



Foxes 开发板 NCJ29D5D 使用 Lauterbach 下载调试

WPI-ARM



Foxes 是一套基于 BLE 芯片 KW38 和 UWB 芯片 NCJ29D5D 的 UWB 开发套件，其中 UWB 芯片 NCJ29D5D 是一种全集成的单芯片脉冲无线电超宽带 (IR-UWB) 低能低成本收发芯片。它是为汽车环境中的安全范围应用而设计的。它可以用于双向测距定位系统，以确定距离的精度高达 10 厘米。该设备配备了专用的 MAC 固件，以简化超宽带范围设置和会话控制，能够按照汽车连接联盟 (CCC) 的指定操作，以实现与汽车接入生态系统的互操作性。

一、芯片介绍

1.1 功能和特点

- 符合 IEEE 802.15.4 HRP UWB PHY 标准
- 芯片上 MAC 固件，以减轻超宽带范围设置和会话控制
- 支持 SHF 超宽频带从 6.0 GHz 到 8.5 GHz 的全球使用
- 中心频率 6.5 GHz 到 8.0 GHz
- 可编程发射机输出功率高达 12 dBm 峰值
- 与 IEEE 802.15.4 相比，128 MHz PRF 模式使可用的平均 TX 功率加倍
- 优化短帧模式操作
- 全相干接收机的最大范围和精度
- 集成所有需要的射频元件(即 balun,TRX 开关)
- 供电电压 1.8 V ~ 3.6 V
- 低能源消耗
- 可配置的电流限制应用与硬币电池供应
- 数据速率 110 kbps (BPRF)， 850 kbps (BRRF)， 6.8 Mbps (BPRF)， 7.8 Mbps (HPRF)
- 支持双向测距
- IEEE 802.15.4z BPRF/HPRF 超宽带 PHY 协议草案，对抗 Cicada 和前导码注入攻击造成的远距离操纵
- 混乱时间戳序列 (STS) 生成兼容 NIST SP 800-90A
- 集成 IQ 相位和振幅失配补偿
- 窄带干扰消除 (NBIC)，具有卓越的同通道干扰抑制能力
- XO 缓冲器，用于与其他设备共享 XTAL(例如 UHF 或 BLE 收发器)
- 6mm x 6mm 40-pin QFN 包带 0.5 mm 铅间距和可湿性侧面
- 外部组件数量少
- ARM@Cortex-M33 32 位处理器 55.2 MHz
- ARM@AHB-Lite 总线矩阵和双主 NS-DMA 快速数据传输
- 256kbyte 非易失性内存
- 40 kByte RAM



- 96 kByte ROM
- ARM@TrustZone 技术和 S-DMA 安全
- 高级加密标准 AES (Advanced Encryption Standard)，具有 128 位密钥和 256 位密钥
- 具有 256 位、384 位和 512 位密钥的椭圆曲线密码 (ECC)
- 256 位的安全哈希算法 2 (SHA2-256)
- 安全接口(启用 GlobalPlatform 协议)
- 循环冗余校验 (CRC) 计算的协处理器
- 真随机数生成 (TRNG)
- SPI, UART 和 LIN 兼容接口
- 简化了定制应用程序的开发的 API
- 应用程序开发的几个计时器
- 低功耗唤醒计时器
- 看狗定时器
- 支持高精度外部温度传感器，精确的晶体温度漂移补偿
- 集成温度传感器用于晶体温度漂移补偿
- 通用 ADC 10 位

1.2 应用

这颗芯片支持以下应用：

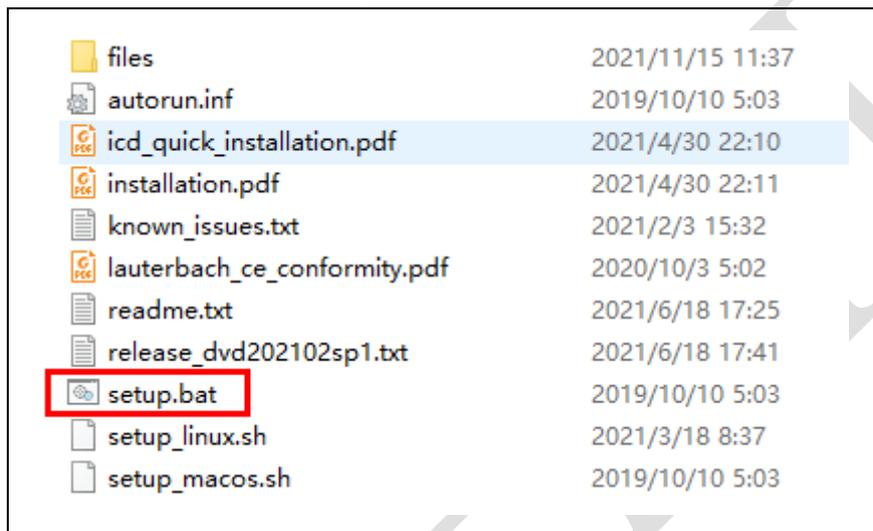
- 基于低频无钥匙进入系统 (PKE) 的中继攻击防御
- 基于超宽带定位的免提智能接入
- 安全测距在汽车系统中的应用 (远程停车控制)
- 更多的用于汽车中的舒适功能



