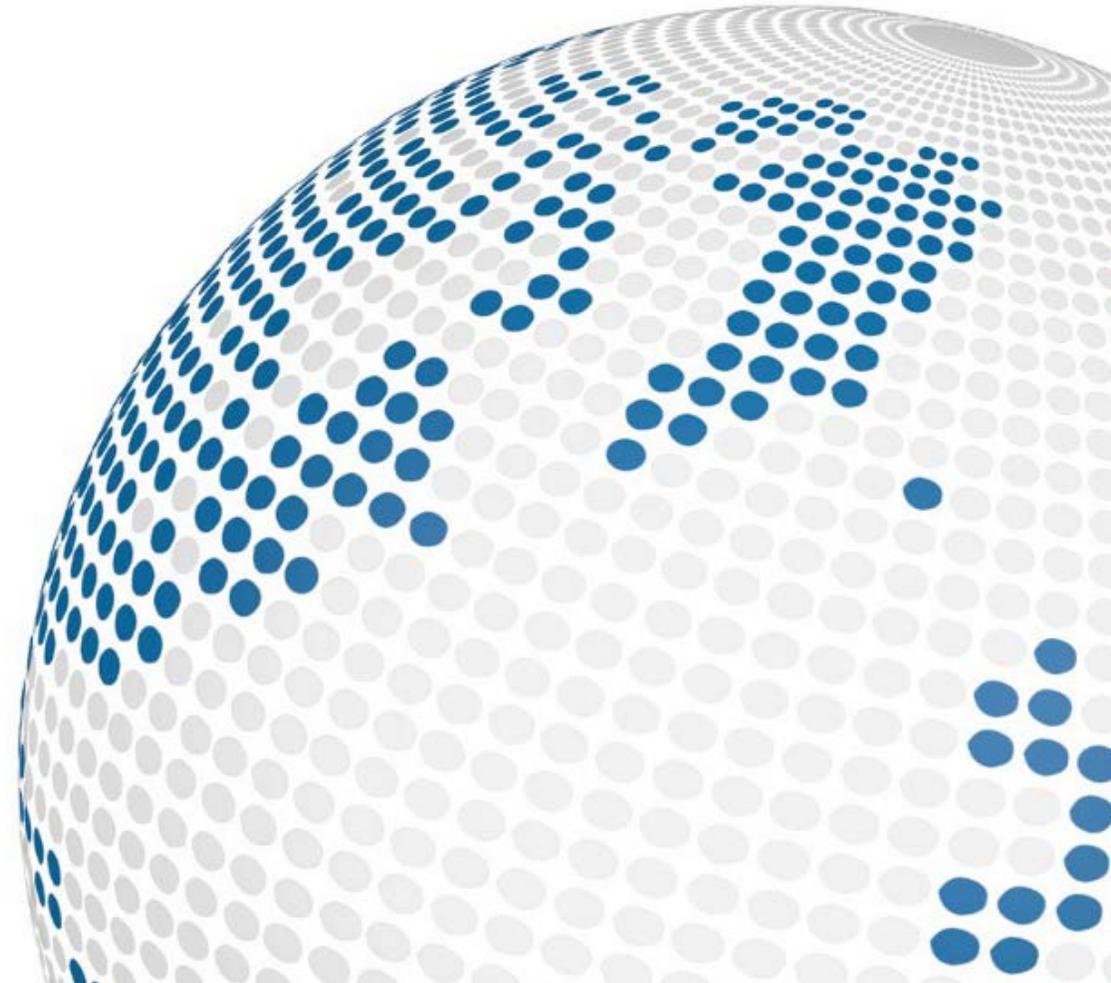


ams

Shaping the world with sensor solutions

Compile and Execute TMF8701 Bare Metal (=MCU)
driver on Raspberry PI

Peter Trattler



Compile and Execute Bare Metal (=MCU) driver on PI

This documentation describes howto compile and execute the bare metal (=MCU) driver on Raspberry PI with and without embedding the firmware inside the driver.

