Product Manual

i.MX-RT1050 使用手册

SylixOS 平台使用手册

PM0010020004 V1.10 Date: 2018/04/20

产品使用手册

类别	内容
关键词	SylixOS RealEvo-IDE i.MX-RT1050
摘 要	i.MX-RT1050 平台开发快速入门

翼辉信息 ACOINFO

修订历史

版本	日期	原因
V1.0	2018/03/07	创建文档
V1.1	2018/04/20	增加一键部署及仿真调试介绍
V1.2	2018/05/03	修改 2.3 章，匹配新版工具软件

目 录

第1章 概述

本文档介绍了如何在 i.MX-RT1050 平台启动 SylixOS 实时操作系统，并且快速开发基于 SylixOS 的应用程序。

1.1 SylixOS 介绍

1.1.1 SylixOS 简介

SylixOS 是一款大型嵌入式实时操作系统，诞生于 2006 年，起初它只是一个小型多任务调度器，经过多年开发，SylixOS 目前已经成为一个功能完善、性能卓越、可靠稳定的嵌入式系统软件开发平台。

与 SylixOS 类似的实时操作系统中，全球比较知名的有 VxWorks（主要应用于航空航天、军事与工业自动化领域）、RTEMS（起源于美国国防部导弹与火箭控制实时系统）等。

SylixOS 作为实时操作系统的后来者，在设计思路上借鉴了众多实时操作系统的设计思想，使得 SylixOS 在功能和具体性能上达到或超过了众多实时操作系统的水平，成为国内实时操作系统的最优秀代表之一。

1.1.2 SylixOS 的功能与特点

SylixOS 作为抢占式多任务硬实时操作系统，具有如下功能与特点：

- 兼容 IEEE1003（ISO/IEC9945）操作系统接口规范；
- 兼容 POSIX 1003.1b（ISO/IEC 9945-1）实时编程的标准；
- 优秀的实时性能（任务调度与切换、中断响应算法都是 O(1)时间复杂度算法）；
- 支持无限多任务；
- 抢占式调度支持 256 个优先级；
- 支持协程（windows 称为纤程）；
- 支持虚拟进程；
- 支持优先级继承，防止优先级反转；
- 极其稳定的内核，很多基于 SylixOS 开发的产品都需要 7×24 小时不间断运行；
- 内核占用 CPU 率低；
- 柔性体系（Scalable）；
- 核心代码使用 C 语言编写，可移植性好；
- 支持紧耦合同构多处理器（SMP），例如：ARM Cortex-A9 SMP Core；
- 独一无二的硬实时多核调度算法；
- 支持标准 I/O、多路 I/O 复用与异步 I/O 接口；
- 支持多种新兴异步事件同步化接口，例如：signalfd、timerfd、eventfd 等；
- 支持众多标准文件系统：TPSFS、FAT、YAFFS、RAMFS、NFS、ROMFS 等；
- 支持文件记录锁，可支持数据库；
- 支持统一的块设备 Cache 模型；
- 支持内存管理单元（MMU）；
- 支持第三方 GUI 图形库，如：Qt、Microwindows、emWin 等；
- 支持动态装载应用程序、动态链接库以及模块；
- 支持扩展系统符号接口；

- 支持标准 TCP/IPv4/IPv6 双网络协议栈，提供标准的 socket 操作接口；
- 支持 AF_UNIX, AF_PACKET, AF_INET, AF_INET6 协议域；
- 内部集成众多网络工具，例如：FTP、TFTP、NAT、PING、TELNET、NFS 等；
- 内部集成 shell 接口、支持环境变量（与 Linux 操作习惯基本兼容）；
- 内部集成可重入 ISO/ANSI C 库（支持 80%以上标准函数）；
- 支持众多标准设备抽象，如：TTY、BLOCK、DMA、ATA、GRAPH、RTC、PIPE 等。同时支持多种工业设备或总线模型，如：PCI、USB、CAN、I2C、SPI、SDIO 等；
- 提供高速定时器设备接口，可提供高于主时钟频率的定时服务；
- 支持热插拔设备；
- 支持设备功耗管理；
- 内核、驱动、应用程序支持 GDB 调试；
- 提供内核行为跟踪器，方便进行应用性能与故障分析。

1.1.3 SylixOS 的应用领域

SylixOS 采用抢占式、多任务、硬实时的方式来设计整个操作系统。其技术实现的核心目标是实时可控，稳定可靠。所以 SylixOS 适用于（但不限于）以下对实时性和稳定性要求尤为突出的领域：

- **工业实时控制领域**：主要包括工业机器人系统、现场安全监控与防护系统、工业现场总线通信管理系统等；
- **航空航天领域**：主要包括航空器飞控系统、航空航天数据采集与记录系统、高精度测绘系统，航空航天通信系统等；
- **国防安全领域**：主要包括加密通信系统、传感器终端系统、虚拟仪表系统、数据采集与记录系统、火控系统等；
- **金融终端领域**：主要包括 POS 收费系统、终端支付系统、ATM 自动柜员机等；
- **可靠民用领域**：主要包括汽车行驶记录仪系统、车辆及船用发动机中央控制系统、生产线测试系统、医疗仪器系统、分布式无人值守系统等。

1.2 i.MX-RT1050 介绍

1.2.1 i.MX-RT1050 芯片介绍

i.MX-RT1050 是 NXP 推出的业界首款跨界处理器，兼具应用处理器的高性能与高度集成，以及微控制器的易用性和实时性。高达 600 MHz 的 ARM Cortex-M7 内核，可提供较高的 CPU 性能与极佳的实时性。

- 高性能 Arm® Cortex-M7®内核；
- 3020 CoreMark/1284 DMIPS @ 600 MHz；
- 高达 512 kB 紧耦合存储器(TCM)；
- 实时低延迟相应，低至 20 ns；
- 行业最低的动态功耗，带集成 DC-DC 转换器；
- 低功耗运行模式下运行频率为 24MHz；
- 面向 GUI 和增强 HMI 的高级多媒体；
- 2D 图形加速引擎；
- 并行摄像头传感器接口；
- LCD 显示屏控制器(高达 WXGA 1366x768)；
- 3x I2S，面向高性能多通道音频；

- 丰富的外部存储接口选项：NAND、eMMC、QuadSPI NOR Flash 和 Parallel NOR Flash；
- 无线连接接口：Wi-Fi®、Bluetooth®、BLE、ZigBee®和 Thread™。

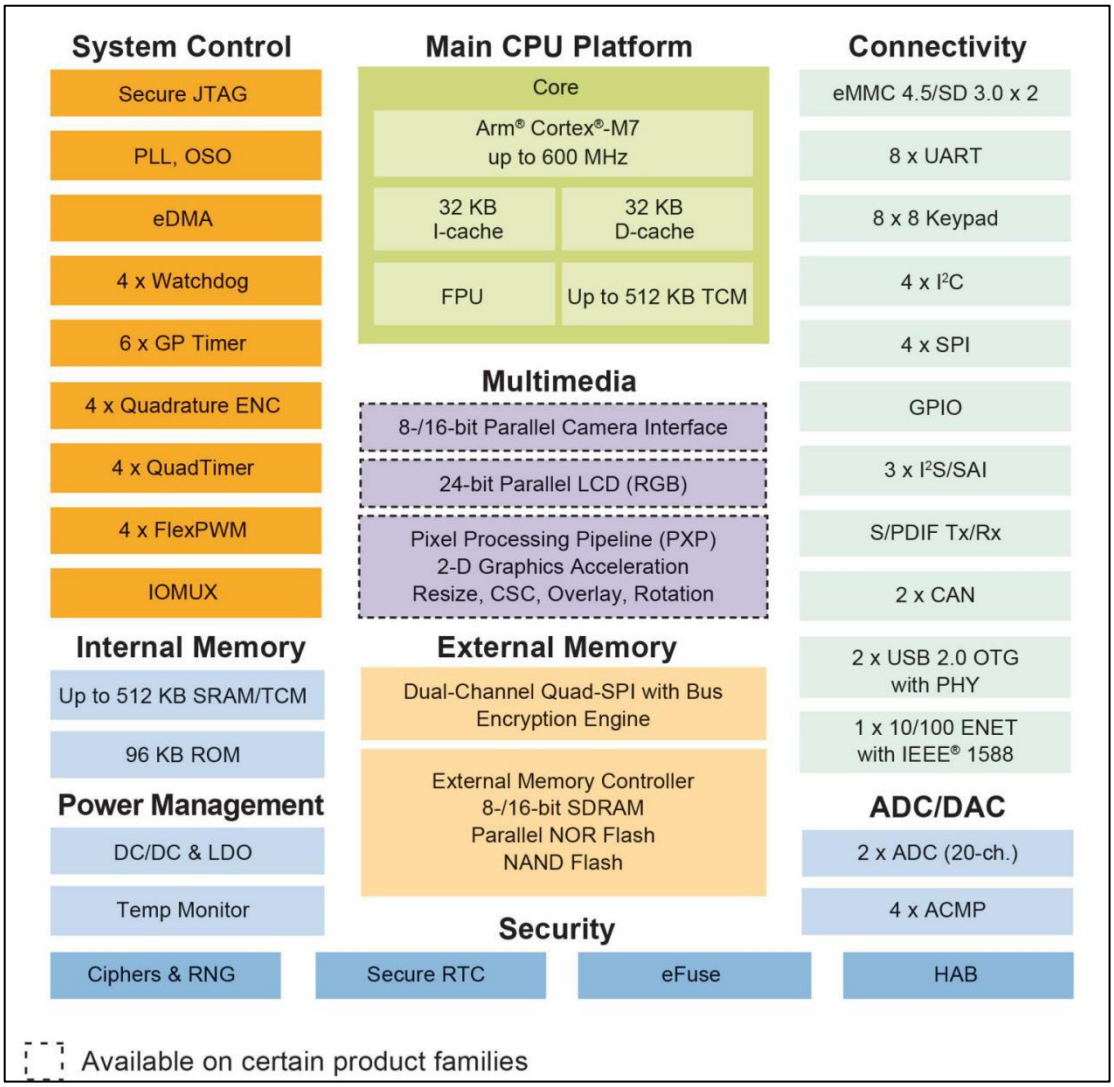

图 1.1 i.MX-RT1050 资源框图

1.2.2 i.MX-RT1050-EVK 开发板介绍

i.MX-RT1050-EVK 是 NXP 为 i.MX RT1050 处理器配套的首套官方开发板。为 4 层穿孔式 PCB，可由 USB 供电，板载 OpenSDA 调试器，一根 USB 线即可完成供电、仿真、串口输入输出的功能。

- PIMXRT1052DVL6A 处理器；
- 256 Mb/32MB SDRAM 存储器；
- 512 Mb/64MB Hyper Flash；
- 64 Mb/8MB QSPI Flash；
- SD 卡 TF 插槽；
- 并行 LCD 接头；
- 摄像头接头；

- WM8960 音频编解码器；
- 4 极音频耳机插孔；
- 板载麦克风；
- SPDIF 接头；
- Micro USBz 主设备和 OTG 接头；
- 以太网(10/100T)连接器；
- CAN 收发器；
- Arduino®接口。

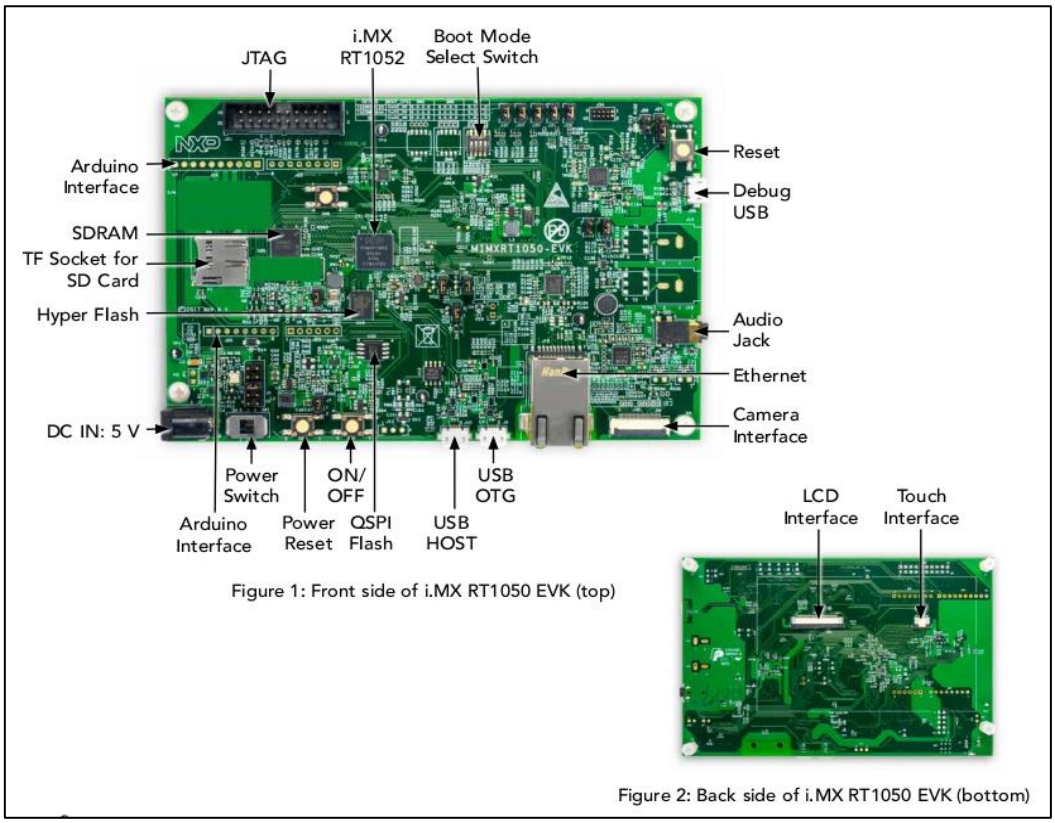

图 1.2　i.MX-RT1050-EVK 指示图

第2章 快速体验 SylixOS

目前 SylixOS 已经支持了 NXP 的 i.MX-RT1050 处理器，并且在 RealEvo-IDE 3.7.3 中已经集成了 i.MX-RT1050-EVK 开发板的 BSP，本章将带领大家快速体验 SylixOS 在 i.MX-RT1050-EVK 开发板上运行的步骤和效果。

首先要获取和安装翼辉信息为 SylixOS 开发推出的 RealEvo-IDE。虽然 SylixOS 是免费开源的，但其集成开发环境（IDE）是需要付费的，想要了解 SylixOS 的用户可以免费申请体验版 IDE。IDE 的支持会使得 SylixOS 开发变得轻松而愉悦，能帮助用户把时间精力投入到更有意义的工程开发中去。

在浏览器上打开免费申请页面 http://www.acoinfo.com/html/experience.php 在这里填写表格，申请提交后，翼辉信息工作人员会主动联系您，向您发放 IDE 下载链接及注册码。

然后参考文档《RealEvo 软件注册步骤》安装 RealEvo-IDE 3.7.3 或者以上版本，安装完成后启动 IDE。

2.1 创建工程

启动 RealEvo-IDE 后需要用户创建或选择工作空间。在非中文路径下创建 imxrt1050 目录，并创建新的工作空间到该目录。如下图所示：

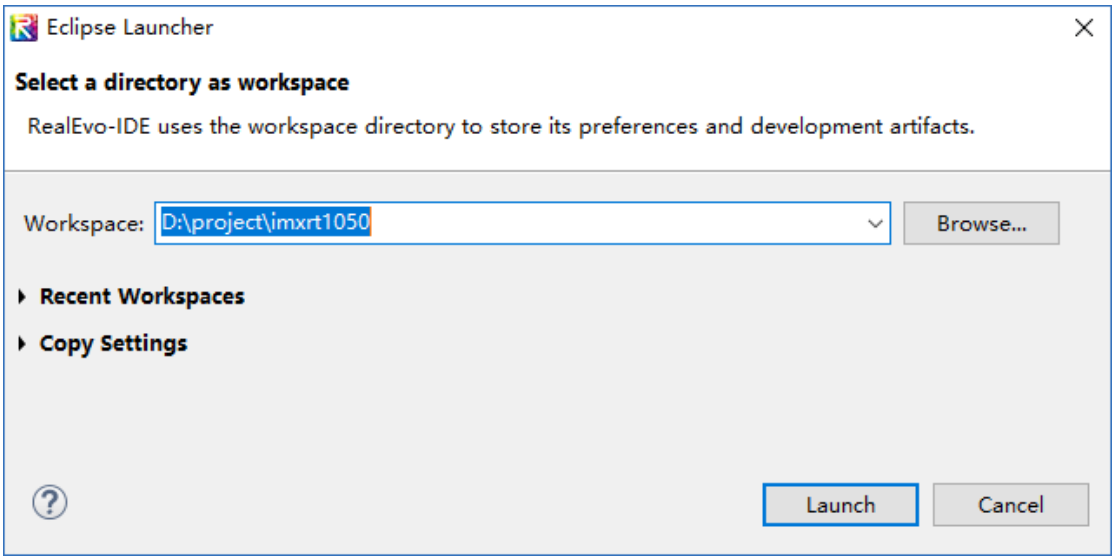

图 2.1 workspace 创建向导

2.1.1 创建 Base 工程

首先创建 SylixOS Lite Base 工程，选择“File→New→SylixOS Base”菜单，打开 SylixOS Base 工程创建向导，如下图所示：