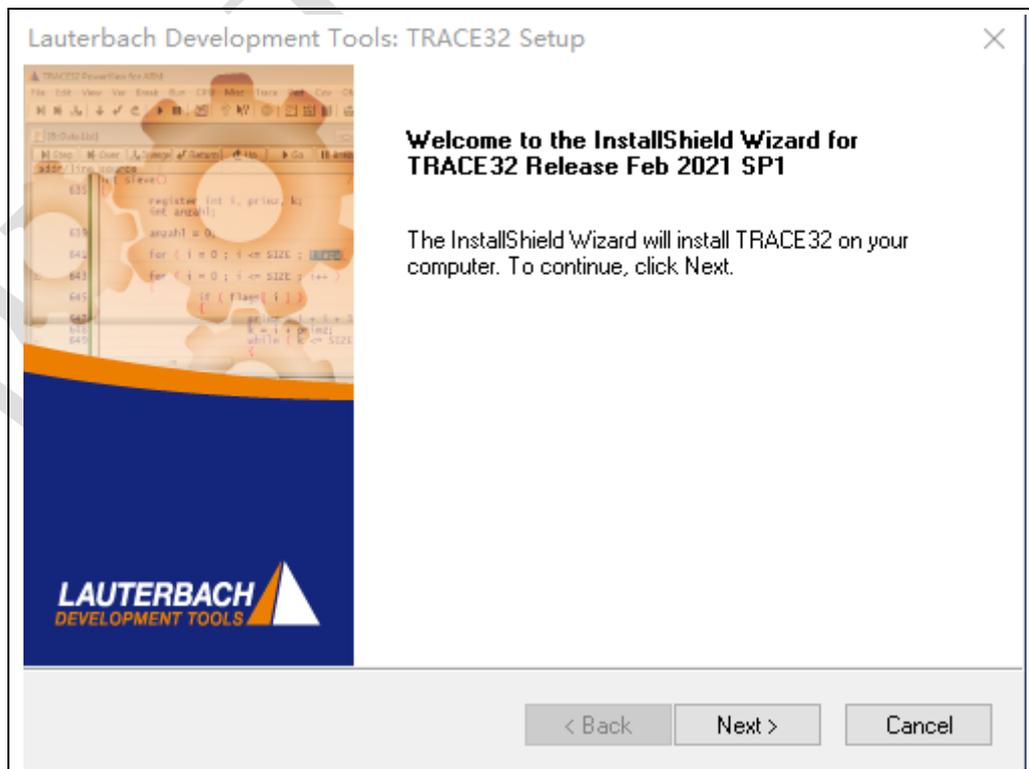
二、使用 Luterbach 进行调试和下载

2.1 安装 LAUTERBACH TRACE32 软件

(1) 从附送的光盘中拷贝得到安装软件,点击 setup.bat 进行安装。

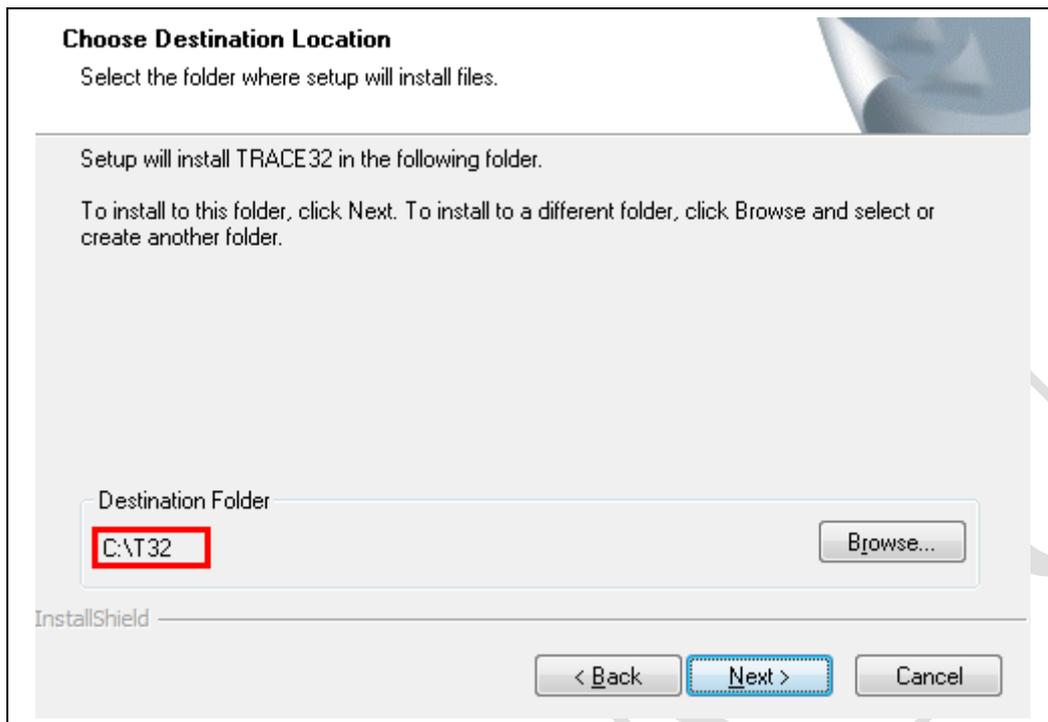


(2) 进入安程序点击 “Next” 进行安装。

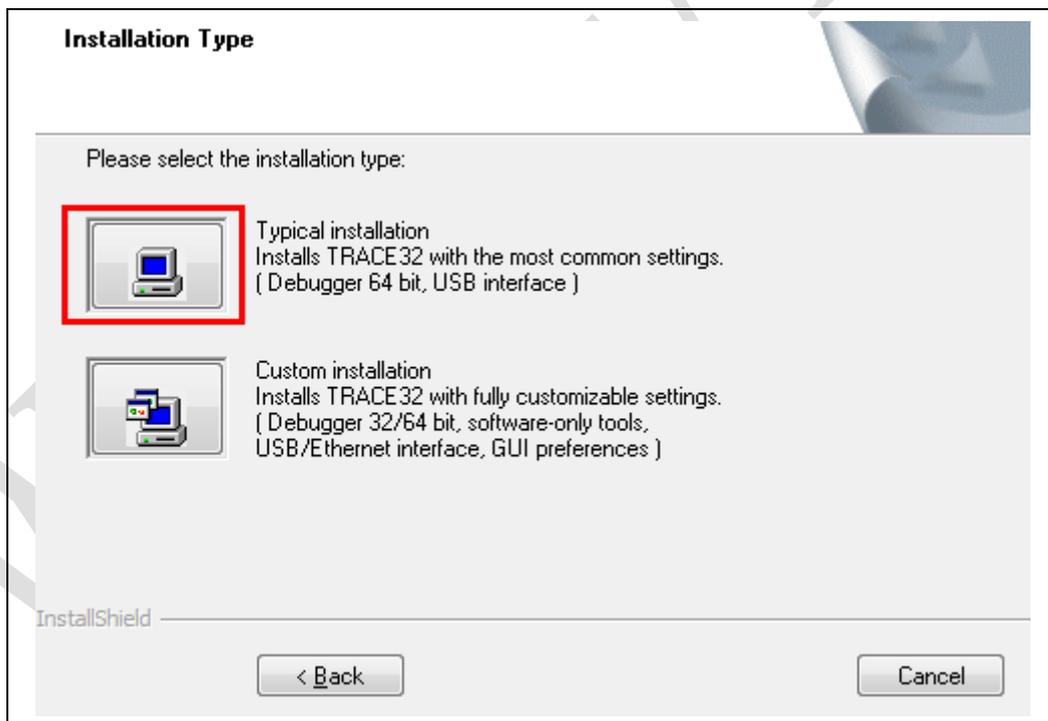




