**Banana Pi BPI-BC95**

**1.** Hardware introduction

**2.** Software introduction

**3**. How to setup and start

**一 Hardware introduction**

There are two major components on the board,namely NB-IoT module and MCU, their specs are below:

1. NB-IoT module:

**NB-IoT module：**Quectel BC95-B5/8/20 or BC95-G

**Model and Frequency band：**



**SIM card：**

1).Micro SD SIM Slot

2).e-SIM optional(footprint reserved on the board)

**Antenna：** IPEX antenna connector

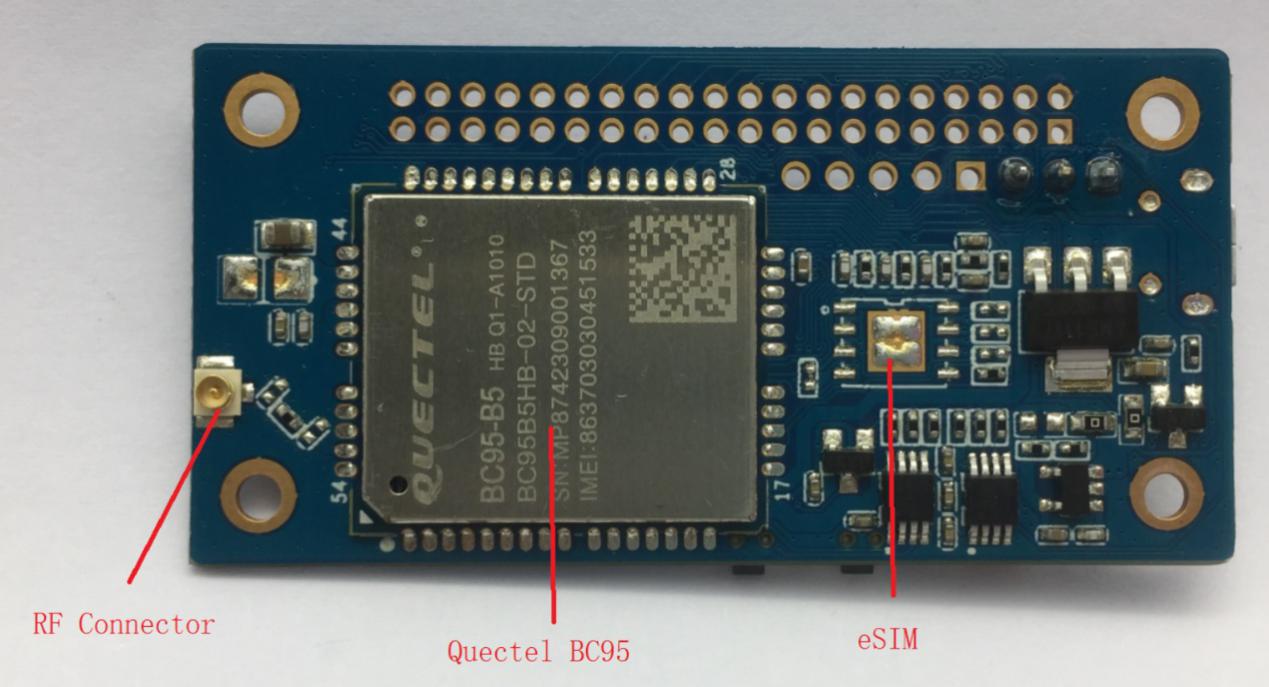
1. MCU:

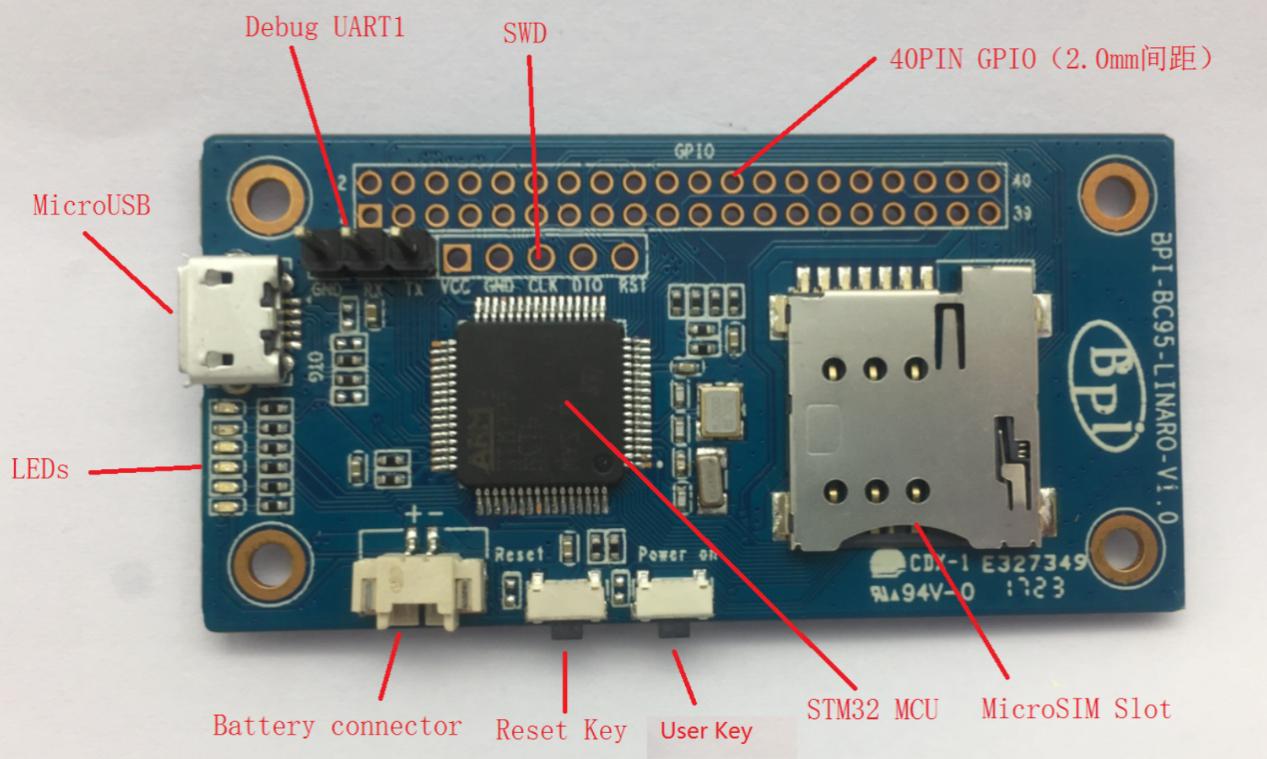


1. Development board :

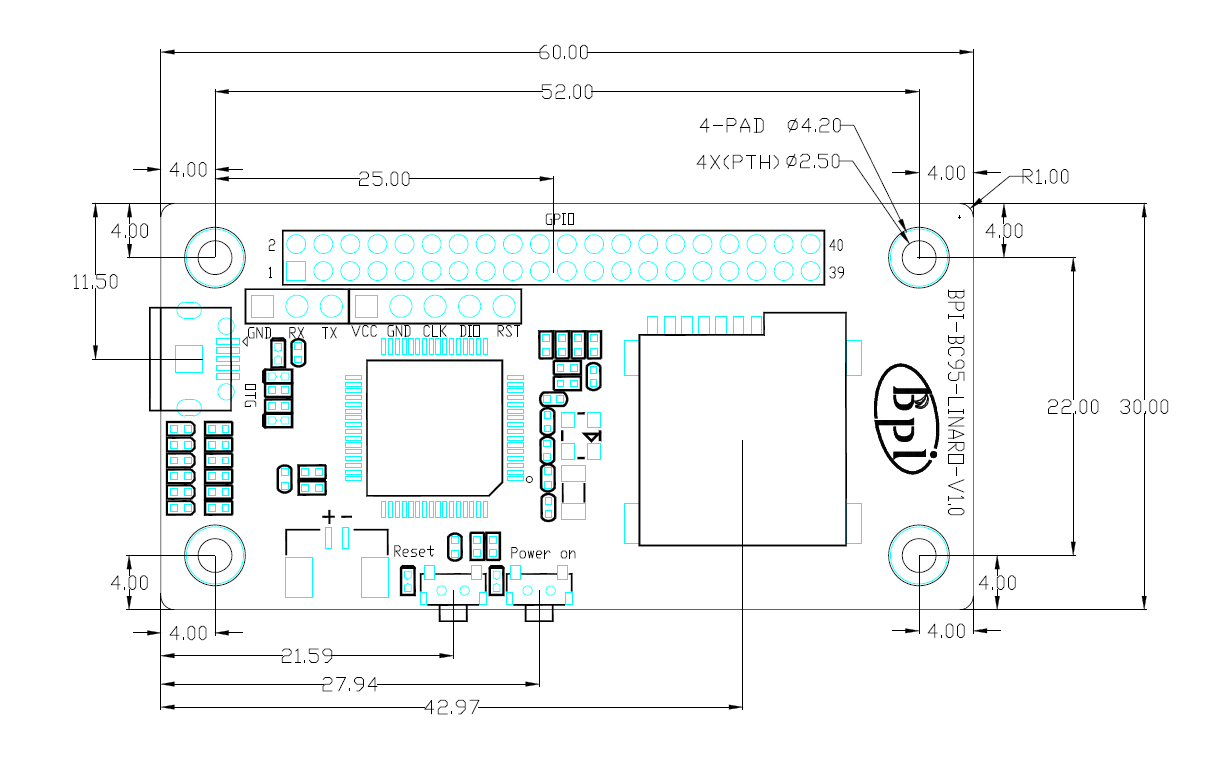
| **Hardware Specification of BPi BC95-LINARO** | |
| --- | --- |
| Soc | STM32F103RBT6 |
| CPU | ARM 32-bit Cortex™-M3 CPU @72MHz |
| SDRAM size | 20 KB |
| Flash size | 128KB |
| Power | 5V/2A via MicroUSB / 2PIN Battery Connector |
| **Features** | |
| Low-level peripherals | 40 Pins Header (2.0mm Pitch), 32×GPIOs,  Some of which can be used for specific functions including UART, I2C, SPI, PWM |
| On board Network | Quectel BC95 |
| On board SIM | MicroSD SIM Slot  eSIM(option) |
| USB | 1 USB 2.0 host |
| Buttons | Reset button, User button |
| Leds | 1 Power status Led and 5 other Leds |
| Sizes | 60 mm × 30mm |
| Weight | 10g |

Pictures of the board:





Dimensions:



[GPIO specification](http://www.bananapi.com.au/banana-pi-technical-specifications/)

**Banana Pi 40-pin GPIO(2.0mm pitch)**

Following is the Banana Pi GPIO Pin-out:

**BPI-BC95 40PIN GPIO (CON1)**

|  |  |  |
| --- | --- | --- |
| **GPIO Pin Name** | **Default Function** | **Function2：GPIO** |
| CON1-P01 | GND |  |
| CON1-P02 | GND |  |
| CON1-P03 | MCU\_USART2\_RTS | PA1 |
