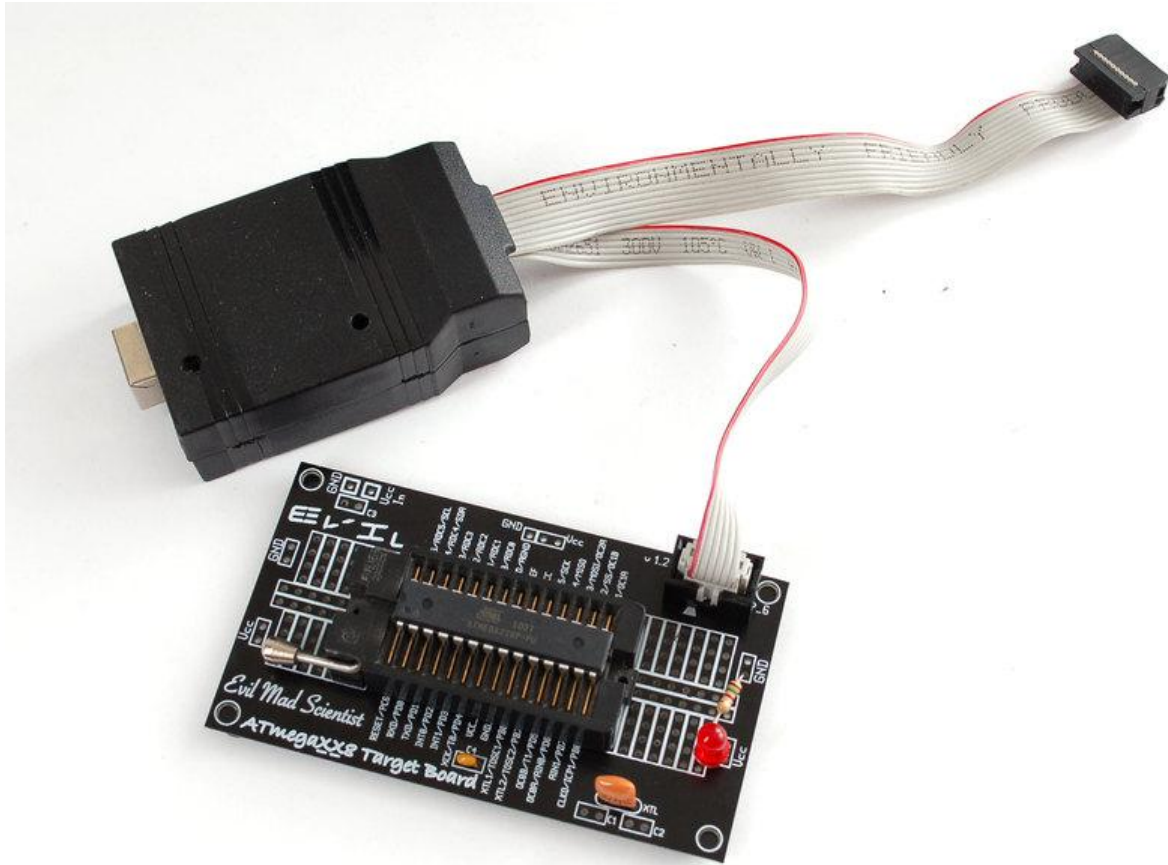




Mass AVR ISP Programmer

Created by lady ada



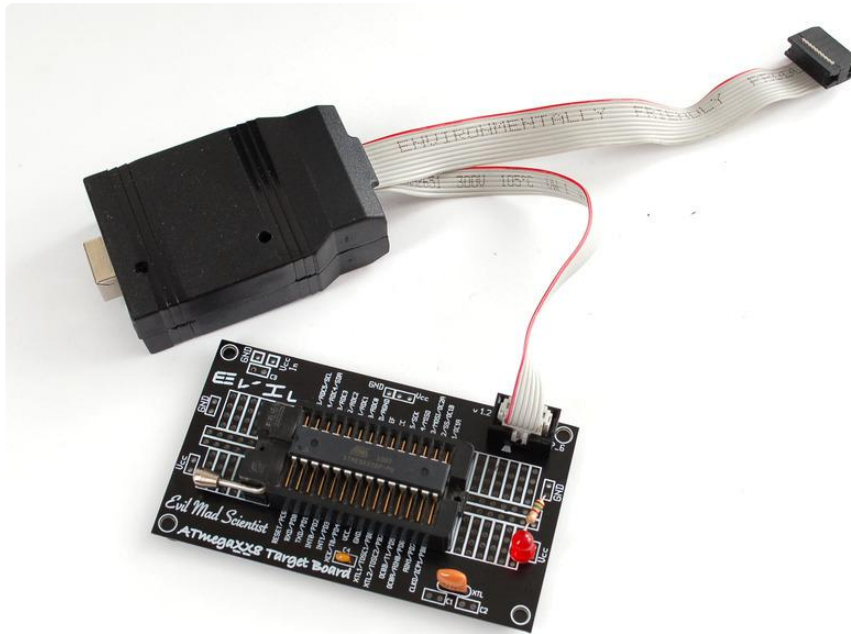
<https://learn.adafruit.com/mass-avr-isp-programmer>

Last updated on 2021-11-15 05:59:22 PM EST

Table of Contents

| | |
