

微信扫码支付在 Qt 下的实现

基于 DCP-3000L 嵌入式工控机

AN01010101 V1.00 Date: 2017/6/5

产品应用笔记

类别	内容
关键词	微信支付、Linux、Qt
摘 要	本文介绍了微信扫码支付在 Qt 下的实现

修订历史

版本	日期	原因
V1.00	2017/6/5	创建文档

目 录

1. 适用范围.....	1
2. 原理概述.....	2
3. 技术实现.....	3
3.1 部署 Web 服务器.....	3
3.1.1 安装 nginx 服务器.....	3
3.1.2 安装 php5-fpm.....	3
3.1.3 安装微信支付 SDK.....	4
3.2 客户端程序开发.....	8
3.2.1 qrencode 工具移植.....	8
3.2.2 Qt 应用程序开发.....	8
3.2.3 微信扫码支付完整测试.....	10
4. 参考资料.....	13
5. 免责声明.....	14
附录 A 程序清单 build.sh	15

1. 适用范围

本文基于嵌入式 Linux 平台和 Qt 4.8.6 版本，主要介绍了如何开启实现微信扫码支付，其应用原理及方法同样适用于致远电子同类型工控机。

工控机链接：<http://www.zlg.cn/ipc/ipc/product/id/129.html>

2. 原理概述

微信支付分为公众号支付、APP 支付、扫码支付以及刷卡支付等模式，其中扫码支付适用于线下支付，其支付流程如图 2.1 所示：

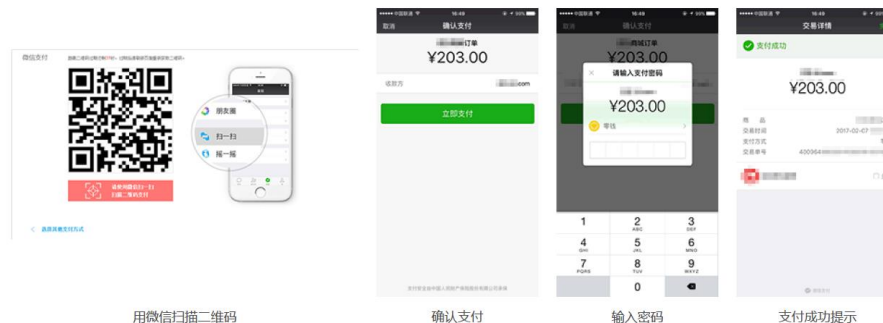


图 2.1 微信扫码支付流程

分析微信扫码支付的流程，本文在嵌入式 Linux 平台上使用 Qt 实现该功能，总流程如图 2.2 所示：

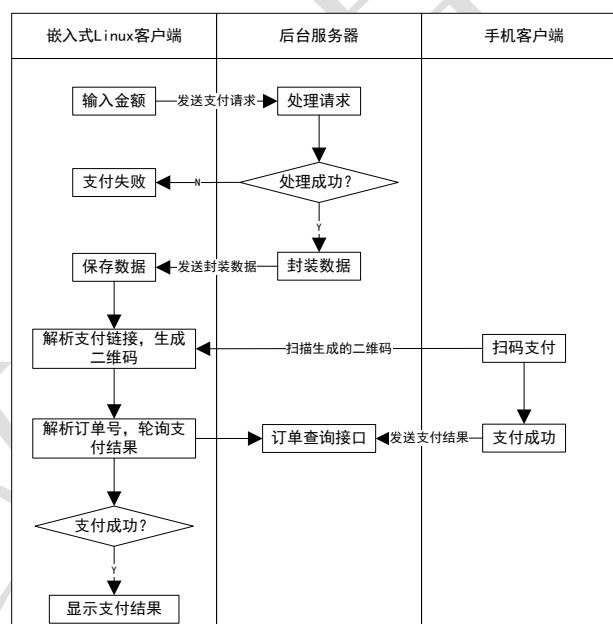


图 2.2 嵌入式 Linux 客户端扫码支付流程

从图 2.2 的支付流程中我们可以看到，微信扫码支付分为三部分，分别为嵌入式 Linux 客户端、后台服务器以及手机客户端，需要实现的有以下三点：

- 服务器部署：本文选择 nginx+php 进行实现，部署在 ubuntu-14.04 系统，需先从微信支付官网下载 php 版本的 SDK，[点击下载](#)；
- 解析支付链接和回调地址：在与服务器的 POST 请求时，可以加入数据分隔符（本文以“###”为分隔符），方便本地解析；
- 生成二维码图片：需借助 qrencode 工具，本文同时移植了 qrencode 及其所依赖的相关库（zlib，libpng），下载的 qrencode 版本为 [3.4.4](#)，zlib 版本为 [1.2.5](#)，libpng 版本为 [1.6.29](#)；