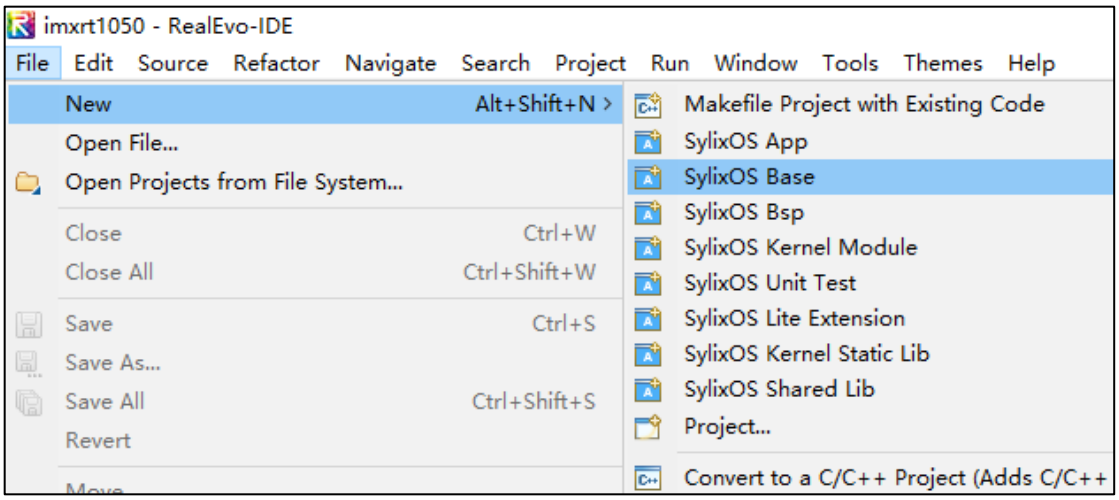

图 2.2　新建 Base 工程选择路径

填写工程名，这里命名为“base” ，勾选“Use default location”选项，继续下一步。

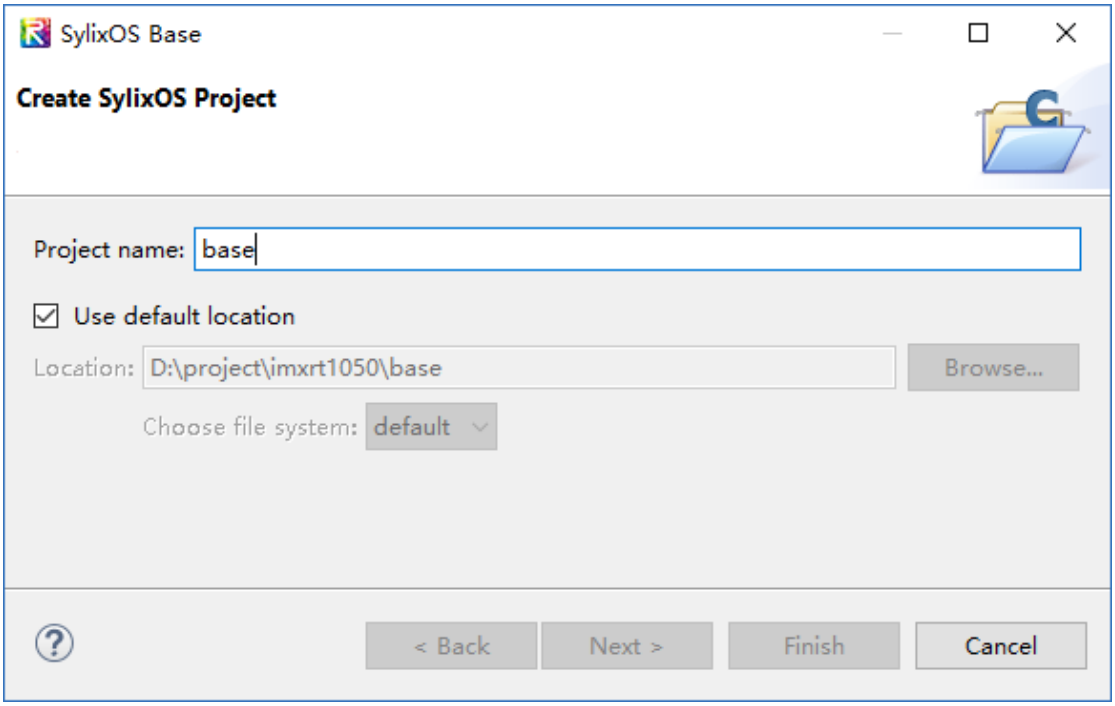

图 2.3　Base 工程创建向导

因为 i.MX-RT1050 不具备 MMU，此处选择“SylixOS Lite Base”，继续下一步。

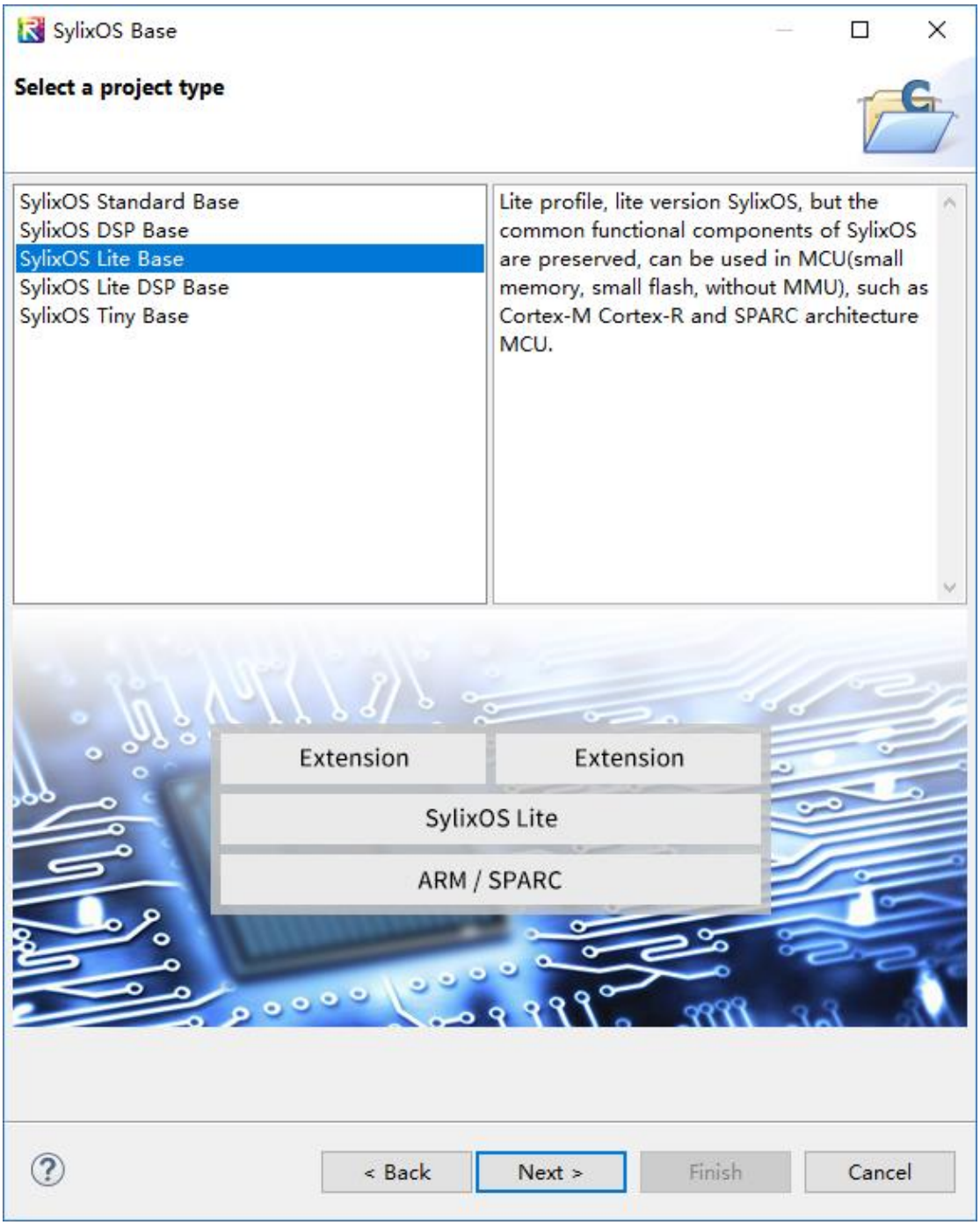

图 2.4 Base 类型选择页面

Toolchain 选择 arm-sylixoslitele-toolchain，Debug Level 选择 debug 模式，CPU Type 选择 cortex-m7，FPU Type 选择 fpv5-sp-d16，然后继续下一步。

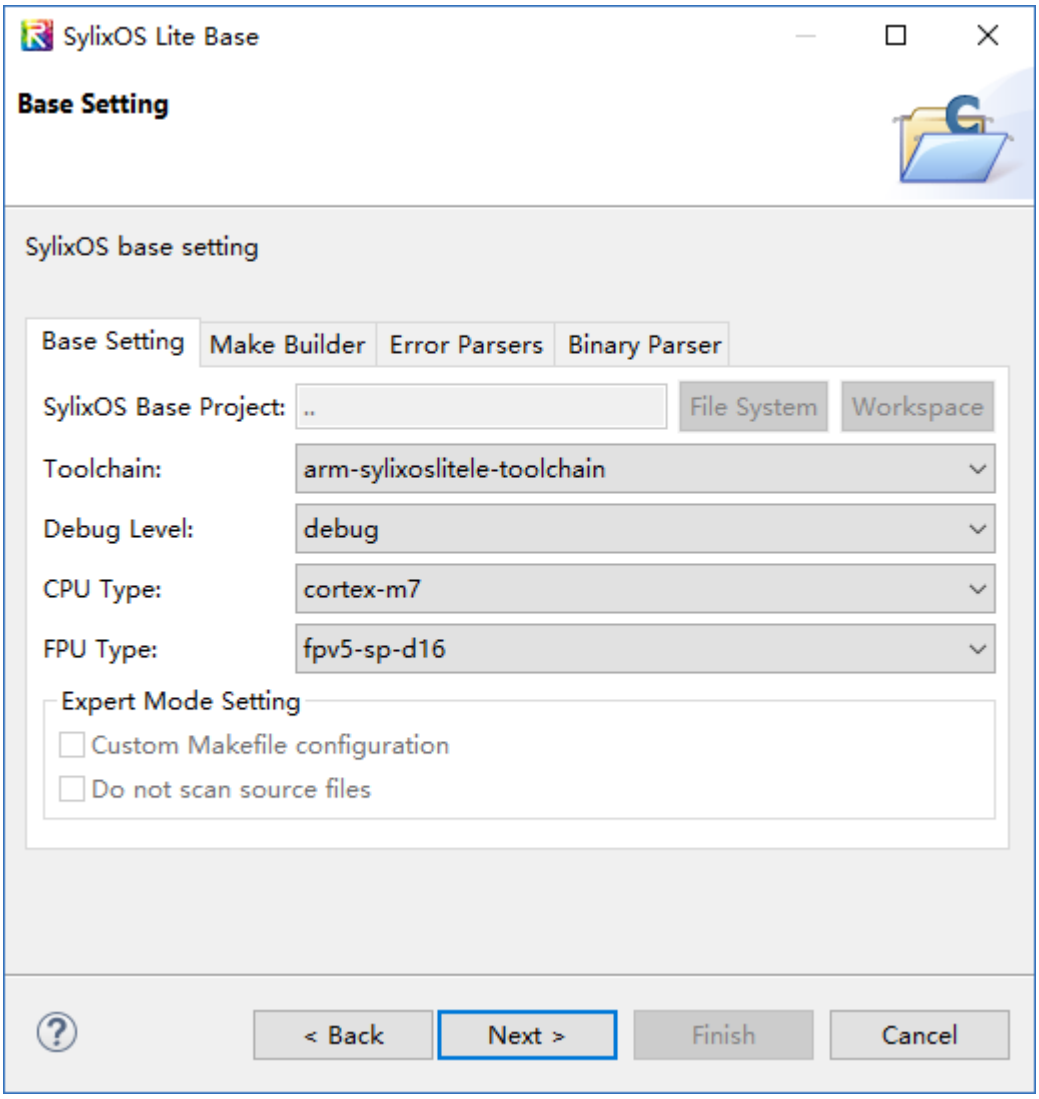

图 2.5　Base 工程创建向导

Lite 版系统组件只有 libsylixos ，勾选 Select All，最后点击 Finish 完成 Base 工程的创建。

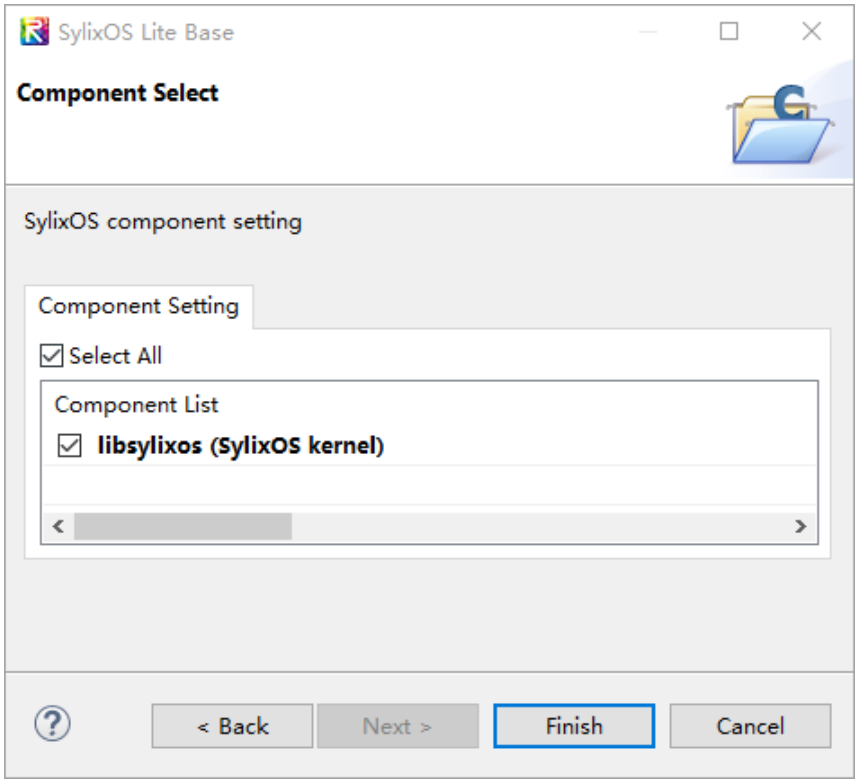

图 2.6　Base 工程创建向导

2.1.2 创建 BSP 工程

创建 BSP 工程，选择“File→New→SylixOS Bsp”菜单，打开 SylixOS Bsp 工程创建向导。填写工程名，这里命名为“bsp”，勾选 Use default location 选项，继续下一步。

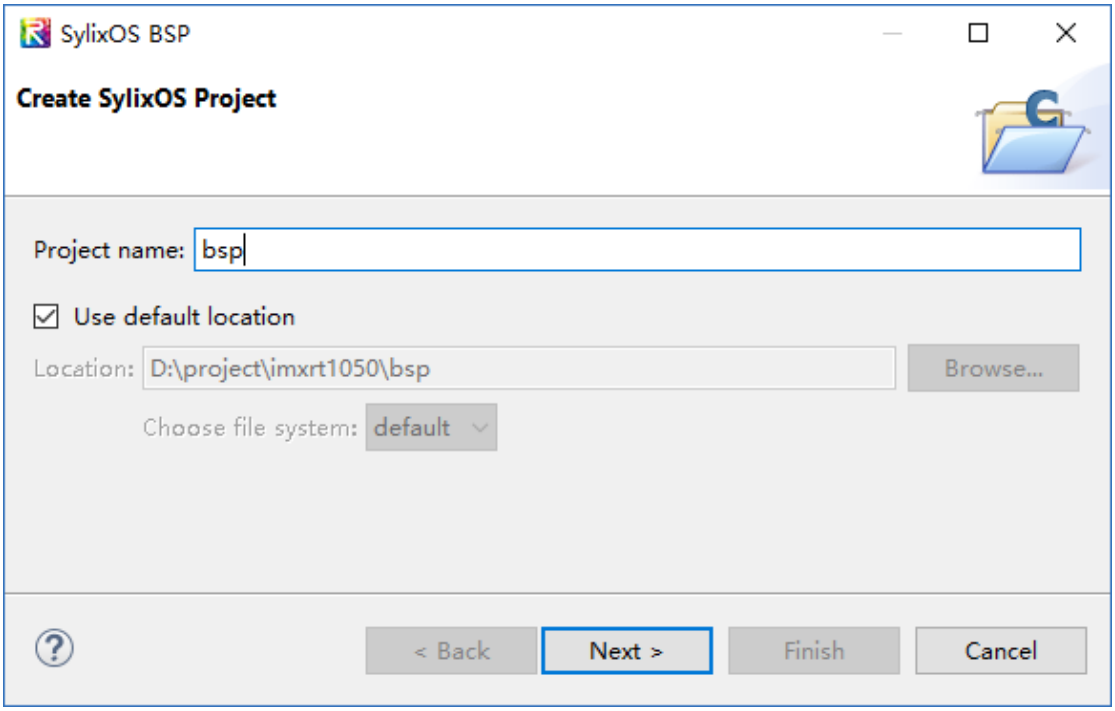

图 2.7　BSP 工程创建向导

SylixOS Base Project 选择该 BSP 工程所依赖的 Base 工程，点击“Workspace”按钮，选择 Base 工程，点击 OK，其他选项就会自动与 Base 工程保持一致。

