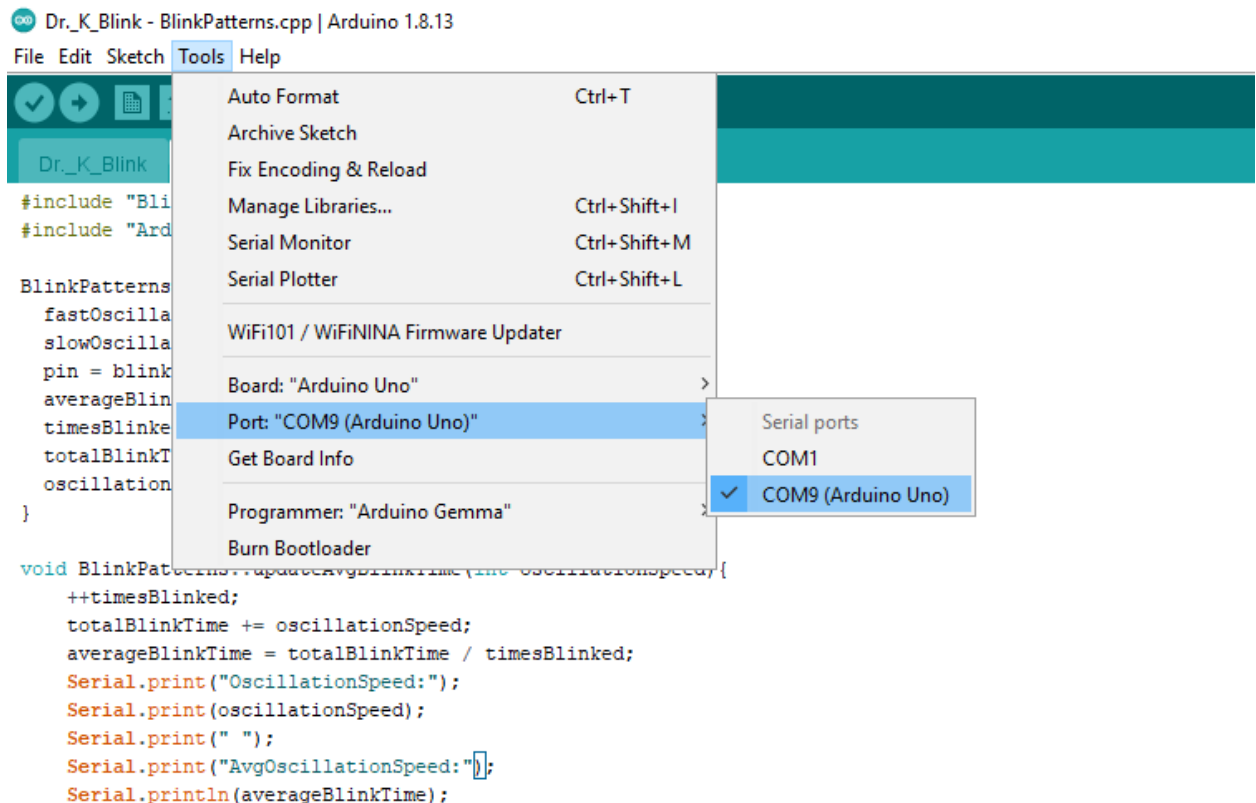


How to Program ATTINY 24/44/84

Setting up the Arduino as a programmer: In this step we set up the Arduino so that it can directly program our ATTINY chips.

1. Download Arduino IDE at: <https://www.arduino.cc/en/software>
2. Open the Arduino IDE
3. Connect your Arduino Uno, or Nano to your computer using the compatible USB Cable.
(NOTE: NOT ALL USB CABLES CONTAIN DATA LINES. CHEAP CABLES WILL PROVIDE POWER BUT BE UNABLE TO TRANSFER DATA / UPLOAD CODE**)**
4. Open Arduino IDE and go to File → Examples → Arduino ISP
5. Select the correct microcontroller board to upload to. Click “Tools” → “Board” and then navigate to the correct microcontroller model.
6. Select the correct port. The port designates which USB port to communicate with the Arduino. Click “Tools” → “Port” then select the correct port which should be labeled with the model of microcontroller connected to computer. If no port is labeled unplug your USB then check the ports. Plug in your USB then check the ports again, there should be a suggested port that was not there, that is the one to select.



7. Now that your board and port are set, click the green arrow at the top left corner of the IDE to start the upload! Alternatively, from the menu bar you can click “Sketch” → “Upload”.