3. 技术实现

3.1 部署 Web 服务器

在 Ubuntu 14.04 系统上部署 nginx 服务器，同时安装微信支付 php 版本的 SDK（下载好的 SDK 名称为 WxpayAPI_php_v3.zip），可以按照以下步骤进行：

3.1.1 安装 nginx 服务器

输入以下命令进行 nginx 的安装：

```
[vmuser@ubuntu: ~]$ sudo apt-get install nginx
```

由于 apache 服务器的监听端口与 nginx 相同，为了不与其冲突，这里我们直接更改 nginx 服务器的配置，将默认的 80 端口修改为 8080：

```
[vmuser@ubuntu: ~]$ vim /etc/nginx/sitesavailable/default
```

```
...
server {
    listen 8080 default_server;
    listen [::]:8080 default_server ipv6only=on;
    ...
}
```

// 修改完成后，需要重启 nginx 服务器

```
[vmuser@ubuntu: ~]$ sudo service nginx restart
```

```
* Restarting nginx nginx [ OK ]
```

接下来在其它主机的浏览器输入服务器地址+端口号（注意修改为服务器实际的 IP 地址），看到如图 3.1 所示界面则说明 nginx 已经安装成功了。

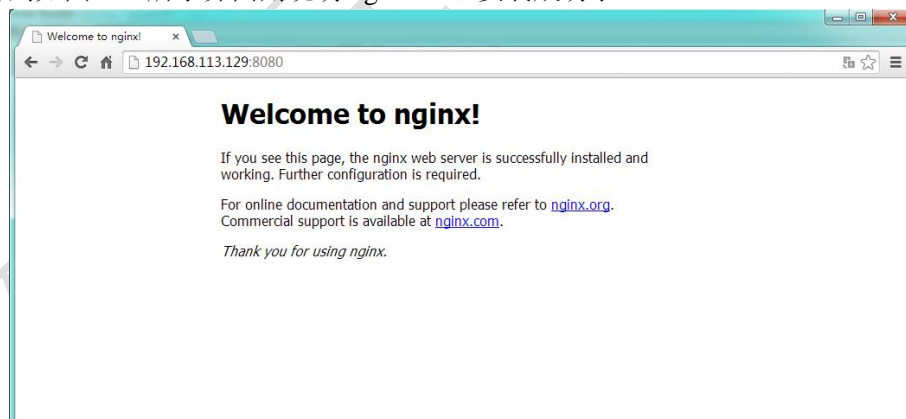


图 3.1 访问 nginx 服务器

3.1.2 安装 php5-fpm

输入以下命令进行 php5-fpm 的安装：

```
[vmuser@ubuntu: ~]$ sudo apt-get install php5-fpm
```

接下来修改 nginx 服务器配置，使其支持 php：

```
[vmuser@ubuntu: ~]$ sudo vim /etc/nginx/sitesavailable/default
```

```
...
root /usr/share/nginx/html;
index index.php index.html index.htm; // 添加 index.php
```

```
...
// 将相关配置注释打开
location ~ \.php$ {
#       fastcgi_split_path_info ^(.+\.php)(/.+)$;
#       # NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini
#
#       # With php5-cgi alone:
#       fastcgi_pass 127.0.0.1:9000;
#       # With php5-fpm:
#       fastcgi_pass unix:/var/run/php5-fpm.sock;
#       fastcgi_index index.php;
#       include fastcgi_params;
}
```

在/usr/share/nginx/html 目录下新建 index.php，重启 php5-fpm 和 nginx 进行测试：

```
[vmuser@ubuntu: ~]$ sudo rm /usr/share/nginx/html/*
[vmuser@ubuntu: ~]$ sudo vi /usr/share/nginx/html/index.php
<?php
phpinfo();
?>
[vmuser@ubuntu: ~]$ sudo /etc/init.d/php5-fpm restart
[vmuser@ubuntu: ~]$ sudo service nginx restart
```

