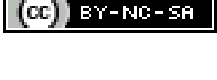
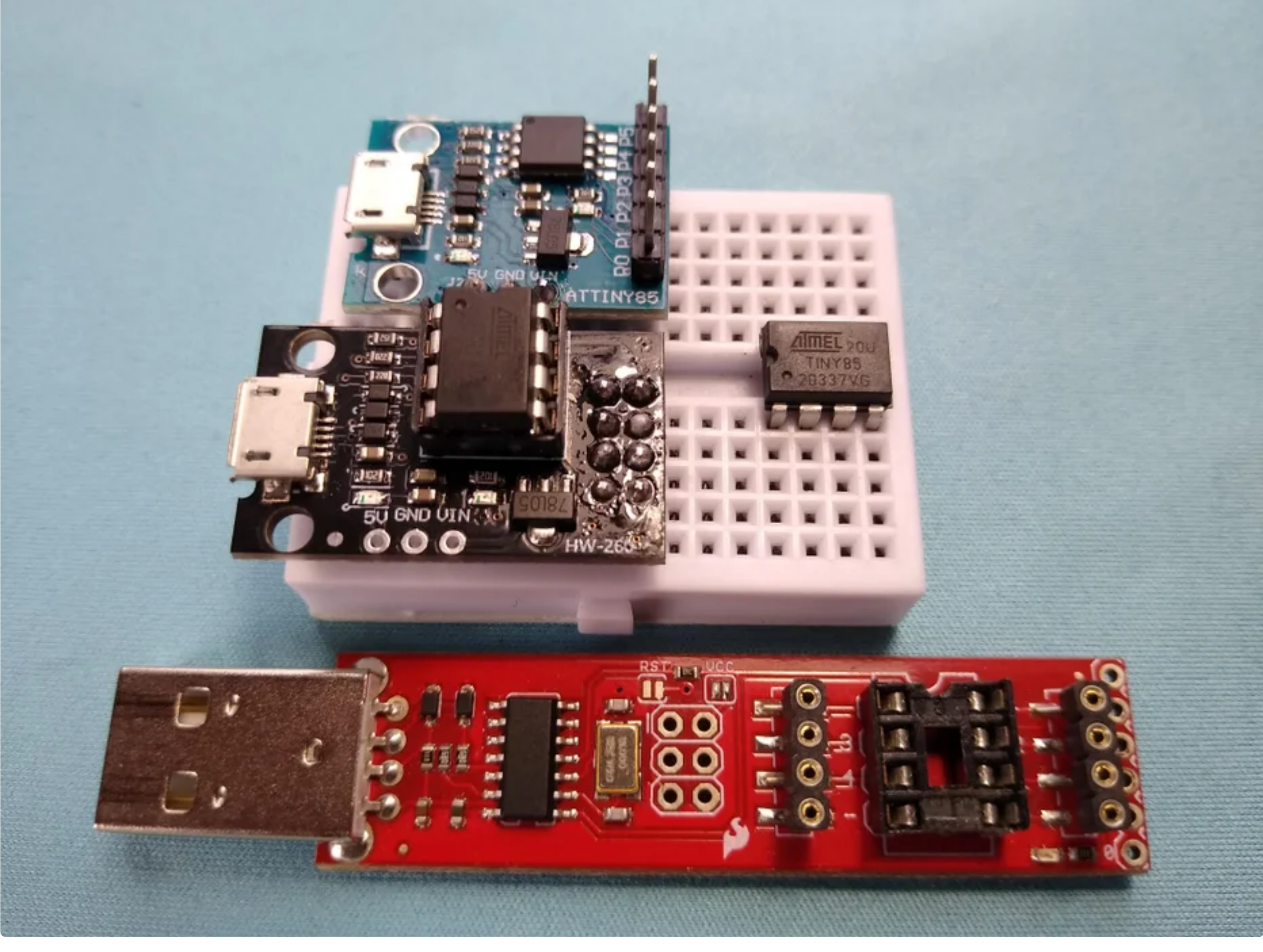


ATTinyCore, HW-260 and the ATTINY85

By grovedc in Circuits > Arduino 9,643 7 1



Save PDF



The ATTiny85 is one of my favorite programmable devices. In this 8 pin device, you get 512 bytes of RAM memory, 8K bytes of ROM memory, EEPROM and so much more. It is very inexpensive, and is also a great place to get started with Arduino.

I was recently asked by a friend to help her get started. This can be a bit confusing, particularly considering the number of options available. Also, the state of play has been moving a lot lately and while many older tutorials are still current, others are not.

I'd like to go through getting started with the HW-260 board. I had not seen this before, and I found it to be very interesting. Before that, though, let's look at the other boards very quickly.

