

TinyTimer - Kickstarter Edition
Assembly Instructions
Released by Nanohawk
Written by Brian Greul
brian@nanohawk.com
www.nanohawk.com

About Nanohawk: Nanohawk is the tradename of Swamp Ventures LLC, a Texas LLC. Founded by Brian Greul in 2012, Nanohawk is focused on cloud and embedded technologies. Nanohawk strives to develop and market kits that are affordable, hackable, useful, and most importantly do-able.

About TinyTimer: Tiny Timer is a complex cycle timer. It is used to control one AC or DC device. It was designed to control a water pump and can be adapted to other applications. The original design requirement is for to come on for 90 seconds, be off for 7.5 minutes. To indicate cycle status it is equipped with 2 LEDs. A Red LED indicates the wait state, A green indicates the operation of the primary relay. Each LED flashes once per second to indicate heartbeat while active. The LED is lit for 950 ms and extinguished for 50ms. The circuit features an input terminal, a logic bypass terminal, and a reset button. The reset button resets the operation and begins operation. The input terminal is for future programming such as a switch to indicate low water level or low pressure and cause the RO system to run. It can be adapted for other purposes. Please see the schematic for more detail. The logic bypass is provided to facilitate operation of heavy loads via an external SSR. It provides logic voltage of 5v DC to the terminals. No protection is provided so adequate protection and a resistor may be appropriate to your application. It can also be used with a multimeter for diagnostic purposes, however the LED's should cover this unless the programming is modified. This device uses an Atmel Atmega328 series microprocessor. This processor supports the full set of Arduino IDE commands.

Support: Should you need assistance with operation or assembly please contact us at sales@nanohawk.com. Please provide as much detail as possible including pictures if appropriate.

License: TinyTimer is released under the CERN Open Hardware License, herein OHL. Supporting materials including this manual, documentation, code, drawings, and other pictures are released under a creative commons (herein CC) Attribution, ShareAlike license. This means that if you use this work or any part of this work you must share it with others, give credit to Nanohawk and Brian Greul and agree to be bound by the CERN OHL and CC licenses. In addition, other licenses may apply such as the GNU Public License, or GPL.

Attributions:

I would like to thank the work of the Arduino team for the work and software that made this product possible. The IDE and code associated with the Arduino project make it easy to utilize the Atmel ATTiny85 chip.

Safety:

This product is intended to control DC and AC current. Electrical current can be dangerous and event fatal. Please consider enlisting the help of a qualified electrician when connecting AC loads. If you are not familiar with safe work practices, National Electric Codes in the United States, and local building codes please get help from someone who is familiar with these things. Above all, **NEVER** touch the circuit while AC power is connected. If you are controlling an AC load it is your responsibility to place this circuit inside an enclosure so that someone else cannot touch it. For more information please visit <http://esfi.org> to learn about Electrical Safety.

If you orient the circuit board with the USB connector in the lower left, the right hand side of the circuit board is where AC current flows when controlling an AC circuit. Please familiarize yourself with this portion of the board and avoid touching it while in operation or when AC power is connected.

If you are uncomfortable working around electricity you may return the product for a full refund excluding shipping costs. Please ship it postage paid to Nanohawk, PO Box 70033, Houston, TX 77270, United States of America. Enclose a brief note that you would like a refund for safety reasons. If you have purchased a kit, the kit must be unassembled in re-saleable condition. If the kit has been started or parts are missing a deduction may be applied in order to restore the kit to saleable condition.

Skills Required:

This product requires a limited understanding of electronics and the ability to make proper solder joints. If you desire to customize the product, an understanding of Arduino programming is required.

Warranty: Please see our website for current warranty information.

[Http://www.nanohawk.com](http://www.nanohawk.com), under About.

Tools Recommended:

- **Small slot screwdriver for connecting terminals.**

For assembly:

- Soldering station or Iron
- A quality Electronics solder in a small gauge such as .020 inches. Kester makes fine solders. DO NOT use “electrical” or plumbing solder. It will destroy your kit.
- A pair of diagonal cutters or snips
- De-soldering wick
- A heat-safe work surface.
- A pen or pencil to mark off the steps as you complete them.
- A vise to hold the circuit during assembly, or an assistant.

Items required/recommended that are not included:

- USB power source such as a phone charger, laptop, or computer.
- USB A to USB B cable of sufficient length.
- Enclosure and cover.
- Mounting hardware for your enclosure.
- A drill and drill bit for the enclosure
- Wire to hook up the device you wish to control.
- An external SSR for heavy duty loads.

