MEasy IOT Development Guide

Make Your Idea Real

Table of Contents

MEasy IOT V1.0 Development Guide

Foreword

This document mainly describes the basic framework of MEasy IOT, and demonstrates the operation of MEasy IOT on the development board of Shenzhen MYIR Electronics Co., Ltd. (hereinafter referred to as "MYIR"), and further explains the construction of the MEasy IOT development environment and the compilation of the source code. Application integration, in the form of an example, describes how to develop more applications based on the MEasy IOT framework.

This document is suitable for embedded linux development engineers QT, Python Web-backend and front-end development engineers with certain development experience.

Version History:

Version	Description	Time
V1.0	Initial Version	2019.7.1

Hardware Version: This document is valid for the MYIR MYD-YA157 development board. The specific information is subject to the release package of the corresponding product..

> Note: The default password for root of the embedded Linux system is not set.

1. MEasy IOT Framework Introduction

MEasy IOT is a human-computer interaction system composed of WEB Demo developed by Shenzhen MYIR Technology Co., Ltd. The hardware platform is based on MYD-YA157; the software is based on the Linux file system produced by YOCTO, including the Python WEB running environment and WEB applications. Web Demo is a B / S architecture application based on Python 2.7, which includes Flask, Flask-Appbuilder, Javascript, css, HTML, socketio, dbus, tdbus and other components. The structure diagram of MEasy IOT is shown below.

MEasy IOT block diagram is shown as below：

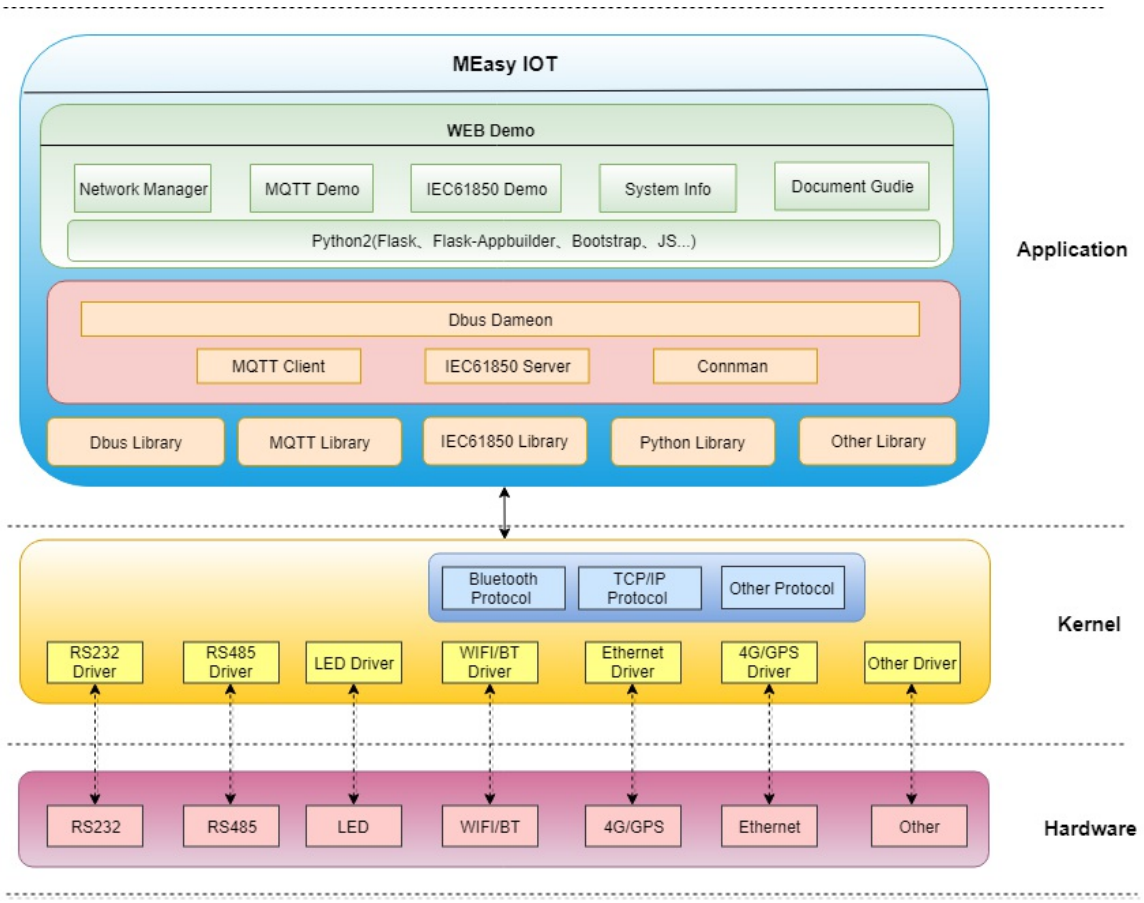

Figure 1-1 MEasy IOT Framework

D-Bus is an advanced inter-process communication mechanism that is provided by the freedesktop.org project and is distributed under the GPL license. The main purpose of D-Bus is to provide communication for processes in the Linux desktop environment, and to pass Linux desktop environment and Linux kernel events as messages to the process.

More details about dbus can be found here https://www.freedesktop.org/wiki/Software/dbus/

MEasy IOT uses D-Bus as the access interface for applications and underlying hardware. For example, ADC, LED control is accessed and controlled by Method and Signal of D-BUS implementation..

Connman is software for managing network devices running embedded linux devices. Connman is a fully modular system that can be expanded by plug-in to support the management of EtherNet, WIFI, 3G/4G, Bluetooth and other network devices. .

For more details on Connman, please refer to https://01.org/en/node/2207

The directory structure of the MEasy IOT software on the target board contains applications and libraries directly generated by YOCTO recipes. This section directly refers to *layers/meta-st/meta-st-openstlinux/recipes samples/measy-iot/measy-iot.bb*

2. Use Introduction

This section mainly introduces the details of the use and use of each APP in the MEasy IOT.

