

# BE230E1 Software Programming Guide

Rev. 1.0



© 2016-2017 Bolymin, Inc. All Rights Reserved.

Version	Note	Date
0.1	Initial version	2016/10/25
0.5	Add Bolymin API definitions	2016/11/15
1.0	1 <sup>st</sup> DVT release	2016/12/05

# **Table of Content**

1 Setup Software Development Environment	5
1.1 Linux Host Machine for Software Development	5
1.2 Install SDK on Linux Host	6
Prerequisite	
Download SDK/Toolchain	
Use SDK/Toolchain	
1.3 Connect BE230E1 Device to PC	7
2 Test Apps for BE230E1	9
2.1 GPIO Test	10
2.1.1 GPIO Pins Definition	
2.1.2 Bolymin GPIO API	11
2.1.3 GUI	
2.2 ADC Test	13
2.2.1 ADC Channels Definition	
2.2.2 Bolymin ADC API	
2.2.3 GUI	
2.3 UART Test	15
2.3.1 UART Port Definition	
2.3.2 Qt Serial Port API	
2.3.2 GUI	
2.4 Bluetooth Test	16
2.4.1 Qt Bluetooth API	
2.4.2 GUI	
2.5 WLAN Test	
2.5.1 Sample Code Source	
2.5.2 GUI	
2.6 LAN Test	
2.6.1 Description	
2.6.2 GUI	
2.7 LCD Test	20
2.7.1 QML	
2.7.2 GUI	
2.8 Backlight Control	21
2.8.1 Backlight Brightness Value	

2.8.2 Bolymin Backlight API	
2.8.3 GUI	
2.9 Touch Panel Test	22
2.9.1 Qt C++ Class	
2.9.2 GUI	
2.10 Speaker Test	23
2.10.1 Qt C++ Class	
2.10.2 GUI	

### **1.1 Linux Host Machine for Software Development**

To do Linux development with the SDK for BE230E1 embedded product, you'll need a host PC running Linux OS. The Linux host is generally much faster and has a lot more memory (both RAM and hard disk space) than the typical embedded system so that it's usually used to install specific cross compiler to build executable programs for target device. There're many choices of Linux distributions for your Linux PC and we choose 64-bit Ubuntu 14.04 LTS (or later version), running either natively or in a Virtual Machine. Note that **it's necessary to use 64-bit Linux OS** to develop application programs on BE230E1 device.



# **1.2 Install SDK on Linux Host**

### Prerequisite

Before using toolchain, please check if some dependent packages are installed in your Linux Host, or install them by

```
$sudo apt-get install xinetd tftpd nfs-kernel-server minicom
build-essential libncurses5-dev autoconf automake
```

### **Download SDK/Toolchain**

Get the SDK and tool chain from Bolymin <u>download link</u>. Use this tool chain to build executable programs for BE230E1 Linux platform.

After downloading the SDK/toolchain file to your Linux host, unzip and untar this tar.bz2 file by

```
$tar xvfj be230e1_toolchain.tar.bz2
```

to be located at some path you assigned.

/incent@rdsw02 ~]\$ ls -al be230e1_toolchain/	
otal 44	
wxr-xr-x 4 vincent vincent 4096 Nov 16 22:17	
wx 10 vincent vincent 4096 Nov 7 17:28	
wxrwxr-x 11 vincent vincent 4096 Jul 5 15:28 cortexa8t2hf-vfp-neon-linux-gnueabi_with_qt5.6	
wxrwxr-x 1 vincent vincent 2964 Aug 17 22:15 environment-setup	
wxrwxr-x 1 vincent vincent 22616 Aug 16 22:38 site-config-cortexa8t2hf-vfp-neon-linux-gnueabi_with_qt5.	6
wxr-xr-x 8 vincent vincent 4096 Aug 16 23:25 toolchain_linaro_4.7	
vincent@rdsw02 ~]\$	

### Use SDK/Toolchain

Before starting to build your applications, please execute **environment-setup** script file to setup/export several environment variables for cross compiler/linker or Qt libraries.



