

MEasy IOT 开发手册



MYiR™ Make Your Idea Real

专业服务只为助您成功！ Chinese 登录

首页

系统

系统信息

设置

以太网设置 OFF

WiFi设置 OFF

移动网络设置 OFF

应用演示

MQTT应用

IEC61850应用

文档指南

文档指南

欢迎使用MEasy IOT!



电话: 0755-25622735 25502782 | 传真: 0755-25532724 | 邮箱: sales.cn@myirtech.com project@myirtech.com
技术支持: support.cn@myirtech.com | 技术支持电话: 027-59621648 | 地址: 深圳市龙岗区坂田街道发达路云里智能园2栋604室 | www.myir-tech.com



目錄

前言	0
1. MEasy IOT框架介绍	1
2. 使用介绍	2
2.1 系统信息	2.1
2.2 以太网设置	2.2
2.3 WIFI设置	2.3
2.4 移动网络设置	2.4
2.5 MQTT应用	2.5
2.6 IEC61850应用	2.6
2.7 文档指南	2.7
3. WEB应用开发	0
3.1 构建运行环境	0.1
3.2 运行WEB应用	0.2
4. MEasy IOT应用集成	1
5. IEC61850应用开发	2
5.1 构建编译环境	2.1
5.2 编译IEC61850应用	2.2
5.3 运行IEC61850应用	2.3
附录A	0
附录B	1

MEasy IOT V1.0 开发手册

前言

本文档主要讲述MEasy IOT的基本框架，并演示MEasy IOT在深圳市米尔电子有限公司(下文简称“米尔”)开发板上的运行，此外还进一步说明了MEasy IOT开发环境的搭建和源码的编译以及应用集成，以实例的形式讲述了如何在MEasy IOT框架的基础上开发更多的应用。

本文档适合有一定开发经验的嵌入式linux开发工程师，QT开发工程师和Web前端和后端开发工程师。

版本历史：

版本号	描述	时间
V1.0	初始版本	2019.7.1

硬件版本：

本文档适用于米尔MYD-YA157开发板，具体信息以相应产品的发布包为准。

注意：开发板Linux系统默认的root账户密码为空。

1. MEasy IOT框架介绍

MEasy IOT是深圳市米尔科技有限公司开发的一套由WEB Demo组成的人机交互系统。硬件平台基于MYD-YA157；软件基于YOCTO制作生成的linux文件系统，包含Python WEB运行环境和WEB应用程序。Web Demo是基于Python2.7编写的B/S架构的应用，包含Flask, Flask-Appbuilder, Javascript, css, HTML, socketio, dbus, tdbus等组件。MEasy IOT的结构框图如下所示：

