

Contents: [Dobrica Pavlinu's random unstructured stuff]

- Dobrica Pavlinu's random unstructured stuff (Similar boards)
 - ◆ Dobrica Pavlinu's random unstructured stuff (B3606)
 - ◆ Dobrica Pavlinu's random unstructured stuff (D3806)
- Dobrica Pavlinu's random unstructured stuff (B3603)
- Dobrica Pavlinu's random unstructured stuff (Specification)
- Dobrica Pavlinu's random unstructured stuff (backup existing flash)
- Dobrica Pavlinu's random unstructured stuff (unprotect stm8 using stm8flash)
- Dobrica Pavlinu's random unstructured stuff (compile)
- Dobrica Pavlinu's random unstructured stuff (flash)
- Dobrica Pavlinu's random unstructured stuff (serial)
- Dobrica Pavlinu's random unstructured stuff (calibrate)

Similar boards

B3606

<https://github.com/UsnameTaken/B3606>

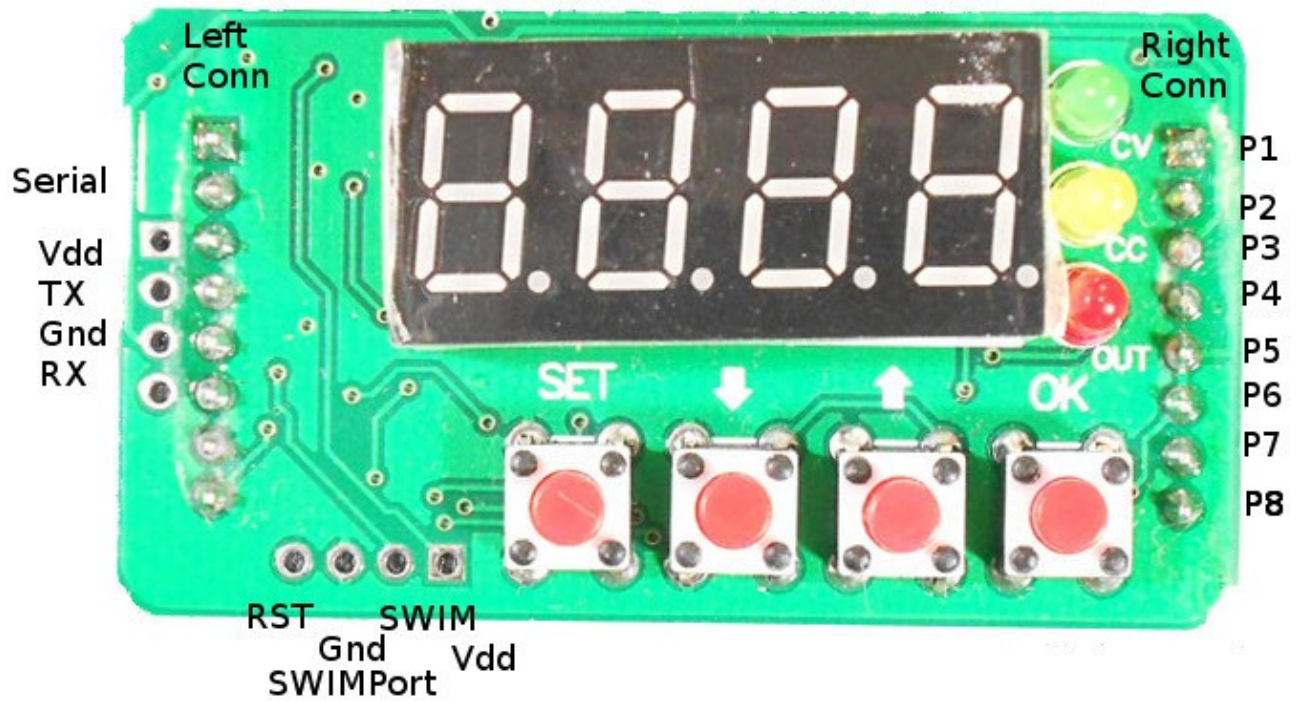
D3806

I have unit, there is no alternative firmware for this model but there is for:

<https://github.com/delboy711/BST900>

<https://www.falatic.com/index.php/161/minghe-buckboost-converters-handy-if-youre-careful>

B3603



working keys: <https://github.com/nlitsme/b3603/network>

<https://hackaday.io/project/4362-power-supply-b3603-alternative-firmware>

MingHe B3603 DC-DC converter pinout

Out: I(out) sense
 $\sim 0.97\text{V/A} + 0.14\text{V}$

Out: V(out) sense
 $72\text{mV/V} + 42\text{mV}$

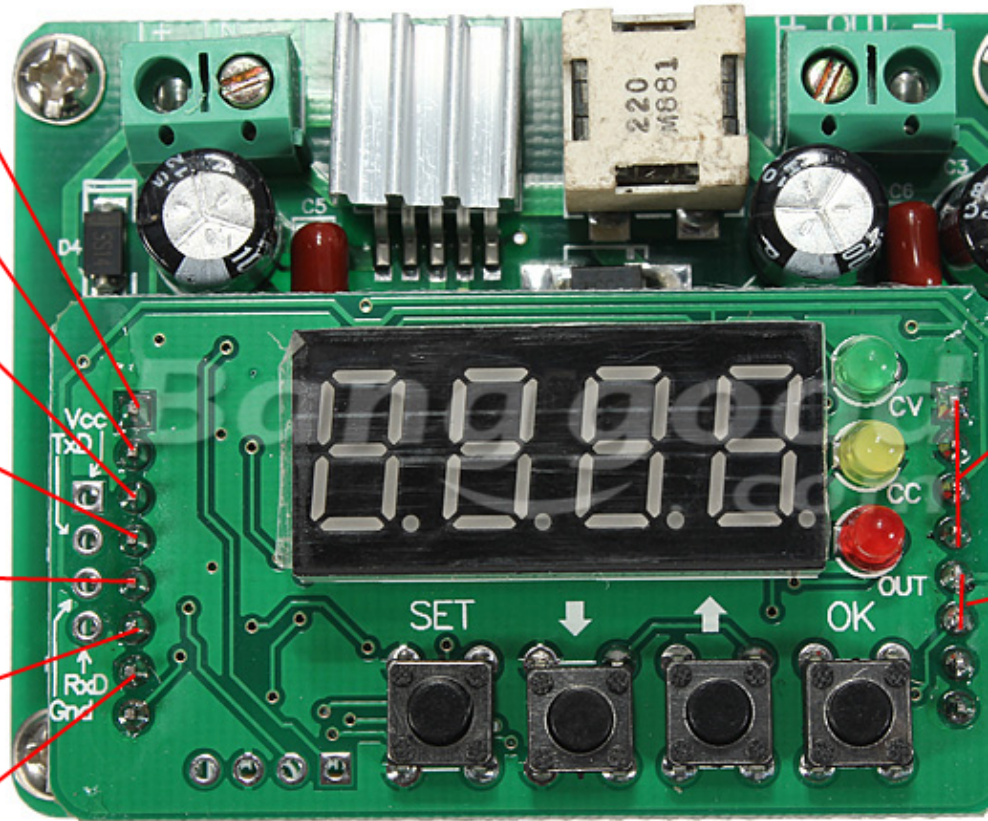
Out: V(in) sense
 62mV/V

In: I(out) set
 $\sim 0.97\text{V/A} + 0.14\text{V}$

In: V(out) set
 $72\text{mV/V} + 42\text{mV}$

In: Enable
0V = output on
5V = output off

Out: CC/CV status
CV = $\sim 0.47\text{V}$
CC = $\sim 2.5\text{V}$



- <http://www.eevblog.com/forum/reviews/b3603-dcdc-buck-converter-mini-review-and-how-the-set-key>
- <https://github.com/baruch/b3603>
- <http://www.banggood.com/DC-LED-Digital-Controlled-Step-Down-Driver-Power-Module-p-910096.ht>