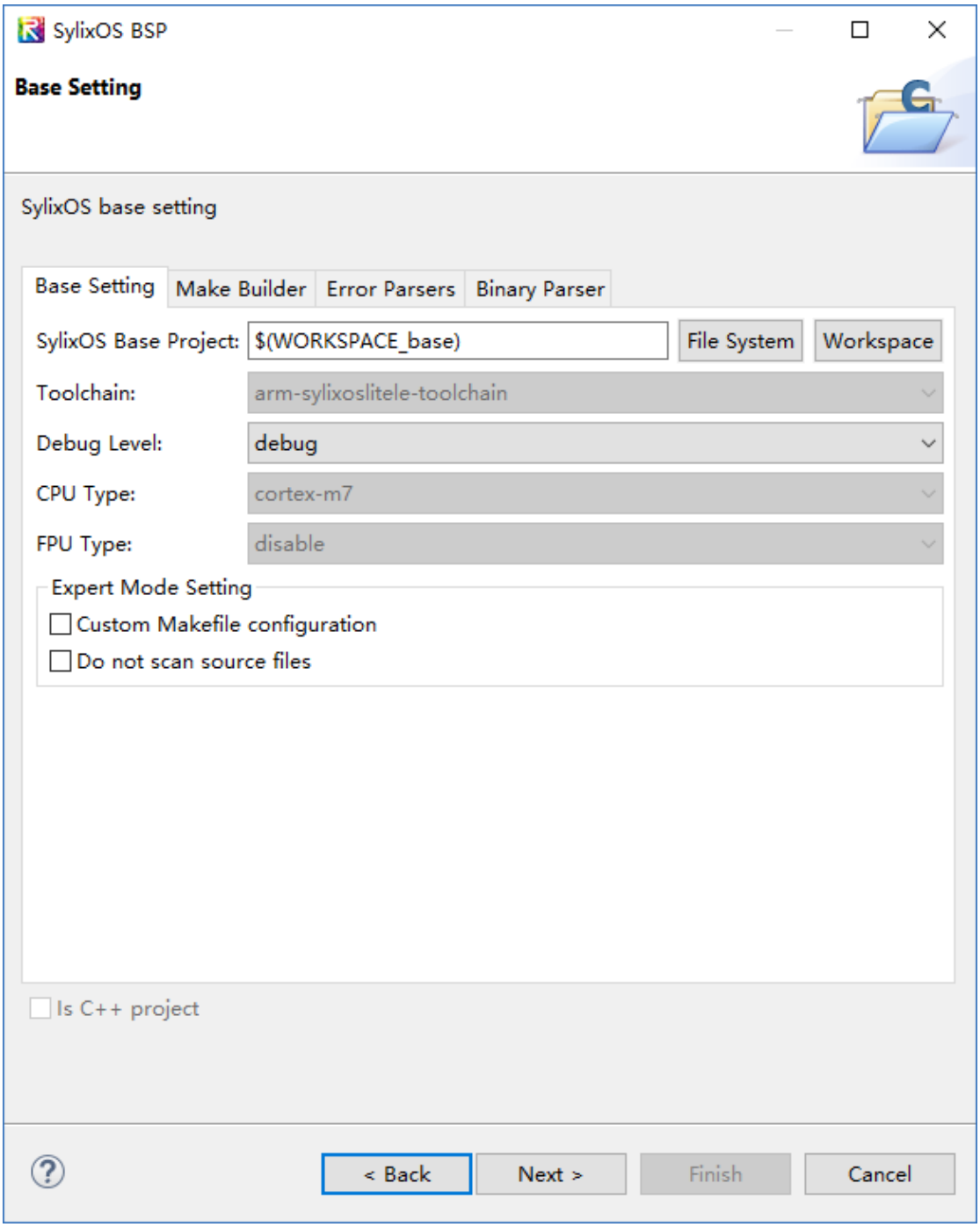

图 2.8　BSP 工程创建向导

继续下一步，进行 BSP 设置。在 3.7.2 版的 IDE 中已经集成了 i.MX-RT1050 的 BSP，所以可以直接选取该模板。 Bsp Template 选择 arm-imxrt1050，Use Extension 选择 yes，Boot Type 选择 hyperflash，最后点击 Finish 完成 BSP 工程的创建设置。

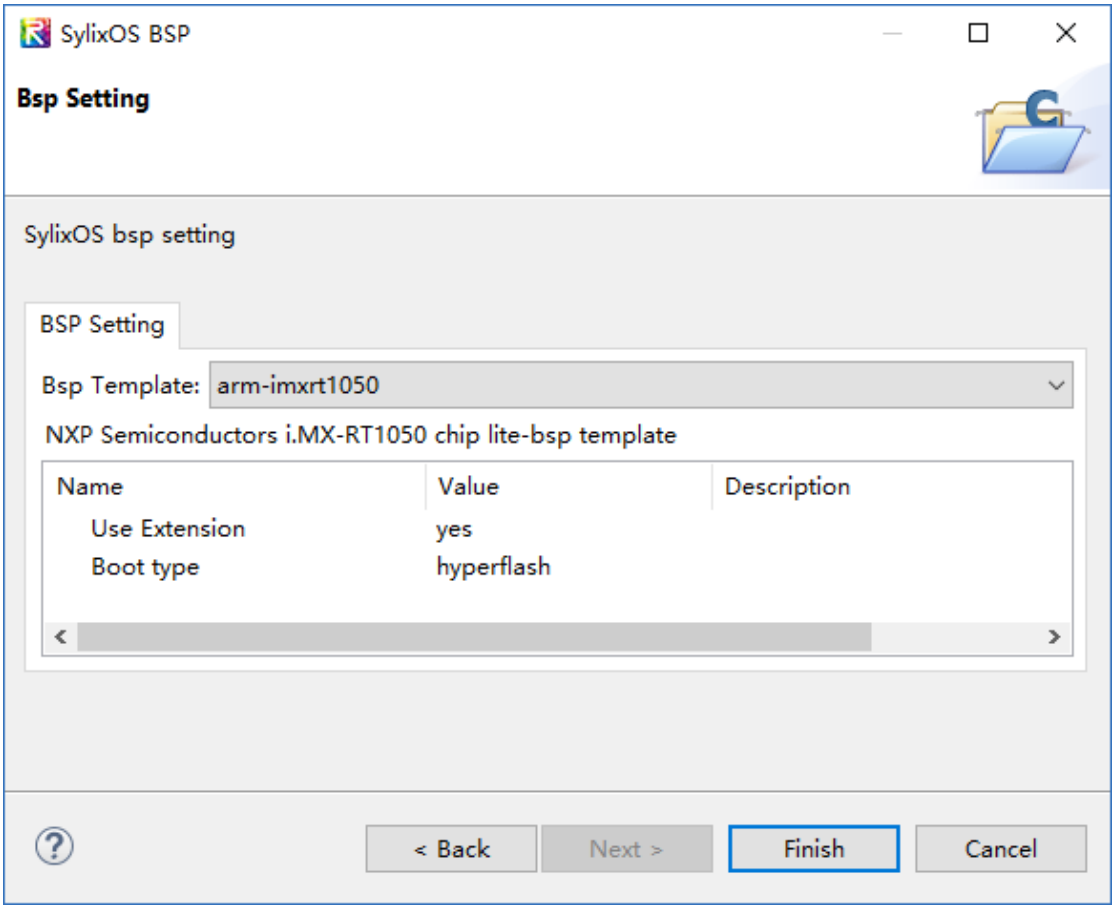

图 2.9　BSP 工程创建向导

2.1.3 创建 Extension 工程

创建 Extension 工程，选择“File→New→SylixOS Lite Extension ”菜单，打开 SylixOS Lite Extension 工程创建向导。填写工程名，这里命名为 ext ，勾选 Use default location 选项，继续下一步。类似 BSP 工程，也需要选择对应的 Base 工程。

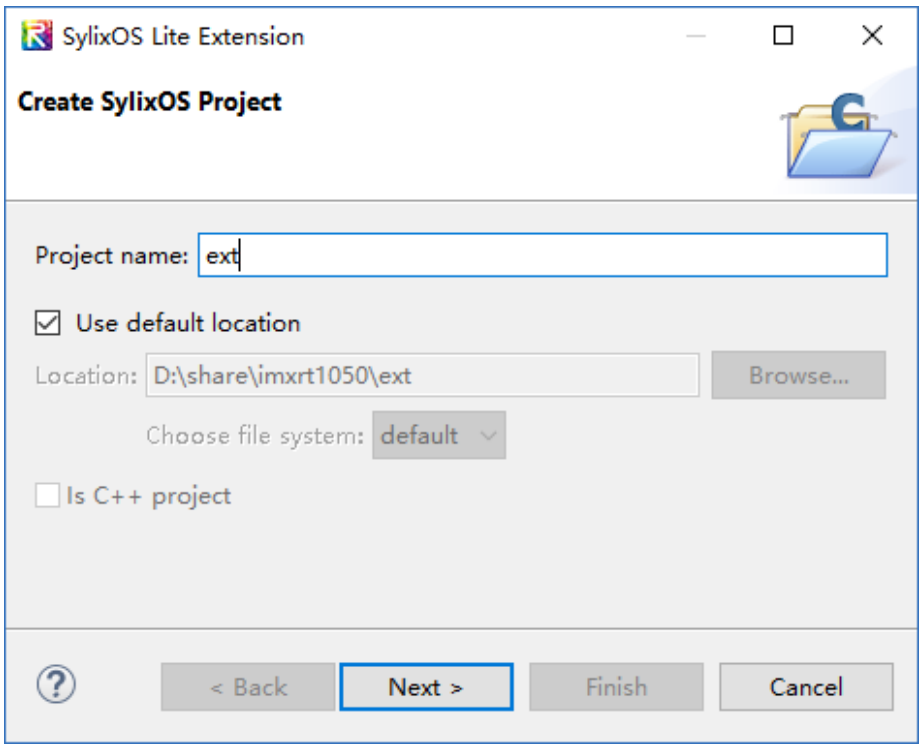

图 2.10　Extension 工程创建向导

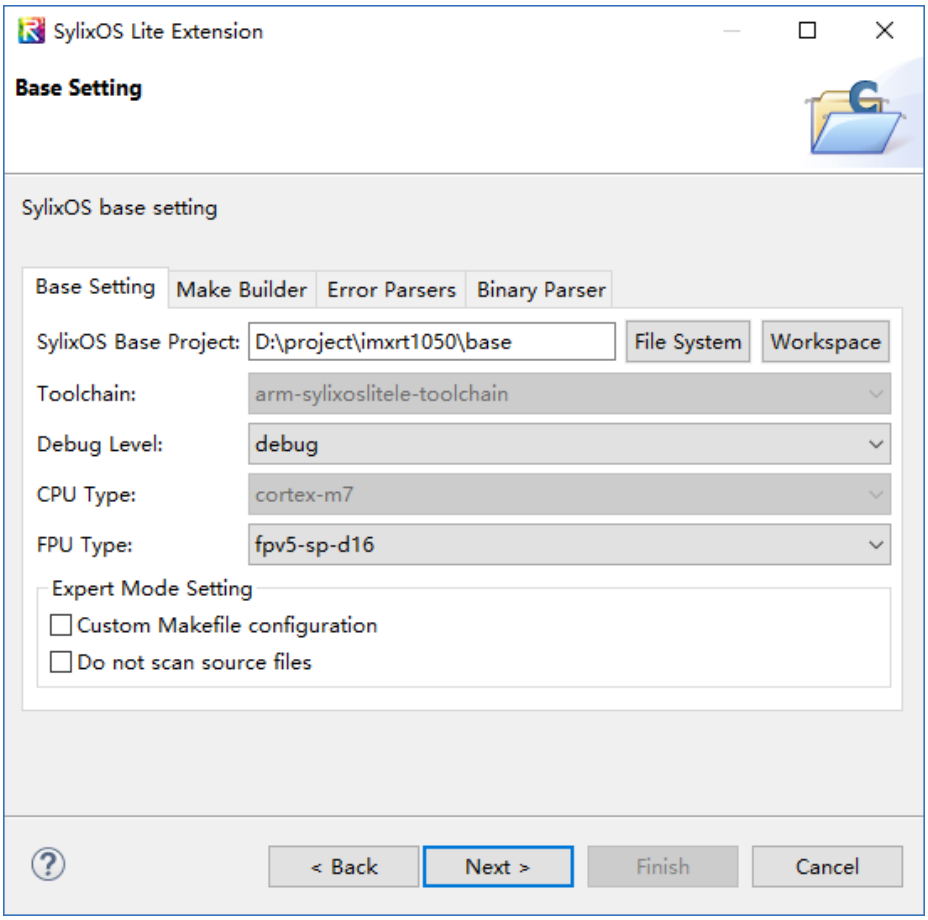

图 2.11　Extension 工程创建向导

继续下一步，选择关联的 BSP 工程。SylixOS BSP Path 选择本工作空间的 BSP 工程，Extension Template 选择默认模板，模板内的各参数按图 2.12 中的值修改，最后点击“Finish”按钮完成 Extension 工程的创建。