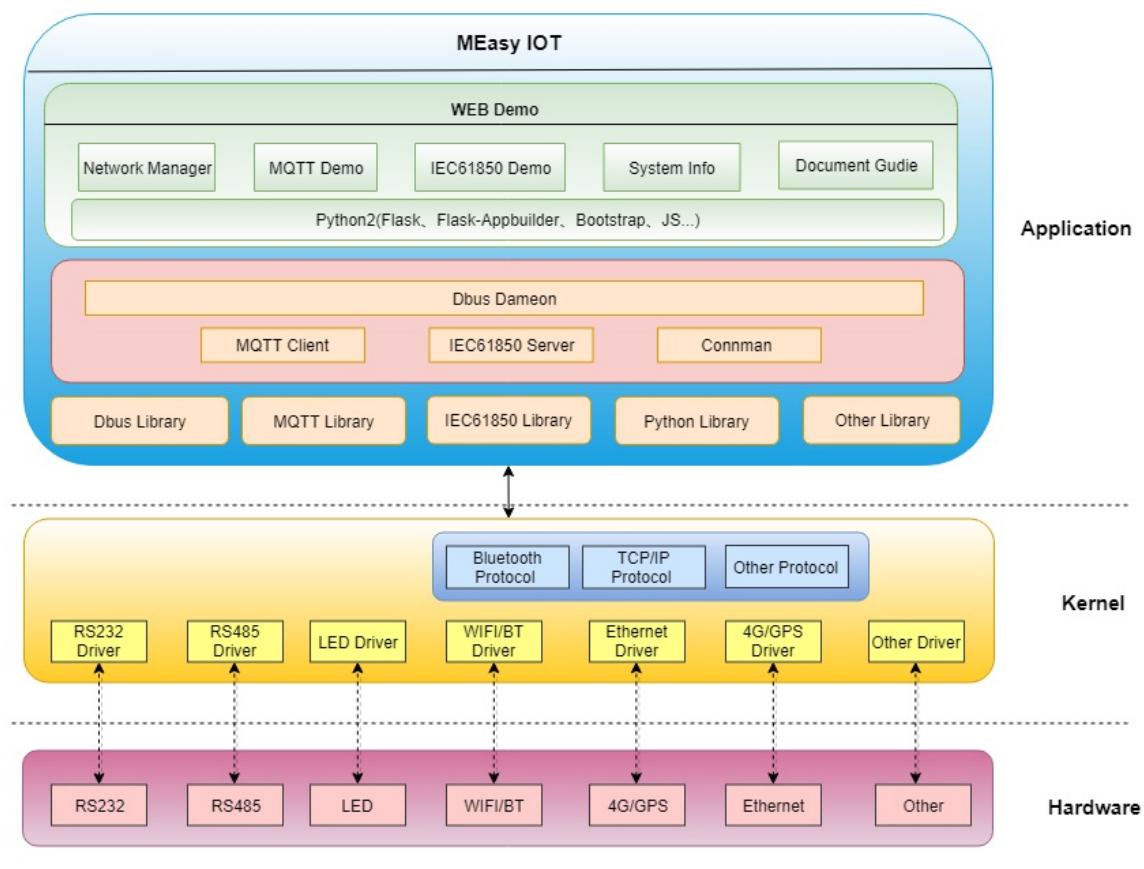


图1-1 MEasy IOT结构框图

MEasy IOT使用D-Bus作为应用程序和底层硬件的访问接口。例如ADC, LED的控制都是通过D-Bus的实现的Method和Signal来进行访问和控制的。关于D-bus的更多细节请参考<http://dbus.freedesktop.org>。

MEasy IOT中的网络管理应用则使用开源的Connman作为中间层来实现对网络设备的控制，Connman也是一个基于D-Bus的完全模块化的系统，可以通过插件化进行扩展，以支持EtherNet、WIFI、3G/4G、Bluetooth等网络设备的管理。关于Connman的更多细节请参考<https://01.org/zh/node/2207>

MEasy IOT软件在目标板上目录结构包含由YOCTO配方直接生成的应用程序和库，这部分直接参考[layers/meta-st/meta-st-openstlinux/recipes-samples/measy-iot/measy-iot.bb](https://git.yoctoproject.org/meta-st/recipes-samples/measy-iot/measy-iot.bb)。

2. 使用介绍

本节主要介绍MEasy IOT的使用及使用过程中注意的细节。软件环境：

- u-boot 2018.11
 - linux-4.19.9
 - 带MEasy IOT运行环境的文件系统
- 以上软件位于光盘目录02-images/MEasy_IOT-images。

硬件环境：

- MYD-YA157开发板
- 网线
- PC主机

硬件连接方式：

表2-1 开发板接口

开发板	网口
MYD-YA157	J3 Ether Net

注意事项：

- 使用前准备 开发板上电前请网线插到对应的接口。开发板网口1设置的静态IP为192.168.1.100，需要在PC添加192.168.1网段的IP，然后将开发板和PC机接入交换机中或者直连。
- WEB登录方式