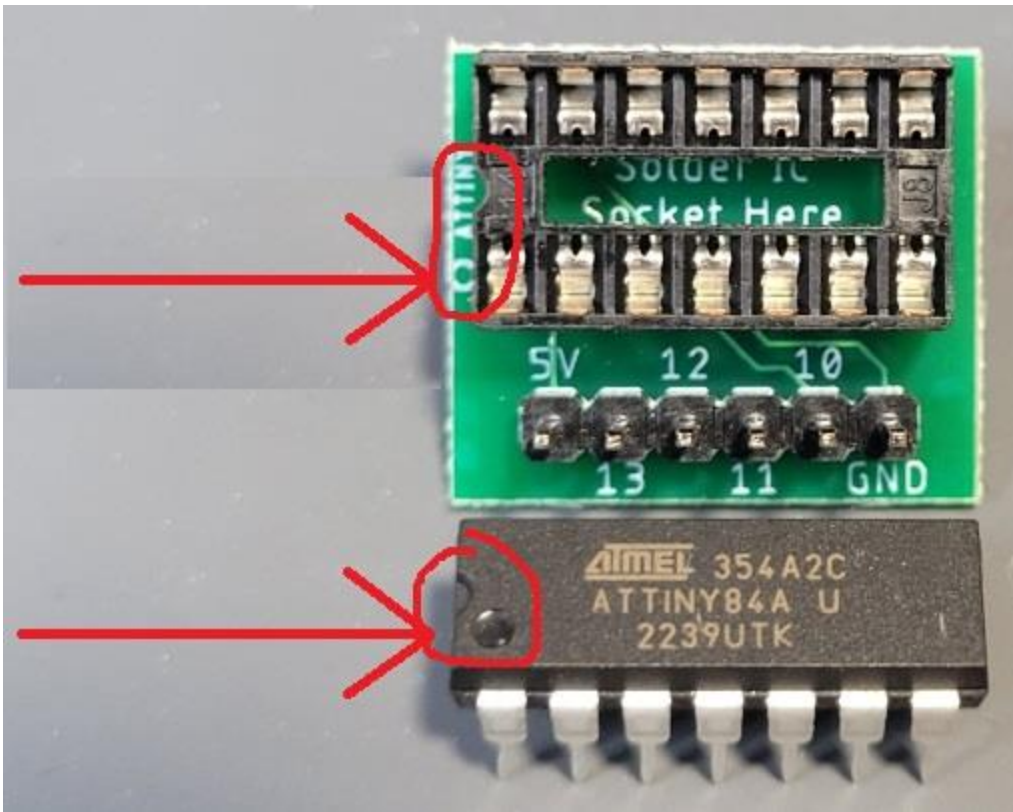
Interfacing Arduino with the ATTINY Easy Programmer

1. Find your Attiny 84 Chip and Easy programmer.

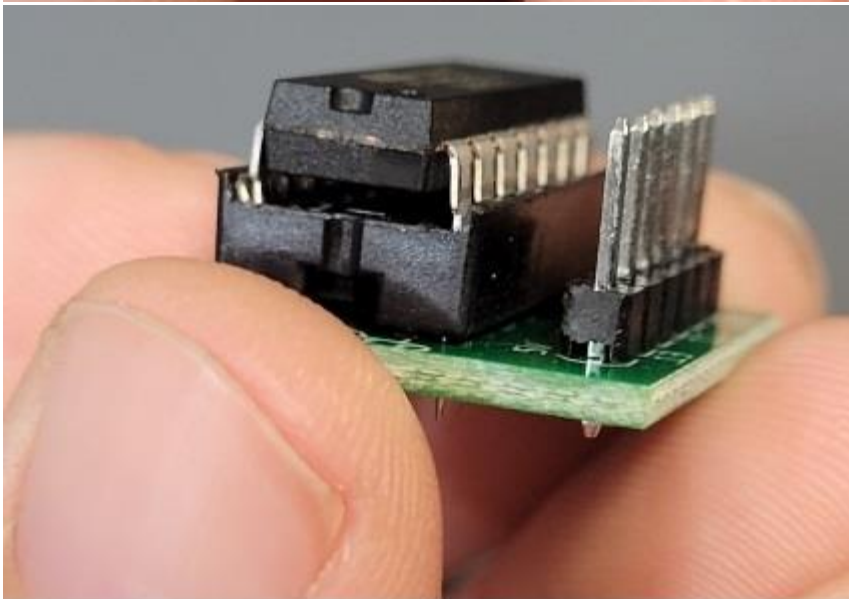
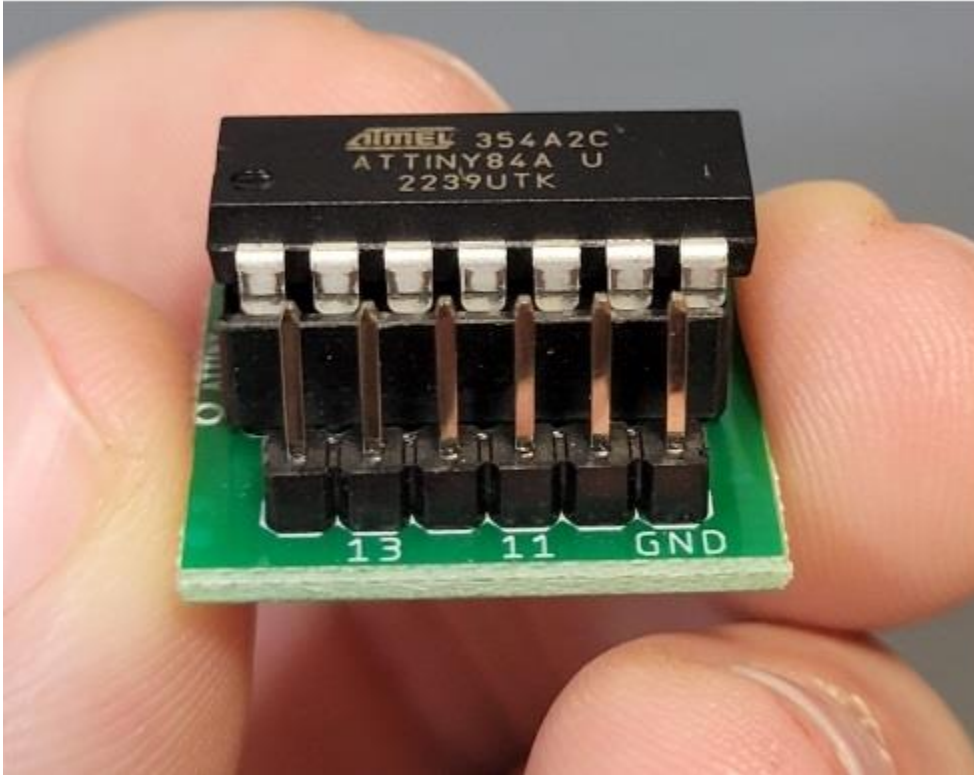