Step 1 – Inventory

- In this step we will identify the various components and make sure they are all present prior to assembly.
- Please locate and check-off the following items:

Description	Qty	Part Number	Schematic Number
-------------	-----	-------------	------------------

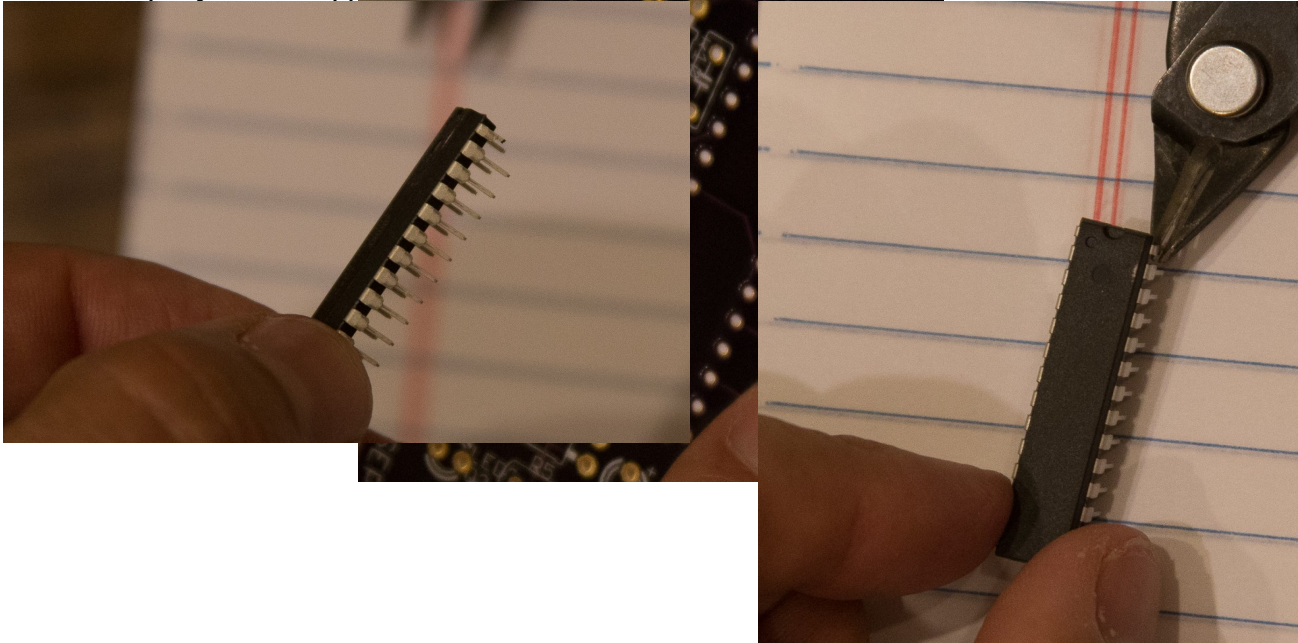
PCB	1	pcb	
Relay	1	Tianbo HK3FF-DC5v-SHG	A-5012
USB-B Connector	1		
2 row pin strip	1		ISP
Jumber block	1		
2 pos terminal block	3		Logic/load/input
Resistor 1M	1	Brown-black-green-gold	R3
Resistor 100 ohm	4	Brown-black-brown-gold	R4/R1/R2/R5
Resistor 10K	4	Brown-black-orange-gold	R7/R10/R9/R6
Resistor 20K	1	Red Black Orange Gold	R8
Red LED 3mm	1		LED 1
Green LED 3 mm	1		LED 2
Tactile Switch	3	6mm x6mm	
Transistor BC517	1	BC517	Q1
Diode	1	1N4004 or equiv	D1
Capacitor	1	.01uf	C1
Capacitor	1	.1uf	C2
Capacitor	2	22pf	
Jumper Block	1		
Xtal	1	16mhz	
LED Driver	1	Max 7219CNG	
LED 1 digit	1		
LED 4 digit	1		
CPU	1	Atmega328 series	

<inventory pic here>

Step 2 – Assembly

1. Locate the two row pin header. Place it in the PCB and solder it in place. This is easiest to do with the PCB resting on the pin-header. The short pins go into the PCB. Be careful not to push too hard or the pin will shift. If this happens place the tip of the soldering iron on the pin to heat it and gently push it back into place.

2. Locate the Atmega chip. Notice the half circle mark. With the mark facing away from you, clip the pin on the far left of the chip. The pin you are clipping the board in only one way.