WEB默认的url为<http://192.168.1.100:8080>，进入后的页面如下图所示。



图2-2 欢迎页面

进入欢迎页面后需要登录才能进行操作，点击右上角的Login按钮进行登录，默认帐号为admin，密码为admin。



图2-3 登录页面

2.1 系统信息

本页面展示了MYD-YA157开发板的软硬件信息。

界面说明：

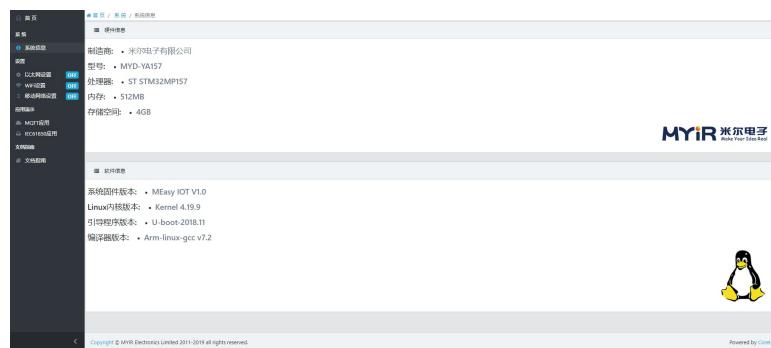


图2-1-1 系统信息页面

测试步骤：

点击导航栏系统信息即可进入本页面。

2.2 以太网设置

本页面提供网口IP地址显示和设置。

硬件环境：

- J3 Ether Net

注意：J3是用于WEB页面访问的网口，如果修改了IP需要重新以新IP地址访问。

界面介绍：



图2-2-1 局域网页面

注意：1.位于导航栏以太网设置状态徽章，在广域网口接入网线并成功获取到IP地址后会显示ON的状态。2.局域网口不支持DHCP获取IP。

测试步骤：

- 设置按钮用于设置当前DCHP方式和IP地址信息。
- 获取按钮用于获取当前DCHP方式和IP地址信息。

2.3 WIFI设置

本页面提供了WIFI STA功能相关操作。

硬件环境：

- 板载WIFI模块
- WIFI天线

界面介绍：



图2-3-1 WIFI设置页面

注意：1.当前不支持中文WIFI名称和隐藏的WIFI名称。2.位于导航栏的WIFI设置状态徽章，在WIFI打开后会显示ON，在WIFI连接成功后会显示对应WIFI AP名称。3.WIFI列表由WIFI信号从高到低排列。