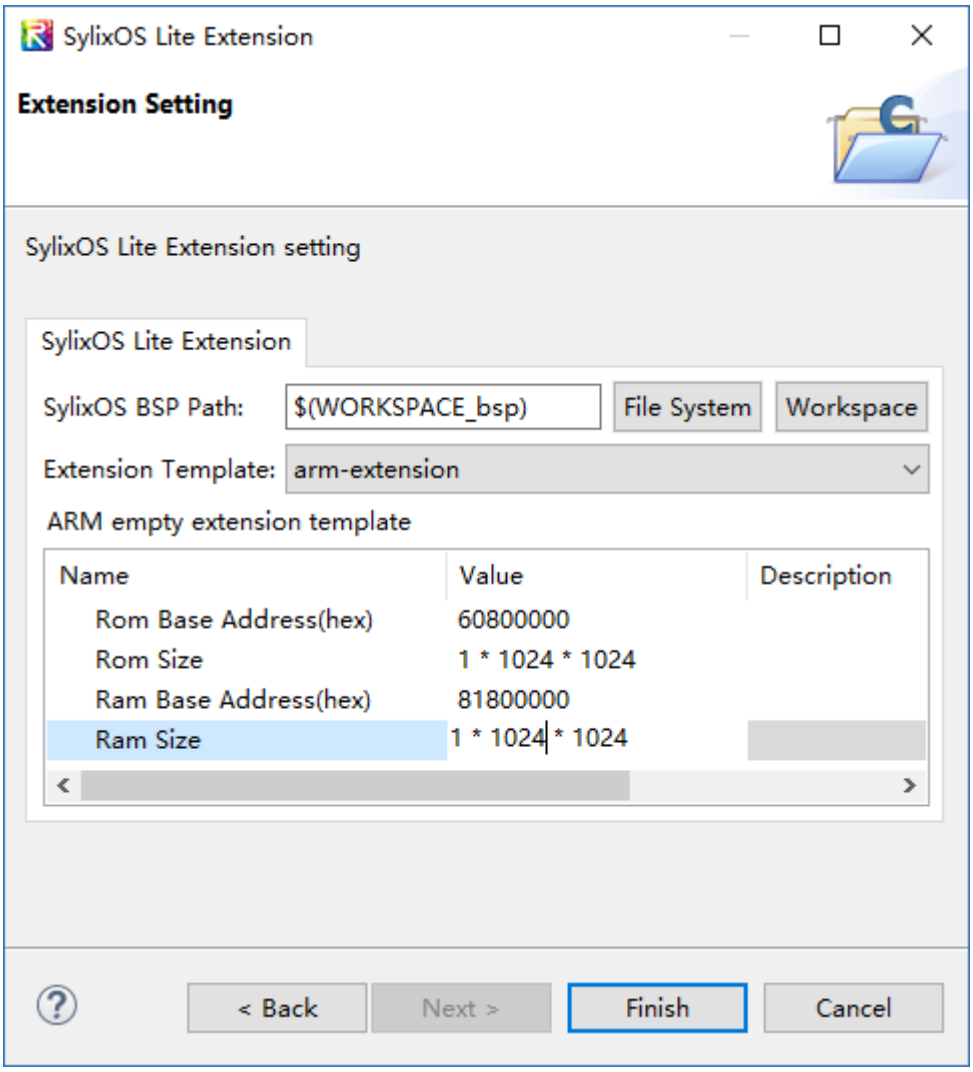

图 2.12　Extension 工程创建向导

经过以上操作，工程空间生成了 base、bsp、ext 三个工程。

2.2　编译工程

右键选中 base 目录，选择“Build Project”菜单，编译 Base 工程。编译过程中 console 窗口会有编译链接的信息输出。

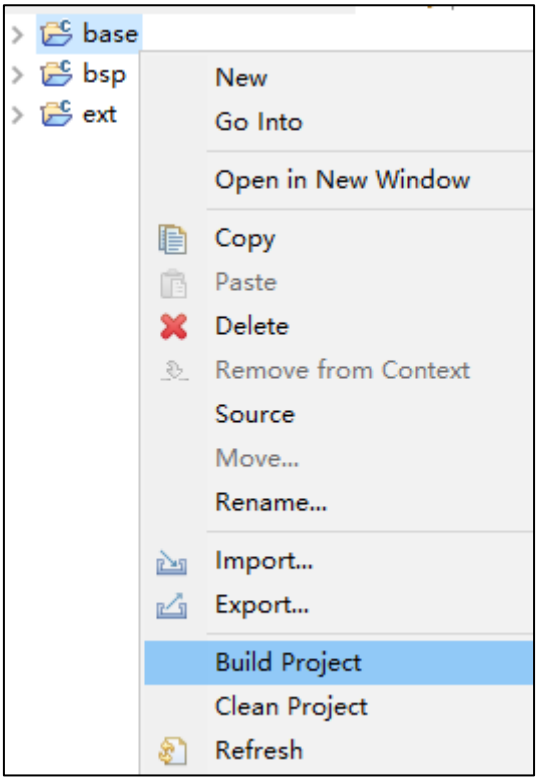

图 2.13　编译工程选择路径

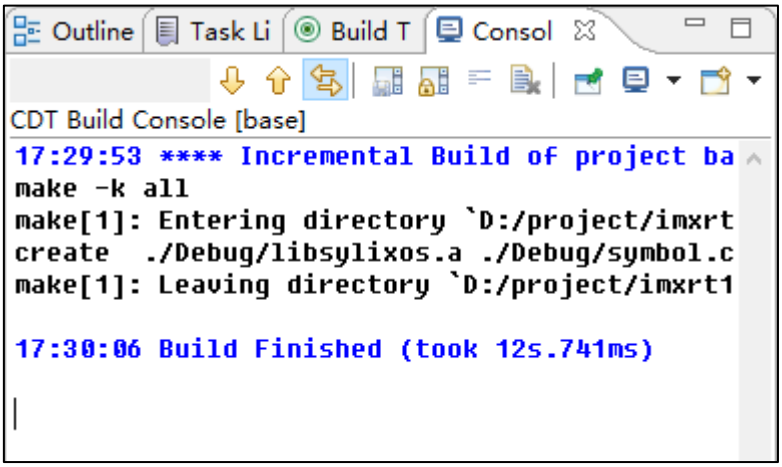

图 2.14　编译输出信息

同样方法编译 bsp 和 ext 工程，在 Debug 目录下会生成镜像文件，分别得到 bsp.bin 及 ext.bin 两个程序镜像，后面步骤会用到这两个镜像。

注意：编译顺序一定是先编译 Base 工程，再编译 bsp 工程，最后再编译 ext 工程。

2.3 制作 SylixOS 启动盘

选择 RealEvo-IDE 的“Tools→RealEvo-SylixOS-Installer”启动安装工具。

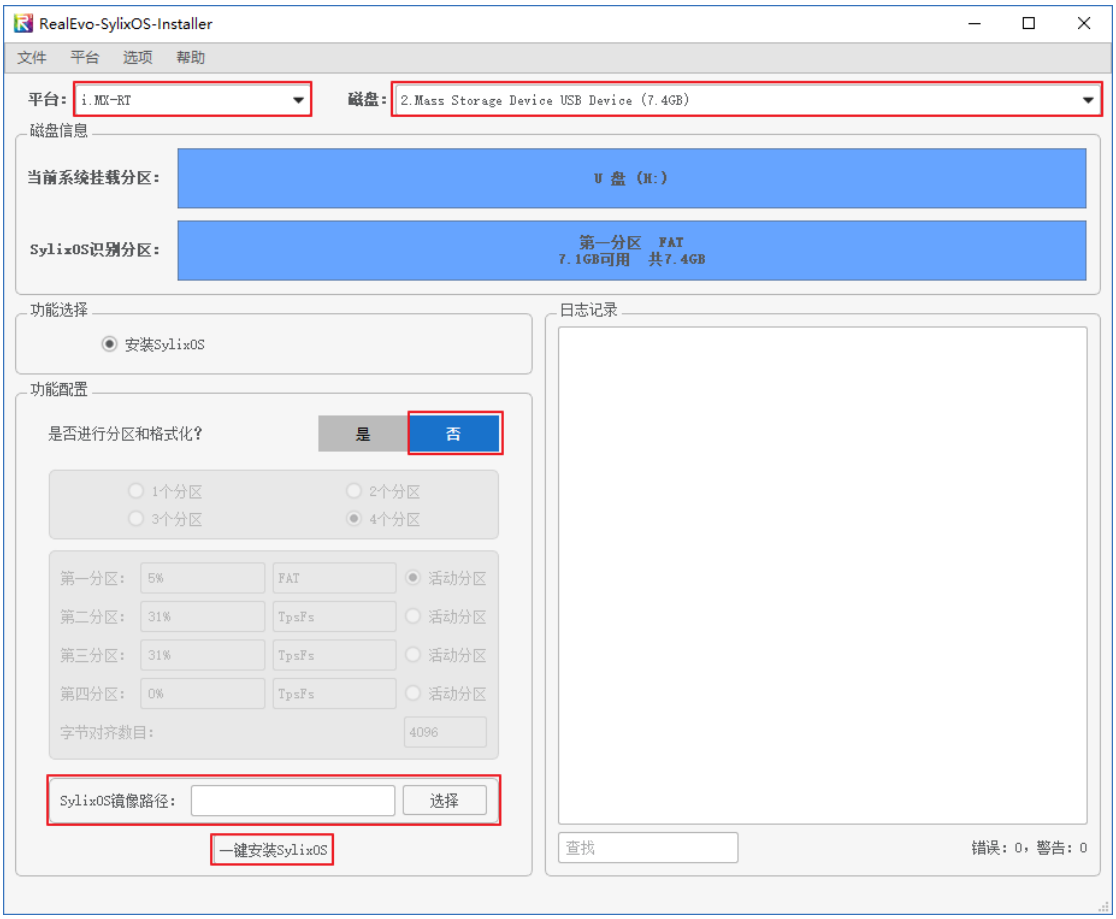

图 2.15 RealEvo-SylixOS-Installer 启动界面

平台选择“i.MX-RT”，磁盘选择对应的 SD 卡设备，如果 SD 卡还未格式化则应该选择进行分区和格式化，第一分区必须为 FAT 格式。SylixOS 镜像路径可以选择 BSP 工程生成的镜像文件 bsp.bin，也可以先空着。点击“一键安装 SylixOS”，会弹出警告对话框，确认选择的磁盘正确，点击“继续”按钮。

图 2.16 设备格式化警告

这样一张 SylixOS 启动卡就制作成功了。磁盘第一分区的根目录会多出刚才选择的那个 SylixOS 系统镜像文件 bsp.bin，后期用户也可以直接通过拷贝的方式在第一分区添加或更新新的系统镜像。这里还需要拷贝 ext 工程生成的 ext.bin 文件到启动卡的第一分区。

2.4 启动 BootLoader

i.MX-RT1050-EVK 开发板默认为 Internal Boot 启动模式，该模式下通过设置 SW7 拨码开关选择启动存储介质，如图 2.17 所示：

SW7-1	SW7-2	SW7-3	SW7-4	Boot Device
OFF	ON	ON	OFF	Hyper Flash
OFF	OFF	ON	OFF	QSPI Flash
ON	OFF	ON	OFF	SD Card

图 2.17 i.MX-RT1050-EVK 内部启动介质选择

将制作好的 SylixOS 启动盘插入 i.MX-RT1050-EVK 开发板，SW7 拨码开关选择 SD 卡启动模式（即拨码为 1010）。J1 短接 5 和 6 脚，用 USB 线连接开发板 J28 口和电脑，电脑上会自动生成一个串口设备，使用 PuTTY 连接该串口，串口配置参数为：

- 波特率 115200；
- 8 位数据位；
- 无校验位；
- 1 位停止位；

设置完成后，按 SW4 按键复位开发板，串口会有如图 2.18 所示的信息输出，此时系统已成功进入 SD 卡启动模式下的 SylixOS Bootloader 中。

```
****************************************
*  SylixOS Bootloader For i.MX-RT1050  *
*     build Apr  8 2018 17:48:15       *
*  Input ? or help to get help menu    *
****************************************

> input any key to cancel auto run! 3

>
```

图 2.18 SylixOS Bootloader 启动打印

```
> ?
?               -       print help menu
help            -       print help menu
md              -       list memory value
ls              -       list files of sdcard
arg             -       get or set auto run arg
program         -       program bsp image from sdcard into hyperflash
ext             -       program extension image from sdcard into hyperflash
load            -       load bsp image from sdcard into sdram, and run
run             -       run bsp image in sdram

>
```

图 2.19 SylixOS Bootloader 支持的命令

各命令的详细说明如下：

- ***md*** 命令用于显示系统内存数据，默认从 0 地址开始，每次列出 1KB 的内存数据，重复输入 md 命令内存地址会自动累加。md 命令也可后跟一个内存地址，方便直接显示某处地址起始的内存。
- ***ls*** 命令用于列出 SD 卡上的文件和目录，方便查看 SD 卡上有哪些文件和目录。
- ***arg*** 命令用于修改或查看自动运行的命令。
- ***ext*** 命令用于将一个能在 HyperFlash 中运行的 Extension 镜像由 SD 卡烧写到 HyperFlash 中。
- ***program*** 命令用于将一个能在 HyperFlash 中运行的操作系统镜像由 SD 卡烧写到 HyperFlash 中；program 通过文件路径参数指定要烧写的文件（默认文件名为 bspimxrt1050.bin）。
- ***load*** 命令用于将一个能在 sdram 中运行的操作系统镜像由 SD 卡加载到 sdram 中并运行；load 通过文件路径参数指定要加载运行的文件（默认文件名为 bspimxrt1050.bin）。
- ***run*** 命令用于直接跳转到某内存地址处的操作系统镜像而运行，默认地址为 0x80000000。

2.5 烧录 SylixOS

输入如下命令烧写操作系统镜像到 HyperFlash 中：

```
program bsp.bin
```

```
> program bsp.bin
Program file bsp.bin(4795128B) from sdcard into hyperflash(at 0x60002000)
Erase sector:...................
Program page:|
Program success!
```

图 2.20 烧系统录镜像过程

输入如下命令烧写 Extension 镜像到 HyperFlash 中：

```
ext ext.bin    0x60800000
```

```
> ext ext.bin 0x60800000
Program file ext.bin(176B) from sdcard into hyperflash(at 0x60800000)
Erase sector:.
Program page/
Program success!
```

图 2.21 烧录 Extension 镜像过程

2.6 运行 SylixOS

修改 SW7 拨码开关为 HyperFlash 启动模式（即拨码为 0110）。复位开发板，系统即可从 HyperFlash 启动。SylixOS 系统启动后的打印信息如下：

```
Block device /dev/blk/sdcard-0 part 0 mount to /media/sdcard0 use vfat file system.
envionment variables load from /etc/profile fail, error: No such file or directory
[ifparam]No network parameter for [i.MXRT1050_0] from /etc/ifparam.ini, default parameters will be used.
Press <n> to NOT execute /etc/startup.sh (timeout: 1 sec(s))
can not open /etc/startup.sh: No such file or directory
sysname  : sylixos
nodename : sylixos
release  : LongYuan
version  : 1.5.6-1
machine  : NXP i.MX-RT1050 (Cortex-M7 600MHz FPv5)
build    : Mar  8 2018 11:24:55

                          [[                            (R)
 [[[[           [[[[      [[             [[[[    [[[[
[[  [[            [[                   [[  [[  [[  [[
[[      [[  [[    [[    [[[[    [[  [[  [[  [[  [[
 [[     [[  [[    [[      [[    [[  [[  [[  [[   [[
  [[    [[  [[    [[      [[     [[[[   [[  [[    [[
   [[   [[  [[    [[      [[      [[    [[  [[     [[
    [[  [[  [[    [[      [[     [[[[   [[  [[      [[
[[  [[  [[  [[    [[      [[    [[  [[  [[  [[  [[  [[
 [[[[    [[[[   [[[[[[  [[[[[[  [[  [[   [[[[    [[[[
           [[
          [[    KERNEL: LongWing(C) 1.5.6-1
       [[[[   COPYRIGHT ACOINFO Co. Ltd. 2006 - 2017

SylixOS license: Commercial & GPL.
SylixOS kernel version: 1.5.6-1 Code name: LongYuan

CPU     : NXP i.MX-RT1050 (Cortex-M7 600MHz FPv5)
CACHE   : 64KBytes L1-Cache (D-32K/I-32K)
PACKET  : NXP i.MX-RT1050 EVK
ROM SIZE: 0x00800000 Bytes (0x80000000 - 0x807fffff)
RAM SIZE: 0x00800000 Bytes (0x80800000 - 0x80ffffff)
BSP     : BSP version 1.0.0 for Octopus
[root@sylixos:/root]#
[root@sylixos:/root]#
```

图 2.22 系统启动打印信息

此时可以使用各种命令来操作 SylixOS，具体命令请查阅《SylixOS shell 用户手册》

执行如下命令，运行 Extension 程序，打印信息如下：

```
tc 0x60800000
```

```
[root@sylixos:/root]# tc 0x60800000
[root@sylixos:/root]# extension: hello SylixOS lite!

[root@sylixos:/root]#
```

图 2.23 运行 Extension 程序示例

第3章 SylixOS Lite 应用开发

3.1 SylixOS Lite 应用开发方法

标准版的 SylixOS 只能运行在支持 MMU 的处理器上，依靠 MMU 实现虚拟内存管理，应用程序是可以单独编译和动态加载运行的。

而 SylixOS Lite 可以运行在无 MMU 的处理器上，但不能实现应用程序的单独编译和动态加载运行。一般的，Lite 版的应用开发和单片机开发类似，应用程序和 BSP 代码属于同一工程，一同编译的，最后得到一个系统镜像。应用程序的源代码习惯放在 BSP 工程的 user 目录下。

为了方便用户开发 Lite 版的应用程序，SylixOS 引入了 Extension（扩展）的概念。Extension 是系统镜像的一个功能扩展，为独立工程，但需要基于对应的 Base 工程和 BSP 工程，独立编译后再通过静态链接的方式获取 Base 和 BSP 中的符号地址，从而生成独立镜像文件。

Extension 并不能动态加载运行，Extension 必须加载或烧录到其预先定义好的地址处，才能正确运行。开发 Extension 工程时需要人为确定好其代码段和数据段的位置，这两个段不能与 BSP 及其他 Extension 工程重叠。

对于 text 段在 HyperFlash 中的 Extension 工程，需要提前借助 BootLoader 的 ext 命令将其镜像文件烧录到要求的位置。系统启动后，可用 tc 或 Extension 命令附加地址参数来运行。对于 text 段在 SDRAM 中的 Extension 工程，镜像文件可以存放于文件系统中，直接使用系统 Extension 命令附加文件名及地址参数即可由系统加载 Extension 镜像到内存中运行。

使用 Extension 的好处：

- 便于工程分割，BSP 开发和应用程序开发可以很好的解耦；
- 免去了更新应用程序时需要一同编译和烧写 BSP 的过程，使得应用程序开发更快捷高效；
- 便于产品的更新维护，大部分情况下，产品软件更新只需要更新 Extension 部分即可，操作起来会更加方便灵活，不改动 BSP 镜像也使得系统可靠性更高；
- 便于技术保密，关键的算法或驱动可以编译到 BSP 镜像中，对 Extension 开发人员只开放 BSP 的符号表文件和头文件，而无须提供 BSP 的源文件。

3.2 工程镜像类型

i.MX-RT1050 的内存空间由以下部分组成：

表 3.1 i.MX-RT1050 的内存用空间分布

起始地址	空间大小	说明
0x00000000	512KB	ITCM
0x20000000	512KB	DTCM
0x40000000	4MB	AIPS
0x60000000	504MB	FLEXSPI
0x80000000	1536MB	SEMC
0xE0000000	1MB	PPB