3. The chip should fit into the PCB in only one direction. It should not be necessary to bend the pins. Test fit the chip and adjust the pins as required. Solder the chip in position from the bottom.
4. Locate the Max 7219CNG chip. Orient the half-moon and adjust the pins as needed to fit in the PCB. Solder it in place.
5. Locate, Install and solder the 6 pin ISP header. Take care not to push too hard or the pins will change heights. If this happens re-heat and adjust.
6. Locate and install the 4 digit and 1 digit LED modules. Take care to align the pins and orient the decimal to the bottom on the 1 digit LED. The 4 digit LED is keyed, the 1 digit is not. Use care when soldering to ensure the component stays fully seated for a clean finished appearance.
7. Locate, install, and solder the 3 pin header. Use care when soldering.
8. Install and solder the two 22uf capacitors and the 16mhz crystal. Once you insert the components bend the leads on the back of the PCB to hold the components in place.
9. Trim the leads on the capacitors and crystal. The leads on the LED displays also need to be trimmed. Use nippers to trim them flush to the solder joint. Do not cut the solder joint it self.
10. Install the diode, D1 taking care to orient the stripe according to the marking on the PCB. Solder and trim the leads.
11. Install the Red and Green LED's, reference the parts sheet and markings on the PCB. Take care to orient the long pin to the + mark on the PCB. Solder and trim the leads.

12. Install R3, and R8. Solder and trim the leads. Reference the parts sheet for resistor codes.
13. Install C2 a 1uF 50V electrolytic capacitor. This capacitor is polarity specific and must be installed in the proper orientation for the system to work. The grey stripe with the negative sign should be towards the red and green LEDs. The positive (+) lead should be towards the outside of the board.
14. Install R6, R7, R9, and R10. Solder and trim the leads.
15. Install and solder R4 and R5. Trim leads after soldering.
16. Install and solder R1 and R2. Trim leads after soldering.
17. Install and solder the 3 push button switches labeled Reset, Next, and INC. Use care when installing them not to have your finger behind them. The leads are sharp.
18. Install the 2 position terminal blocks. Use care to ensure they are fully seated when soldering and that the terminal openings face outward.
19. Install and solder the USB-B connector. Go slowly on this part and make sure that the pins are aligned properly before seating the connector. If the pins are bent it will bend the pins and may require replacing the part.
20. Install the relay, use care to ensure it is fully seated when soldering. Trim the excess leads once soldered. These leads may carry controlled current (AC or DC) and should be as short as possible once installed, or protected from accidental contact.
21. Locate and install Q1, BC517. Solder the leads and trim. Use a small pair of needlenose pliers to adjust the center lead. It will take some careful effort to align the leads and the component may not sit all the way down on the board.
22. Install C1. Solder and trim the leads.
23. Congratulations! You have assembled your TinyTimer KickStarter Kit.

Step 4 - Programming

- Connect an ISP cable to the 6 pin header.
- Set the programmer type to match your ISP Programmer.
- Check the com port and ensure it is set correctly.
- Set the board type to Arduino Duemilanove w/ Atmega 328
- Use the Arduino IDE to burn a bootloader.
- If it complains that it can't see the chip, reverse the cable connection. If it still does not work check your programmer settings and chip type.
- If it still fails you will need to troubleshoot the PCB. Contact sales@nanohawk.com for assistance.
- Send a test program to the chip and see if it runs. Use the File>Upload Using Programmer to write to the chip.
- The test code will run the Green and Blue LEDs for 3 seconds and the Red LED for 2 seconds. Once per second the LED will flash to indicate a heartbeat.
- Unplug the ISP and connect a USB cable to the USB Port. Verify that the system boots and runs.
- It is now ready for customization and use in your applications.

Step 5: Operation and Theory

- The board operates in run mode unless a jumper is connected on the two right pins in the run/prog header.
- In program mode the next and inc buttons are used to change digits and modes and increment the setting of the digit.
- Reset is the equivalent of unplugging and plugging the board in.
- Left most digit indicates mode, the other 4 indicate the number of increments.
- For example: A display of 00100 would indicate mode 0, qty of 100 units.
- The red LED indicates the total cycle time, for example 5 minutes.
- The green LED is the relay active period in the cycle. For example 1 minute.
- In this example, the green led would be on for the first 1 minute of every 5 minute cycle. The programming calculates the “stop” or “wait” time for you. This allows great flexibility in setting intervals.
- The timer operates in ticks. A tick is $1/10^{\text{th}}$ of a second and is a name I have given this interval.
- The modes are as follows:
 - 0 is ticks ($1/10^{\text{th}}$ sec)
 - 1 is seconds
 - 2 is minutes
 - 3 is hours
 - 4 is days
 - Sample settings:
 - 20005 = 5 minutes
 - 10060 = 60 seconds
 - 00030 = 30 ticks or 3 seconds ($30 \times .10$)
 - 40001 = 1 day.

Flexible examples:

To set an on time of 5 minutes once per day, set green to 20005 and red to 40001.

To set an on time of 4.5 seconds once per 90 seconds set green to 00045 and red to 10090.

To set an on time of 7.5 minutes once every hour set green to 10450 and red to 30001. (i'm using 450 seconds on this once)

The output or logic header is provided for an SSR connection. Please reference the SSR requirements for safe operation. The logic out is a 5v signal from the transistor. No load should be connected here and it has a resistor for protection against shorts.

The input header