Specification

Input voltage: 6V ~ 40V
Output voltage: 0V ~ 36V
Output current: 0A ~ 3A
Conversion efficiency: up to 92%
The output ripple: $\hat{a} \approx 50\text{mV}$
Working temperature: $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$
Working frequency: 150KHz
Short circuit protection: constant current
Voltage regulation/display resolution: 0.01V
Current regulation/display resolution: 0.001A
The minimum display resolution of power: 0.001W
The minimum display resolution of capacity: 0.001AH
Input reverse connect protection: no, if necessary please install diode
Connection mode: terminals
Size: 66 x 50 x 21mm
Weight: 44g

backup existing flash

```
pi@rpi2 ~/stm8flash $ sudo ./stm8flash -c stlinkv2 -p stm8s003f3 -s eeprom -r b3606/eeprom.bin
action = 1Determine EEPROM area
Reading 128 bytes at 0x4000... OK
Bytes received: 128
pi@rpi2 ~/stm8flash $ sudo ./stm8flash -c stlinkv2 -p stm8s003f3 -s flash -r b3606/flash.bin
action = 1Determine FLASH area
Reading 8192 bytes at 0x8000... OK
Bytes received: 8192
pi@rpi2 ~/stm8flash $ sudo ./stm8flash -c stlinkv2 -p stm8s003f3 -s ram -r b3606/ram.bin
action = 1Determine RAM area
Reading 1024 bytes at 0x0... OK
Bytes received: 1024
```

unprotect stm8 using stm8flash

<https://github.com/vdudouyt/stm8flash/issues/38#issuecomment-195123081>

```
$ echo "00" | xxd -r -p >ROP_CLEAR.bin
$ stm8flash -c stlinkv2 -p stm8s103f3 -s opt -w ROP_CLEAR.bin
```

compile

```
pi@rpi2 ~ $ git clone https://github.com/swegener/b3603
Cloning into 'b3603'...
remote: Counting objects: 1109, done.
remote: Total 1109 (delta 0), reused 0 (delta 0), pack-reus
Receiving objects: 100% (1109/1109), 11.04 MiB | 2.90 MiB/s
Resolving deltas: 100% (770/770), done.
Checking connectivity... done.
```

```
pi@rpi2 ~/b3603/stm8 $ sudo apt-get install sdcc
```

```
pi@rpi2 ~/b3603/stm8 $ make
```

```
Code fits the flash, it is 7930
```

```
dpavlin@nuc:/nuc/b3603/stm8$ uname -a
```

```
Linux nuc 4.4.0-1-amd64 #1 SMP Debian 4.4.6-1 (2016-03-17) x86_64 GNU/Linux
```

```
dpavlin@nuc:/nuc/b3603/stm8$ sdcc -v
```

```
SDCC : mcs51/z80/z180/r2k/r3ka/gbz80/tlcs90/ds390/TININative/ds400/hc08/s08/stm8 3.5.0 #9253 (Mar
```

```
published under GNU General Public License (GPL)
```

```
dpavlin@nuc:/nuc/b3603/stm8$ scp b3603.ihx rpi2:stm8flash/
```

flash

```
pi@rpi2 ~/stm8flash $ sudo ./stm8flash -c stlinkv2 -p stm8s003f3 -w ./b3603.ihx
```

```
action = 2Determine FLASH area
```

```
Writing Intel hex file 8160 bytes at 0x8000... Tries exceeded
```

```
# unprotect flash
```

```
pi@rpi2 ~/stm8flash $ echo "00" | xxd -r -p >ROP_CLEAR.bin
```

```
pi@rpi2 ~/stm8flash $ sudo ./stm8flash -c stlinkv2 -p stm8s003f3 -s opt -w ROP_CLEAR.bin
```

```
action = 2Determine OPT area
```

```
Writing binary file 1 bytes at 0x4800... OK
```

```
Bytes written: 1
```

```
#
```

serial

```
dpavlin@x200:/x200/stm8$ microcom -p /dev/ttyUSB0 -s 38400
```

```
connected to /dev/ttyUSB0
```

```
Escape character: Ctrl-\
```

```
Type the escape character followed by c to get to the menu or q to quit
```

```
B3603 starting: Version 1.0.1
```

```
pi@rpi2 ~/b3603/stm8 $ cat ~/b3603.sh
```

```
#!/bin/sh -x
```

```
microcom -s 38400 -p /dev/serial/by-path/platform-3f980000.usb-usb-0\:1.5.1\:1.0-port0
```