测试步骤：

- 点击WIFI开关，打开WIFI。

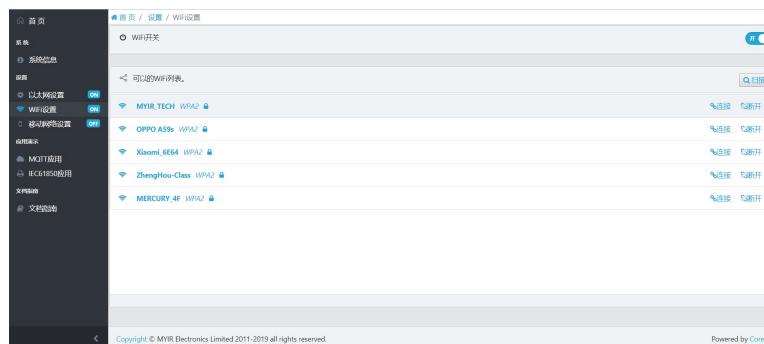


图2-3-2 打开WIFI后得页面

- 点击需要连接WIFI AP的连接按键，弹出WIFI连接页面输入密码。WIFI连接成功后会自动关闭密码输入页面。

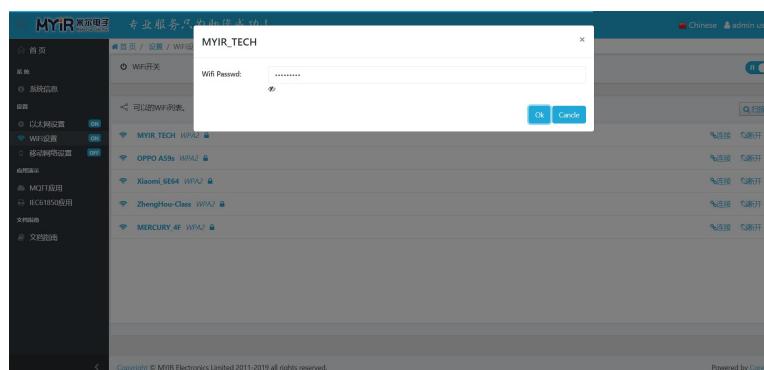


图2-3-3 WIFI连接页面

- WIFI连接成功后，位于导航栏的WIFI设置状态徽章会显示当前连接的WIFI AP名称。



图2-3-4 WIFI连接成功的页面

2.4 移动网络设置

本页面提供4G模块的开关操作和IP信息显示。

硬件环境：

- 板载4G模块
- 4G模块天线

界面介绍：



图2-4-1 移动网络设置页面

注意：1.SIM卡必须在开发板上电前插入4G模块中。

测试步骤：

- 点击4G开关，打开4G功能，等待几秒，4G连接成功后IP地址信息会更新到页面中。

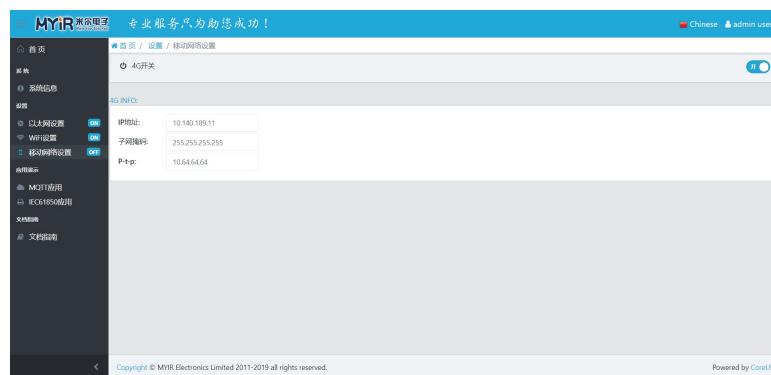


图2-4-2 4G连接成功后的页面

2.5 MQTT应用

本页面提供一个MQTT Client应用例程用于MQTT消息订阅和收发测试。

硬件环境：

- 一个可连接外网的WIFI路由器

界面介绍：



图2-5-1 MQTT应用页面

MQTT服务器：MQTT服务器配置页面。

订阅：MQTT消息订阅与退订页面。

发布消息：MQTT消息发布页面。

注意：

1.MQTT功能需要通过wifi连接外网来实现。2.MQTT服务器页面中的默认参数是连一个公开的MQTT服务器，可以直接使用。

测试步骤： 1.通过WIFI设置页面进行WIFI连接，等待左侧的导航栏上面WIFI状态徽章变为你所连接的WIFI名称。 2.进入MQTT应用界面，点击MQTT服务器页面中的连接按钮，来连接MQTT服务器。

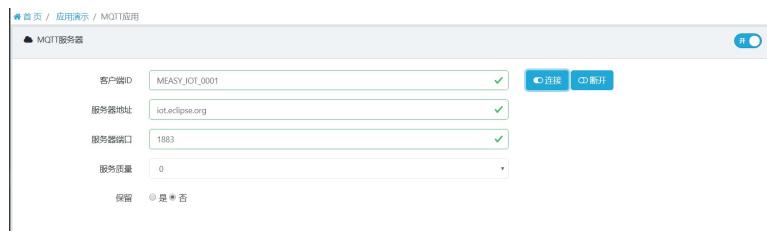


图2-5-2 MQTT连接服务器成功

3.点击MQTT订阅页面中的订阅按钮，然后再点击MQTT发布页面中的发布按钮，订阅页面中即可收到消息。

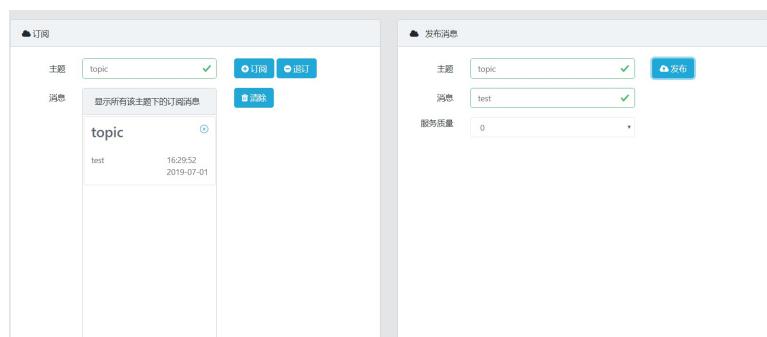


图2-5-3 MQTT消息订阅与发布

2.6 IEC61850应用

本页面提供了DBUS和IEC61850服务端通信的演示例程。 软件环境：

- PC端软件IEDScout

此软件需要用户自行下载安装，下载地址<https://www.omicronenergy.com/cn/products/iedscout/options/>

界面介绍：



图2-6-1 IEC61850页面

测试步骤：

- 测试LED

1. 进入IEC61850应用界面，打开PC端IEDScout软件。

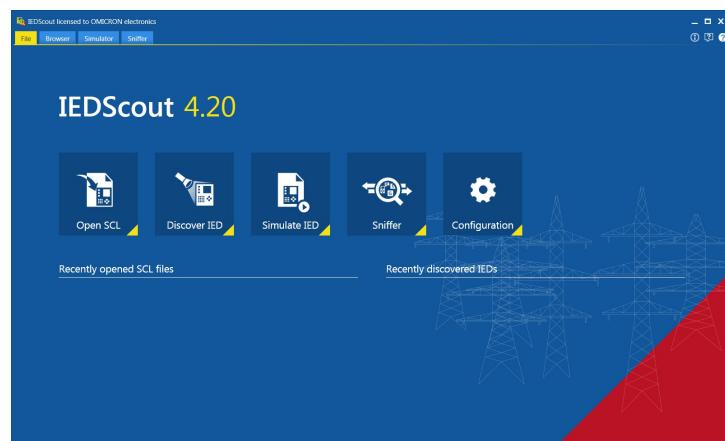


图2-6-2 IEDScout软件界面

1. 点击IECScout主界面下面的 discover IED 按钮，然后输入输入服务端的IP地址 192.168.1.100， 输入完成后点击 Discover 按钮。