2. Notice that on one side of the chip and one side of the programmer there is a circle and a half circle cut out of the chip and programmer socket, make sure that when you place the chip on the socket that the shapes are on the same side and matching. If you put the

chip in backwards you can damage it when power is applied.

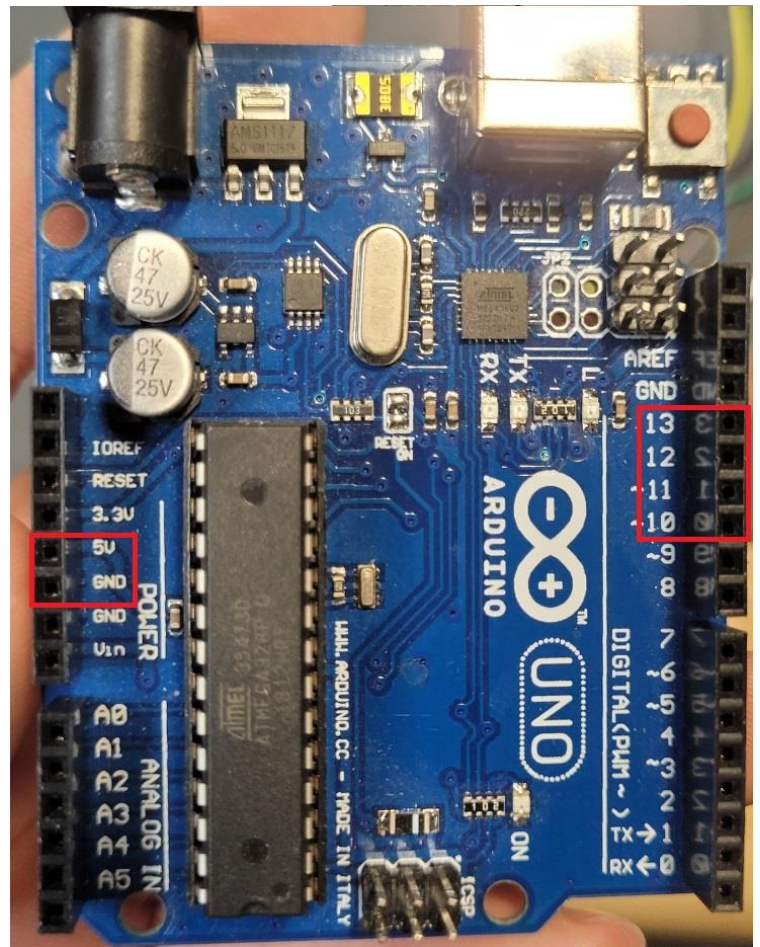
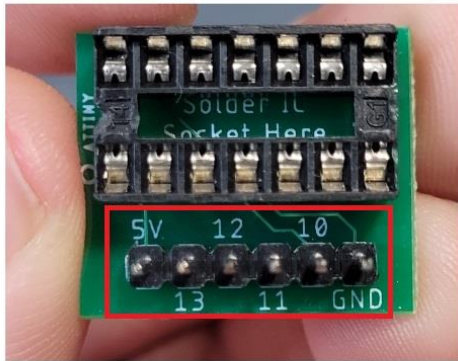


3. Place the chip on top of the socket with the correct orientation and use thumbs on both sides of the chip to gently press and wiggle the pin down into the socket until it is firmly in place. If you need to slightly bend some of the legs to better fit that is 100% fine.