The ATTINY85 board has been around for some time now. For about \$2 you get the ATTiny85, a USB connector, and a 5 volt voltage regulator. An LED is connected on pin 1. With the voltage regulator, you can connect an external power source and have the voltage regulated down to 5 volts. The ATTINY85 requires a bootloader to function. The bootloader will watch the USB port when first powered on. This allows the device to be programmed. If no USB activity is detected, then the bootloader will hand off to the user code. I am using Micronucleus version 2.5, and this version leaves 6586 bytes available for the user code. Note that most ATTINY85 boards come preinstalled with an older version of Micronucleus, usually 1.6 or 1.2, and you can use ATTinyCore to update the bootloader.

The red board is the Sparkfun Tiny Programmer. There are many ISP programmers available. I have found this programmer to be very reliable, and I recommend it. I will be using this board in this tutorial to program the ATTiny85.

The HW-260 board is available on AliExpress for 45 cents US. The listing calls the board:

Pluggable ATTINY Development Board For ATtiny13A/ATtiny25/ATtiny45/ATtiny85 Programming Editor Micro Usb Power Connector

As mentioned, I had never heard of this board before. So what exactly is this board? I couldn't see how it could be used as a programmer. It would need a processor on the board to be an ISP. You can't just plug in a Tiny85 and program it. So I uploaded a blink sketch to a Tiny85 using the Sparkfun Tiny Programmer. This worked just fine, with the LED being on pin 1. So this got me thinking, can I actually load a bootloader on to a bare Tiny85 and put it on the HW-260. As it turns out, yes you can. Effectively, the HW-260 is the same as an ATTINY85, but with a socket for a ATTiny85. This means that you can use the HW-260 just like a ATTINY85, or you can have no bootloader at all and have all 8K of ROM available. The HW-260 actually be programmed and used as an ISP.

This can all be done using just the ATTinyCore board in the Arduino IDE.

Supplies

You will need the Arduino IDE installed, as well as:

- An an ISP programmer. I am using the Sparkfun Tiny Programmer.
- A HW-260 board
- One or more ATTiny85 chips

Step 1: Configure the Arduino IDE



Providing support for the attinycore is done by installing the boards manager URL on the preferences page, and then installing the board in the boards manager. This can be done as described on the projects [Home Page](#).

Note as well that on the preferences page, there are two buttons next to, "Show verbose output during:". Enabling these two options show a lot of very good information for the process described below.

You can check the installation in the boards manager as shown in the image above.

Step 2: Programming the ATTiny85 With the ISP



Place your ATTiny85 chip into the Sparkfun Tiny Programmer. Take care to get the notch on the chip aligned with the notch in the center of the socket on the ISP. The ATTiny85 is shipped with the clock set to 1 mHz. The first step will be change the clock to 8 mHz.

This is shown in the first image above. Note that the programmer is set to "for new or 1 mHz parts".

Go ahead and hit the "Burn Bootloader" button at the end of the list. Note that we aren't actually burning a bootloader yet. If you have the verbose logging enabled, you can see that the file that is loaded is:

```
arduino15/packages/ATTinyCore/hardware/avr/1.5.2/bootloaders/empty/empty_all.hex
```

Or, an empty bootloader. But the ATTiny85 is now configured for 8mHz internal clock.

Step 3: Burn the Initial Bootloader



Now, with the ATTiny85 chip still in the Sparkfun Tiny Programmer ISP, configure the board as shown above. Note that the board is: ATTiny85 (micronucleus DigiSpark). The Clock is 8 mHz (No USB). The Burn Bootloader Method is "Fresh Install (via ISP)", and that the programmer is the "USBtinyISP", but now for the faster parts.

This will install:

```
arduino15/packages/ATTinyCore/hardware/avr/1.5.2/bootloaders/micronucleus/t85_entry_on_power_on_no_pullup_fast_exit_on_no_USB.hex
```

on to the ATTiny85 chip.

I usually get an error at the end of this burn, but it does complete successfully.

Step 4: Burn the Final Bootloader



Now, take the ATTiny85 chip from the Sparkfun Tiny Programmer ISP and place it on the HW-260. Once again watching the proper orientation of the notch.

Configure you board as shown above and burn the bootloader. This will install:

```
arduino15/packages/ATTinyCore/hardware/avr/1.5.2/bootloaders/micronucleus/upgrade-t85_entry_on_power_on_no_pullup_fast_exit_on_no_USB.hex
```

With the clock set to 16.5 mHz on the ATTiny85.

You can now load your sketch to your HW-260 board with these same settings.



Categories

- Circuits
- Workshop
- Design
- Craft
- Cooking
- Living
- Outside
- Teachers

About Us

- Who We Are
- Why Publish?

Resources

- Get Started
- Help
- Sitemap

Find Us