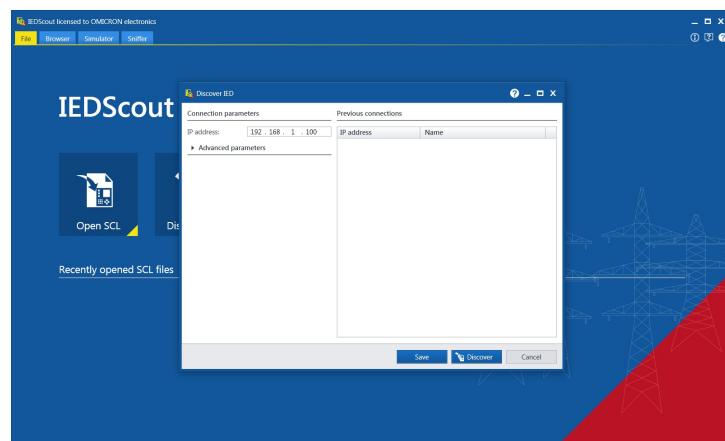


图2-6-3 Discover IED界面

1. 进入 Browser 页面，可以看到以MYIR1命名的IED设备模型。

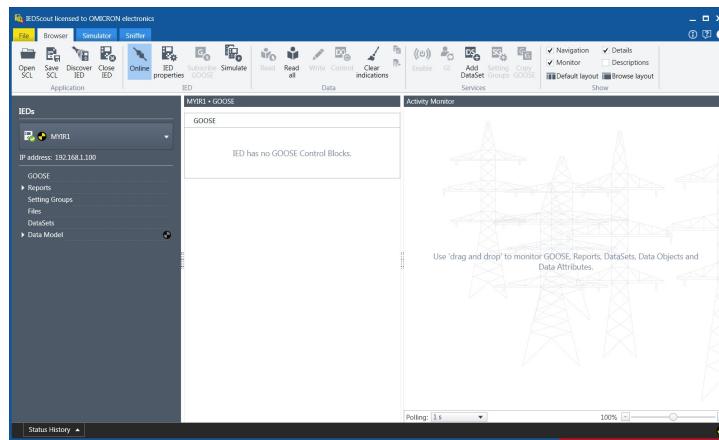


图2-6-4 Brower IED界面

4.点击IED设备模型下面的Data Models->LD1->GGIO1进入通用I/O控制界面。

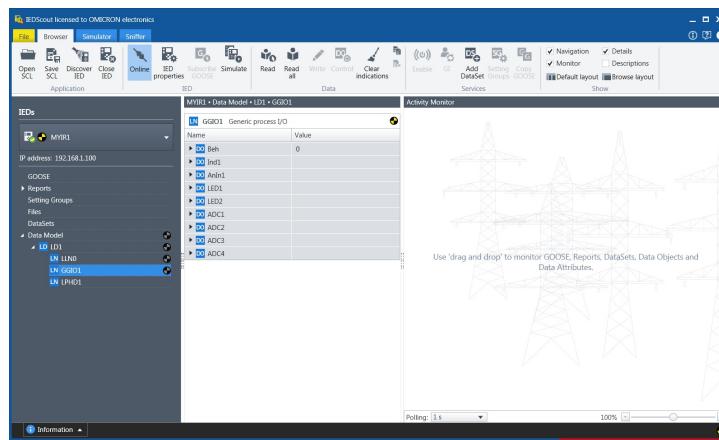


图2-6-5 通用I/O控制界面

5.双击GGIO1下面的LED1数据对象，可以看到Switch这个数据属性，然后点击上方菜单栏的Write按钮即可对LED1进行控制.

