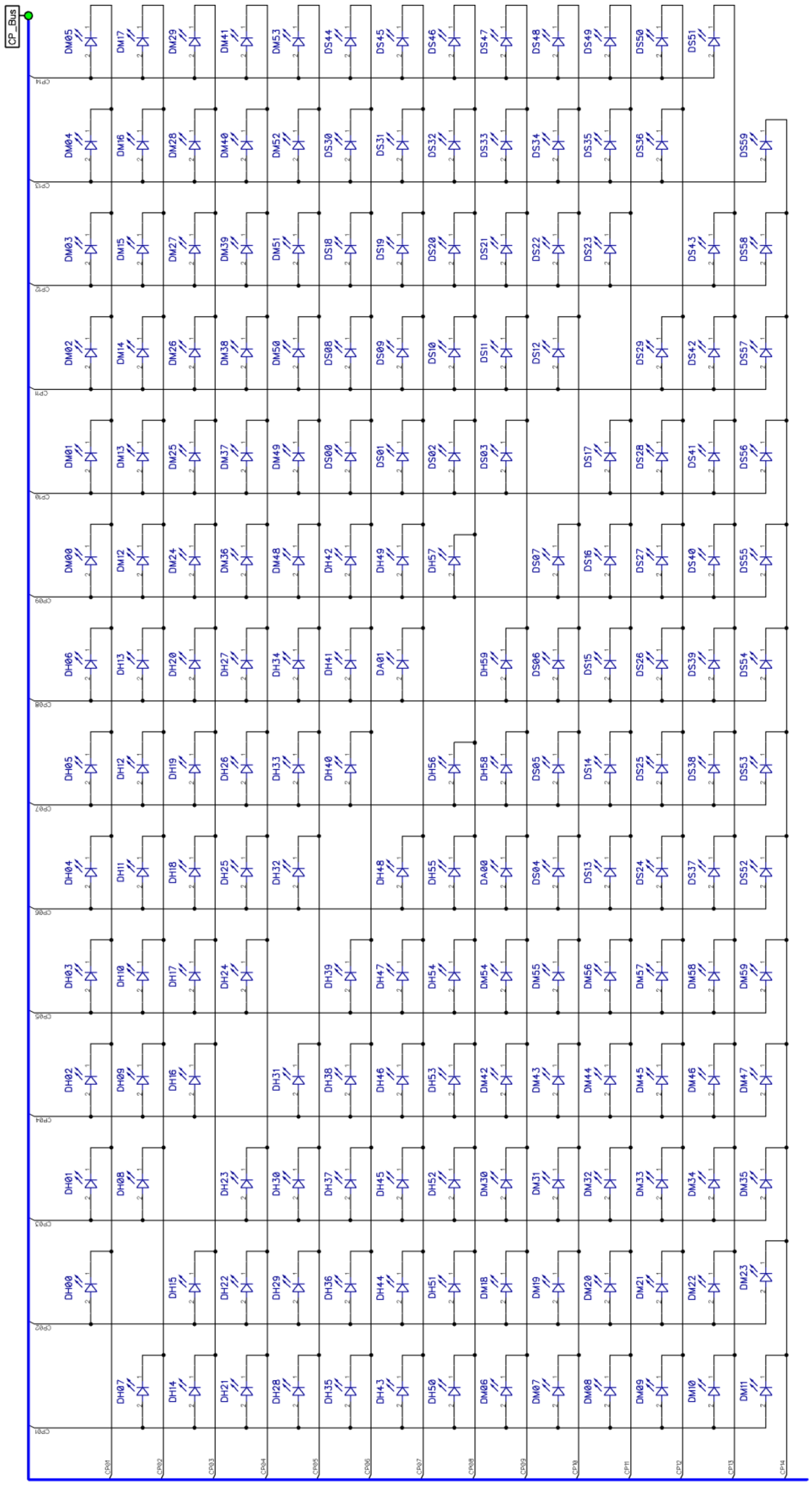


Figure 49 – Schematic, Charlieplexed LED Matrix

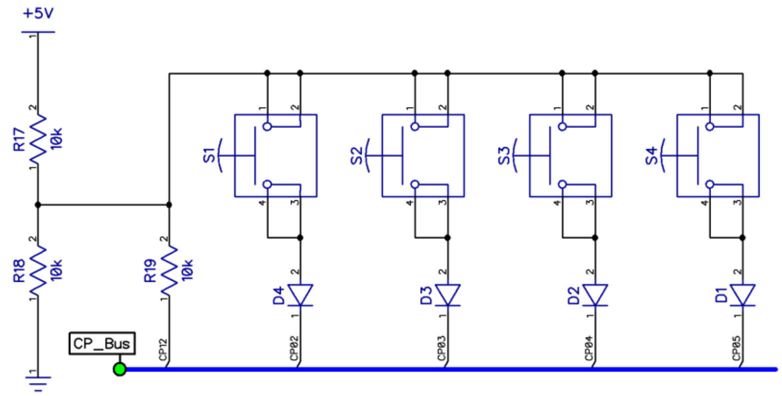


Figure 50 – Schematic, Pushbuttons

Software Operation

The microcontroller program was written in assembly language and was developed and assembled using Microchip's free MBLab Integrated Development Environment (IDE). To make writing, debugging and editing the software easier, the program was broken up into different sections, each in its own file. The software source files can be downloaded at www.LEDClockKit.com and should be referred to while reading this section.

Timing issues are crucial to the operation of the clock. It keeps correct time by monitoring the 50 or 60Hz line signal coming from the wall transformer. In addition, since only one LED can be lit at a time, the multiplexing of the LEDs must be carried out fast enough that the human eye cannot detect that they are being multiplexed, and with precise timing so that the LEDs appear consistent in brightness. For example, the brightness of the Second, Minute and Hour LEDs should not change when the Alarm LED is turned on or off.

An interrupt routine is used to accomplish these tasks (see interrupt.inc). The subroutine at initInt sets up Timer0 to interrupt the processor every 256 instruction cycles, or every 256µs. This provides a way that events can occur at a fixed and reliable rate. initInt also sets up the programmable voltage reference and comparator for the reading of the pushbuttons and performs various other file register initializations. Finally, it enables the interrupt routine, appropriately labeled Interrupt.

Eight interrupt cycles (tracked by register intPWMcount) are used to drive of each of five LEDs: Seconds, Minutes, Hours, Alarm and PM. This time is allocated whether or not the LED is to be turned on. Each LED can be turned on any number of the eight interrupt cycles, allowing fine-tuned adjustment to its brightness by varying its on time, otherwise known as pulse width modulation (PWM). Each of the five LEDs are turned on in sequence, with the register intLEDCount keeping track of which LED is active. Therefore, each lit LED is turned on for 2.048ms each multiplexing cycle.

