

CC3200-WiFi 套件

用户使用手册

资料版本：V 0.1

归档日期：2015 年 06 月

BOM 编码：201506080011

目录

一、整体概述	8
1.1 简介	8
1.2 套件特点	8
1.3 硬件组成	10
1.4 硬件描述	12
1.4.1 硬件框图	12
1.4.2 主要 IC	12
1.4.3 接口描述	13
1.4.4 LED	18
1.4.5 在线调试 CC3200	20
1.4.6 KEY	22
二、开发环境介绍	25
2.1 Code Composer Studio 6.1.0 的使用	25
2.1.1 选择工作空间	25
2.1.2 在 Code Composer Studio 6.1.0 上运行一个已经存在的程序	25
2.1.3 目标板设置	26
2.1.4 编译并下载程序	28
三、CC3200 MCU 实验	29
3.1 CC3200 Blinky Application	29
3.1.1 概述	29
3.1.2 应用说明	29
3.1.3 使用说明	29
3.2 CC3200 Timer Application	30
3.2.1 概述	30
3.2.2 应用说明	30
3.2.3 使用说明	30
3.3 CC3200 Watchdog Application	30
3.3.1 概述	30
3.3.2 应用说明	30

3.3.3	使用说明	30
3.4	CC32xx UART Demo Application	31
3.4.1	概述	31
3.4.2	应用说明	31
3.4.3	使用说明	31
3.5	CC3200 Interrupt Demo Application	32
3.5.1	应用说明	32
3.5.2	使用说明	32
3.6	CC3200 Sleep-DeepSleep Application	33
3.6.1	概述	33
3.6.2	应用说明	33
3.6.3	使用说明	34
3.7	CC3200 μ DMA Application	35
3.7.1	概述	35
3.7.2	应用说明	35
3.7.3	使用说明	35
3.8	CC32xx FreeRTOS Application	36
3.8.1	概述	36
3.8.2	应用说明	37
3.8.3	使用说明	37
3.9	CC3200 AES Demo Application.....	38
3.9.1	概述	38
3.9.2	应用说明	38
3.9.3	使用说明	38
3.10	CC3200 DES Demo Application.....	39
3.10.1	概述	39
3.10.2	应用说明	39
3.10.3	使用说明	40
3.11	CC3200 CRC Demo Application	41
3.11.1	概述	41

3.11.2 应用说明	41
3.11.3 使用说明	41
3.12 CC3200 SHA-MD5 Demo Application	42
3.12.1 概述	42
3.12.2 应用说明	43
3.12.3 使用说明	43
3.13 CC3200 ADC	44
3.13.1 概述	44
3.13.2 应用说明	44
3.13.3 使用说明	45
3.14 CC3200 PWM	46
3.14.1 概述	46
3.14.2 应用说明	46
3.14.3 使用说明	46
3.15 CC3200 SPI Demo	46
3.15.1 概述	46
3.15.2 应用说明	47
3.15.3 使用说明	47
3.16 CC3200 UART DMA Application	49
3.16.1 概述	49
3.16.2 应用说明	49
3.16.3 使用说明	49
3.17 CC3200 Timer Count Capture Application	50
3.17.1 概述	50
3.17.2 应用说明	51
3.17.3 使用说明	51
3.18 CC3200 Dynamic Library Loader	52
3.18.1 概述	52
3.18.2 应用说明	52
3.18.3 使用说明	52

四、CC3200 WIFI 实验	54
4.1 CC3200 Connection Policy	54
4.1.1 概述	54
4.1.2 应用说明	54
4.1.3 使用说明	54
4.2 CC32xx File Operations	55
4.2.1 概述	55
4.2.2 应用说明	55
4.2.3 使用说明	56
4.3 CC3200 Info Center Get Time Application	56
4.3.1 概述	56
4.3.2 应用说明	56
4.3.3 使用说明	57
4.4 CC3200 Info Center Get Weather Application	58
4.4.1 概述	58
4.4.2 应用说明	58
4.4.3 使用说明	58
4.5 CC3200 Getting Started with WLAN AP	59
4.5.1 概述	59
4.5.2 应用说明	59
4.5.3 使用说明	60
4.6 CC3200 Getting Started with WLAN Station	61
4.6.1 概述	61
4.6.2 应用说明	61
4.6.3 使用说明	62
4.7 CC32xx HTTP Server	63
4.7.1 概述	63
4.7.2 使用说明	63
4.8 CC3200 mDNS	67
4.8.1 概述	67

4.8.2 应用说明	67
4.9.3 使用说明	67
4.9 CC3200 Mode-Configuration Application.....	69
4.9.1 概述	69
4.9.2 应用说明	70
4.9.3 使用说明	70
4.10 CC32xx NWP Filter Application.....	70
4.10.1 概述	70
4.10.2 使用说明	71
4.11 P2P (Wi-Fi Direct)	71
4.11.1 概述	71
4.11.2 应用说明	72
4.11.3 使用说明	72
4.12 Provisioning AP	74
4.12.1 概述	74
4.12.2 使用说明	74
4.13 Provisioning with SmartConfig	75
4.13.1 概述和应用介绍	75
4.13.2 应用说明	75
4.13.3 使用说明	75
4.14 CC32xx Provisioning WPS Application.....	76
4.14.1 概述	76
4.14.2 应用说明	76
4.14.3 使用说明	77
4.15 Scan Policy	78
4.15.1 概述	78
4.15.2 应用说明	78
4.15.3 使用说明	78
4.16 TCP Socket	79
4.16.1 概述	79

4.16.2 应用说明	79
4.16.3 使用说明	79
4.17 Transceiver Mode	81
4.17.1 概述	81
4.17.2 应用说明	81
4.17.3 使用说明	82
4.18 UDP Socket	83
4.18.1 概述	83
4.18.2 应用说明	83
4.18.3 使用说明	83
4.19 Idle Profile	85
4.19.1 概述	85
4.19.2 应用说明	85
4.19.3 使用说明	86
4.20 File Download	88
4.20.1 概述和应用说明	88
4.20.2 使用说明	88
4.21 Deep-sleep	89
4.21.1 概述	89
4.21.2 应用说明	89
4.21.3 使用说明	89
4.22 Hibernate	90
4.22.1 概述	90
4.22.2 应用说明	90
4.22.3 使用说明	90
4.23 Watchdog System Demo	92
4.23.1 概述	92
4.23.2 应用说明	92
4.23.3 使用说明	92
4.24 TFTP Client	93

4.24.1 概述	93
4.24.2 应用说明	93
4.24.3 使用说明	93
4.25 Idle Profile (Non OS)	94
4.25.1 概述	95
4.25.2 应用说明	95
4.25.3 使用说明	95
4.26 MQTT Client.....	96
4.26.1 概述	96
4.26.2 应用说明	96
4.26.3 使用说明	98

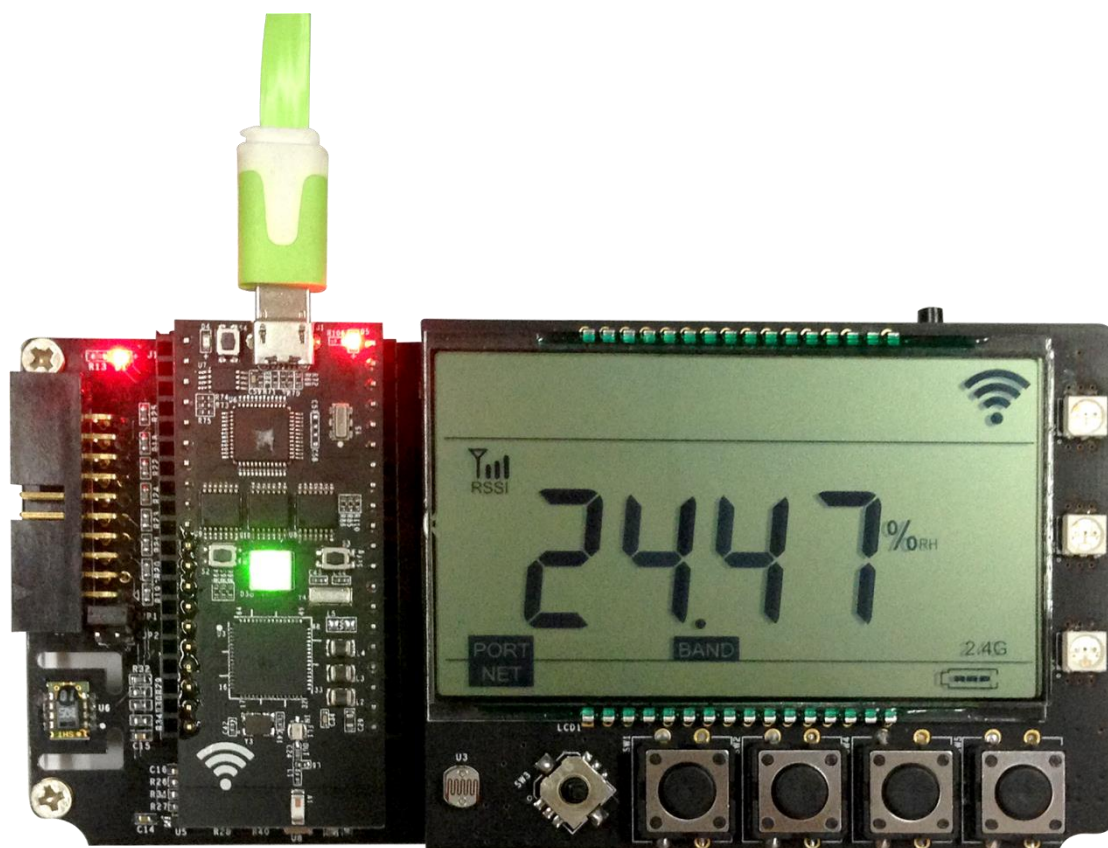
一、整体概述

1.1 简介

基于 TI 最新推出的 CC3200 平台。CC3200 使得客户能够轻松地众多的家用、工业和消费类电子产品增添嵌入式 Wi-Fi 和互联网功能，所凭借的特性包括：业界最低的功耗（适用于电池供电式设备），以及低功耗射频和高级低功耗模式；高度的灵活性，利用 CC3200 的集成型可编程 ARM® Cortex®-M4 MCU，从而允许客户添加其特有的代码。

可利用快速连接、云支持和片上 Wi-Fi、互联网和稳健的安全协议实现针对 IoT 的简易型开发，无需具备开发连接型产品的先前经验。

能够采用某种手机或平板电脑应用程序或者一种具有多种配置选项（包括 SmartConfig™ 技术、针对 WPS 和 AP 模式的网络浏览器简单且安全地将其设备连接至 Wi-Fi。



该实训套件由组成如下：

编号	名称/型号	数量	备注
1	WF 通信模块 OURS-WF-CORE	1	
2	无线模块开发板 OURS-SDK-EB	1	
3	MicroUSB 数据线	1	
4	光盘	1	

1.2 套件特点

- 强大 ARM Cortex-M4F

- 高达 80MHz 时钟速度
- 高达 256KB RAM
- 支持外部串行 Flash
- 丰富的外设接口
 - 8 位并行 Camera 接口
 - 1 个多通道音频串行接口
 - 1 个 SD/MMC 接口
 - 1 个 S P I 接口
 - 2 个 UART
 - 4 个通用的 Timer
 - 1 路看门狗 Timer
 - 4 路 12Bit ADC
- WiFi 网络子系统
 - Featuring WiFi Internet-On-a-Chip
 - 专用 ARM MCU 将 WiFi 和网络协议与应用处理器完全独立出来
 - 内置 802.11b/g/n 基带、MAC、射频，WiFi 驱动和其他相关附件
 - 内置 TCP/IP 协议栈
 - ◆ 工业级 BSD Socket API
 - ◆ 同时支持 8 路 TCP 或者 UDP Socket 通信
 - ◆ 同时支持 2 路 TLS 和 SSL Socket 通信
 - 强大加密引擎，通过 256Bit-AES 加密，以适应安全而且快速的 WiFi 和 Internet 连接
 - 支持 Station、AP 和 WiFi 直连模式
 - 支持个人 WPA2 和企业级安全
 - 支持 SmartConfig、WPS2 技术，以满足简易而又灵活的 WiFi Provisioning
- 发射功率
 - 18.0dBm @ 1 DSSS
 - 14.5dBm @ 54 OFDM
- 接收灵敏度
 - -95.7dBm @ 1 DSSS
 - -74.0dBm @ 54 OFDM
- 电源系统
 - 单板电源系统
 - USB 接口 5V 供电
 - 外部 5V 供电
 - 外部 3.3V 供电
 - USB 接口供电有效时，外部 5V/3.3V 供电接口不能使用
 - 外部 5V 供电有效时，USB 接口 5V/外部 3.3V 接口不能使用
 - 外部 3.3V 供电有效时，USB 接口 5V/外部 5V 接口不能使用
 - WiFi 电源子系统
 - 集成宽压输入的 DC-DC
 - ◆ 宽压模式：2.1 V to 3.6V
 - ◆ 1.8V 预调整模式：1.85V
 - 高级低功耗模式

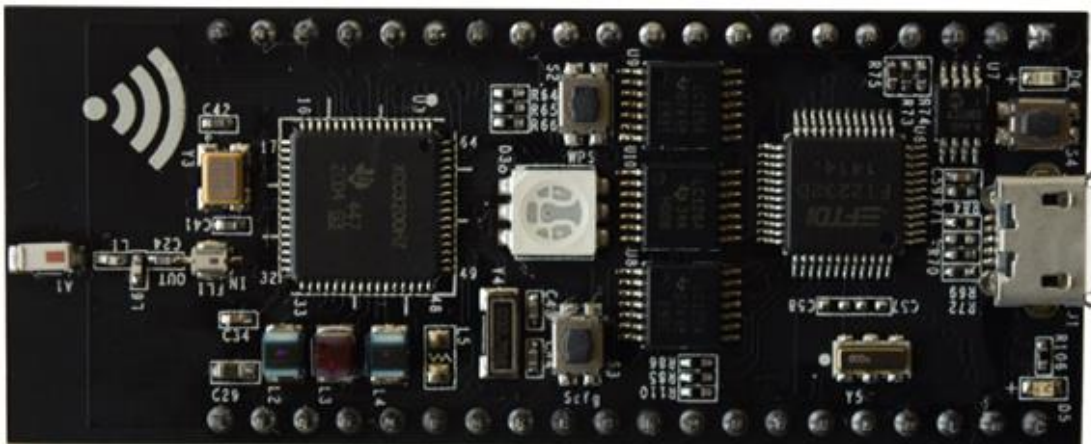
- ◆ 休眠:4uA
- ◆ 低功耗深度睡眠: 120uA
- ◆ 接收 (MCU 运行态): 59mA @54OFDM
- ◆ 发送 (MCU 运行态): 229mA @ 54OFDM 最大功耗
- 模块外部接口
 - USB 接口
 - 1 路 SPI
 - 2 路串口
 - 4 路 ADC
 - 2 路 PWM 输出接口
 - 1 个 RGB 三色灯
 - 两个用户按键
- 板载段式 LCD, 便于交互
- 集成多种传感器

1.3 硬件组成

OURS-SDK-WFB 采用 OURS-WF-CORE 加无线模块开发板 (OURS-SDK-EB) 的形式。

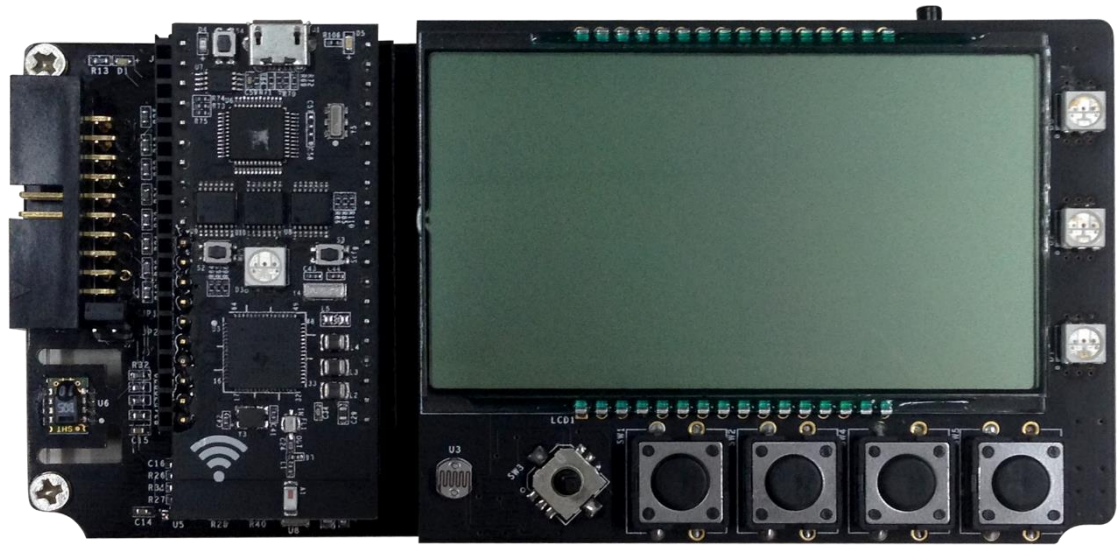
OURS-WF-CORE:

OURS-WF-CORE 基于 CC3200 实现, 搭载 FT2232 实现在线调试和串口通讯功能。同时提供丰富的扩展接口, 例如 SPI、UART、GPIO、ADC、PWM 等产品形式:



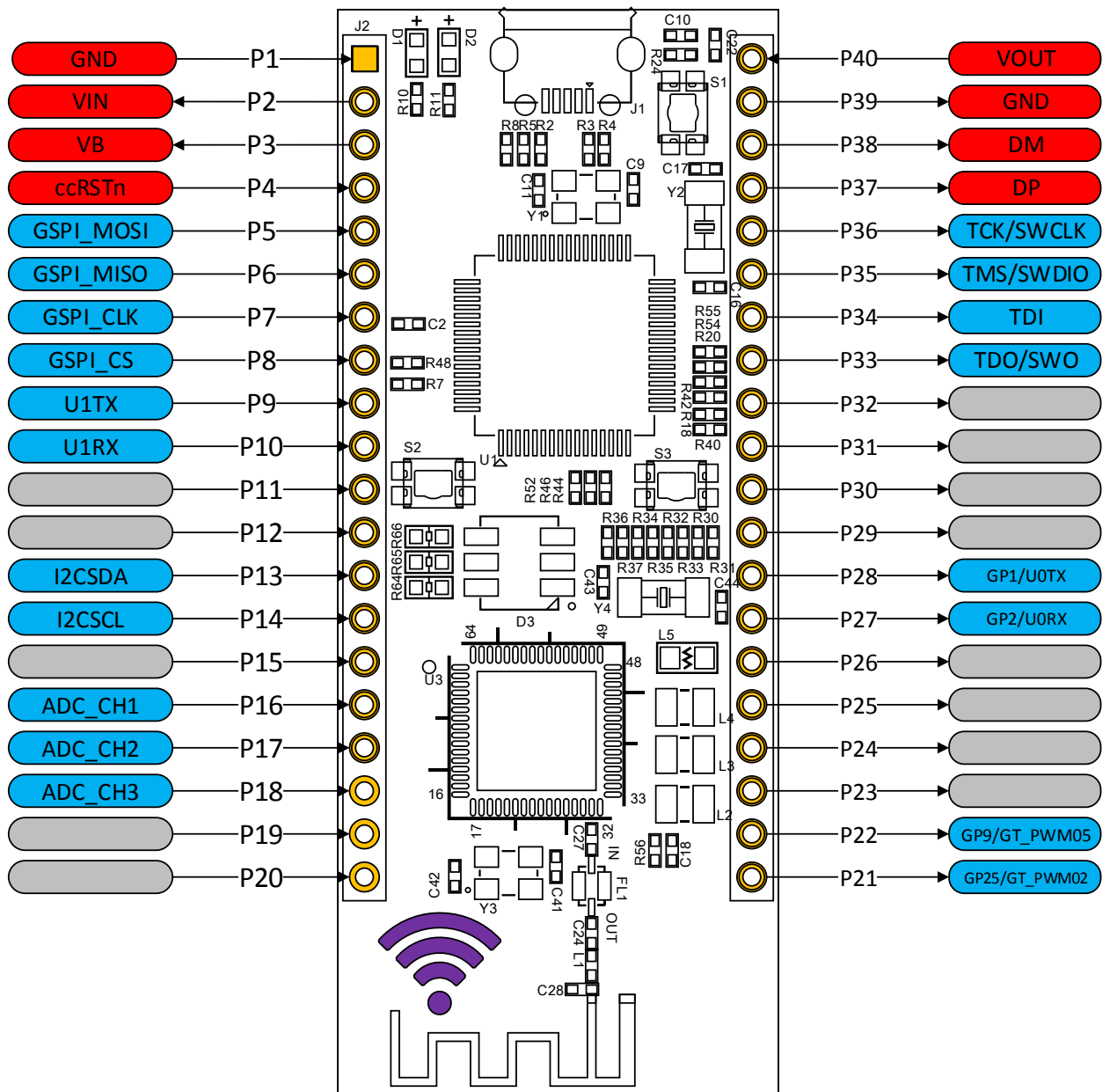
OURS-SDK-EB:

无线模块评估底板, 板载温湿度、光照度、加速度、气压传感器和简单的人机交互 LCD。OURS-SDK-WFB 产品形式如下:



Number	Part Number	Manufacturer	Description
1	FT2232D	FTDI	IC Dual USB UART/FIFO Support IIC SPI Jtag
3	CC3200-M2RTD[T/R]	TI	IC WiFi SoC M4 256K RAM NWP Sec WiFi VQFN64

1.4.3 接口描述



接口定义描述如下

Pin Number	Pin Location @CC3200	Pin Name	Description
P1	N/A	GND	模块输入电源地
P2	N/A	VIN	输入 5V 直流电源，该电源有效时，P3 引脚不再输入 3.3V 电源
P3	N/A	VB	输入 3.3V 直流电源，该直流电源有效时，P2 引脚不再输入 5V 电源
P4		nR	外部输入的模块复位信号，实现对模块的复位控制 整个复位系统的设置存在 Bug： 当 TM4C123H6PM 作为 obed-if 或者仅仅实现 USB 转串口功能的时候，按下复位按键会将

			<p>obed-if 和 CC3200 同时复位， 实际需要的是：按下复位按键的时候，仅仅复位 CC3200，同时也可以通过 TM4C123H6PM 实现对 CC3200 的复位</p>
P5	PIN7	GSPI_MOSI	<p>Default: General-Purpose I/O GPIO16 GPIOA2 Bit0 Configured: General SPI MOSI Option: CAM_D6: Parallel Camera Data Bit 6 UART1_TX: UART1 TX Data GT_CCP07: Timer Capture Port 07</p>
P6	PIN6	GSPI_MISO	<p>Default: General-Purpose I/O GPIO15 GPIOA1 Bit7 Configured: General SPI MISO Option: I2C_SDA I2c Data CAM_D5 : Parallel Camera Data Bit 5 GT_CCP06: Timer Capture Port 06</p>
P7	PIN5	GSPI_CLK	<p>Default : General-Purpose I/O GPIO14 GPIOA1 Bit6 Configured: General SPI CLK Option : I2C_SCL : I2C Clock CAM_D4 : Parallel Camera Data Bit 5 GT_CCP05: Timer Capture Port 05</p>
P8	PIN8	GSPI_CS	<p>Default : General-Purpose I/O GPIO17 GPIOA2 Bit1 Configured : General SPI Chip Select Option : UART1_RX : UART1 RX Data CAM_D7 : Parallel Camera Data Bit 7</p>
P9	PIN3	UART0_TX	<p>Default : General-Purpose I/O GPIO12 GPIOA1 Bit5 Configured : UART0 Tx Data Option : McACLK : I2S Audio Port Clock O pVS(VSYNC) : Parallel Camera Vertical Sync GT_CCP03 : Timer Capture Port 03</p>
P10	PIN4	UART0_RX	<p>Default : General-Purpose I/O GPIO13 GPIOA1 Bit6 Configured : UART0 RX Data Option : I2C_SDA : I2C Data pHS(HSYNC) : Parallel Camera Horizontal Sync</p>

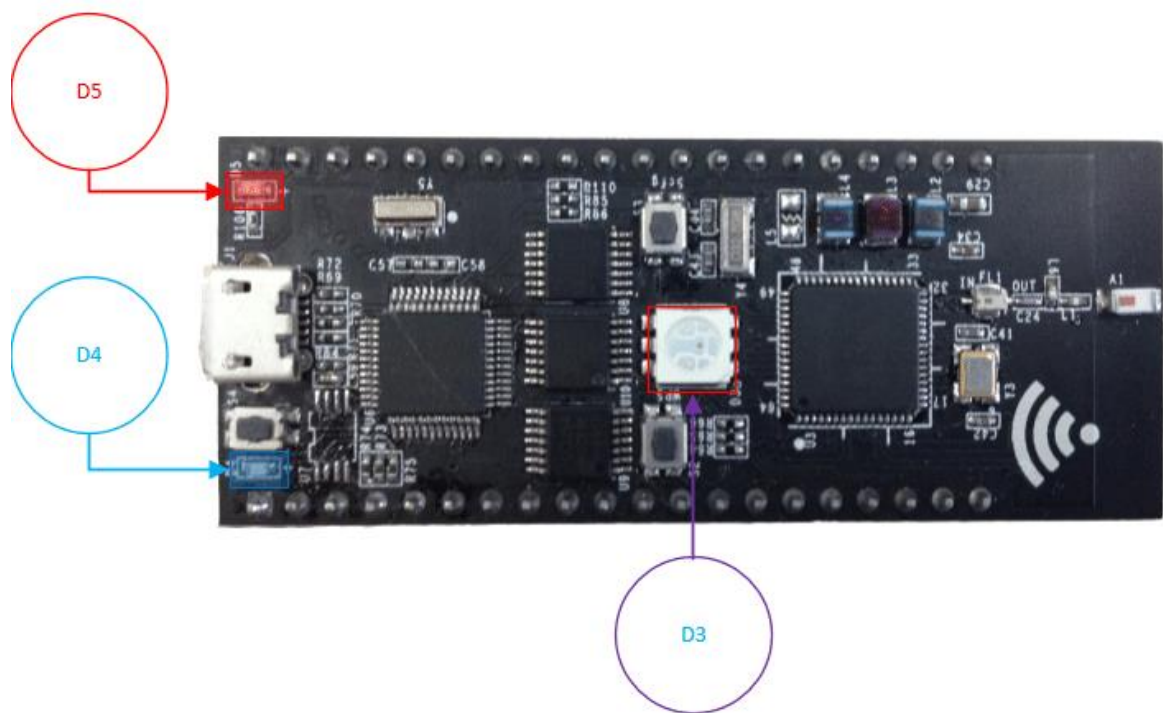
			GT_CCP04 : Timer Capture Port 04
P11	PIN50	UART0_CTS	Default : General-Purpose I/O GPIO0 (GPIOA0 Bit0) Configured : UART 0 Clear to Send input Option : McAXR1 – I2S Audio Port Data1(Rx/Tx) GT_CCP00 –Timer Capture Port 00 GSPI_CS –General SPI Chip Select UART1_RTS-Uart1 Request to Send Output UART1_CTS-Uart 1 Clear to Send Input McAXR0-I2S Audio Port Data0(Rx/Tx)
P12	PIN62	UART0_RTS	Default : General-Purpose I/O GPIO7 (GPIOA0 Bit7) Configured : UART0_RTS-Uart0 Request to Send Output Option : CAM_D0-Parallel Camera Data Bit 0 UART1_CTS-Uart1 Clear to Send Input UART0_CTS-UART0 Clear to Send Input GT_CCP06-Timer Capture Port 06
P13	PIN2	I2CSDA	Default : General-Purpose I/O GPIO11(GPIOA1 BIT3) Configured : I2C_SDA I2C Data Option : GT_PWM07-Pulse-width modulated O/P 07 pXCLK(XVCLK)-Free Clock to Parallel Camera UART1_RX –UART TX Data SDCARD_CMD-SD Card Command Line GT_CCP02-Timer Capture Port 02 McAFSX-I2S Audio Port Frame Sync
P14	PIN1	I2CSCL	Default : General-Purpose I/O GPIO10(GPIOA1 BIT2) Configured : I2C_SCL-IIC Clock Option : GT_PWM06-Pulse-width modulated O/P 06 UART1_TX-UART Tx Data SDCARD_CLK-SD Card Clock GT_CCP01-Timer Capture Port
P15	PIN57	ADC_CH0	Default : General-Purpose I/O GPIO2(GPIOA1 Bit2) Configured : ADC_CH0-ADC Channel 0 Input(1.5V Max) Option : UART0_RX-UART0 RX Data

			UART1_RX-UART1 RX Data GT_CCP02-Timer Capture Port 02
P16	PIN58	ADC_CH1	Default: General-Purpose I/O GPIO3(GPIOA0 BIT3) Configured: ADC_CH1-ADC Channel1 Input(1.5V max) Option: UART1_TX-UART1 Tx Data CAM_D3-Parallel Camera Data Bit 3
P17	PIN59	ADC_CH2	Default: General-Purpose I/O GPIO4(GPIOA0 BIT4) Configured: ADC_CH2-ADC Channel2 Input(1.5V max) Option: UART1_RX-UART1 Rx Data CAM_D2-Parallel Camera Data Bit 2
P18	PIN60	ADC_CH3	Default: General-Purpose I/O GPIO5(GPIOA0 BIT5) Configured : ADC_CH3-ADC Channel3 Input(1.5V Max) Option: CAM_D1-Parallel Camera Data Bit 1 McAXR1-I2S Audio Port Data1(RX/TX) GT_CCP05-Timer Capture Port
P19	N/A	N/A	
P20	N/A	N/A	
P21	PIN61	GT_CCP06	Default: General-Purpose I/O GPIO6(GPIOA0 BIT6) Configured: GT_CCP06-Timer Capture Port 06 Option : UART0_RTS-UART0 Request to Send Output CAM_D0-Parallel Camera Data Bit 0 UART1_CTS-UART1 Clear to Send Input UART0_CTS-UART0 Clear to Send Input
P22	PIN55	GT_CCP01	Default: General-Purpose I/O GPIO1(GPIOA0 BIT1) Configured : GT_CCP01-Timer Capture Port 01 Option : UART0_TX-UART0 Tx Data pCLK(PIXCLK)-Pixel Clock From Parallel Camera Sensor UART1_TX-UART1 TX Data
P23	N/A	N/A	
P24	N/A	N/A	

P25	N/A	N/A	
P26	N/A	N/A	
P27	PIN58	GPIO3/UART_TX	<p>注意：与 P16 冲突</p> <p>Default: General-Purpose I/O GPIO3(GPIOA0 BIT3)</p> <p>Configured UART1_TX-UART1 Tx Data</p> <p>Option:</p> <p>ADC_CH1-ADC Channel1 Input(1.5V max)</p> <p>CAM_D3-Parallel Camera Data Bit 3</p>
P28	PIN59	GPIO4/UAT1_RX	<p>Default: General-Purpose I/O GPIO4(GPIOA0 BIT4)</p> <p>Configured:UART1_RX-UART1 RX Data</p> <p>Option :</p> <p>CAM_D2-Parallel Camera Data Bit 2</p>
P29	N/A	N/A	
P30	N/A	N/A	
P31	N/A	N/A	
P32	N/A	N/A	
P33	PIN17	TDO/SWO	Default : Jtag TDO
P34	PIN16	TDI	Default : Jtag TDI
P35	PIN20	TMS/SWDIO	Default : Jtag TMS
P36	PIN19	TCK/SWCLK	Default : Jtag TCK
P37	N/A	DP	FT2232 USB 信号负
P38	N/A	DM	FT2232 USB 信号正
P39	N/A	GND	模块输出 5V 电源
P40	N/A	VOUT	模块输出 3.3V 电源

1.4.4 LED

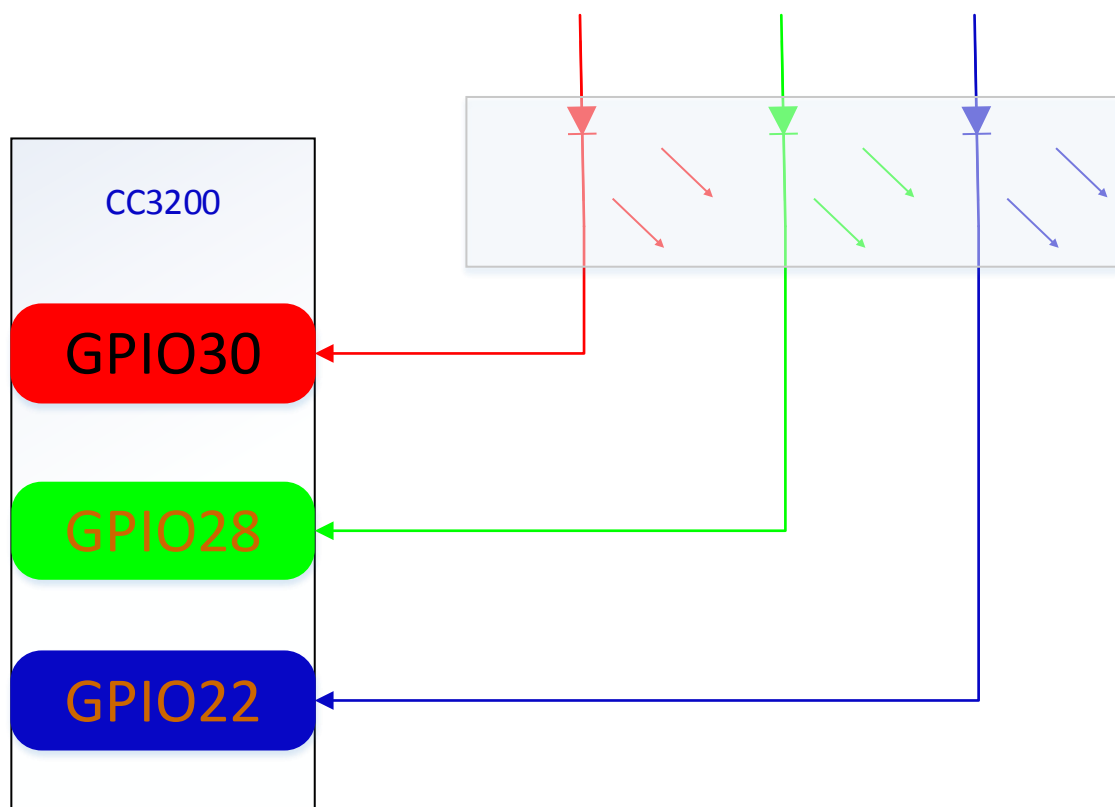
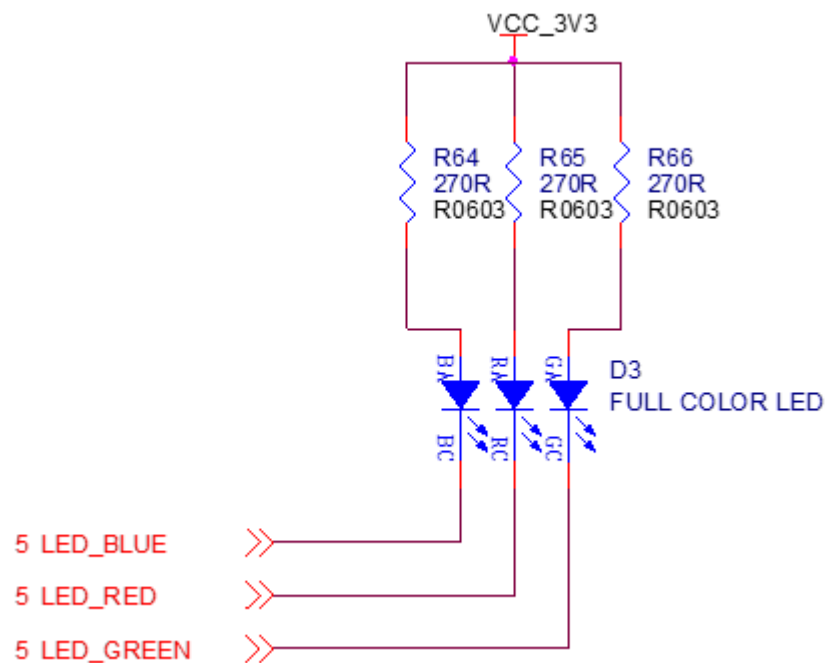
OURS-WIFI-CORE 使用了 1 个 RGB 灯、1 个红色 LED、1 个蓝色 LED，如



三个 LED 灯的功能说明如下：

LED Number	Function	Controlled By GPIO	Description
D1	红色 LED，用于指示 USB 转串口在接收数据	PD2@TM4C123H6PM	PD2 低电平控制有效，PD2 为电平的时候的 D2 灯亮
D2	蓝色 LED，用于指示 USB 转串口在发送数据	PD3@TM4C123H6PM	PD3 低电平控制有效，PD3 为电平的时候的 D2 灯亮
D3	5050 RGB LED，用于 CC3200 工作状态指示 红色：CC3200 心跳灯 蓝色：CC3200 网络建立成功 绿色：CC3200 通过 RF 收发数据	红 色： GPIO30@CC3200	GPIO30 低电平控制有效，GPIO30 位低电平的时候红色灯亮
		蓝色： GPIO28@CC3200	GPIO28 低电平控制有效，GPIO28 位低电平的时候蓝色灯亮
		绿色： GPIO22@CC3200	GPIO22 低电平控制有效，GPIO22 位低电平的时候绿色灯亮

D3 与 CC3200 的连接关系如下：



三个 GPIO 的信息如下图

编号	颜色	GPIO	Pin Number	GPIO_BaseAddress	GPIO_BitMask
1	Red	GPIO_30	PIN_53	GPIOA3_BASE	0x40
2	Green	GPIO_28	PIN_18	GPIOA3_BASE	0x10
3	Blue	GPIO_22	PIN_15	GPIOA2_BASE	0x40

1.4.5 在线调试 CC3200

1.4.5.1 模块独立工作模式

CC3200 支持三种 Boot 方式

SoP Mode	SoP[2]	SoP[1]	SoP[0]	NAME	Comment
LDfrUART	PullUP	PullDown	PullDown	UART DOWNLOAD (4-wire Jtag)	Factory/Lab Flash/SRAM load through UART
Fn2WJ	PullDown	PullDown	PullUP	FUNCTIONAL_2WJ	Functional development mode. In This Mode ,two pin JTAG is available to the developer. TMS and TCK are available for Debug Connection.
Fn4WJ	PullDown	PullDown	PullDown	FUNCTIONAL_4WJ	Functional development mode. In This Mode ,Four pin JTAG is available to the developer. TMS,TCK ,TDI and TDO are available for Debug Connection.