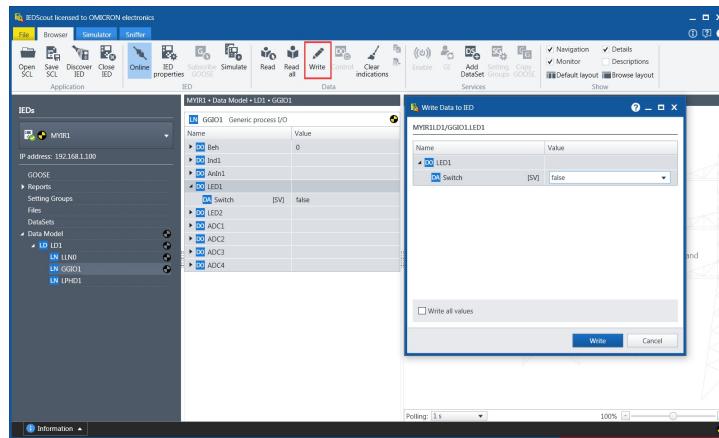


图2-6-6 数据属性操作界面

6.在Value栏选择true，然后点击Write按键，此时开发板上的LED D2就被点亮了，同时WEB页面IEC61850应用中LED1的状态也被打开了。

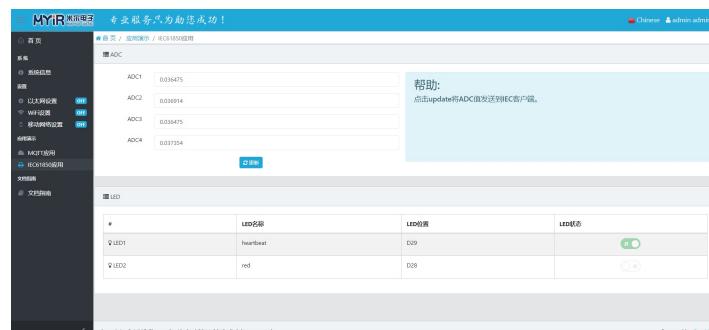


图2-6-7 WEB IEC6185 应用界面

- 测试ADC

1.在IECScout软件通用I/O控制界面，依次点击ADC1->adcMag即可看到ADC1的值，这个值和当前WEB页面IEC61850应用中的ADC的值是一样的。

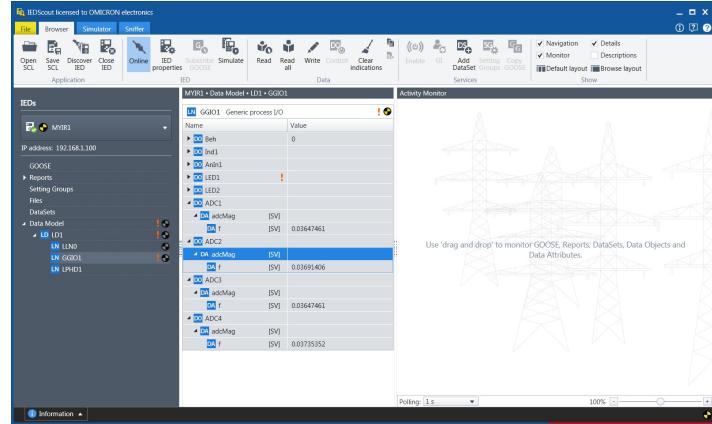


图2-6-8 ADC的值

2.在WEB页面IEC61850应用中点击ADC下面的更新按键，此时WEB页面中ADC的值更新到了最新值，同时ADC的值也写入到了IEC Server中，此时需要在IECScout软件中点击菜单栏的Read all按钮来更新通用I/O控制界面中ADC的值。

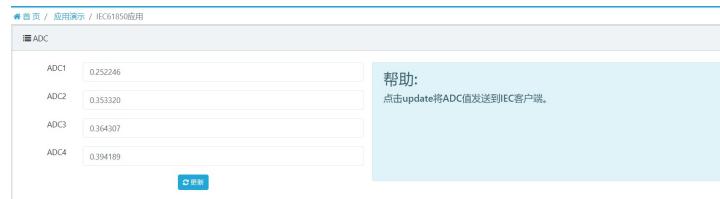


图2-6-9 WEB ADC的更新

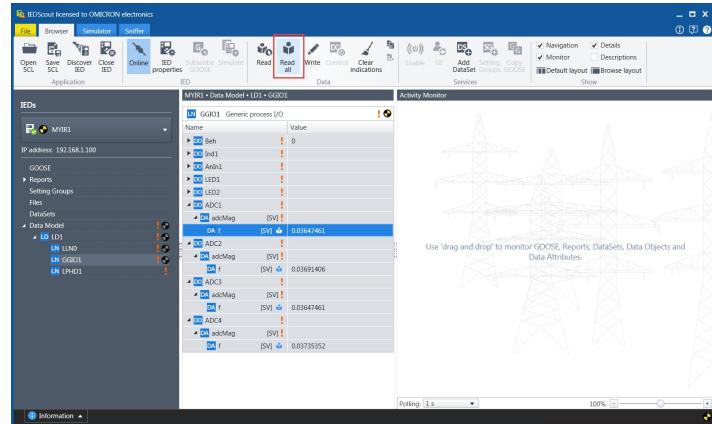


图2-6-10 IEC ADC的更新

2.7 文档指南

本页面展示MEasy IOT开发手册的文档。

界面介绍：



图2-7-1 文档指南

使用方法：

点击导航栏文档指南即可。

3 Web 应用开发

本章重点介绍如何构建MEasy IOT Web Demo应用的运行环境。Web Demo是基于Python2.7编写的B/S架构的应用，包含Flask, Flask-Appbuilder, Javascript, css, HTML, socketio, dbus,tdbus等组件。