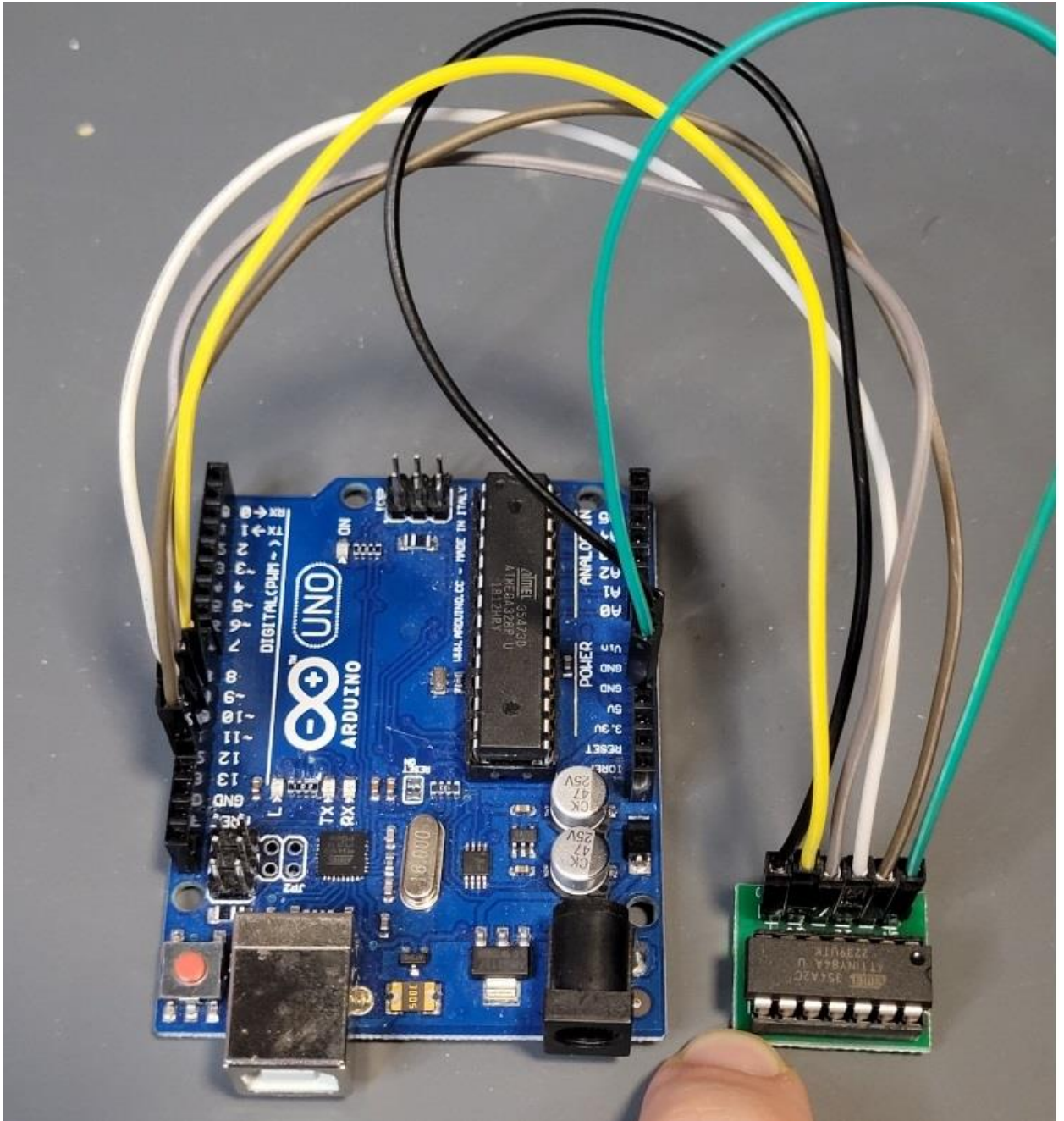


4. Now that our Attiny is in its programming socket we need to connect it to the Arduino. Find the 6 Jumper wires included with the Attiny Programmer.
5. Gently separate the wires one at a time so they are easier to work with.
6. On the Attiny Programmer there are pin designations: 5V, 13, 12, 11, 10, GND. Find these on the Arduino and use the jumper wires to connect them. Double check that the wires are in the correct place by counting the pins! Sometimes the numbers and Text can be