# **1.3 Connect BE230E1 Device to PC**

#### 0. Get IP address of LAN port

Assume there's DHCP server in your network environment, you can get current IP address of LAN port after plugging RJ45 cable into BE230E1 and then run 'System Info' test app.

÷	BE230E1 System Information	
CPU	ARMv7 Processor rev 2 (v7l)	
RAM (usage)	MemTotal:         509552 kB         2           MemFree:         346484 kB         2           MemAvailable:         459428 kB         2	•
ROM (usage)	NAMEMAJ:MIN RMSIZE RO TYPE MOUNTPOINTmmcblk1boot0179:2402M1 diskmmcblk1boot1179:3202M1 diskmmcblk1boot1170:0027C0 disk	-
OS	Linux be230e 3.14.26 #1.02 Tue Oct 11 13:44:35 CST 2016 armv7l GNU/Linux	_
IP Address (eth0)	192.168.1.117	
Netmask	255.255.255.0	
Broadcast	192.168.1.255	_
Date/Time	2016/11/17 01:58:54 (UTC)	

#### 1. Telnet connection (thru LAN or Wi-Fi)

The BE230E1 device has a LAN and WLAN interface so that you can do remote connection to login the device thru network by some telnet client software like <u>SSH</u>.



Default login account is "root", password is "root".

#### 2. Put your app or files to the BE230E1 device

To transfer your built Apps or other files to the BE230E1 device, the easy and fast way is to use the file transfer software such as <u>SFTP</u> or <u>SCP</u>. Thus you could transfer files between PC (Windows or Linux) and BE230E1 target device.

77.2 GiB free out of	98.7 Gi	B (78%) on	/[(ext4)]		E.					
home/vincent			·· ,	/ 🗇 🙀					. /	. ⊙+ €
Name 🔻	Ext	Size	Modified	rw)	Name	~	Ext	Size	Modified	rv
<b>}</b>		< DI	3>							
.cache		6	) SFTP Login							
.compiz			111							
.compiz-1			21							
.config		0	Dease er	ter vour u	sername and r	assword				
dbus.			- ricuse er	icer your u	sernanc and p	/0330/010.				
.gconf		si	te: sftp://19	2.168.1.1	84:22					
.gnome2			sername: root							
.gstreamer-0.10		<u> </u>								
.gvfs		P	assword							
.installjammerinfo										
kde		4			Cance	♥0	к			
local		1								
.mission-control		<di< td=""><td>R&gt; 11/11/14 10:06</td><td>AM rwx</td><td></td><td></td><td></td><td></td><td></td><td></td></di<>	R> 11/11/14 10:06	AM rwx						
.mozilla		<dii< td=""><td>R&gt; 11/11/14 11:26</td><td>AM rwx</td><td></td><td></td><td></td><td></td><td></td><td></td></dii<>	R> 11/11/14 11:26	AM rwx						
.node-gyp		<dii< td=""><td>R&gt; 11/27/14 02:43</td><td>PM rwx 🕃</td><td>L</td><td></td><td></td><td></td><td></td><td>_</td></dii<>	R> 11/27/14 02:43	PM rwx 🕃	L					_
out of 48, 0 B (0) out o	f 155.9	KiB (159,66	50)		>> Reading: C	pening SFT	o conne	ction to ho	st 192.168.1.184:2	2 🥝 /
vincent				20		/				C
				1.000		252				-

Login account is "root", password is "root".

# 2 Test Apps for BE230E1

On this BE230E1 device, there are several test apps written in Qt/C++ and/or QML/QtQuick that can be used to simply verify the functionalities of corresponding hardware module or interfaces. In addition, we've offered Qt 4.6 framework porting on the board and it could be a choice for you to write your own GUI application on BE230E1 device.