Software Environment：

- u-boot 2018.11
- linux-4.19.9
- File system with MEasy IOT operating environment
- MEasy IOT V1.0 application

> The above software is located in the CD-ROM directory 02-images/MEasy_IOT-images.

Hardware Environment：

- MYD-YA157 development board
- net cable
- PC

Hardware connection method:

Table 2-1 Development board display interface

Board	**Net Cable**
MYD-YA157	J3 Ether Net

Precautions:

- Preparation before use

  Insert the network cable into the corresponding interface before developing the board. The static IP address set by the development board network port J3 is 192.168.1.100. You need to add the IP of the 192.168.1 network segment to the PC, and then connect the development board and the PC to the switch or directly.

- WEB login method

  The default url of WEB is http://192.168.1.100:8080. The page after entering is as shown below.

Figure 2-2 Welcome Page

After entering the welcome page, you need to log in to operate. Click the Login button in the upper right corner to log in. The default account is admin and the password is admin..

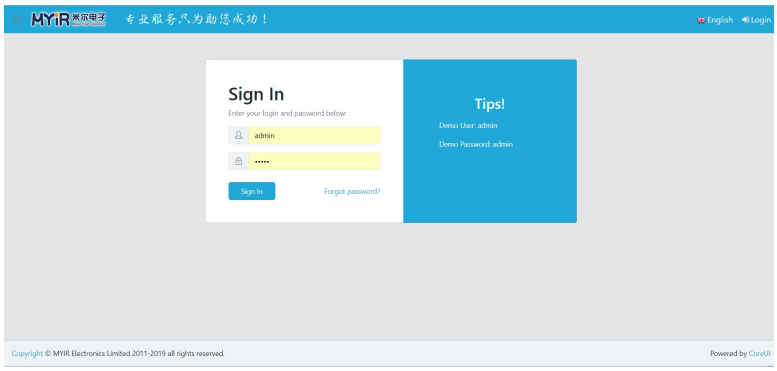

Figure 2-3 Login page

2.1 System Info

This page shows the software and hardware information of the MYD-YA157 development board.

Interface Description：

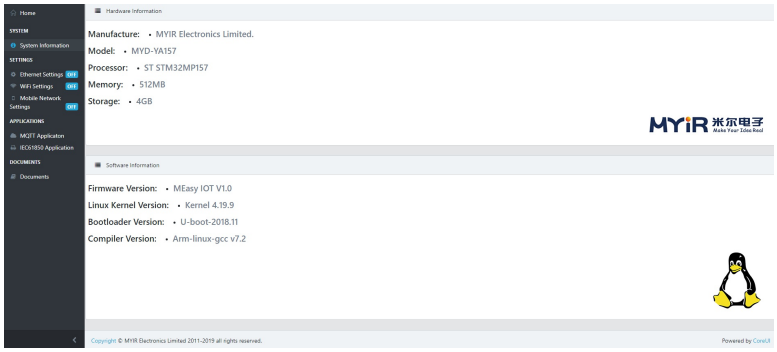

Figure 2-1-1 System Information Page

Test Procedure：

Click on the navigation bar system information to enter this page.

2.2 Ethernet Settings

This page provides network port IP address display and settings.

Hardware Environment:

- J3 Ether Net

  > Note: J3 is the network port used for WEB page access. If you modify the IP address, you need to re-access the new IP address.

Interface introduction:

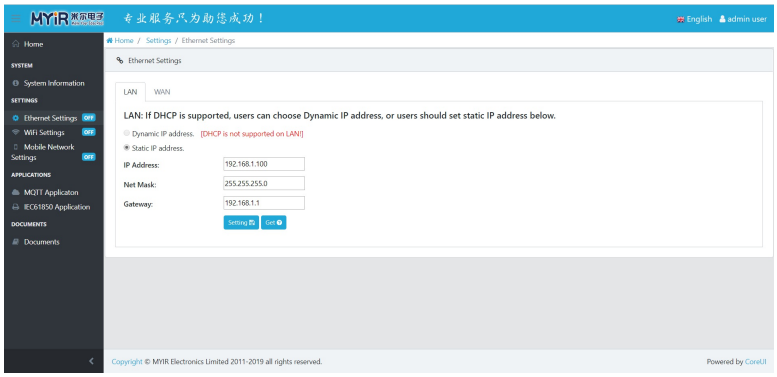

Figure 2-2-1 LAN page

> Note: 1. Located in the navigation bar Ethernet setting status badge, the status will be displayed after the WAN port accesses the network cable and successfully obtains the IP address. 2.J3 does not support DHCP to obtain IP.

Test steps:

- The Settings button is used to set the current DCHP mode and IP address information.
- The Get button is used to obtain the current DHCP mode and IP address information.

2.3 WIFI Setting

This page provides WIFI STA related operations.

Hardware environment:

- Onboard WIFI module
- WIFI antenna

Interface introduction:

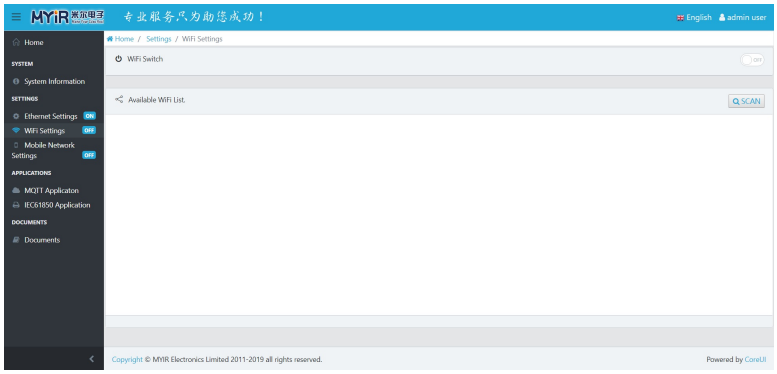

Figure 2-3-1 WIFI Settings Page

> Note: 1. Chinese WIFI name and hidden WIFI name are not currently supported.
>
> 1. The WIFI setting status badge located in the navigation bar will display ON after WIFI is turned on. After the WIFI connection is successful, the corresponding WIFI AP name will be displayed.
> 2. The WIFI list is arranged by the WIFI signal from high to low.

