BM1880 EDB Software
User Manual
## Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Author</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Liang.Wang02</td>
<td>2018.10.12</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>1.0</td>
<td>Liang.Wang02</td>
<td>2018.11.1</td>
<td>Update version to 1.0</td>
</tr>
</tbody>
</table>
# Catalog Index

REVISION HISTORY .................................................................................................................. 2  
OVERVIEW .................................................................................................................................. 4  
EDGE TPU DEVELOPER BOARD ............................................................................................ 5  
RESOURCES ON EDB .............................................................................................................. 5  
SPECIFICATIONS ...................................................................................................................... 6  
EXPANSION CONNECTOR ........................................................................................................... 7  
40Pin low speed expansion interface .......................................................................................... 7  

OPERATING SYSTEM INSTALLATION ...................................................................................... 7  
SOFTWARE RELEASE AND DOWNLOAD ............................................................................... 7  
BOOTUP MODES ......................................................................................................................... 8  
SD Boot .................................................................................................................................. 8  
USB Boot .................................................................................................................................. 9  
eMMC Boot ................................................................................................................................. 10  

SOFTWARE DEVELOPMENT ................................................................................................. 11  
GPIO CONTROL EXAMPLE ................................................................................................. 11  
UVC CAMERA .......................................................................................................................... 11
Overview

Product Overview
The Bitmain Sophon™ Edge Developer Board is designed for bringing powerful Deep Learning capability to various type of applications through its quick prototype development. Sophon™ Edge Developer Board is powered by BM1880, which equips tailored TPU supporting DNN/CNN/RNN/LSTM operations and models. The edge developer board is compatible with Linaro 96boards while supporting modules for Arduino and Raspberry Pi. Developers can leverage off the shelf modules and develop cutting edge DL/ML applications, like facial detection and recognition, facial expression analysis, object detection and recognition, vehicle license plate recognition, voiceprint recognition, etc.

Product Features
– Supports DNN/CNN/RNN/LSTM models profiling, compiling and tuning
– Real time inference in edge device
– Quickly deploy existing DNN/CNN/RNN/LSTM models or uniquely trained networks
– Features Bitmain Sophon™ BM1880 with energy efficient DNN/CNN/RNN/LSTM processing
– Compatible to 96Boards Consumer Edition Spec.
– Support Caffe, ONNX, Pytorch, Tensorflow framework
– Support ResNet50, Yolo V2, GoogleNet V1, MobileNet v1/v2, SSD300, Alexnet, VGG16...etc
Resources on EDB

- Resources On EDB

BM1880 SoC
USB Port
USB 2.0 Host/Service Mode/OTG
Micro USB
Debug for ARM Core
JTAG1
Power 5V/12V, 3A
Power Reset
Chip Reset
Ethernet
5GbE Ethernet
SD Card Slot
50 x 30, UHS-I 104, up to 187 Mbps
USB Port
Support USB 3.1 Host Mode
DRAM
LPDDR4, 16 GB
Wi-Fi (Backside)
IEEE 802.11 b/g/n/ac
Bluetooth (Backside)
BT 4.2/2.1 + EDR
JTAG1
Debug for PMC-V Core
Expansion Connector
48-Pin Low Speed Expansion Connector
GPI, SPI, I2C, UART and GPIO
<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Chip</td>
<td>BM1880</td>
</tr>
<tr>
<td>Processor</td>
<td>Dual Cortex <a href="mailto:A53@1.5Ghz">A53@1.5Ghz</a></td>
</tr>
<tr>
<td></td>
<td>RISC-V: 750Mhz</td>
</tr>
<tr>
<td>TPU</td>
<td>Up to 2TOPS by INT8 Winograd implementation</td>
</tr>
<tr>
<td>Memory</td>
<td>1GB LPDDR4 3200Mhz</td>
</tr>
<tr>
<td>Storage</td>
<td>8GB eMMC + micro SD card slot</td>
</tr>
<tr>
<td>Camera</td>
<td>Support USB Camera (UVC)</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Gigabit Ethernet Wifi/BT Combo</td>
</tr>
<tr>
<td>USB</td>
<td>USB 3.0 x 3 (1 with OTG support)</td>
</tr>
<tr>
<td>Expansion</td>
<td>40-pin low-speed expansion header</td>
</tr>
<tr>
<td>Audio</td>
<td>I2S x 2 support 4-Mic + speaker</td>
</tr>
<tr>
<td>Power supply</td>
<td>12V@2A</td>
</tr>
<tr>
<td>Debug</td>
<td>JTAG, UART</td>
</tr>
<tr>
<td>OS</td>
<td>Linux</td>
</tr>
<tr>
<td>Dimensions</td>
<td>85mm x 55mm</td>
</tr>
</tbody>
</table>
Expansion connector