实现时采用 4 位的拨码开关进行切换.

这里推荐使用 Fn2WJ 进行在线调试；可以使用 J-Link 在线调试 CC3200，CC3200 支持四线 JTAG 调试模式或者 SWD 调试模式，通过 SoP 进行设置

在不使用用 OURS-SENSOR-NODE 的情况下，推荐使用 SWD 调试模式，需要确保 SoP[2:0]="001"

在 OURS-WIFI-CORE-REVC 中通过拨码开关实现：

Mode	1(SoP0)	2(SoP1)	3(SoP2)	4(NULL)
LDfrUART	0	0	1	NA
Fn2WJ	1	0	0	NA

Fn4WJ	0	0	0	NA
-------	---	---	---	----

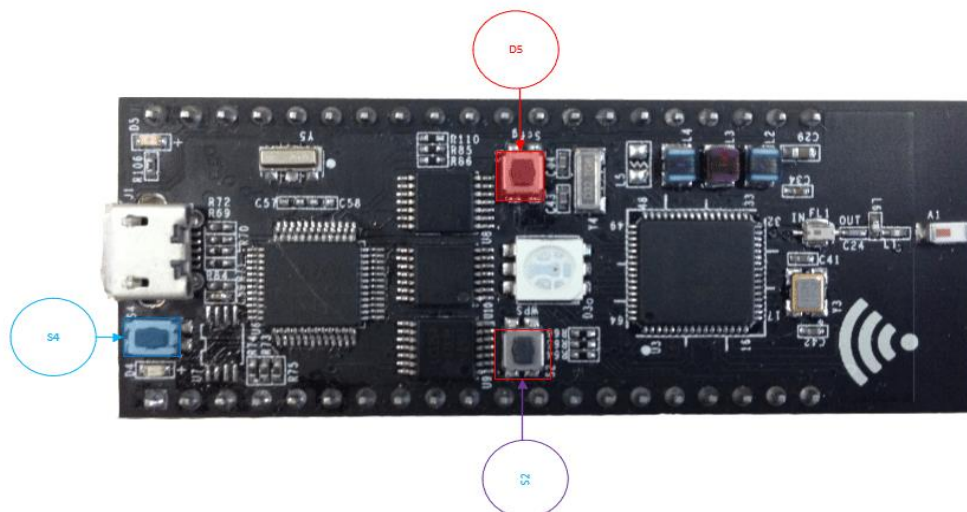
■

1.4.5.2 WiFi 智能传感器模式

Sop 设置为“001”，如下图：

1.4.6 KEY

OURS-WIFI-CORE 单板有三个按键，分别实现单板复位、CC3200 功能按键，三个按键在单板上面的布局如下：

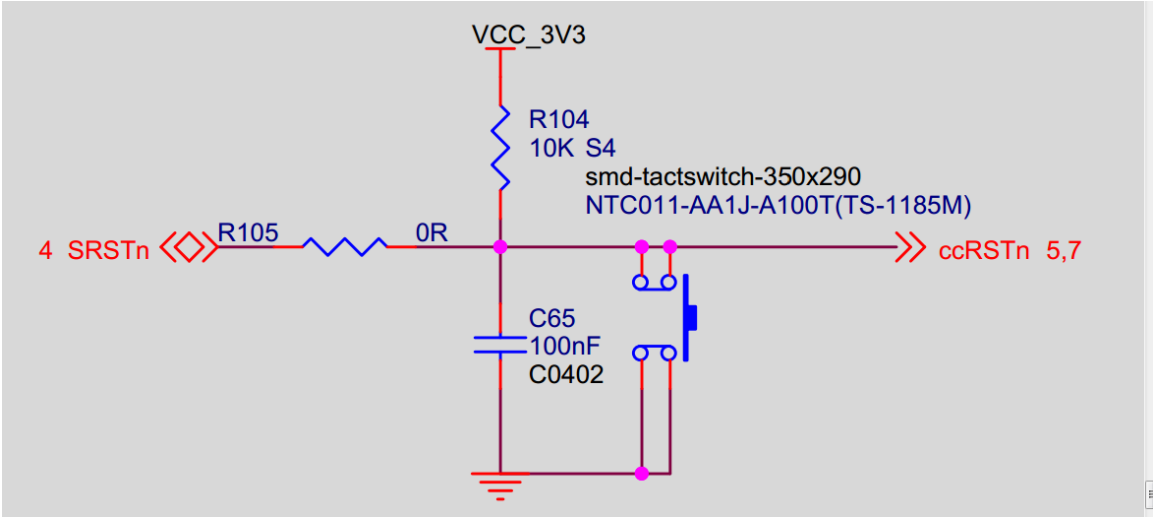


按键功能描述

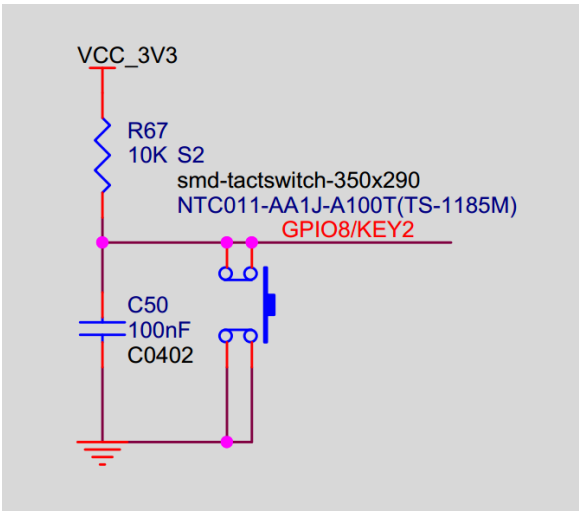
按键编号	功能	关联 IO	备注
S1	单板复位按键	N/A	实现对整个单板的复位 注意：实际上需要只是复位 CC3200,

			但是目前的设计中，由于将 CC3200 的复位信号与 TM4C123H6PM 的复位信号接在了一起，这样就导致 TM4C123H6PM 与 CC3200 同时被复位。当 TM4C123H6PM 枚举出串口的时候，串口会因为 TM4C123H6PM 被复位而小时，再次枚举。给系统调试带来了不便
S2	CC3200 功能按键	GPIO8@CC3200	按键功能未定
S3	CC3200 功能按键	GPIO9@CC3200	按键功能未定

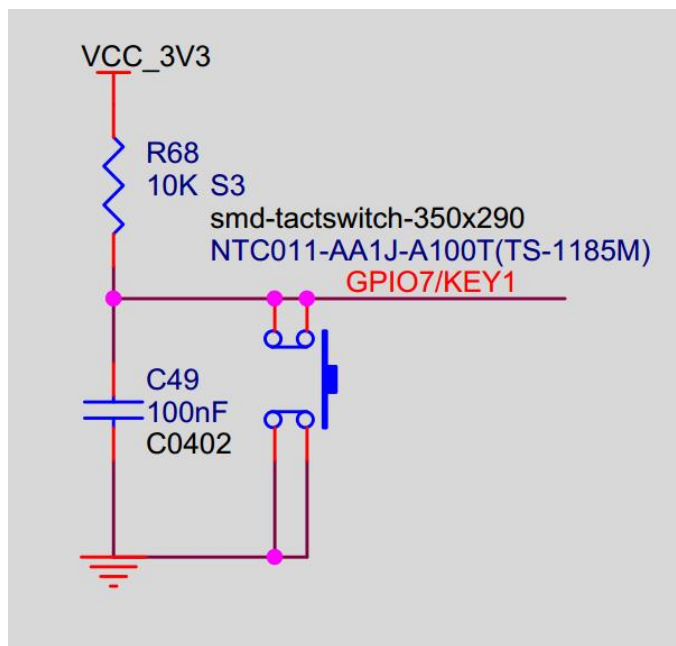
按键 S1 与单板主要 IC 的连接关系（单板复位）



按键 S2 与 CC3200 的连接关系如下：



按键 S3 与 CC3200 的连接关系如下：

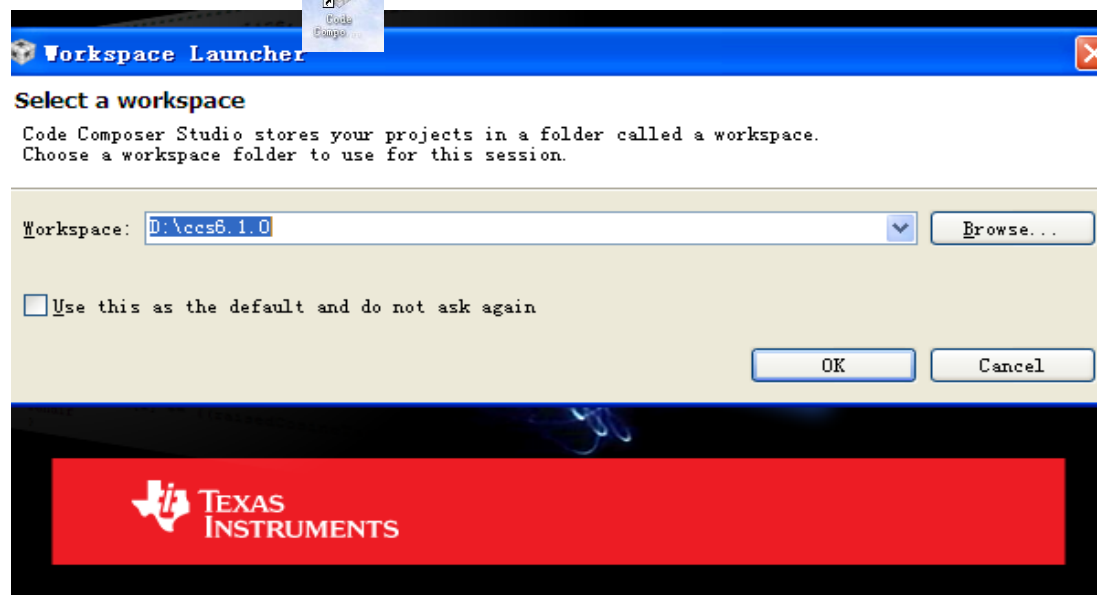


二、开发环境介绍

2.1 Code Composer Studio 6.1.0 的使用

2.1.1 选择工作空间

双击桌面快捷方式 ，跳出 workspace 保存路径对话框，选择相应的路径，如图

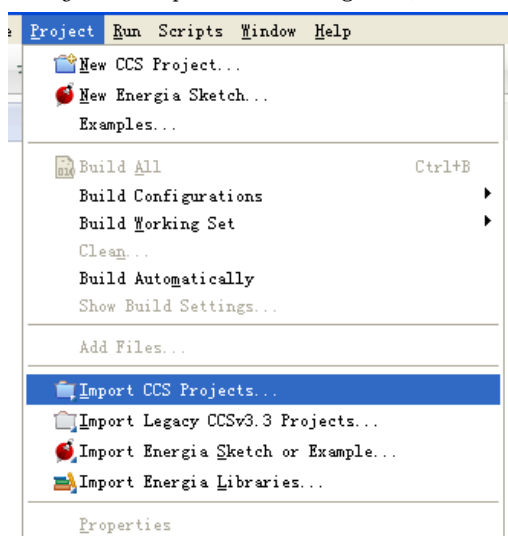


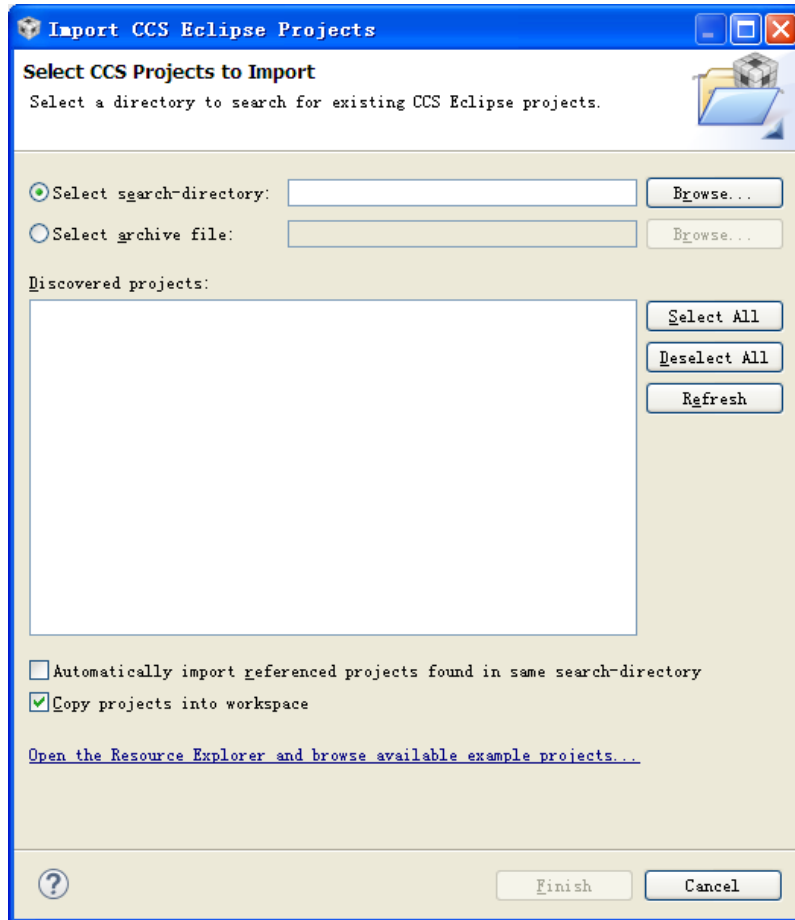
选择好 workspace 保存路径后，点击 OK。

2.1.2 在 Code Composer Studio 6.1.0 上运行一个已经存在的程序

一、打开一个已存在的项目文件，这里选择 C:\TI\CC3200SDK_1.1.0\cc3200-sdk\example\ 中的 timer 例子做相应说明，步骤如下：

(1) 在项目菜单中 Project→Import Existing CCS/CCE Eclipse Project 如图。

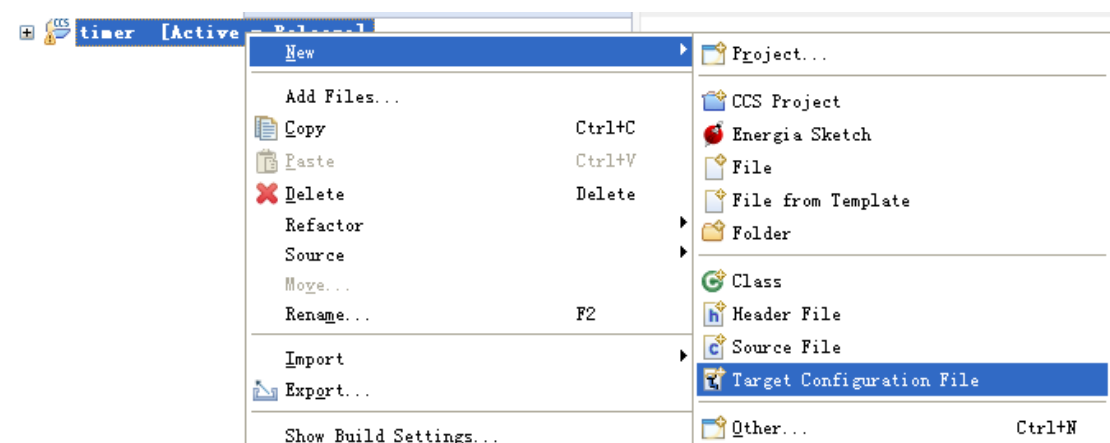


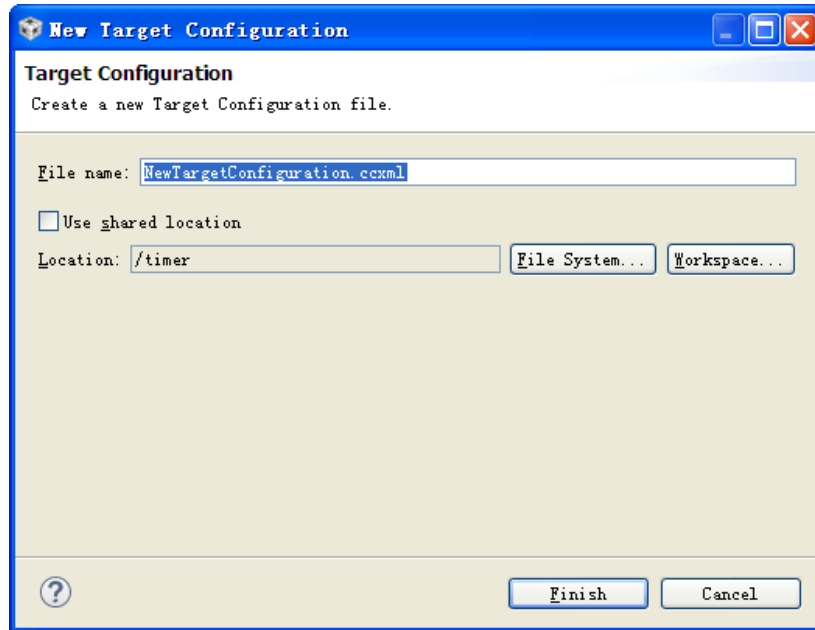


(2) select search-directory 中选择这个项目所在的位置，点击 Browse 找到 timer 例子程序的位置，选择好后点击 Finish。

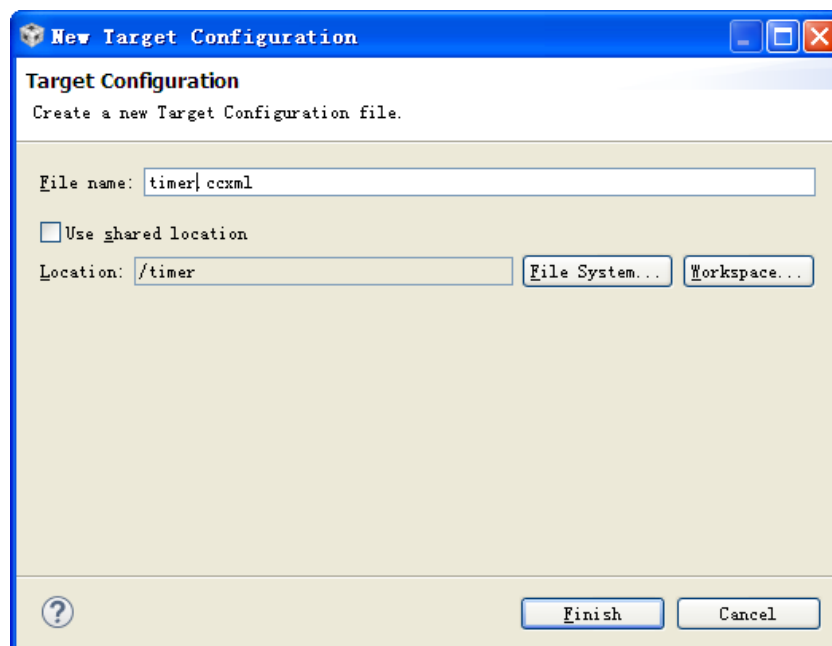
2.1.3 目标板设置

一、选定的项目例程，设置相应的硬件， 选择项目文件点击右键 New->Target Configurarion File 操作，如图。

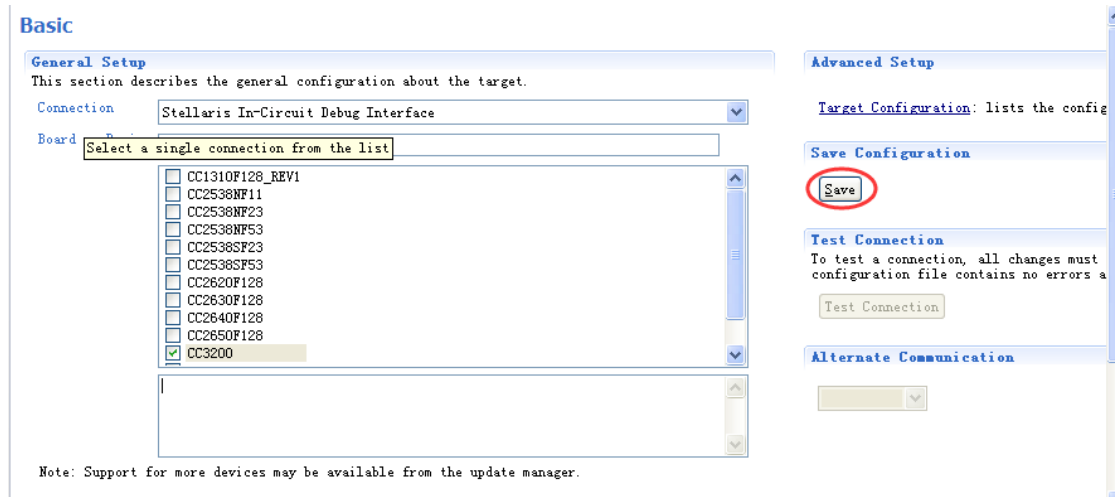




若不需要更改名称则直接点 Finish; 一般更改成自己的明确的目标板比较好, 更改后如图



二、点击 Finish 如图 1.7, Connection 选项框右边的下拉菜单选择仿真器的名称 Stellaris In-Circuit Debug Interface, Board or Device 中选择 CC3200, 选择好后点击 save。

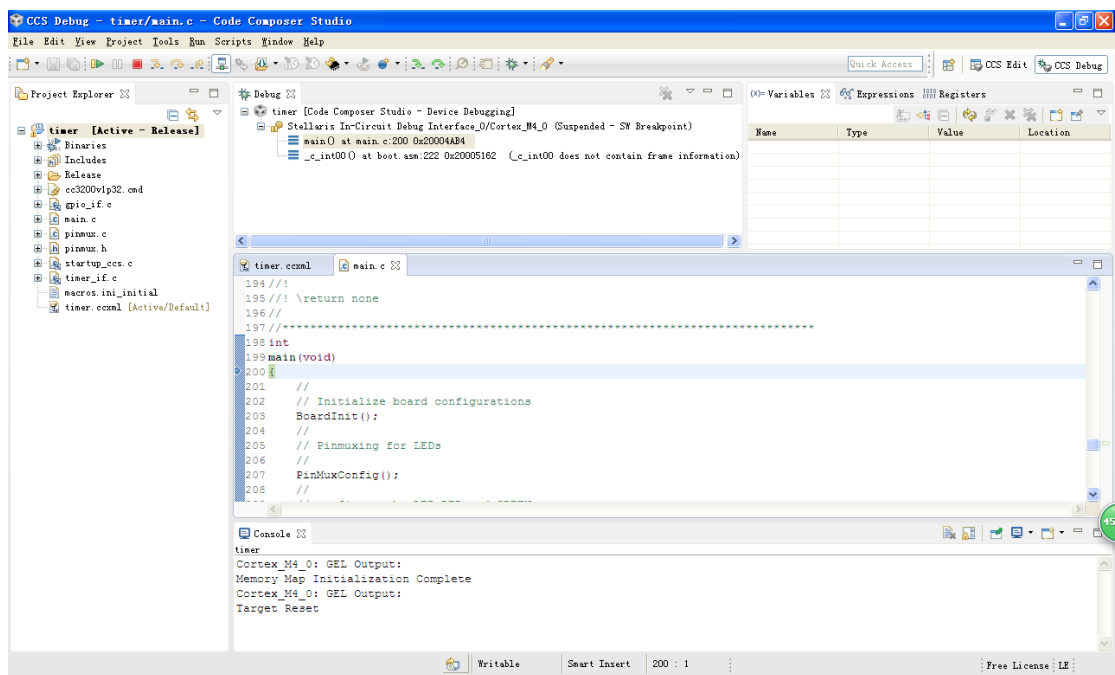


2.1.4 编译并下载程序

编译项目，先连接目标板并上电，然后点击工具栏上的绿色按钮，如图



如同 Project→Build Project，通过这个操作，激活状态的项目被编译，且 Target 和 CCS 自动连接并下载项目到 Target 中。当操作结束后整个界面跳转到 Debug 模式，如图



- (1) 运行参考程序(Flashing the bin/IAR/CCS)。
 - 打开应用程序；
 - 编译下载程序到目标板。
- (2) 目标板上的 LED 灯 (D3) 连续地亮灭。

3.2 CC3200 Timer Application

3.2.1 概述

每个 GPTM(通用定时器模块)模块可以作为两个 16 位定时器/计数器(称为定时器 A 和定时器 B),可以配置为独立运作模式来作为定时器或事件计数器,或联合起来作为一个 32 位定时器。

定时器也可以用来触发 uDMA 传输。

支持以下操作模式:

- 16 或 32 位的可编程一次性定时器;
- 16 或 32 位的可编程周期定时器;
- 16 位的通用定时器和一个 8 位预定标器。

3.2.2 应用说明

这个应用程序的目的是展示定时器 DriverLib 的 API 的使用。使用 16 位定时器产生中断从而切换 GPIO 的状态(led)。

两个具有不同超时值(一个是另一个的两倍)的定时器被设置去切换两个不同的接有 LED 灯的 GPIO,展示 LED 灯闪烁的效果。



3.2.3 使用说明

- (1) 运行参考程序(Flashing the bin/IAR/CCS)。
- (2) 观察 LED 灯不同的闪烁频率(一个是另一个的两倍)。

3.3 CC3200 Watchdog Application

3.3.1 概述

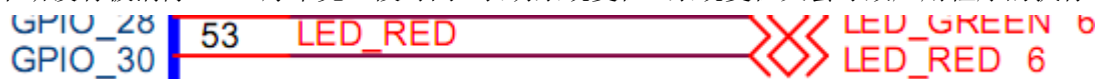
当超时值到达时,看门狗定时器产生一个中断或复位。看门狗定时器用于当操作系统由于软件错误或在预期的时间内外部设备回应失败时的重新控制。

看门狗定时器有以下特点:

- (1) 带有一个可编程加载寄存器的 32 位计数器;
- (2) 可编程中断可以产生中断屏蔽逻辑;
- (3) 调试期间,当微控制器维护 CPU 停止时,User-enabled 停滞。

3.3.2 应用说明

这个应用程序展示了看狗定时器(WDT)DriverLib 的 API 的使用,是为了展示当系统失败时看门狗定时器的复位功能,在这里,10 个计数值后没有清除看门狗中断,本质上是模拟系统故障的状况。看门狗定时器到期时,清除中断,LED 灯作出相应闪烁指示,十次之后,中断没有被清除,LED 灯常亮一段时间,表明系统复位。系统复位又会导致应用程序的执行。



3.3.3 使用说明

- (1) 运行参考程序(Flashing the bin/IAR/CCS)。
- (2) 观察红色 LED 灯去理解应用程序被执行的顺序

3.4 CC32xx UART Demo Application

3.4.1 概述

对 UART 功能设备的支持, 它有各种标准, 比如可编程波特率、单独传送和接收 FIFO、完全可编程单独接口特性等。

3.4.2 应用说明

这个应用的目的是为了展示 UART 的用处, 包括获取来自用户的输入信息和在终端上显示信息, 下面这个例子输入一行字, 按 enter, 然后原样显示出来。

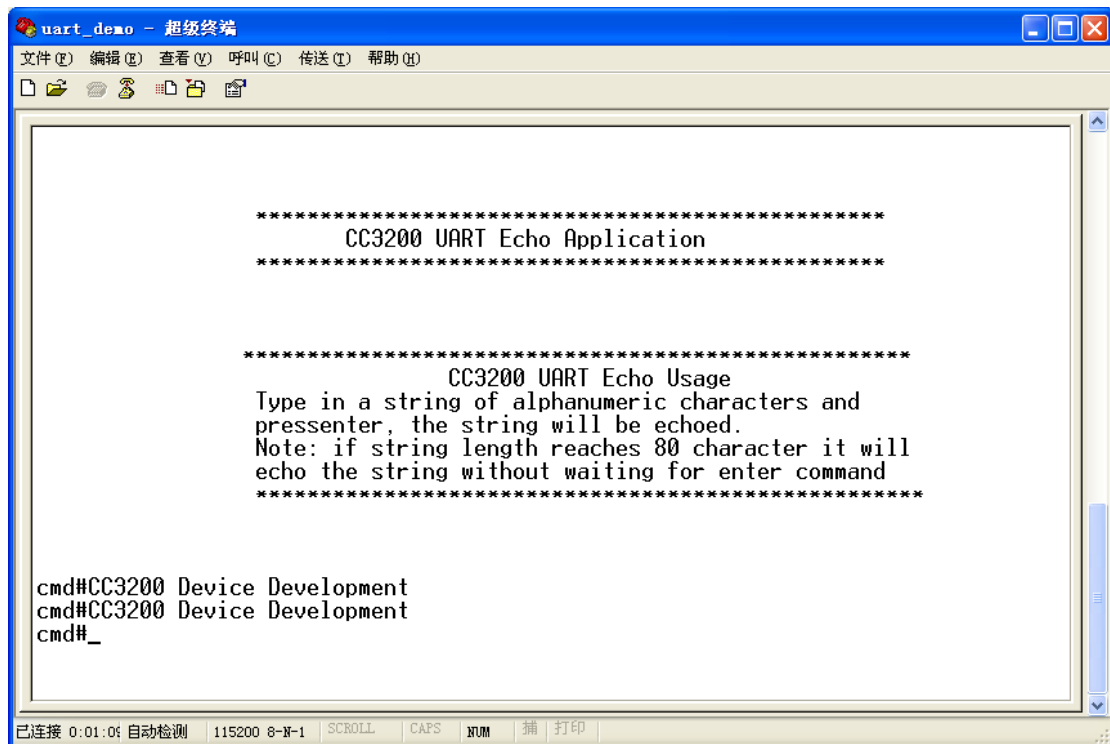
55	CC_GPIO01		GP0/CCF0 1
57	CC_GPIO02		GP1/U0TX 3,7
58	CC_GPIO03		GP2/U0RX 3.7

