TinyTracker Configuration with AVRDude

While this technique may look a bit more complicated than using the Arduino IDE to upload a program that writes desired values to EEPROM, the Arduino IDE approach then requires us to still resort to a command line to upload the original tracking program firmware again from a hex file. The techniques outlined here only edit the EEPROM memory containing the tracking control parameters, and not the program flash memory.

Necessary Hardware

These instructions are written for USBasp AVR programmers as these are relatively cheap and commonly available. If you have another compatible AVR programmer (with 10 to 6 pin adapter) then substitute the appropriate label for the "-c usbasp" parameter used below.

Getting Started

Open a command window in the firmware directory where the avrdude.exe program is located. (Windows users right-click on the Windows icon in your task bar and select **Command Prompt,** then enter CD commands as needed to navigate to the firmware directory**)**

Plug-in your USBasp programmer into your computer and connect the end of the cable with the 10 to 6 pin adapter to the ISP header on the TinyTracker board. Make sure the MISO signal from the adapter connects to the pin 1 on the pcb ISP header.

First-time update

* TinyTracker comes with Hfuse set to 0xD7 which preserves EEPROM during AVRdude flash write operations. In order to upload new values the Hfuse has to be changed to 0xDF.

Command line for settting fuse to enable avrdude to write to eeprom:
avrdude -C ..\etc\avrdude.conf -p attiny85 -c usbasp -U hfuse:w:0xdf:m

This only needs to be done once and we don't ordinarily need to worry about changing it back.

Backup Current EEPROM Configuration

This command will allow you to make a backup of eeprom to file (save.out):
avrdude -C ..\etc\avrdude.conf -p attiny85 -c usbasp -U eeprom:r:save.out:i

Restoring/Uploading EEPROM Configuration

The syntax for restoring a backup eeprom file (save.out) is the same as uploading customized hex configuration values :
avrdude -C ..\etc\avrdude.conf -p attiny85 -c usbasp -U eeprom:w:save.out:i

Configuration Hex File Format

The ":i" option added to our avrdude command line instructs avrdude to use what is known as standard

Intel binary hex file format. Each line starts with a colon followed by 8 digits which encode the starting memory address for each line. The sixty-four digits that follow represent 32 bytes of memory, and are followed by a CRC check sum value based on all 72 hex values.

Default values look like this:

```
-address|bal day sun period  max_run max_trakEIaxPm|--------------------CHECKSUM
:200000000000C201E001B8F9020030E60200881300000000FFFFFFFFFFFFFFFFFFFFFFFFE2
:00000001FF
```

The last line tells avrdude that there is no more input and should never be changed.

Note that byte order within each field is reversed, and two hex digits form one byte!

8+ Offset	Parameter	Default Decimal	Hex Value	#Bytes	Reversed Byte Order	Units
0	Balance	0	0000	2	0000	N/A
4	Daylight	450	01C2	2	C201	N/A
8	Sunlight	480	01E0	2	E001	N/A
12	Track period	1900000	0002F9B8	4	B8F90200	milliseconds
20	Max run	1800000	0002E630	4	30E60200	milliseconds
28	Max track		00001388	4	88130000	milliseconds
36	East-Inhibit	0	00	1	00	N/A
38	Alt-axis	0	00	1	00	N/A
40	PWM	255	FF	1	FF	N/A

Making a custom configuration file for uploading

Example: daylight threshold value in our file is C201, but the hex conversion for 450 (the default value) is 0x01C2.

Copy the **save.out** EEPROM data file to a new file. In this example we'll be using "TTday460.hex"

If we want to increase the daylight value to 460 (hex: 01CC) then the new eeprom file line would like like:

```
:200000000000CC01E001B8F9020030E60200881300000000FFFFFFFFFFFFFFFFFFFFFFFF
               ^changed hex digit
```

We've left the checksum off of the end of the new line, because we still need to calculate the new checksum. Copy the entire line (without the colon in front) and paste that into the CheckSum calculator link show below.

The CheckSum calculator says the new check sum is D8, so now the line looks like:

```
:200000000000CC01E001B8F9020030E60200881300000000FFFFFFFFFFFFFFFFFFFFFFFFD8
```

```
So we save the following to our input file, TTday460.hex
:200000000000CC01E001B8F9020030E60200881300000000FFFFFFFFFFFFFFFFFFFFFFFFD8
:00000001FF
```

Lastly, tell AVRDude to upload our input file to eeprom:

```
avrdude -C ..\etc\avrdude.conf -p attiny85 -c usbasp -U eeprom:w:TTday460.hex:i
```

Tracking Firmware Uploads

In the event that you issue an AVRDude command which renders the tracking controller inoperable, you will need to reinstall the baseline program firmware.

```
avrdude -C "..\etc\avrdude.conf" -c usbasp -p t85 -U flash:w:"C:\Users\YourName\Documents\TinyTracker_revH.hex":i
```

Useful links:
For converting decimal numbers to hex values:
http://www.binaryconvert.com/convert_signed_short.html

For calculating new checksum component:
http://easyonlineconverter.com/converters/checksum_converter.html