Test App Items	Description
GPIO	To verify 8 GPIO ports functionality. Written in Qt/C++ code.
ADC	To verify 4 ADC channels functionality. Written in Qt/C++ code.
	To verify UART functionality, used for RS232/RS485 connection. Written in
UART	Qt/C++ code.
Dhustsst	To verify Bluetooth basic functions, like scanning and pairing/unpairing.
Didelootii	Written in Qt/C++ code.
WLAN	To verify WLAN functionality. Written in Qt/C++ code.
LAN	To verify accessing network thru LAN port. Written in Qt/C++ code.
LCD	To verify LCD display. Written in Qt/C++ and QML/QtQuick code.
Backlight	To verify screen backlight brightness. Written in Qt/C++ and QML/QtQuick
Touch Panel	To verify touch panel functionality. Written in Qt/C++ code.
Speaker	To verify speaker functionality. Written in Qt/C++ code.



# 2.1 GPIO Test

This test app is used to test built-in 8 GPIO pins on BE230E1 device. Users can change both the direction (either input or output) and value (either high or low) of these 8 pins by API functions provided by Bolymin. Developers could refer to our sample code for more details.

### 2.1.1 GPIO Pins Definition

The value of 8 GPIO pins is located in below list. It could only set as "**0**" (meaning LOW) or "**1**" (meaning HIGH). The default for all pins is "**1**" after reboot.

```
static QString const SYSFS_GPIO_PINS_VALUE[] =
                     {
                      "/sys/class/gpio/gpio48/value",
                                                       // Pin 1
                      "/sys/class/gpio/gpio7/value",
                                                        // Pin 2
                      "/sys/class/gpio/gpio53/value",
                                                        // Pin 3
                      "/sys/class/gpio/gpio54/value",
                                                        // Pin 4
                                                        // Pin 5
                      "/sys/class/gpio/gpio67/value",
                      "/sys/class/gpio/gpio61/value",
                                                       // Pin 6
                      "/sys/class/gpio/gpio68/value",
                                                        // Pin 7
                      "/sys/class/gpio/gpio66/value"
                                                        // Pin 8
                     };
```

The direction of 8 GPIO pins is located in below list. It could only set as "**in**" or "**out**". The default for all pins is "**in**" after reboot.

```
static QString const SYSFS_GPIO_PINS_DIRECTION[] =
                     {
                      "/sys/class/gpio/gpio48/direction",
                                                            // Pin 1
                      "/sys/class/gpio/gpio7/direction",
                                                           // Pin 2
                      "/sys/class/gpio/gpio53/direction",
                                                           // Pin 3
                      "/sys/class/gpio/gpio54/direction",
                                                           // Pin 4
                      "/sys/class/gpio/gpio67/direction",
                                                           // Pin 5
                      "/sys/class/gpio/gpio61/direction",
                                                            // Pin 6
                      "/sys/class/gpio/gpio68/direction",
                                                            // Pin 7
                      "/sys/class/gpio/gpio66/direction"
                                                            // Pin 8
                     };
```

### 2.1.2 Bolymin GPIO API

Source: gpioaccess.c Header: gpioaccess.h

Function Syntax	int GPIO_Get_Pin_Direction(int pinNum)			
Description	To get the GPIO pin direction by assigned pin number.			
Parameters	pinNum	Pin number, between 1 and 8.		
	1	It means input direction		
Return Value	2	It means output direction		
	0	Undefined		

Function Syntax	<pre>void GPIO_Set_Pin_Direction(int pinNum, bool bInput)</pre>			
Description	To set the GPIO pin direction by assigned pin number.			
	pinNum	Pin number, between 1 and 8.		
Parameters	1.1.	'true' is for input direction;		
	omput	'false' is for output direction		
Return Value	none			