3.4.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



- (2) 运行参考程序(Flashing the bin/IAR/CCS)。
- (3) 观察终端上的状态信息理解应用程序执行的顺序。
- (4) 运行结果如下图所示:



3.5 CC3200 Interrupt Demo Application

3.5.1 应用说明

这个应用程序的目的是展示中断抢占和 tail-chaining 能力。当中断有相同的优先级时会合成嵌套中断。高优先级情况下，优先级抢占发生；在其他两个情况下 tail-chaining 将发生。

3.5.2 使用说明

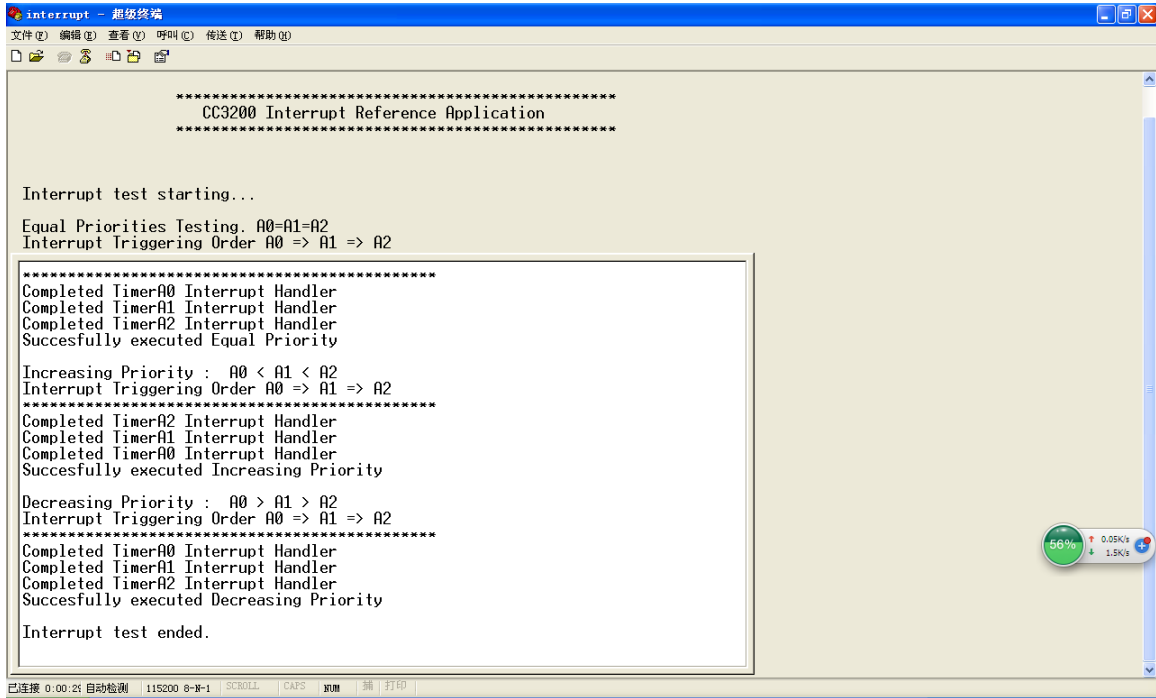
(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(2) 运行参考程序(Flashing the bin/IAR/CCS)。

(3) 观察终端上的状态信息理解应用程序执行的顺序。

(4) 运行结果如下图所示：



```
*****
CC3200 Interrupt Reference Application
*****

Interrupt test starting...

Equal Priorities Testing. A0=A1=A2
Interrupt Triggering Order A0 => A1 => A2

*****
Completed TimerA0 Interrupt Handler
Completed TimerA1 Interrupt Handler
Completed TimerA2 Interrupt Handler
Successfully executed Equal Priority

Increasing Priority : A0 < A1 < A2
Interrupt Triggering Order A0 => A1 => A2
*****
Completed TimerA2 Interrupt Handler
Completed TimerA1 Interrupt Handler
Completed TimerA0 Interrupt Handler
Successfully executed Increasing Priority

Decreasing Priority : A0 > A1 > A2
Interrupt Triggering Order A0 => A1 => A2
*****
Completed TimerA0 Interrupt Handler
Completed TimerA1 Interrupt Handler
Completed TimerA2 Interrupt Handler
Successfully executed Decreasing Priority

Interrupt test ended.
```

3.6 CC3200 Sleep-DeepSleep Application

3.6.1 概述

睡眠和深度睡眠是 CC3200 设备支持的两个低功率模式。在低功率模式下，通过对 PRCM 模块指定的寄存器的适当配置，不同的选项来选择模块活跃模式。因此，导致从低功率模式退出是以活跃的模块为基础的，对于其他模块的时钟，没有被选择为积极的，禁用。假如是深度睡眠模式时钟也减半。

不同时钟选通（启用、禁止）模块是：

- (1) Camera
- (2) McASP
- (3) MMCHS
- (4) McSPI
- (5) uDMA
- (6) GPIOs
- (7) WatchDOG
- (8) UART
- (9) GPT
- (10) Crypto
- (11) I2C

通过数值天气预报触发的 Wifi 网络活跃模式也可以唤醒模块。

3.6.2 应用说明

这个应用程序的目的是展示 CC3200 设备支持的两个模块的睡眠和深度睡眠。
在本例中选择的模块是：

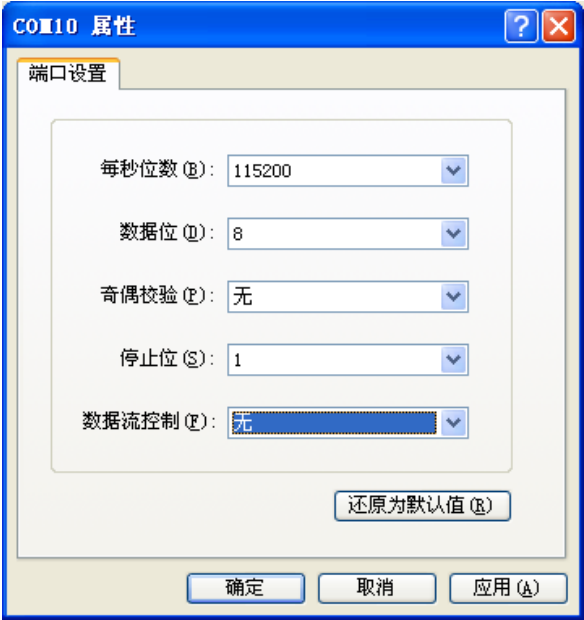
- (1) 基于看门狗定时器的睡眠和深度睡眠；

(2) 基于通用定时器 (GPT) 的睡眠和深度睡眠。

设备进入低电平模式执行无线指令，在设备进入低电平模式之前，有一些先决条件的设置被执行。

3.6.3 使用说明

(3) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(4) 运行参考程序(Flashing the bin/IAR/CCS)。

(5) 观察终端上的状态信息理解应用程序执行的顺序。

(6) 运行结果如下图所示：



3.7 CC3200 μ DMA Application

3.7.1 概述

μ DMA 控制器是一个灵活和高度可配置的 DMA 控制器，它具有微处理器 Cortex-M4 处理器核心设计的工作效率。它支持多个数据大小和地址增量计划，多个 DMA 通道的优先级别，和几个允许复杂的编程数据传输模式。

支持的不同传输模式是：

➤ **基本模式：**

在基本模式，只要提出传输请求，有更多的项目要传输， μ DMA 控制器就执行传输。每当外围设备准备数据传输，这种模式用来外设断言 μ DMA 请求信号。

➤ **自动模式：**

自动模式类似于基本模式，除了一旦收到传输请求时，即使 μ DMA 请求删除，也要将传输运行完成。这种模式适用于软件触发传输。一般来说，自动模式不用在外围设备。

➤ **乒乓模式：**

乒乓模式是用于支持发送连续数据流给外设或从外设传出连续数据流。使用乒乓球模式，主要的和备用的数据结构必须实现。结构都是由处理器建立，用于内存和外围设备的数据传输。传输开始时使用主控制结构，当传输使用的主要控制结构完成， μ DMA 控制器读取该通道的备用控制结构继续传输，每次这种情况发生的时候，产生一个中断，并且处理器为刚完成的传输重新加载控制结构。

➤ **分散收集模式：**

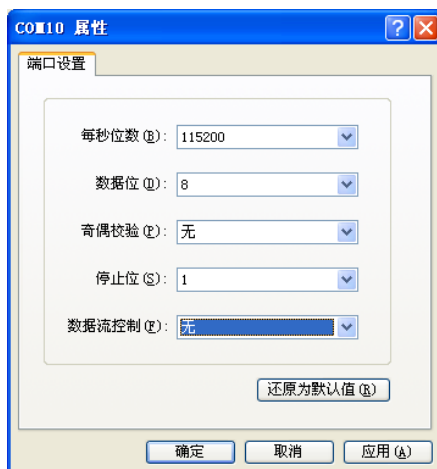
散集模式是一个复杂模式，当数据被转移到内存中或者在内存的不同位置传输才会被用到，不是一组连续的内存缓冲区地址。

3.7.2 应用说明

这个应用程序的目的是展示不同的 DMA 传输方式。演示了使用 μ DMA 控制器在内存缓冲区传输数据，用自动模式和散集模式，使用乒乓模式传输数据到 UART。在自动模式和散集模式下，一旦数据传输完成，数据传输的正确性就要被验证，在乒乓模式下，传输速率被计算并且显示。

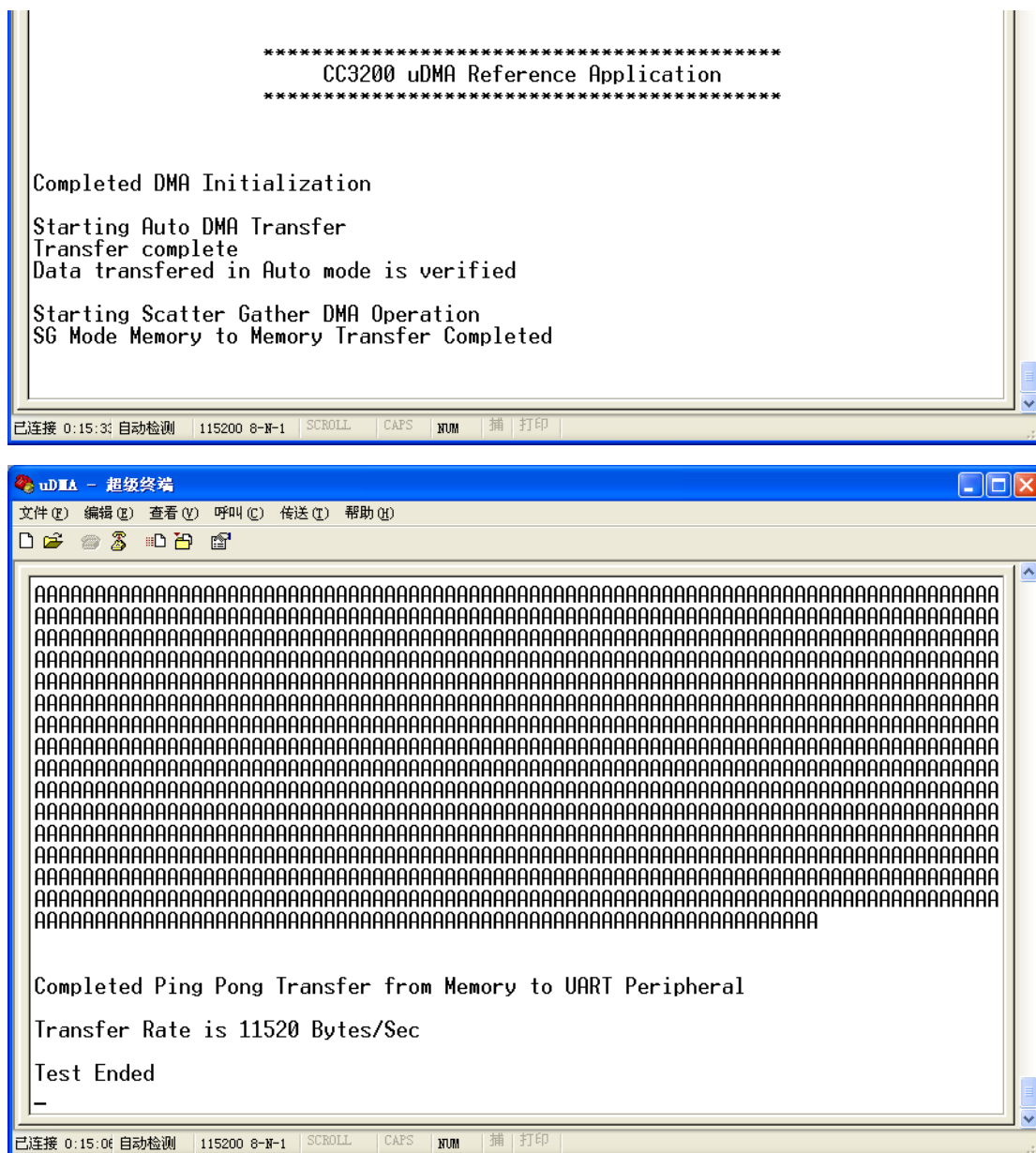
3.7.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(2) 运行参考程序(Flashing the bin/IAR/CCS)。

- (3) 观察终端上的状态信息理解应用程序执行的顺序。
- (4) 运行结果如下图所示：



3.8 CC32xx FreeRTOS Application

3.8.1 概述

FreeRTOS 是开源实时操作系统。它的实现只有一组简约的功能:基本任务处理和内存管理, 并且没有提供网络通信, 外部的硬件驱动程序, 或访问文件系统。但是, 它包括以下特点:

- 抢占任务;
- 占用空间小;
- 用 C 编写的, 并且用不同的 C 编译器编译;
- 无限数量的任务可以同时运行;
- 实现队列、二进制和计数信号量, 互斥;
- 互联网通信任务是使用队列来完成的。

3.8.2 应用说明

这个应用程序的目的是展示 FreeRTOS 特性,例如多任务创建,使用队列来完成互联网通信任务,本例程创建两个任务和一个队列,一个任务发送一个常量信息到队列中,另一个任务从队列中接收相同的信息,收到信息后,通过 UART 显示信息。

3.8.3 使用说明

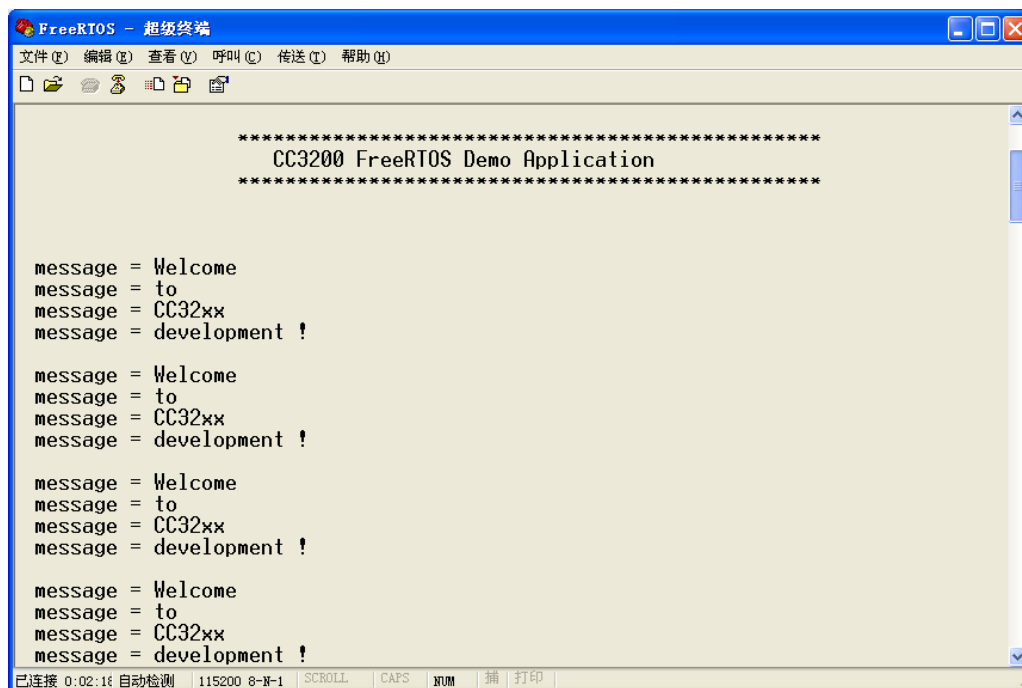
(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



(2) 运行参考程序(Flashing the bin/IAR/CCS)。

(3) 在终端上观察第二个任务的状态信息。

(4) 运行结果如下图所示:



3.9 CC3200 AES Demo Application

3.9.1 概述

推进加密标准 (AES) 安全模块, 提供硬件加速数据的加密和解密基于二进制的 AES 是一个对称密码模块, 它支持 128, 192, 或 256 位密钥加密和解密的硬件。AES 模块是基于对称算法, 这意味着加密和解密密钥是相同的。加密数据意味着从纯文本转换为一种特殊的形式称为密文。解密密文意味着将加密的数据改回到原来的纯文本形式。

3.9.2 应用说明

应用程序的引用使用 AES DriverLib 函数。开发人员/用户可以参考这个简单的应用程序和重用这个功能在应用程序。这个应用程序在有或没有“Uart 终端”的情况下都可以使用。

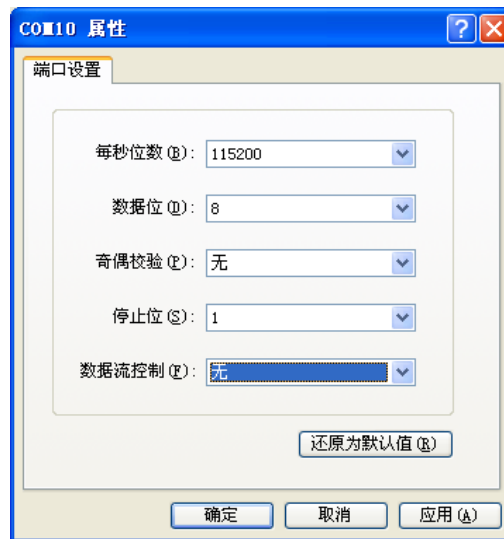
如果用户希望使用“Uart 终端”输入, 按照下面操作, 通过定义“USER_INPUT”在 aes_main.c 文件中。

- aesdemo: 此命令允许用户运行 AES 功能。需要两个参数 aes-mode 和 key-len。
 - aes-mode 是用户可以选择的 AES 算法, 值可以是 ECB、CBC、CTR、ICM 或者 CFB。
 - key-len 是用户需要定义的长度, 值可以使 128、192、或者 256。

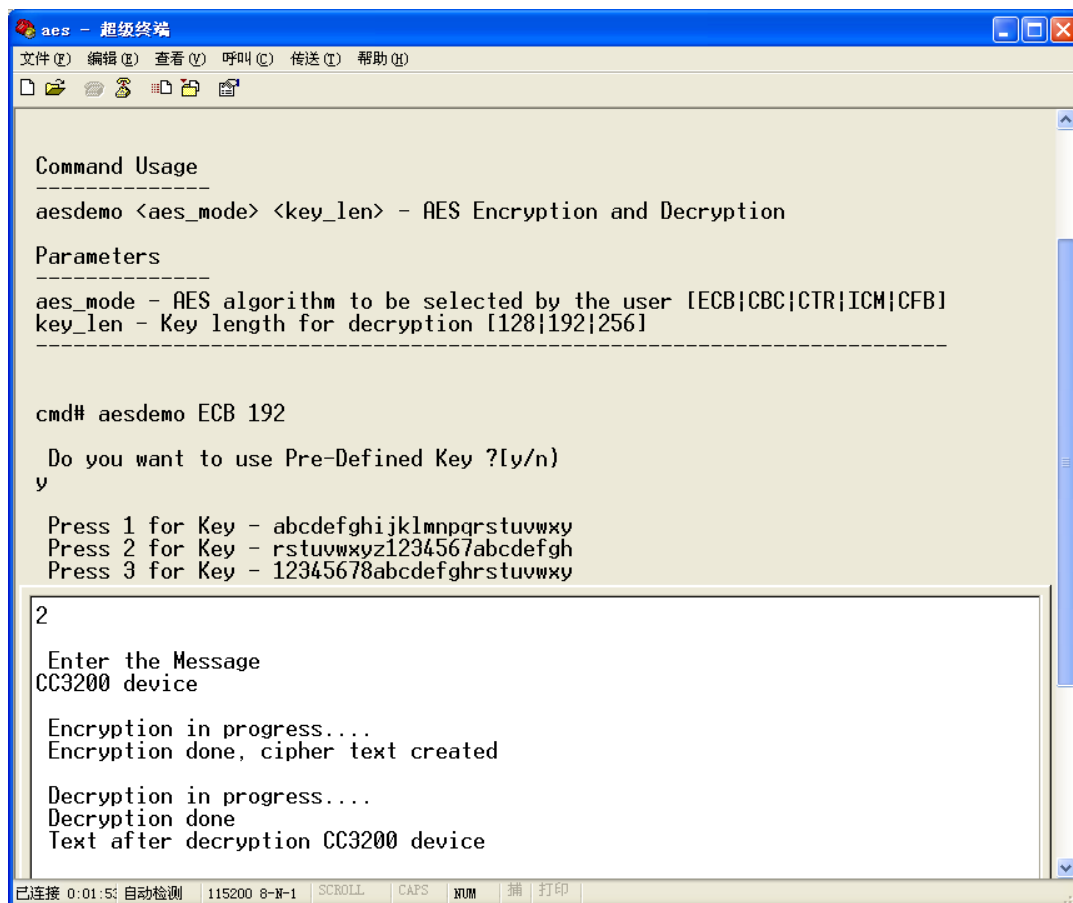
接下来, 系统将提示用户输入” key”和” 纯文本输入”, 没有定义 USER-INPUT, 将允许用户按照执行路径在 IAR 或 CCS 集成开发环境, 在“调试”模式, 用户不需要输入