Prerequisites

- Need ssh tool on PC (e.g. Cygwin, Ubuntu or PuTTY)
- Need file transfer tool over ssh (scp) (e.g. Cygwin or WinSCP)

- Connect TMF8X01 EVM to PC
 - Will generate a drive on PC (typically USB Drive E)
 - Start E:\drivers\win\ams_raspirndis_setup.exe to install driver
 - Copy aos_tmf8701_mcu_driver_src_2.0.33.0_1.1.zip to Raspberry by scp (or WinSCP) /home/pi directory

Note: If aos_tmf8701_mcu_driver_**bin**_2.0.33.0_1.1.zip is used instead (pre-compiled), unzip driver (next slide) and then skip all pages up to “Execute Driver”

Setup Raspberry PI - 1

```
ssh pi@169.254.0.2
# password is raspberry
# now all commands below are within Raspberry PI shell – pi@raspberrypi:~$

# extract zip file of driver
unzip aos_tmf8701_mcu_driver_src_2.0.33.0_1.1.zip
# connect Raspberry PI to internet
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
# add WLAN network in nano – replace with actual network information
network={
    ssid="testing"
    psk="testingPassword"
}
# connect
wpa_cli -i wlan0 reconfigure
```

Setup Raspberry PI - 2

```
# check internet connection
```

```
ping ams.com
```

```
# output : 64 bytes from www.ams.com (212.166.112.3): icmp_seq=1 ttl=254 time=20.0 ms
```

```
# update apt database
```

```
sudo apt-get update
```

```
# install cmake
```

```
sudo apt-get install cmake
```

```
# prepare driver
```

```
cd aos_tmf8701_mcu_driver_src_2.0.33.0_1.1/
```

```
# edit nano toolchain_raspberrypi3.cmake and comment out following lines – add '#'
```

```
nano toolchain_raspberrypi3.cmake
```

```
#SET(toolchain $ENV{RPI_TOOLCHAIN}/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/)
```

```
#SET(CMAKE_C_COMPILER ${toolchain}/bin/arm-linux-gnueabihf-gcc)
```

```
#SET(CMAKE_FIND_ROOT_PATH ${toolchain}/arm-linux-gnueabihf/)
```

Cmake & Compile – Firmware not Embedded

```
# use these command if the firmware is provided on the command line
mkdir build
cd build
cmake -DCMAKE_TOOLCHAIN_FILE=../toolchain_raspberrypi3.cmake ..
# compile
make
```

Cmake & Compile – Firmware Embedded



```
# use these command if the firmware is embedded in the driver
mkdir build
cd build
# install srecord tool
sudo apt-get install srecord
cmake -DFW_PATH=../main_app_3v3_k2.hex -DCMAKE_TOOLCHAIN_FILE=../toolchain_raspberrypi3.cmake ..
# compile
make
```

Execute driver 1/2

Following needs to be done only once if the EVM was installed previously

1. comment out the tof device tree from /boot/config.txt
sudo nano /boot/config.txt
 - a. dtoverlay=tof8701-overlay --> #dtoverlay=tof8701-overlay
2. remove the tof kernel module from the software framework
sudo rm /opt/USBSensorBridgeRuntime/modules/tmf8?01.ko
3. add 'i2c-dev' to /etc/modules
sudo nano /etc/modules
 - a. Check for a line with "i2c-dev"
4. sync && reboot
sudo reboot now

Execute driver 2/2 – Firmware not embedded

Version with firmware provided on the command line

5. After the raspberry rebooted, log into the raspberry again
6. Change to the directory where the binary of bare metal linux driver is installed
7. Run `./mcu_tmf8701 -h` to display the help screen for running the bare metal driver

```
cd aos_tmf8701_mcu_driver_src_2.0.33.0_1.1/build/bin
```