在实际的 i.MX-RT1050-EVK 开发板上，FLEXSPI 链接的是 HyperFlash，共有 64MB 空间。SEMC 链接的是外部 SDRAM，共有 32MB 空间。

BSP 将 32MB 的外部 SDRAM 均分为了 4 个块，每块 8MB。第一块用于 text 段，第二块用于 data 段，第三块用于 DMA，第四块用于 Extension。

BSP 工程中的/bsp/config.h 中的宏 BSP_CFG_BOOT_MODE 用于条件编译系统镜像的 text 段是在 HyperFlash 中还是在 SDRAM 中，两种镜像除 text 段的链接位置不同外其他配置都是相同的。

Extension 的 text 段同样可以选择是在 HyperFlash 中还是在 SDRAM 中。如果在 HyperFlash 中则要跳过 BSP 占用的空间（预定义为 8MB），即(0x60000000 + 8 * 1024 * 1024)之后；如果在 SDRAM 中，则要链接到 Extension 块，即(0x80000000 + 24 * 1024 * 1024)之后。Extension 的 data 段则一定在 Extension 块中。

第二章介绍的快速体验中，系统镜像和 Extension 镜像都是固化到 HyperFlash 中的，便于演示和产品化，此时/bsp/config.h 中的设置为：

```
#define BSP_CFG_BOOT_MODE               BSP_CFG_HYPERFLASH_BOOT
```

此时/ext/config.h 中的设置为：

```
#define BSP_CFG_ROM_BASE_HYPERFLASH     (0x60000000 + 8    * 1024 * 1024)
#define BSP_CFG_ROM_BASE_SDRAM          (0x80000000 + 24 * 1024 * 1024)
#define BSP_CFG_ROM_BASE      (BSP_CFG_ROM_BASE_HYPERFLASH)
#define BSP_CFG_ROM_SIZE      (1 * 1024 * 1024)
#define BSP_CFG_RAM_BASE      (BSP_CFG_ROM_BASE_SDRAM)
#define BSP_CFG_RAM_SIZE      (1 * 1024 * 1024)
```

在 HyperFlash 中运行并不适合需要频繁更新镜像的开发阶段，为了提高开发效率，系统镜像和 Extension 镜像都建议加载到 SDRAM 中运行的。

此时/bsp/config.h 中的设置为：

```
#define BSP_CFG_BOOT_MODE               BSP_CFG_SDRAM_BOOT
```

此时/ext/config.h 中的设置为：

```
#define BSP_CFG_ROM_BASE_HYPERFLASH     (0x60000000 + 8    * 1024 * 1024)
#define BSP_CFG_ROM_BASE_SDRAM          (0x80000000 + 24 * 1024 * 1024)
#define BSP_CFG_ROM_BASE      (BSP_CFG_ROM_BASE_SDRAM)
#define BSP_CFG_ROM_SIZE      (1 * 1024 * 1024)
#define BSP_CFG_RAM_BASE      (BSP_CFG_ROM_BASE + BSP_CFG_ROM_SIZE)
#define BSP_CFG_RAM_SIZE      (1 * 1024 * 1024)
```

3.3 系统加载运行方法

固化两种镜像的方法在第二章中已经有详细介绍，现在介绍加载运行镜像的方法和步骤。

1. 设置 BSP_CFG_BOOT_MODE 为 BSP_CFG_SDRAM_BOOT，编译 BSP 工程，得到 bsp.bin 文件，即可在内存中运行的系统镜像。
2. 接下来就是如何把镜像文件放入 SD 卡中了。最简单的办法就是连接电脑直接拷贝文件。连接电脑可以通过 USB 读卡器，或大小卡转换器，或者有的电脑能直接插入 micrSD 卡，总之连接后 SD 卡会以一个 U 盘的形式出现在电脑上，此时就可以拷贝 bsp.bin 到 SD 卡上的 FAT 文件系统中。

3. SW7 拨码开关设置为 SD 卡启动（即拨码为 1010）；SD 启动卡插入开发板，网线连接网口，USB 线连接电脑和板载仿真器 USB 口（J28），开启串口调试窗口，按 SW4 按键复位开发板，进入 BootLoader 命令行界面。
4. 输入“load　bsp.bin”命令，加载系统镜像并运行。

```
****************************************
*  SylixOS Bootloader For i.MX-RT1050 *
*     build Mar  8 2018 14:36:09      *
*  Input ? or help to get help menu   *
****************************************

> ls
2000/ 1/ 1  8: 2:21        3392B  ext.bin
2018/ 3/10 18:41: 6        <DIR>  System Volume Information
2000/ 1/ 1  8: 5:16        2048B  a.txt
2018/ 3/12 14:43:20        <DIR>  sdram
2000/ 1/ 1  8: 0:22     4748944B  bsp.bin

> load bsp.bin
Load file bsp.bin(4748944B) from sdcard into sdram(at 0x80000000) and run
-
Load finished, run at 80000000
```

图 3.1 系统镜像加载过程

3.4 FTP 方式部署镜像文件

文件拷贝方式需要 SD 卡不断的在 PC 和开发板间插拔和文件拷贝，比较繁琐，只有在 SD 卡驱动或网络驱动还没有开发好时才使用。

通常情况下都是使用 FTP 来下载镜像文件到开发板的 SD 卡上。操作步骤如下：

1. 准备好一个已具备 SD 卡访问和以太网通信的开发板，连接好网线。
2. 输入 ifconfig 命令查看开发板的 IP 地址。

```
[root@sylixos:/root]# ifconfig
en1        Link encap: Ethernet HWaddr: 30:e0:0c:07:9b:8a
           Dev: i.MXRT1050 0 Ifidx: 2 DHCP: D4 D6 Spd: 100 Mbps
           inet addr: 10.4.0.196 netmask: 255.255.255.0
           gateway: 10.4.0.1 broadcast: 10.4.0.255
           inet6 addr: fe80::32e0:cff:fe07:9b8a Scope:link
           UP BROADCAST RUNNING MULTICAST MTU:1500 METRIC:1
           RX ucast packets:8383 nucast packets:420 dropped:0
           TX ucast packets:3 nucast packets:10 dropped:0
           RX bytes:567789 TX bytes:902

lo0        Link encap: Local Loopback
           Dev: N/A Ifidx: 1 DHCP: D4 D6 Spd: N/A
           inet addr: 127.0.0.1 netmask: 255.0.0.0
           P-to-P: 127.0.0.1 broadcast: Non
           inet6 addr: ::1 Scope:loopback
           UP LOOPBACK RUNNING MTU:0 METRIC:1
           RX ucast packets:3 nucast packets:0 dropped:0
           TX ucast packets:3 nucast packets:0 dropped:0
           RX bytes:168 TX bytes:168

dns0: 0.0.0.0
dns1: 0.0.0.0
default device is: en1
total net interface: 2
[root@sylixos:/root]#
```

图 3.2 ifconfig 命令

3. 输入 ping 命令，检测开发板与 PC 的连通性。

```
[root@sylixos:/root]# ping 10.4.0.195
Pinging 10.4.0.195

Reply from 10.4.0.195: bytes=32 time=1ms TTL=128
Reply from 10.4.0.195: bytes=32 time=0ms TTL=128
Reply from 10.4.0.195: bytes=32 time=0ms TTL=128
Reply from 10.4.0.195: bytes=32 time=0ms TTL=128

Ping statistics for 10.4.0.195:
    Packets: Send = 4, Received = 4, Lost = 0(0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

[root@sylixos:/root]#
```

图 3.3 ping 命令

4. 点击工具栏右侧 Device 按钮，进入 Device 视图，然后在左侧 Remote System Navigator 空白处右键，选择“New Device”菜单，如图 3.4 所示：

