

MT SDK User Manual

- V1.4

<Contents>

1. System Hardware and Configuration
 - 1.1 Hardware Specifications
 - 1.2 Hardware Interface
 - 1.2.1 3 UART
 - 1.2.2 Enable WiFi Module
 - 1.2.3 1 Ethernet port eth0
 - 1.2.4 1 WiFi port
 - 1.2.5 1 3G/4G port(optional)
2. SDK User Guide
 - 2.1 SDK Installation Path and File
 - 2.2 SDK Access Control
3. Software Development
 - 3.1 Develop in C language
 - 3.1.1 Compile with gcc
 - 3.1.2 Compile with Makefile
 - 3.2 Develop with PHP
 - 3.3 Develop with Python
 - 3.4 Develop with Java
 - 3.5 Apache Instructions
 - 3.6 crond Instructions
4. Drive Related to Hardware
 - 4.1 GPIO Drive
 - 4.2 GPIO IO Control Command
 - 4.3 GPIO Pin Numbers
 - 4.3.1 HJ8X00 IO Pin Instructions
5. Software and Library Installation
 - 5.1 Set System Time
 - 5.2 Python Setuptools Installation
 - 5.3 Python pip Installation
6. Issue Software and Self-start Software
 - 6.1 Realize Self-starting by Start-up File

1. System Hardware and Configuration

1.1 Hardware Parameters

| | |
|---------------------|---|
| Master Chip | MIPS24KEC |
| Dominant Frequency | 580Mhz |
| Flash | 32MB |
| RAM | 128MB DDR2 |
| WiFi | 802.11n, 2T2R 300M MIMO technology |
| Power Supply | POE 24V, DC 9~24V |
| WiFi RF Power | 18dBm ± 2dBm |
| WiFi RF Band | Support 5,10,20,HT40MHz |
| Receive Sensitivity | -62dBm@150Mbps, -65dBm@128Mbps,-69dBm@78Mbps, |
| Software Upgrade | Firmware and software all can be upgraded |
| Work Temperature | -20°C to 80°C |
| Work Humidity | 10% to 90%RH non-condensing |
| Storage Humidity | 5% to 90%RH |

1.2 Hardware Interface Instructions

1.2.1 3 UART /dev/ttyS0 /dev/ttyS1 /dev/ttyS2. /dev/ttyS0 is only used for system but cannot be used in other ways

1.2.2 1 Ethernet port eth0

```
eth0.2  Link encap:Ethernet  HWaddr C2:48:00:01:25:86
        inet addr:192.168.33.100  Bcast:192.168.33.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:12064 errors:0 dropped:2692 overruns:0 frame:0
        TX packets:4190 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:1539313 (1.4 MiB)  TX bytes:616277 (601.8 KiB)
```

The MAC address in Ethernet port is factory settings. Each device has unique MAC address.

1.2.3 1 WiFi port

```
wlan0  Link encap:Ethernet  HWaddr 00:19:82:00:47:86
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1755408 (1.6 MiB)  TX bytes:0 (0.0 B)
        Interrupt:4
```

MAC address in WiFi is factory settings. Each device has unique MAC address.

1.2.4 Multiple IO interfaces which can be accessed through /dev/mtxdev device.

1.2.5 1x 3G/4G interface(optional)

1.2.6 1 MMC/SD interface(optional) which can be accessed through /dev/mmcblk0p1 device. System mount it in /mmz path by default.

1.2.7 1 433M/315M interface(optional) which can be accessed through /dev/mtsi device.

