LOL, it has been like a year since I've done this, and it was difficult to find then, and apparently still is. Stuart needs to make a video about it :-)

For now, dummy guide:

Download AVRDUDE 6.3, extract in ease directory, like c:\tmp or so.

insert USBASP programmer into USB port

To get the programmer working (windows drivers) you need to use Zadig http://zadig.akeo.ie/

plug in the controller. You can try to start avrdude but it will not find it.

C:\tmp\avrdude-64>avrdude -C avrdude.conf -P usb -c usbasp -p t841 avrdude: error: could not find USB device with vid=0x16c0 pid=0x5dc vendor='www.fischl.de' product='USBasp'

zadig will do it automatically, click install driver **libusb-win32**

Zadig	- 🗆 ×
Device Options Help	
Driver libusb0 (v1.2.6.0) USB ID 16C0 05DC WCID ² X	More Information <u>WinUSB (libusb)</u> <u>libusb-win32</u> <u>libusbK</u> <u>WinUSB (Microsoft)</u>
6 devices found.	Zadig 2.7.765

(choose this one, **libusb-win32**)

If it doesn't work, check out this:

https://electronics.stackexchange.com/questions/416714/avrdude-does-not-recognize-usbaspdevice/417509#417509

or

https://www.robolab.in/installing-usbasp-driver-for-windows-7-and-above/

command prompt (windows key + "R", type cmd, press enter), type: C:\tmp\avrdude-64>avrdude -C avrdude.conf -P usb -c usbasp -p t841

(you can copy and paste)

C	C:\Windows\system32\cmd.exe					
ā	Restore Move	avrdude.conf	-P usb -	c usbasp -p t841		
-	Size Minimize Maximize	requency to 187500 Hz cannot set sck period. please check for usbasp firmware update. e initialized and ready to accept instructions				
×	Close					
	Edit >	Mark	Ctrl-M			
	Defaults	Сору	Enter	obably t841)		
	Properties	Paste	Ctrl-V			
av	rauae: satemoae	Select All	Ctrl-A	L:62)		
avrdude done. Tha		Scroll				
		Find	Ctrl-F			
C:	C:\tmp>_					

It still won't work yet, as it standart doesn't support Attiny 841.. Step 9 of stuart: The standard avrdude tool doesn't include support for ATTINY841 chips. So download and overwrite the file **avrdude.conf using the file from <u>here</u>**

After this, start cmd, navigate to the correct path.

If you used c:\tmp you need to first go to the root: type "cd\" Sometimes a few times, depending where you started (it does one level) if all correct you see "C:\ >"

type "CD tmp" (or the name of your directory that contains avrdude)

type or copy the code

avrdude -C avrdude.conf -P usb -c usbasp -p t841 -e -B 8 -U efuse:w:0xF4:m -U hfuse:w:0xD6:m -U lfuse:w:0x62:m -U flash:w:diybms_module_firmware_400.hex:i

result of programming :

C:\tmp>avrdude -C avrdude.conf -P usb -c usbasp -p t841 -e -B 8 -U efuse:w:0xF4:m -U hfuse:w:0xD6:m -U lfuse:w:0x62:m -U flash:w:diybms_module_firmware_400.hex:i

avrdude: set SCK frequency to 93750 Hz avrdude: warning: cannot set sck period. please check for usbasp firmware update. avrdude: AVR device initialized and ready to accept instructions

//. I Made the output short here. //
avrdude: safemode: Fuses OK (E:F4, H:D6, L:62)

avrdude done. Thank you.

https://github.com/stuartpittaway/diyBMSv4Code

Programming the modules

Module code runs on the ATTINY841 micro controller, it is important to program the chip with the correct version of code depending on your PCB version.

You will need a programming device capable of programming ATMEL AVR chips - like the <u>USBASP programmer</u>

Setup the programmer

- 1. Connect the USBASP programmer to the computer
- 2. On the programmer, move the jumper pin (normally marked JP1) to use 3.3 volt programming settings (instead of 5 volt)
- 3. Completely disconnect the module from any battery/cell and the TX/RX connectors should also be unconnected.
- 4. Connect the programmer to the module using the 6 pin ISP connector on the module take great care to ensure PIN 1 is aligned to PIN 1 of the programmer. PIN 1 is marked on the PCB.
- 5. Download AVRDUDE 6.3 or newer, for <u>Windows</u> other versions are <u>here</u>
- 6. Extract the AVRDUDE zip file
- 7. Open a console/command window and change to the folder where you extracted the AVRDUDE program, on Windows this looks similar to this

cd C:\temp\avrdude-6.3-mingw32

- 9. The standard avrdude tool doesn't include support for ATTINY841 chips. So download and overwrite the file avrdude.conf using the file from <u>here</u>
- 10. Lets test connectivity to the programmer and module. Back in the console window, run the command below. On Linux and Mac operating systems, you may need to use a different port insteoad of "usb" for example /dev/tty1 but this will vary depending on the computer. Note that the parameters ARE case sensitive.

avrdude -C avrdude.conf -P usb -c usbasp -p t841

11. All being well, it should report something similar to the below. If not, check the wiring and ensure you are using the correct COM port.

Programming the module

Programming the module takes around 12 seconds.

- 1. Identify which module/board you have using the details found at the end of this document.
- 2. Copy the required ".hex" file to the same folder where you extracted the avrdude tool to.
- 3. Now we shall program the module, run the command line similar to below, replacing the "diybms_module_firmware_400" filename where applicable.
- 4. The fuse settings are important, and are in the filename for example "eF4_hD6_l62" means efuse=0xF4, hfuse=0xD6, lfuse=0x62

avrdude -C avrdude.conf -P usb -c usbasp -p t841 -e -B 8 -U efuse:w:0xF4:m -U hfuse:w:0xD6:m -U lfuse:w:0x62:m -U flash:w:diybms_module_firmware_400.hex:i it should output

```
avrdude: set SCK frequency to 187500 Hz
avrdude: AVR device initialized and ready to accept instructions
avrdude: Device signature = 0x1e9315 (probably t841)
avrdude: erasing chip
avrdude: set SCK frequency to 187500 Hz
avrdude: reading input file "0xF4"
avrdude: writing efuse (1 bytes):
avrdude: 1 bytes of efuse written
avrdude: verifying efuse memory against 0xF4:
avrdude: load data efuse data from input file 0xF4:
avrdude: input file 0xF4 contains 1 bytes
avrdude: reading on-chip efuse data:
avrdude: verifying ....
avrdude: writing hfuse (1 bytes):
avrdude: 1 bytes of hfuse written
avrdude: reading on-chip hfuse data:
avrdude: verifying ...
avrdude: writing lfuse (1 bytes):
avrdude: 1 bytes of lfuse written
avrdude: verifying lfuse memory against 0xE2:
avrdude: reading on-chip lfuse data:
avrdude: verifying ....
avrdude: writing flash (7718 bytes):
```

- 1. If programming fails, but the programmer appears to be communicating, try increasing the value of the "B" setting from 8 to 16 to slow down the USBASP device.
- 2. Check that the fuses report as "OK" and read E:F4, H:D6, L:E2
- 3. That module can now be disconnected from the USBASP programmer, connect the next module and repeat the avrdude command to program the next one.