比特大陆 BITMAIN | 算丰 SOPHON

BM1880 EDB Software
User Manual

Revision History

Revision Number	Author	Date	Description
0.1	Liang.Wang02	2018.10.12	Initial Draft
1.0	Liang.Wang02	2018.11.1	Update version to 1.0

Catalog Index

Overview

Product Overview

The Bitmain SophonTM Edge Developer Board is designed for bringing powerful Deep Learning capability to various type of applications through its quick prototype development. SophonTM 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.

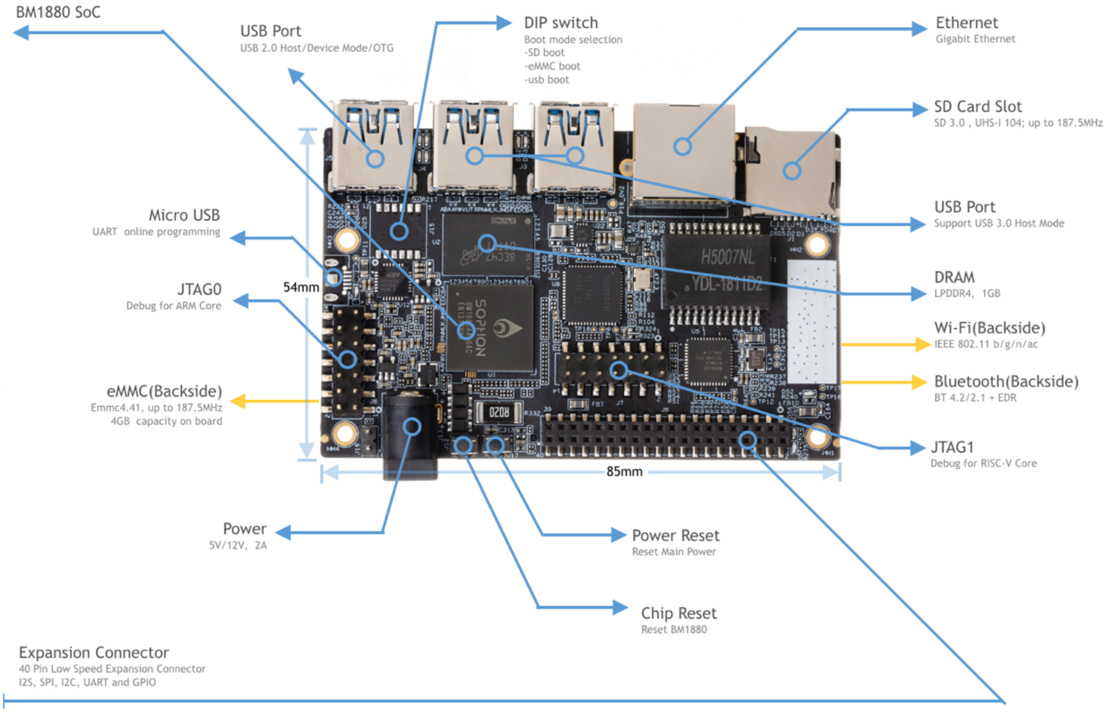
Product Features

– 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 SophonTM BM1880 with energy e和cient 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

Edge TPU Developer Board

Resources on EDB

- Resources On EDB



BM1880 SoC

USB Port
USB 2.0 Host/Device Mode/OTG

DIP switch
Boot mode selection
-SD boot
-eMMC boot
-usb boot

Ethernet
Gigabit Ethernet

SD Card Slot
SD 3.0 , UHS-I 104; up to 187.5MHz

Micro USB
UART online programming

JTAG0
Debug for ARM Core

54mm

USB Port
Support USB 3.0 Host Mode

DRAM
LPDDR4, 1GB

Wi-Fi(Backside)
IEEE 802.11 b/g/n/ac

Bluetooth(Backside)
BT 4.2/2.1 + EDR

eMMC(Backside)
Emmc4.41, up to 187.5MHz
4GB capacity on board

JTAG1
Debug for RISC-V Core

85mm

Power
5V/12V, 2A

Power Reset
Reset Main Power

Chip Reset
Reset BM1880

Expansion Connector
40 Pin Low Speed Expansion Connector
I2S, SPI, I2C, UART and GPIO