同样地在浏览器输入 nginx 服务器的地址加端口号，可以看到如图 3.2 所示的界面，此时已经完成了 nginx 和 php 的安装：



图 3.2 nginx 服务器支持 php5-fpm

3.1.3 安装微信支付 SDK

将下载好的微信支付 php 版本的 SDK 进行解压，并将相关文件拷贝到 nginx 服务器根目录（默认为/usr/share/nginx/html，可通过修改/etc/nginx/sitesavailable/default 的 root 参

数进行指定，这里我们使用默认参数)：

```
[vmuser@ubuntu: ~]$ unzip WxpayAPI_php_v3.zip
[vmuser@ubuntu: ~]$ ls WxpayAPI_php_v3
cert doc example image index.php lib logs
[vmuser@ubuntu: ~]$ sudo cp WxpayAPI_php_v3/* /usr/share/nginx/html/ -r
```

打开浏览器，输入服务器地址，可以看到微信支付 SDK 已经成功运行起来了，如图 3.3 所示：



图 3.3 微信支付 php 版本 SDK 主页面

此时点击任意方块可以跳转至微信支付 SDK 的示例网址，本文完成本地扫码支付只需要用到微信支付 SDK 中的扫码支付接口以及查询接口，接下来对相关文件进行修改：

1) 修改/example/native.php:

```
[vmuser@ubuntu: ~]$ sudo vi /usr/share/nginx/html/example/native.php
<?php
ini_set('date.timezone','Asia/Shanghai');
//error_reporting(E_ERROR);

require_once "../lib/WxPay.Api.php";
require_once "WxPay.NativePay.php";
require_once 'log.php';

$notify = new NativePay();

$input = new WxPayUnifiedOrder();
$input->SetBody("test");
$input->SetAttach("test");
// 保存订单号码
$order_num = WxPayConfig::MCHID.date("YmdHis");
$input->SetOut_trade_no($order_num);
// 获取输入金额
$money = $_POST['money'];
if (empty($money)) {
    $money = "1";
}
```



```
$input->SetTotal_fee($money);
$input->SetTime_start(date("YmdHis"));
$input->SetTime_expire(date("YmdHis", time() + 600));
$input->SetGoods_tag("test");
// 设置支付回调地址，这里直接使用微信自带的回调地址
$input->SetNotify_url("http://paysdk.weixin.qq.com/example/notify.php");
$input->SetTrade_type("NATIVE");
$input->SetProduct_id("123456789");
$result = $notify->GetPayUrl($input);
// 获取支付链接
$url = $result["code_url"];
if (!empty(url)) {
    // 使用###作为分隔符
    echo $order_num . "###" . $url;
}
?>
```

2) 修改/example/orderquery.php:

```
[vmuser@ubuntu: ~]$ sudo vi /usr/share/nginx/html/example/orderquery.php
<?php
ini_set('date.timezone','Asia/Shanghai');
error_reporting(E_ERROR);
require_once "../lib/WxPay.Api.php";
require_once 'log.php';

//初始化日志
$logHandler= new CLogFileHandler("./logs/" . date('Y-m-d') . '.log');
$log = Log::Init($logHandler, 15);

if(isset($_REQUEST["transaction_id"]) && $_REQUEST["transaction_id"] != ""){
    $transaction_id = $_REQUEST["transaction_id"];
    $input = new WxPayOrderQuery();
    $input->SetTransaction_id($transaction_id);
    // printf_info(WxPayApi::orderQuery($input));
    $result = WxPayApi::orderQuery($input);
    // 返回支付结果
    echo $result['trade_state'];
    exit();
}

if(isset($_REQUEST["out_trade_no"]) && $_REQUEST["out_trade_no"] != ""){
    $out_trade_no = $_REQUEST["out_trade_no"];
    $input = new WxPayOrderQuery();
    $input->SetOut_trade_no($out_trade_no);
    // printf_info(WxPayApi::orderQuery($input));
```

```
$result = WxPayApi::orderQuery($input);  
// 返回支付结果  
echo $result['trade_state'];  
exit();  
}  
?>
```