|------------|---|
| Overview | 3 |
| Parts List | 3 |
| Assembly | 8 |
| Use It! | 9 |

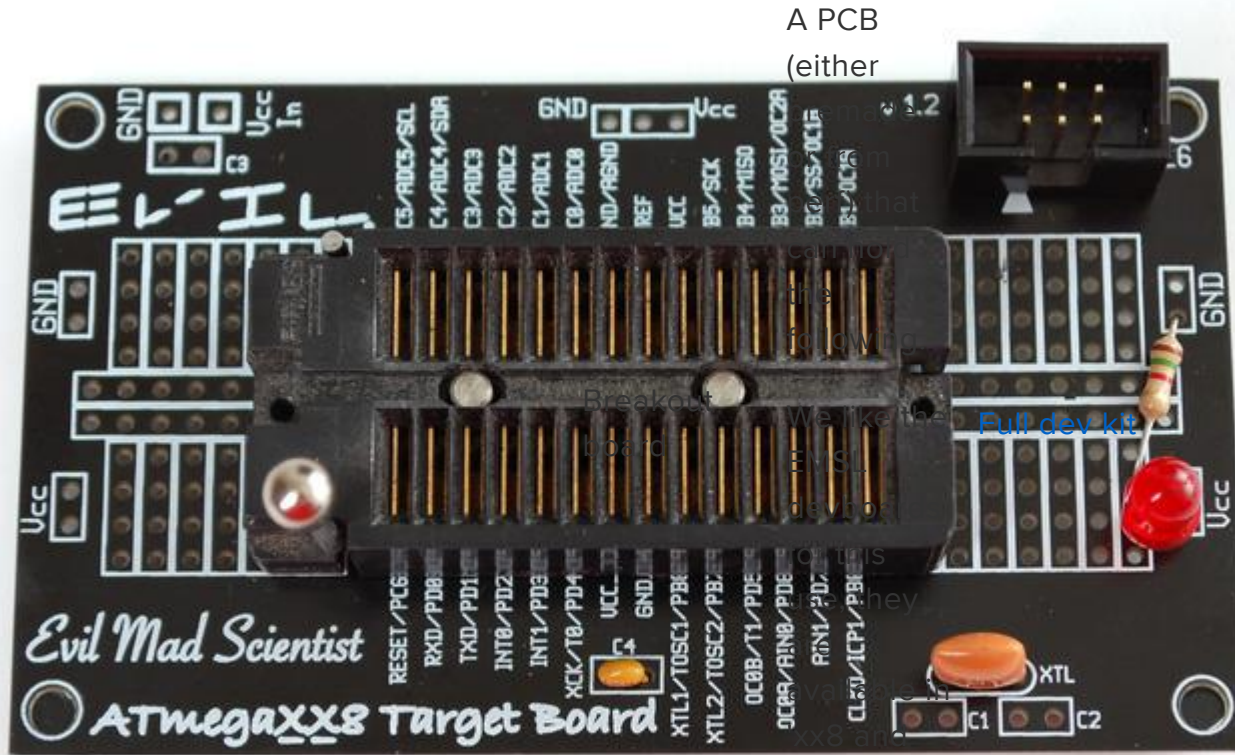
Overview



We often get the question "How do you program all the chips in your kits?" Well! Now we have a nice tutorial showing how its done!