3.1 构建运行环境

我们提供MEasy IOT WEB Demo源码位于光盘文件的/04-Linux_Source/MEasy_IOT目录，将MEasy_IOT_WEB.tar.gz拷贝到ubuntu目录工作目录，并解压出来。

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

目录中的 requirements.txt 是python web运行环境所依赖的组件。

```
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
ipparser==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
```

上述python依赖组件都已包含在了yocto所生成的文件系统里面。

出厂光盘里面包含这个文件系统位于光盘02-Images/MEasy_IOT-images目录。

3.2 运行WEB应用

拷贝MEasy_IOT_WEB.tar.gz 到开发板，解压出来。

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

运行MEasy IOT WEB之前还需要启动dbus后台进程和myir_iec61850_server程序， myir_iec61850_server这个程序包含IEC61850 server和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
```

通过python解释器来运行WEB应用。

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

4 MEasy IOT应用集成

在前面的章节中我们介绍了MEasy IOT运行环境的构建过程以及如何在开发板上运行MEasy IOT。本章将重点介绍如何将MEasy IOT应用集成到目标板系统当中，使之开机就能启动。

MEasy IOT包含python web 运行环境和Web端应用，开发编译完成之后，最终都是在yocto中打包生成的文件系统。已经配置好的bb文件位于layers/meta-st/meta-st-openstlinux/recipes-samples/measy-iot目录。

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

在yocto执行bitbake myir-image-measy-iot的时候，上述目录的文件都会通过measy-iot.bb配方文件里面的do_install()函数安装到文件系统对应的目录，例如 files/home/myir 目录下的文件会拷贝至文件系统 /home/myir 目录下面。

进入yocto目录执行下列命令即可生成包含MEasy IOT的文件系统。

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

编译完成后的镜像位于编译目录的/tmp-glibc/deploy/images/stm32mp1目录下面。

5. IEC61850应用开发

IEC61850是变电站自动化系统（SAS）中通信系统和分散能源（DER）管理的国际标准。它通过标准的实现，实现了智能变电站的工程运作标准化。使得智能变电站的工程实施变得规范、统一和透明。

在Linux系统上可以使用开源的libIEC61850库，该库提供了用C语言编写的IEC 61850 / MMS, IEC 61850 / GOOSE和IEC 61850-9-2 /采样值通信协议的服务端和客户端库。了解更多请访问<http://libiec61850.com/libiec61850/>

5.1 配置编译环境

本章节讲述libIEC61850库的编译环境配置过程。

1.IEC61850库中的ICD文件需要一个JAVA工具来进行转换，所以需要先安装JAVA运行环境。米尔提供的JDK安装包位于光盘03-Tools目录，拷贝jdk-8u191-linux-x64.tar.gz到ubuntu工作目录，并解压出来。

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

配置JAVA环境变量，修改/etc/profile文件，在里面添加如下内容：

```
# 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
```

修改完成保存退出，使用下面命令来验证JAVA运行环境是否安装成功：

```
# 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.配置交叉编译工具链

这里直接使用MYIR制作的交叉编译工具链。拷贝位于光盘04-Linux_source\Toolchain\arm-myir-linux-gnueabihf-gcc.tar.gz到ubuntu的工作目录。通过下面操作设置交叉编译工具链环境变量。

```
# 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
```

设置完成后使用如下命令验证设置是否成功。

```
# arm-myir-linux-gnueabihf-gcc -v
使用内建 specs。
COLLECT_GCC=/home/qinlh/buildroot/buildroot-2019.02.2/output/host/bin/arm-myir-linux-gnueabihf-gcc.br_real
COLLECT_LTO_WRAPPER=/home/qinlh/buildroot/buildroot-2019.02.2/output/host/libexec/gcc/arm-myir-linux-gnueabihf/7.4.0/
lto-wrapper
...
...
线程模型: posix
gcc 版本 7.4.0 (Buildroot 2019.02.2-g04eff54)
```

5.2 编译IEC61850库

本章节讲述IEC61850库的编译和MYIR IEC61850演示程序myir_iec61850_server的编译过程。

我司提供libIEC61850源码位于光盘文件的/04-Linux_Source/IEC61850目录，将libiec61850-1.3.0.tar.gz拷贝到ubuntu目录工作目录，并解压出来。

```
# 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
```