| CON1-P04 | GPIO\_PC6 | PC6 |
| CON1-P05 | MCU\_USART2\_RX | PA3 |
| CON1-P06 | RST\_BTN |  |
| CON1-P07 | MCU\_USART2\_TX | PA2 |
| CON1-P08 | SPI1\_SCK | PA5 |
| CON1-P09 | MCU\_USART2\_CTS | PA0 |
| CON1-P10 | SPI1\_MISO | PA6 |
| CON1-P11 | GPIO\_PB8 | PB8 |
| CON1-P12 | SPI1\_NSS | PA4 |
| CON1-P13 | GPIO\_PB9 | PB9 |
| CON1-P14 | SPI1\_MOSI | PA7 |
| CON1-P15 | I2C1\_SCL | PB6 |
| CON1-P16 | GPIO\_PC13 | PC13 |
| CON1-P17 | I2C1\_SDA | PB7 |
| CON1-P18 | GPIO\_PA15 | PA15 |
| CON1-P19 | I2C2\_SCL | PB10 |
| CON1-P20 | GPIO\_PB0 | PB0 |
| CON1-P21 | I2C2\_SDA | PB11 |
| CON1-P22 | GPIO\_PB1 | PB1 |
| CON1-P23 | USART1\_CK | PA8 |
| CON1-P24 | SPI2\_NSS | PB12 |
| CON1-P25 | USART1\_TX | PA9 |
| CON1-P26 | SPI2\_SCK | PB13 |
| CON1-P27 | USART1\_RX | PA10 |
| CON1-P28 | SPI2\_MISO | PB14 |
| CON1-P29 | USART5\_CK | PB5 |
| CON1-P30 | SPI2\_MOSI | PB15 |
| CON1-P31 | USART5\_TX | PB3 |
| CON1-P32 | USART4\_RX | PC11 |
| CON1-P33 | USART5\_RX | PB4 |
| CON1-P34 | USART4\_TX | PC10 |
| CON1-P35 | VCC\_1.8V |  |
| CON1-P36 | NC |  |
| CON1-P37 | VCC\_5V |  |
| CON1-P38 | NC |  |
| CON1-P39 | GND |  |
| CON1-P40 | GND |  |