(3) 选择默认路径进行安装。

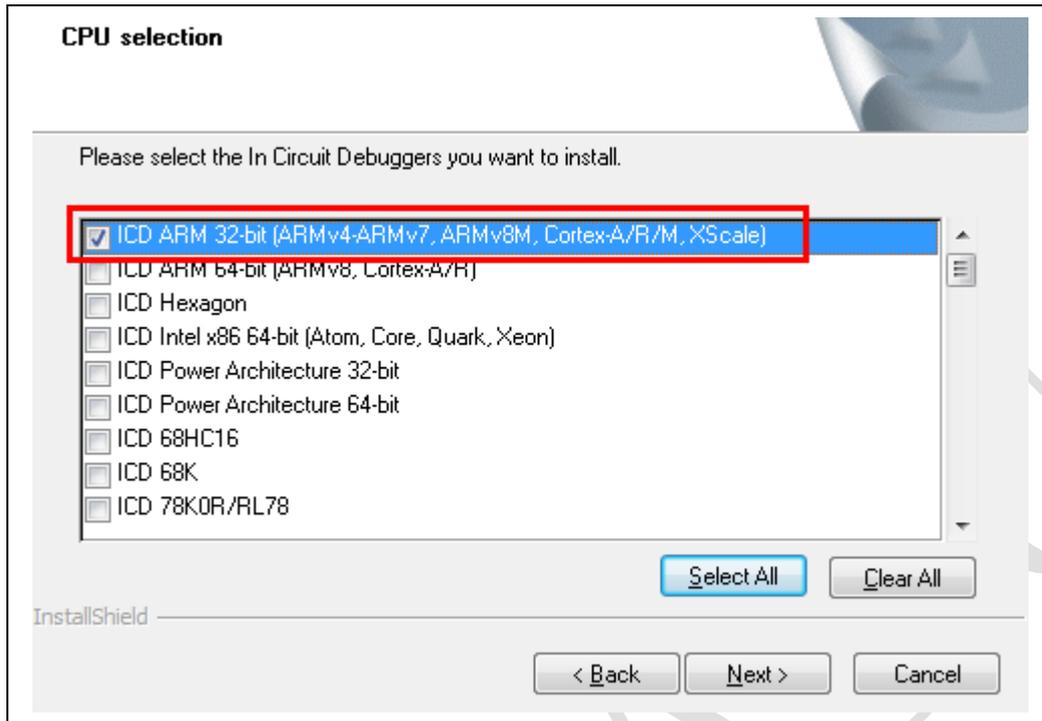


(4) 选择典型安装。





(5) 选择正确的 CPU 类型。

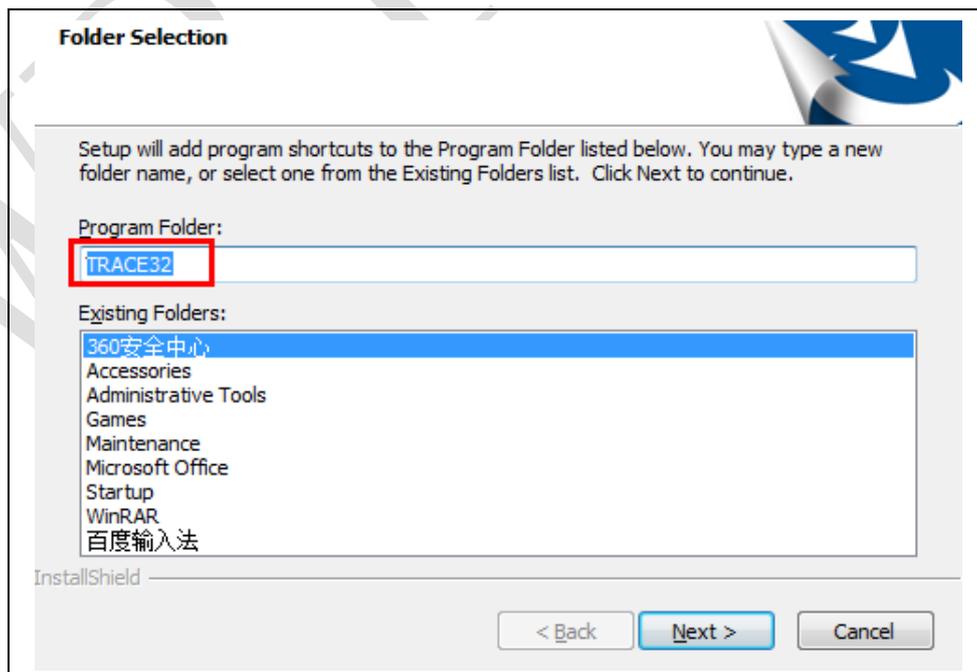


(6) 接下来安装驱动程序。



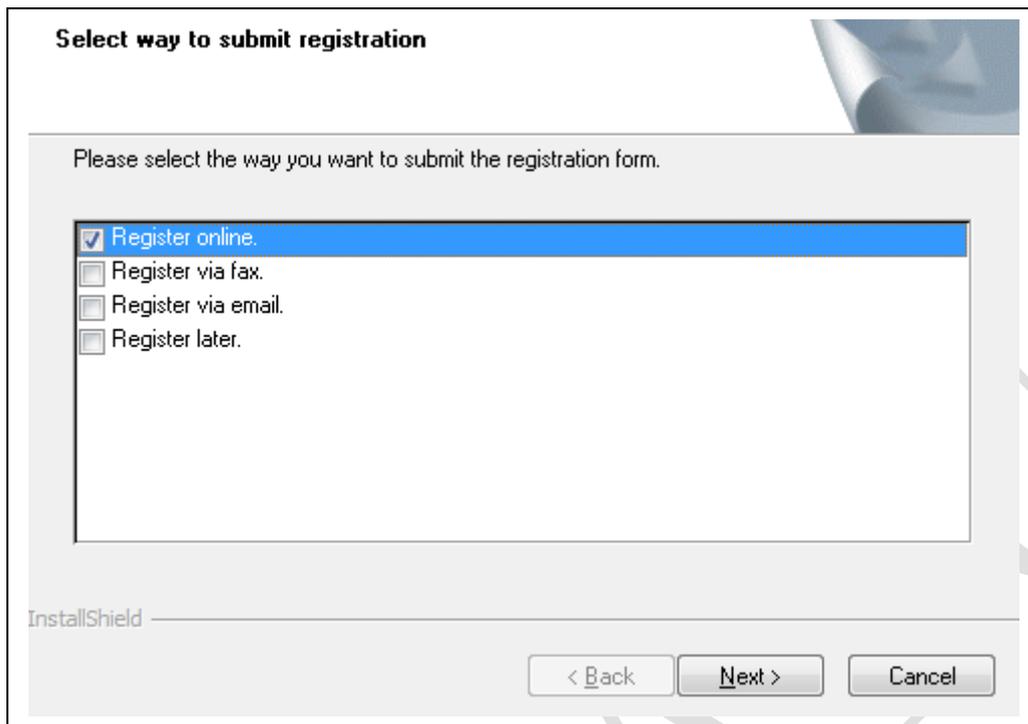


(7) 设置一个快捷目录名称。



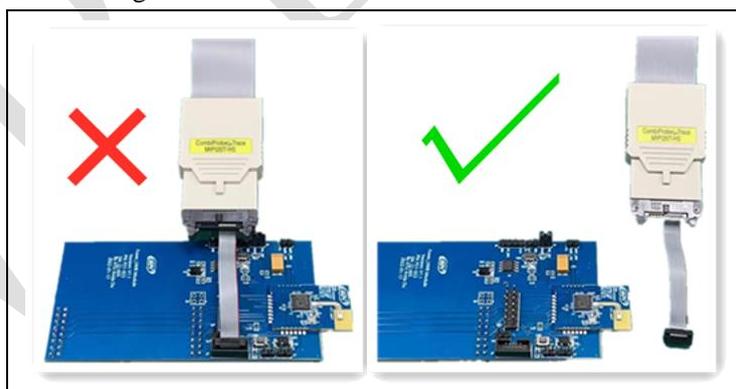