After eight interrupt cycles, the LED is turned off and the next interrupt cycle is dedicated to reading a pushbutton. The value of intLEDCount, if it is between 1 and 4, determines which button is read. If it any other value, no button is read during that interrupt. The button is read by bringing low the line that the button is on and reading the value returned by the comparator. The status of the button is stored for use by the main routine. In addition, its current status is compared to its previous status and if it has changed, then a button-pressed or button-released flag is set as appropriate. When the main routine processes the pressing or releasing of the button, it resets the flag. In the case of the [+] and [-] pushbuttons, as long as the button is held down, the interrupt routine will continue to set the button-pressed flag in accordance with the values in delayTable and dNumTable. These tables define respectively the number of multiplex cycles to wait and the number of times to repeat before advancing to the next entry of the tables, where the repeat speed increases.

It takes nine interrupt cycles to light each LED, eight cycles for lighting the LED plus one cycle for reading a button. Since there are a total of five LEDs to cycle through, each multiplexing cycle take 45 interrupt cycles or 11.52ms. This period is equivalent to 86.6Hz which is fast enough that the human eye cannot distinguish any flickering.

In addition to multiplexing the LEDs and reading the buttons, the interrupt also monitors the 50 or 60Hz line and drives the buzzer. After counting 50 or 60 transitions of the AC line, the interrupt routine advanced the time by one second by calling incSecTime in time.inc. When the buzzer needs to be driven, the interrupt alternately drives the buzzer line high and low, each for the duration of one interrupt cycle. The result is that the buzzer is driven by a signal with a period of two interrupt cycles for a total of 512µs or a frequency of 1953Hz. This is very close to the buzzer's resonance frequency of 2kHz.