**SWD(CN1)**

|  |  |  |
| --- | --- | --- |
| **CSI Pin Name** | **Default Function** | **Function2：GPIO** |
| CN1-P1 | VCC |  |
| CN1-P2 | GND |  |
| CN1-P3 | SWCLK | PA14 |
| CN1-P4 | SWDIO | PA13 |
| CN1-P5 | RESET-ST |  |

**Debug UART(CON2)**

|  |  |  |
| --- | --- | --- |
| CON2-P1 | GND |  |
| CON2-P2 | USART1\_RX | PA9 |
| CON2-P3 | USART1\_TX | PA10 |

**二 Software introduction**

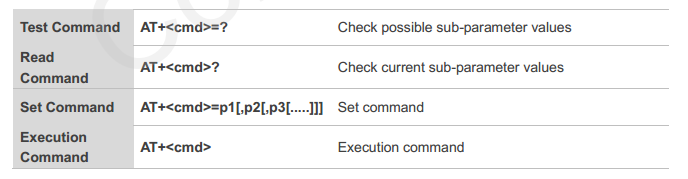
1. Lite OS brief

Huawei LiteOS is a lightweight open-source IoT OS and a smart hardware development platform. It simplifies IoT device development and device connectivity, makes services smarter, delivers superb user experience, and provides better data protection. Huawei LiteOS is designed for smart homes, wearables, IoV, and intelligent manufacturing applications.

For more about Lite OS please refer to: *http://developer.huawei.com/ict/en/site-iot/product/liteos*

**IDE：**recommended development environment is MDK521 + STM32F1xxx library

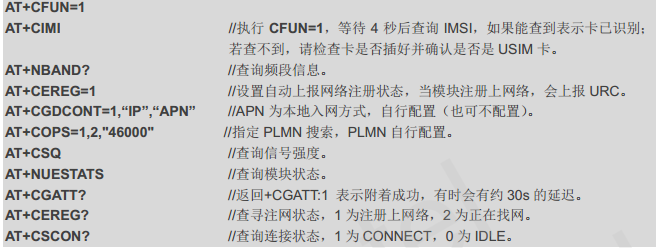
1. BC95 software brief:

Command format:

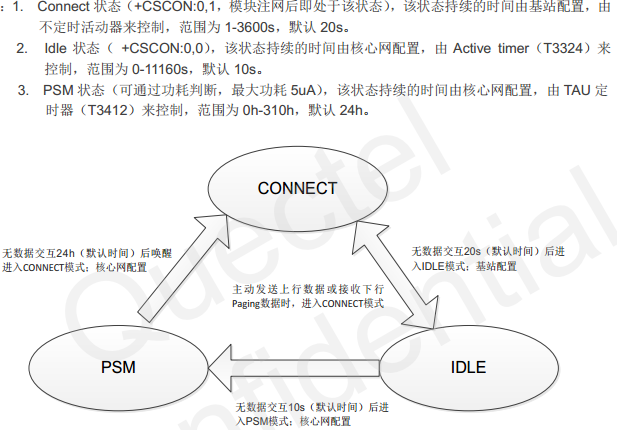
Searching network steps:



Manually configure network if fails to register network automatically:



Network connectivity status chart:



**三 How to setup and start to use**

1. Hardware preparation:

BPI OPEN DEBUGGER or J-Link DEBUGGER \*1

BPI BC95-Linaro development board \*1

Micro USB cable \*2

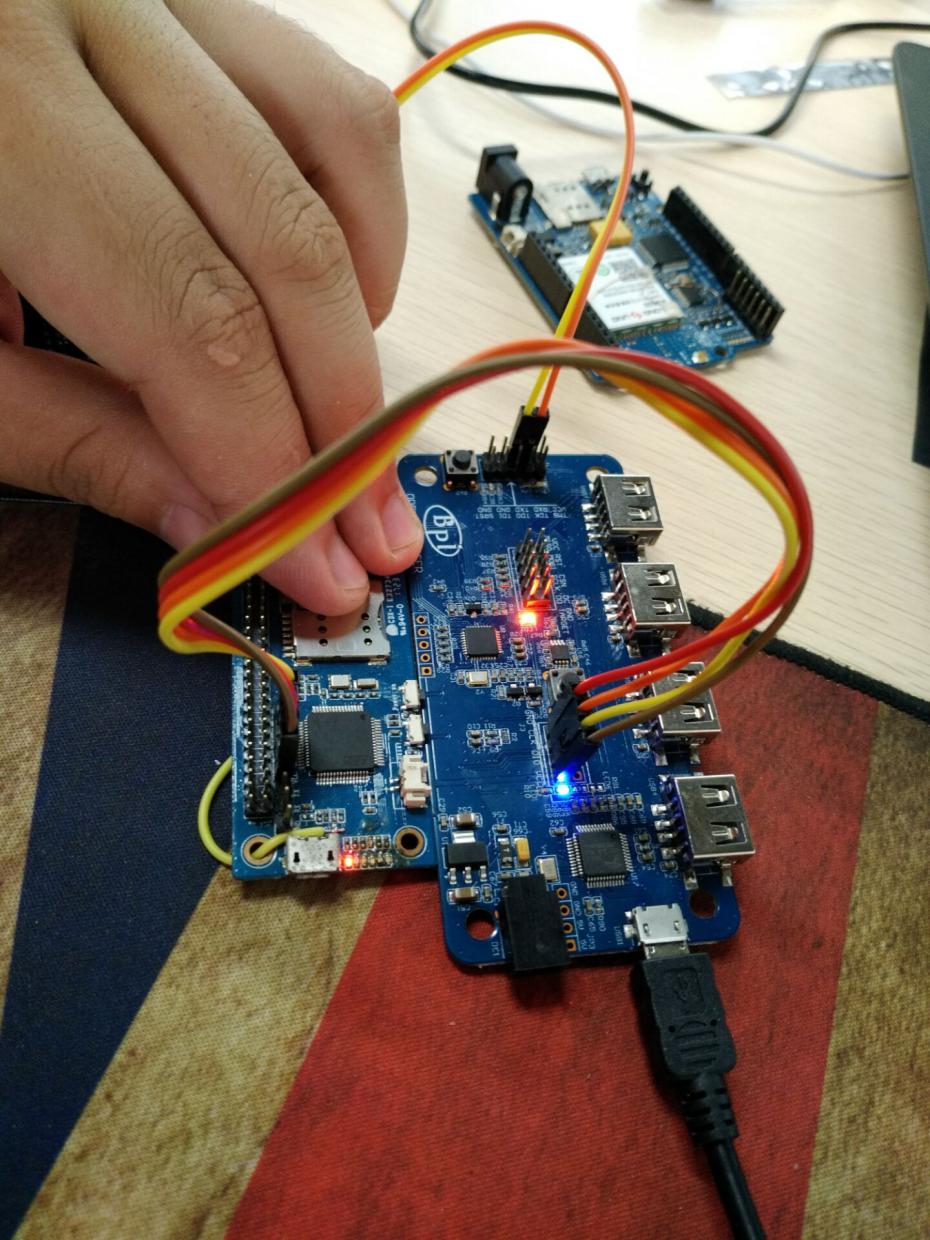
Dupont lines (female female) \*4

1. Hardware hookup

There are two ways to flash by either BPI OPEN DEBUGGER or J-Link DEBUGGER:

1. BPI OPEN DEBUGGER：

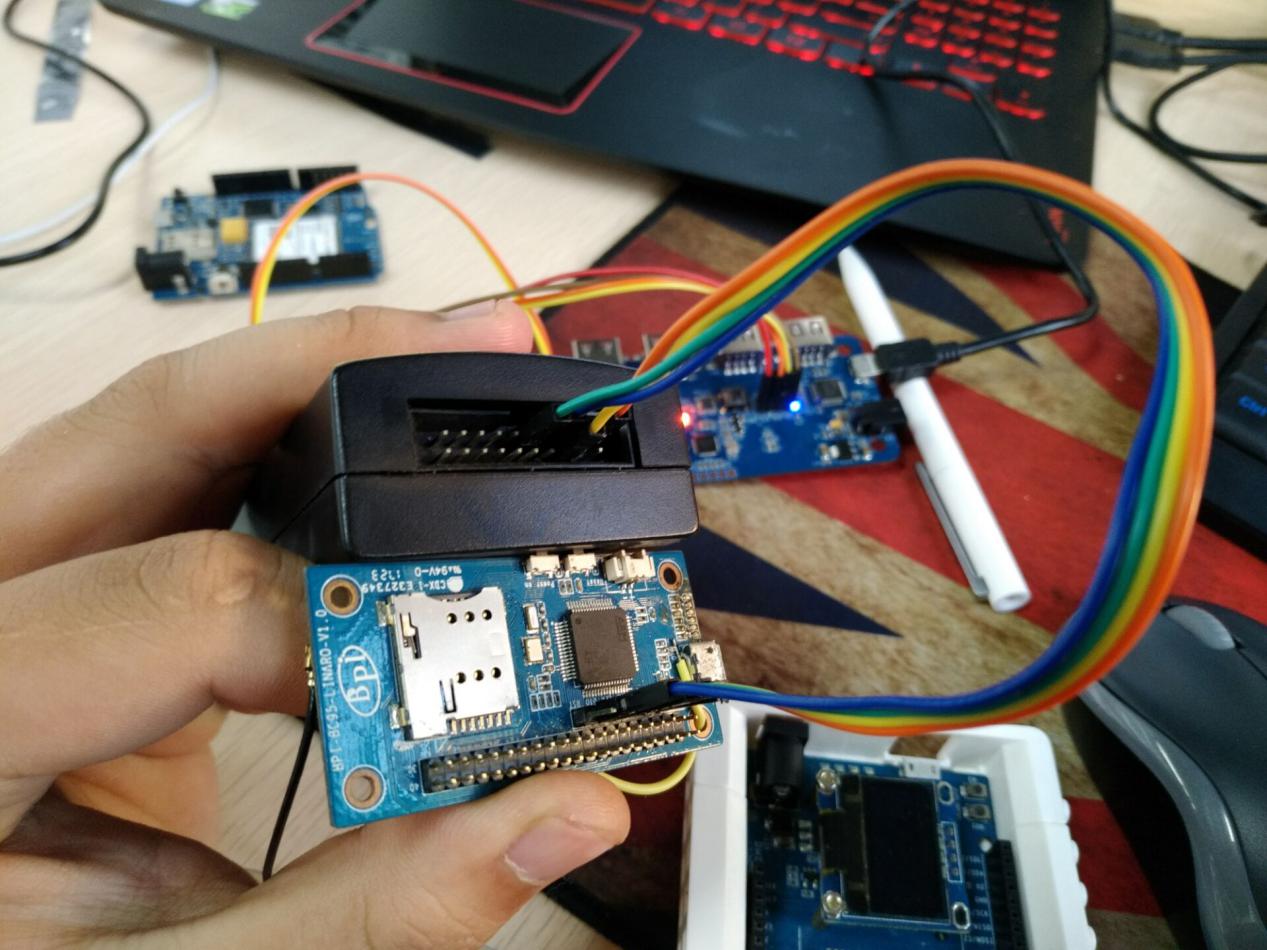
Using Dupont lines to connect BPI OPEN DEBUGGER with BPI BC95-Linaro SWD port，the connection are below:



|  |  |
| --- | --- |
| **BPI OPEN DEBUGGER** | **BPI NB-IoT Linaro SWD** |
| **3V3** | **VCC** |
| **GND** | **GND** |
| **DIO** | **DIO** |
| **CLK** | **CLK** |