(8) 然后会询问你是否注册官网账号，不注册也可以使用 TRACE32 软件。



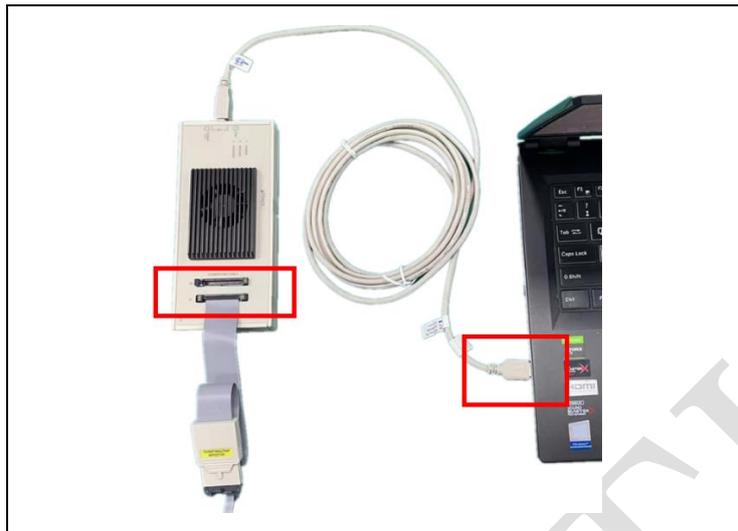
2.2 硬件连接

(1) 在目标板上电之前，确保 Debug cable 已经与目标板断开





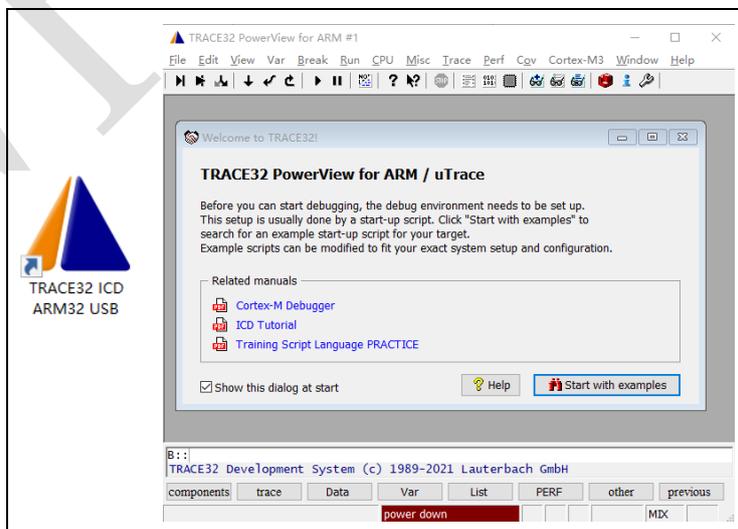
(2) TRACE32 连接电脑，把 Debug cable 接到 TRACE32 的 A 或 B 连接槽