Specifications

Main Chip	BM1880
Processor	Dual Cortex A53@1.5Ghz RISC-V: 750Mhz
TPU	Up to 2TOPS by INT8 Winograd implementation
Memory	1GB LPDDR4 3200Mhz
Storage	8GB eMMC + micro SD card slot
Camera	Support USB Camera (UVC)
Connectivity	Gigabit Ethernet Wifi/BT Combo
USB	USB 3.0 x 3 (1 with OTG support)
Expansion	40-pin low-speed expansion header
Audio	I2S x 2 support 4-Mic + speaker
Power supply	12V@2A
Debug	JTAG, UART
OS	Linux
Dimensions	85mm x 55mm

Expansion connector

40Pin low speed expansion interface

GND	Pin 1	Pin 2	GND
UART0_CTS	Pin 3	Pin 4	PWR_BTN_N
UART0_TxD	Pin 5	Pin 6	RST_BTN_N
UART0_RxD	Pin 7	Pin 8	SPI1_SCLK
UART0_RTS	Pin 9	Pin 10	SPI1_SDI
UART1_TxD	Pin 11	Pin 12	SPI1_CS
UART1_RxD	Pin 13	Pin 14	SPI1_SDO
I2C0_SCL	Pin 15	Pin 16	I2S0_FS
I2C0_SDA	Pin 17	Pin 18	I2S0_SCLK
I2C1_SCL	Pin 19	Pin 20	I2S0_SDO
I2C1_SDA	Pin 21	Pin 22	I2S0_SDI
GPIO0	Pin 23	Pin 24	GPIO62
GPIO1	Pin 25	Pin 26	GPIO64/I2S1_FS
GPIO3	Pin 27	Pin 28	GPIO63/I2S1_SCLK
GPIO7	Pin 29	Pin 30	GPIO66/I2S1_SDO
GPIO50	Pin 31	Pin 32	GPIO65/I2S1_SDI
GPIO51	Pin 33	Pin 34	GPIO67
+1V8	Pin 35	Pin 36	+12V
+5V	Pin 37	Pin 38	+12V
GND	Pin 39	Pin 40	GND

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/

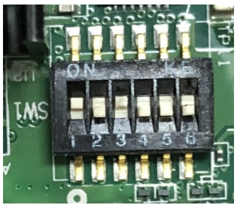
Released package includes:

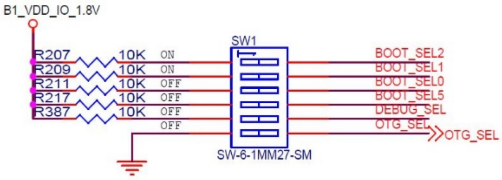
- fip.bin: A53 bootloader + uboot
- ramboot_mini.itb: Linux kernel image (on RAM/USB/TF-card with RAMdisk)
- sdboot.itb: Linux kernel image (on TF-card)
- emmcboot.itb: Linux kernel image (on eMMC)