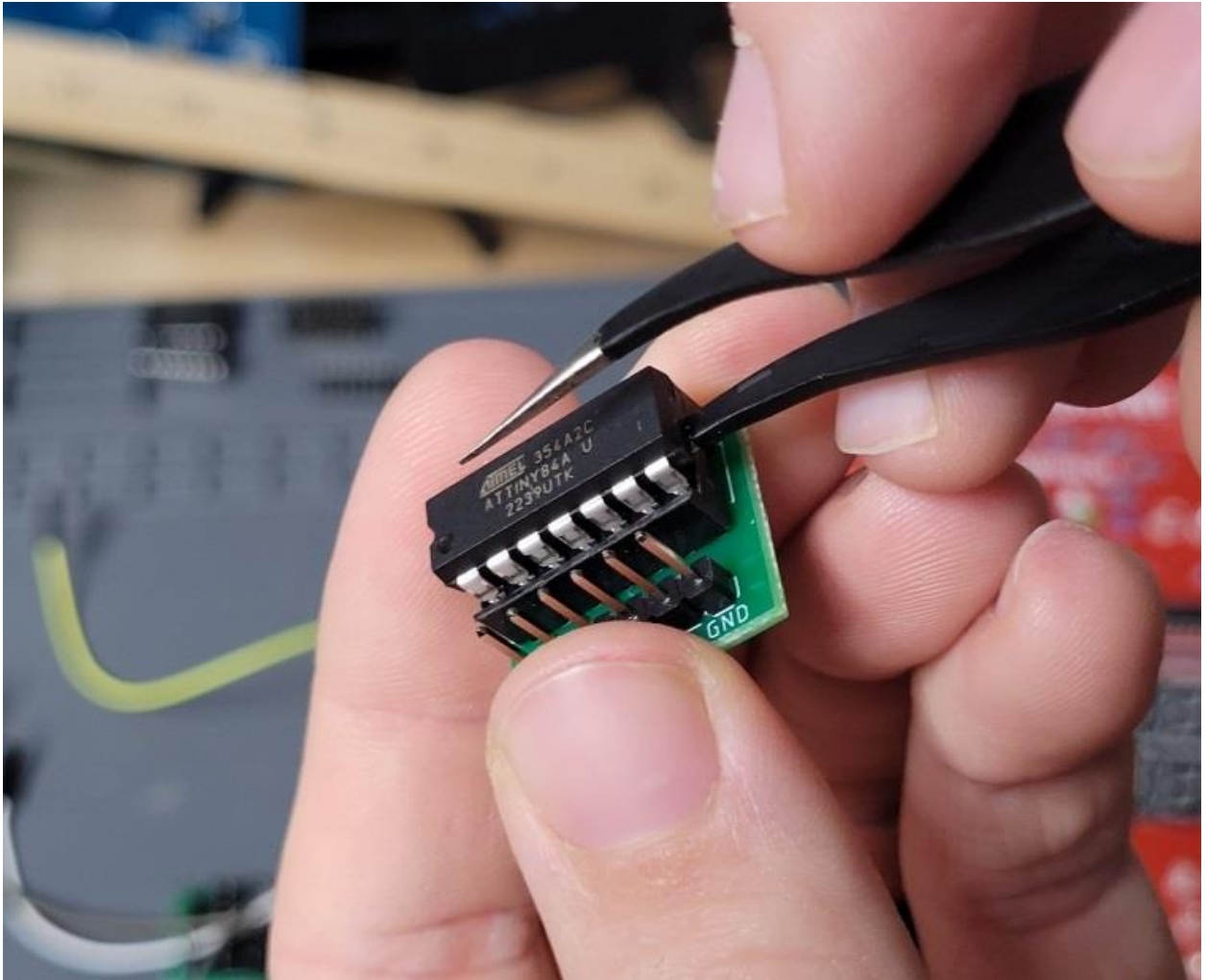
misleading.



7. Now that the Easy Programmer and Arduino are interfaced, it's time to upload some code!



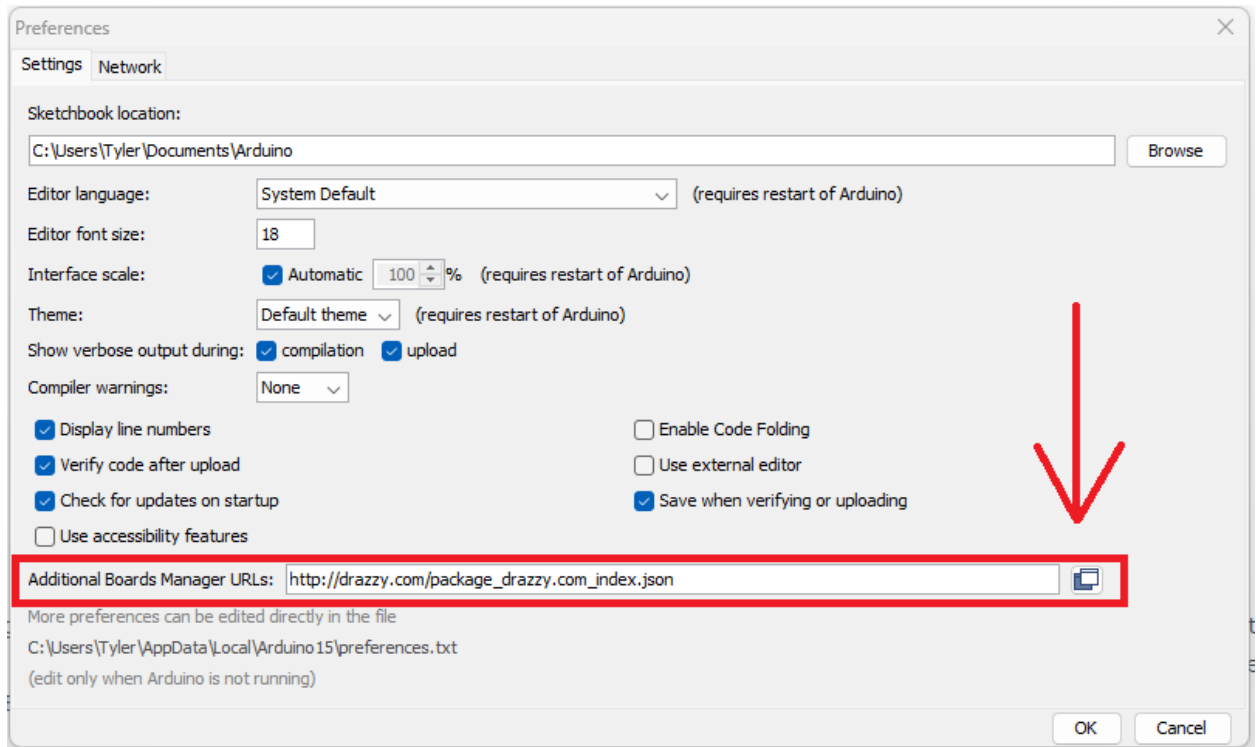
8. **IMPORTANT:** To Remove Attiny Chip from programmer, Use a thin pair of tweezers. Gently slide the tweezers between the gap of the chip and socket from the left- or right-hand side, and then gently rock the chip out of the socket, alternating from both sides until it has completely dislodged.