1.2.8 Enable WiFi module

Configure /etc/Wireless/RT2860/RT2860.dat and main parameters SSID1, Channel, AuthMode, EncrypType, WPAPSK1, etc

Enable AP by command line

```
rmmod mt_wifi_ap
```

```
insmod /lib/modules/mt_wifi_ap.ko
```

Enable STA by command line

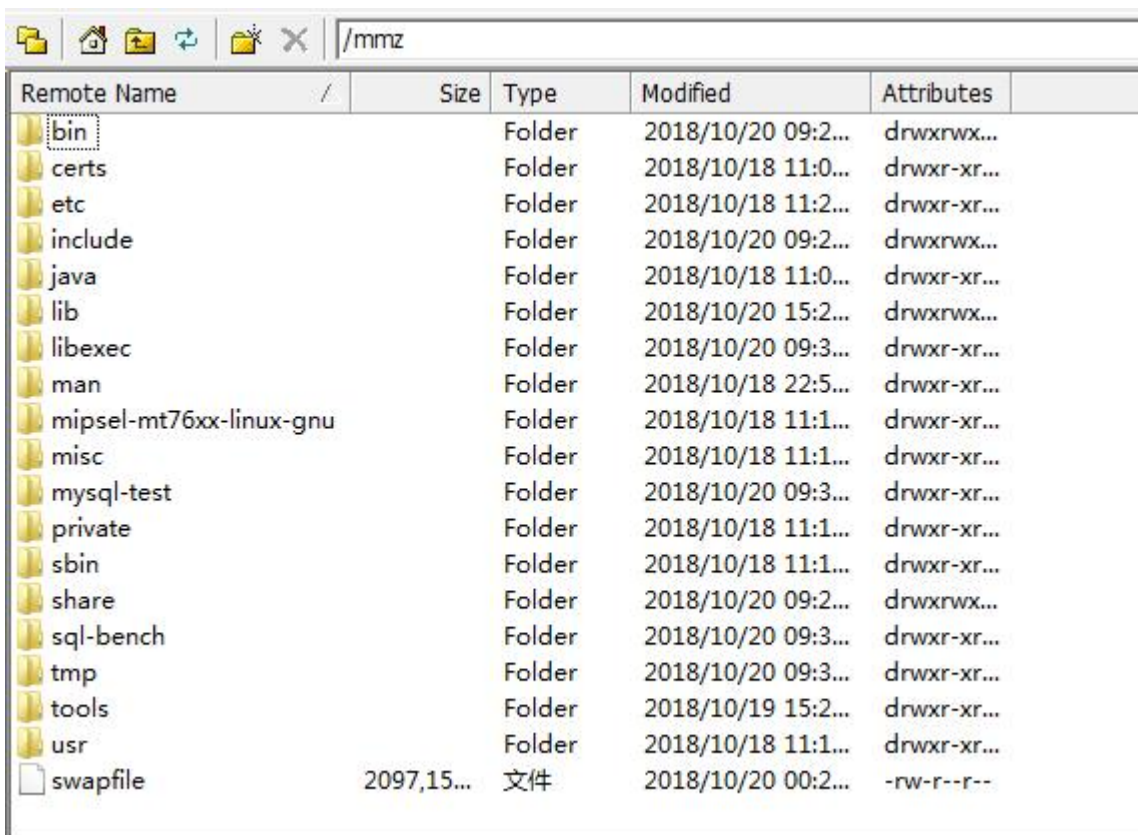
```
rmmod mt_wifi_sta
```

```
insmod /lib/modules/mt_wifi_sta.ko
```