Test steps:

- Click the WIFI switch to turn on WIFI.

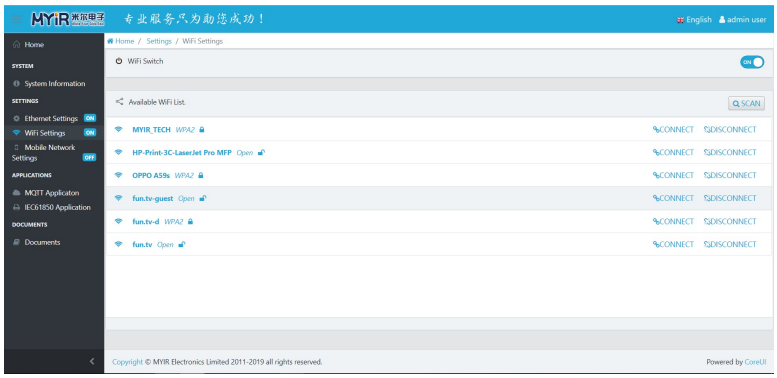

Figure 2-3-2 Page after opening WIFI

- Click the connection button that needs to be connected to the WIFI AP, and the WIFI connection page will pop up to enter the password.The password input page will be automatically closed after the WIFI connection is successful.

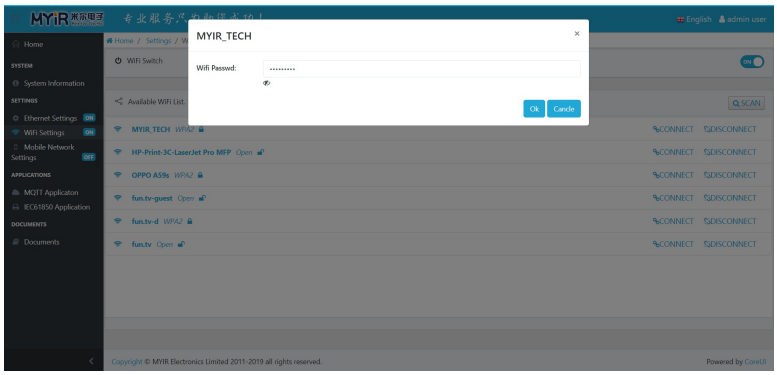

Figure 2-3-3 WIFI connection page

- After the WIFI connection is successful, the WIFI setting status badge located in the navation bar will display the currently connected WIFI AP name.

Figure 2-3-4 Successful WIFI connection page

2.4 Mobile Network Settings

This page provides the switching operation and IP information display of the 4G module.

Hardware environment:

- Onboard 4G module
- 4G module antenna

Interface introduction:

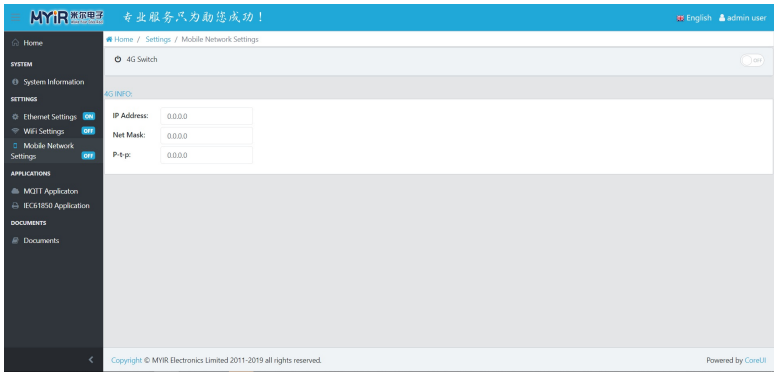

Figure 2-4-1 Mobile web settings page

Test steps:

- Click the 4G switch to turn on the 4G function and wait a few seconds. After the 4G connection is successful, the IP address information will be updated to the page.

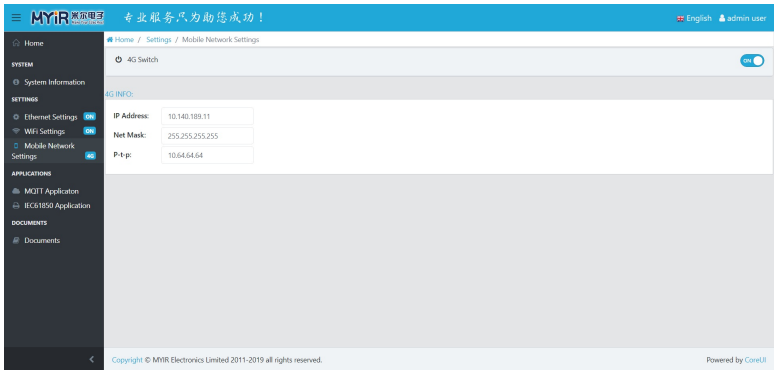

Figure 2-4-2 Page after successful 4G connection

2.5 MQTT Application

This page provides an MQTT Client application routine for MQTT message subscription and transceiving tests.