3) 修改/lib/WxPay.Api.php, 定位到 538 行, 将 TRUE 更改为 FALSE:

```
[vmuser@ubuntu: ~]$ sudo vi /usr/share/nginx/html/lib/WxPay.Api.php  
...  
private static function postXmlCurl($xml, $url, $useCert = false, $second = 30)  
{  
    $ch = curl_init();  
    //设置超时  
    curl_setopt($ch, CURLOPT_TIMEOUT, $second);  
  
    //如果有配置代理这里就设置代理  
    if(WxPayConfig::CURL_PROXY_HOST != "0.0.0.0"  
        && WxPayConfig::CURL_PROXY_PORT != 0){  
        curl_setopt($ch, CURLOPT_PROXY, WxPayConfig::CURL_PROXY_HOST);  
        curl_setopt($ch, CURLOPT_PROXYPORT, WxPayConfig::CURL_PROXY_PORT);  
    }  
    curl_setopt($ch, CURLOPT_URL, $url);  
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);  
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2); //严格校验  
    //设置 header  
    curl_setopt($ch, CURLOPT_HEADER, FALSE);  
    //要求结果为字符串且输出到屏幕上  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);  
    ...
```

4) 修改/lib/WxPay.Config.php, 将微信支付商家相关信息填入 (在申请微信支付成功后, 可从微信服务号后台登录获取, 这里不赘述):

```
class WxPayConfig  
{  
    //=====【基本信息设置】=====  
    //  
    /**  
     * TODO: 修改这里配置为您自己申请的商户信息  
     * 微信公众号信息配置  
     *  
     * APPID: 绑定支付的 APPID (必须配置, 开户邮件中可查看)  
     *  
     * MCHID: 商户号 (必须配置, 开户邮件中可查看)  
     *  
     * KEY: 商户支付密钥, 参考开户邮件设置 (必须配置, 登录商户平台自行设置)
```

```

* 设置地址: https://pay.weixin.qq.com/index.php/account/api_cert
*
* APPSECRET: 公众帐号 secert (仅 JSAPI 支付的时候需要配置, 登录公众平台, 进入开发者中心
可设置),
* 获取地址:
https://mp.weixin.qq.com/advanced/advanced?action=dev&t=advanced/dev&token=2005451881&lang=zh_CN
* @var string
*/
const APPID = 'wx426b3015555a46be';
const MCHID = '1900009851';
const KEY = '8934e7d15453e97507ef794cf7b0519d';
const APPSECRET = '7813490da6f1265e4901ffb80afaa36f';
...

```

至此, 我们已经完成了服务器的部署以及微信 SDK 的修改安装, 接下来进行客户端应用程序的开发。

3.2 客户端程序开发

3.2.1 qrencode 工具移植

为了方便生成二维码图片, 接下来需要移植 qrencode 工具, 提前下载好依赖库以及源码, 下载的 qrencode 版本为 3.4.4, zlib 版本为 1.2.5, libpng 版本为 1.6.29。

为了方便移植, 我们新建一个目录进行 qrencode 的交叉编译, 并将下载好的源码包放到该目录中, 假设我们新建的目录为 qrencode-dev, 用 ls 命令查看目录内容的结果如下:

```

[vmuser@ubuntu: ~]$ ls qrencode-dev/
libpng-1.6.29.tar.gz  qrencode-3.4.4.tar.gz  zlib-1.2.5.tar.bz2

```

在该目录下建立一个编译脚本, 脚本内容见附录 A 程序清单 build.sh, 执行结果如下:

```

[vmuser@ubuntu: ~]$ cd qrencode-dev/
[vmuser@ubuntu: ~/qrencode-dev]$ chmod +x build.sh
[vmuser@ubuntu: ~/qrencode-dev]$ ./build.sh
...
All have done, copy the output file qrcode-dev.tar.gz to your board.
[vmuser@ubuntu: ~/qrencode-dev]$ ls
build.sh      libpng-1.6.29.tar.gz  qrencode-3.4.4.tar.gz
install       qrcode-dev.tar.gz     zlib-1.2.5
libpng-1.6.29 qrencode-3.4.4        zlib-1.2.5.tar.bz2