3.9.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



- (2) 运行参考程序(Flashing the bin/IAR/CCS)。
- (3) 在超级终端上, 出现一个提示:
需要下达 AES 命令, 可以看到结果。
- (4) 运行结果如下图所示:



```

aes - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

Command Usage
-----
aesdemo <aes_mode> <key_len> - AES Encryption and Decryption

Parameters
-----
aes_mode - AES algorithm to be selected by the user [ECB|CBC|CTR|ICM|CFB]
key_len - Key length for decryption [128|192|256]
-----

cmd# aesdemo ECB 192

Do you want to use Pre-Defined Key ?[y/n]
y

Press 1 for Key - abcdefghijklmnopqrstuvwxyz
Press 2 for Key - rstuvwxyz1234567abcdefgh
Press 3 for Key - 12345678abcdefghrstuvwxyz

2

Enter the Message
CC3200 device

Encryption in progress....
Encryption done, cipher text created

Decryption in progress....
Decryption done
Text after decryption CC3200 device

已连接 0:01:53 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

3.10 CC3200 DES Demo Application

3.10.1 概述

DES 模块提供了硬件加速的数据加密和解密功能。模块运行单一 DES 或三倍 DES (3 DES) 算法符合 FIPS 46-3 标准和支持以下操作模式：

- 电子码 (ECB)
- 密码块链接 (CBC)
- 密码反馈 (CFB)

基于对称算法 DES 模块, 这意味着加密和解密密钥是相同的。加密数据意味着从纯文本转换为一种特殊的形式称为密文。解密密文意味着将加密的数据改回到原来的纯文本形式。

3.10.2 应用说明

应用程序的引用使用 DES DriverLib 函数。开发人员/用户可以参考这个简单的应用程序和重用这个功能在应用程序。这个应用程序在有或没有“Uart 终端”的情况下都可以使用。

如果用户希望使用“Uart 终端”输入, 按照下面知识操作, 通过定义“USER_INPUT”在 des_main.c 文件中。

- desdemo: 此命令允许用户运行 DES 功能。需要一个参数 des-mode:
 - des-mode 是用户可以选择的 DES 算法, 值可以是 ECB、CBC、CFB、TECB、TCBC 或者 TCFB。

接下来，系统将提示用户输入”key”和”纯文本输入”，没有定义USER-INPUT，将允许用户按照执行路径在IAR或CCS集成开发环境，在“调试”模式，用户不需要输入。

3.10.3 使用说明

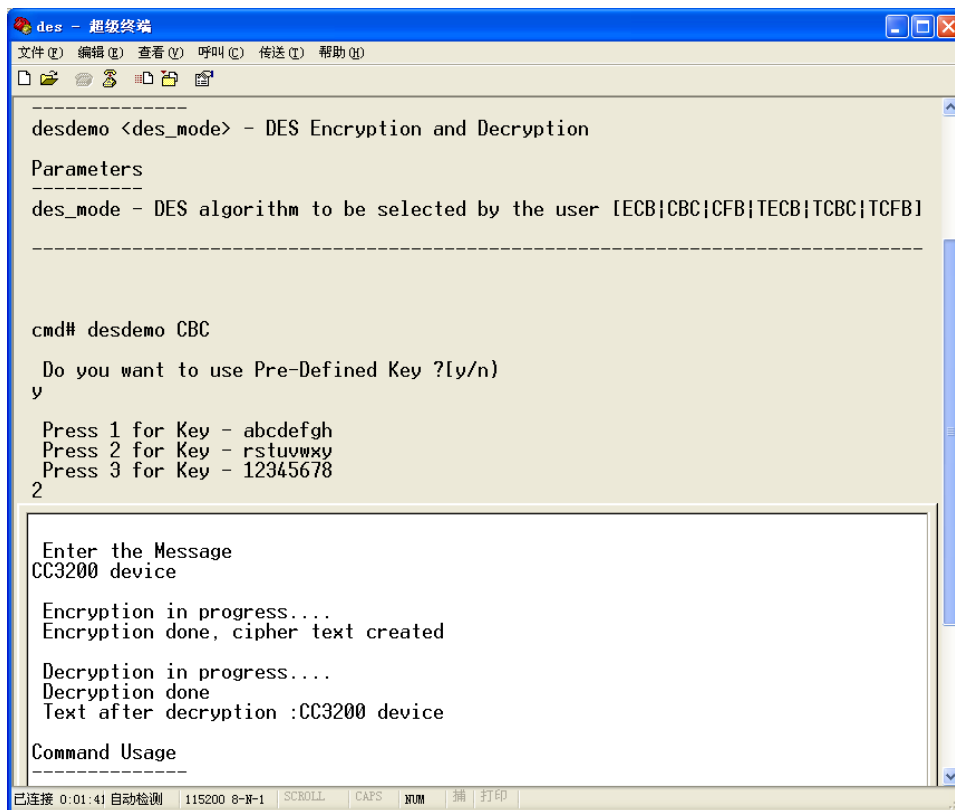
- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 运行参考程序(Flashing the bin/IAR/CCS)。

- (3) 在超级终端上，出现一个提示：
需要下达DES命令，可以看到结果。

- (4) 运行结果如下图所示：



3.11 CC3200 CRC Demo Application

3.11.1 概述

循环冗余校验(CRC)计算模块可用于消息传输和安全系统检查。CRC 引擎的目的是加速 CRC 和 TCP 校验和操作。CRC 操作的结果是一个 32 位和 16 位信号,可以用来检查数据的完整性。支持以下特性:

四个主要的 CRC 形式:

- CRC16-IBM as used by USB and ANSI;
- CRC32-IEEE as used by IEEE 802.3 and MPEG-2;
- CRC32C as used by G.Hn;
- CRC16-CCITT as used by CCITT/ITU X.25;
- 校验和 (TCP) ;
- 字和字节的提供;
- 字节、半字输入和输出数据的交换;
- 位反转。

3.11.2 应用说明

应用程序的引用使用 CRC DriverLib 函数。开发人员/用户可以参考这个简单的应用程序和重用这个功能在应用程序。这个应用程序在有或没有“Uart 终端”的情况下都可以使用。

如果用户希望使用“Uart 终端”输入,按照下面知识操作,通过定义“USER_INPUT”在 des_main.c 文件中。

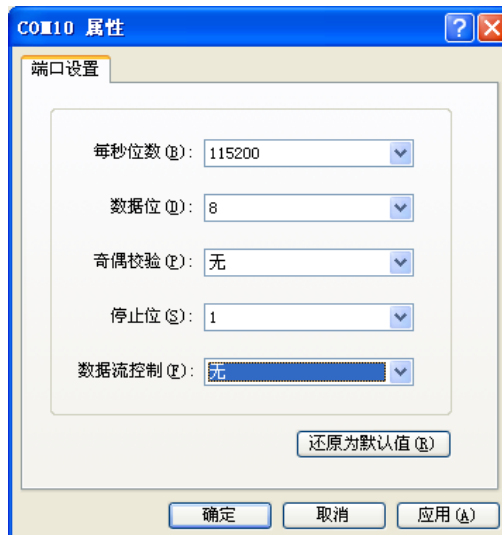
- crcdemo: 此命令允许用户运行 CRC 功能。需要一个参数 crc-mode:
 - crc-mode 是用户可以选择的 CRC 算法,值可以是 P8005、P1021、P4C11DB7、P1EDC6F41、或者 TCPCHKSUM。

P8005 - CRC-16-IBM 0x8005
P1021 - CRC-16-ICCITT 0x1021
P4C11DB7 - CRC-32 0x4C11DB7
P1EDC6F41 - CRC-32 0x1EDC6F41
TCPCHKSUM - TCP Checksum

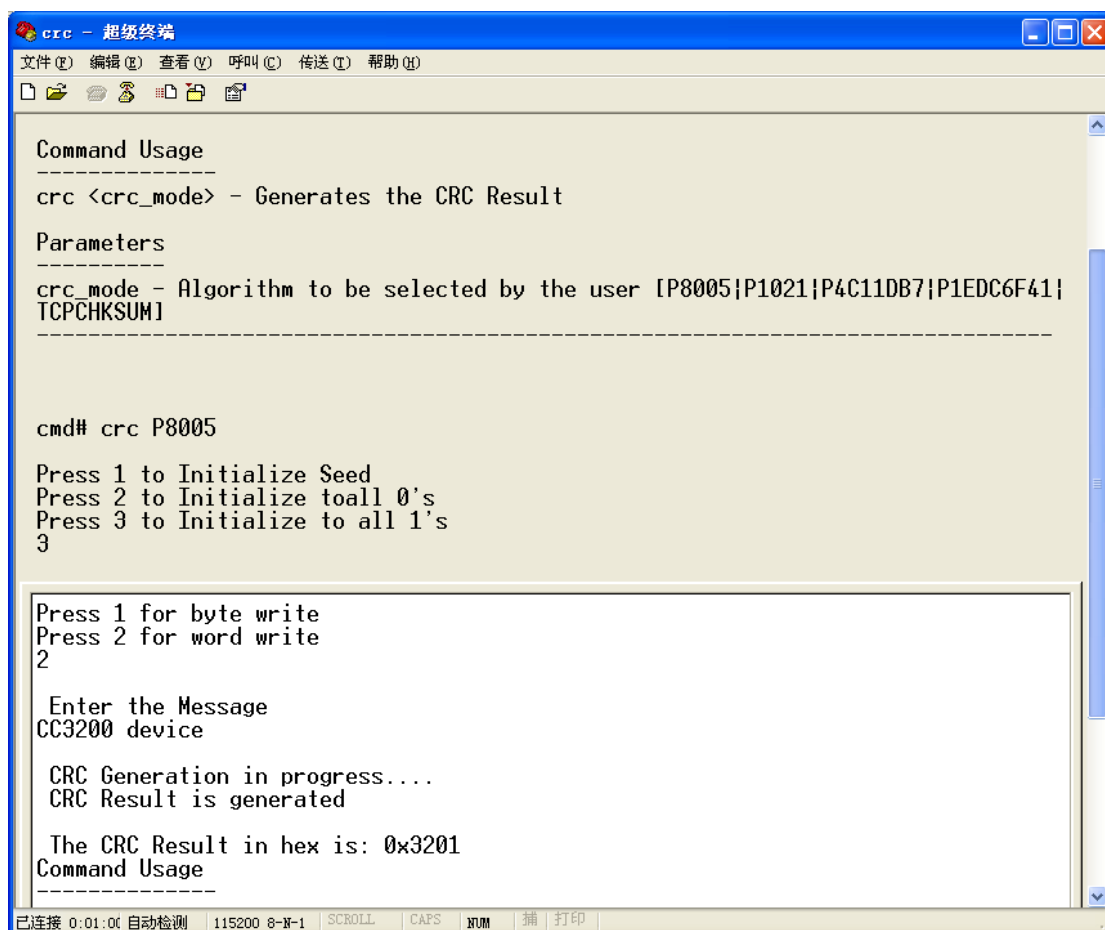
接下来,用户将被提示输入。没有定义 USER-INPUT,将允许用户按照执行路径在 IAR 或 CCS 集成开发环境,在“调试”模式,用户不需要输入。

3.11.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



- (2) 运行参考程序 (Flashing the bin/IAR/CCS)。
- (3) 在超级终端上，出现一个提示：
需要下达 CRC 命令，可以看到结果。
- (4) 运行结果如下图所示：



3.12 CC3200 SHA-MD5 Demo Application

3.12.1 概述

SHA/MD5 模块提供硬件加速散列函数,可以运行:

- MD5 消息摘要算法，Ron Rivest 在 1991 年开发的；
- SHA-1 算法符合 FIPS 180 - 3 标准；
- 算法产生一个压缩消息或数据文件的表示，称为摘要，可以用于验证消息的完整性；
- 0 到 233 的散列 - 2 字节的数据(232 - 1 字节是一部分)使用 MD5、SHA-1、SHA-224,或 SHA-256 散列算法；
- 自动 HMAC 的键的键预处理,最高可达 64 个字节；
- Host-assisted HMAC 的键的键预处理大于 64 字节；
- HMAC 从预先执行(内部/外部消化)改进的性能在小块；
- 支持 μ DMA 操作数据和上下文/结果传输；
- 支持中断阅读摘要。

3.12.2 应用说明

应用程序的引用使用 SHAMD5 DriverLib 函数。开发人员/用户可以参考这个简单的应用程序和重用这个功能在应用程序。这个应用程序在有或没有“Uart 终端”的情况下都可以使用。

如果用户希望使用“Uart 终端”输入，按照下面知识操作，通过定义“USER_INPUT”在 shamd5_main.c 文件中。

- hash: 此命令允许用户运行 SHAMD5 功能。需要一个参数 shamd5-mode:
 - shamd5-mode 是用户可以选择的 SHAMD5 算法，值可以是 MD5、SHA1、SHA224、SHA256、HMAC_MD5、HMAC_SHA1、HMAC_SHA224 或者 HMAC_SHA256。

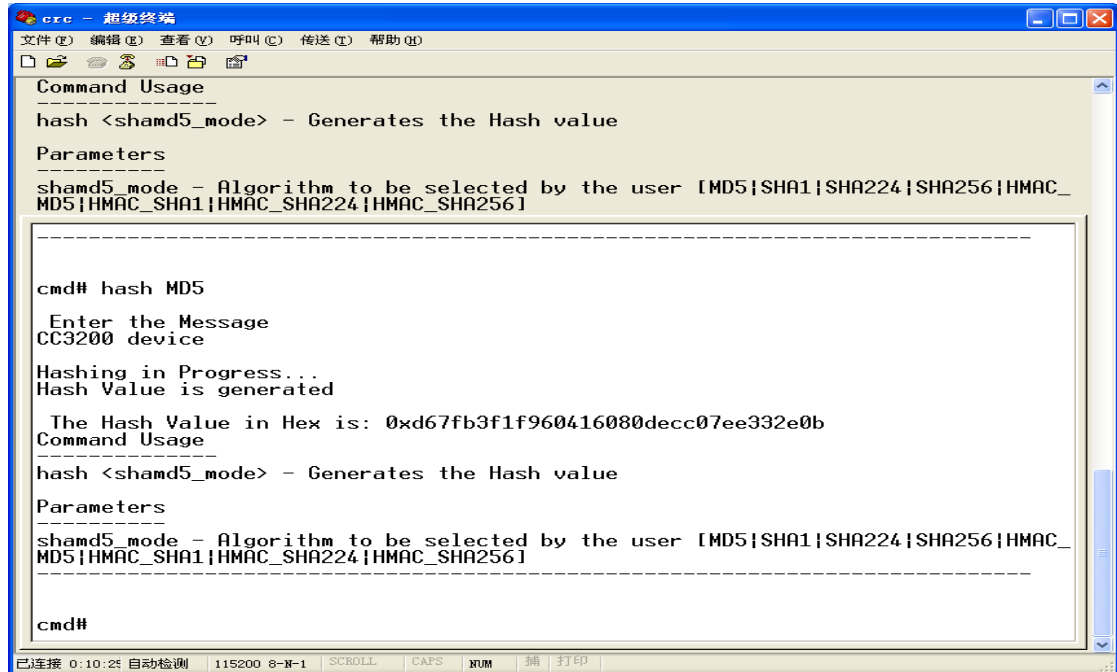
接下来,用户将被提示输入。没有定义 USER-INPUT, 将允许用户按照执行路径在 IAR 或 CCS 集成开发环境, 在“调试”模式, 用户不需要输入

3.12.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 运行参考程序(Flashing the bin/IAR/CCS)。
- (3) 在超级终端上，出现一个提示：
 - 需要下达 SHA-MD5 命令，可以看到结果。
- (4) 运行结果如下图所示：



3.13 CC3200 ADC

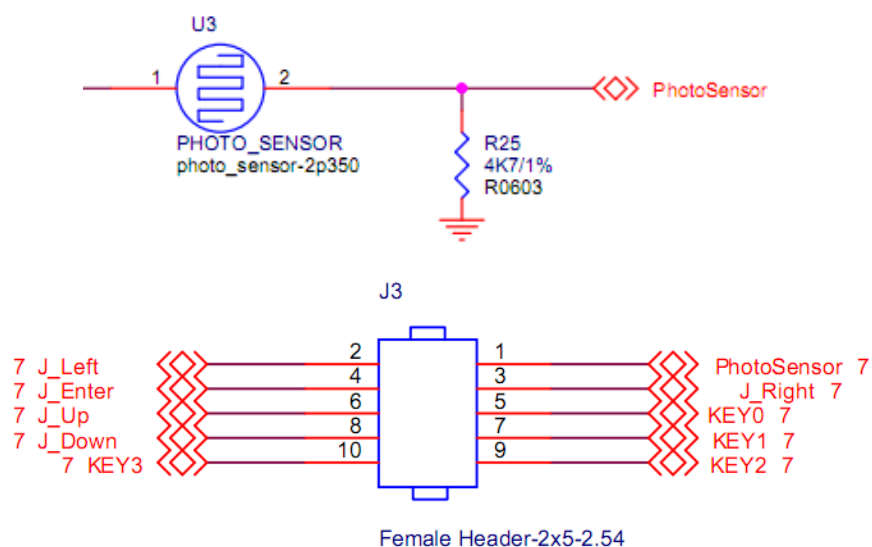
3.13.1 概述

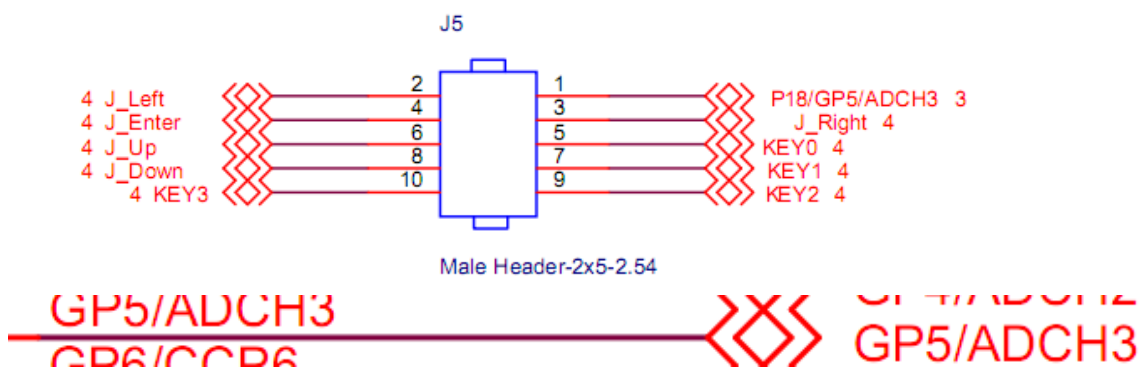
ADC 外围将连续的模拟电压转换为离散的数字号码。CC3200R 装置包括四个输入通道的 ADC 模块。每个 ADC 模块具有四个输入通道 12 位转换解决方案的特点。它每通道支持的采样率 16μs。

3.13.2 应用说明

这个应用程序的目的是展示 ADC DriverLib 的 API 的使用。是使用内部的 ADC 来采集光敏电阻的电压。

原理图:





3.13.3 使用说明

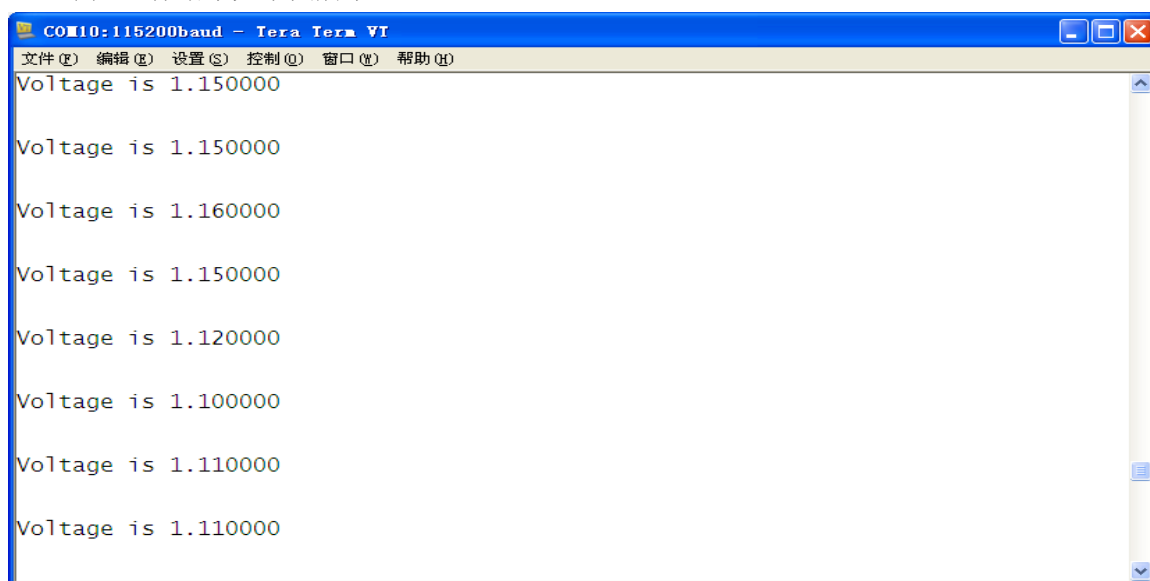
(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(2) 运行参考程序(Flashing the bin/IAR/CCS)。

(3) 在超级终端上，出现一个提示：
需要下达 ADC 命令，可以看到结果。

(4) 运行结果如下图所示：



3.14 CC3200 PWM

3.14.1 概述

通用定时器 (gpt) 支持 16 位脉冲宽度调制 (PWM) 模式，带 8 位预分频器和软件可编程输出反相。

驱动 led 的引脚具有多路复用去驱动 PWM 输出的能力。在这个例子中，通过适当的不同的责任周期，LED 灯从灭到亮，它的亮度是不同的，。

3.14.2 应用说明

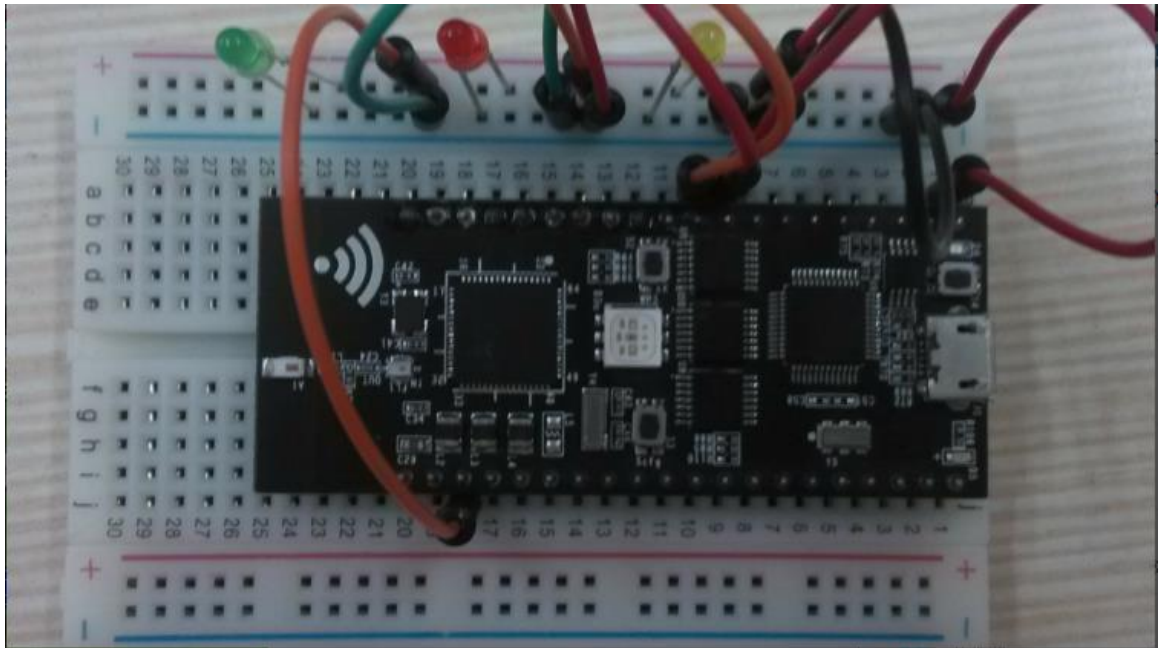
应用程序使用下面的引脚针来驱动 PWM 输出：

- PIN_64 for TIMERPWM5
- PIN_01 for TIMERPWM6
- PIN_02 for TIMERPWM7

在目标板上的以上引脚来外接三个 LED 灯，LED 灯的亮度随着不同的责任周期的输出而改变，这个应用程序产生 0.5s PWM 输出时间，责任周期不断变化从而改变 LED 灯的亮度。

3.14.3 使用说明

- (1) 在 PIN_64、PIN_01、PIN_02 三个引脚上外接 3 个 LED 灯。



- (2) 运行参考程序 (Flashing the bin/IAR/CCS)。
- (3) 3 个 LED 灯的亮度同时逐步改变从低（灭）到高（亮），不断重复。

3.15 CC3200 SPI Demo

3.15.1 概述

串行外围接口 (SPI) 是同步串行接口允许全双工设备与外部 SPI 兼容的主设备或从设备通信。SPI 设备操作在主从模式下，主设备发起通信，可以运行在两个模式 4 线或 3 线模式。SPI 指定以下四种信号：

- SCLK：串行时钟从主设备到从设备；
- MOSI：串行数据从主设备到从设备；

MISO: 串行数据从从设备到主设备;
CS: 芯片选择(只在 4 线模式)。

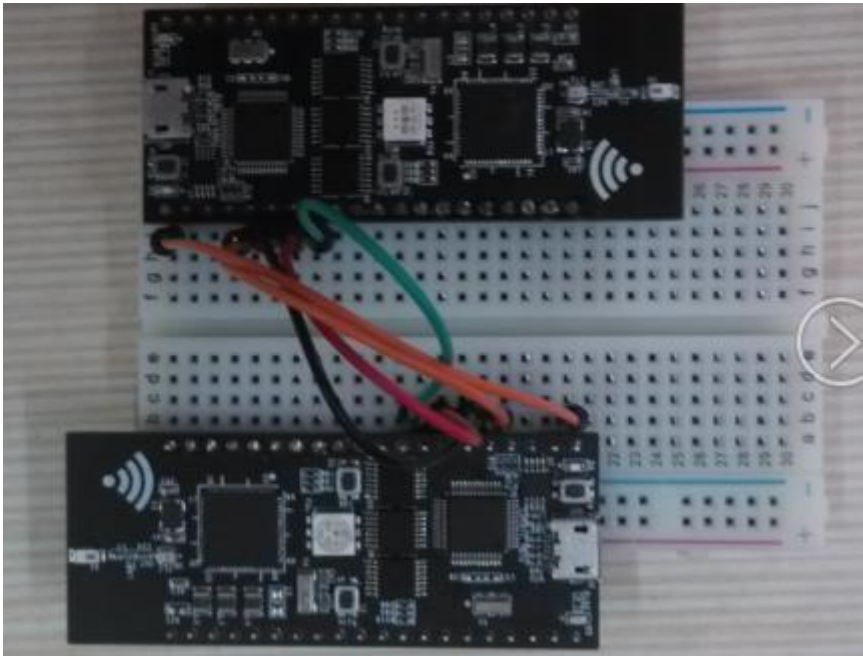
3.15.2 应用说明

这个应用程序重点在于展示在全双工四线主从模式使能 CC3200 SPI 模块所需的初始化顺序。

3.15.3 使用说明

(1) 连接两个目标板如下所示，其中一个目标板做为 SPI 的主设备，另一个作为 SPI 的从设备。

目标板 1	目标板 2	引脚名称
P1. 7	P1. 7	SCLK
P2. 3	P2. 3	CS
P2. 6	P2. 6	MOSI
P2. 7	P2. 7	MISO
P2. 1	P2. 1	GND



(2) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



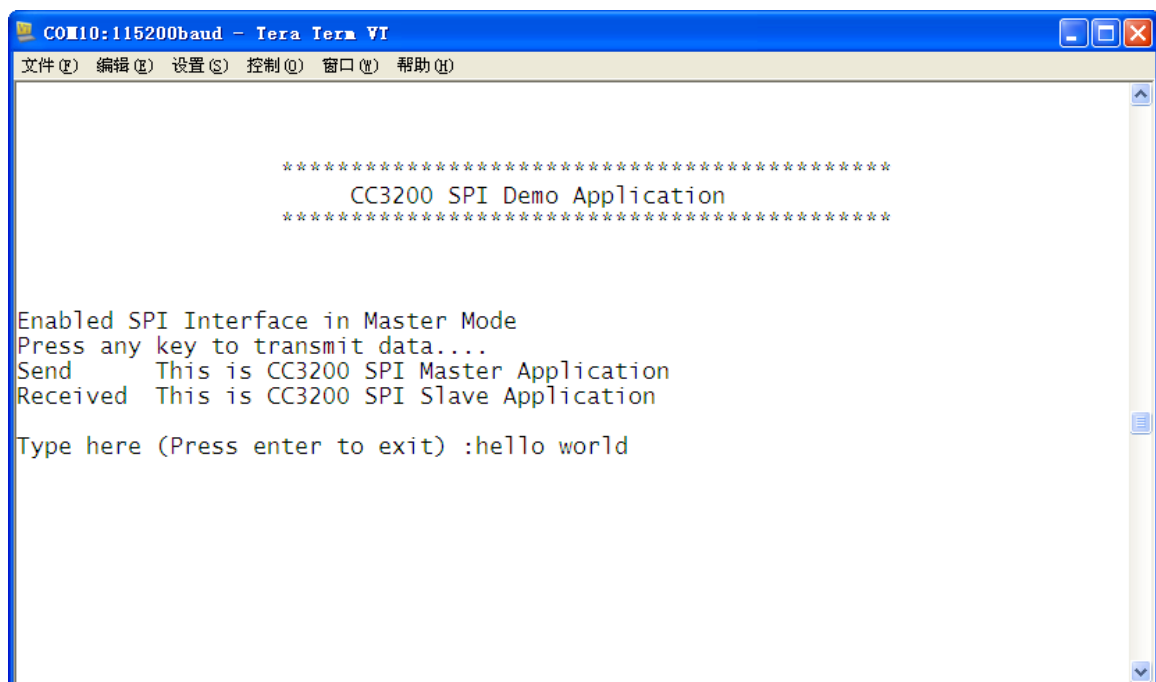
(3) 运行应用程序 (Flashing the bin/IAR/CCS) 在目标板上, 一个编译为主模式, 另一个用于从模式通过改变下面的宏定义:

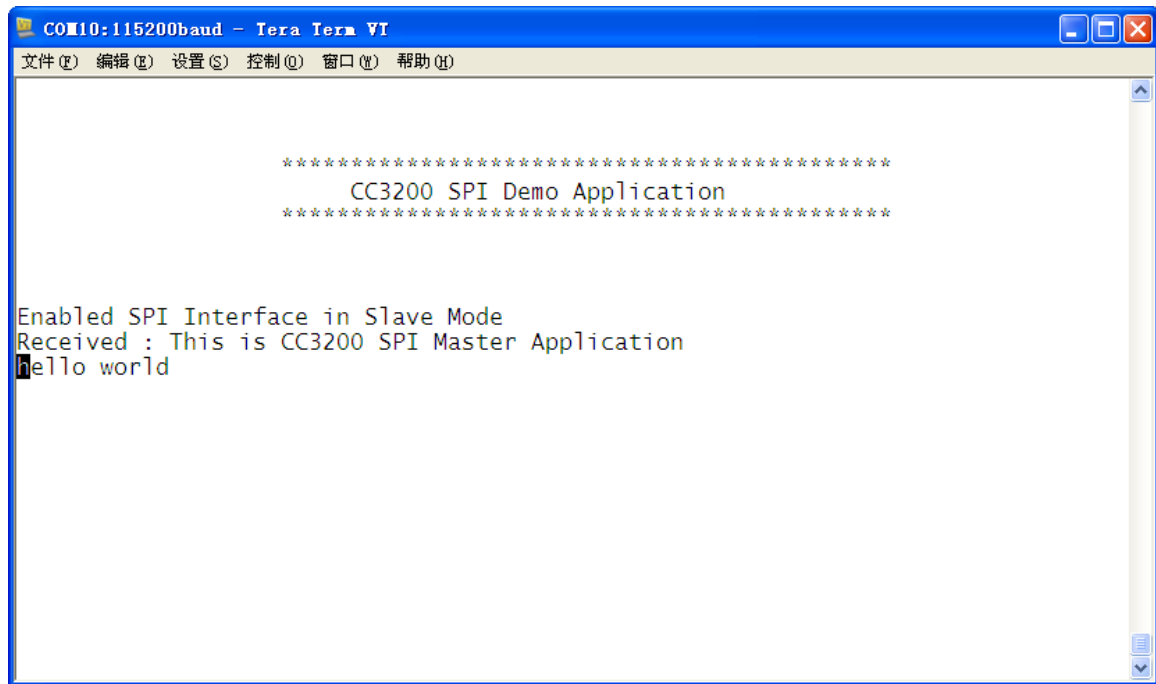
```

//*****
//
// Application Master/Slave mode selector macro
//
// MASTER_MODE = 1 : Application in master mode
// MASTER_MODE = 0 : Application in slave mode
//
//*****
#define MASTER_MODE      0

```

(4) 下面是 UART 终端显示。





3.16 CC3200 UART DMA Application

3.16.1 概述

设备的硬件支持 UART 的功能。，它有各种标准，比如可编程波特率、单独传送和接收 FIFO、完全可编程单独接口特性等。

3.16.2 应用说明

这个应用程序的目的是展示 UART 的 μ DMA 和中断使用。包括获取来自用户的输入信息和在终端上显示信息。这个例子包括将用户输入的 8 个字符传输到本地缓冲区使用 μ DMA Rx 通道。本地缓冲区接收到 8 个字符后, 使用 μ DMA Tx 通道通过 UART 将字符发送到终端。



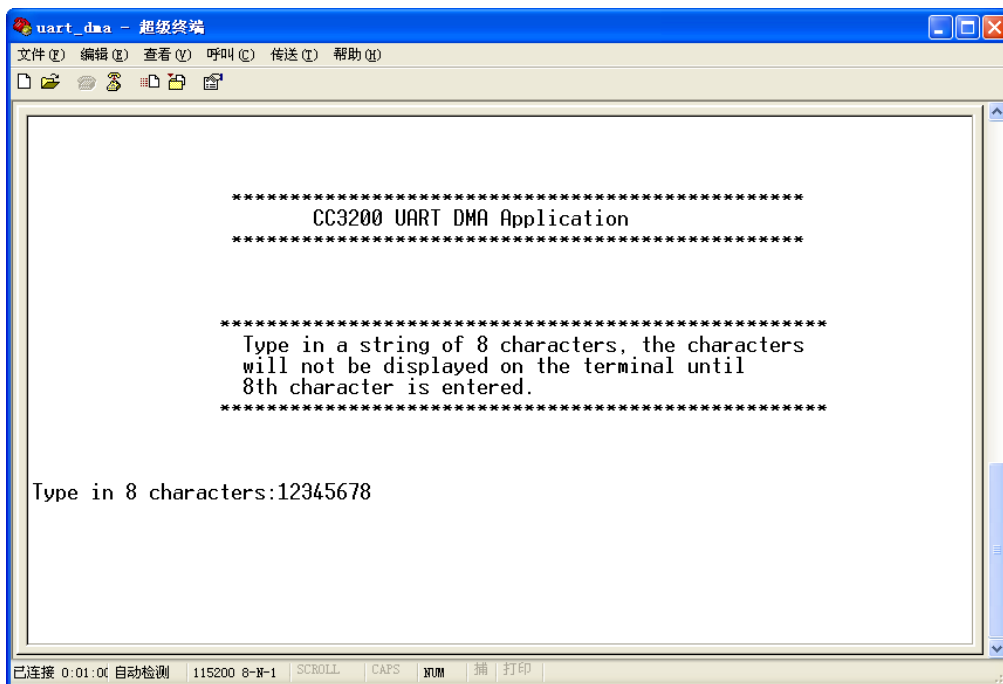
3.16.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(2) 运行应用程序 (Flashing the bin/IAR/CCS)

(3) 终端显示:



3.17 CC3200 Timer Count Capture Application

3.17.1 概述

每个 GPTM (通用定时器模块) 块可以作为两个 16 位定时器/计数器 (称为定时器 A 和定时器 B), 可以配置为独立运作来作为定时器或事件计数器, 或联合起来作为一个 32 位定时器。

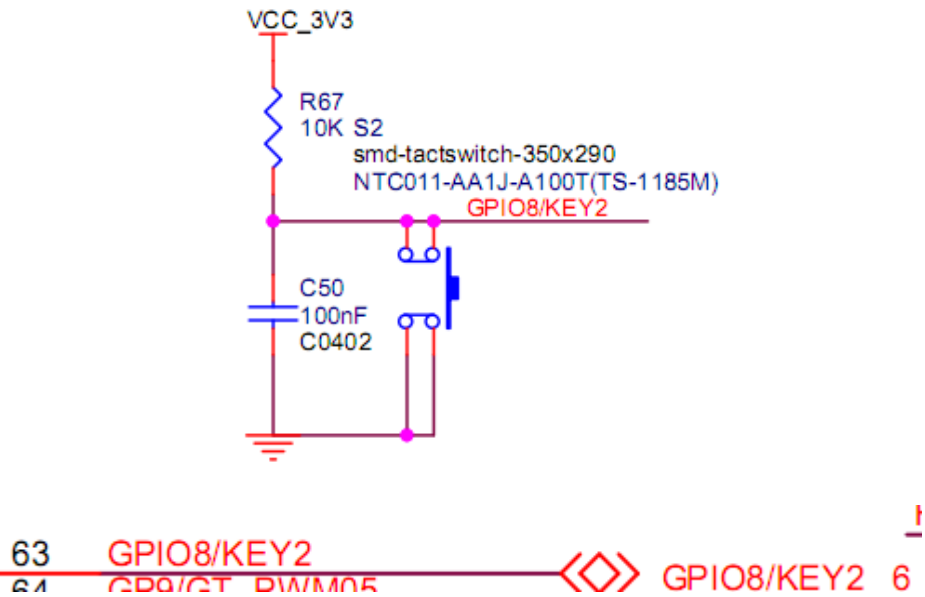
定时器也可以用来触发 μ DMA 传输。

支持以下操作模式:

- 16 或 32 位的可编程一次性定时器;
- 16 或 32 位的可编程周期定时器;
- 16 位的通用定时器和一个 8 位预定标器。

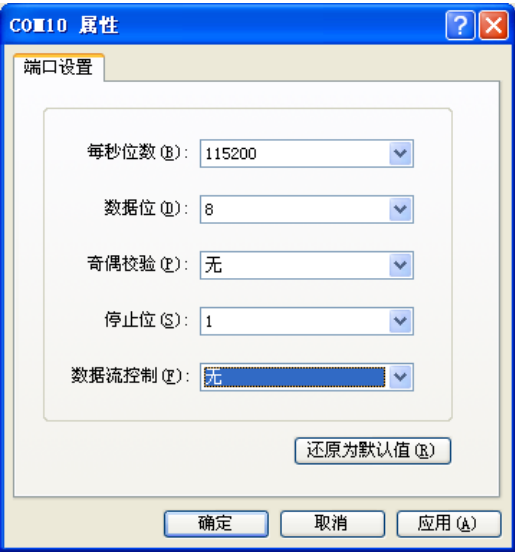
3.17.2 应用说明

这个应用程序展示了使用定时器的计数捕捉功能来衡量外部信号的频率。

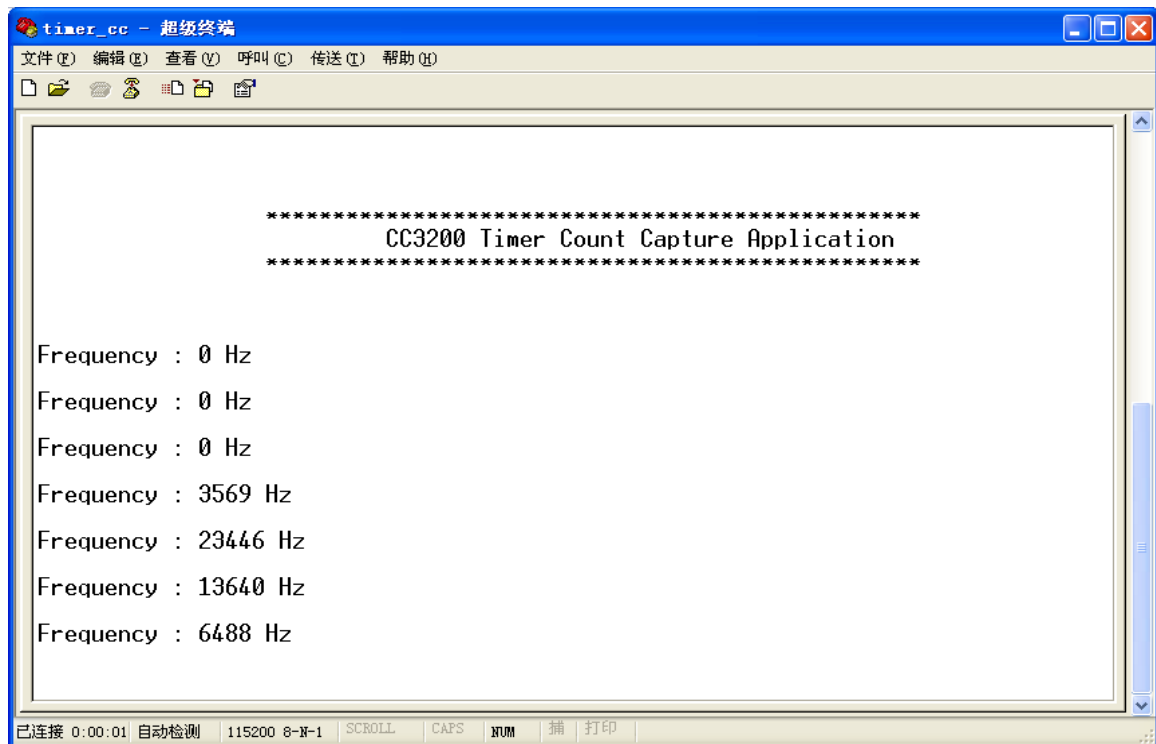


3.17.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 运行应用程序(Flashing the bin/IAR/CCS)。
- (3) 给一个外部信号在目标板 PIN_62。
- (4) 在终端上，观察得到的频率。
- (5) 显示结果如下：



3.18 CC3200 Dynamic Library Loader

3.18.1 概述

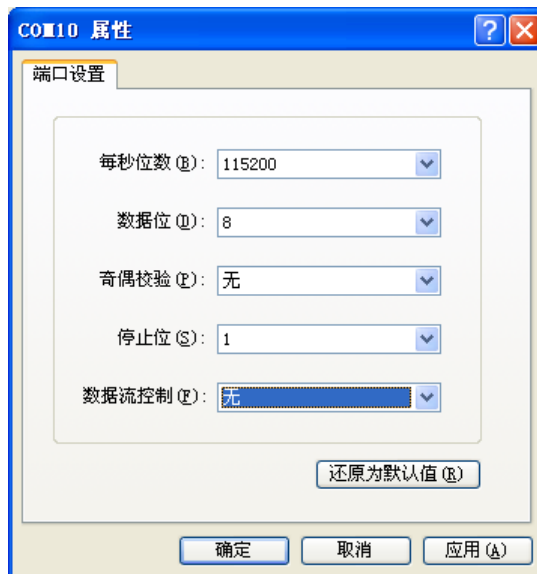
这个应用程序主要演示在程序运行时，对于有公共接口的 API 但实现目的不同时，该如何挂接不同的库。在一定条件下，对于有限的内存系统，在运行时如果应用程序要加载库到内存，那么这个特点是非常重要的。

3.18.2 应用说明

这是一个非常基本的应用程序，两个有相同的 API 接口的不同的库一个接一个地被加载，紧随其后的是它们的调用顺序。这两个库的实现不同，这样给它们相同的参数调用这两个库时会有不同的输出。

3.18.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 使用 uniflash 下载库并且加载应用程序(提供 IAR 和 CCS 的 ucf 文件)。
使用 uniflash 下载时, SW1 拨码开关拨到英文字母的方向。运行时, 拨到数字方向。
- (3) 运行从 debug 或 flash 中加载的程序。
- (4) 终端上显示加载动态库并且调用相同的 API, 但是有不同的输出:

```
Application Name: Dynamic Lib Loader
*****

Loading library 1.. /tmp/app1.bin
Library loaded successfully
Calling common APIs..
This function returns summation of the arguments
AddNums(40, 10): 50
This function returns difference of arguments (first - second)
SubNums(40, 10): 30

Loading library 2.. /tmp/app2.bin
Library loaded successfully
Calling common APIs..
This function returns twice the summation of the arguments
AddNums(40, 10): 100
This function returns twice the difference of arguments 2*(first - second)
SubNums(40, 10): 60