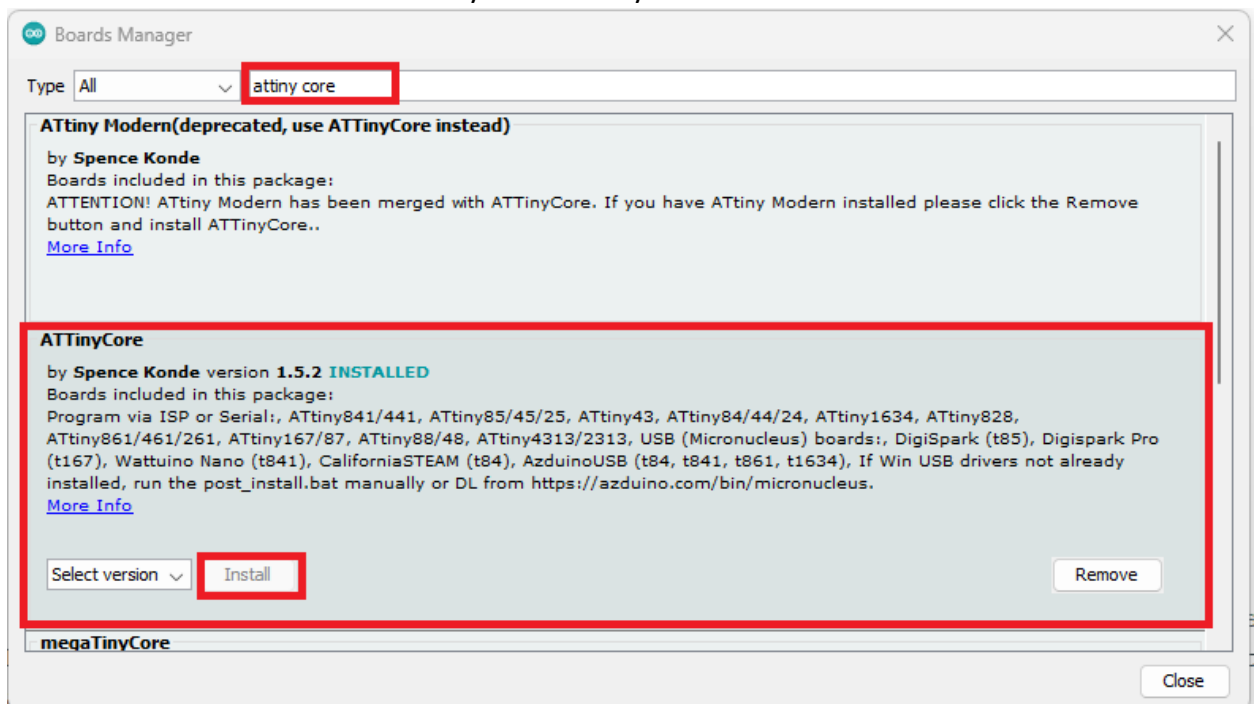
Programming the ATTINY 24/44/84: Now that our Arduino is setup as a programmer and connected to our Attiny 24/44/84 chip, we lastly need to install a library and set the correct upload settings for which ATTINY we are using.

1. Open the Arduino IDE code you wish to upload.
2. Now that your Arduino is connected to the easy programmer, connect your Arduino Uno, or Nano to your computer using the compatible USB Cable.
(NOTE: NOT ALL USB CABLES CONTAIN DATA LINES. CHEAP CABLES WILL PROVIDE POWER BUT BE UNABLE TO TRANSFER DATA / UPLOAD CODE**)**
3. Select File → Preferences