Function Syntax         int GPIO_Get_Pin_Value(int pinNum)				
Description	To get the GPIO pin value by assigned pin number.			
Parameters	pinNum	Pin number, between 1 and 8.		
	0	It means 'Low' state		
<b>Return Value</b>	1	It means 'High' state		
	others	Undefined state		

Function Syntax	void GPIO_Set_Pin_Value(int pinNum, bool bPullLow)			
Description	To set the GPIO pin value by assigned pin number and			
	boolean type value to set low or high.			
	pinNum	Pin number, between 1 and 8.		
Parameters	1-D111	'true' means Low state		
	OPUILOW	'false' means High state		
Return Value	none			

### 2.1.3 GUI

4	BE230	DE1 GPIO T	ſest	
PIN NUM	DIRECTIO	N STATE	DIRECTIO	N TRIGGER
GPIO 1	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 2	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 3	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 4	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 5	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 6	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 7	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
GPIO 8	Input	🔘 High	<ul> <li>Output</li> </ul>	🗆 High
		EXIT		

## 2.2 ADC Test

This test app is used to test the four ADC channels of BE230E1. The resolution of all ADC channels is **12-bit** and each value range is from 0 volt to 3.3 volt.

### 2.2.1 ADC Channels Definition

```
static QString const SYSFS_ADC_CHANNELS[] =
{
    "/sys/bus/iio/devices/iio\:device0/in_voltage4_raw", // ADC CH1
    "/sys/bus/iio/devices/iio\:device0/in_voltage5_raw", // ADC CH2
    "/sys/bus/iio/devices/iio\:device0/in_voltage6_raw", // ADC CH3
    "/sys/bus/iio/devices/iio\:device0/in_voltage7_raw", // ADC CH4
};
```

#### 2.2.2 Bolymin ADC API

Source: adcreader.c Header: adcreader.h

Function Syntax	float Read_Adc_Value(unsigned int chNum)	
Description	To read current ADC value by assigned channel number.	
Parameters	chNum Channel number, from 1 to 4	
Return Value	ADC value in voltage, ranged from 0 ~ 3.3 volt.	

#### 2.2.3 GUI



### 2.3 UART Test

This test app is modified from Qt's <u>Terminal sample</u> of Serial Port Example, which shows the main features of the QSerialPort class, like configuration, I/O implementation and so forth.

### 2.3.1 UART Port Definition

On BE230E1, UART ports can be used for RS232 or RS485. Please see the user manual for the detailed pin definition of Multi I/O cable.

Device Name	Description
/dev/ttyO2	For RS232
/dev/ttyO4	For RS422/RS485 (selected by cable)

#### 2.3.2 Qt Serial Port API

Please refer to <u>Qt Serial Port C++ Classes</u> for more details.

#### 2.3.2 GUI

Calls Tools Sen	d			
🌜 😳 🔊 🧭	Select Serial Port	Select Param	eters	
	ttyO2 Description: N/A Manufacturer: N/A Serial number: N/A	BaudRate: Data bits: Parity:	9600 ▼ 8 ▼ None ▼	
	Location: /dev/ttyO2 Vendor Identifier: N/A Product Identifier: N/A	Stop bits: Flow control:	1   •     None   •	
	Additional options ☑ Local echo			
			Apply	

# 2.4 Bluetooth Test

This test app is modified from Qt's <u>Bluetooth Scanner</u> example. It could be used to scan the available Bluetooth devices nearby and do pairing/unpairing to assigned remote device.

#### 2.4.1 Qt Bluetooth API

This example utilizes several Qt Class of Bluetooth related like <u>QBluetoothDeviceInfo</u>, <u>QBluetoothLocalDevice</u>, and <u>QBluetoothDeviceDiscoveryAgent</u>.