8. Generate calibration file (no object in front of EVM within 40cm)

```
sudo ./mcu_tmf8701 -h
```

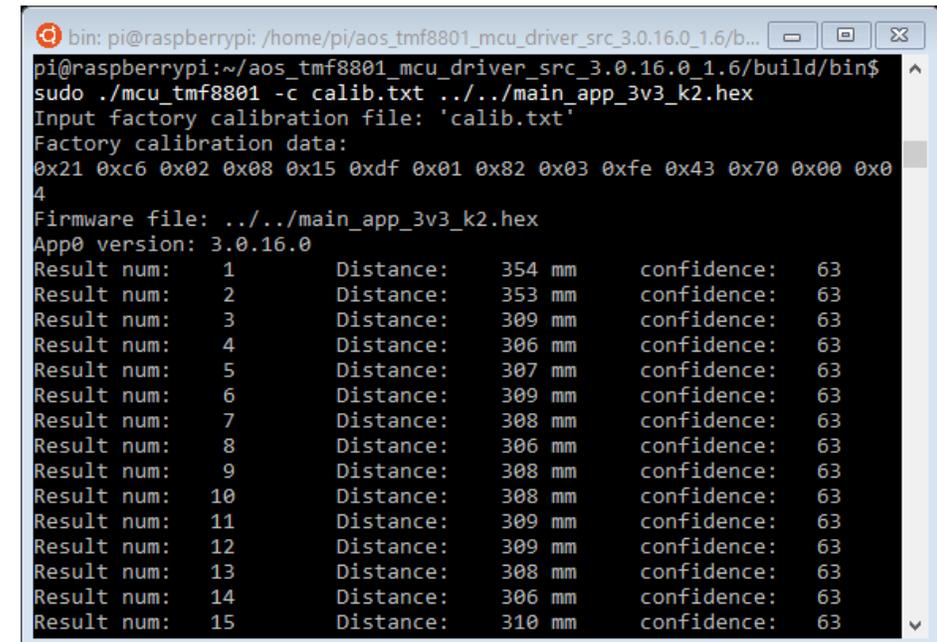
```
sudo ./mcu_tmf8701 -z calib.txt ../../main_app_3v3_k2.hex
```

9. Run measurement

```
sudo ./mcu_tmf8701 -c calib.txt ../../main_app_3v3_k2.hex
```

10. Run measurement for 100 times:

```
sudo ./mcu_tmf8701 -c calib.txt ../../main_app_3v3_k2.hex -n 100
```



```
bin: pi@raspberrypi: /home/pi/aos_tmf8801_mcu_driver_src_3.0.16.0_1.6/b...
pi@raspberrypi:~/aos_tmf8801_mcu_driver_src_3.0.16.0_1.6/build/bin$
sudo ./mcu_tmf8801 -c calib.txt ../../main_app_3v3_k2.hex
Input factory calibration file: 'calib.txt'
Factory calibration data:
0x21 0xc6 0x02 0x08 0x15 0xdf 0x01 0x82 0x03 0xfe 0x43 0x70 0x00 0x0
4
Firmware file: ../../main_app_3v3_k2.hex
App0 version: 3.0.16.0
Result num: 1      Distance: 354 mm      confidence: 63
Result num: 2      Distance: 353 mm      confidence: 63
Result num: 3      Distance: 309 mm      confidence: 63
Result num: 4      Distance: 306 mm      confidence: 63
Result num: 5      Distance: 307 mm      confidence: 63
Result num: 6      Distance: 309 mm      confidence: 63
Result num: 7      Distance: 308 mm      confidence: 63
Result num: 8      Distance: 306 mm      confidence: 63
Result num: 9      Distance: 308 mm      confidence: 63
Result num: 10     Distance: 308 mm      confidence: 63
Result num: 11     Distance: 309 mm      confidence: 63
Result num: 12     Distance: 309 mm      confidence: 63
Result num: 13     Distance: 308 mm      confidence: 63
Result num: 14     Distance: 306 mm      confidence: 63
Result num: 15     Distance: 310 mm      confidence: 63
```