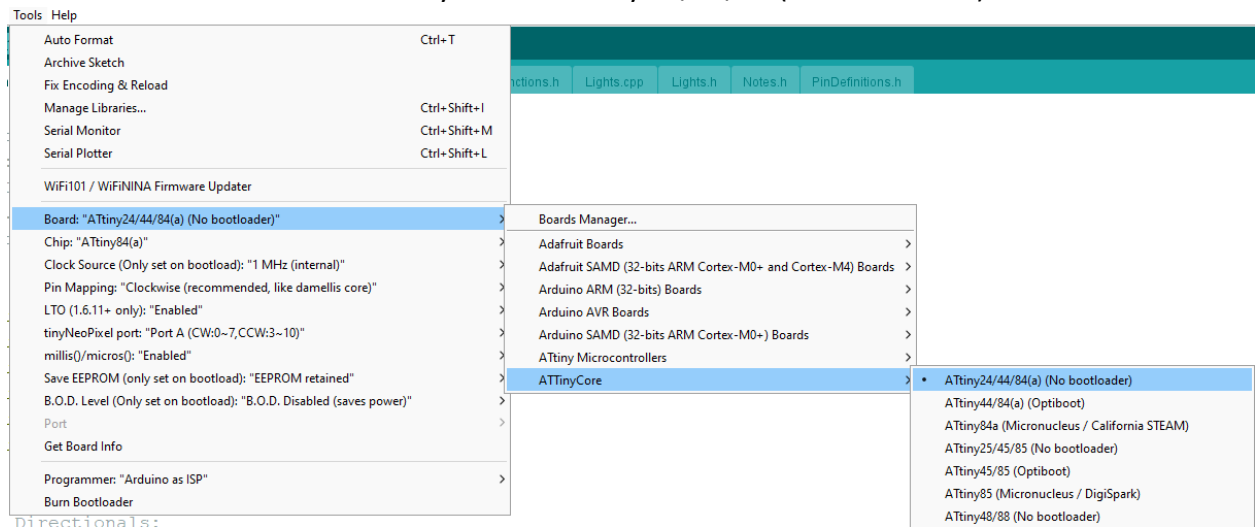
- A menu will pop up, find the section labeled “Additional Boards Manager URLs:” and paste this URL: http://drazzy.com/package_drazzy.com_index.json



- Click “OK”
- Click Tools → Board → Boards Manager
- The boards manager window will pop up. Enter “Attiny Core” into the search bar, and install the latest version of the Attiny Core Library



8. Now Click Tools → Board → Attiny Core → Attiny 24/44/84 (No bootloader)

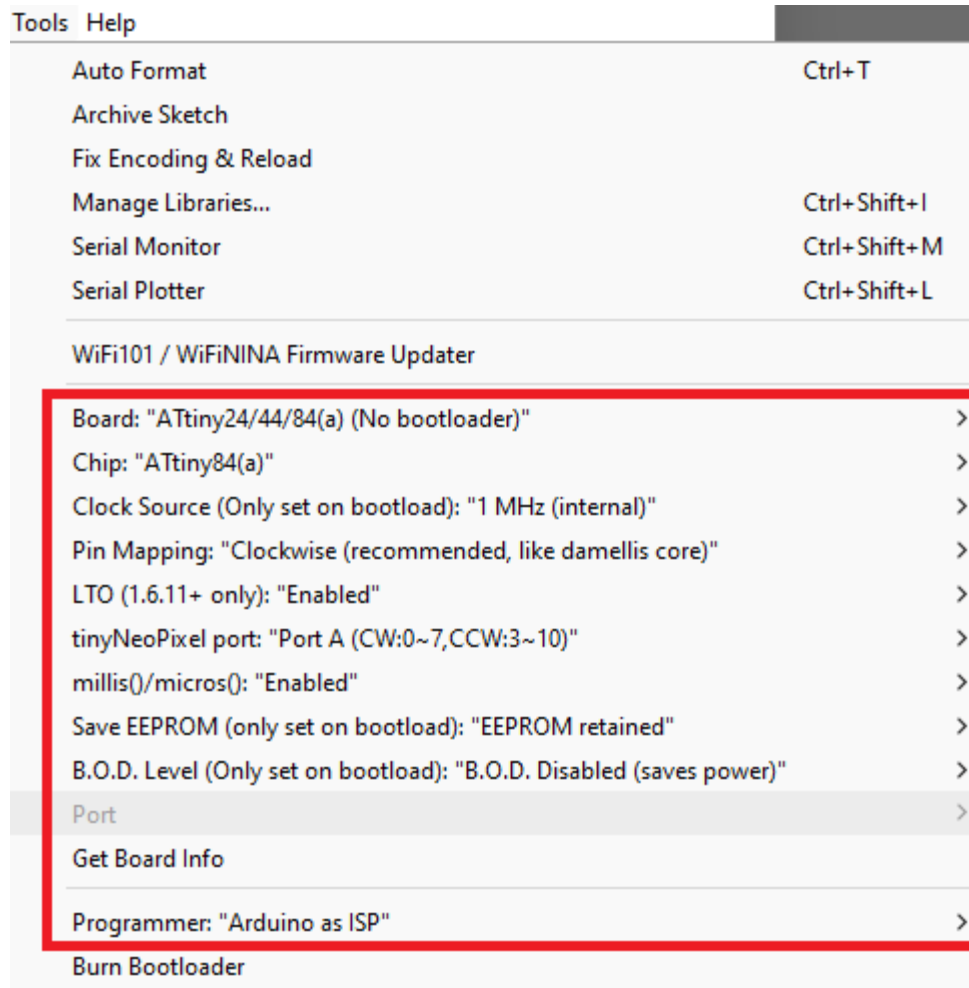


9. Click Tools → Chip → Attiny 84 (or whatever attiny chip your using)

10. Click Tools → Clock Source → 1MHz (internal)

11. Click Tools → Programmer → Arduino as ISP

12. Now all our other settings should be the recommended settings, but just to be sure take a minute and make sure your settings are the same as the image below.