Application Tested Successfully
Exiting Application
```

已连接 0:03:38 ANSI 115200 8-N-1 SCROLL CAPS NUM 捕 打印

四、CC3200 WIFI 实验

4.1 CC3200 Connection Policy

4.1.1 概述

连接政策决定如何连接到 AP:

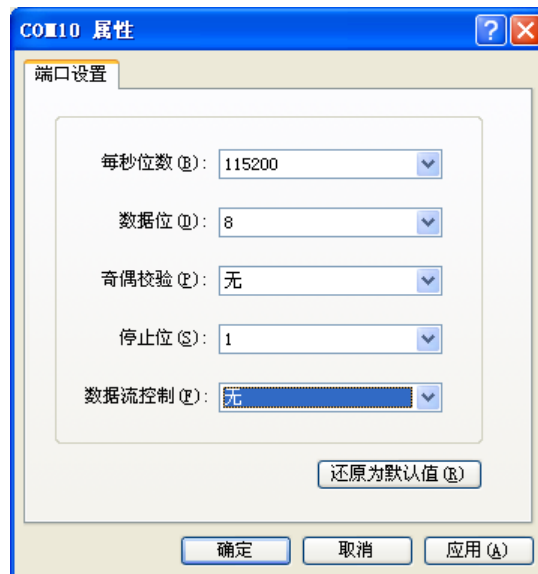
-auto
-fast
-smartconfig

4.1.2 应用说明

这个应用程序演示了使用连接策略。应用程序将根据用户的输入去使用对应的连接策略。

4.1.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



(2) 运行应用程序(Flashing the bin/IAR/CCS)

(3) 在终端上会出现提示需要用户输入

(4) 终端显示:

```
COM10:115200baud - Tera Term VI
文件(F) 编辑(E) 设置(S) 控制(C) 窗口(W) 帮助(H)

*****
CC3200 connection_policy Application
*****

AP NAME:      cc3200demo
AP Password:   12345678
security type: WPA

Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.4.1.1.5.3.3
Device is configured in default state
Device started as STATION

Please Choose Connect Policy
1.Auto
2.Autosmartconfig
3.Fast
4.Exit
Please enter the number of connection policy:1

device connect to AP in auto mode

[WLAN EVENT] STA Connected to the AP: cc3200demo ,BSSID: 0:27:19:23:4c:56
[NETAPP EVENT] IP Acquired: IP=192.168.100.113 , Gateway=192.168.100.1
[WLAN EVENT]Device disconnected from the AP: cc3200demo,BSSID: 0:27:19:23:4c:56
on application's request

Please Choose Connect Policy
1.Auto
2.Autosmartconfig
3.Fast
4.Exit
Please enter the number of connection policy:
```

4.2 CC32xx File Operations

4.2.1 概述

这个例子表明应用程序可以执行文件的操作。应用程序使用 serial-flash 作为存储介质。CC3200 目标板有 1 MB Sflash，下面给出了页、扇区、块的大小：

页 - 256 字节

扇区 - 4Kb

块 - 64Kb

在 SFlash 文件系统中，simplelink 库公开的 API 通过存储在 SFlash 中的内容去执行文件的操作，这个应用程序展示了 API 的使用。

4.2.2 应用说明

应用程序如下：

- 打开用户文件并且写文件
- 写“Old MacDonalds”儿歌多次得到一个小于 64KB 的文件
- 关闭用户文件
- 再次打开用户文件并且读文件
- 读数据并且和存储缓冲区中的作比较
- 关闭用户文件

- 成功点亮绿灯，失败点亮红灯。

4.2.3 使用说明

- (1) 运行应用程序(Flashing the bin/IAR/CCS)
- (2) 如果文件操作成功,完成后,绿灯被点亮。在任何错误的情况下,红灯被点亮。

4.3 CC3200 Info Center Get Time Application

4.3.1 概述

获取时间的应用程序连接到网络自动对时服务,请求时间、处理数据并且在终端上显示时间,设备和服务之间的通信依赖网络时间协议(NTP)。

4.3.2 应用说明

时间应用程序连接到一个 SSID 是"cc3200demo"的 AP,如果没有成功连接到这个默认的 AP,用户可以根据终端的相应提示输入要连接的 AP,AP 必须按照下面指定的格式输入:

<ap_ssid>:<security_type>:<password>:<wep_key_id>:

- ap_ssid - ssid of the AP to be connected
- security_type - values 1(for Open) or 2(for WEP) or 3(for WPA)
- password - network password in case of 2(for WEP) or 3(for WPA)
- wep_key_id - key ID in case of 2(for WEP)

用户也可以在 SDK\example\common\common.h 文件中修改如下宏定义:

```
#define SSID_NAME "cc3200demo" /* AP SSID */
#define SECURITY_TYPE SL_SEC_TYPE_WPA/* Security type (OPEN or WEP or WPA*/
#define SECURITY_KEY "12345678" /* Password of the secured AP */
```

只要没有成功连接到 AP,红灯就不断的闪烁,连接成功红灯保持常亮状态。

用户可以通过改变变量 acSNTPserver,来选择下面服务列表中的任何一个。

#!/	##### list of SNTP servers #####		
#!/	##		
#!/	##	hostname	IP location
#!/	##	-----	
#!/	##	nist1-nj2.ustiming.org	165.193.126.229 Weehawken, NJ
#!/	##	nist1-pa.ustiming.org	206.246.122.250 Hatfield, PA
#!/	##	time-a.nist.gov	129.6.15.28 NIST, Gaithersburg, Maryland
#!/	##	time-b.nist.gov	129.6.15.29 NIST, Gaithersburg, Maryland
#!/	##	time-c.nist.gov	129.6.15.30 NIST, Gaithersburg, Maryland
#!/	##	ntp-nist.ldsbc.edu	198.60.73.8 LDSBC, Salt Lake City, Utah
#!/	##	nist1-macon.macon.ga.us	98.175.203.200 Macon, Georgia
#!/			
#!/	##	For more SNTP server link visit 'http://tf.nist.gov/tf-cgi/servers.cgi'	
#!/			
#####			
const char g_acSNTPserver[30] = "nist1-nj2.ustiming.org";			

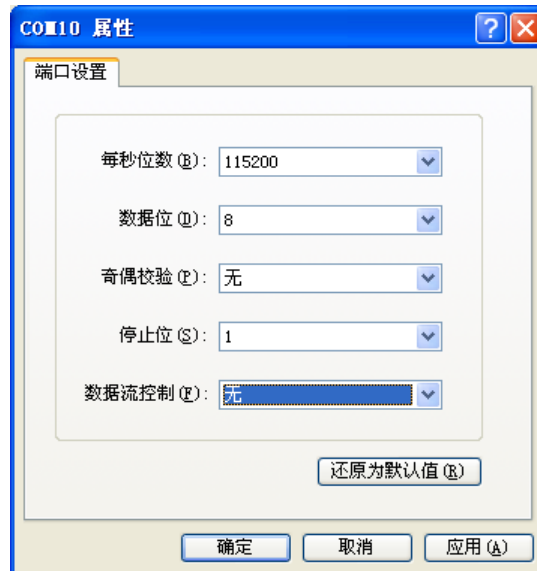
根据感兴趣的地区,通过改变宏定义来改变 GMT 偏移:

```
#define GMT_DIFF_TIME_HRS 5  
#define GMT_DIFF_TIME_MINS 30
```

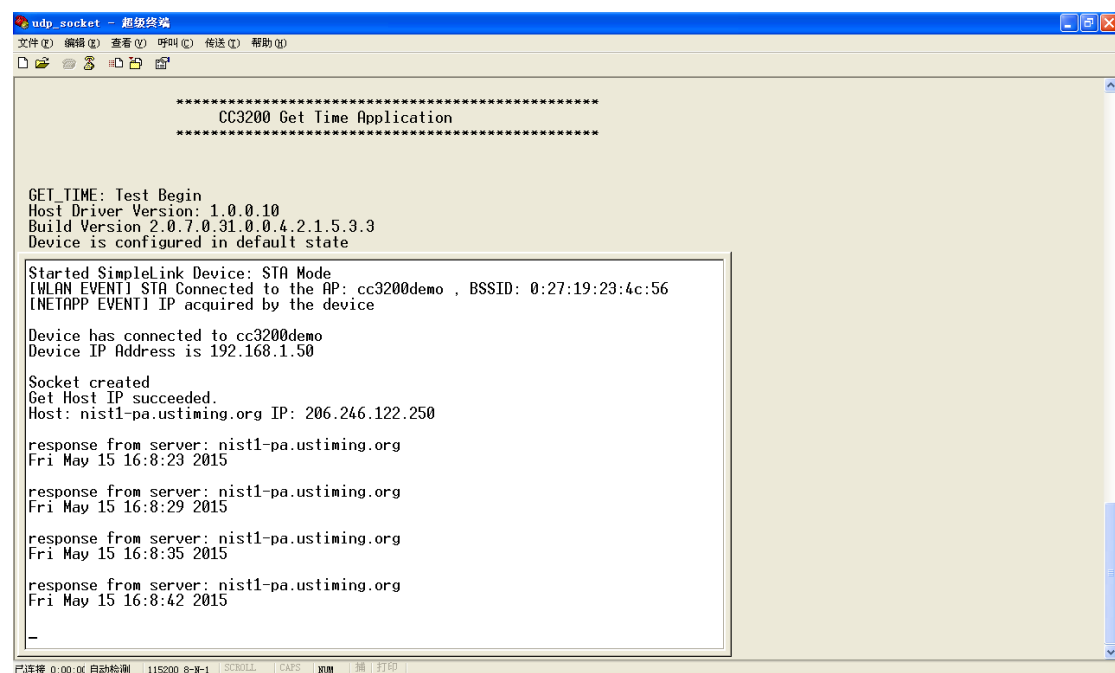
获取时间的应用程序得到了配置区域的当前时间。这个应用程序可以用来熟悉使用 UDP 套接字与服务器之间的通信。

4.3.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



- (2) 运行应用程序(Flashing the bin/IAR/CCS)
- (3) 应用程序要求 AP 连接网络
- (4) 如果没有成功连接到默认的 AP, 用户可以按照格式输入 AP
- (5) 只要没有成功连接到 AP, 红灯就不断的闪烁, 一旦连接成功红灯保持常亮状态。
- (6) 从 SNTP 服务获取的时间会定期的显示在终端上



4.4 CC3200 Info Center Get Weather Application

4.4.1 概述

获取天气的应用程序连接到“openweathermap.org”服务，请求指定城市的天气信息，处理数据并且在终端上显示天气情况。

4.4.2 应用说明

时间应用程序连接到一个 SSID 是“cc3200demo”的 AP, 如果没有成功连接到这个默认的 AP, 用户可以根据终端的相应提示输入要连接的 AP, AP 必须按照下面指定的格式输入:

<ap_ssid>:<security_type>:<password>:<wep_key_id>:

- ap_ssid - ssid of the AP to be connected
- security_type - values 1(for Open) or 2(for WEP) or 3(for WPA)
- password - network password in case of 2(for WEP) or 3(for WPA)
- wep_key_id - key ID in case of 2(for WEP)

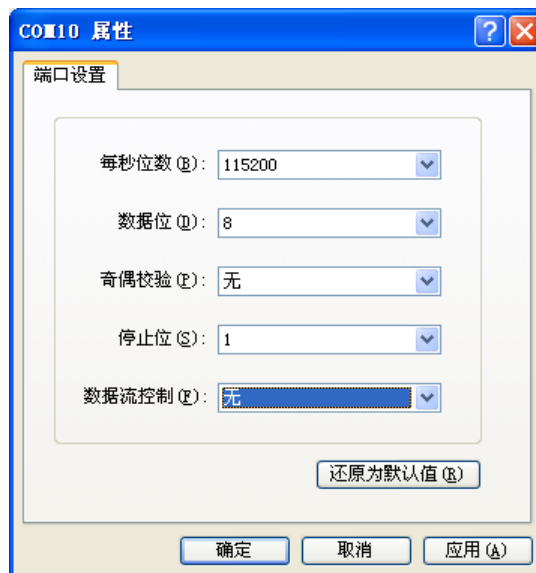
用户也可以在 SDK\example\common\common.h 文件中修改如下宏定义:

```
#define SSID_NAME "cc3200demo" /* AP SSID */  
#define SECURITY_TYPE SL_SEC_TYPE_WPA /* Security type (OPEN or WEP or WPA*/  
#define SECURITY_KEY "12345678" /* Password of the secured AP */
```

只要没有成功连接到 AP, 红灯就不断的闪烁, 一旦连接成功红灯保持常亮状态。提示用户输入城市名称获取天气信息, 从“openweathermap.org”服务得到指定城市的天气信息, 这是通过打开 TCP 套接字和服务, 发送一个 HTTP GET 请求来完成的, 数据处理并且显示天气信息

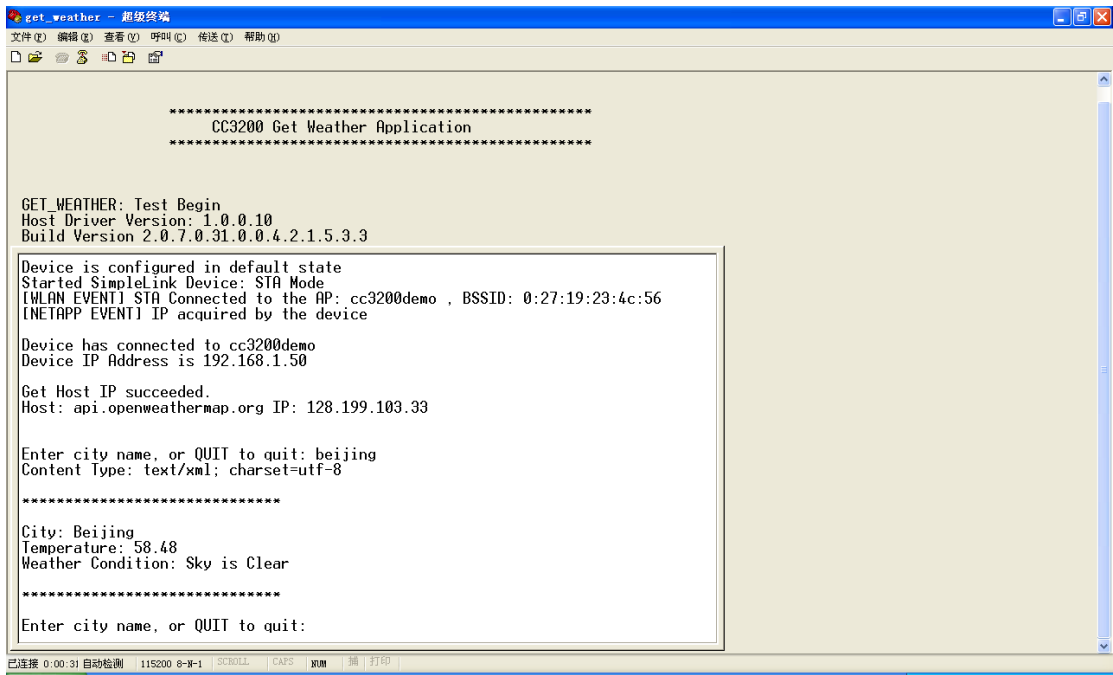
4.4.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下:



- (2) 运行应用程序(Flashing the bin/IAR/CCS)
- (3) 应用程序要求 AP 连接网络
- (4) 如果没有成功连接到默认的 AP, 用户可以按照格式输入 AP
- (5) 只要没有成功连接到 AP, 红灯就不断的闪烁, 一旦连接成功红灯保持常亮状态。
- (6) 提示用户输入城市名称
- (7) 显示获取到的指定城市的天气信息

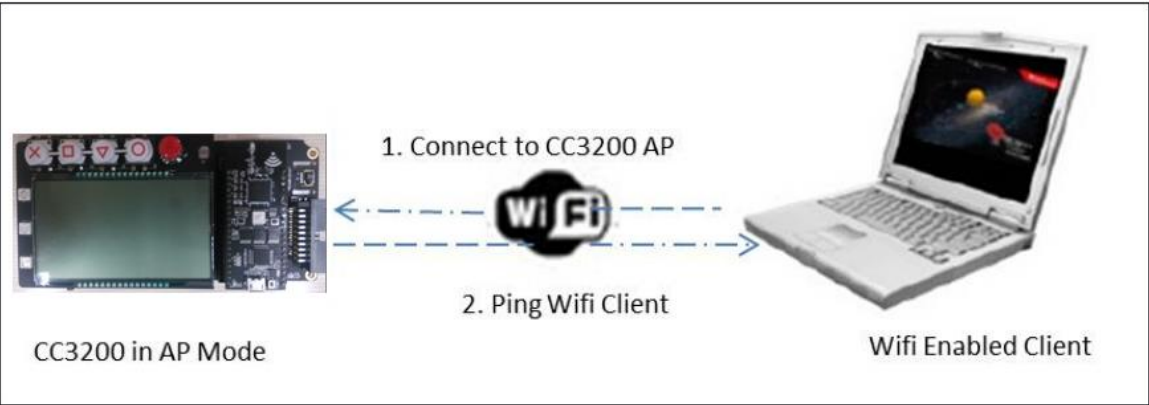
- (8) 出现提示后，用户也可以输入 QUIT 退出程序
- (9) 终端显示结果：



4.5 CC3200 Getting Started with WLAN AP

4.5.1 概述

SimpleLink CC3200 设备可以作为一个 AP(接入点), 任何在它范围内开启的 wifi 客户端都可以到连接设备，并且按照网络标准可以进行



4.5.2 应用说明

这个应用程序目的在于演示 CC3200 设备作为 AP(接入点)，开发者/用户写新的程序时，可以参考这个功能或重用它们。

设备作为 AP(接入点)，然后等待站点去连接，如果连接成功，就会 Ping 这个站点，0 是预期的返回值，如果返回其他信息表示 Ping 这个站点没有成功。

这个例子可以被用在 TI-RTOS 或者 FreeRTOS。ti_rtos 项目和 ti_rtos_config 项目需要被导入到应用程序空间。这两个项目在 CC3200-SDK 目录中的 ti_rtos 文件夹下。需要安装 TI-SDK-WF\Tools\tirtos_2_01_00_03\tirtos_simplelink_setupwin32_2_01_00_03.exe。

使用 CC3200 作为 AP 是简单的两个步骤。

- (1) 通过调用 sl_Start() API 来启动 SimpleLink。
 - (2) 等待设备获得 ip 地址。
- 设备作为 AP(接入点)模式后，接下来的两个步骤是确保设备作为 AP(接入点)。
- (1) 等待站点连接到设备(你需要连接一台机器到设备上)。
 - (2) 然后 Ping 这个机器(站点)。

更多的细节请参考 main.c 文件

注意：如果设备没有 Ping 通，试着去关闭你机器上的杀毒软件。

用户可以在 CC3200-SDK\example\common\common.h 文件中修改如下宏定义：

```
#define SSID_NAME      "cc3200demo"    /* AP SSID */
#define SECURITY_TYPE  SL_SEC_TYPE_WPA /* Security type (OPEN or WEP or WPA*/
#define SECURITY_KEY   "12345678"      /* Password of the secured AP */
```

4.5.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 运行应用程序(getting_started_with_wlan_ap) (Flashing the bin/IAR/CCS).
- (3) 如果不是 AP 模式，应用程序将切换到 AP 模式。
- (4) 客户端连接到设备，设备(AP)就会 Ping 客户端，并且通过 UART 打印结果
- (5) 终端显示结果：

```
COM10:115200baud - Tera Term V1
文件(F) 编辑(E) 设置(S) 控制(C) 窗口(W) 帮助(H)

*****
CC3200 WLAN AP Application
*****

Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.4.1.1.5.3.3
Device is configured in default state
Device started as STATION
Enter the AP SSID name: wlan_ap
Device is configured in AP mode
Connect a client to Device
[NETAPP EVENT] IP Leased to Client: IP=192.168.1.2 , Client is connected to Device
Running Ping Test...
Ping Test successful
WLAN AP example executed successfully
```

4.6 CC3200 Getting Started with WLAN Station

4.6.1 概述

CC3200 设备可以作为一个典型的网络系统站点(基本配置)，它能连接到接入点(安全或不安全)，并且可以通过这个接入点(可用的)连接网络。



4.6.2 应用说明

这个应用程序目的在于演示在简单网络当中 CC3200 设备作为一个站点，开发者/用户写新的程序时，可以参考这个功能或重用它们。设备连接到 AP(接入点)，AP 的配置在应用程序中以宏的形式存储。如果连接成功就会去获取 “www.ti.com” 的 IP 地址，然后 Ping 这个地址，0 是预期的返回值，不同的代码被返回表示网络连接是不可用的或 Ping 这个连接是不成功的。

用户可以在 CC3200-SDK\example\common\common.h 文件中修改如下宏定义：

#define SSID_NAME	"cc3200demo"
#define SECURITY_TYPE	SL_SEC_TYPE_OPEN
#define SECURITY_KEY	""
#define WEP_KEY_ID	

这个例子可以被用在 TI-RTOS 或者 FreeRTOS。ti_rtos 项目和 ti_rtos_config 项目需要被导入到应用程序空间。这两个项目在 CC3200-SDK 目录中的 ti_rtos 文件夹下。需要安

装 TI-SDK-WF\Tools\tirtos_2_01_00_03\tirtos_simplelink_setupwin32_2_01_00_03.exe。

CC3200 作为站点需要三个步骤：

- (1) 通过调用 `sl_Start()` API 来启动 SimpleLink。
- (2) 通过调用 `sl_WlanConnect()` API 去连接 AP(接入点)。
- (3) 用 `sl_NetCfgGet()` API 或者检查 `SL_NETAPP_IPV4_IPACQUIRED_EVENT` NetApp 事件去得到设备 IP 地址

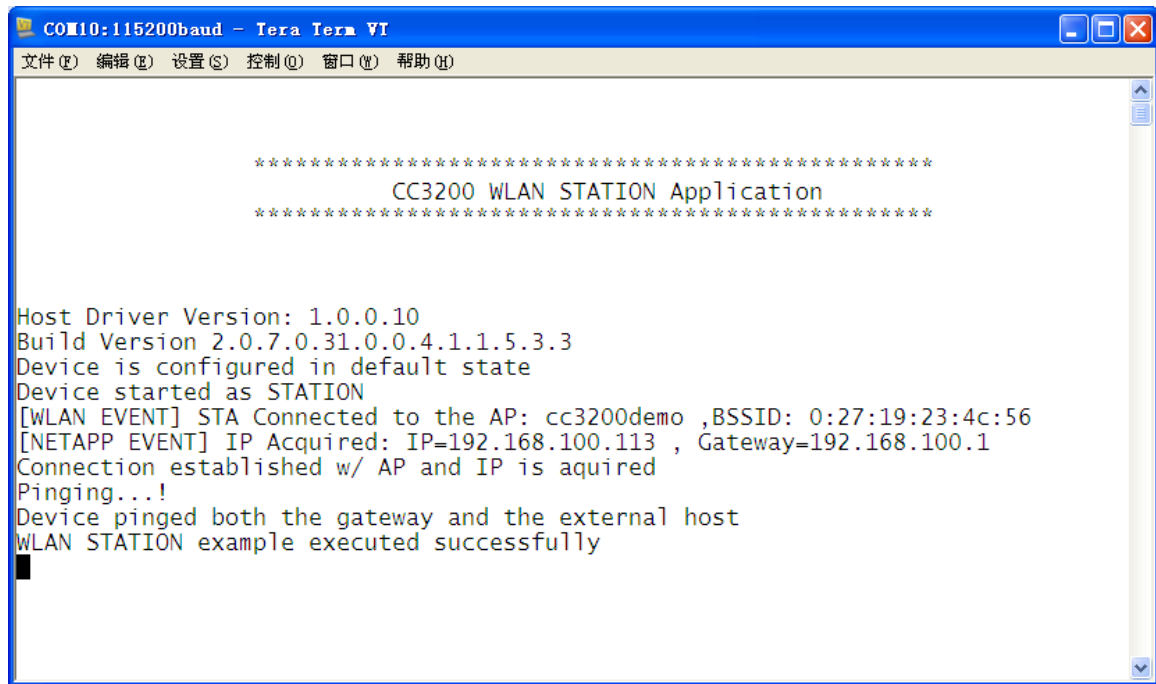
更多的细节请参考 `main.c` 文件

4.6.3 使用说明

- (1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 运行应用程序(`getting_started_with_wlan_ap`) (Flashing the bin/IAR/CCS).
- (3) 如果是其他模式，应用程序将切换到站点模式。
- (4) 连接 AP, 如果连接成功红灯被点亮。
- (5) 连接到设备，然后 Ping AP(接入点)，成功绿灯被点亮。
- (6) 设备会进一步的去检查网络连接 (Ping “`www.ti.com`”), 成功蓝灯被点亮。
- (7) 观察 LED 灯的状态和代码输出以确保正确执行。



```
COM10:115200baud - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(C) 窗口(W) 帮助(H)

*****
CC3200 WLAN STATION Application
*****

Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.4.1.1.5.3.3
Device is configured in default state
Device started as STATION
[WLAN EVENT] STA Connected to the AP: cc3200demo ,BSSID: 0:27:19:23:4c:56
[NETAPP EVENT] IP Acquired: IP=192.168.100.113 , Gateway=192.168.100.1
Connection established w/ AP and IP is aquired
Pinging...!
Device pinged both the gateway and the external host
WLAN STATION example executed successfully
```

4.7 CC32xx HTTP Server

4.7.1 概述

HTTP 服务应用程序演示了 HTTP 客户端（浏览器）和 SimpleLink 设备之间的交互。SimpleLink 设备运行 HTTP 服务并且用户可以通过浏览器和设备进行交互。用户能够交互使用简单的 HTTP GET 和 POST 命令。

使用 HTTP 服务器, 用户可以做以下:

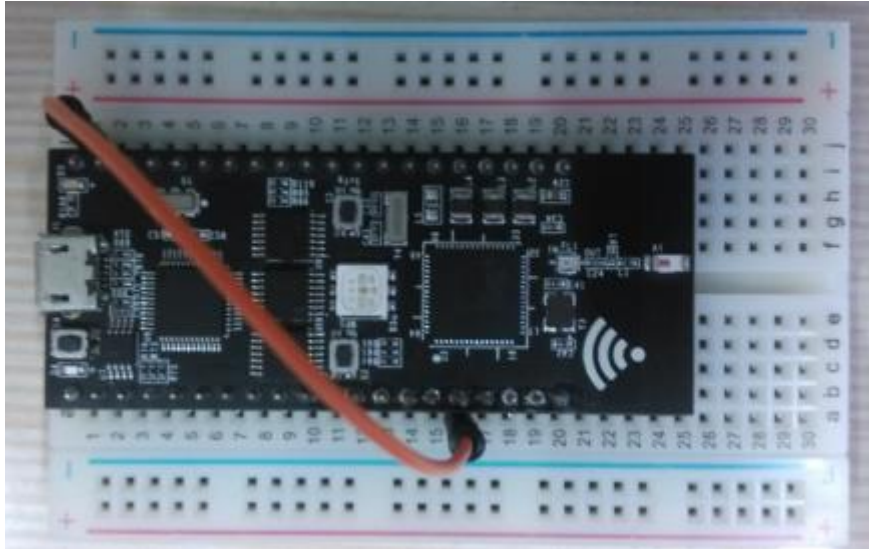
- (1) 静态页面存储在 ROM / SFLash
- (2) 通过在静态 HTML 页面填写表单来更新设备设置
- (3) 在 HTML 页面上, 使用符号来获取动态数据例如设备的状态
- (4) 在 HTML 页面上, 使用符号发送命令给设备, 例如控制设备上的 LED 灯。

4.7.2 使用说明

- (1) 按照下面的步骤加载示例内容 (html, image, Application binary) 。
 - 用 uniflash 打开 <cc3200-sdk>\examples\httpserver\html\httpserver.ucf 文件
 - 下载这个文件到设备中, 详细说明在 Uniflash User Guide。

注意: 下载时, SW1 拨码开关 3 拨到字母方向。

- (2) 运行应用程序(IAR/CCS).
 - (3) 从 flash 中运行二进制文件
 - 将 SW1 拨码开关 3 拨到数字方向然后按下复位键
 - 设备启动然后检查 AP 跳线, 如下所示, 如果跳线存在, 以 AP 模式启动。
- GP3 接入 3.3V 为 AP 模式 不接时为 STA 模式



Case 1 - AP 模式

(1) CC3200 设备以 AP 模式启动，在终端上打印 SSID 和 HTTP 服务器域名

```
*****
CC3200 HTTP Server Application
*****

Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.3.0.1.1.8.8
Device is configured in default state