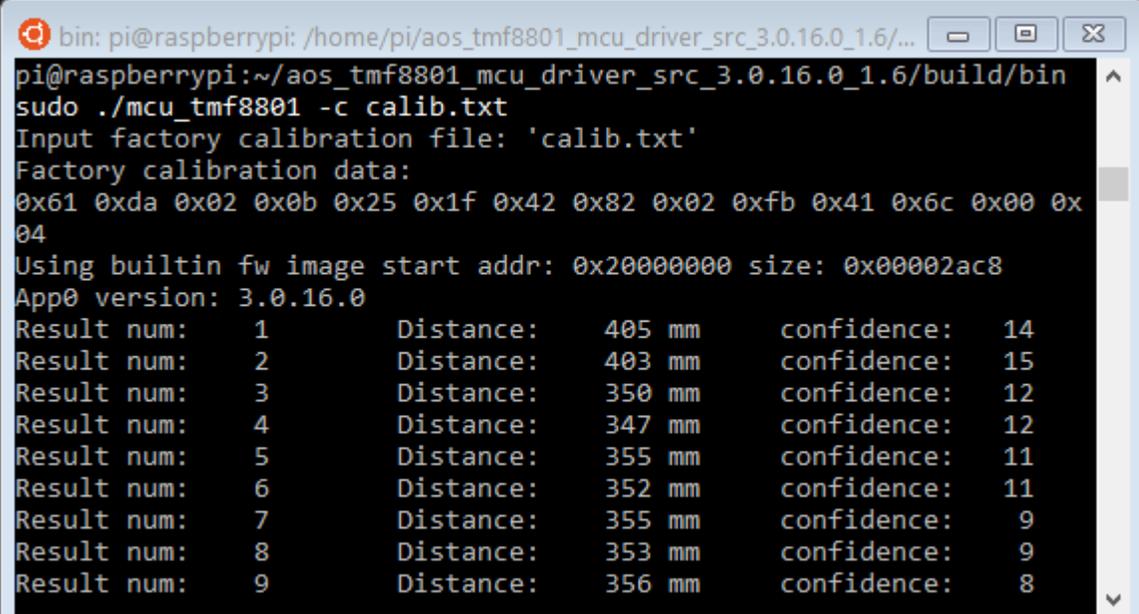
Execute driver 2/2 – Firmware embedded

Version with firmware embedded in the driver

5. After the raspberry rebooted, log into the raspberry again
6. Change to the directory where the binary of bare metal linux driver is installed
7. Run `./mcu_tmf8701 -h` to display the help screen for running the bare metal driver

```
cd aos_tmf8701_mcu_driver_src_2.0.33.0_1.1/build/bin
```

8. Generate calibration file
(no object in front of EVM within 40cm)
`sudo ./mcu_tmf8701 -z calib.txt`
9. Run measurement
`sudo ./mcu_tmf8701 -c calib.txt`
10. Run measurement for 100 times:
`sudo ./mcu_tmf8701 -c calib.txt -n 100`



```
bin: pi@raspberrypi: /home/pi/aos_tmf8801_mcu_driver_src_3.0.16.0_1.6/...
pi@raspberrypi:~/aos_tmf8801_mcu_driver_src_3.0.16.0_1.6/build/bin
sudo ./mcu_tmf8801 -c calib.txt
Input factory calibration file: 'calib.txt'
Factory calibration data:
0x61 0xda 0x02 0x0b 0x25 0x1f 0x42 0x82 0x02 0xfb 0x41 0x6c 0x00 0x
04
Using builtin fw image start addr: 0x20000000 size: 0x00002ac8
App0 version: 3.0.16.0
Result num: 1      Distance: 405 mm      confidence: 14
Result num: 2      Distance: 403 mm      confidence: 15
Result num: 3      Distance: 350 mm      confidence: 12
Result num: 4      Distance: 347 mm      confidence: 12
Result num: 5      Distance: 355 mm      confidence: 11
Result num: 6      Distance: 352 mm      confidence: 11
Result num: 7      Distance: 355 mm      confidence: 9
Result num: 8      Distance: 353 mm      confidence: 9
Result num: 9      Distance: 356 mm      confidence: 8
```