(3) 连接 TRACE32 硬件的电源

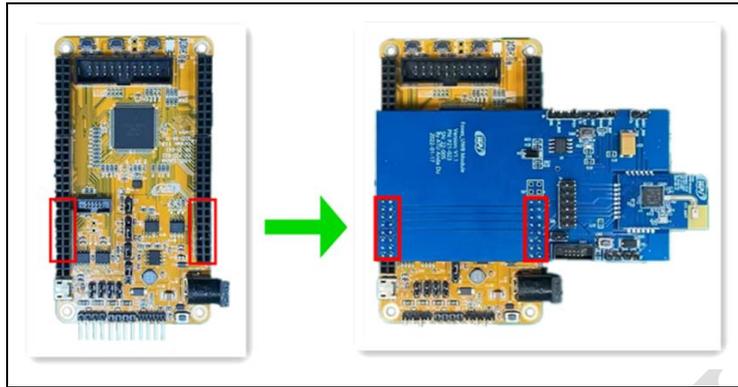


(4) 启动 TRACE32 的软件

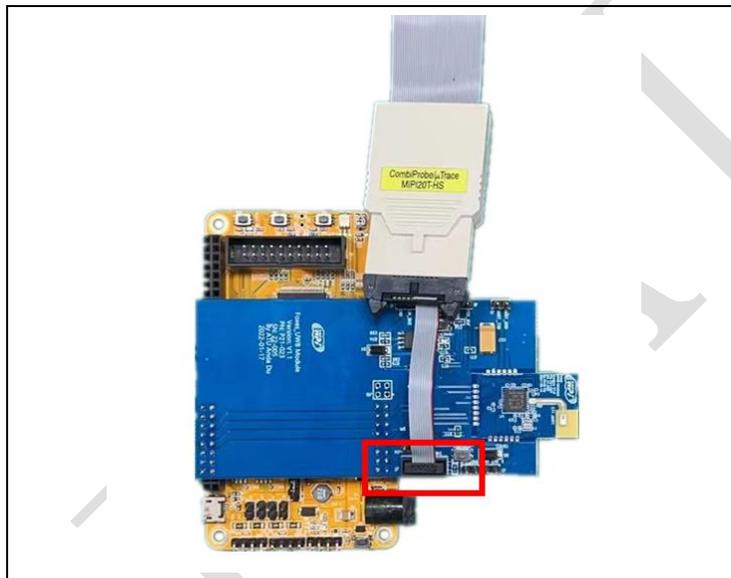




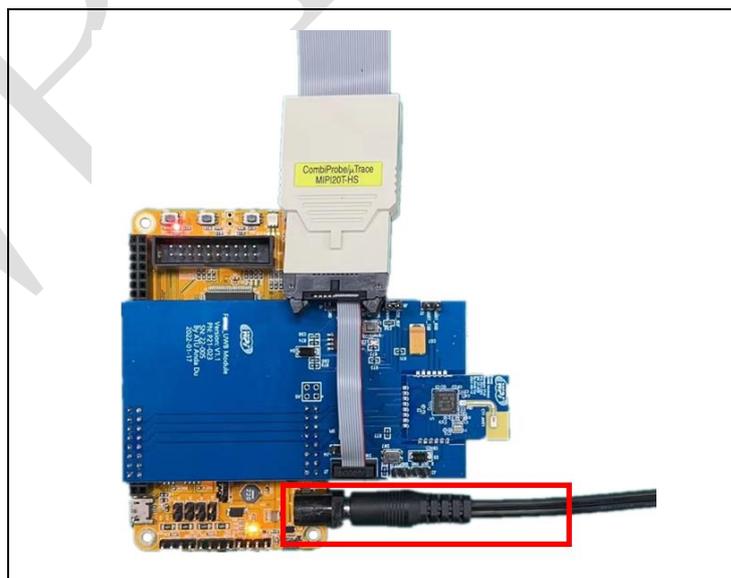
(5) 根据对应接口把 UWB 板连接到 S32K 板



(6) 把 Debug cable 连接到目标板



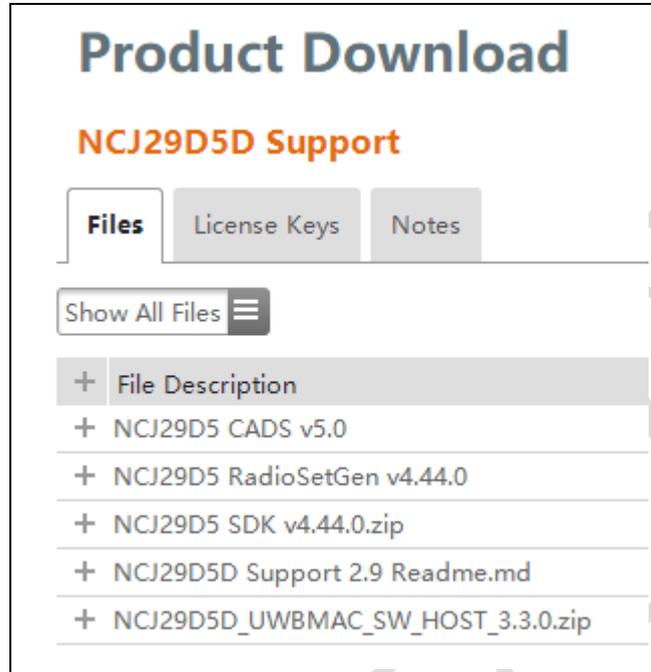
(7) 连接目标板的电源



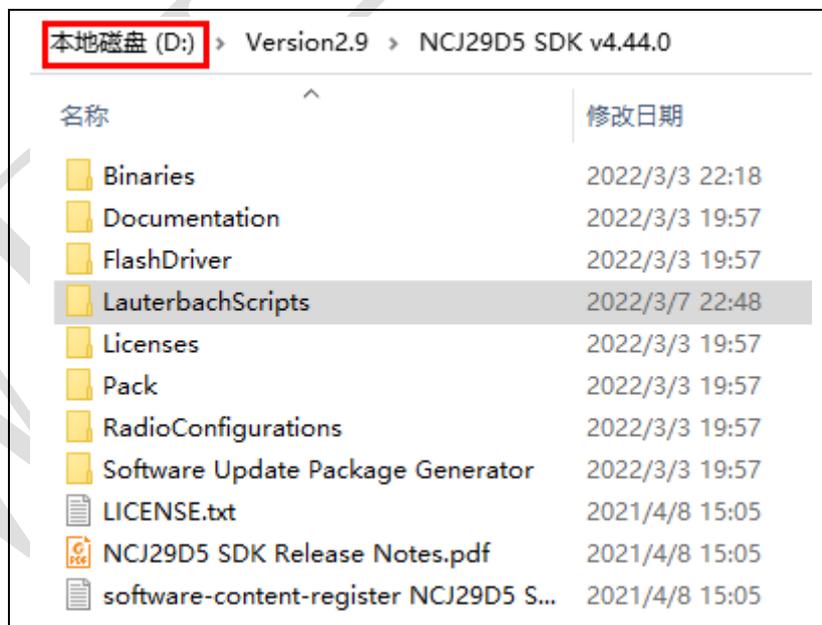


2.3 下载固件到 UWB 板

(1) 首先需要向 NXP 申请 UWB 资料权限，然后就可以去官网下载资料了，这里下载的 2.9 版本。

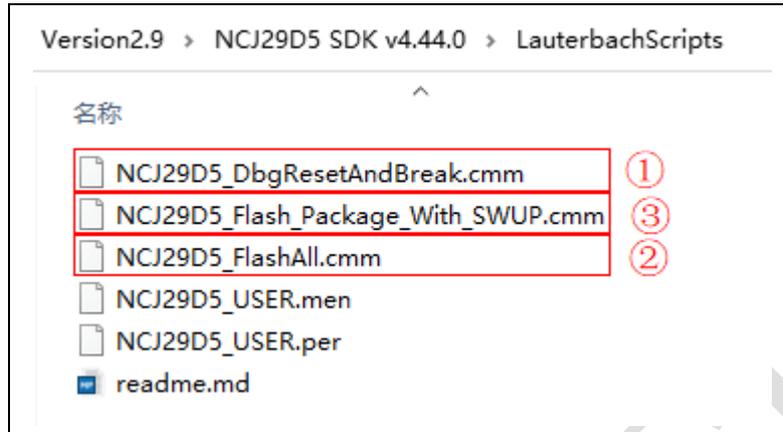


(2) 准备好相关资料包，最好放到无中文且比较短的路径下，比如 D 盘根目录。





(3) 进入 NCJ29D5 SDK v4.44.0 目录中, LauterbachScripts 是能够运行在 Lauterbach 上的脚本文件, 请按下图中 ①②③ 的顺序运行脚本, 进行下载。



① NCJ29D5_DbgResetAndBreak.cmm 脚本内容: 重启, 设置调试方式, 选择 CPU 类型以及进行一次错误测试。

```

NCJ29D5_DbgResetAndBreak.cmm 3
14 ; =====
15 ;
16 ; Detach (in case that we are already attached)
17 SYSTEM.DOWN
18 ;
19 ; OPTIONAL: Clear the previous system configuration
20 SYSTEM.RESET
21 ;
22 ; Select SWD debug port and Cortex-M33 CPU
23 SYSTEM.CONFIG.DEBUGPORTTYPE SWD
24 SYSTEM.CPU CortexM33
25 ;
26 ; add chip specific pull-down menu and peripheral register view
27 ; this assumes NCJ29D5_USER.men and NCJ29D5_USER.per are in the current directory
28 PER.ReProgram NCJ29D5_USER.per
29 MENU.ReProgram NCJ29D5_USER.men
30 ;
31 ; OPTIONAL: Force use hardware (onchip) breakpoints in user flash range (0x00200000..0x0021FFFF)
32 MAP.RESET
33 MAP.BONCHIP 0x00200000--0x0021FFFF
34 ;
35 ; OPTIONAL: Map secure domain memory as inaccessible by default (prevent access by TRACE32)
36 MAP.DENYACCESS 0x00000000--0x0000FFFF
37 MAP.DENYACCESS 0x20000000--0x20001FFF
38 ;
39 ; =====
    
```

- | | |
|------------------------------------|---|
| SYSTEM.DOWN | // 系统关闭。 |
| SYSTEM.RESET | // 系统重启。 |
| SYSTEM.CONFIG.DEBUGPORTTYPE SWD | // 选择调试 接口为 SWD。 |
| SYSTEM.CPU CortexM33 | // 选择 CPU 类型为 CortexM33。 |
| PER.ReProgram NCJ29D5_USER.per | // 指定设备的外围寄存器视图, 此文件需在当前目录下。 |
| MENU.ReProgram NCJ29D5_USER.men | // 指定设备的特定下拉菜单。 |
| MAP.RESET | // 映射前重置。 |
| MAP.BONCHIP 0x00200000--0x0021FFFF | // 在用户闪存 0x00200000--0x0021FFFF 强制使用硬件断点。 |