Hardware environment:

- A WIFI router that can connect to the Internet

Interface introduction:

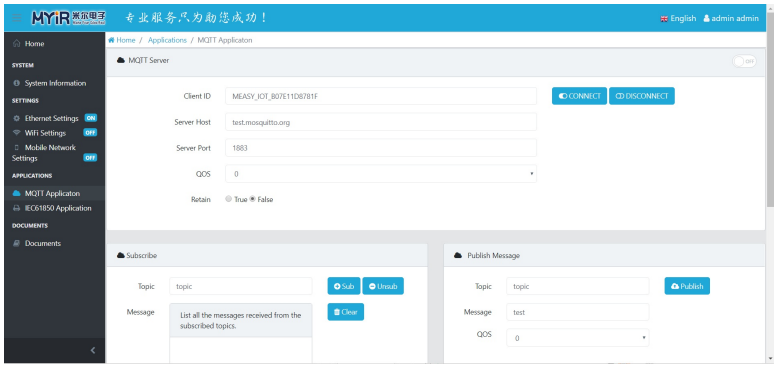

Figure 2-5-1 MQTT application page

MQTT server: MQTT server configuration page. Subscribe: MQTT Message Subscription and Unsubscribe page. Release message: MQTT message release page.

> Note:
>
> 1. MQTT function needs to be connected to the external network via WIFI to achieve.
> 2. The default parameter in the MQTT server page is even a public MQTT server, which can be used directly.

Test steps:

1. Make a WIFI connection through the WIFI setting page, and wait for the WIFI status badge on the left navigation bar to change to the name of the WIFI you are connected to.
2. Enter the MQTT application interface and click the connection button on the MQTT server page to connect to the MQTT server.

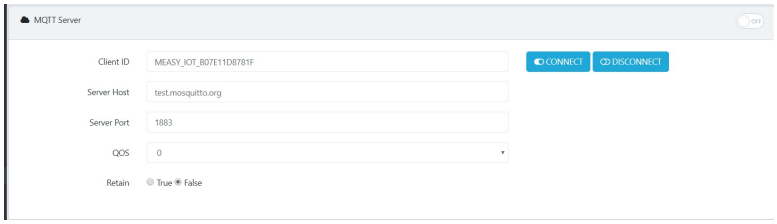

Figure 2-5-2 MQTT connection server succeeded

1. Click the Subscribe button in the MQTT subscription page, then click the Publish button in the MQTT publish page to receive the message on the subscription page.

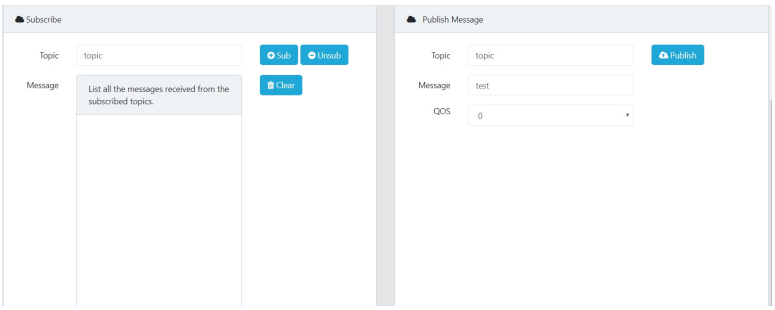

Figure 2-5-3 MQTT message subscription and release

2.6 IEC61850 Application

This page provides demonstration routines for DBUS and IEC61850 server communication.

Software Environment:

- PC side software IEDScout

  > This software requires the user to download and install it at
  > https://www.omicronenergy.com/cn/products/iedscout/options/

Interface introduction:

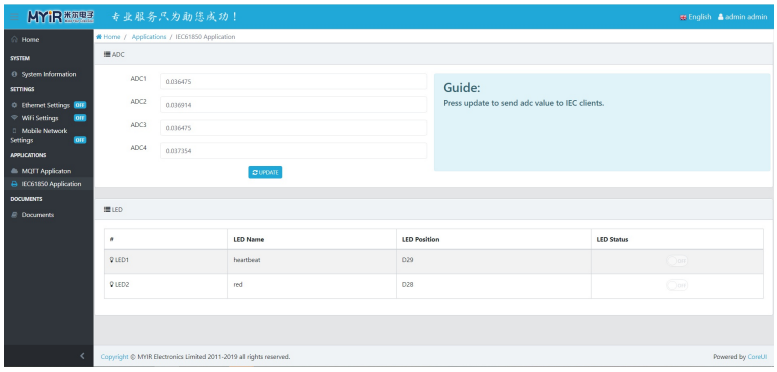

Figure 2-6-1 IEC61850 page

Test steps:

- Test LED

1.Enter the IEC61850 application interface and open the PC-side IEDSCout software.

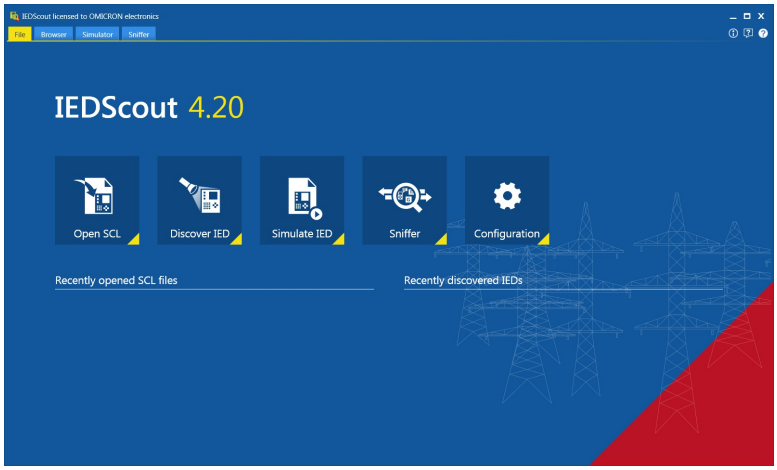

Figure 2-6-2 IEDScout software interface

1. Click the `Discover IED` button under the main interface of IECScout, and then enter the IP address `192.168.1.100` of the input server. After the input is complete, click the `Discover` button.