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_bm1880_asic_edb_sdboot.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
```

- Select boot mode as sd boot (BOOT_SEL[2:0] = 3'b101).
- Insert sd card to start Linux system.

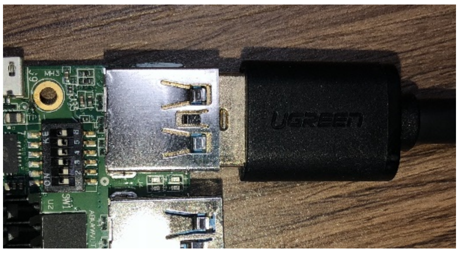
USB Boot

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

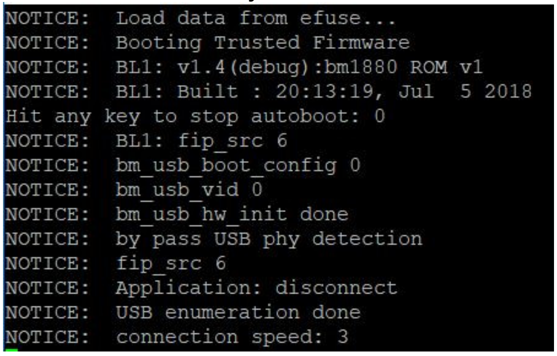
```
python -m pip install --upgrade pip
python -m pip install pyserial
```

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:

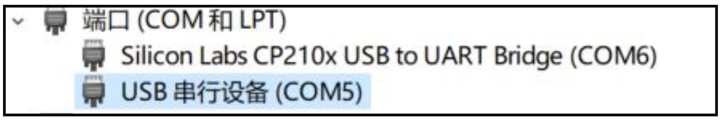


After the power is turned on, you can see the following from the serial port log:



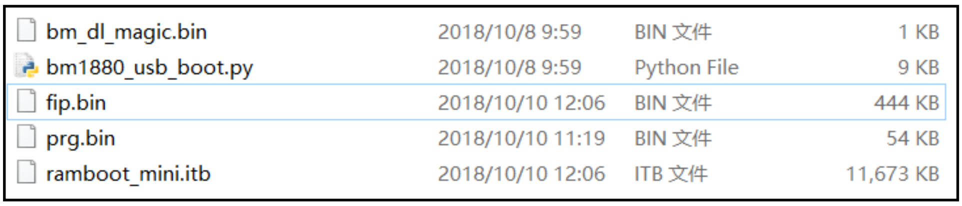
```
NOTICE:  Load data from efuse...
NOTICE:  Booting Trusted Firmware
NOTICE:  BL1: v1.4(debug):bm1880 ROM v1
NOTICE:  BL1: Built : 20:13:19, Jul  5 2018
Hit any key to stop autoboot: 0
NOTICE:  BL1: fip_src 6
NOTICE:  bm_usb_boot_config 0
NOTICE:  bm_usb_vid 0
NOTICE:  bm_usb_hw_init done
NOTICE:  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

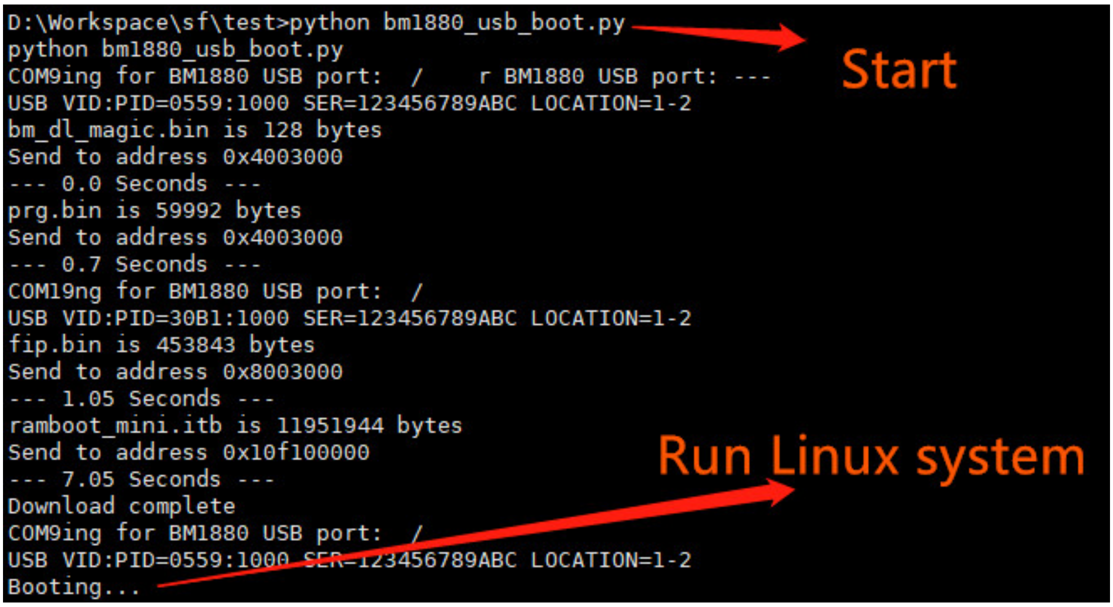


端口 (COM 和 LPT)
Silicon Labs CP210x USB to UART Bridge (COM6)
USB 串行设备 (COM5)