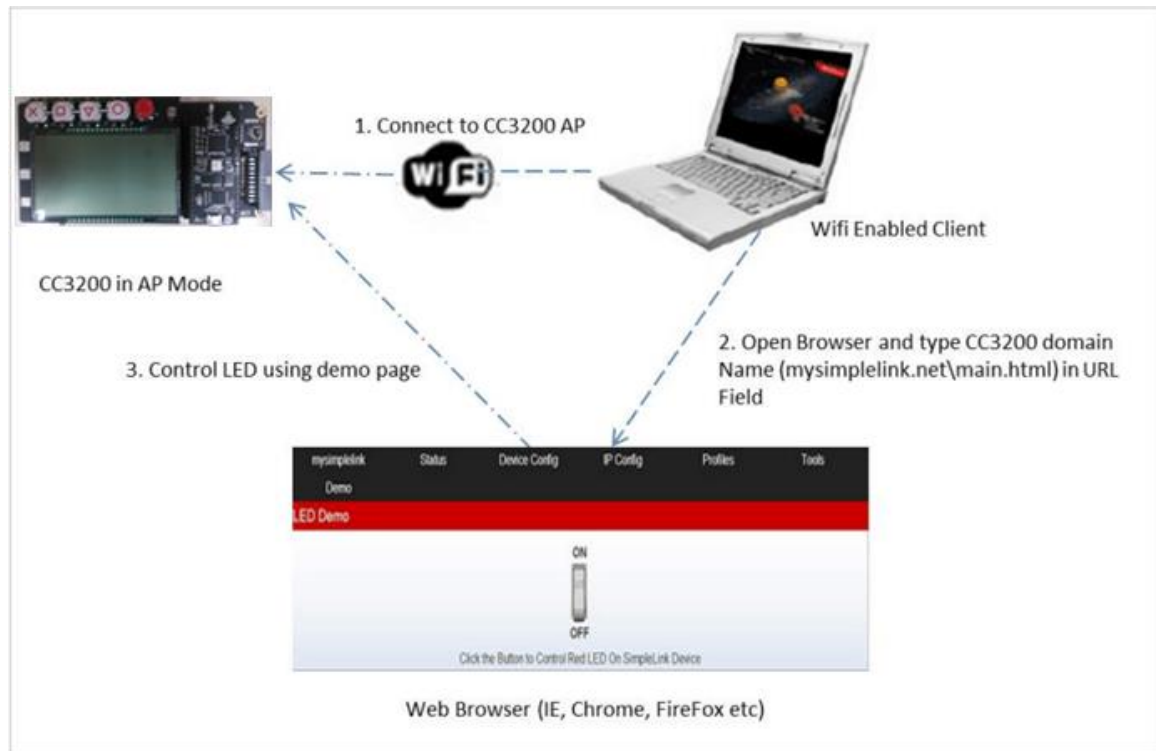
Device is in AP Mode, Please Connect to AP [mysimplelink-10D355] and type [mysimplelink.net] in the browser
-
```

(2) 用户开启 PC/MAC/智能手机 wifi, 连接设备 AP (默认 SSID: mysimplelink-MACAddr)

(3) 用户打开浏览器, 在浏览器地址栏输入 mysimplelink.net \ main.html

(4) 浏览器显示主页面。用户选择演示选项卡如下所示

(5) 浏览器显示演示 web 页面如下所示



(6) 用户点击开/关按钮控制设备上的红灯

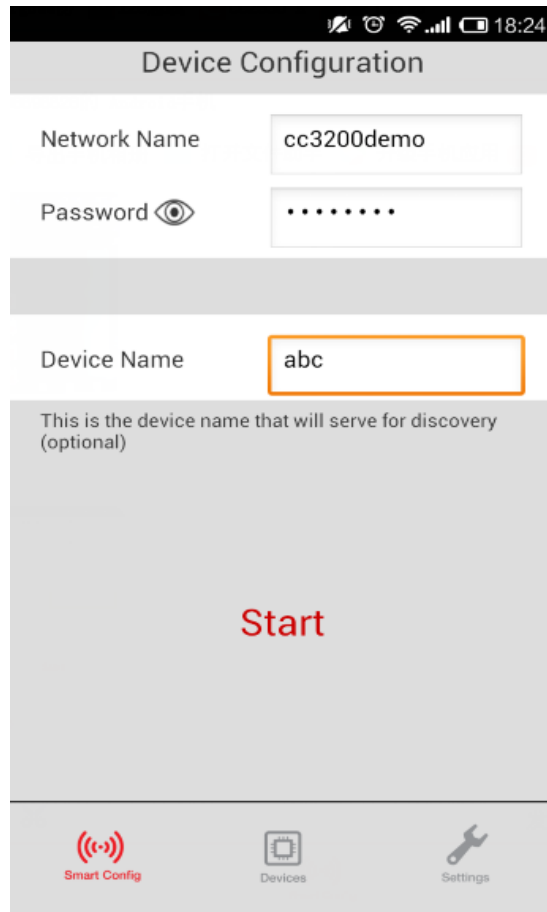
Case 2 - STA 模式

(1) 在你的手机上安装 TI SmartConfig 应用程序

(2) CC3200 设备连接到 AP(如果已经使用 SmartConfig 应用程序配置过)

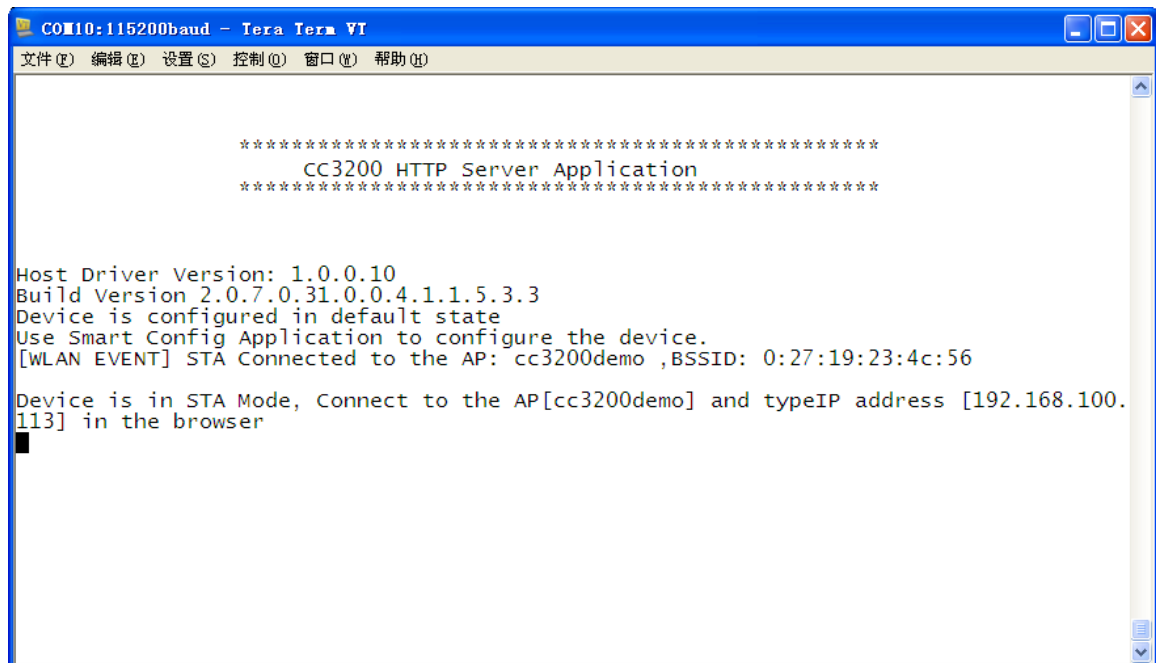
(3) 第一次连接 AP 是没有配置过的，用户需要使用 SmartConfig 应用程序连接

- 3.1 连接手机到选定的 AP
- 3.2 打开 SmartConfig 应用程序
- 3.3 网络的名称会自动出现在网络名称字段。输入密码。开放网络保留‘密码’字段空白。
- 3.4 输入设备名称。用于识别你的设备。
- 3.5 按下” Start” 按钮。



➤ 一旦 smartconfig 完成后,弹出窗口显示表明,新设备连接到网络。

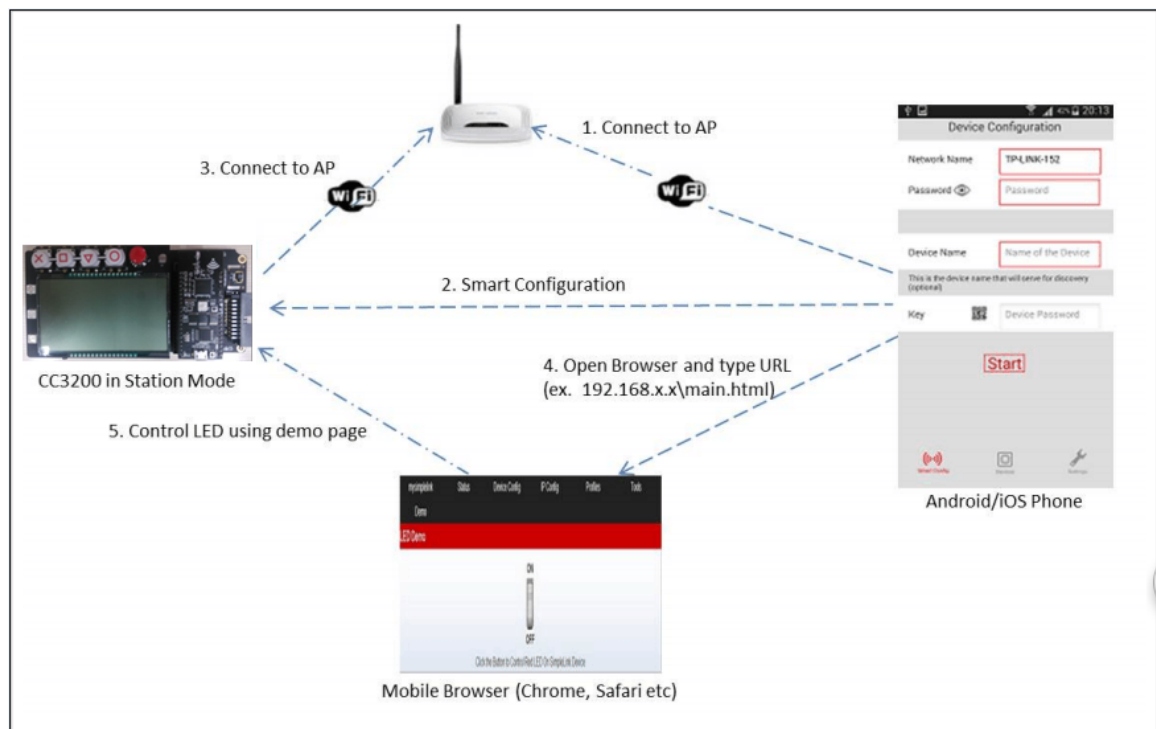
(4) 一旦成功, CC3200 设备连接到 AP 通过 UART 终端显示设备的 IP 地址



(5) 用户打开浏览器和在浏览器的地址栏中输入 < ipaddr > \led_demo.html, 例如
192.168.100.113\main.html

(6) 浏览器显示主页面。用户选择演示选项卡如下所示

(7) 浏览器显示演示 web 页面



4.8 CC3200 mDNS

4.8.1 概述

mDNS (多播域名系统) 是一个主机名称解析服务。Mdns 主机可以多播消息包含关于服务、ip 地址、文本信息在 Mdns 端口上。Mdns 客户端在同一子网监听这些消息。CC3200 设备可以使用这些服务。

4.8.2 应用说明

这个应用程序既可以被用作广播 (Mdns 主机) 应用程序也可以用作监听 (Mdns 客户机) 应用程序。

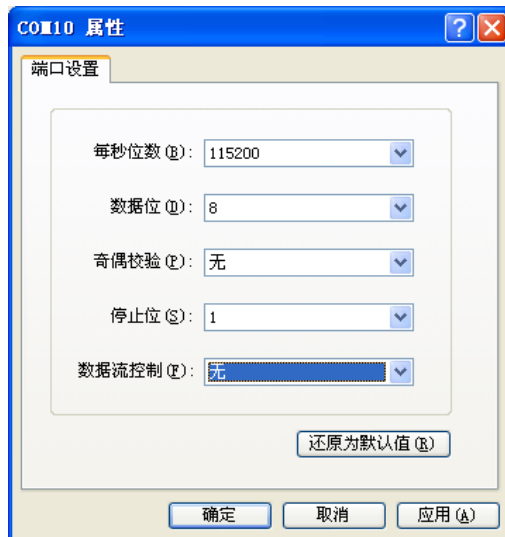
一个预定义的符号“MDNS_ADVERTISE”可以在两种之间做切换。

每个案例的第一步都是连接到一个 AP. 可以通过修改“SSID_NAME”宏定义 (common.h) 修改 AP 名称。广播应用程序注册和多播一个服务 (CC3200._uart._tcp.local) 在 Mdns 端口。监听应用程序获得另一个设备 (mdn 主机) 发出多播服务的 IP、端口和文本。

如果一个服务 w/ 相同的名称已经在 n / w 入册过, 新的服务注册将会失败

4.9.3 使用说明

(1) 设置一个串行通信程序 (超级终端/ TeraTerm)。终端设置如下:

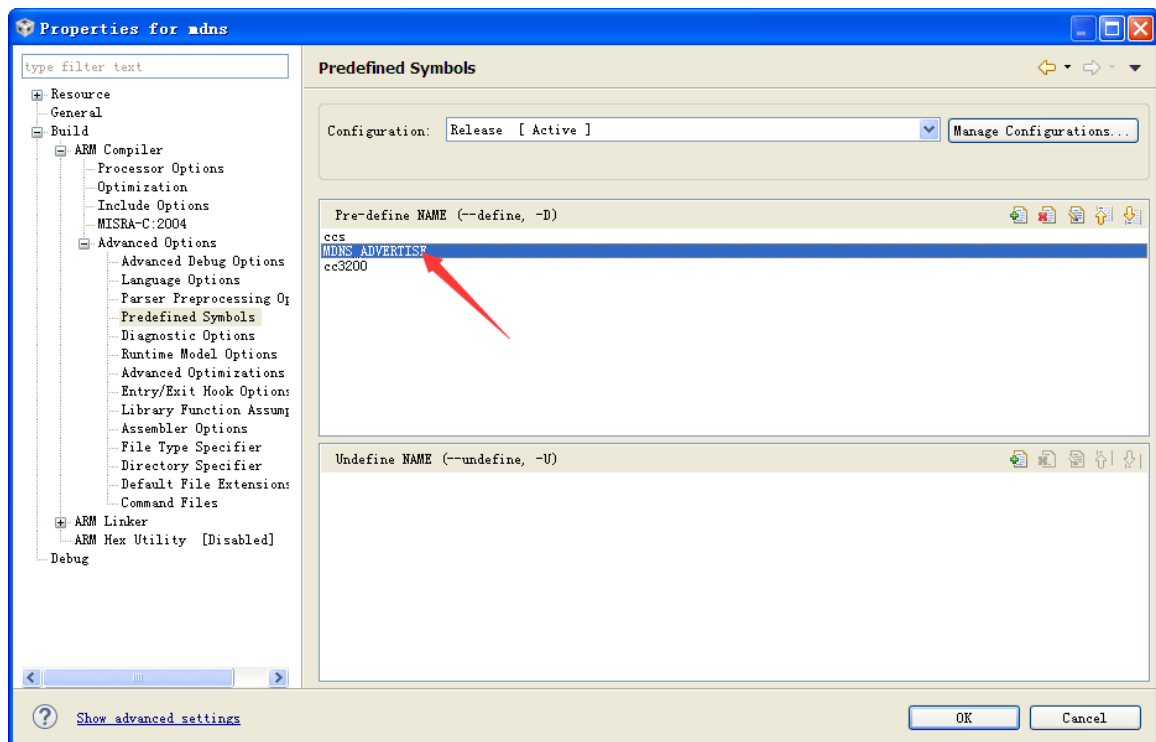


(2) 运行程序作为广播应用程序或者监听应用程序

通过修改是否使用图中所指预定义

使能“MDNS_ADVERTISE”预定义，作为广播应用程序

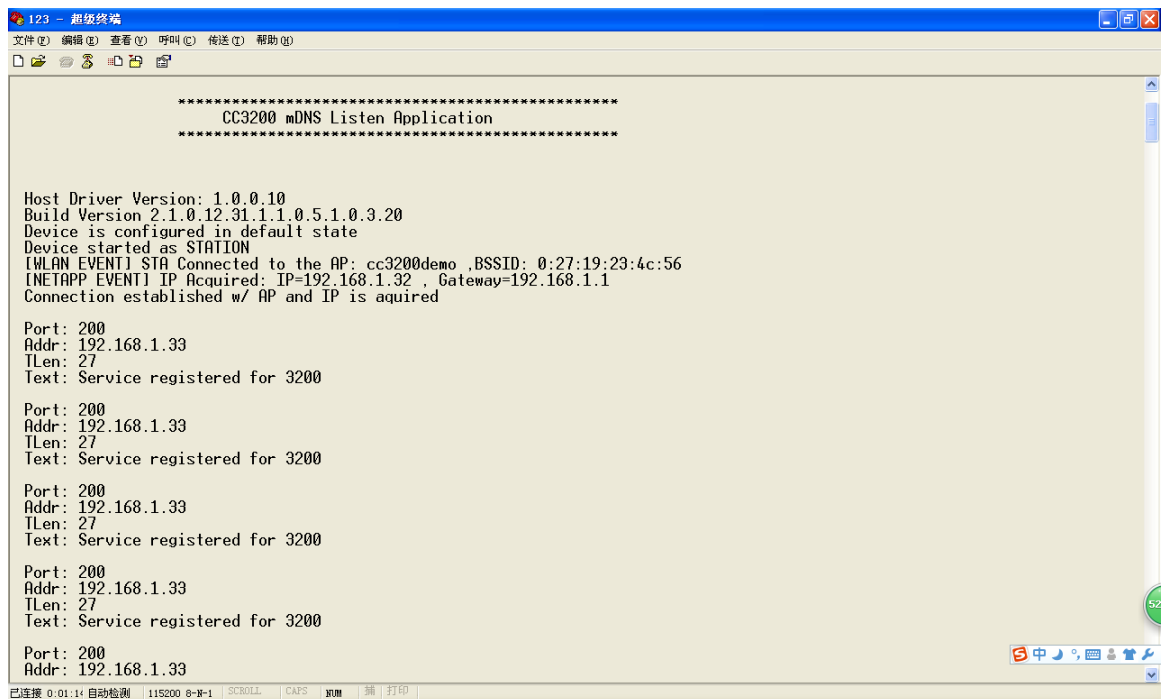
关闭“MDNS_ADVERTISE”预定义，作为监听应用程序



(3) 如果作为广播应用程序运行，终端上出现：



- (4) 如果作为听应用程序运行, 需要运行 mdns 主机在同一子网(可以是一个 PC、移动手机或另一个 CC3200 设备运行作为一个广播)。在接收多播消息:



4.9 CC3200 Mode-Configuration Application

4.9.1 概述

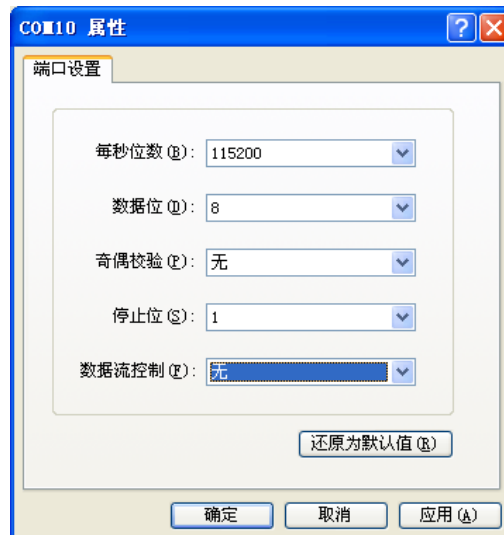
设备可以作为一个站点或 AP（接入点）或 P2P 模式。也可以通过修改适当的文件来调整网络参数

4.9.2 应用说明

应用程序的目的是为了让用户在网络模式（站点或者 AP 或者 P2P）之间进行切换和在 AP 模式下配置 SSID。

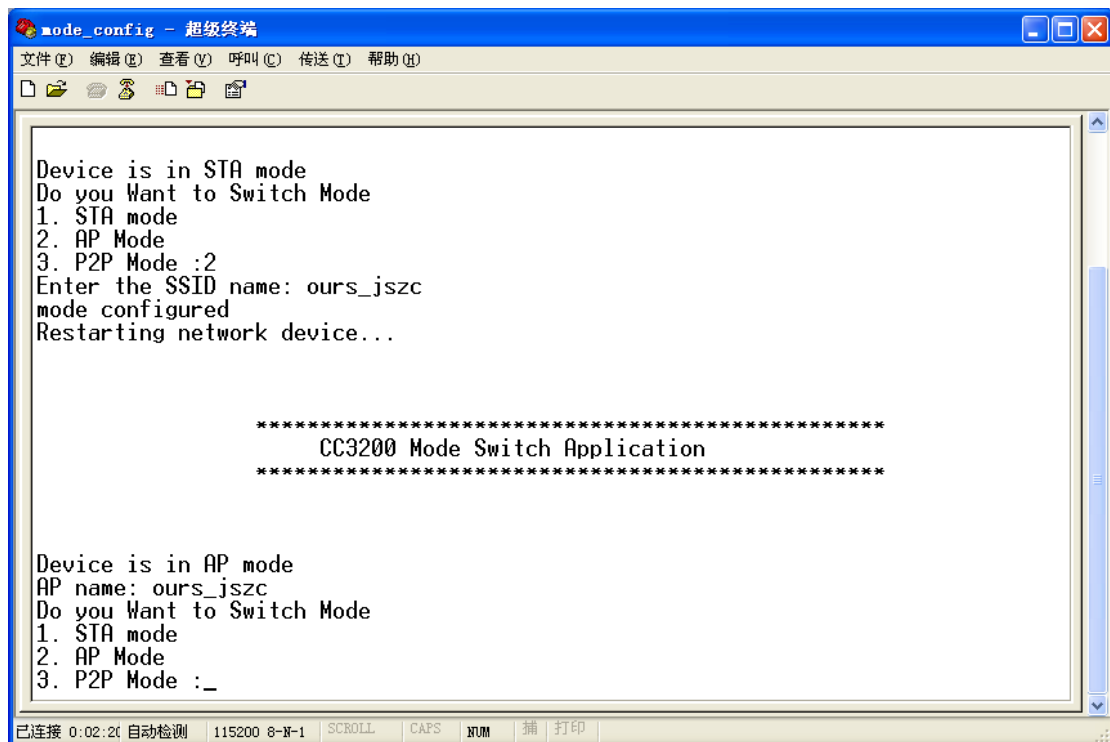
4.9.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(2) 运行应用程序(Flashing the bin/IAR/CCS)

(3) 在终端上提示输入去配置网络模式和 SSID 名称（AP 模式下）



4.10 CC32xx NWP Filter Application

4.10.1 概述

Rx-Filters 特性允许用户简单的定义和管理 Rx-filtering 过程。它减少了传输到主机的网络流量, 并且实现了高效的电源管理。

4.10.2 使用说明

- (1) 修改 IP 地址和 MAC 地址去匹配正在运行 iperf 的机器。
- (2) 修改 SSID_NAME 去连接想要连接的网络
- (3) 运行应用程序
- (4) 一旦设备连接到指定 IP 的网络，我们就可以在变量 pucCC31xx_Rx_Buffer 中看到 IP 地址
- (5) 在想要发送数据的机器上运行 iperf。
- (6) 在 IP 地址和 MAC 地址的基础上应用程序会创建并使能过滤器，并且过滤数据包。
- (7) 下面的截图是被过滤的 IP 地址和 MAC 地址的机器发送数据包的显示结果

```
D:\NewIperf\iperf>iperf.exe -c 192.168.1.101
connect failed: Connection timed out.
```

- (8) 下面的截图是没有被过滤的 IP 地址和 MAC 地址的机器发送数据包的显示结果

```
D:\NewIperf\iperf>iperf.exe -c 192.168.1.101
-----
Client connecting to 192.168.1.101, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[156] local 192.168.1.100 port 49331 connected with 192.168.1.100 port 49331
[ ID] Interval      Transfer    Bandwidth
[156] 0.0-67.9 sec  32.0 KBytes  3.86 Kbits/sec
```

4.11 P2P (Wi-Fi Direct)

4.11.1 概述

这个样例展示如何连接一个 P2P 设备，应用程序启动 TCP 服务，端口是 5001，然后等待 P2P 设备的连接并发送数据。

CC3200 支持的不同 P2P 角色：

SL_P2P_ROLE_GROUP_OWNER: CC3200 被配置成 “Group-Owner” 模式。

SL_P2P_ROLE_CLIENT: CC3200 被配置成 “Client” 模式。

SL_P2P_ROLE_NEGOTIATE: CC3200 与作为客户端的远程设备通信。

CC3200 可以被配置成以下模式进行通信：

SL_P2P_NEG_INITIATOR_ACTIVE: CC3200 执行发现，一旦发现远程设备，就立即发送通信请求。

SL_P2P_NEG_INITIATOR_PASSIVE: CC3200 将执行发现，一旦发现远程设备，CC3200 等待它开始通信。

SL_P2P_NEG_INITIATOR_RAND_BACKOFF: CC3200 将执行发现，一旦发现远程设备，它触发一个随机计时器(1 - 7 秒)，并等待远程设备通信。计时器到期，CC3200 自己开始通信。

支持安全类型中使用 p2p 通信：

使用 P2P 通信支持的安全类型有：

SL_SEC_TYPE_P2P_PBC

SL_SEC_TYPE_P2P_PIN_DISPLAY

SL_SEC_TYPE_P2P_PIN_KEYPAD

CC3200 可以被配置成任何 P2P 模式，CC3200 将执行发现并且使用' SL_SEC_TYPE_P2P_PBC' 安全类型连接第一个被发现的设备

4.11.2 应用说明

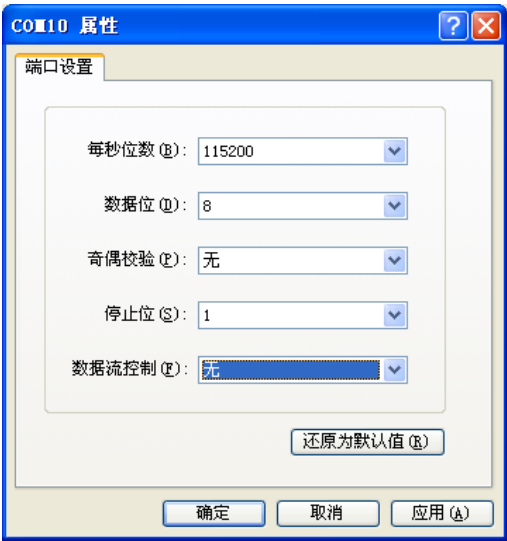
这个样例目的在于展示如何配置和使用 P2P 模式。应用程序将按照以下设置去配置设备
P2P role negotiate (SL_P2P_ROLE_NEGOTIATE)

```
#define P2P_ROLE_TYPE_NEGOTIAE
```

P2P 通信积极
P2P 设备监听通道 11 和 P2P 设备的操作通道设置为 6
P2P 连接的安全类型' PBC'

4.11.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 编译并运行程序
- (3) 启动 P2P 设备

在远程设备上将会发现 CC3200 的 P2P_DEVICE_NAME（程序中设置的）。



(4) 连接 CC3200。一旦连接成功，在终端上会显示 CC3200 的 IP 地址。

(5) 在远程 P2P 设备上打开 'Ipef' 客户端然后连接到 'PORT_NUM' 端口

```
lprf.exe -c <DEST_IP_ADDR> -p <PORT_NUM> -i 1
```

```
COM10:115200baud - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(C) 窗口(W) 帮助(H)

*****
CC3200 P2P Demo Application
*****

Scan Wi-Fi direct device in your handheld device
Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.4.1.1.5.3.3
Device is configured in default state
Started SimpleLink Device: P2P Mode
Connect to cc3200-p2p-device
CC3200 Device IP : IP=192.168.49.184
Send TCP packets from your handheld device to CC3200's IP
Received TCP packets successfully
Test passed, exiting application...
```

4.12 Provisioning AP

4.12.1 概述

这个应用程序展示了用户可以在浏览器上给 CC3200 设备配置 AP 信息。CC3200 扫描接入点，并把它们存储在内存中，然后创建一个带有 AP 列表的 HTML 页面，用户可以打开 HTML 页面，选择想要配置的 AP。

4.12.2 使用说明

- (1) 按照下面的步骤加载示例内容 (html, image, Application binary) 。
 - 用 uniflash 打开 <cc3200-sdk>\examples\httpserver\html\httpserver.ucf 文件
 - 下载这个文件到设备中，详细说明在 Uniflash User Guide。

注意：下载时，SW1 拨码开关 3 拨到字母方向。

- (2) 运行应用程序 (IAR/CCS)。
- (3) 从 flash 中运行二进制文件
 - 将 SW1 拨码开关 3 拨到数字方向然后按下复位键
 - 设备以 STA 模式启动，扫描有效的接入点，然后以 AP 模式重新启动（默认的 SSID: mysimplelink-<MacAddr>）
- (4) 用户使能 PC/MAC/智能手机的 wifi，并且连接上面的 AP (mysimplelink-<MacAddr>)，
- (5) 用户打开浏览器，在地址栏中输入 “mysimplelink.net\profiles_config.html”。
- (6) 页面上会显示 AP 列表
- (7) 通过单击单选按钮选择 AP
- (8) 在相应的区域，用户还需要提供安全类型，密码和优先级
- (9) 单击 Add 按钮

mysimplelink Status Device Config IP Config Profiles Tools

WiFi Connectivity Profiles Settings

Add Profile

SSID: ☐ Cisco_C6 ☐ CC3200-PLT ☐ TP-LINK-APK ☒ oob ☐ NETGEAR-24G

Security Type: ☐ Open ☐ WEP ☒ WPA

Security Key: Hexadecimal digits - any combination of 0-9, a-f and A-F

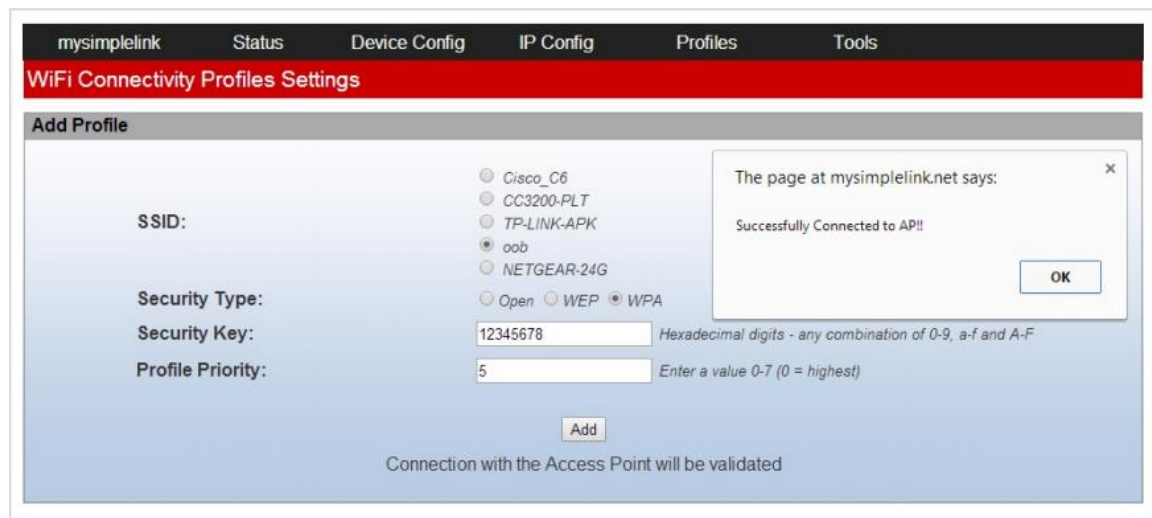
Profile Priority: Enter a value 0-7 (0 = highest)

Add

Connection with the Access Point will be validated

The page at mysimplelink.net says: Checking Status.. OK

- (10) 然后设备切换到站点模式，连接选择的 AP
- (11) 连接成功或失败后，会有相应的提示信息



4.13 Provisioning with SmartConfig

4.13.1 概述和应用介绍

什么是 SmartConfig?

首先 CC3200 开启 WIFI 功能, 配置 CC3200 连接到 WIFI 网络, 这需要 AP(接入点) 信息, 当 WEP/WPA/WPA2 启用时, 需要 SSID 名称和密码。考虑到嵌入式无线网络应用程序通常缺少用户接口, 例如键盘或触摸屏, 如果没有输入输出, 那么这个过程就会是复杂的。

为了提供一个良好的用户体验, TI 创造了 SmartConfig 技术——CC3200 设备连接到无线网络是一步并且一次性的过程, 这与其他供应商需要多个步骤配置一个设备到网路是非常不同的。

4.13.2 应用说明

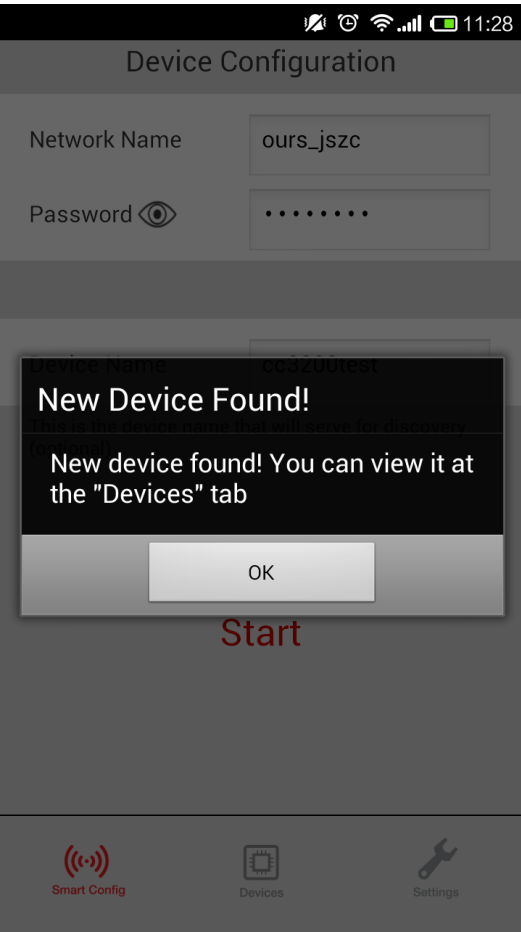
应用程序执行以下步骤:

- (1) 初始化设备网络层。
- (2) 删除所有配置文件, 以确保 CC32xx 设备没有连接到任何 AP
- (3) 设置自动连接政策。确保一旦智能配置完成, 设备自动连接到 AP。
- (4) 等待来自 TI SmartConfig 应用程序的配置
- (5) 等待连接 AP

4.13.3 使用说明

- (1) 选择给 CC3200 设备配置的 AP
- (2) 在你的手机上安装 TI SmartConfig 应用程序
- (3) 运行应用程序(Flashing the bin/IAR/CCS)
- (4) 在 SmartConfig 模式配置设备
应用程序配置设备在 SmartConfig 模式。设备等待来自 TI SmartConfig 应用程序的配置
- (5) 连接手机到选择的 AP
- (6) 打开 TI SmartConfig 应用程序
- (7) 网络的名称会自动出现在网络名称字段。输入密码。开放网络保留‘密码’字段空白。
- (8) 输入设备名称。用于识别你的设备。
- (9) 按下” Start” 按钮。

- (10)一旦 smartconfig 完成后,弹出窗口显示表明,新设备连接到网络。
- (11)同时, 如果连接成功 CC3200 目标板上的红色 LED 灯将被点亮。



4.14 CC32xx Provisioning WPS Application

4.14.1 概述

样例演示了如何使用 WPS wi - fi 和 CC31xx 配置。通过 WPS 演示了两个可用的选项：

- “Push button”
- “PIN code”

4.14.2 应用说明

应用程序执行下面的步骤：

初始化设备网络层

AP SSID 是“cc3200demo”，SSID 可以通过改变下面的宏定义来修改