Parts List

| Image | Name | Description | More info |
|-------|------|-------------|-----------|
|-------|------|-------------|-----------|



A PCB
(either

Breakout
board

that
can be
used
to
program
the
microcontroller
or
to
test
it
in
a
circuit.

Full dev kit

Evil Mad Scientist
ATmegaXX8 Target Board

'2313
formats.

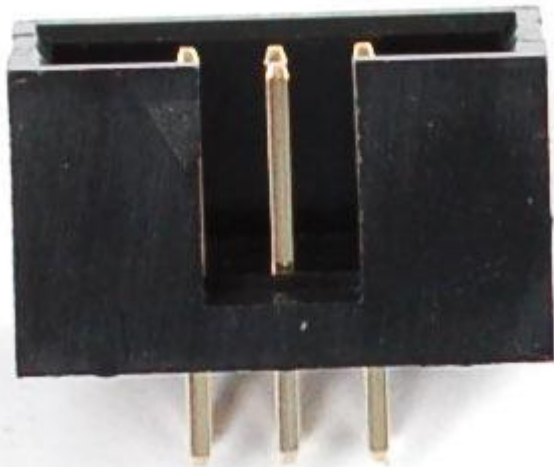


An easy-to-open ("zero insertion force") and close socket, we prefer the ones with gold contacts. ZIF socket a little pricier but will last a lot longer. Make sure they are 'universal' - can take both 0.3" and 0.6" wide chips!



Indicator LED

Use it for 'power good' or other blinky. 5mm red diffused

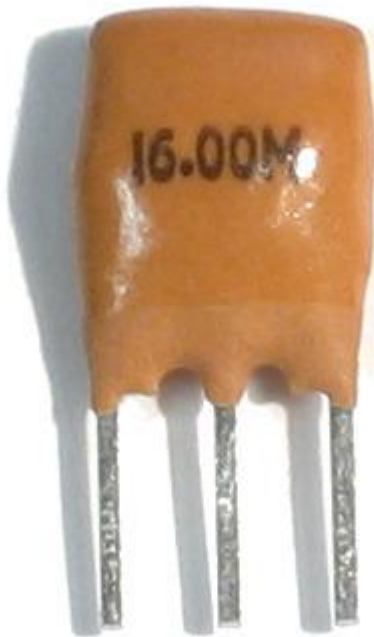


6-pin box header

With a key, its less likely to insert the cable backwards.

[2x3 pin header](#)

[6pinbox.jpg](#)



Ceramic Oscillator

16MHz is good, but really anything >4MHz is OK.

[Ceramic resonator](#)

[xtal16m.jpg](#)

| | | | |
|---|--|---|--|
|  | <p>Ceramic 0.1uF capacitor (104)</p> | <p>Nearly any capacitor will do, just to provide a little power supply smoothing.</p> | <p>Ceramic Capacitor</p> |
|  | <p>Rubber bumper</p> | <p>For protecting the PCB!</p> | <p>Rubber bumper</p> |



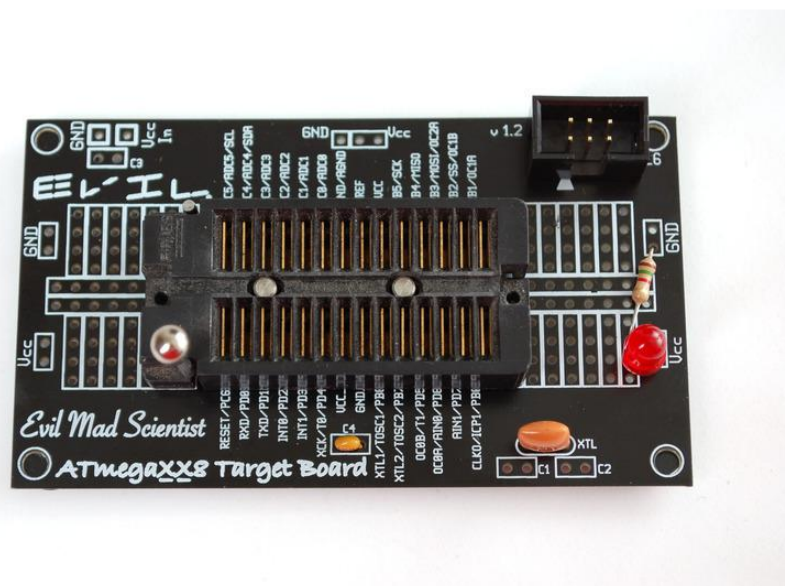
We like the USBtinyISP - not just because we designed it but also because it can provide 'target power' - so you don't need a separate power plug!