1. J-Link DEBUGGER：





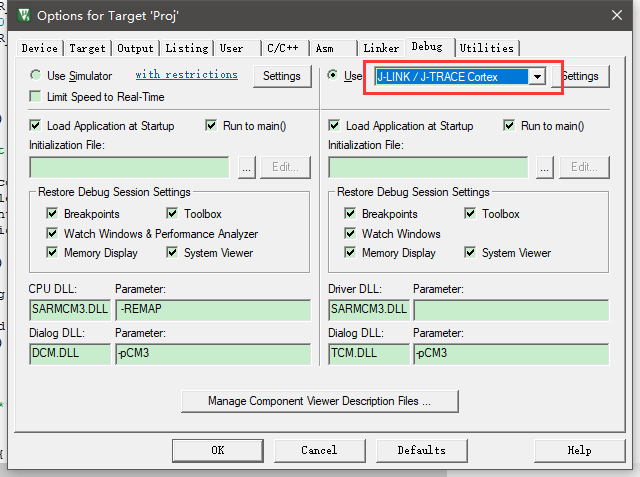
The corresponding connection between JLink and BPI BC95 are the following:

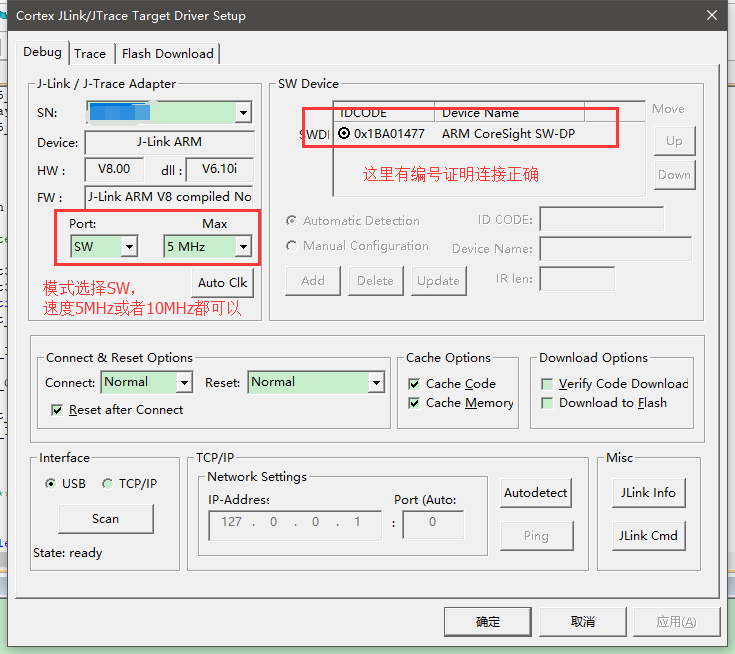
|  |  |
| --- | --- |
| **J-link** | **BPI NB-IoT Linaro SWD** |
| **3V3 (Pin1)** | **VCC** |
| **GND (Pin4.6.8.10……)** | **GND** |
| **SWDIO (Pin7)** | **DIO** |
| **SWCLK (Pin9)** | **CLK** |
| **nJTRST （Pin3）optional** | **RST (optional)** |

1. Software debugging setting:

MDK5 setting:

Create a project choosing the right chip,and set the Options of Target as follows:





1. Software sample cases:

There are two samples in BPI forum, please refer to the following links:

1. http://forum.banana-pi.org.cn/thread-2200-1-1.html

2. http://forum.banana-pi.org.cn/thread-2215-1-1.html

And there are some more information about the development board in the below link:

<http://forum.banana-pi.org.cn/forum-121-1.html>

1. Github code :

https://github.com/yelvlab/BPI\_NB-IoT\_Linaro\_96Boards