```
#define SSID_NAME "cc3200demo"
```

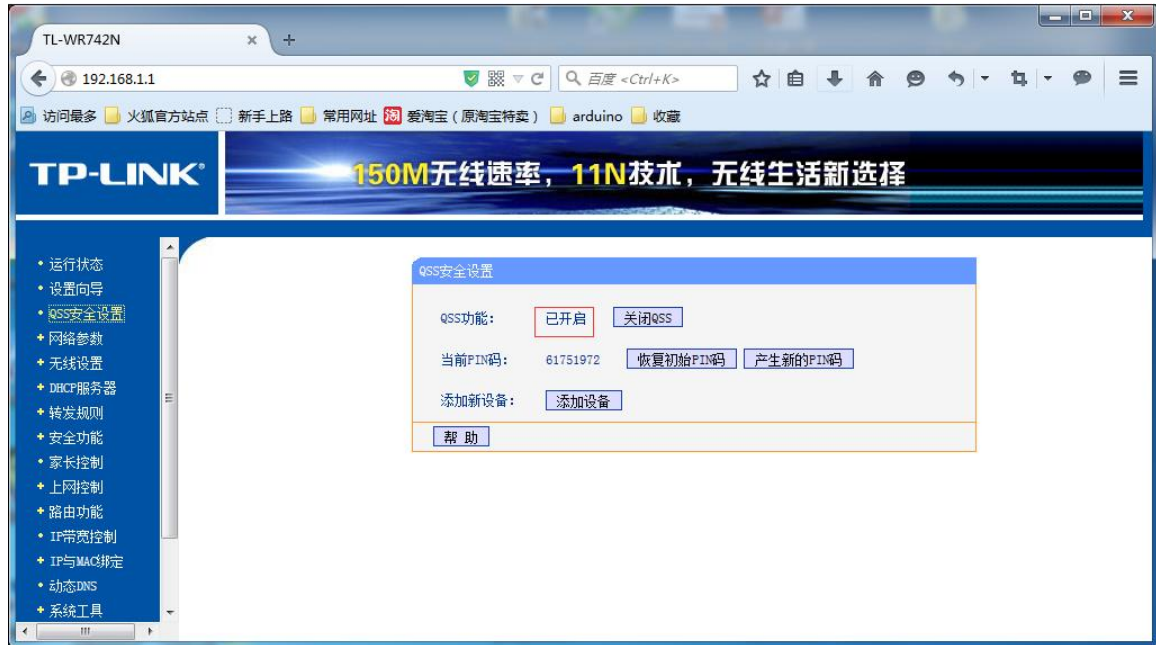
使用按钮的方法，安全类型为 WPS 连接到 AP

- 一旦连接成功，红色灯被点亮
- 一段时间后，从 AP 断开连接，红色灯熄灭。
- 使用 PinCode 的方法，安全类型为 WPS 连接到 AP. 默认的 PinCode 是 88664422
- 一旦连接成功，红色灯被点亮
- 一段时间后，从 AP 断开连接，红色灯熄灭。

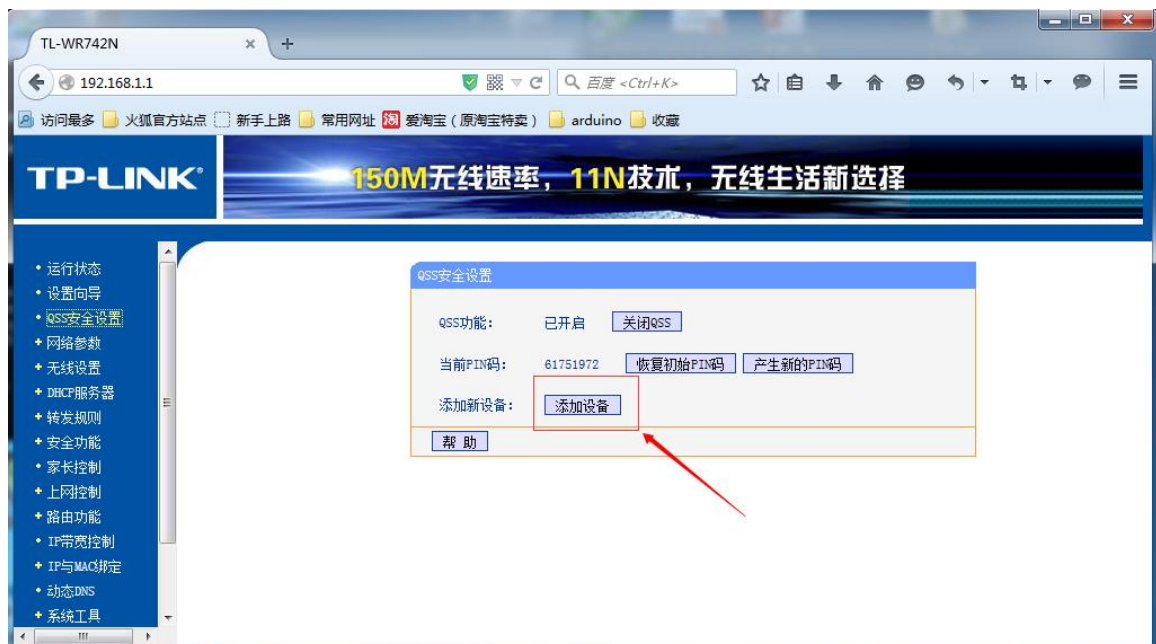
4.14.3 使用说明

- (1) 选择一个有 WPS（按钮和 PinCode）支持 WIFI 配置的 AP
- (2) 用笔记本连接到 AP, 并且打开 AP 网页（以 `http://192.168.1.1` 为例, 然后然后输入管理密码）

首先开启路由器的 QSS 功能



- (3) 运行应用程序(Flashing the bin/IAR/CCS).
- (4) 运行应用程序, 在 WPS（按钮）模式下, 设备等待连接
- (5) 在路由上按下 WPS 按钮。
- (6) 一旦连接成功, 红色灯被点亮, 一段时间后断开连接, 红色灯熄灭。
- (7) 在 WPS（PinCode）模式下, 设备等待连接
- (8) 在 AP 页面中输入 PIN 码 88664422 建立 WPS 连接。





(9) 一旦连接成功，红色灯被点亮，一段时间后断开连接，红色灯熄灭。

4.15 Scan Policy

4.15.1 概述

CC3200 器件允许设置扫描策略并使用设置的策略进行扫描。可以使用 `sl_WlanPolicySet` API 设置其扫描策略为 `SL_POLICY_CONNECTION`、`SL_POLICY_SCAN` 和 `SL_POLICY_PM`。一旦按照设置的策略启动一个周期性的扫描，可以使用 `sl_WlanGetNetworkList` API 在扫描返回的结果中检索。可以得到 RSSI (接收信号强度)、安全信息、BSSID 和 SSID 等信息。

4.15.2 应用说明

以下阐述如何在 CC3200 策器件中设置扫描策略。该过程包含以下步骤：

- (1)、确保当前没有设置连接策略（没有在后台运行扫描）
- (2)、设置扫描策略并设置扫描周期为 10s，启用扫描
- (3)、获取扫描结果中单次事务的所有 20 个条目
- (4)、获取扫描结果中 5 个事务的 4 次交互
- (5)、关闭扫描

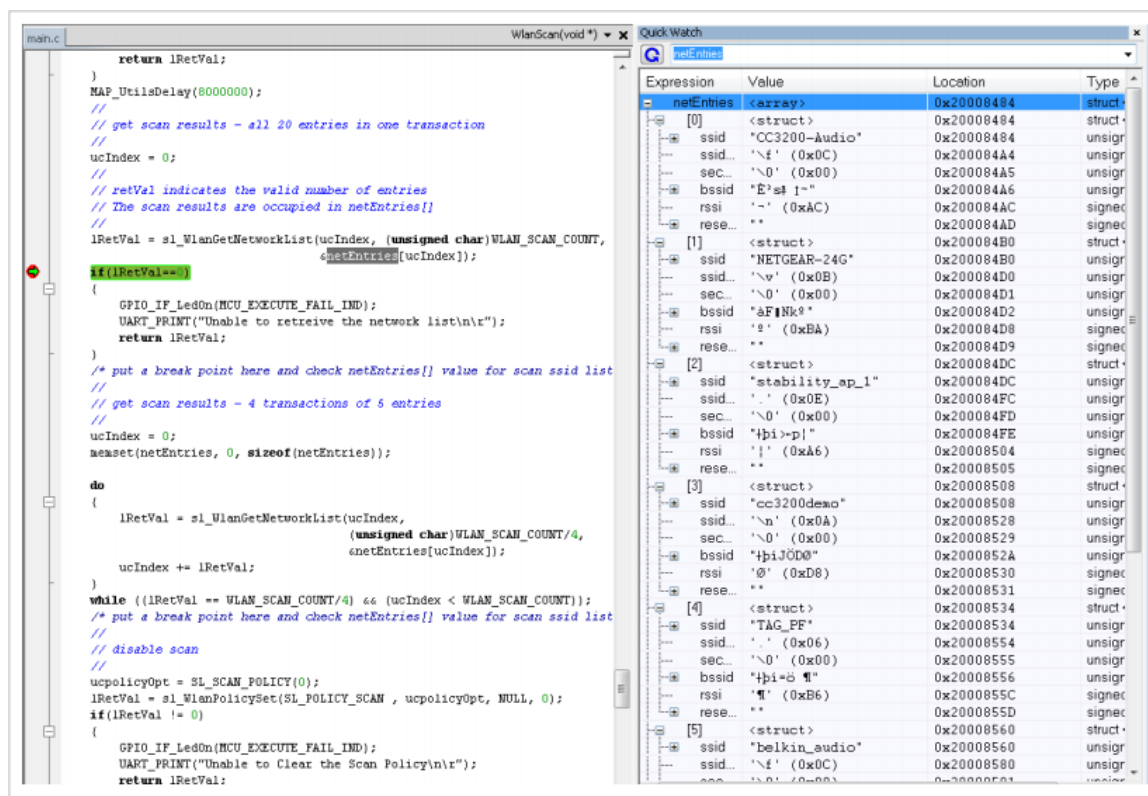
4.15.3 使用说明

- (1)、运行参考程序

在 IAR 或者 CCS 开发环境中打开工程，编译并且下载程序到目标板。

- (2)、应用程序启动后将会扫描并且获取网络列表

下面的截图显示获取到网络列表中的网络数量；`lRetVal` 表示发现的网络数量和网络信息中包含的网络列表。



4.16 TCP Socket

4.16.1 概述

CC3200 可以在不运行 RTOS 操作系统的情况下使用标准的 TCP 和 UDP 协议进行网络通信。

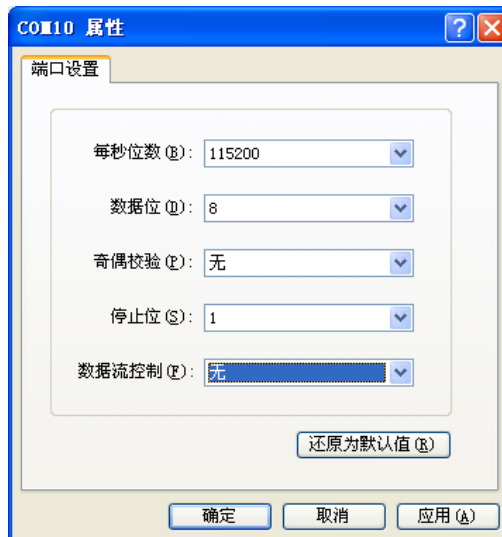
4.16.2 应用说明

本应用说明了如何将 CC3200 设备作为 TCP Client 或 TCP server 进行 TCP 通信。用户可以参考这些函数去重新编写新的应用程序。设备可以连接到一个应用程序宏定义中所指定的 SSID 的 AP。该应用程序实现一个 TCP client 并且发送 1000 个 TCP 数据包到宏定义中指定的套接字地址、端口号和 IP 地址。程序执行成功将返回 0。若返回非 0 值意味着发生套接字错误。缺省的设置以下宏定义中，可以在源代码中更改设置或者在程序运行时更改设置

```
#define SSID_NAME "cc3200demo"
#define IP_ADDR 0xc0a80167
#define PORT_NUM 5001
#define TCP_PACKET_COUNT 1000'
```

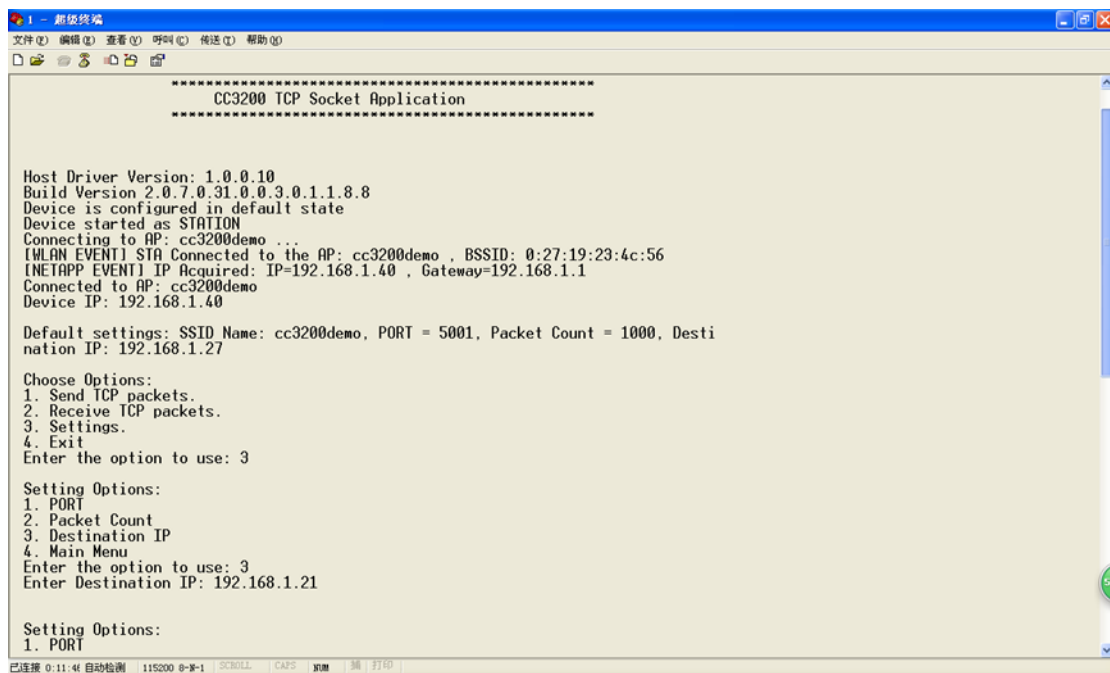
4.16.3 使用说明

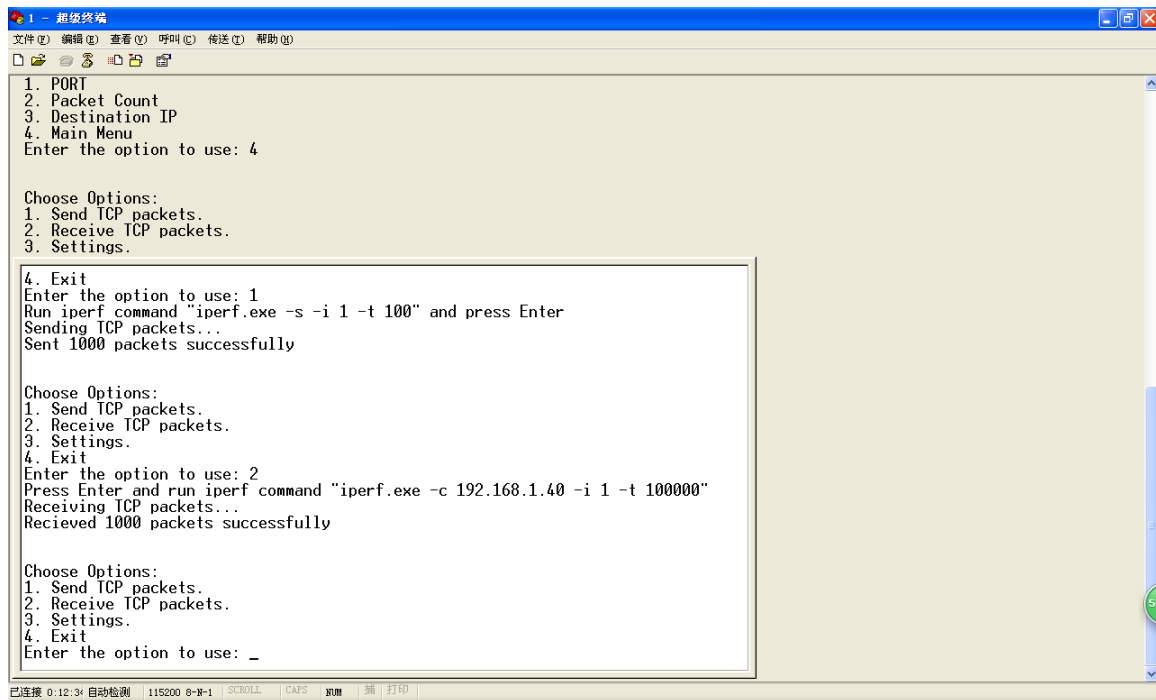
(1) 设置串口通信程序 (HyperTerminal/TeraTerm)，以下详细说明超级终端的设置，提示：在运行 iperf 时请关闭防毒软件



(2) 在 IAR 或 CCS 中打开 tcp_socket 工程，编译并下载应用程序到目标板。

(3) 将 PC 连接到 CC3200 设备所连接的同一个 AP 上。查看 PC 的 IP 地址，并将源码宏定义中的 IP_ADDR 改写为 PC 的 IP 地址，或在 CC3200 程序运行时按照截图的方法设置。





4.17 Transceiver Mode

4.17.1 概述

TX 功能是 SimpleLink 收发模式的一部分。在这种模式下，SimpleLink 装置能够直接在无线物理层进行通信，即绕过网络堆栈、WiFi 驱动程序和 MAC 层。在这种模式下，用户被给予充分的灵活性去建立发送的数据包。

RX 统计特性被用来确定一些关于介质和 CC3200 RX 机制的重要参数。RX 统计特性提供的数据是关于以下几方面：

- RSSI—接收的功率以 dbm 为单位
- RSSI 直方图—在-40—87dbm 之间，分辨率 8dbm
- RSSI 平均值—DATA + CTRL / MANAGEMENT
- 接收帧—分为有效帧、FCS 错误，PLCP 错误帧
- 速率柱状图—创建一个包含 BGN 协议速率从 1mbps-MCS7 的柱状图

4.17.2 应用说明

这个例子说明了在无线网络 PHY 层之上建立私有协议，并给予用户充分的灵活性，以建立自己的包。特别需要注意的是：前面两个字节的原始数据是 WiFi PHY 层的特定数据，它的说明如下：

- 1st byte:无线网络速率。它是定义在 wlan.h 文件中的 RateIndex_e 结构体中
- 2nd byte:4 bits 的功率信息和 4 bits 同步码。

该 RX 统计功能用于检查媒介的拥堵，距离情况并帮助验证射频硬件，在理想情况下使用 RSSI 信息，定位 CC3200 设备。

Note: 更多关于 CC3200 的说明请查阅 CC3x00 收发器模式章节

4.17.3 使用说明

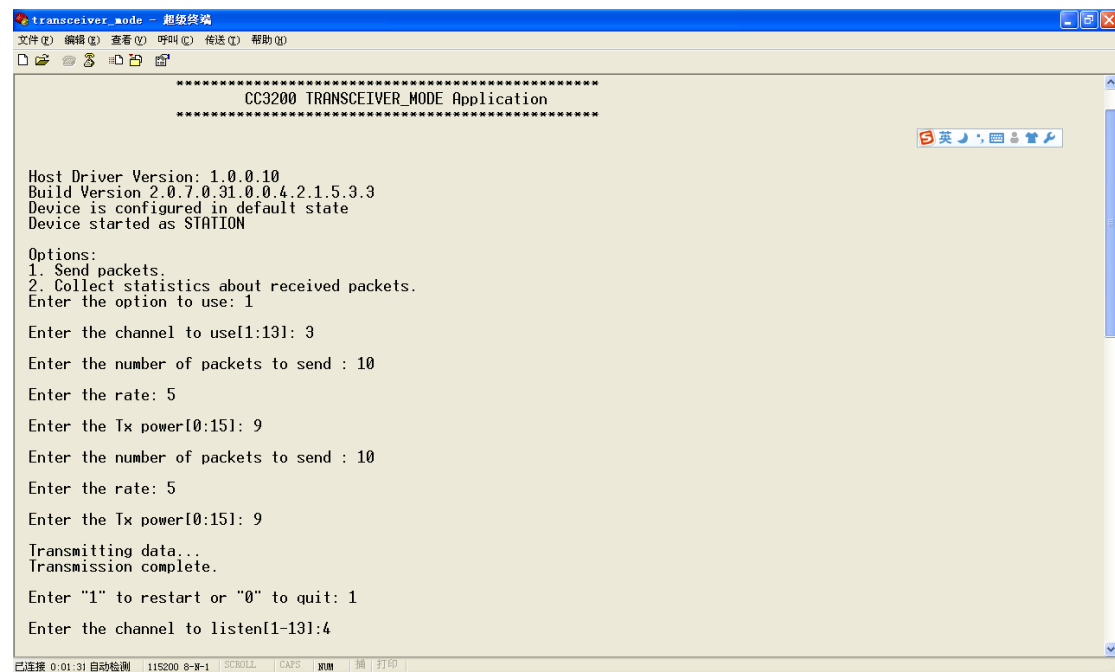
(1) 设置串口通信程序（HyperTerminal/TeraTerm），下面介绍详细设置，在主机 PC 上设置如下：



(2) 运行该应用程序（使用 IAR/CCS 在线调试运行或者将 bin 文件烧写至目标板中）

(3) 观察主机通过串口显示的状态信息，从而去理解应用程序执行的操作顺序

设备运行时超级终端的截图如下：



4.18 UDP Socket

4.18.1 概述

CC3200 设备可以在不使用 RTOS 操作系统的情况下，使用标准 TCP、UDP 网络通信协议进行通信。

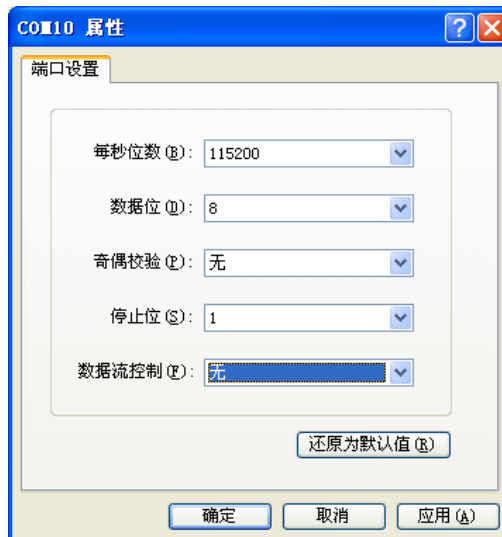
4.18.2 应用说明

这个特定的应用程序说明了如何将 CC3200 设备用作 UDP 通信的客户端或服务端。用户可以使用这些函数或者使用它们编写全新的应用程序。通过该应用程序 CC3200 设备将连接到应用程序宏定义中指定的 SSID 对应的 AP 上。该应用程序实现一个 UDP Client 并且发送 1000 个 UDP 数据包到宏定义中指定的套接字地址、端口、IP 地址。程序正常执行成功将会返回 0，所有非 0 的返回值意味着已经发生套接字错误。缺省的定义在以下宏中指定，可以在源代码或是程序运行时改变缺省设置。

#define SSID_NAME	"cc3200demo"
#define IP_ADDR	0xc0a80167
#define PORT_NUM	5001
#define UDP_PACKET_COUNT	1000

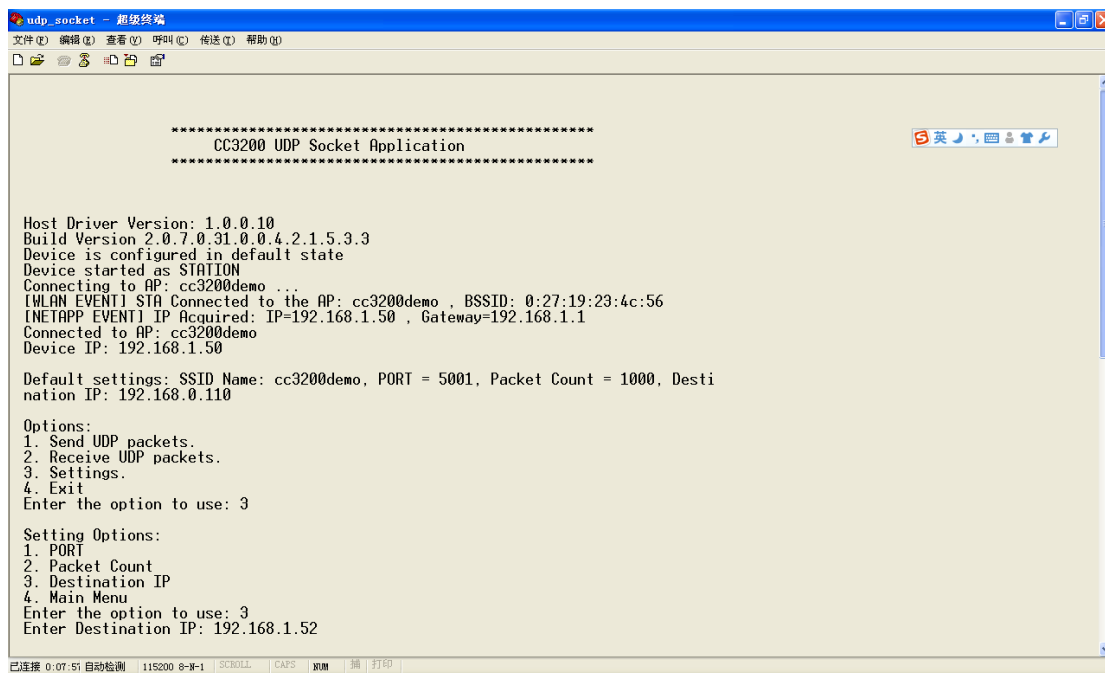
4.18.3 使用说明

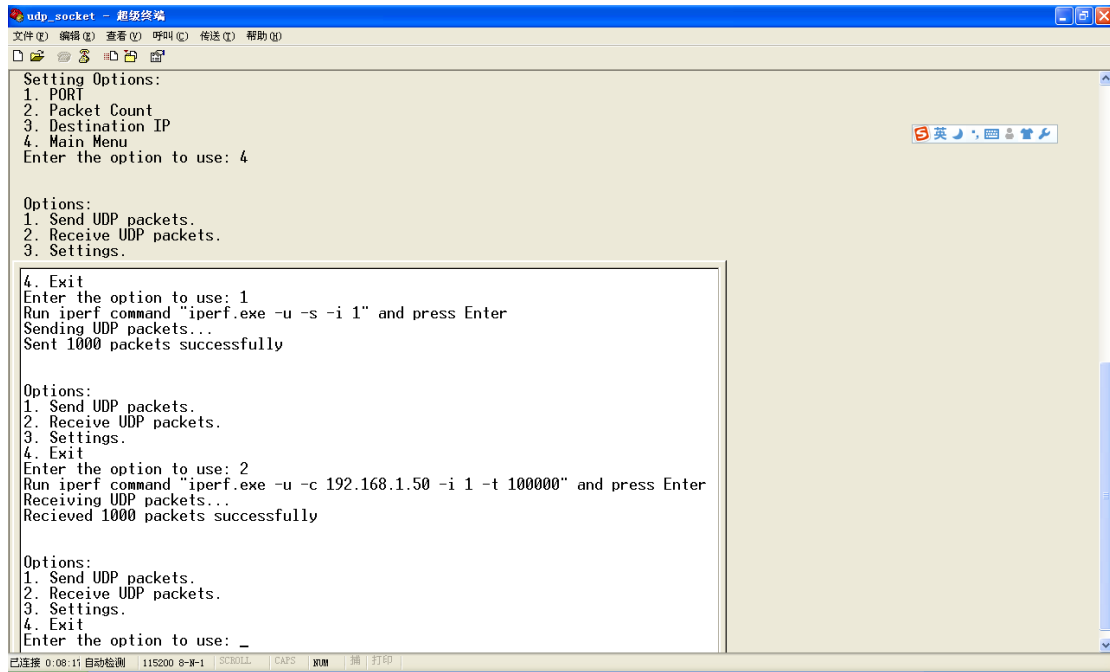
(1) 设置串口通信应用程序（HyperTerminal/TeraTerm），详细的设置如下图。提示：运行 iperf. 时请关闭防毒软件。PC 主机端设置：端口-枚举 COM 口（CC3200LP 双端口）



(2) 运行 UDP_SOCKET 应用程序（IAR/CCS 在线调试或将 bin 文件下载到 flash 运行）。

(3) 将 PC 连接到 CC3200 设备所连接的同一个 AP 上。查看 PC 的 IP 地址，并将源码宏定义中的 IP_ADDR 改写为 PC 的 IP 地址，或在 CC3200 程序运行时按照截图的方法设置。





4.19 Idle Profile

4.19.1 概述

闲置配置文件允许 CC3200 用户测量 CC3200 的当前值, 功耗等参数, 当设备基本上是空闲的(数值天气预报和应用子系统在低功率深度睡眠条件)。应用程序的另一个目的是介绍用户可以很容易配置电源管理框架。

4.19.2 应用说明

电源管理框架

CC3200 设备有多个电源模式, 用户可以根据他们应用程序的要求使用电源模块。电源管理框架使电源管理的应用程序变得容易, 用户可以配置设备进入最低功率模式, 电源管理框架隐藏了底层错综复杂的设置, 帮助用户使用做简单的方式去体验设备的低功率模式。如果用户希望自定义不同电源模式电源, 或者给不同的设备使用框架, 可以通过改变两个文件。关于电源管理框架的更多信息, 请参考 CC32xx Power Management Framework.pdf。

测量当前值

这个应用程序指定低功率深睡眠模式为最低功率模式。有时, 数值天气预报和应用子系统都是在最低功率模式, 在这种情况下, 当前值可能低至几百微妙, 可以在 CC3200 Low Power Modes Current Measurement 页看到当前电流的测量

程序流程

用户需要修改大多数宏定义参数

SSID_NAME - 接入点的名称

GPIO_SRC_WKUP - 作为唤醒资源的 GPIO 号

APP_UDP_PORT - 等待 udp 数据包的端口号

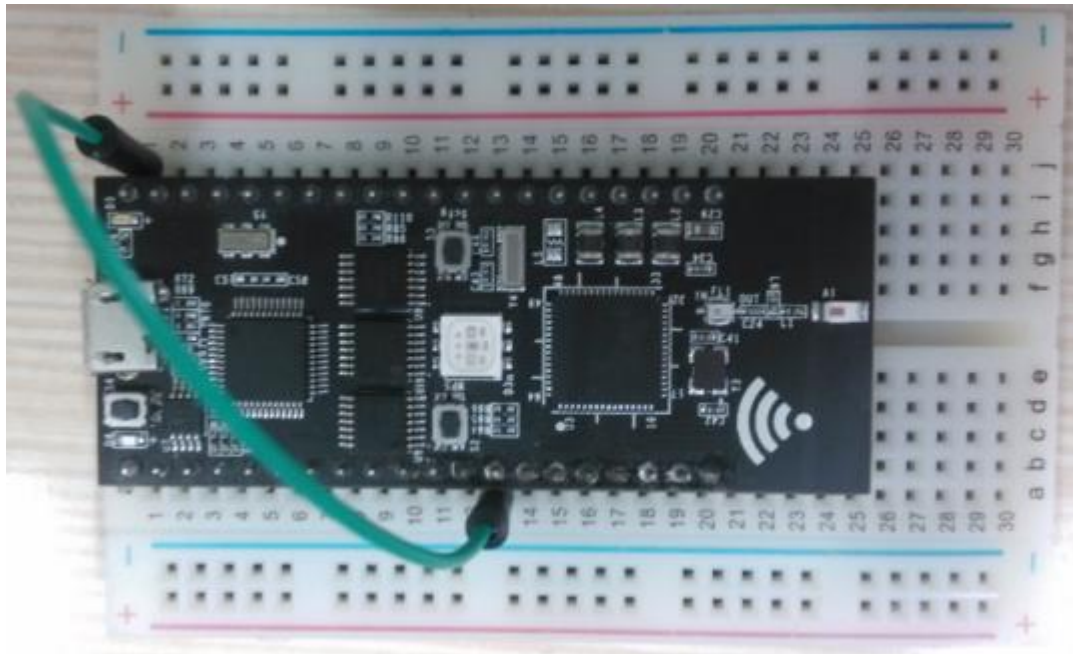
LPDS_DUR_SEC - 从最低功率模式跳出来的的时间

LPDS_DUR_NSEC - 从最低功率模式跳出来的的纳秒时间（除了 HIB_DUR_SEC）

#define SSID_NAME	"cc3200demo"
#define GPIO_13	13
#define GPIO_SRC_WKUP	GPIO_13
#define APP_UDP_PORT	5001
#define LPDS_DUR_SEC	60
#define LPDS_DUR_NSEC	0

4.19.3 使用说明

(1) 连接 VCC 和 GPIO_13 (PIN_4)，作为 GPIO 唤醒资源，如下图所示：

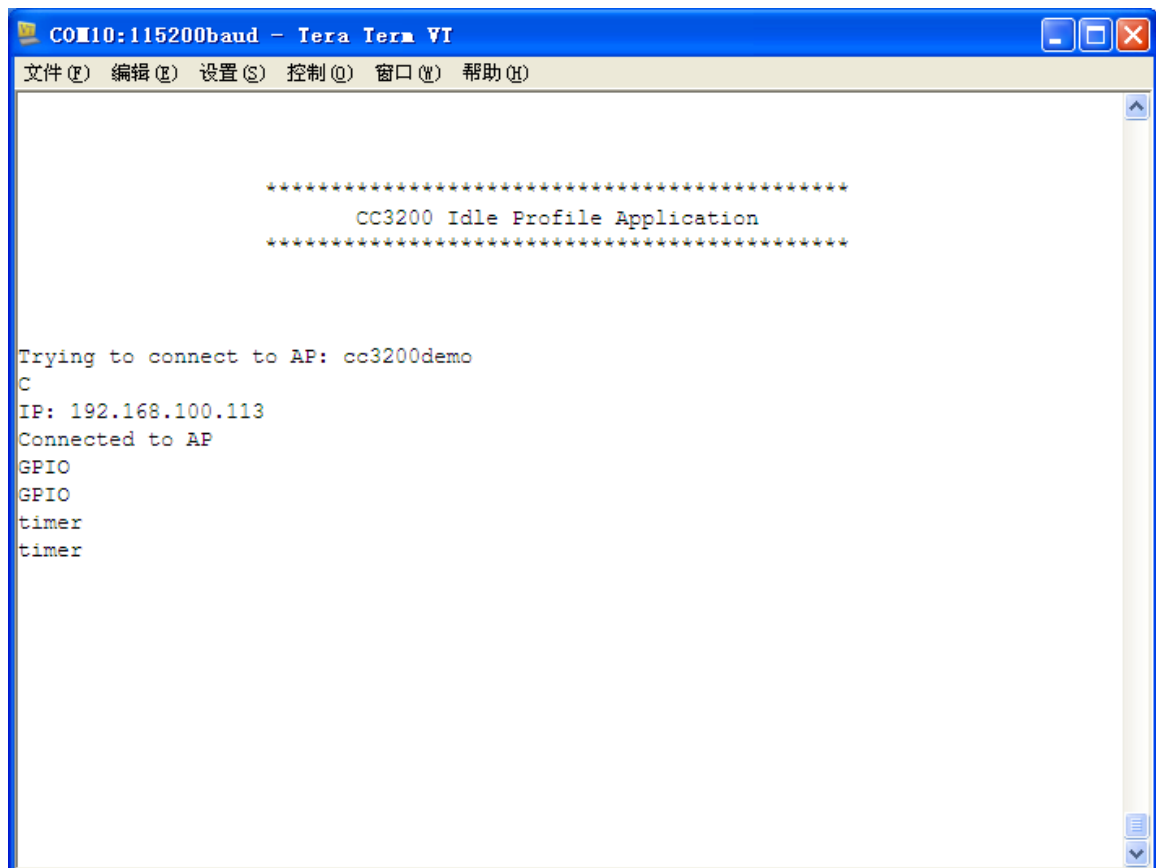


(2) 根据要求修改宏定义然后编译。

(3) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (4) 最好从 FLASH 运行应用程序而不是从 debug
- (5) 在终端上的输出告诉我们唤醒 LDPS 的原因
- (6) 只要设备进入 LPDS, 红色 LED 灯熄灭



4.20 File Download

4.20.1 概述和应用说明

此示例代码的目的是演示如何使用 HTTP 客户端库连接到一台 HTTP 服务器, 下载一个文件, 并将其保存在 Flash 中。这个应用可用于通过网络更新文件。

本应用程序允许用户执行以下操作:


- 连接到 AP 接入点
- 连接到 www.ti.com 服务器
- 从服务器下载 12.8 KB 文件
- 将下载的文件作为临时文件保存到 Flash 中
- 验证文件下载是否完成

用户可以通过更改宏定义中的“PREFIX_BUFFER”, “POST_BUFFER” 和 “HOST_NAME” 来更改服务器和对应文件。改变保存的文件大小可以通过改变宏定义中“SIZE_40K” 来实现, 但需要删除现有以“FILE_NAME” 宏定义指定的文件, 需要重新指定对应大小的文件名。

Note: 此应用程序需要连接能够正常接入互联网的 AP

4.20.2 使用说明

- (1) 打开 ‘main.c’ 更改 ‘SSID_NAME’, ‘SEC_TYPE’ 和 ‘PASSKEY’ 为你当前所连接的 AP 的 SSID、加密方式和密码。
- (2) 编译并运行工程, 应用程序将会尝试连接指定的 AP
- (3) 设备正常连接上 AP 之后, 应用程序尝试连接 www.ti.com
- (4) 设备正常连接上服务器后, 应用程序请求一个以 ‘FILE_NAME’ 为文件名的文件, 并将其保存到 Flash 中
- (5) 当以上都成功时, 超级终端将会显示下面的信息