The interrupt, therefore, takes care of the minutiae of multiplexing the LEDs with precise timing, monitoring the buttons and AC line, keeping time, and sounding the buzzer. This leaves the main routine (in clock.asm) to take care of the "big picture" part of the program, in a foreground/background type of program structure. The main routine responds to the buttons, and handles the setting of the time and alarm, and the sounding, snoozing, and silencing of the alarm.

The main routine starts off by initializing the interrupt by calling initInt. Then it calls other subroutines to set up the Alarm and PM LEDs, display the software version, run the LED Test Mode, and determine if the AC frequency is 50Hz or 60Hz. Then it enters the main loop (mainLoop) where it responds to buttons and alarm events.

If the [Set Time] or [Set Alarm] buttons are held down, the main routine monitors the button-pressed flags for the [+] and [-] buttons and increases or decreases the time or alarm by one minute accordingly. It calls incMinTime, decMinTime (in time.inc), incMinAlrm, or decMinAlrm (in alarm.inc) as appropriate.

The main routine monitors the simultaneous pressing of the [+] and [-] buttons and toggles the alarm on flag and LED. The main routine checks when the current time has reached the alarm time and if the alarm is on. It also continually checks the snooze time and the alarm silence time. If the alarm is to be sounded, it simply set a flag or resets it to silence the alarm. The interrupt routine monitors the flag and drives the buzzer accordingly.

The main routine also does all the lookup and decoding of the Port and Tris values for the LEDs. It does this when the time advances a second as well as during the setting of the time and alarm. To save program space, a two-step lookup approach was used. Each LED has a single byte that is returned from the lookup tables in tables.inc under the heading "Output Tables". The upper nibble contains the line number that is to be driven high and the lower nibble contains the line number that is to be drive low. Each nibble is in the range 1..14 decimal. The nibbles are each looked up using the tables in tables.inc under the heading "Charlieplex Port Masks" to calculate and set the Port and Tris values.

To set the Port and Tris values, the main routine calls timeLEDout or alrmLEDout (in led.inc) as appropriate. For each LED to be lit, these routines perform the lookups outlined above and calculates the Port and Tris values for both Port A and Port B. The Port and Tris values are not sent to the actual Port and Tris registers, but are stored in shadow registers. The interrupt routine copies the shadow registers to the actual Port and Tris registers at the appropriate time for each LED.

Two additional files, definitions.inc and registers.inc, contain definitions for constants, file registers, bit assignments, time calculations and port definitions.

Specifications

Assembled Clock Dimensions (without power adaptor): 8.5 x 8.5 x 0.7” (216 x 216 x 18mm)

Power Adaptor Dimensions: 2.0 x 2.2 x 2.4” (50 x 60 x 65mm)

PCB Dimensions: 6.2 x 6.2 x 0.063” (158 x 158 x 1.6mm)

Kit Weight (approx.): 2 lbs. (0.9kg)

Maximum Power Consumption: 0.5 watt

Power Adaptor Output:
- 9 to 12V AC
- 100mA (minimum)

Microcontroller:
- Microchip PIC16F628A
- 2048 words of program memory
- 224 bytes volatile RAM memory
- 128 bytes non-volatile EEPROM memory
- 16 I/O lines
- 4 MHz clock speed

Software:

The software comes preloaded on the Microcontroller. As of the writing of this manual, the current version of the software is 2.0.0.

The version of the software is displayed for two seconds when the clock is first plugged in. One LED in each ring will be lit. The numbering of the LEDs start with 00 at the 12 o’clock position, and increment in the clockwise direction. The version number is read from the inside ring to the outside ring. For example, for software version 2.0.0, the following LEDs would be lit: inner ring 02, middle ring 00, and outer ring 00.

Warranty:

Due to the nature of this product we cannot offer a warranty of any kind. However, we want you to succeed and we are happy to provide assistance through our website, www.LEDClockKit.com. Please note, we cannot accept returns on this product if any of the parts bags have been opened, or if the kit is not in its original condition. See website for additional terms and conditions.

RoHS:

Kits are now RoHS compliant. Either RoHS compliant or leaded solder can be used to assemble the kit. To maintain compliance, the kit must be soldered with the included solder, or solder that is RoHS compliant.