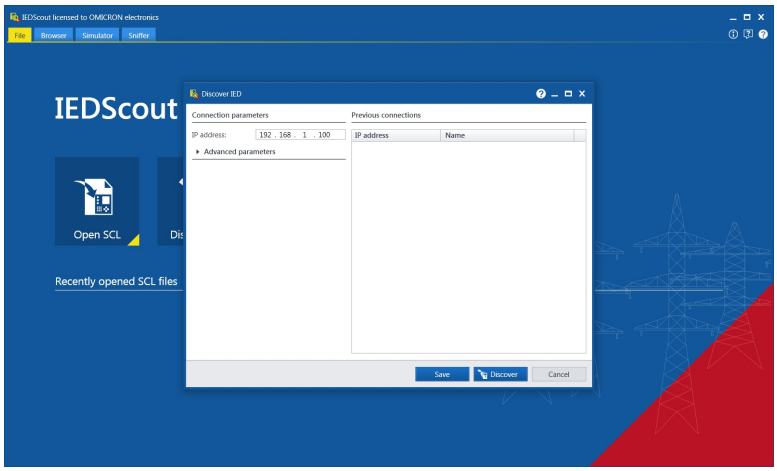

Figure 2-6-3 Discover IED interface

1. Go to the `Browser` page and you will see the IED device model named after MYIR1.

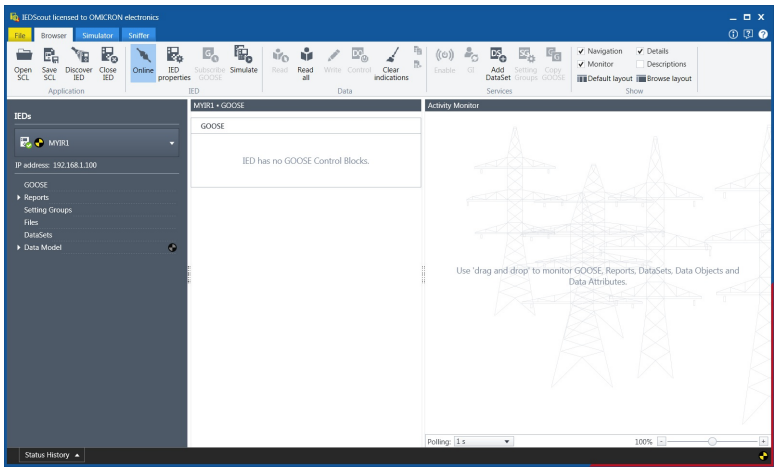

Figure 2-6-4 Brower IED interface

1. Click Data Models->LD1->GGIO1 under the IED device model to enter the general I/O control interface.

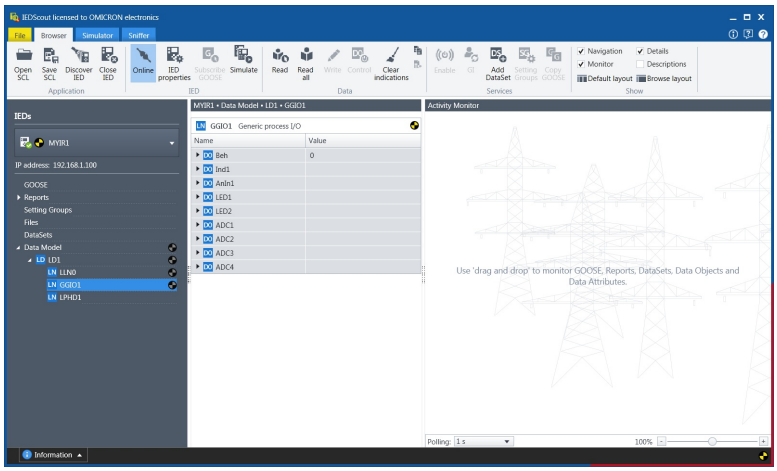

Figure 2-6-5 General I/O Control Interface

1. Double-click the LED1 data object under GGIO1, you can see the data property of Switch, and then click the Write button in the upper menu bar to control LED1.

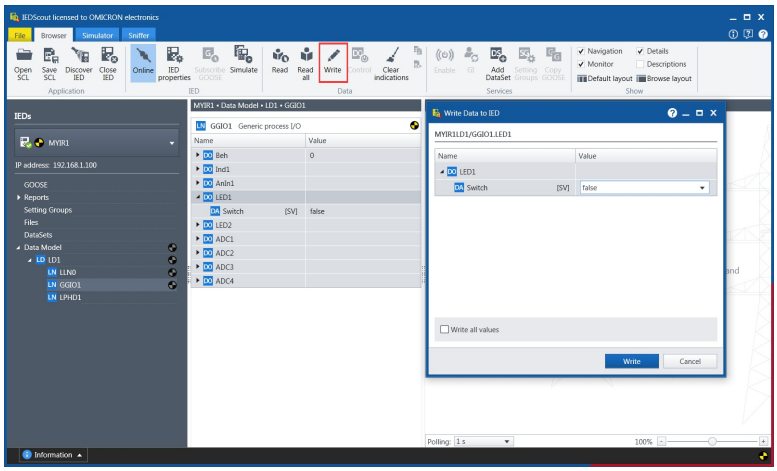

Figure 2-6-6 Data Attributes Operation Interface

1. Select true in the Value column, then click the Write button. At this time, the LED D2 on the development board is illuminated, and the status of LED1 in the IEC61850 application of the WEB page is also opened.