```

编译完成后, 我们得到了编译好的压缩包 qrcode-dev.tar.gz, 接下来进行 Qt 应用程序的开发。

3.2.2 Qt 应用程序开发

关于 Qt 的应用程序开发, 本文主要集中在与服务器的通信上, 对于本地 UI 界面的处理不做过多介绍, 下面依次分析需要实现的四个关键功能:

1) HTTP POST 请求处理:

我们需要与服务器进行通信, 这里我们选择使用 POST 方式发送数据到服务器, 需要用到 QNetworkRequest 模块, 下面将 post 请求封装成函数, 方便调用:

```
#include <QNetworkAccessManager>
```

```
#include <QNetworkReply>
#include <QNetworkRequest>
#include <QString>
#include <QEventLoop>
#include <QByteArray>
#include <QVariant>

//Post http request
/*
    Example: Post("http://google.com/", "header1=one&header2=two");
*/
QString Post(QString uri, QString header)
{
    QString result;
    QEventLoop eventLoop;
    QNetworkAccessManager manager;
    QUrl url(uri);
    QNetworkRequest request(url);
    request.setHeader(QNetworkRequest::ContentTypeHeader, "application/x-www-form-urlencoded");
    QObject::connect(&manager, SIGNAL(finished(QNetworkReply*)), &eventLoop, SLOT(quit()));
    QNetworkReply *reply = manager.post(request, header.toUtf8());
    eventLoop.exec();
    result = reply->readAll();
    return result;
}
```

通过调用 Post 函数，可以实现与服务器的 POST 请求通信。

2) 解析服务器数据，保存支付链接和商家订单号：

```
void MainWindow::wx_handler() {
    ...
    QString pay_money = "money=" + money;
    /* 服务器地址需根据实际情况指定 */
    QString post_ret = Post("http://192.168.1.164:8080/example/native.php", pay_money);
    /* 以 ### 分隔服务器返回数据 */
    QStringList strlist = post_ret.split("###");
    this->wx_query_url = "out_trade_no=" + strlist.at(0);
    this->wx_pay_url = strlist.at(1);
    ...
}
```

3) 根据支付链接生成二维码图片，显示在 UI 界面上：

```
void MainWindow::wx_handler() {
    ...
    /* 使用 3.2.1 交叉编译生成的 qrencode 工具生成二维码 */
    QString cmd = "qrencode " + wx_pay_url + " -s 6 -o /opt/qrcode.png";
    system(cmd.toAscii().data());
}
```

```
system("sync");

/* 将生成的二维码显示在 UI 界面上 */
QPixmap pixmap_qrcode("/opt/qrcode.png");
ui->qrcode_img->setGeometry(200, 40, 400, 400);
ui->qrcode_img->setPixmap(pixmap_qrcode);
ui->qrcode_img->setVisible(true);
...
}
```

4) 根据商家订单号轮询服务器，获得支付结果：

```
void MainWindow::wx_handler() {
    ...
    /* 服务器地址需根据实际情况指定 */
    QString pay_status = Post("http://192.168.1.164:8080/example/orderquery.php", wx_query_url).toUpper();
    qDebug() << "pay_status " << pay_status;
    if (pay_status.compare("SUCCESS") == 0) {
        /* 支付成功，显示支付成功图片 */
        this->ui->label_success->setVisible(true);
    }
    /* 轮询时间为 3s 一次，不可过于频繁 */
    this->timer->start(3000);
    ...
}
```

以上是 Qt 应用程序四个关键功能的具体实现，UI 界面的处理可在此基础上自行添加，接下来需要进行的是应用软件在嵌入式平台上的部署和测试。

3.2.3 微信扫码支付完整测试

在测试之前，需要按照 3.1 小节中的步骤先将微信 SDK 部署到服务器上，然后按照以下步骤进行测试：

1) 部署二维码生成工具：

将 3.2.1 中生成的 qrcode-dev.tar.gz 拷贝到嵌入式工控机的文件系统中，这里拷贝到/opt目录下，按照以下命令进行解压和环境变量的设置：

```
[root@DCP-3000L opt]# tar xf qrcode-dev.tar.gz
[root@DCP-3000L opt]# export LD_LIBRARY_PATH= $LD_LIBRARY_PATH: /opt/qrcode-dev/lib
[root@DCP-3000L opt]# export PATH=$PATH:/opt/qrcode-dev/bin
[root@DCP-3000L opt]# qrencode "http://www.zlg.cn" -o qr-test.png
```