40Pin low speed expansion interface

```
<table>
<thead>
<tr>
<th>Pin</th>
<th>Pin</th>
<th>Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>PWR_BTN_N</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>RST_BTN_N</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>SPI1_SCLK</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>SPI1_SDI</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>SPI1_CS</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>SPI1_SDO</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>I2S0_FS</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>I2S0_SCLK</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>I2S0_SDO</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>I2S0_SDI</td>
</tr>
<tr>
<td>23</td>
<td>24</td>
<td>GPIO62</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>GPIO64/I2S1_FS</td>
</tr>
<tr>
<td>27</td>
<td>28</td>
<td>GPIO63/I2S1_SCLK</td>
</tr>
<tr>
<td>29</td>
<td>30</td>
<td>GPIO66/I2S1_SDO</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>GPIO65/I2S1_SDI</td>
</tr>
<tr>
<td>33</td>
<td>34</td>
<td>GPIO67</td>
</tr>
<tr>
<td>35</td>
<td>36</td>
<td>+12V</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>+12V</td>
</tr>
<tr>
<td>39</td>
<td>40</td>
<td>GND</td>
</tr>
</tbody>
</table>
```

Building a complete development environment, you may also need to prepare these materials:
- 12V@2A or 5V@2A power adapter
- Micro USB cable
- Male to male USB cable
- Network cable

Operating system installation

Software Release and Download

To download the latest version of the SDK package, please visit the website [https://www.sophon.cn/](https://www.sophon.cn/)
Released package includes:

- fip.bin: A53 bootloader + uboot
- ramboot_mini.itb: Linux kernel image (on RAM/USB/TF-card with RAMdisk)
- sdboot.itb: Linux kernel image (on SD-card)
- emmcboot.itb: Linux kernel image (on eMMC)

**Bootup Modes**

EDB Board supports sd boot, usb boot, emmc boot. A variety of boot modes can be selected by SW1’s toggle switch:

- **BOOT_SEL[2:0] = 3'b100**: loading linux kernel from eMMC.
- **BOOT_SEL[2:0] = 3'b101**: loading linux kernel from SD card.
- **BOOT_SEL[2:0] = 3'b110**: loading linux kernel from USB.

**SD Boot**

It is recommended to use ubuntu 16.04 OS, format SD and make two partitions:

- IMAGES partition (FAT file system, 128MB).
- Rootfs partition (EXT4 file system, recommended 3GB or more)

You can refer to our sd_create_rootfs.sh (can you need to install expect : sudo apt-get install expect .) to create the partitions, enter the command:

```
$ sudo sd_create_rootfs.sh /dev/sdc
```

**Note:**

```
/dev/sdc is the sd card recognized by ubuntu Nodes may be different, please confirm with the fdisk -l command.
```

"vfat_part=${device}1" in the script, "ext4_part=${device}2" needs to be modified according to your actual pc identification, please note. After formatting successfully, you can see the following two partitions:

```
/dev/sdc1 on /media/bitmain/IMAGES type vfat
/dev/sdc2 on /media/bitmain/rootfs type ext4
```

- Copy the files needed by sdboot to the sd partition

```
tar –xvf soc_bm1880asic_edbsdboot.tar
cp soc_bm1682_asic_edb_sdboot/fip.bin /media/bitmain/IMAGES; sync
cp soc_bm1682_asic_edb_sdboot/sdboot.itb /media/bitmain/IMAGES; sync
sudo cp -fr install/soc_bm1682_asic_edb/rootfs/* /media/bitmain/rootfs/; sync
```
USB Boot