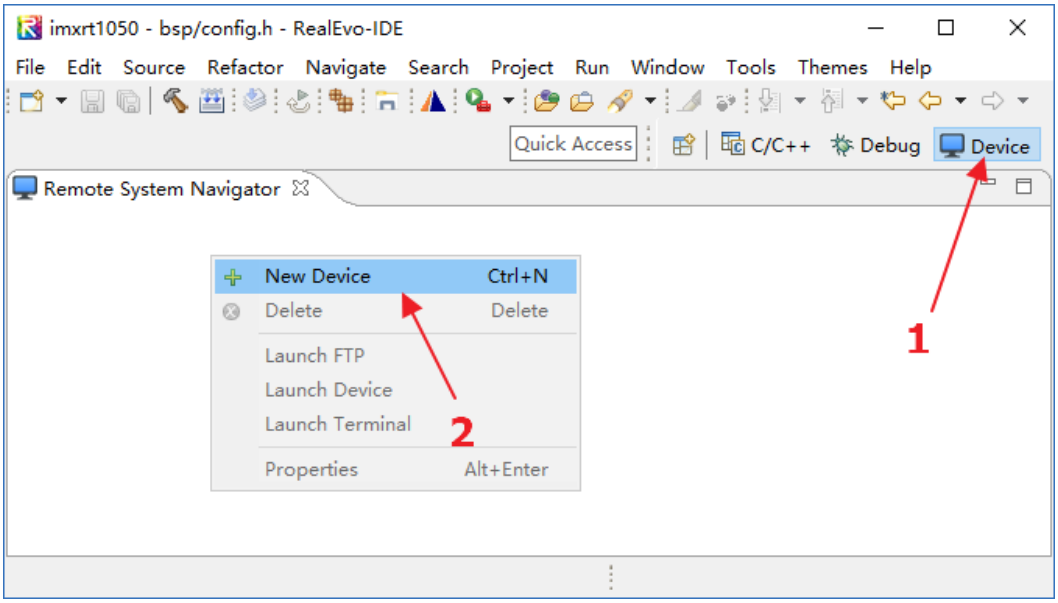

图 3.4 创建设备过程

5. “Device IP”栏填写开发板 IP 地址，其他选项默认值即可，点击“Finish”按钮完成 Device 的添加。

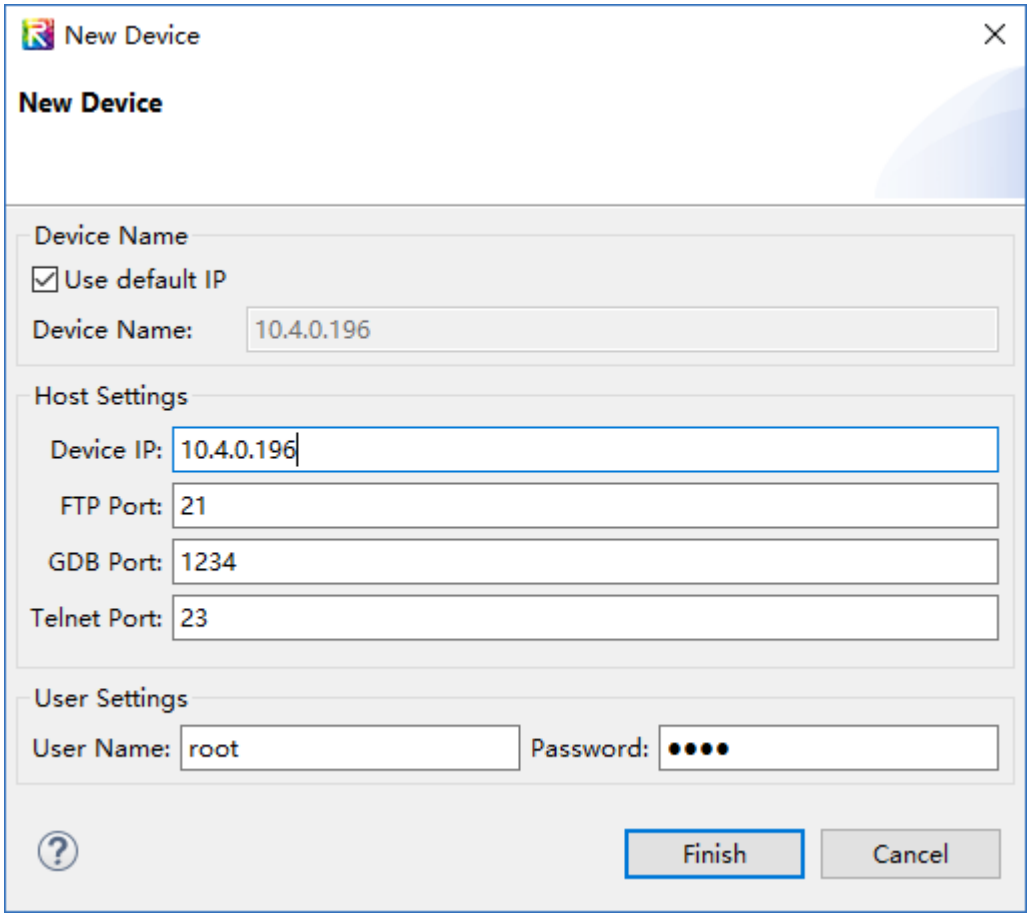

图 3.5 配置设备参数

6. 目标设备条目上右键，选择 Launch FTP，启动 PC 到开发板的 ftp 链接，如图 3.6 所示：

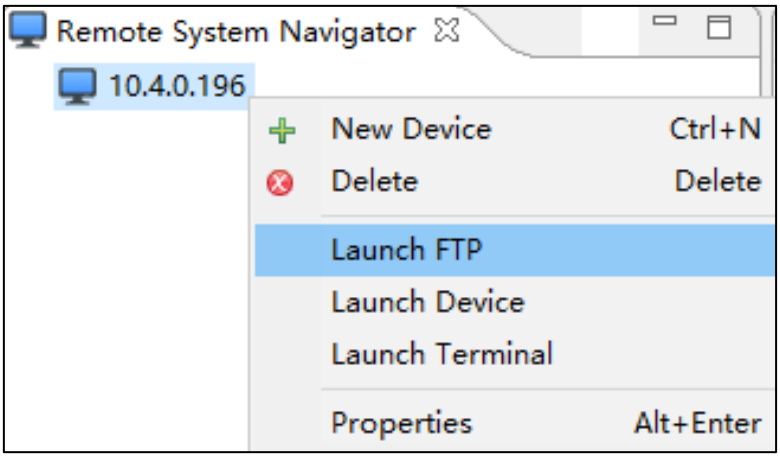

图 3.6　启动 FTP

7. 右侧开发板打开“/media/sdcard0”目录。

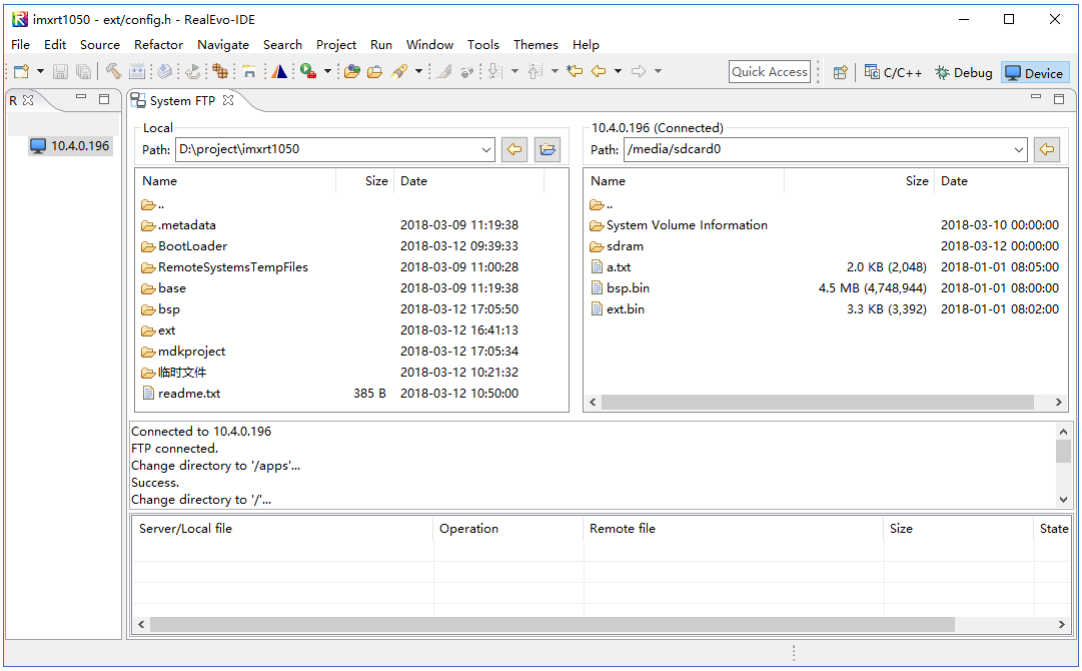

图 3.7　通过 FTP 访问目标文件系统

8. Local Path 找到需要下载的文件，右键该文件，选择“Upload”菜单，开启 FTP 下载。

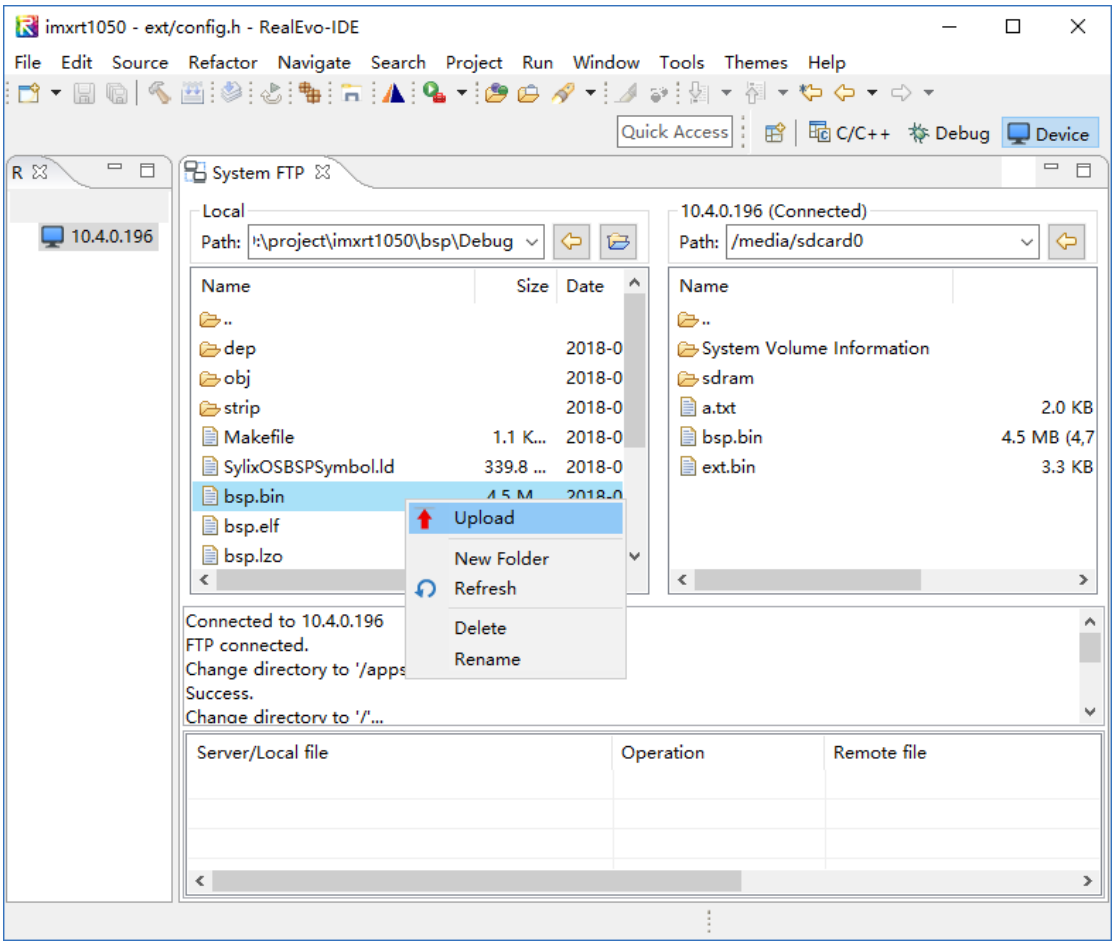

图 3.8　上传文件到目标系统

9. 如果目标目录有同名文件会有提示窗口弹出，选择 Overwrite 和 Always use this action，每次下载文件都会自动覆盖原有的同名文件。

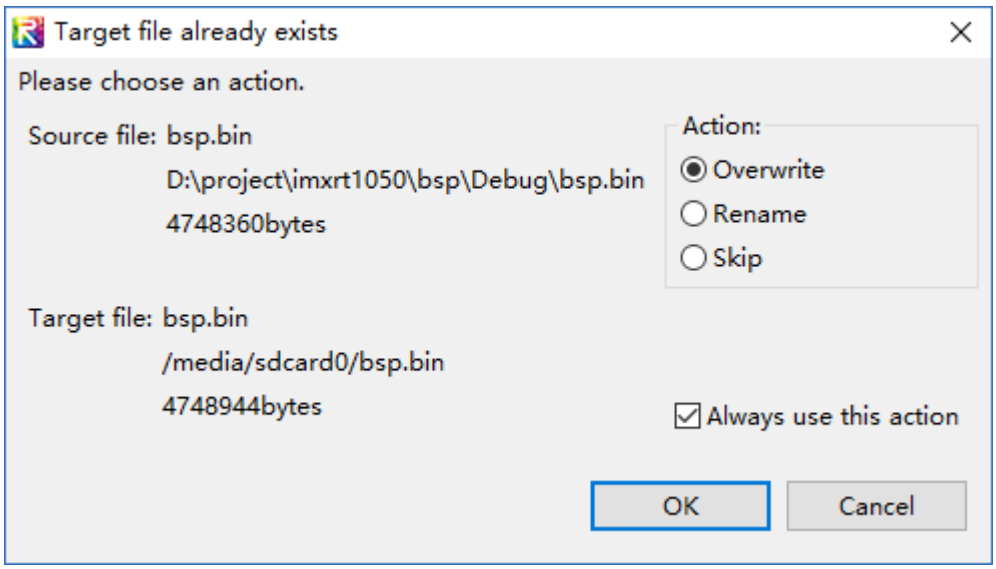

图 3.9　文件覆盖设置

10. 下载过程中会有进度提示，如图 3.11 所示：

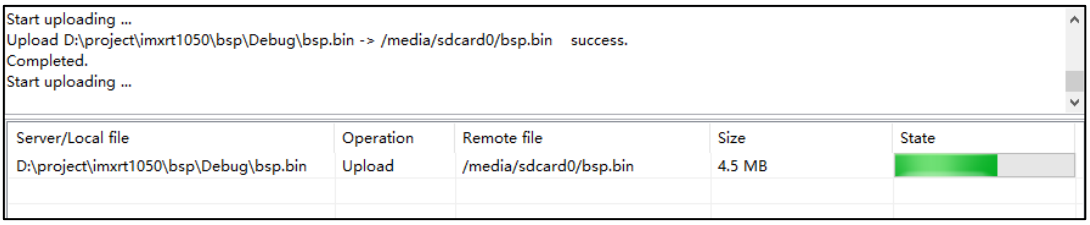

图 3.10 上传文件提示

11. 待下载完成，开发板上可以执行一下磁盘同步命令“sync”，防止下载内容未回写到 SD 卡上。

```
[root@sylixos:/root]#
[root@sylixos:/root]# sync
[root@sylixos:/root]#
```

图 3.11 磁盘同步命令

3.5 一键部署功能

为了最大可能的提高工程部署速度，IDE 集成了一键部署功能。一键部署的原理依旧是 FTP 文件传输，但工具的优化使得下载镜像极为便利。

1. 右键 Project Explorer 中的 BSP 工程，选择“Properties→SylixOS Project→Device Setting”打开 Device Setting 设置页面。

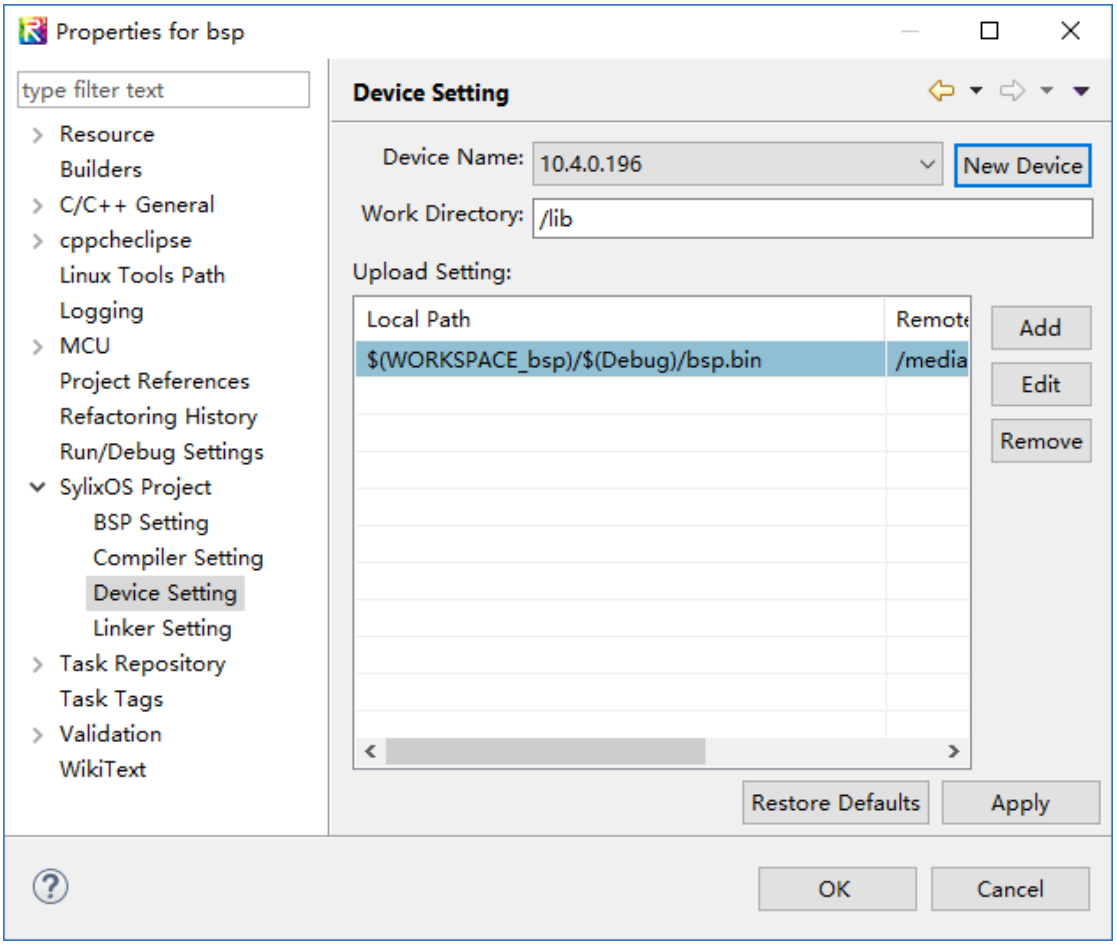

图 3.12 Device Setting 设置页面

2. Device Name 中选择目标设备，如果没有可以点击右边的 New Device 按钮创建设备。Device IP 填写开发板 IP 地址，其他配置按照图 3.13 填写，Password 默认为“root”。

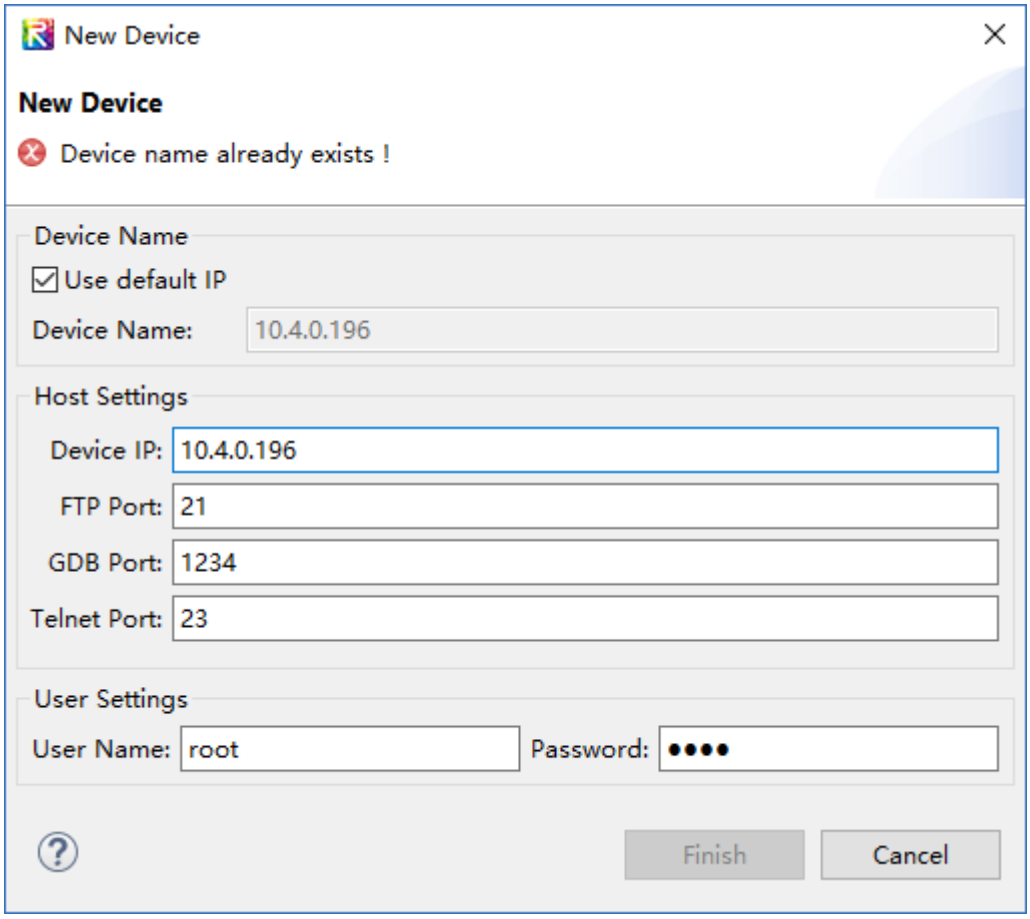

图 3.13 创建 device 窗口

3. 选择好目标设备后，点击按钮“Add”或“Edit”设置下载文件的源文件和目标文件。

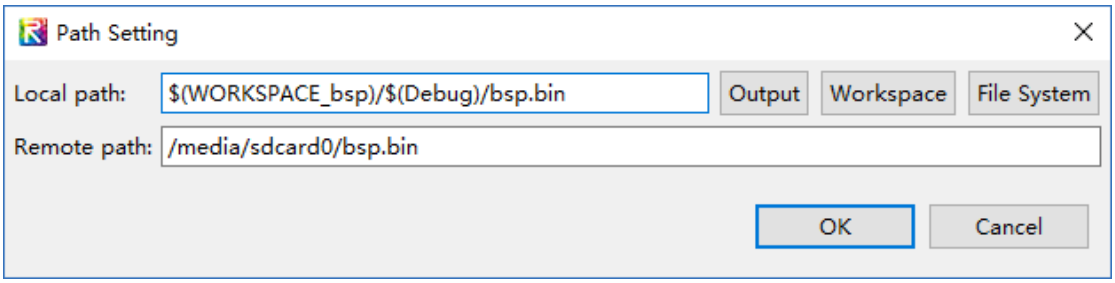

图 3.14 一键部署路径设置

4. 设置好一键部署路径后，点击“OK”，完成设置工作。
5. 编译工程后，右键 BSP 工程，选择“SylixOS→Upload”便可进行镜像部署。通常是使用快捷键“Alt + D”。