USBtinyISP

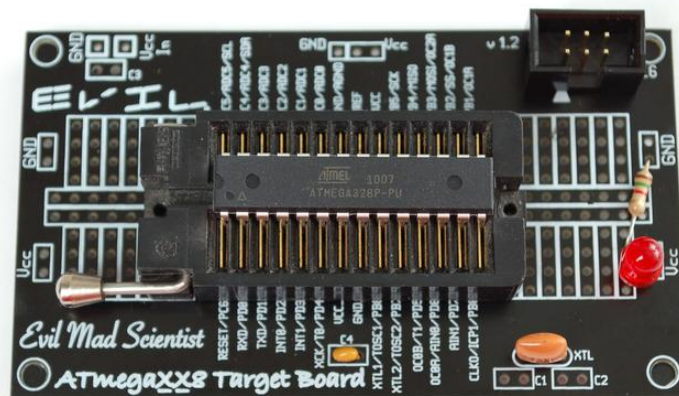
Assembly

There's not a lot going on here, basically solder the ZIF socket in and wire up the socket! The crystal will make it easy to program lots of Arduino chips (which is what we've used this setup for)!

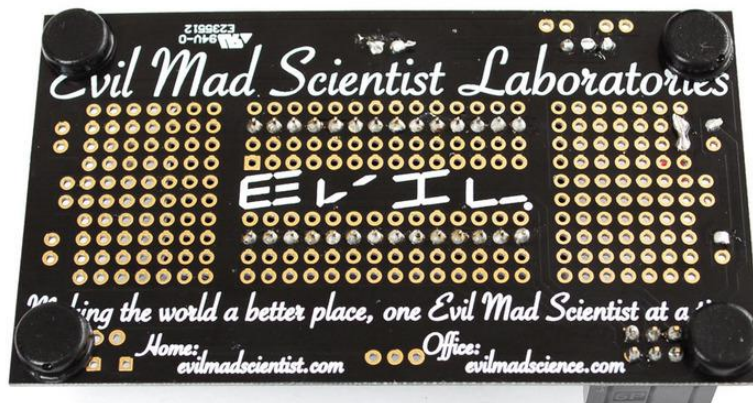
The magic in the pudding is the ZIF socket, which is basically a latching socket that when open is easy to insert chips without bending any pins and when closed is very secure. They are very satisfying to use and last for tens of thousands of cycles!



The latch up, you can see the gold contacts.



And down (holding a chip).



We put bumpers on the bottom - this will prevent accidental shorts (if you have a messy desk - which we do).

Use It!

This kind of board works best with a target-powering programmers such as an STK500 (or dragon) or a USBtinyISP.

```
C:\WINXP\system32\cmd.exe
C:\>avrdude -c usbtiny -p attiny2313 -U flash:w:test_leds.hex
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.02s
avrdude: Device signature = 0x1e910a
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "test_leds.hex"
avrdude: input file test_leds.hex auto detected as Intel Hex
avrdude: writing flash (260 bytes):
Writing | ##### | 100% 0.73s

avrdude: 260 bytes of flash written
avrdude: verifying flash memory against test_leds.hex:
avrdude: load data flash data from input file test_leds.hex:
avrdude: input file test_leds.hex auto detected as Intel Hex
avrdude: input file test_leds.hex contains 260 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 0.50s

avrdude: verifying ...
avrdude: 260 bytes of flash verified
avrdude: safemode: Fuses OK
avrdude done. Thank you.

C:\>_
```

For software we use [AVRDUDE \(https://adafru.it/a15\)](https://adafru.it/a15) with a Makefile so we can just type in "make program" or "make full" or simply a command like above and then watch it go. For the next chip, press the Up Arrow to re-run the last command. Zoom!