Restoring EVM Functionality

Bare metal driver stops TMF8701 EVM functionality

To restore EVM functionality:

```
ssh pi@169.254.0.2
```

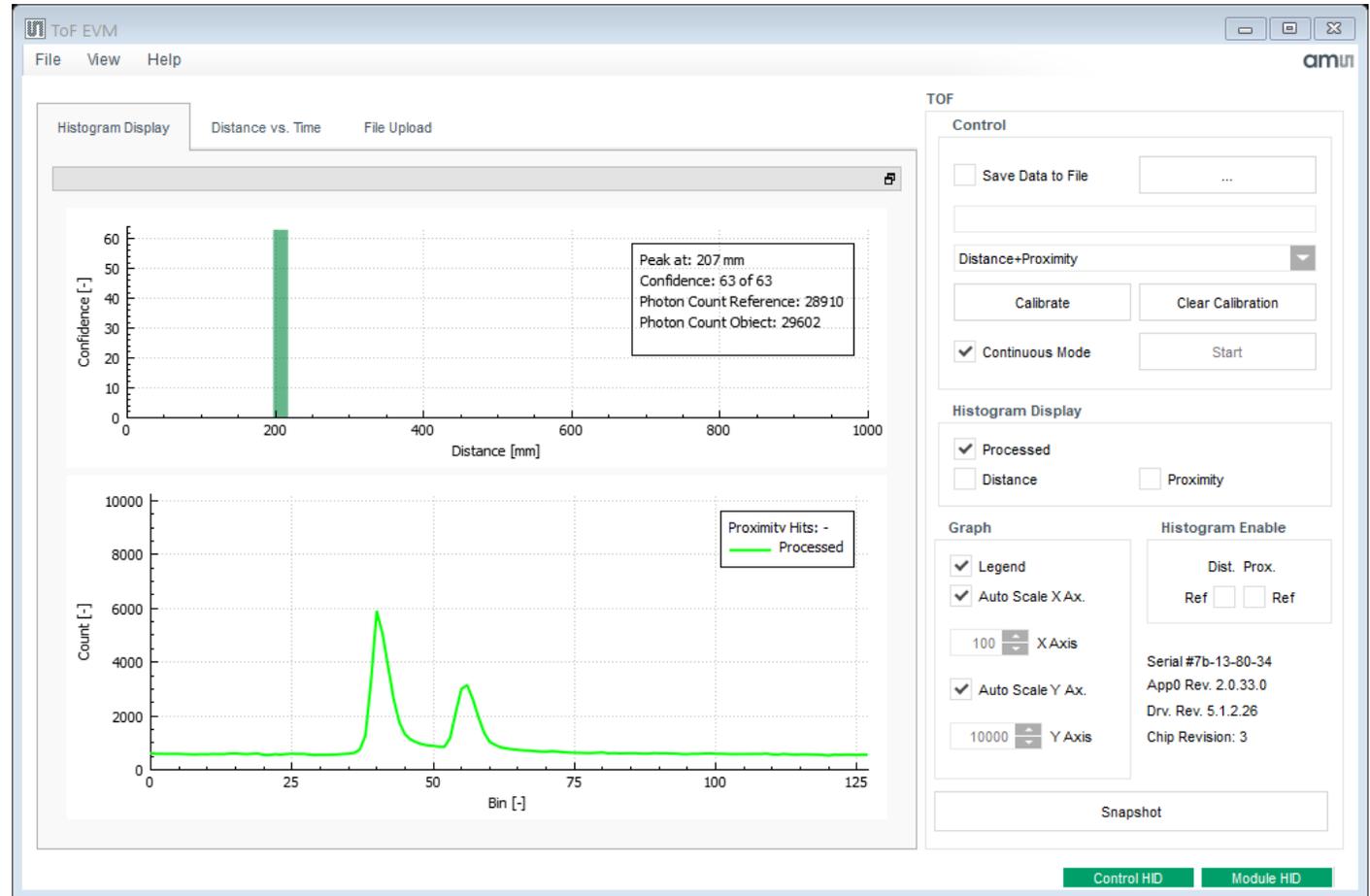
```
# password is raspberry
```

```
cd 2.0.33.0_5.1.2.26_tof_core1_evm_linux/  
sudo ./install_3v3.sh k2
```

```
# next cmd can fail if already done once  
sudo rm /usr/local/lib/libtof.so  
sudo reboot now
```

```
# start GUI on PC
```

Press 'Calibrate' (low IR light, no target within 40cm to device) to re-calibrate





Thank you!

Please visit our website
www.ams.com