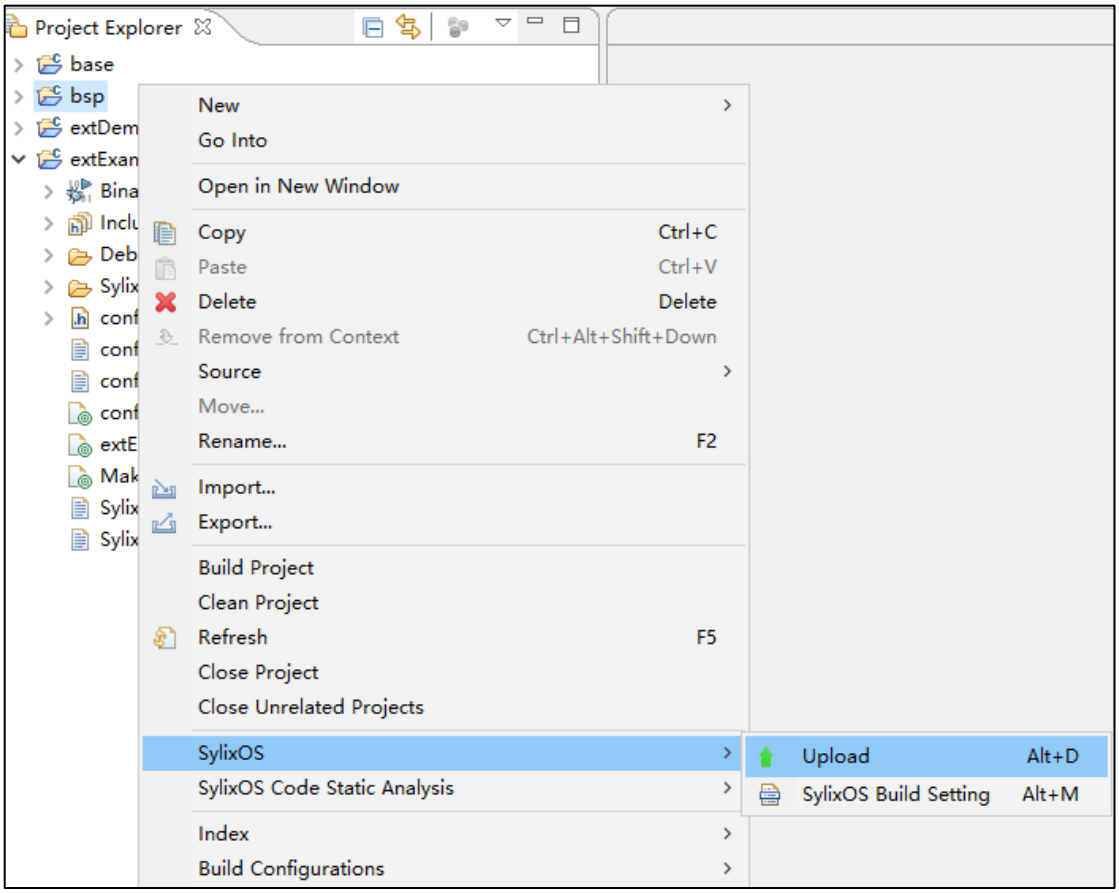

图 3.15 Upload 操作路径

6. 一键部署是会有下载进度的结果提示。

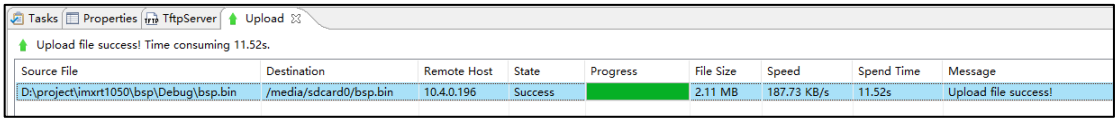

图 3.16 Upload 进度提示

3.6 开发 BSP 工程应用示例

现在我们尝试开发一个包含在 BSP 工程中的 hello world 程序。

1. 在 BSP 工程中，选中 user 目录上右键，选择“New→Source File”菜单，新建一个 bspHelloWorld.c 文件。

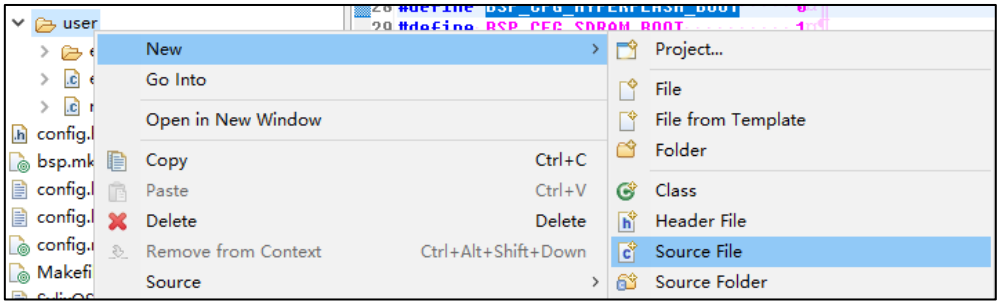

图 3.17 创建 C 源文件

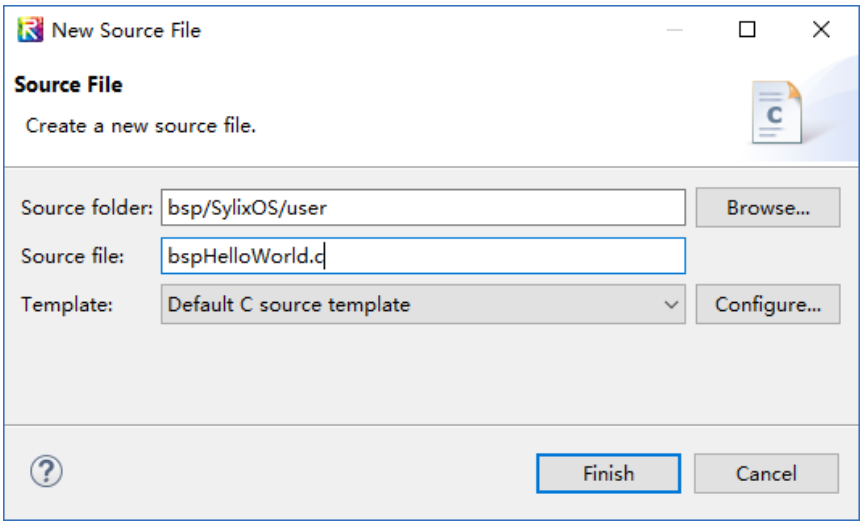

图 3.18　设置 C 源文件名

2. 打开 bspHelloWorld.c 文件编写如下代码：

```
#include <stdio.h>

int bspHelloWorld(int    argc, char    *argv[])
{
    printf("bsp:hello world!\n");

    return    0;
}
```

3. 打开 user/main.c 文件，t_main 函数最后面插入如下代码：

```
extern    int bspHelloWorld(int    argc, char    *argv[]);
API_TShellKeywordAdd("bspHelloWorld", bspHelloWorld);
```

注：API_TShellKeywordAdd 函数能向系统添加一个用户命令，便于在命令行下调用调试。

4. 编译 BSP 工程，获取 bsp.bin 文件。
5. 参考 3.3 章节加载运行系统镜像步骤，运行新的系统镜像。
6. 命令行输入“bspHelloWorld”命令，查看打印结果。

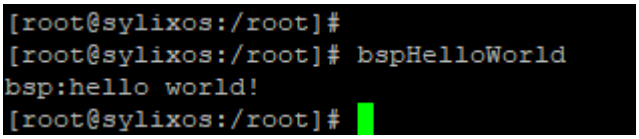

图 3.19　命令执行效果

3.7　开发 Extension 工程应用示例

现在我们尝试开发一个 Extension 版的 hello world 程序。

1. 参考 2.1.3 章节创建一个 extExample 工程，如图 3.20 所示。

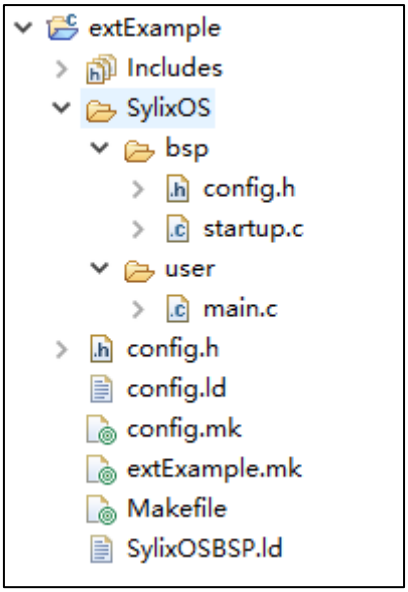

图 3.20　extExample 工程目录

2. 打开 extExample/SylixOS/user/main.c 文件，修改代码如下：

```
#include <stdio.h>

int    main (void)
{
       printf("ext:hello world!\n");

       return    (0);
}
```

3. 打开 extExample/config.h 文件，修改代码如下：

```
#define BSP_CFG_ROM_BASE_HYPERFLASH        (0x60000000 + 8    * 1024 * 1024)
#define BSP_CFG_ROM_BASE_SDRAM             (0x80000000 + 24 * 1024 * 1024)

#define BSP_CFG_ROM_BASE       (BSP_CFG_ROM_BASE_SDRAM)
#define BSP_CFG_ROM_SIZE       (1 * 1024 * 1024)
#define BSP_CFG_RAM_BASE       (BSP_CFG_ROM_BASE + BSP_CFG_ROM_SIZE)
#define BSP_CFG_RAM_SIZE       (1 * 1024 * 1024)
```

4. 编译 extExample 工程，下载 extExample.bin 到 SD 卡。

5. 输入“extension　0x81800000　/media/sdcard0/extExample.bin”命令，加载并运行 extExample.bin 镜像。

```
[root@sylixos:/root]#
[root@sylixos:/root]# extension 0x81800000  /media/sdcard0/extExample.bin
ext:hello world!
[root@sylixos:/root]#
```

图 3.21　运行 Extension 程序示例

第4章 仿真调试

SylixOS IDE 集成多种仿真调试功能，对于 i.MX-RT1050 平台可以使用 J-Link 仿真器进行代码调试。

4.1 准备工作

1. 首先安装好需要的软件，以及准备好开发板和 J-Link 仿真器，本文档调试使用的各软硬件版本如下：
   - 操作系统 ：Windows10 64 位教育版
   - RealEvo-IDE ：V3.7.3 Ultimate
   - 目标开发板 ：MIMXRT1050-EVK
   - J-Link 调试器：硬件为 J-LinkV9，固件版本 V9.40，软件版本 V6.31a
2. 去掉开发板上 J32，J33 的跳帽，断开板载 DAP 仿真器与 SOC 的 SW 仿真接口，防止与 J-Link 仿真器硬件冲突。用跳帽短接 J1 的 5 脚和 6 脚，用 USB 线连接电脑与 J28 口，使用 USB 接口为开发板供电，同时使用板载 DAP 仿真器提供的 USB 转串口功能作为串口调试接口。
3. 关闭或断开 J-Link 仿真接口 2 脚的电源输出功能，使用 20PIN 排线连接 J-Link 与开发板，J-Link 的 USB 口连接电脑。
4. 修改 BSP 工程中的 BSP_CFG_BOOT_MODE 宏，定义为 BSP_CFG_SDRAM_BOOT，编译获取 sdram 版系统镜像。镜像运行前需要初始化外部 sdram 并加载镜像到 sdram 中。

4.2 设置调试参数

1. 选择“Run→Debug Configurations”或者点击 Debug 按钮（小虫子图标）右边三角下拉框选择“Debug Configurations”。打开 Debug Configurations 设置页面。右键“GDB SEGGER J-Link Debugging”选择“New”新建一个调试对象。

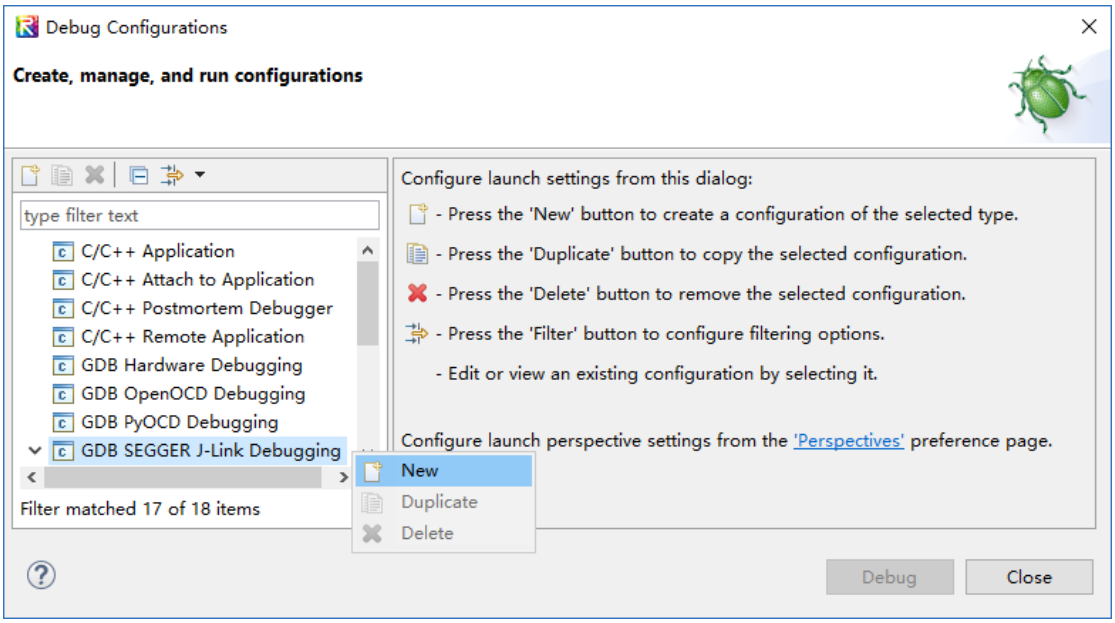

图 4.1 新建调试对象

2. “Name”栏为调试对象名称。“Name”栏下为 5 个设置标签页。

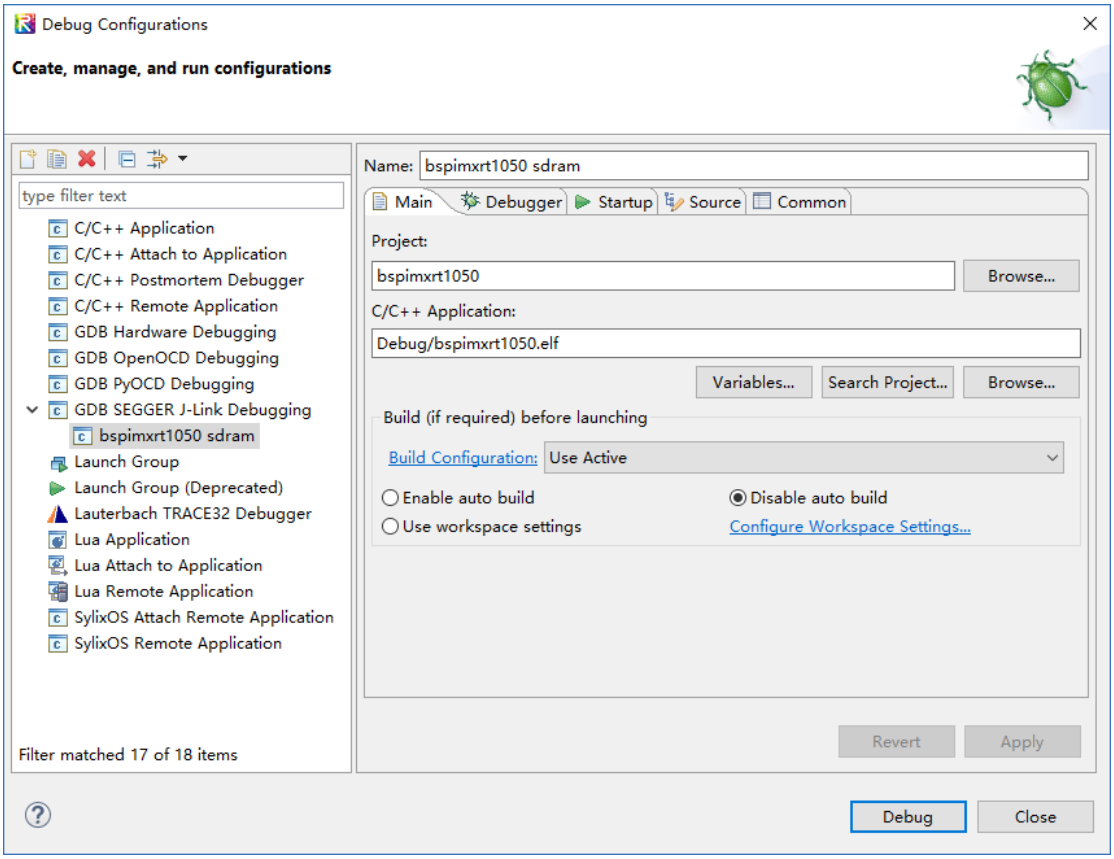

图 4.2 调试对象设置界面

3. “Main”标签页

- “Project”栏填写要仿真的工程名称。或点击“Browse”按钮，在弹出的对话框中选择。
- “C/C++ Application”栏填写要仿真的.elf 目标文件，或通过下方三个按钮进行选择，一般使用“Search Project...”。
- “Build before launching”栏是选择启动仿真前是否要编译工程，一般默认即可。

4. “Debugger”标签页

- “Actual executable”填写实际要运行的 J-Link GDB Server 程序，需要到 J-Link 驱动安装路径下查找，一般都能自动识别。这里的值为“C:/Program Files (x86)/ SEGGER/ JLink_V631a// JLinkGDBServerCL.exe”。
- “Device name”填写目标处理器型号，这里是“MCIMXRT1052”
- “Executable”填写要调用的本地 GDB Client 可执行程序，也可以通过右边的按钮进行选择。这里使用的值为“C:\ACOINFO\RealEvo\compiler\arm-sylixoslitele-toolchain\bin\ arm-sylixoslitele-eabi-gdb.exe”。
- 其他选项按图 4.3 设置。