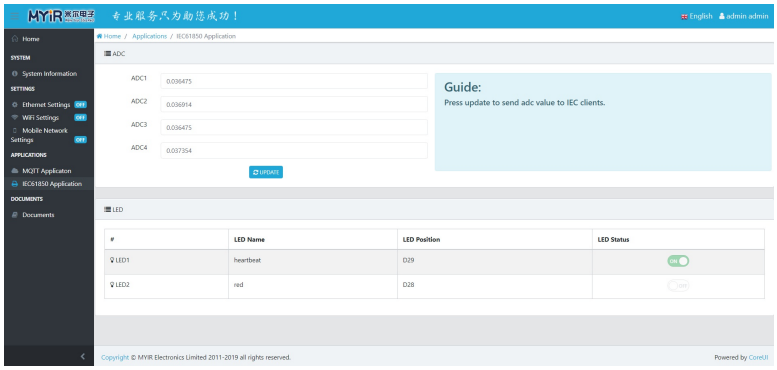

Figure 2-6-7 WEB IEC6185 application interface

- Test ADC

1.In the general I/O control interface of the IECScout software, click ADC1->adcMag to see the value of ADC1. This value is the same as the value of the ADC in the current WEB page IEC61850 application.

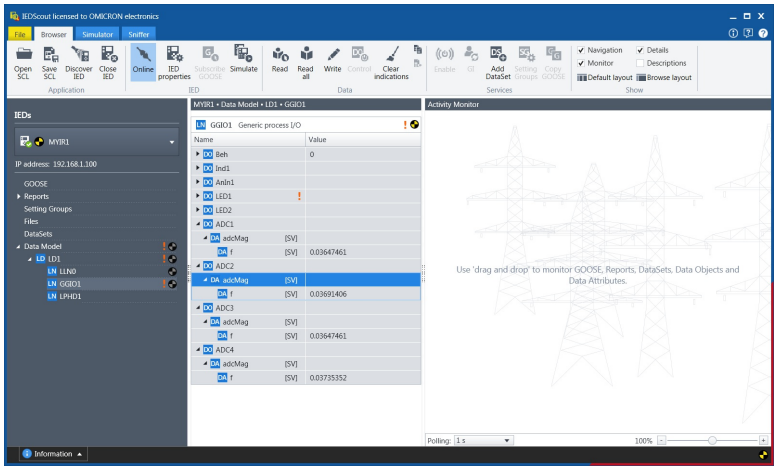

Figure 2-6-8 ADC value

1. Click the update button under the ADC in the WEB page IEC61850 application. At this time, the value of the ADC in the WEB page is updated to the latest value, and the value of the ADC is also written into the IEC Server. In this case, you need to click the menu bar in the IECScout software. The Read all button updates the value of the ADC in the general purpose I/O control interface.

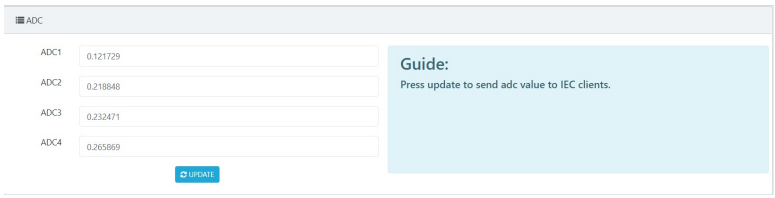

Figure 2-6-9 Update of WEB ADC

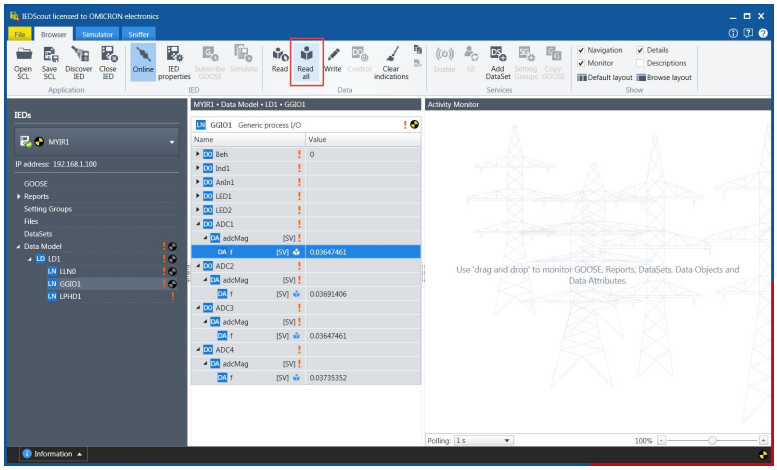

Figure 2-6-10 Update of IEC ADC

2.7 Document Guide

This page shows the documentation for the MEasy IOT Development Manual.

Interface introduction:

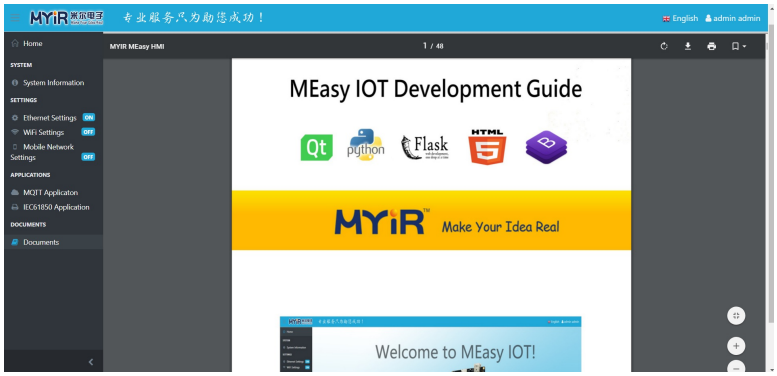

Figure 2-7-1 Document Guide

Use steps:

- Click on the navigation bar documentation guide.

3 Web Application Development

This chapter focuses on how to build the runtime environment for the MEasy IOT Web Demo application. Web Demo is an application based on B/S architecture written in Python 2.7, including Flask, Flask-Appbuilder, Javascript, css, HTML, socketio, dbus, tdbus and other components.