qrencode 命令没有报错则说明部署完成，且生成的 qr-test.png 图片可以用手机进行扫码，将会跳转至致远电子首页。

2) 启动应用程序：

我们在 3.2 中生成的 Qt 应用程序名称为 qt-wxpay，实现了以下功能：

- 键盘输入金额，以分为单位；
- 请求服务器，生成二维码支付链接；
- 扫码倒计时处理；
- 支付结果查询及显示。

将 qt-wxpay 拷贝到工控机文件系统的/opt目录下，使用以下命令进行程序的启动：

```
[root@DCP-3000L ~] cd /opt
[root@DCP-3000L opt]# ./qt-wxpay -qws
```

应用初始化界面如图 3.4 所示：

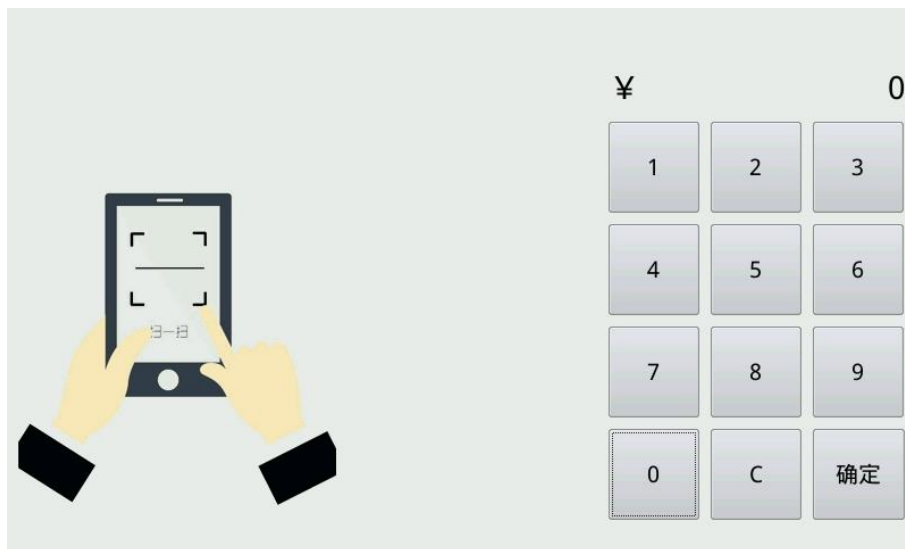


图 3.4 Qt 应用程序界面

3) 发送支付请求，生成二维码：

点击右侧键盘输入具体金额，点击确定发送支付请求，服务器将返回支付链接，本地应用程序将支付链接处理成二维码图片并进行显示，此时请求结束完成的界面如图 3.5 所示：



图 3.5 本地二维码页面

4) 手机扫码支付：

使用手机微信客户端的“扫一扫”功能，扫描界面生成的二维码，根据支付页面提示输入支付密码，支付完成界面如图 3.6 所示：



支付成功

ZLG致远电力

¥ 0.08

完成

□ 致远电力

图 3.6 手机客户端扫码支付结果

5) 支付完成结果:

在第三步生成二维码后，客户端会根据商家订单号轮询服务器，当完成第四步的手机扫码支付后，服务器将会回复“SUCCESS”字段，此时我们可以确认支付已经完成，从而进行相应处理，支付完成界面如图 3.7 所示：



图 3.7 支付完成界面

至此，微信扫码支付在 Qt 下的实现已经完成，如需完整的服务器代码示例以及客户端应用程序示例，请向我司技术支持申请。

4. 参考资料

微信扫码支付接入流程指引.pdf

5. 免责声明

广州致远电子有限公司所提供的服务内容旨在协助客户加速产品的研发进度,在服务过程中所提供的任何程序、文档、测试结果、方案、支持等资料和信息,都仅供参考,客户有权不使用或自行参考修改,本公司不提供任何的完整性、可靠性等保证,若在客户使用过程中因任何原因造成的特别的、偶然的或间接的损失,本公司不承担任何责任。