Enable ifconfig wlan0 up

2. SDK User Guide

2.1 SDK Installation Patch and File



| Remote Name | Size | Type | Modified | Attributes |
|-------------------------|------------|--------|--------------------|-------------|
| bin | | Folder | 2018/10/20 09:2... | drwxrwx... |
| certs | | Folder | 2018/10/18 11:0... | drwxr-xr... |
| etc | | Folder | 2018/10/18 11:2... | drwxr-xr... |
| include | | Folder | 2018/10/20 09:2... | drwxrwx... |
| java | | Folder | 2018/10/18 11:0... | drwxr-xr... |
| lib | | Folder | 2018/10/20 15:2... | drwxrwx... |
| libexec | | Folder | 2018/10/20 09:3... | drwxr-xr... |
| man | | Folder | 2018/10/18 22:5... | drwxr-xr... |
| mipsel-mt76xx-linux-gnu | | Folder | 2018/10/18 11:1... | drwxr-xr... |
| misc | | Folder | 2018/10/18 11:1... | drwxr-xr... |
| mysql-test | | Folder | 2018/10/20 09:3... | drwxr-xr... |
| private | | Folder | 2018/10/18 11:1... | drwxr-xr... |
| sbin | | Folder | 2018/10/18 11:1... | drwxr-xr... |
| share | | Folder | 2018/10/20 09:2... | drwxrwx... |
| sql-bench | | Folder | 2018/10/20 09:3... | drwxr-xr... |
| tmp | | Folder | 2018/10/20 09:3... | drwxr-xr... |
| tools | | Folder | 2018/10/19 15:2... | drwxr-xr... |
| usr | | Folder | 2018/10/18 11:1... | drwxr-xr... |
| swapfile | 2097,15... | 文件 | 2018/10/20 00:2... | -rw-r--r-- |

SDK is installed in MMC card and mounted in /mmz path. It includes below environments:

C/C++ compiler

Glibc library
Python development environment
Java compiling and operating environment
PHP development environment
MySQL database
SSH tool

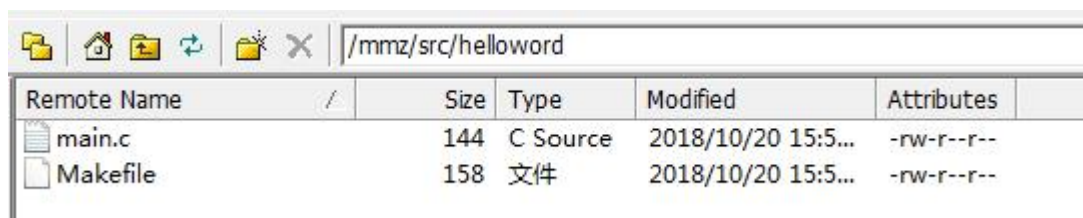
2.2 SDK Access Control

SDK can be accessed through SSH. The device can be used as a Linux server with MIPS processor.
Common SSH client SSHSecureShellClient include ssh terminal and sftp client two parts. It is easy for use.

3. Software Development

3.1 Develop in C language

Source file path



| Remote Name | Size | Type | Modified | Attributes |
|-------------|------|----------|--------------------|------------|
| main.c | 144 | C Source | 2018/10/20 15:5... | -rw-r--r-- |
| Makefile | 158 | 文件 | 2018/10/20 15:5... | -rw-r--r-- |

Source file content

```
[root@localhost helloworld]$ cat main.c
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{   printf("Hello Word\r\n");
    return 1;
}
```

3.1.1 Compile with gcc

```
[root@localhost helloworld]$ gcc main.c
[root@localhost helloworld]$ ls
Makefile  a.out    main.c
```

Output a.out

```
[root@localhost helloworld]$ ./a.out
Hello Word
```

3.1.2 Compile with Makefile

It will check system time to use Makefile. Some hardware may not have clock circuit, so you would need to update system time manually. For example, use

```
date -s "2015-7-27 11:06:04" to change the time
```

Makefile content:

```

[root@localhost helloworld]$ cat Makefile
CC=gcc
AR=ar
CFLAGS = -Wall -O2
all : HelloWorld
%.o : %.c
    $(CC) $(CFLAGS) -c $<
HelloWord: main.o
    $(CC) -o $@ $(CFLAGS) $<
clean:
    rm -f *.o HelloWorld *.out

```

Makefile compiling programs:

```

[root@localhost helloworld]$ make
gcc -Wall -O2 -c main.c
gcc -o HelloWorld -Wall -O2 main.o

```

Run compiling result:

```

[root@localhost helloworld]$ ls
HelloWord  Makefile  main.c  main.o
[root@localhost helloworld]$ ./HelloWord
Hello Word

```

3.2 Develop in PHP

The simplest is phpinfo.

```

[root@blueone src]$ ls
Makefile  hello  hello.c  hello.o  phpinfo.php
[root@blueone src]$ php-cgi phpinfo.php
You will see that php will list all information it supports.