```
file_download - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]

*****
CC3200 File Download Application
*****

Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.4.2.1.5.3.3
Device is configured in default state
Device started as STATION
[WLAN EVENT] STA Connected to the AP: cc3200demo , BSSID: 0:27:19:23:4c:56
[NETAPP EVENT] IP Acquired: IP=192.168.1.34 , Gateway=192.168.1.1
Connected to the AP: cc3200demo
Connection to server created successfully
Start downloading the file
Content length: 13173
Total bytes received: 13173
Downloading File Completed

Thank you
_

已连接 0:09:04 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

4.21 Deep-sleep

4.21.1 概述

深度睡眠状态是由 CC3200 设备所支持的低功耗模式中的一个。通过适当配置 PRCM 模块寄存器相关选项可以使能模块进入低功耗模式或者退出低功耗模式。其他未被激活的时钟将会被禁用，在深度睡眠状态下，激活的时钟会减半运行。

以下模块可以使能/禁用时钟：

- (1) UART
- (2) UDMA
- (3) TIMER
- (4) WDT
- (5) GPIO
- (6) I2C

此外，在深睡眠模式中，由 NWP 触发 WiFi 网络上的活动，也可以选择作为唤醒源。

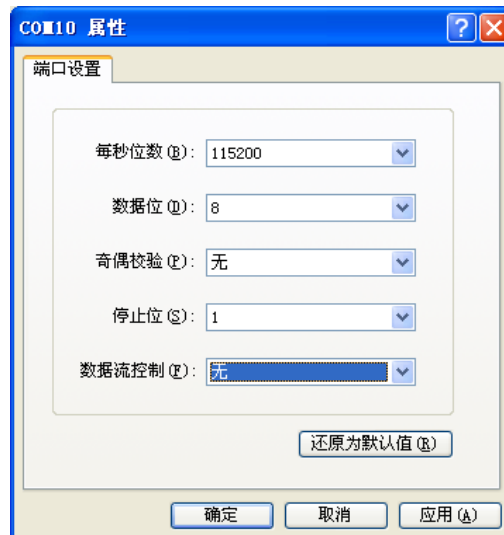
4.21.2 应用说明

本程序的目的是展示在联网的情况下（作为 TCP Server）进入深度睡眠，以实现低功耗。在这个例子中利用 FreeRTOS 提供给应用程序的空闲任务钩子函数进入深度睡眠模式。每当网络上有活动，该设备将会退出深度睡眠状态并提供相关网络服务，因此当网络相关 API 被拦截系统处于空闲情况下，设备将进入深度睡眠状态

4.21.3 使用说明

Note: 注意针对此设备建议将程序编译为 .bin 文件并烧写到 Flash 上进行测试。如果通过 IAR/CCS 等进行在线调试当设备进入低功耗状态将会断开与调试器的连接，后续的调试将无法进行。

- (1) 设置串口通信程序（HyperTerminal 或 TeraTerm），在 PC 主机上超级终端设置如下

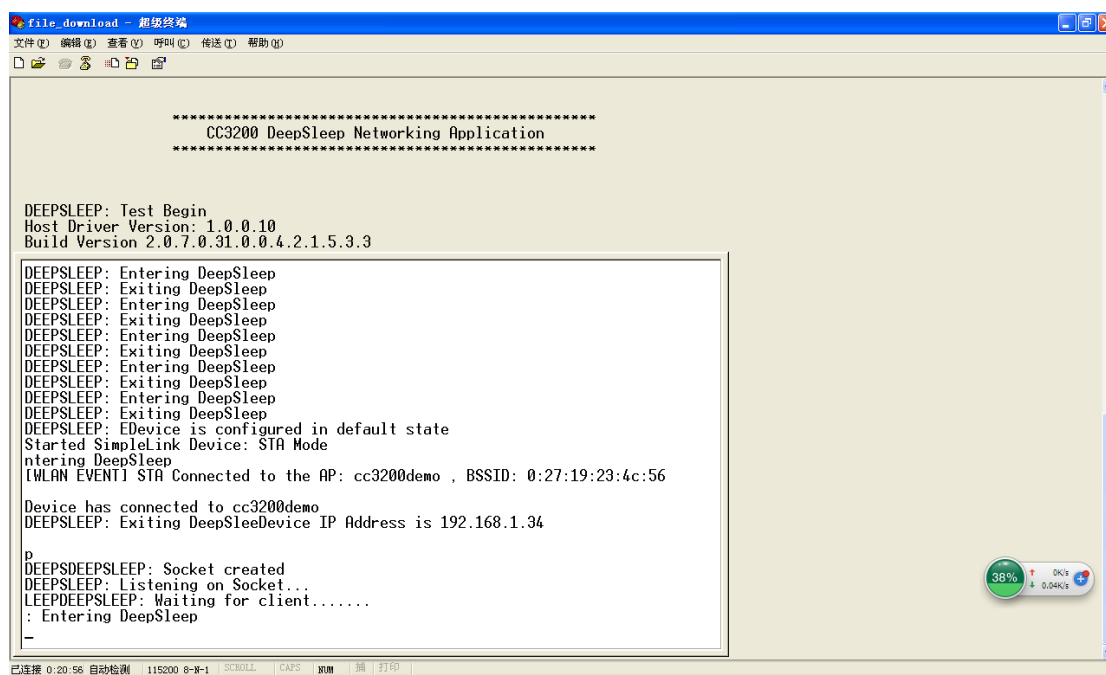


- (2) 将应用程序编译生成 .bin 文件，并烧写到 Flash 中，启动应用程序
- (3) 通过主机串口观察运行状态，了解程序执行流程
- (4) 默认情况下，程序尝试连接名字为 DemoAP 的 AP 接入点，如果连接失败，系统将会通过终端提示用户输入新的 AP 名称（必须是未加密的开放 AP）
- (5) 当程序正常运行将会启动一个 TCP Server 并等待 Client 连接。可以使用网络调试助手类似的外部网络应用程序进行连接，TCP Server 端口号为 5001，IP 地址从串口输出信

息中获取。

(6) 每当一个新的客户端进行连接将会通知系统退出 deep sleep，还有一些其他的是设备进入深度睡眠的实例。

应用程序在设备上正常执行的终端截图如下：



```
*****
CC3200 DeepSleep Networking Application
*****

DEEPSLEEP: Test Begin
Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.4.2.1.5.3.3

DEEPSLEEP: Entering DeepSleep
DEEPSLEEP: Exiting DeepSleep
DEEPSLEEP: Entering DeepSleep
DEEPSLEEP: Exiting DeepSleep
DEEPSLEEP: Entering DeepSleep
DEEPSLEEP: Exiting DeepSleep
DEEPSLEEP: Entering DeepSleep
DEEPSLEEP: Exiting DeepSleep
DEEPSLEEP: Entering DeepSleep
DEEPSLEEP: Exiting DeepSleep
DEEPSLEEP: EDevice is configured in default state
Started SimpleLink Device: STA Mode
ntering DeepSleep
[WLAN EVENT] STA Connected to the AP: cc3200demo , BSSID: 0:27:19:23:4c:56

Device has connected to cc3200demo
DEEPSLEEP: Exiting DeepSleepDevice IP Address is 192.168.1.34

p
DEEPSDEEPSLEEP: Socket created
DEEPSLEEP: Listening on Socket...
LEEPSDEEPSLEEP: Waiting for client.....
: Entering DeepSleep
-
```

4.22 Hibernate

4.22.1 概述

用户应用程序可以通过调用 `PRCMHibernateEnter()` API 进入“休眠模式(HIB)”。这个 API 将会使模块进入 HIB 模式，当进入 HIB 模式后会禁用所有模块的电源只保留 32KHz 时钟在运行。该 HIB 模块支持 RTC 定时器使能去激活，一旦启动其将会在整个 HIB 活动周期内自由运行。在进入 HIB 模式后，系统只能被 Wake Timer 唤醒定时器唤醒。只有在 HIB 模块被激活的情况下系统大部分 GPIO 是关闭的包括网络处理器。该 HIB 模块有 2 个 32-bit 寄存器，用于保存整个进入和退出 HIB 模式的所有信息。该代码从复位向量开始执行。

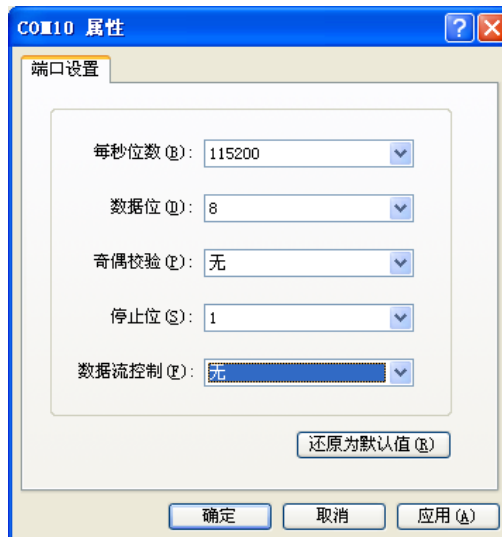
4.22.2 应用说明

本程序的目的是展示设备在联网的情况下（作为 UDP Client）使用休眠达到低功耗。在这个例子中，设备从休眠状态定期唤醒并广播一个消息然后再进入休眠状态。

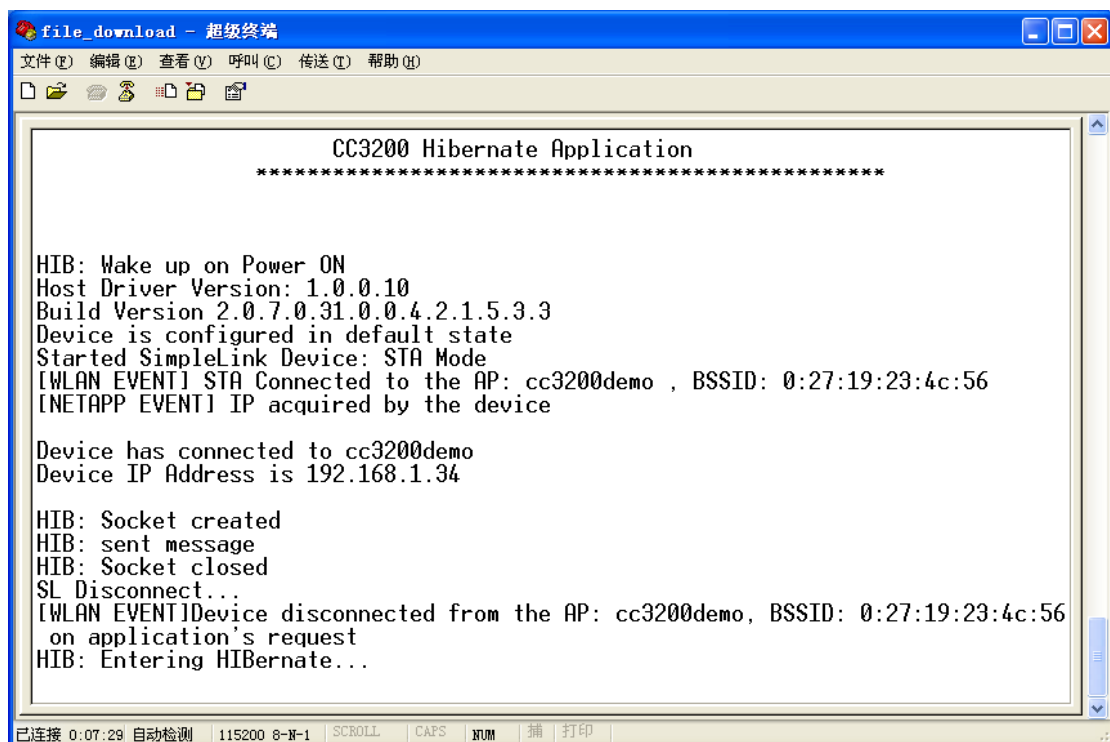
4.22.3 使用说明

Note: 注意针对此设备必须将程序编译为 .bin 文件并烧写到设备 Flash 中进行测试。如果通过 IAR/CCS 等在线调试，当设备进入休眠模式后，将会断开与调试器的连接，将无法进行后续调试。

(1) 配置串口应用程序(HyperTerminal 或 TeraTerm)，在 PC 主机具体的配置如下图



- (2) 将程序编译生成.bin 文件，并烧写到设备 Flash 中
- (3) 如果从 IDE 在线调试运行，当设备进入休眠状态，将会断开与 IDE 连接
- (4) 通过主机串口观察状态信息，了解程序的执行流程
- (5) 在默认情况下该应用将会尝试连接名字为 DemoAP 的 AP，如果使用默认情况连接失败，系统将会在 HyperTerminal 或 TeraTerm 中提示用户输入 AP 名称（必须为未加密的开放 AP）
- (6) 可以利用网络调试助手等能够接收 UDP 广播消息的外部工具监听 UDP 广播，广播的端口号为：5001. 以下是在设备运行此应用程序的超级终端截图



4.23 Watchdog System Demo

4.23.1 概述

当超时值到达时，看门狗定时器产生一个中断或复位。看门狗定时器用于重新控制当操作系统由于软件错误或在预期的时间内外部设备回应失败。

4.23.2 应用说明

这个应用程序展示了在完整的带有 MCU 的系统和网络子系统中看门狗定时器的使用。应用程序的目的是展示当系统失败时，看门狗定时器恢复完整的系统，包括网络子系统的功能，这里，使用发送 20 个 UDP 数据包并且看门狗没有响应来模拟系统故障。

从看门狗复位退出时，系统立即请求休眠一段时间，并且从休眠返回恢复其全部功能。确保从任何复杂的逻辑场景包括网络子系统恢复

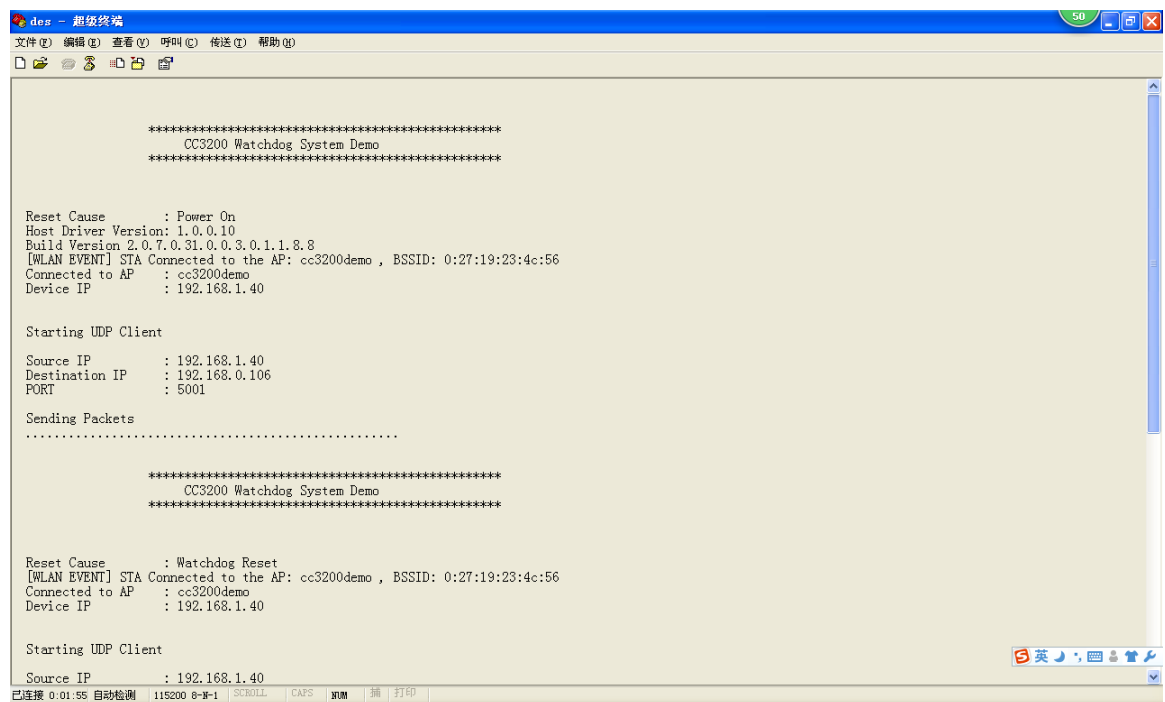
4.23.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



(2) 运行程序 (Flash the bin).

(3) 显示结果如下：



```
des - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]

*****
CC3200 Watchdog System Demo
*****

Reset Cause       : Power On
Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.3.0.1.1.8.8
[WLAN EVENT] STA Connected to the AP: cc3200demo, BSSID: 0:27:19:23:4c:56
Connected to AP   : cc3200demo
Device IP        : 192.168.1.40

Starting UDP Client

Source IP       : 192.168.1.40
Destination IP  : 192.168.0.106
PORT           : 5001

Sending Packets
.....

*****
CC3200 Watchdog System Demo
*****

Reset Cause       : Watchdog Reset
[WLAN EVENT] STA Connected to the AP: cc3200demo, BSSID: 0:27:19:23:4c:56
Connected to AP   : cc3200demo
Device IP        : 192.168.1.40

Starting UDP Client

Source IP       : 192.168.1.40

已连接 0:01:55 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
```

4.24 TFTP Client

4.24.1 概述

简单文件传输协议(TFTP)是一个简单的、同步的、允许客户端获取或放置一个文件到远程主机的文件传输协议

4.24.2 应用说明

这个应用程序演示了客户端应用程序如何从 TFTP 服务读文件或者向 TFTP 服务写文件，首先，应用程序从服务读一个已经存在的文件并且写文件到设备的 sFlash 的内存中。然后，应用程序向服务写不同名字但是同样内容的文件。默认的设置是被定义在下列宏定义中，可以在源代码中修改或者在运行时修改。

```
#define SSID          "cc3200demo"
#define TFTP_IP       0xc0a80164 /* TFTP server IP */
#define FILE_SIZE_MAX (20*1024) /* Max File Size set to 20KB */
```

4.24.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 使 PC 连接到设备连接的 AP
- (3) 获取 PC 的 IP 地址然后修改 TFTP_IP 宏定义的值 of PC 的 IP, 然后编译应用程序 (IAR/CCS)
- (4) 运行应用程序 (flash/IAR/CCS)。
- (5) 终端显示结果:



4.25 Idle Profile (Non OS)

4.25.1 概述

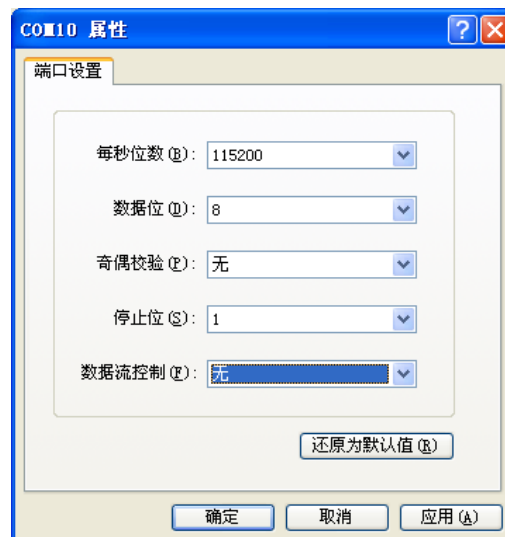
闲置配置文件允许 CC3200 用户测量 CC3200 的当前值, 功耗等参数, 当设备基本上是空闲的 (数值天气预报和应用子系统在低功率深度睡眠条件)。应用程序的另一个目的是介绍用户可以很容易配置电源管理框架。

4.25.2 应用说明

应用程序展示了设备的输入和输出处于深度睡眠模式期间按照每个用户的要求使用电源管理框架 (非 OS 框架)。这个应用程序为用户提供更多的控制。重新配置外围设备 (从深度睡眠模式返回的设备) 的责任留给了应用程序。关于更多的电源管理信息, 请参考 CC32xx Power Management Framework. pdf

4.25.3 使用说明

- (1) 根据要求修改宏定义然后重新编译。
- (2) 设置一个串行通信程序 (超级终端/ TeraTerm)。终端设置如下:



- (3) 最好从 FLASH 运行应用程序而不是从 debug
- (4) 输入命令, 如图所示:


```
COM10:115200baud - Tera Term VT
文件(F) 编辑(E) 设置(S) 控制(O) 窗口(W) 帮助(H)

*****
CC3200 Idle Profile Non Os Application
*****

Trying to Connect to AP: cc3200demo ...
C
IP: 192.168.100.113
enter one of the following command:
sleep - for putting the system into LPDS mode
        GPIO 13 and timer(5 sec) are the wk source configured
recv  - for receiving 1000 UDP packets
send  - for broadcasting 1000 UDP packets
cmd#sleep
cmd#send
Sent 1000 packets successfully
cmd#recv
Recieved 8 packets successfully
timed out
cmd#
```

4.26 MQTT Client

4.26.1 概述

MQTT(消息队列遥测传输)协议是一个轻量级连接协议。它是基于发布/订阅消息传递模型而设计 TCP / IP 协议。协议的特点包括代码占用空间小, 网络带宽要求低, 还用响应时间短, 功耗低, 容易扩展。这个协议的关键点包括代码占用空间小, 网络带宽要求低。其他功能包括更快的响应时间, 低功率需求, 良好的可伸缩性。这些优势, 使其成为实现物联网应用的嵌入式设备通信协议中一个理想的选择。我们可以从最新的 MQTT 协议规范中获得更多关于 MQTT 协议的信息。

4.26.2 应用说明

MQTT 网络

一个简单的 MQTT 网络包含一个服务器/代理(如中央枢纽), 可以处理来自多个客户端连接。每个连接的客户端都可以使用令牌发布数据。然而, 服务器/代理负责转发客户端发布的数据给指定客户端。这是 SDK 中一个非常简单的描述 MQTT 网络的应用程序。

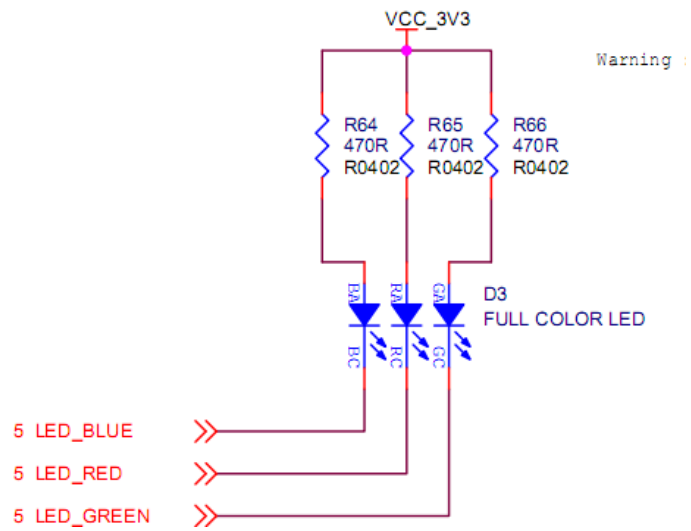
MQTT 库

MQTT 库抽象了底层 MQTT 错综复杂的网络, 为用户提供一个直观的, 易于在 CC3200 的设备实现 MQTT 协议的 API。

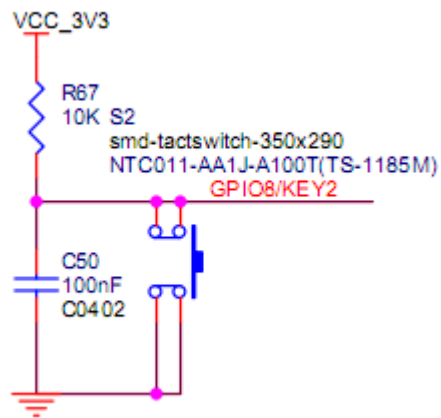
应用程序

这个应用程序利用 MQTT 客户断 API 库和 IBM 代理的客户端之间使用命令进行通信。CC3200 设备上面的可以通过 web 客户端发布的命令来控制三个 led 灯。同样, 也可以使用预先配置在代码中的 CC3200 设备上按键来发送信息

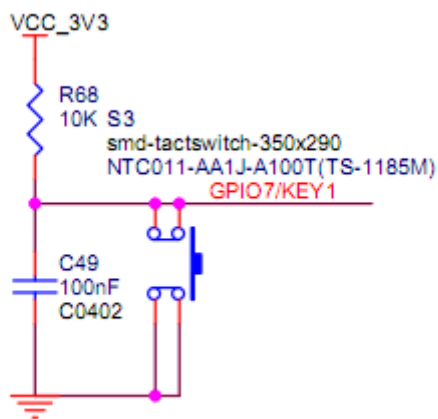




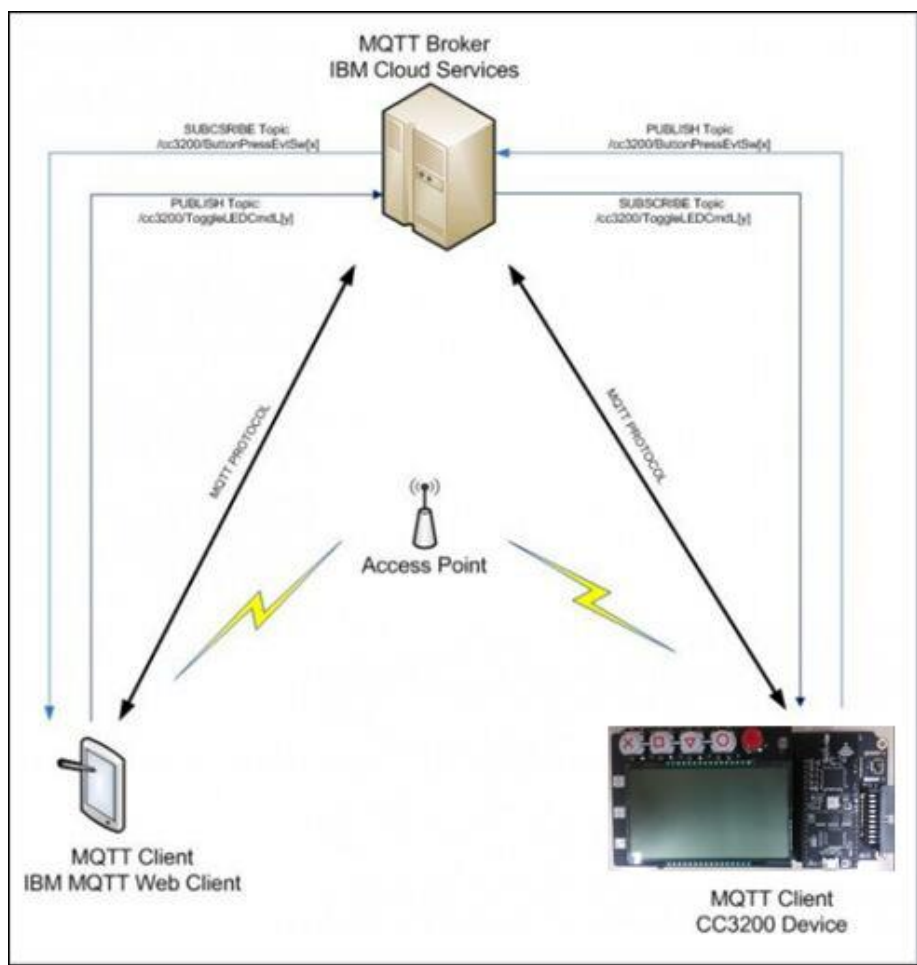
Led 图



按键 2



按键 3



4.26.3 使用说明

(1) 设置一个串行通信程序(超级终端/ TeraTerm)。终端设置如下：



- (2) 从 WEB 客户端发送 `/cc3200/ButtonPressEvtSw2` 和 `/cc3200/ButtonPressEvtSw3` 命令。
- (3) 运行应用程序 (Flashing the bin/IAR/CCS).
- (4) 应用程序要求 AP 是可以连接网络的
- (5) 如果连接 AP 不成功，终端上会提示输入 AP 格式的详细信息。

- (6) 网络处理成功绿色 LED 灯被点亮，否则红色 LED 灯被点亮，连接建立后红灯亮 1s, 然后所有灯熄灭
- (7) 在另外一台电脑上，打开一个浏览器在地址栏中输入 `http://m2m.demos.ibm.com/mqttclient/`，将会访问一个 IBM 的客户端。
- (8) 这个 web 客户端通过填写必填字段连接到 IBM 代理（使用下面的截图作为参考字段值）。
- (9) 通过发布 `/cc3200/ToggleLEDCmdL1`，`/cc3200/ToggleLEDCmdL2`
- (10) 或者 `/cc3200/ToggleLEDCmdL3` 来分别控制红色 LED，绿色 LED 和蓝色 LED.
- (11) 可以通过按 S2 或者 S3 来分别发布 `/cc3200/ButtonPressEvtSw2` 或 `/cc3200/ButtonPressEvtSw3` 信息.
- (12) 下面是截图信息：

MQTT Client

Connected!

Connect

Server

messagesight.demos.ibm.com

Port

1883

Connect

Disconnect

Client ID

Client80248

Username

(optional)

Password

(optional)

Clean Session

OFF

ON

SSL

OFF

ON

Subscribe

Publish

MQTT Client

Connected!

Connect

Subscribe

Topic

/cc3200/ButtonPressEvtSw3

QOS

1

Subscribe

/cc3200/ButtonPressEvtSw2

/cc3200/ButtonPressEvtSw3

Publish

Topic

/cc3200/ToggleLEDCmdL3

Message

toggle the state of GREEN LED

QOS

1

Retained

OFF

ON

Publish

Log 11

Clear

Follow

```

(04:46:21.648) Connected to messagesight.demos.ibm.com:1883
(04:46:25.870) >> [/cc3200/ButtonPressEvtSw2][qos 1][retained] Push button sw2 is pressed on CC32XX device
(04:46:25.873) Subscribed to [/cc3200/ButtonPressEvtSw2][qos 1]
(04:46:29.756) >> [/cc3200/ButtonPressEvtSw3][qos 1][retained] Push button sw3 is pressed on CC32XX device
(04:46:29.759) Subscribed to [/cc3200/ButtonPressEvtSw3][qos 1]
(04:46:34.638) << [/cc3200/ToggleLEDCmdL1][qos 1] toggle the state of RED LED
(04:46:43.012) << [/cc3200/ToggleLEDCmdL2][qos 1] toggle the state of ORANGE LED
(04:46:54.550) << [/cc3200/ToggleLEDCmdL3][qos 1] toggle the state of GREEN LED
(04:46:59.589) >> [/cc3200/ButtonPressEvtSw3][qos 1] Push button sw3 is pressed on CC32XX device
(04:47:01.690) >> [/cc3200/ButtonPressEvtSw2][qos 1] Push button sw2 is pressed on CC32XX device
(04:47:02.598) >> [/cc3200/ButtonPressEvtSw2][qos 1] Push button sw2 is pressed on CC32XX device

```

终端显示：

```
1 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]

*****
CC3200 MQTT_Client Application
*****

Host Driver Version: 1.0.0.10
Build Version 2.0.7.0.31.0.0.3.0.1.1.8.8
Device is configured in default state
Started SimpleLink Device: STA Mode
[WLAN EVENT] STA Connected to the AP: cc3200demo , BSSID: 0:27:19:23:4c:56
[NETAPP EVENT] IP acquired by the device

Device has connected to cc3200demo
Device IP Address is 192.168.1.49

Version: Client LIB 1.0.3, Common LIB 1.1.1.
C: FH-B1 0x10 to net 17, Sent (45 Bytes) [0 1574]
C: Rcvd msg Fix-Hdr (Byte1) 0x20 from net 17 [0 1575]
C: Cleaning session for net 17
C: Msg w/ ID 0x0000, processing status: Good

Success: conn to Broker no. 1
C: FH-B1 0x82 to net 17, Sent (79 Bytes) [0 1575]
C: Rcvd msg Fix-Hdr (Byte1) 0x31 from net 17 [0 1575]

Publish Message Received
Topic: /cc3200/ToggleLEDCmdL1 [Qos: 0] [Retained]
Data is: Hello world!
C: Msg w/ ID 0x0000, processing status: Good
C: Rcvd msg Fix-Hdr (Byte1) 0x35 from net 17 [0 1575]
C: FH-B1 0x50 to net 17, Sent (4 Bytes) [0 1575]

Publish Message Received
Topic: /cc3200/ToggleLEDCmdL2 [Qos: 2] [Retained]
Data is: Hello world!

已连接 0:01:04 自动检测 115200 8-N-1 SCROLL CAPS 关闭 编辑 打印
```

44%

0.05K/s

0.1K/s