附录 A 程序清单 build.sh

```
#!/bin/sh

# 交叉编译自动化脚本
# 请自行指定交叉编译工具链
HOST=arm-none-linux-gnueabi
CROSS_COMPILE=arm-none-linux-gnueabi-
PREFIX=${PWD}/install

if [ ! -d install ]; then
    mkdir install
fi

sync

ZLIB_PKG=`ls *zlib* | grep tar`
if [ $ZLIB_PKG == "" ]; then
    echo "qrencode packets no found"
    exit
fi

PNG_PKG=`ls *png* | grep tar`
if [ $PNG_PKG == "" ]; then
    echo "libpng packets no found"
    exit
fi

QRCODE_PKG=`ls *qrencode* | grep tar`
if [ $QRCODE_PKG == "" ]; then
    echo "qrencode packets no found"
    exit
fi

ZLIB_PKG_PATH=`echo $ZLIB_PKG | awk -F '.tar' '{print $1}'`
PNG_PKG_PATH=`echo $PNG_PKG | awk -F '.tar' '{print $1}'`
QRCODE_PKG_PATH=`echo $QRCODE_PKG | awk -F '.tar' '{print $1}'`

if [ -d $ZLIB_PKG_PATH ]; then
    rm $ZLIB_PKG_PATH -r
fi

if [ -d $PNG_PKG_PATH ]; then
```

```
rm $PNG_PKG_PATH -r
fi

if [ -d $QRCODE_PKG_PATH ]; then
    rm $QRCODE_PKG_PATH -r
fi

sync

tar xf $ZLIB_PKG
tar xf $PNG_PKG
tar xf $QRCODE_PKG

sync

sleep 1


echo ""
echo ">>>>>>>>>>> Start compile zlib"
cd $ZLIB_PKG_PATH
CC=${CROSS_COMPILE}gcc ./configure --prefix=${PWD}/../install/
sync
sleep 1
make
sync
sleep 1
make install
cd ..
if [ -f install/lib/libz.so ]; then
    echo ">>>>>>>>>>>>>>>>>>>"
    echo "zlib compile success"
    echo ">>>>>>>>>>>>>>>>>>>"
else
    echo "zlib compile failed"
    exit
fi

sync

sleep 1


echo ""
echo ">>>>>>>>>>> Start compile libpng"
cd $PNG_PKG_PATH
cp ../install/include/*.h ./
./configure --prefix=${PWD}/../install/ --host=${HOST} CC=${CROSS_COMPILE}gcc
CFLAGS=-I${PWD}/../install/include LDFLAGS=-L${PWD}/../install/lib
sync
```

©2017 Guangzhou ZHIYUAN Electronics Co., Ltd.

销售与服务网络

广州致远电子有限公司

地址：广州市天河区车陂路黄洲工业区 7 栋 2 楼

邮编：510660

网址：www.zlg.cn

全国销售与服务电话：400-888-4005



全国服务电话：400-888-4005

销售与服务网络：

广州总公司

广州市天河区车陂路黄洲工业区 7 栋 2 楼

电话：(020)28267985 22644261

上海分公司：上海

上海市北京东路 668 号科技京城东楼 12E 室

电话：(021)53865521 53083451

北京分公司

北京市海淀区知春路 108 号豪景大厦 A 座 19 层

电话：(010)62536178 62635573

上海分公司：南京

南京市珠江路 280 号珠江大厦 1501 室

电话：(025)68123923 68123920

深圳分公司

深圳市福田区深南中路 2072 号电子大厦 12 楼

电话：(0755)83640169 83783155

上海分公司：杭州

杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)89719491 89719493

武汉分公司

武汉市洪山区广埠屯珞瑜路 158 号 12128 室（华中电脑数码市场）

电话：(027)87168497 87168397

重庆分公司

重庆市九龙坡区石桥铺科园一路二号大西洋国际大厦（赛格电子市场）2705 室

电话：(023)68796438 68797619

成都分公司

成都市一环路南二段 1 号数码科技大厦 403 室

电话：(028)85439836 85432683

西安办事处

西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881295 87881296

请您用以上方式联系我们，我们会为您安排样机现场演示，感谢您对我公司产品的关注！