MAP.DENYACCESS 0x00000000--0x0000FFFF // 默认情况下, 将安全域内存映射为不可访问的
MAP.DENYACCESS 0x20000000--0x20001FFF //(即阻止 TRACE32 访问)。

```

NCJ29D5_DbgResetAndBreak.cmm
54 ;
55 SYSTEM.ATTACH
56
57 ; Disable the WDT as reset source (via non-invasive memory access)
58 D.SET EN:0x4000A834 %Long 0
59
60 ; At this point we do not know anything about the state of the ARM core (typically we are somewhere
61 ; in the middle of an application)
62 ;
63 ; To get a clean initial state we manually setup HW breakpoints on the possible application domain
64 ; entrypoints (first instruction in flash and safe mode RAM loop), then we trigger an SCB->AIRCR based reset
65 ;
66 ; NOTE: This is basically the same sequence that is used on the J-Link startup code
67 ;
68
69 ; Manually configure the Cortex-M33 breakpoint unit
70 ; FP_COMP0: Break @ 0x00200000 (flash entrypoint)
71 ; FP_COMP1: Break @ 0x20002000 (safe mode RAM loop entrypoint)
72 ; FP_CTRL: Enable breakpoints
73 ;
74 ;
75 DATA.SET EN:0xE0002008 %Long 0x00200001
76 DATA.SET EN:0xE000200C %Long 0x20002001
77 DATA.SET EN:0xE0002010 %Long 0x20004001
78 DATA.SET EN:0xE0002014 %Long 0x20005001
79 DATA.SET EN:0xE0002000 %Long 0x10000083
80
81 ; Trigger a software reset via SCB->AIRCR
82 ;
83 ; NOTE: The following operation will report a "Bus Error" in the TRACE32 status line (which is expected
84 ; behavior, as we are initiating a type of software reset via SCB.AIRCR); the ON ERROR and ERROR.RESET statement
85 ; clears the error status for this (expected) event in order to allow the script to proceed
86
87 ON ERROR CONTINUE
88 DATA.SET EN:0xE000ED0C %Long 0x05FA0004
89
90 ON ERROR INHERIT
91 ERROR.RESET
92
93 ; Ensure that we are attached and in halt state (again)
94 SYSTEM.ATTACH
95
96 ; Disable the watchdog reset source
97 DATA.SET EN:0x4000A834 %Long 0
98
99 ; Ensure that the system is halted again
100 BREAK
101
102 ; Done
103 ENDDO

```

SYSTEM.ATTACH	// 建立调试器与核心之间的通信。
D.SET EN:0x4000A834 %Long 0	// 禁用看门狗 (通过非侵入性内存访问)。
DATA.SET EN:0xE0002008 %Long 0x00200001	// 手动配置 Cortex-M33 断点单元。
DATA.SET EN:0xE000200C %Long 0x20002001	// 手动配置 Cortex-M33 断点单元。
DATA.SET EN:0xE0002010 %Long 0x20004001	// 手动配置 Cortex-M33 断点单元。
DATA.SET EN:0xE0002014 %Long 0x20005001	// 手动配置 Cortex-M33 断点单元。
DATA.SET EN:0xE0002000 %Long 0x10000083	// 手动配置 Cortex-M33 断点单元。



```

ON ERROR CONTINUE // 这里的操作将在 TRACE32 状态行中报告一个“总线错误”
DATA.SET EN:0xE000ED0C %Long 0x05FA0004 // ( 这是预期的行为, 因为我们正在通过 SCB.AIRCR 启动一
ON ERROR INHERIT // 种类型的软件重置 ); ON ERROR 和 ERROR。重新声明清
ERROR.RESET // 除此 ( 预期的 ) 事件的错误状态, 以便允许脚本继续执行。
SYSTEM.ATTACH // 确保我们被连接并处于停止状态。
DATA.SET EN:0x4000A834 %Long 0 // 关闭看门狗复位源。
BREAK // 确保系统停止。
ENDDO // 脚本完成。
    
```