3.1 Build Runtime Environment

We provide MEasy IOT WEB Demo source code in the /04-Linux_Source/MEasy_IOT directory of the CD-ROM file, copy MEasy_IOT_WEB.tar.gz to the ubuntu directory working directory, and extract it.

```
├── app
├── app.db
├── babel
├── config.py
├── README.rst
├── requirements.txt
└── run.py
```

The `requirements.txt` in the directory is the component that the python web runtime environment depends on.

```
Babel==2.6.0
backports-abc==0.5
certifi==2018.10.15
chardet==3.0.4
Click==7.0
colorama==0.4.1
Flask==0.12.2
Flask-AppBuilder==1.12.3
Flask-Babel==0.11.1
Flask-Cors==3.0.3
Flask-JSONRPC==0.3.1
Flask-Login==0.4.0
Flask-OpenID==1.2.5
Flask-SocketIO==3.3.2
Flask-SQLAlchemy==2.3.2
Flask-WTF==0.14.2
gevent==1.2.2
greenlet==0.4.12
ifparser==0.8.2
itsdangerous==0.24
iw-parse==0.0.2
Jinja2==2.10
MarkupSafe==1.0
paho-mqtt==1.4.0
python-dateutil==2.7.5
python-engineio==3.5.1
python-openid==2.2.5
python-socketio==3.1.2
python-tdbus==0.11
pytz==2018.7
simplejson==3.11.1
singledispatch==3.4.0.3
six==1.11.0
smbus==1.1
SQLAlchemy==1.2.2
uWSGI==2.0.15
Werkzeug==0.12.2
```

The above python dependent components have been included in the file system generated by yocto.

The factory CD contains this file system located in the CD 02-Images/MEasy _IOT-images directory.

3.2 Run the WEB application

Copy MEasy_IOT_WEB.tar.gz to the development board and extract it.

```
tar zxvf MEasy_IOT_WEB.tar.gz
```

Before running MEasy IOT WEB, you need to start the dbus background process and myir_iec61850_server program. The program myir_iec61850_server contains IEC61850 server and dbus server.

```
# dbus-launch
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-c0AGn2s5XN,guid=d6544df82962d617c2dc37805c2c2d68
DBUS_SESSION_BUS_PID=5649
# export DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-c0AGn2s5XN


# myir_iec61850_server &
[1] 5755
# Using libIEC61850 version 1.3.0
```

Run the web application through the python interpreter.

```
# cd MEasy_IOT_WEB
# python2.7 run.py
/usr/lib/python2.7/site-packages/flask_sqlalchemy/__init__.py:794: FSADeprecationWarning: SQLALCHEMY_
TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future.
Set it to True or False to suppress this warning.
```

4 MEasy IOT Application Integration

In the previous chapters we introduced the directory structure of the MEasy IOT on the target board, as well as the runtime environment and development process of the MEasy IOT. This chapter will focus on how to integrate the MEasy IOT application into the target board system so that it can be booted up.

MEasy IOT includes the python web runtime environment and web-side applications. After the development and compilation are completed, the file system is ultimately packaged in yocto. The configured bb file is located in the `layers/meta-st/meta-st-openstlinux/recipes-samples/measy-iot` directory.

```
├── files
│   ├── etc
│   │   ├── dbus-1
│   │   └── systemd
│   ├── home
│   │   └── myir
│   ├── lib
│   │   ├── libiw.so -> libiw.so.30
│   │   └── systemd
│   └── usr
│       ├── bin
│       ├── lib
│       └── share
└── measy-iot.bb
```

When the `bitbake myir-image-measy-iot` command is executed in the yocto directory, the files in the above directories will be installed to the corresponding directory of the file system through the do_install () function in the measy-iot.bb recipe,For example, files in the `files/home/myir` directory will be copied to the file system `/home/myir` directory.

Enter the yocto directory and execute the following command to generate a file system containing MEasy IOT.

```
# DISTRO=openstlinux-eglfs MACHINE=stm32mp1 source layers/meta-st/scripts/envsetup.sh
# bitbake myir-image-measy-iot
```

The compiled image is located under the */tmp-glibc/deploy/images/stm32mp1* directory of the compilation directory.

5 IEC61850 Application Development

IEC 61850 is the international standard for communication systems and distributed energy (DER) management in substation automation systems (SAS). It realizes the standardization of engineering operation of intelligent substation through standard implementation. Make the engineering implementation of smart substation standardized, unified and transparent.

The open source libIEC61850 library is available on Linux systems, which provides server and client libraries for IEC 61850 / MMS, IEC 61850 / GOOSE and IEC 61850-9-2 / sampled value communication protocols written in C. For more information please visit http://libiec61850.com/libiec61850/

5.1 Configuration Compilation Environment

This section describes the compilation environment configuration process for the libIEC61850 library.

1.The ICD file in the IEC61850 library requires a JAVA tool for conversion, so you need to install the JAVA runtime environment first. The JDK installation package provided by Mill is located in the CD-03-Tools directory, copy jdk-8u191-linux-x64.tar.gz to the ubuntu working directory, and extract it.

```
# cd <WORKDIR>/JDK
# tar zxvf jdk-8u191-linux-x64.tar.gz
# cd jdk1.8.0_191
```

Configure the JAVA environment variable, modify the /etc/profile file, and add the following content to it:

```
# vi /etc/profile
export JAVA_HOME=<WORKDIR>/JDK/jdk1.8.0_191
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export  PATH=${JAVA_HOME}/bin:$PATH
```

After the modification is completed, save and exit. Use the following command to verify whether the JAVA runtime environment is successfully installed:

```
# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
```