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



bm_dl_magic.bin	2018/10/8 9:59	BIN 文件	1 KB
bm1880_usb_boot.py	2018/10/8 9:59	Python File	9 KB
fip.bin	2018/10/10 12:06	BIN 文件	444 KB
prg.bin	2018/10/10 11:19	BIN 文件	54 KB
ramboot_mini.itb	2018/10/10 12:06	ITB 文件	11,673 KB



```
D:\Workspace\sf\test>python bm1880_usb_boot.py
python bm1880_usb_boot.py
COM9ing for BM1880 USB port:  /    r BM1880 USB port: ---
USB VID:PID=0559:1000 SER=123456789ABC LOCATION=1-2
bm_dl_magic.bin is 128 bytes
Send to address 0x4003000
--- 0.0 Seconds ---
prg.bin is 59992 bytes
Send to address 0x4003000
--- 0.7 Seconds ---
COM19ng for BM1880 USB port:  /
USB VID:PID=30B1:1000 SER=123456789ABC LOCATION=1-2
fip.bin is 453843 bytes
Send to address 0x8003000
--- 1.05 Seconds ---
ramboot_mini.itb is 11951944 bytes
Send to address 0x10f100000
--- 7.05 Seconds ---
Download complete
COM9ing for BM1880 USB port:  /
USB VID:PID=0559:1000 SER=123456789ABC LOCATION=1-2
Booting...
```

eMMC Boot

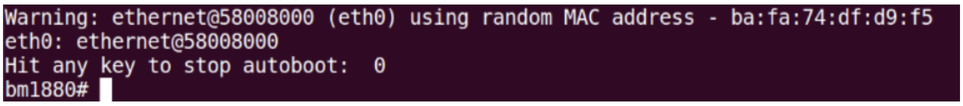
Extract the file edb_emmc_boot.tar.gz.
There are many ways to write EMMC. Only the method of programming from SD card and tftp is introduced here.

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.



```
Warning: ethernet@58008000 (eth0) using random MAC address - ba:fa:74:df:d9:f5
eth0: ethernet@58008000
Hit any key to stop autoboot:  0
bm1880#
```

The command line executes the bm_update command.

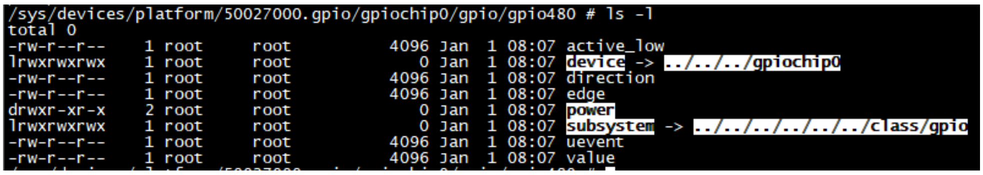
```
tftp upgrade input the following command:
tftp 0x120000000 bm_update.img
bm_update 0x120000000
```

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:



```
/sys/devices/platform/50027000.gpio/gpiochip0/gpio/gpio480 # ls -l
total 0
-rw-r--r--    1 root     root          4096 Jan  1 08:07 active_low
lrwxrwxrwx    1 root     root             0 Jan  1 08:07 device -> ../../../gpiochip0
-rw-r--r--    1 root     root          4096 Jan  1 08:07 direction
-rw-r--r--    1 root     root          4096 Jan  1 08:07 edge
drwxr-xr-x    2 root     root             0 Jan  1 08:07 power
lrwxrwxrwx    1 root     root             0 Jan  1 08:07 subsystem -> ../../../../../../class/gpio
-rw-r--r--    1 root     root          4096 Jan  1 08:07 uevent
-rw-r--r--    1 root     root          4096 Jan  1 08:07 value
```

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.