- 编译IEC61850库。

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

- MYIR IEC61850演示程序用到了一些外部库，需要修改位于examples\myir_iec61850_server\目录下的Makefile来指定外部库的路径，将<WORKDIR>替换为用户实际的工作路径。

```
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/include/ \
-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
```

如果上述编译过程报错请检查交叉编译工具链设置是否有问题。

- 编译ICD文件。

```
# 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
```

如果上述过程报错请检查JAVA运行环境是否设置成功。

- 编译MYIR IEC61850演示程序

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

5.3 运行IEC61850应用

编译完成后MYIR IEC61850应用位于examples/myir_iec61850_server/目录下面，将myir_iec61850_server拷贝至开发板，运行过程如下所示。

```
# 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
```

运行成功后，可以参考第2.6章来进行测试。

附录一 联系方式

销售联系方式

- 网址: www.myir-tech.com
- 邮箱: sales.cn@myirtech.com

深圳总部

- 负责区域: 广东 / 四川 / 重庆 / 湖南 / 广西 / 云南 / 贵州 / 海南 / 香港 / 澳门
- 电话: 0755-25622735 0755-22929657
- 传真: 0755-25532724
- 邮编: 518020
- 地址: 深圳市龙岗区坂田街道发达路云里智能园2栋6楼04室

上海办事处

- 负责区域: 上海 / 湖北 / 江苏 / 浙江 / 安徽 / 福建 / 江西
- 电话: 021-60317628 15901764611
- 传真: 021-60317630
- 邮编: 200062
- 地址: 上海市普陀区中江路106号北岸长风I座1402

北京办事处

- 负责区域: 北京 / 天津 / 陕西 / 辽宁 / 山东 / 河南 / 河北 / 黑龙江 / 吉林 / 山西 / 甘肃 / 内蒙古 / 宁夏
- 电话: 010-84675491 13269791724
- 传真: 010-84675491
- 邮编: 102218
- 地址: 北京市昌平区东小口镇中滩村润枫欣尚2号楼1009

技术支持联系方式

- 电话: 027-59621648
- 邮箱: support.cn@myirtech.com

如果您通过邮件获取帮助时, 请使用以下格式书写邮件标题:

【公司名称/个人--开发板型号】问题概述

这样可以使我们更快速跟进您的问题, 以便相应开发组可以处理您的问题。

附录二 售后服务与技术支持

凡是通过米尔科技直接购买或经米尔科技授权的正规代理商处购买的米尔科技全系列产品，均可享受以下权益：

- 1、6个月免费保修服务周期
- 2、终身免费技术支持服务
- 3、终身维修服务
- 4、免费享有所购买产品配套的软件升级服务
- 5、免费享有所购买产品配套的软件源代码，以及米尔科技开发的部分软件源代码
- 6、可直接从米尔科技购买主要芯片样品，简单、方便、快速；免去从代理商处购买时，漫长的等待周期
- 7、自购买之日起，即成为米尔科技永久客户，享有再次购买米尔科技任何一款软硬件产品的优惠政策
- 8、OEM/ODM服务

如有以下情况之一，则不享有免费保修服务：

- 1、超过免费保修服务周期
- 2、无产品序列号或无产品有效购买单据
- 3、进液、受潮、发霉或腐蚀
- 4、受撞击、挤压、摔落、刮伤等非产品本身质量问题引起的故障和损坏
- 5、擅自改造硬件、错误上电、错误操作造成的故障和损坏
- 6、由不可抗拒自然因素引起的故障和损坏

产品返修：用户在使用过程中由于产品故障、损坏或其他异常现象，在寄回维修之前，请先致电米尔科技客服部，与工程师进行沟通以确认问题，避免故障判断错误造成不必要的运费损失及周期的耽误。

维修周期：收到返修产品后，我们将即日安排工程师进行检测，我们将在最短的时间内维修或更换并寄回。一般的故障维修周期为3个工作日（自本公司收到物品之日起，不计运输过程时间），由于特殊故障导致无法短期内维修的产品，我们会与用户另行沟通并确认维修周期。

维修费用：在免费保修期内的产品，由于产品质量问题引起的故障，不收任何维修费用；不属于免费保修范围内的故障或损坏，在检测确认问题后，我们将与客户沟通并确认维修费用，我们仅收取元器件材料费，不收取维修服务费；超过保修期限的产品，根据实际损坏的程度来确定收取的元器件材料费和维修服务费。

运输费用：产品正常保修时，用户寄回的运费由用户承担，维修后寄回给用户的费用由我司承担。非正常保修产品来回运费均由用户承担。