② NCJ29D5_FlashAll.cmm 脚本内容: 烧写 RX TX 的射频配置信息, 以及 RCI 测距应用程序。

```

NCJ29D5_FlashAll.cmm
10 ; Clear the current flash configuration
11 FLASH.RESET
12
13 ; Define target flash algorithm
14 FLASH.TARGET 0x20004000 0x20008000 0x1000 "..\FlashDriver\NCJ29D5_Raw_FlashDriver.bin" /STACKSIZE 0x400
15
16 FLASH.CREATE 1. 0x00238a00--0x0023a9ff 0x2000 TARGET LONG /INFO "RX Radio Config"
17 FLASH.PROGRAM 0x00238a00--0x0023a9ff
18 PRINT "* Pre-loading Rx Cfgs. NCJ29D5 supports 16 Rx indices"
19 ; Radio Cfg. Index 0
20 DATA.LOAD.BINARY "../RadioConfigurations/Rx Eco Mini Data Phr Key Fob.bin" 0x00238a00..0x238bff
21 ; Radio Cfg. Index 1
22 DATA.LOAD.BINARY "../RadioConfigurations/Rx Eco Mini Ranging Data Phr Key Fob.bin" 0x00238c00..0x238dff
23 ; Radio Cfg. Index 2
24 DATA.LOAD.BINARY "../RadioConfigurations/Rx Eco Mini Ranging Key Fob.bin" 0x00238e00..0x238fff
25
26 ; Flash programming is done. Uninitialized the flash driver
27 FLASH.PROGRAM.OFF
28
29 FLASH.CREATE 2. 0x0023aa00--0x0023b9ff 0x1000 TARGET LONG /INFO "TX Radio Configs"
30 FLASH.PROGRAM 0x0023aa00--0x0023b9ff
31 PRINT "* Pre-loading Tx Cfgs. NCJ29D5 supports 16 Tx indices"
32 ; Radio Cfg. Index 16
33 DATA.LOAD.BINARY "../RadioConfigurations/Tx Eco Mini Data Phr Key Fob.bin" 0x0023aa00..0x23aaff
34 ; Radio Cfg. Index 17
35 DATA.LOAD.BINARY "../RadioConfigurations/Tx Eco Mini Ranging Data Phr Key Fob.bin" 0x0023ab00..0x23abff
36 ; Radio Cfg. Index 18
37 DATA.LOAD.BINARY "../RadioConfigurations/Tx Eco Mini Ranging Key Fob.bin" 0x0023ac00..0x23acff
38
39 ; Radio Cfg. Index 22
40 DATA.LOAD.BINARY "../RadioConfigurations/Tx DDFSCW.bin" 0x0023b000..0x23b0ff
41 ; Flash programming is done. Uninitialized the flash driver
42 FLASH.PROGRAM.OFF
43
44 ; Define flash region
45 FLASH.CREATE 3. 0x00200000--0x0021ffff 0x20000 TARGET LONG /INFO "Application Flash"
46 FLASH.PROGRAM 0x00200000--0x0021ffff
47 PRINT "* Pre-loading RCI Ranging Application"
48 ; Download the executable image; note that the target memory layout for AXF (ELF) images is described in the image itself
49 ; See the Lauterbach TRACE32 DATA.LOAD family of commands for other options to download raw binaries, Intel HEX images, etc.
50 ; (specifically see DATA.LOAD.BINARY and DATA.LOAD.INTELHEX)
51 ; DATA.LOAD "../Binaries/RciRangingApp.axf"
52 DATA.LOAD.INTELHEX "../Binaries/RciRangingApp.hex" /VERIFY
53 ; Leave flash programming mode
54 FLASH.PROGRAM.OFF
55
56 ;
57 ; Script is done
58 ;
59 ENDDO
    
```

```

FLASH.RESET // 清除闪存中的当前配置。
FLASH.TARGET 0x20004000 0x20008000 0x1000 "..\FlashDriver\NCJ29D5_Raw_FlashDriver.bin" /STACKSIZE 0x400
// 定义闪存驱动存放的位置。
    
```



```
FLASH.CREATE 1. 0x00238a00--0x0023A9FF 0x2000 TARGET LONG /INFO "RX Radio Config"  
// 创建一个区域存放接收端射频配置。  
FLASH.PROGRAM 0x00238A00--0x0023A9FF // 闪存开始编程。  
PRINT "* Pre-loading Rx Cfgs. NCJ29D5 supports 16 Rx indices" // 打印信息。  
DATA.LOAD.BINARY "../RadioConfigurations/Rx_Eco_Mini_Data_Phr_Key_Fob.bin" 0x00238A00..0x238BFF  
// 加载接收端配置信息 0, bin 文件要存放在如上相对路径下。  
DATA.LOAD.BINARY "../RadioConfigurations/Rx_Eco_Mini_Ranging_Data_Phr_Key_Fob.bin"0x00238C00..0x238DFF  
// 加载接收端配置信息 1, bin 文件要存放在如上相对路径下。  
DATA.LOAD.BINARY "../RadioConfigurations/Rx_Eco_Mini_Ranging_Key_Fob.bin" 0x00238E00..0x238FFF  
// 加载接收端配置信息 2, bin 文件要存放在如上相对路径下。  
FLASH.PROGRAM.OFF // 闪存编程结束。  
FLASH.CREATE 2. 0x0023AA00--0x0023B9FF 0x1000 TARGET LONG /INFO "TX Radio Configs"  
// 创建一个区域存放发射端射频配置。  
FLASH.PROGRAM 0x0023AA00--0x0023B9FF // 闪存开始编程。  
PRINT "* Pre-loading Tx Cfgs. NCJ29D5 supports 16 Tx indices" // 打印信息。  
DATA.LOAD.BINARY "../RadioConfigurations/Tx_Eco_Mini_Data_Phr_Key_Fob.bin" 0x0023AA00..0x23AAFF  
// 加载接收端配置信息 16, bin 文件要存放在如上相对路径下。  
DATA.LOAD.BINARY "../RadioConfigurations/Tx_Eco_Mini_Ranging_Data_Phr_Key_Fob.bin"  
0x0023AB00..0x23ABFF  
// 加载接收端配置信息 17, bin 文件要存放在如上相对路径下。  
DATA.LOAD.BINARY "../RadioConfigurations/Tx_Eco_Mini_Ranging_Key_Fob.bin" 0x0023AC00..0x23ACFF  
// 加载接收端配置信息 18, bin 文件要存放在如上相对路径下。  
DATA.LOAD.BINARY "../RadioConfigurations/Tx_DDFSCW.bin" 0x0023B000..0x23B0FF  
// 加载接收端配置信息 22, bin 文件要存放在如上相对路径下。  
FLASH.PROGRAM.OFF // 闪存编程结束。  
  
FLASH.CREATE 3. 0x00200000--0x0021FFFF 0x20000 TARGET LONG /INFO "Application Flash"  
// 创建一个区域存放应用程序。  
FLASH.PROGRAM 0x00200000--0x0021FFFF // 闪存开始编程。  
PRINT "* Pre-loading RCI Ranging Application" // 打印信息。  
DATA.LOAD.INTELHEX "../Binaries/RciRangingApp.hex" /VERIFY // 下载可执行程序并验证。  
FLASH.PROGRAM.OFF // 闪存编程结束。  
ENDDO // 脚本完成。
```



③ NCJ29D5_Flash_Package_With_SWUP.cmm 脚本内容: 烧写软件升级程序, 烧写 DSP 固件程序。

```

1 ; =====
2 ; Activation of SWUP
3 ; =====
4 PRINT "* Attaching to target"
5 SYSTEM.ATTACH
6 BREAK
7 PRINT "* Disabling Watchdog (WDOG_RST_ENBL = 0)"
8 PER.Set.simple D:0x4000A834 %Long 0x0
9
10 ; Clear the current flash configuration
11 FLASH.RESET
12
13 PRINT "* loading program into RAM"
14 DATA.LOAD.INTELHEX "..\Binaries\SwupActivation.hex" /Verify
15
16 PRINT "* set program counter to RAM"
17 REGISTER.SET PC 0x20004000
18
19 PRINT "* executing RAM program"
20 WAIT 10ms
21 GO
22
23 PRINT "* Attaching to target"
24 SYSTEM.ATTACH
25 BREAK
26
27 PRINT "* soft reset to load SWUP"
28 ON ERROR CONTINUE
29 DATA.SET EN:0xE000ED0C %Long 0x05FA0004
30
31 ON ERROR INHERIT
32 ERROR.RESET
    
```