13. Now we are ready to program! Select the port by clicking Tools → Port → (your port)

14. Now simply upload the code by clicking Sketch → Upload!

15. Remove the chip From the easy uploader by using tweezers and gently rocking it out of the socket, then place the Attiny Chip into its new home!

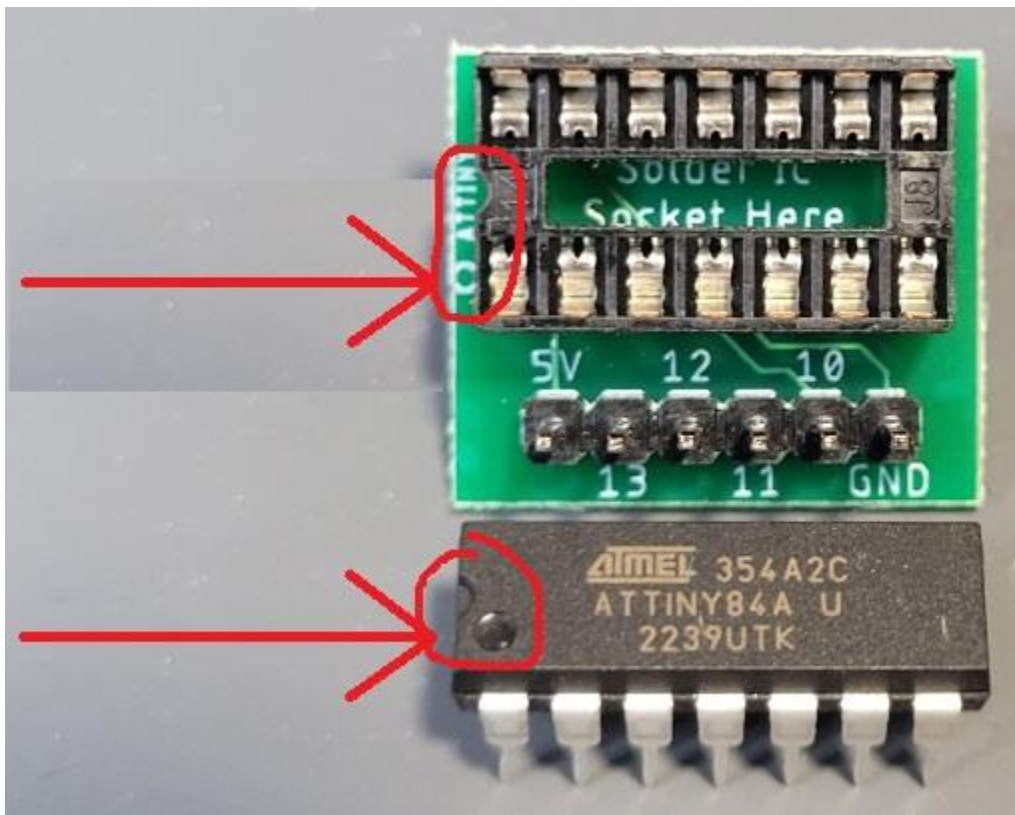
TROUBLESHOOTING

Most common error when trying to upload code to the attiny is invalid device signature:

```
avrdude: Device signature = 0x000000
avrdude: Yikes!  Invalid device signature.
        Double check connections and try again, or use -F to override
        this check.
```

This means your chip is not properly connected to your Arduino. To resolve this make sure:

1. Your chip is inserted properly into the Attiny Easy programmer. Make sure the part with the circle indent and semicircle chip line up with the circle print and semicircle chip of the socket.



2. Your wires have the proper connections. Count and make sure the physical pin location matches with where your wire is because sometimes the little silkscreen numbers can be very misleading about which pin they correspond to.
3. Make sure you properly set your Arduino to program the attiny. Revisit the Steps on setting up Arduino as Programmer by uploading the ISP code. **If you are using a nano make sure you uploaded the sketch using the proper bootloader, the default is Atmega328p but sometimes the old bootloader is used so try: Tools → Bootloader → Atmega328p (old bootloader)**
4. If the problem persists, you have rewired the board, checked the chip was inserted, Arduino set up as programmer then it could be that an internal fuse was set to NON_PROGRAMMABLE. This should never happen with the chips I provide you because I buy mine brand new direct from American suppliers, and I test every single chip before I send it to you. This means your chip was used or damaged and the only way to fix it is HIGH VOLTAGE REPROGRAMMING. <https://www.electronics-lab.com/recover-bricked-attiny-using-arduino-as-high-voltage-programmer/>
Don't try to reprogram it yourself, I just put that there for documentation and education purposes. Reach out to me via email or in the comments section of QR page and I can help you figure out how to trouble shoot your issue.