Need to install python2.7.x (https://www.python.org/downloads/).
Install pyserial (Windows OS)

```
python -m pip install --upgrade pip
python -m pip install pyserial
```

Settings as below:
- Toggle switch OTG_SEL (6) is selected to ground.
- Boot mode is selected to usb boot (BOOT_SEL[2:0] = 3'b110).
- USB male to bus is inserted into EDB P3 port:

After the power is turned on, you can see the following from the serial port log:

```
NOTICE: Load data from tfu...
NOTICE: Booting Trusted Firmware
NOTICE: BLI: v1.4(debug):bl1880 ROM v1
NOTICE: BLI: Built : 20:13:19, Jul 5 2018
Hit any key to stop autoboot: 0
NOTICE: BLI: fip_src 0
NOTICE: bm_usb_boot config 0
NOTICE: bm_usb_vid 0
NOTICE: bm_usb_hw_init done
NOTICE: by pass USB phy detection
NOTICE: fip_src 6
NOTICE: Application: disconnect
NOTICE: USB enumeration done
NOTICE: connection speed: 3
```

Open the “Device Manager” of Windows, you can see a "USB Serial Device (COM5)", this is the identified EDB board.

Burning program
Unzip edb_boot_from_usb.tar.gz and get the following file:

```
<table>
<thead>
<tr>
<th>File</th>
<th>Date</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>bm_dl_magic.bin</td>
<td>2018/10/8</td>
<td>BIN file</td>
<td>1 KB</td>
</tr>
<tr>
<td>bm1880_usb_boot.py</td>
<td>2018/10/8</td>
<td>Python File</td>
<td>9 KB</td>
</tr>
<tr>
<td>fip.bin</td>
<td>2018/10/10</td>
<td>BIN file</td>
<td>444 KB</td>
</tr>
<tr>
<td>prg.bin</td>
<td>2018/10/10</td>
<td>BIN file</td>
<td>54 KB</td>
</tr>
<tr>
<td>ramboot_mini.itb</td>
<td>2018/10/10</td>
<td>ITB file</td>
<td>11,673 KB</td>
</tr>
</tbody>
</table>
```
### eMMC Boot

Extract the file `edb_emmc_boot.tar.gz`. There are many ways to write EMMC. Only the method of programming from SD card and tftp is introduced here.

**SD card burning**
- Please make two FAT32 partitions on the SD card:
  - One 128M, put into `fip.bin`;
  - The other partition is larger than the size of the upgrade image `bm_update.img`, put `bm_update.img`

Start the board with the SD boot mode to the u-boot command line:
- copy `bm_update.img` to the sd card.
- connect the sd card to the board or the local tftp servers directory.

Start the board with the SD boot mode to the u-boot command line:
- copy `bm_update.img` to the sd card.
- connect the sd card to the board or the local tftp servers directory.

The command line executes the `bm_update` command.

```bash
tftp upgrade input the following command:
tftp 0x120000000 bm_update.img
bm_update 0x120000000
```
Software development

GPIO Control Example

The EDB platform uses the Linux GPIO operation interface, which is implemented by "/sys/class/gpio". Take GPIO0 as an example:
Echo 480 > /sys/class/gpio/export => GPIO0 needs to add an offset number of 480, and so on.

After the above command is successful, the gpio480 directory will be generated in the /sys/class/gpio directory and entered into the gpio480 directory as follows:

```
ls -l /sys/devices/platform/50027000.qpio/gpiochip0/gpio/gpio480
```

The direction file defines the input input direction, and the parameters accepted by direction: in, out.
The value file is the value of the port, which is 1 or 0.
For example, if you want to output 1 on GPIO0, you need the following operations:

```
Echo out > /sys/class/gpio/gpio480/direction
Echo 1 > /sys/class/gpio/gpio480/value
```

UVC Camera

The EDB Linux system has turned on the support of UVC Camera by default, inserting the UVC Camera into the USB port, and if the platform recognizes it as /dev/video0, it means success.