```

3.3 Develop in Python

```

[root@localhost ~]$ python3
Python 3.7.0 (default, Oct 19 2018, 01:21:57)
[GCC 5.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World!')
Hello World!
>>> exit();
[root@localhost ~]$

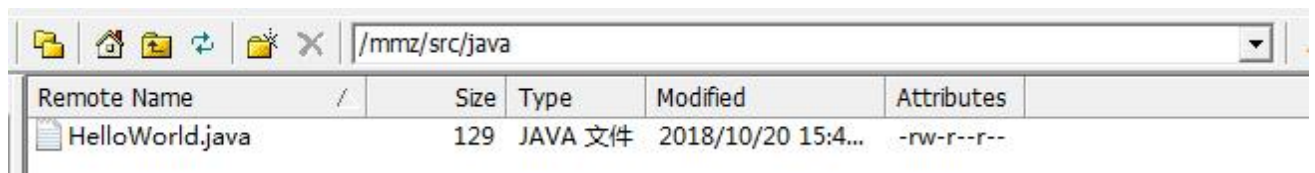
```

Pip3 setup library and SDK


```
[root@localhost mmz]$ pip3 install redis
The directory '/root/.cache/pip/http' or its parent directory is
ecuting pip with sudo, you may want sudo's -H flag.
The directory '/root/.cache/pip' or its parent directory is not a
pip with sudo, you may want sudo's -H flag.
Collecting redis
  Downloading https://files.pythonhosted.org/packages/3b/f6/7a76:
    100% |#####| 71kB 4.5kB/s
Installing collected packages: redis
Successfully installed redis-2.10.6
You are using pip version 10.0.1, however version 18.1 is availa
You should consider upgrading via the 'pip install --upgrade pip'
```

3.4 Develop in Java

Test file



File content

```
[root@localhost java]$ cat HelloWorld.java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Compiling file

```
[root@localhost bin]$ cd /mmz/src/java/
[root@localhost java]$ javac HelloWorld.java
[root@localhost java]$ ls
HelloWorld.class HelloWorld.java
[root@localhost java]$
```

Operating file

```
[root@localhost java]$ java HelloWorld
Hello World!
[root@localhost java]$
```

3.5 Apache Instructions

The device has been installed with Apache https-2.4.25 version. The default path is /mmz. Configuration file is /mmz/conf/httpd.conf. Webpage path is /mmz/htdocs. Webpage homepage is index.html.

You can configure parameters by the configuration file httpd.conf to realize different functions.

3.6 crond Instructions

Crond configuration file is /var/spool/cron/crontabs/root. You can modify this file to realize automatic process management.

4. Drive Related to Hardware

4.1 GPIO drive

Run device