2.configure the cross-compilation toolchain
We directly use the cross-compilation toolchain made by MYIR, copy the 04-Linux_source\Toolchain\arm-myir-linux-gnueabihf-gcc.tar.gz located on the CD-ROM to the working directory of ubuntu,Set the cross-compilation toolchain environment variables by the following operations.

```
# mkdir arm-myir-linux-gnueabihf-gcc
# tar xvf arm-myir-linux-gnueabihf-gcc.tar.gz -C arm-myir-linux-gnueabihf-gcc
# export ARCH=arm
# export CROSS_COMPILE=arm-myir-linux-gnueabihf-
# export PATH=$PATH:/<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/bin
```

After the setup is complete, use the following command to verify that the setup was successful.

```
# arm-myir-linux-gnueabihf-gcc -v
Use built-in specs。
COLLECT_GCC=/home/qinlh/buildroot/buildroot-2019.02.2/output/host/bin/arm-myir-linux-gnueabihf-gcc.br_real
COLLECT_LTO_WRAPPER=<workdir>/buildroot/output/host/libexec/gcc/arm-myir-linux-gnueabihf/7.4.0/lto-wrapper
...
Threading model：posix
gcc version 7.4.0 (Buildroot 2019.02.2-g04eff54)
```

5.2 Compile IEC61850 Library

This chapter describes the compilation of the IEC61850 library and the compilation process of the MYIR IEC61850 demo program myir_iec61850_server.

Our company provides libIEC61850 source code located in the /04-Linux_Source/IEC61850 directory of the CD-ROM file, copy libeec61850-1.3.0.tar.gz to the ubuntu directory working directory, and extract it.

```
# cp libiec61850-1.3.0.tar.gz <WORKDIR>/libiec61850
# cd <WORKDIR>/libiec61850
# tar zxvf libiec61850-1.3.0.tar.gz
# cd libiec61850-1.3.0
```

- Compile the IEC61850 library.

```
# make TARGET=LINUX-ARM
...
...
arm-myir-linux-gnueabihf-ar: creating ./build-arm/libiec61850.a
arm-myir-linux-gnueabihf-ranlib ./build-arm/libiec61850.a
```

- The MYIR IEC61850 demo uses some external libraries. You need to modify the Makefile located in the examples\myir_iec61850_server directory to specify the path to the external library and replace <WORKDIR> with the actual work path of the user.

```
CFLAGS +=-I./ \
          -I/<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/arm-myir-linux-gnueabihf/sysroot/usr/include/glib-2
.0/ \
          -I/<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/arm-myir-linux-gnueabihf/sysroot/usr/lib/glib-2.0/i
nclude/ \
          -I/<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/arm-myir-linux-gnueabihf/sysroot/usr/include/cjson/
 \
          -I/<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/arm-myir-linux-gnueabihf/sysroot/usr/include \
          -I/<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/rm-myir-linux-gnueabihf/sysroot/usr/include/libxml2
LDFLAGS +=  -lpthread  -ldbus-1 -lxml2 -lcjson \
-L /<WORKDIR>/arm-myir-linux-gnueabihf-gcc/usr/arm-myir-linux-gnueabihf/sysroot/usr/lib
```

If the above compilation process reports an error, please check if there is a problem with the cross-compilation toolchain settings.

- Compile the ICD file.

```
# cd examples/myir_iec61850_server
# make model
java -jar ../../tools/model_generator/genmodel.jar myir_iec61850_server.icd
Select ICD File myir_iec61850_server.icd
parse data type templates ...
parse IED section ...
parse communication section ...
Found connectedAP ap1 for IED MYIR1
print report instance 01
print report instance 02
```

If the above process is reported incorrectly, please check if the JAVA runtime environment is set successfully.

- Compile MYIR IEC61850 demo program.

```
# cd examples/myir_iec61850_server
# make  TARGET=LINUX-ARM
```

5.3 Running IEC61850 Application

After the compilation is completed, the MYIR IEC61850 application is located under the examples/myir_iec61850_server/ directory, and the myir_iec61850_server is copied to the development board. The running process is as follows.

```
# dbus-launch
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-c0AGn2s5XN,guid=d6544df82962d617c2dc37805c2c2d68
DBUS_SESSION_BUS_PID=5649
# export DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-c0AGn2s5XN


# ./myir_iec61850_server
# Using libIEC61850 version 1.3.0
```

After successful operation, you can refer to Chapter 2.6 for testing.

Appendix A Warranty & Technical Support Services

MYIR Tech Limited is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded
applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR's products.

Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

Delivery Time

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support

MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers one year free technical support and after-sales maintenance service from the purchase date. The service covers:

Technical support service

- MYIR offers technical support for the hardware and software materials which have provided to customers;
- To help customers compile and run the source code we offer;
- To help customers solve problems occurred during operations if users follow the user manual documents;
- To judge whether the failure exists;
- To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

- Hardware or software problems occurred during customers' own development;
- Problems occurred when customers compile or run the OS which is tailored by themselves;
- Problems occurred during customers' own applications development;
- Problems occurred during the modification of MYIR's software source code.

After-sales maintenance service

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

- The warranty period is expired;
- The customer cannot provide proof-of-purchase or the product has no serial number;
- The customer has not followed the instruction of the manual which has caused the damage the product;
- Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;
- Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;
- Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;
- Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips:

1. MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.
2. Please do not use finger nails or hard sharp object to touch the surface of the LCD.
3. MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.
4. Do not clean the surface of the screen with chemicals.
5. Please read through the product user manual before you using MYIR's products.

6. For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR's support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

Maintenance period and charges

- MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
- For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

Shipping cost

During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

Products Life Cycle

MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

1. MYIR provides services of driver development base on MYIR's products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

MYIR Tech Limited

Address: Room 04, 6th Floor, Building No.2, Fada Road, Yunli Smart Park, Bantian, Longgang District, Shenzhen, Guangdong, China 518129

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836

Fax: +86-755-25532724

Website: www.myirtech.com