#### 2.4.2 GUI

BE230E1 Bluetooth Test		
C0:18:85:DB:CB:9D N16_001		
14:DD:A9:A0:05:E2 ASUS_Z00AD		
E4:CE:8F:32:50:B3		
E4:CE:8F:32:50:B3 Apple Bluetooth Device		
Local Device		
E Blucketh Brunned On	Disease bla	
Bluetooth Powered On	M Discoverable	
🗹 General Unlimited Inquiry		
Scan Clear	Show Service	Quit

BE230E1 Bluetooth	Test			
C0:18:85:DB:CB:9D	116 001			
14:DD:A9:A0:05:E2 E4:CE:8F:32:50:B3 E4:CE:8F:32:50:B3 /	Object Push Audio Source Audio/Video Remote Control Target Audio/Video Remote Control Headset AG Personal Area Networking (PANU) Personal Area Networking (NAP) Hands-Free AG Phonebook Access PSE Message Access Server Device Identification Generic Access			
			Close	
-Local Device			-	
Bluetooth Powered	l On	☑ Discovera	able	
☑ General Unlimited In	nquiry			
Scan	Clear	Show S	Service	Quit

### 2.5 WLAN Test

This example can be used to verify the wireless LAN functionality of BE230E1.

#### 2.5.1 Sample Code Source

This test app is the front-end GUI part (wpa\_gui-qt4) from <u>wpa\_supplicant</u> and we've modified it only for simple testing.

#### 2.5.2 GUI

Current Wi-Fi Status	Manage Netork
Status: Completed (station)	0: bolymin_guest
Last messa - Connection to 00:1d:aa:81:9d:8	
Authentical NONE	
Encryption: NONE	
SSID: bolymin_guest	
BSSID: 00:1d:aa:81:9d:8d	
IP address:	
dapter: wlan0 Connect	© Enabled Add Remove
Jetwork: 0: bolymin guest 💌 Disconnect	O Disabled Edit Scan
} E×i	t

## 2.6 LAN Test

#### 2.6.1 Description

This test app uses a QWebView class acting as a web browser to access network.

### 2.6.2 GUI



### 2.7 LCD Test

This app is used to test the LCD screen by showing different colors (Red, Green, Blue, Black, and White) sequentially.

#### 2.7.1 QML

This app is written in <u>QML</u> primarily. Please find main.qml in LcdTest source example for more details.

#### 2.7.2 GUI

Ç;	BE230E1 LCD Test	
	change color automatica	lly
	start	

# 2.8 Backlight Control

This app is used to control the screen backlight brightness ranged from 1 to 8.

#### 2.8.1 Backlight Brightness Value

The exposed Linux kernel parameter to control brightness value is located at "/sys/class/backlight/backlight.8/brightness".

The available configured value is between 0 and 8. Note that the screen would be full black if you set brightness to be 0.

#### 2.8.2 Bolymin Backlight API

Function Syntax	int getBrightness()
Description	Get backlight brightness value
Parameters	none
Return Value	brightness value : between 1 and 8

Function Syntax	void setBrightness(int value)	
Description	Set backlight brightness value	
Parameters	value Between 1 and 8	
Return Value	none	

#### 2.8.3 GUI



# 2.9 Touch Panel Test

This test app is modified from Qt's <u>Scribble Example</u> and could be used to draw lines while user's figure is pressing and moving on the screen.

#### 2.9.1 Qt C++ Class

This example utilizes <u>QTouchEvent</u> related class of Qt to detect user's touch motion on the screen. Programmer could visit Qt's website to see more detailed information.

#### 2.9.2 GUI

Dptions	
<u>C</u> lear Screen	
E <u>x</u> it	
	Server and the server

# 2.10 Speaker Test

This app is modified from Qt's <u>Media Player</u> example that can be used to play the audio file (e.g. MP3) to test the built-in speaker of BE230E1.

#### 2.10.1 Qt C++ Class

The example uses a <u>QMediaPlayer</u> object to control the play/pause/stop functions. To give the application playlist capability it also uses a <u>QMediaPlayList</u> object.

#### 2.10.2 GUI