```
int blue_io_open(void)
```

```
{    return open("/dev/mtxdev", O_RDWR);
```

```
}
```

IO control

4.2 GPIO IO Control Command

```
ioctl(fd,GPIO_CMD,GPIO_NUMBER);
```

GPIO_CMD fetch below numbers

```
#define GPIO_IO_MAGIC      'E'    /* for linux not used */
```

```
#define GPIO_IO_BASE      0
```

```
#define GPIO_CMD_0        _IO(GPIO_IO_MAGIC,(GPIO_IO_BASE+1))
```

```
#define GPIO_CMD_1        _IO(GPIO_IO_MAGIC,(GPIO_IO_BASE+2))
```

```
#define GPIO_CMD_DI       _IO(GPIO_IO_MAGIC,(GPIO_IO_BASE+3))
```

```
#define GPIO_CMD_DOUT     _IO(GPIO_IO_MAGIC,(GPIO_IO_BASE+4))
```

```
#define GPIO_CMD_READ     _IO(GPIO_IO_MAGIC,(GPIO_IO_BASE+5))
```

GPIO_CMD_0 indicates IO output is low level

GPIO_CMD_1 indicates IO output is high level

GPIO_CMD_DI indicates IO is set as input

GPIO_CMD_DOUT indicates IO is set as output

GPIO_CMD_READ indicates reading level of IO

4.3 GPIO Pin Numbers

GPIO_NUMBER. Different devices have different definitions. You can refer to PIN definition of device type.

4.3.1 HJ8000 and HJ8100 IO Pin instructions

36 ==> GPIO-36 COM1 direction control PIN when working on RS485 mode is controlled by the kernel.

The program only need to be initiated to output IO.

1 ==> GPIO-1 COM2 direction control PIN when working on RS485 mode is controlled by the kernel. The

program only need to be initiated to output IO.

19 ==> GPIO-19 COM1 RS485 and RS232 mode switch function pin,0==>RS232, 1==>RS485/RS422

6 ==> GPIO-6 COM2 RS485 and RS232 mode switch function pin,0==>RS232, 1==>RS485/RS422

5 ==> GPIO-5 3G/4G enable control pin, output a low pulse of >500ms

11 ==> GPIO-11 3G/4G reset pin,output a low pulse of >300ms

3 ==> GPIO-3 green LED control pin,0==>on, 1==>off

4 ==> GPIO-4 blue LED control pin,0==>on, 1==>off

0 ==> GPIO-0 power control pin,1==>turn on the power again after turning off the power for 5 seconds

4.3.2 HJ8300 and HJ8500 IO definition

- 5 → GPIO5 3G/4G enable control pin,output a low pulse of >500ms
- 11 → GPIO-11 3G/4G reset pin,output a low pulse of >300ms
- 3 → GPIO-3 green LED control pin,0==>on, 1==>off
- 0 → GPIO-0 power control pin,1==>turn on the power again after turning off the power for 5 seconds
- 44 → GPIO44 factory settings input pin

5. Software and Library Installation

5.1 System Time Setting

As some hardware do not have time circuit, you would need to set system time manually.

```
[root@blueone pip-8.1.2]$ date -s "2015-7-27 11:06:04"
```

5.2 python setuptools installation

```
[root@blueone python-install]$ tar xzf setuptools-24.0.2.tar.gz
[root@blueone python-install]$ cd setuptools-24.0.2/
[root@blueone setuptools-24.0.2]$ python setup.py install
running install
running bdist_egg
running egg_info
writing requirements to setuptools.egg-info/requirements.txt
```

5.3 python pip installation

```
[root@blueone python-install]$ ls
pip-8.1.2.tar.gz          setuptools-24.0.2          setuptools-24.0.2.tar.gz
[root@blueone python-install]$ tar xzf pip-8.1.2.tar.gz
[root@blueone python-install]$ cd pip-8.1.2/
[root@blueone pip-8.1.2]$ python setup.py install
.....
Installed /mmz/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg
Processing dependencies for pip==8.1.2
Finished processing dependencies for pip==8.1.2
[root@blueone pip-8.1.2]$
```

6. Issue Software and Self-start Software

6.1 Realize Self-starting by Start-up File

After developing software in SDK environment, you will need to install the software in the system and let the program self-start after enabling it in the system. This can be realized by modifying the start-up file.

Start-up file is in /etc/init.d/rcS

Please open and edit start-up file by vi command

```
#!/bin/sh
/bin/echo "configure mdev"
mount -t tmpfs -o size=64k,mode=0755 tmpfs /dev
mkdir /dev/pts
mount -t devpts devpts /dev/pts
mount -t proc proc /proc
```



```
mount -t sysfs sysfs /sys
mount -t usbfs none /proc/bus/usb
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s
source /etc/profile
/sbin/ldconfig
mount -t ext4 /dev/mmcblk0p1 /mmz
ulimit -n 128
/bin/echo "start main"
/mmz/sbin/sshd &
/bin/mainwdt
```

Notice: Do not change contents in above shaded part, otherwise it will cause device crash and not work.