calibrate

```
dpavlin@cubieboard:/mnt/nuc/b3603/stm8$ git diff ./calibrate.py
diff --git a/stm8/calibrate.py b/stm8/calibrate.py
index f94f866..313e1f6 100755
--- a/stm8/calibrate.py
+++ b/stm8/calibrate.py
@@ -153,7 +153,8 @@ class Multimeter(object):
     return self._sample() != None

     def _sample(self):
-        p = os.popen('sigrok-cli -d %s:conn=%s --samples 1' % (self.model, self.portname))
+        print('## sigrok-cli -d %s:conn=%s --samples 1' % (self.model, self.portname))
+        p = os.popen('sigrok-cli -d %s:conn=%s --samples 1 -O analog' % (self.model, self.portname))
         s = p.read()
         p.close()
         return float(s.split(' ')[1])
@@ -214,9 +215,9 @@ def calibration_voltage(auto):
     return

     if auto == True:
-        dmm = Multimeter(sys.argv[3], sys.argv[4])
+        dmm = Multimeter(sys.argv[4], sys.argv[5])
         if not dmm.open():
-            print 'Failed to open serial port to multimeter on serial %s model %s' % (sys.argv[3], sys.argv[4])
+            print 'Failed to open serial port to multimeter on serial %s model %s' % (sys.argv[4], sys.argv[5])
             psu.close()
             return
```

```
dpavlin@cubieboard:/mnt/nuc/b3603/stm8$ ./calibrate.py -a voltage /dev/ttyUSB0 /dev/ttyUSB1 uni-t
er
OPEN "M: B3603"
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
PSU Input voltage is 11305.0 mV, will use 10 steps between 10 mV and 4000 mV
0 . Setting voltage to 10 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Multimeter samples vary too much, stddev=0.244349, data: [1.2358, 0.8785, 0.6413]
Failed to read stable value, trying again, maybe
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 0 Set voltage 10.000000 mV Read voltage 0.273667 mV PWM 114.0 ADC 132.0 (0.149)
1 . Setting voltage to 409 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 1 Set voltage 409.000000 mV Read voltage 0.403467 mV PWM 185.0 ADC 172.0 (0.377)
2 . Setting voltage to 808 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 2 Set voltage 808.000000 mV Read voltage 0.807800 mV PWM 257.0 ADC 244.0 (0.772)
3 . Setting voltage to 1207 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 3 Set voltage 1207.000000 mV Read voltage 1.212667 mV PWM 329.0 ADC 316.0 (1.179)
4 . Setting voltage to 1606 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
```

```
Step 4 Set voltage 1606.000000 mV Read voltage 1.617200 mV PWM 401.0 ADC 387.0 (1.574)
5 . Setting voltage to 2005 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 5 Set voltage 2005.000000 mV Read voltage 2.022333 mV PWM 473.0 ADC 459.0 (1.975)
6 . Setting voltage to 2404 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 6 Set voltage 2404.000000 mV Read voltage 2.425333 mV PWM 545.0 ADC 531.0 (2.376)
7 . Setting voltage to 2803 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 7 Set voltage 2803.000000 mV Read voltage 2.829333 mV PWM 617.0 ADC 603.0 (2.782)
8 . Setting voltage to 3202 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 8 Set voltage 3202.000000 mV Read voltage 3.232000 mV PWM 689.0 ADC 675.0 (3.178)
9 . Setting voltage to 3601 mV
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
## sigrok-cli -d uni-t-ut61e-ser:conn=/dev/ttyUSB1 --samples 1
Step 9 Set voltage 3601.000000 mV Read voltage 3.635667 mV PWM 761.0 ADC 747.0 (3.579)
['OK', '']
ADC
(0.005373500905021045, -0.8923354860819778) 352 58480
['OK', '']
PWM
(179.88709677419354, 185.25806451612908) 11789080 12141072
['OK', '']
```