PRINT "* Attaching to target"	// 打印信息。
SYSTEM.ATTACH	// 连接调试器和核心。
BREAK	// 系统停止。
PRINT "* Disabling Watchdog (WDOG_RST_ENBL = 0)"	// 打印信息。
PER.Set.simple D:0x4000A834 %Long 0x0	// 关闭看门狗。
FLASH.RESET	// 闪存重置, 以清除当前配置。
PRINT "* loading program into RAM"	// 打印信息。
DATA.LOAD.INTELHEX "..\Binaries\SwupActivation.hex" /Verify	// 加载启动软件升级的程序并验证。
PRINT "* set program counter to RAM"	// 打印信息。
REGISTER.SET PC 0x20004000	// 设置 PC 寄存器位置。
PRINT "* executing RAM program"	// 打印信息。
WAIT 10ms	// 等待 10ms。
GO	// 程序开始执行。
PRINT "* Attaching to target"	// 打印信息。
SYSTEM.ATTACH	// 调试器连接核心。
BREAK	// 程序停止。



```
PRINT "*" soft reset to load SWUP" // 打印信息。
ON ERROR CONTINUE // 人为报告一个错误新信息并继续。
DATA.SET EN:0xE000ED0C %Long 0x05FA0004 // 一个预期报错的位置。
ON ERROR INHERIT // 继承上个报错。
ERROR.RESET // 报错后进行一个软重置。
```

```

NCJ29D5_Flash_Package_With_SWUP.cmn
37 PRINT "*" Attaching to target"
38 SYSTEM.ATTACH
39 BREAK
40 PRINT "*" Disabling Watchdog (WDOG_RST_ENBL = 0)"
41 PER.Set.simple D:0x4000A834 %Long 0x0
42
43 ; Clear the current flash configuration
44 FLASH.RESET
45
46 ; Define target flash algorithm
47 PRINT "*" loading driver to communicate with SWUP"
48 FLASH.TARGET 0x20006000 0x20008000 0x1000 "..\FlashDriver\NCJ29D5_SWUP_FlashDriver.bin" /STACKSIZE 0x800
49
50 ; Define flash region #1 in address range 0x00500000..0x0051ffff (flash via target flash algorithm)
51 ; address will be handled via flash driver. This address ensure that no data is overwritten with a wrong loaded flash driver
52
53 ; Flash.Create is needed for lauterbach to load the flash driver
54 FLASH.CREATE 1. 0x00500000--0x0051ffff 0x20000 TARGET LONG /INFO "invalid flash area"
55
56 ; flash.program all is needed for lauterbach to trigger the write command
57 FLASH.PROGRAM ALL
58
59 PRINT "*" reading package"
60 DATA.LOAD.BINARY "..\Binaries\NCJ29D5D_DSP_Firmware.pkg" 0x00500000
61
62 ; Leave flash programming mode
63 FLASH.PROGRAM.OFF
64
65 ; after a successful update, SWUP will trigger a restart. This point should not be reached.
66 PRINT "*" Update Package failed!"
67 ;
68 ; Script is done
69 ;
70 ENDDO

```

```
PRINT "*" Attaching to target" // 打印信息。
SYSTEM.ATTACH // 连接调试器和核心。
BREAK // 程序停止。
PRINT "*" Disabling Watchdog (WDOG_RST_ENBL = 0)" // 打印信息。
PER.Set.simple D:0x4000A834 %Long 0x0 // 关闭看门狗。
FLASH.RESET // 闪存重置，以清除当前配置。
PRINT "*" loading driver to communicate with SWUP" // 打印信息。
FLASH.TARGET 0x20006000 0x20008000 0x1000 "..\FlashDriver\NCJ29D5_SWUP_FlashDriver.bin" /STACKSIZE
0x800 // 定义闪存驱动存放的位置。
FLASH.CREATE 1. 0x00500000--0x0051ffff 0x20000 TARGET LONG /INFO "invalid flash area"
// 在闪存上创建一个 127k 无错误区域。
; flash.program all is needed for lauterbach to trigger the write command
FLASH.PROGRAM ALL // 触发执行闪存的写命令。
PRINT "*" reading package" // 打印信息。
```



```
DATA.LOAD.BINARY "..\Binaries\NCJ29D5D_DSP_Firmware.pkg" 0x00500000
// 加载 DSP 固件在指定闪存区域上。

FLASH.PROGRAM.OFF
// 闪存编程结束。

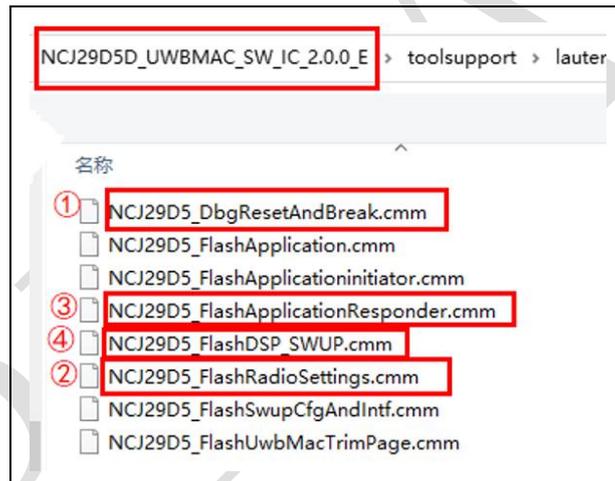
PRINT "* Update Package failed!"
// 以上程序更新后，会触发 Reset，而不会打印此处。

ENDDO
// 脚本结束。
```

P.S. 有几个经验可以分享:

- a. 把需要烧写的 sdk 包拷贝到无中文且较短路径下成功率更高。
- b. 如果某次下载失败后，再进行下载的话，可能因为环境变量已经发生了变化，无法找到脚本中原本可以找到的文件，所以，建议一旦下载失败一次，就完全退出软件，重新进入再进行下一次下载。

(3) 新的芯片建议先进行如上烧写，说明芯片可以使用，后续再根据需求烧写对应固件版本，比如我这里后续需要使用“NCJ29D5D_UWBMAC_SW_IC_2.0.0_E”版本的固件，如下是下载的版本和下载的顺序，此处第③步建议下载 ***Responder.cmm，如果之前下载 ***Initiator.cmm 就会进入低功耗模式，如果想再次下载固件，就会因为芯片上电进入低功耗模式无法完成下载，如果遇到 DSP 固件无法下载的情况可以参考我的另一篇博文《NCJ29D5D 下载 package 失败的解决方法》。





三、总结

本文介绍了基于 BLE 和 UWB 的 Foxes 开发板的 UWB 芯片 NCJ29D5D 的参数介绍，调试工具的使用，调试软件的安装，固件的下载方法，并且详细分析了下载脚本的内容和脚本具体进行的操作。

欢迎在博文下方留言评论，我们会及时回复您的问题。如有更多需求，欢迎联系大联大世平集团 ATU 部门：atu.cn@wpi-group.com。

WPI-ATU



参考资料

- [1] 《an523019 - AN-SCA-1901 NCJ29D5 Tools (1.9)》
- [2] 《ds652611_1.1_NCJ29D5DHN – Ranger-4 Ultra Wideband Transceiver with MAC support》 (datasheet)

下一篇：

《NCJ29D5D 下载 package 失败的解决方法》

我是好景虚设，欢迎在大大通上关注我！

好景虚设
青出於藍

★ 2 ★ 8 ★ 56

編輯

4184 經驗

粉絲 43 | 追蹤 108 | 提問 197 | 回答 93

有興趣/擅長的領域或技術 **Rockchip · Wi-Fi · BLE · 机器人 · 广告屏**

自我介紹
12 顆 IPU 在燃燒!!!

論壇成績
頭銜 **青出於藍** ?
積分 **280** ?
勳章 ?

2 8 56

UWB 相关文档：

《NCJ29D5D 下载 package 失败的解决方法》