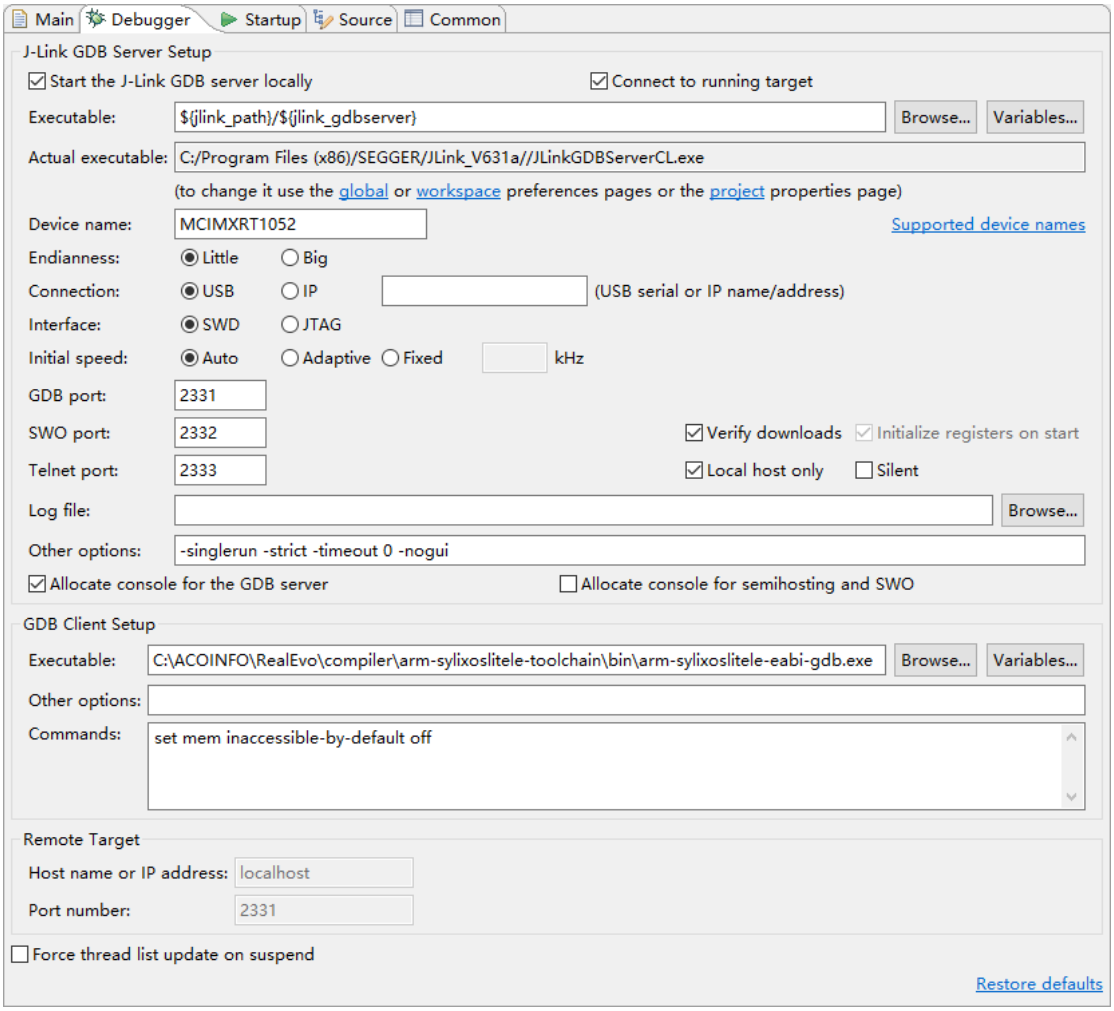

图 4.3 Debugger 标签页设置

5. “Startup”标签页

- “Initialization Commangds”栏用于设置目标初始化命令，方框中用于填写要执行的命令。因为 sdram 版的系统镜像是要在外部 sdram 中运行，所以在仿真运行前需要通过此处添加的命令来初始化外部 sdram，初始化外部 sdram 命令如下：

```
monitor writeu32 0x400FC068=0xffffffff
monitor writeu32 0x400FC06C=0xffffffff
monitor writeu32 0x400FC070=0xffffffff
monitor writeu32 0x400FC074=0xffffffff
monitor writeu32 0x400FC078=0xffffffff
monitor writeu32 0x400FC07C=0xffffffff
monitor writeu32 0x400FC080=0xffffffff
monitor writeu32 0x400D8030=0x00002001
monitor writeu32 0x400D8100=0x001d0000
monitor writeu32 0x400FC014=0x00010D40
monitor writeu32 0x401F8014=0x00000000
monitor writeu32 0x401F8018=0x00000000
monitor writeu32 0x401F801C=0x00000000
```

```
monitor writeu32 0x401F8020=0x00000000
monitor writeu32 0x401F8024=0x00000000
monitor writeu32 0x401F8028=0x00000000
monitor writeu32 0x401F802C=0x00000000
monitor writeu32 0x401F8030=0x00000000
monitor writeu32 0x401F8034=0x00000000
monitor writeu32 0x401F8038=0x00000000
monitor writeu32 0x401F803C=0x00000000
monitor writeu32 0x401F8040=0x00000000
monitor writeu32 0x401F8044=0x00000000
monitor writeu32 0x401F8048=0x00000000
monitor writeu32 0x401F804C=0x00000000
monitor writeu32 0x401F8050=0x00000000
monitor writeu32 0x401F8054=0x00000000
monitor writeu32 0x401F8058=0x00000000
monitor writeu32 0x401F805C=0x00000000
monitor writeu32 0x401F8060=0x00000000
monitor writeu32 0x401F8064=0x00000000
monitor writeu32 0x401F8068=0x00000000
monitor writeu32 0x401F806C=0x00000000
monitor writeu32 0x401F8070=0x00000000
monitor writeu32 0x401F8074=0x00000000
monitor writeu32 0x401F8078=0x00000000
monitor writeu32 0x401F807C=0x00000000
monitor writeu32 0x401F8080=0x00000000
monitor writeu32 0x401F8084=0x00000000
monitor writeu32 0x401F8088=0x00000000
monitor writeu32 0x401F808C=0x00000000
monitor writeu32 0x401F8090=0x00000000
monitor writeu32 0x401F8094=0x00000000
monitor writeu32 0x401F8098=0x00000000
monitor writeu32 0x401F809C=0x00000000
monitor writeu32 0x401F80A0=0x00000000
monitor writeu32 0x401F80A4=0x00000000
monitor writeu32 0x401F80A8=0x00000000
monitor writeu32 0x401F80AC=0x00000000
monitor writeu32 0x401F80B0=0x00000010
monitor writeu32 0x401F80B4=0x00000000
monitor writeu32 0x401F80B8=0x00000000
monitor writeu32 0x401F8204=0x000110F9
monitor writeu32 0x401F8208=0x000110F9
monitor writeu32 0x401F820C=0x000110F9
monitor writeu32 0x401F8210=0x000110F9
monitor writeu32 0x401F8214=0x000110F9
```

```
monitor writeu32 0x401F8218=0x000110F9
monitor writeu32 0x401F821C=0x000110F9
monitor writeu32 0x401F8220=0x000110F9
monitor writeu32 0x401F8224=0x000110F9
monitor writeu32 0x401F8228=0x000110F9
monitor writeu32 0x401F822C=0x000110F9
monitor writeu32 0x401F8230=0x000110F9
monitor writeu32 0x401F8234=0x000110F9
monitor writeu32 0x401F8238=0x000110F9
monitor writeu32 0x401F823C=0x000110F9
monitor writeu32 0x401F8240=0x000110F9
monitor writeu32 0x401F8244=0x000110F9
monitor writeu32 0x401F8248=0x000110F9
monitor writeu32 0x401F824C=0x000110F9
monitor writeu32 0x401F8250=0x000110F9
monitor writeu32 0x401F8254=0x000110F9
monitor writeu32 0x401F8258=0x000110F9
monitor writeu32 0x401F825C=0x000110F9
monitor writeu32 0x401F8260=0x000110F9
monitor writeu32 0x401F8264=0x000110F9
monitor writeu32 0x401F8268=0x000110F9
monitor writeu32 0x401F826C=0x000110F9
monitor writeu32 0x401F8270=0x000110F9
monitor writeu32 0x401F8274=0x000110F9
monitor writeu32 0x401F8278=0x000110F9
monitor writeu32 0x401F827C=0x000110F9
monitor writeu32 0x401F8280=0x000110F9
monitor writeu32 0x401F8284=0x000110F9
monitor writeu32 0x401F8288=0x000110F9
monitor writeu32 0x401F828C=0x000110F9
monitor writeu32 0x401F8290=0x000110F9
monitor writeu32 0x401F8294=0x000110F9
monitor writeu32 0x401F8298=0x000110F9
monitor writeu32 0x401F829C=0x000110F9
monitor writeu32 0x401F82A0=0x000110F9
monitor writeu32 0x401F82A4=0x000110F9
monitor writeu32 0x401F82A8=0x000110F9
monitor writeu32 0x402F0000=0x10000004
monitor writeu32 0x402F0008=0x00030524
monitor writeu32 0x402F000C=0x06030524
monitor writeu32 0x402F0010=0x8000001B
monitor writeu32 0x402F0014=0x8200001B
monitor writeu32 0x402F0018=0x8400001B
monitor writeu32 0x402F001C=0x8600001B
```

```
monitor writeu32 0x402F0020=0x90000021
monitor writeu32 0x402F0024=0xA0000019
monitor writeu32 0x402F0028=0xA8000017
monitor writeu32 0x402F002C=0xA900001B
monitor writeu32 0x402F0030=0x00000021
monitor writeu32 0x402F0004=0x000079A8
monitor writeu32 0x402F0040=0x00000F31
monitor writeu32 0x402F0044=0x00652922
monitor writeu32 0x402F0048=0x00010920
monitor writeu32 0x402F004C=0x50210A08
monitor writeu32 0x402F0080=0x00000021
monitor writeu32 0x402F0084=0x00888888
monitor writeu32 0x402F0094=0x00000002
monitor writeu32 0x402F0098=0x00000000
monitor writeu32 0x402F0090=0x80000000
monitor writeu32 0x402F009C=0xA55A000F
monitor sleep 100
monitor writeu32 0x402F0090=0x80000000
monitor writeu32 0x402F009C=0xA55A000C
monitor sleep 100
monitor writeu32 0x402F0090=0x80000000
monitor writeu32 0x402F009C=0xA55A000C
monitor sleep 100
monitor writeu32 0x402F00A0=0x00000033
monitor writeu32 0x402F0090=0x80000000
monitor writeu32 0x402F009C=0xA55A000A
monitor sleep 100
monitor writeu32 0x402F004C=0x50210A09
```

- “RAM application”是否为 RAM 中的应用程序，需要勾选。
- “Run/Restart Commands”栏方框内填写加载或下载完成后仿真运行前要操作的命令。此处需要设置向量表偏移，初始化栈地址及 PC 值。命令如下：

```
monitor writeu32 0xE000ED08=0x80000000
monitor reg sp = (0x80000000)
monitor reg pc = (0x80000004)
```

- 其他项按照图 4.4 进行设置。

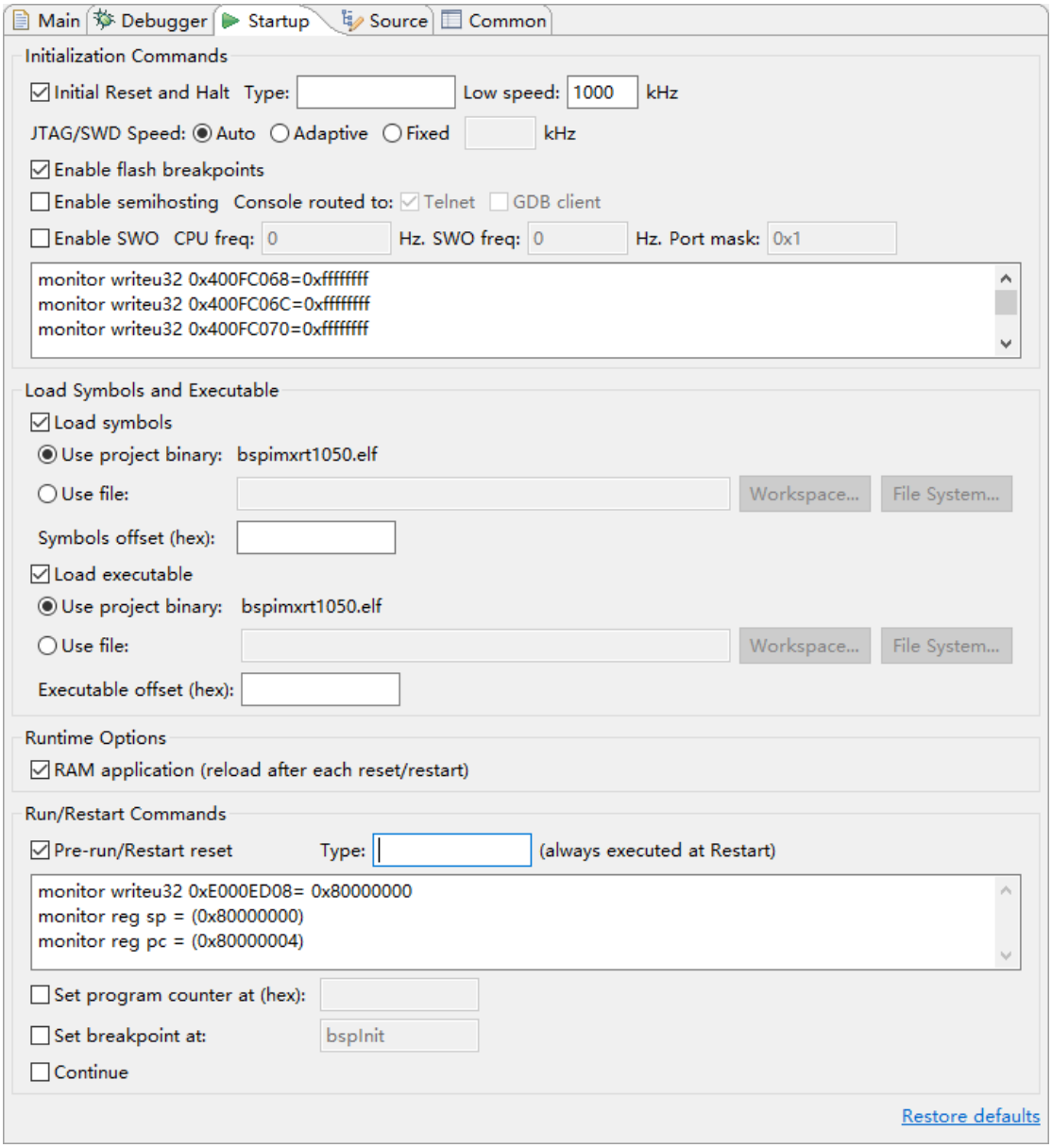

图 4.4 “Start”标签页设置

4.3 启动调试

点击“Debug”按钮启动仿真调试，IDE 会进入 Debug 视图模式，不同配置下界面会有所区别，此处的界面如下图所示：

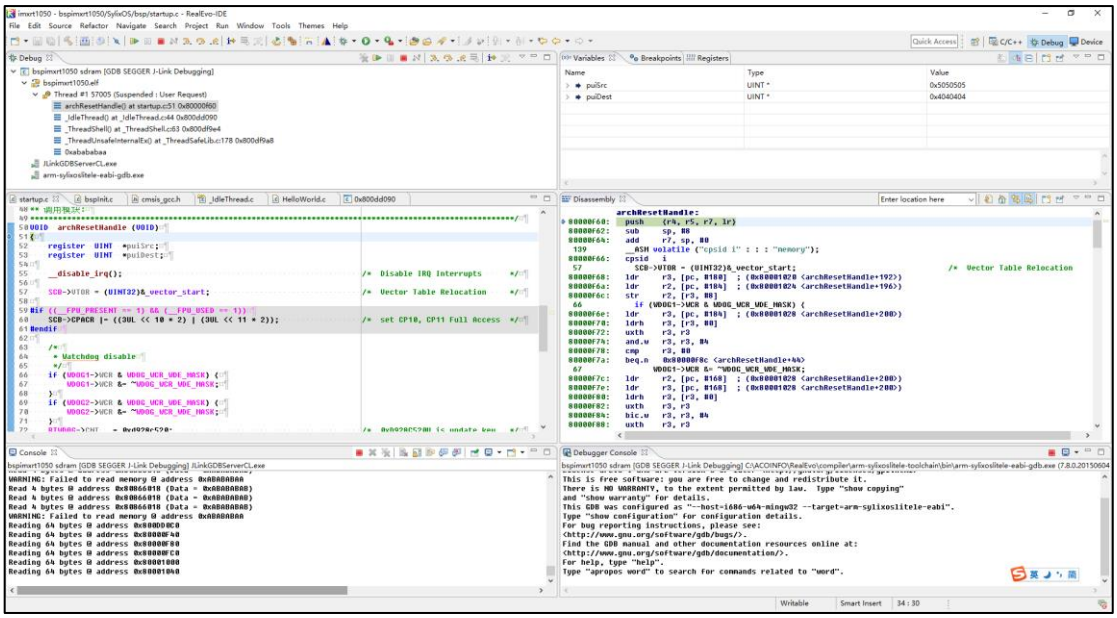

图 4.5 Debug 视图模式

按 F8 快捷键即可进行全速仿真，另外单步执行、暂停、设置断点、查看变量等都可以使用。

4.4 调试 Extension

1. 编译好 sdram 版的 BSP 工程和 Extension 工程。
2. 启动 BSP 工程仿真，等待镜像加载完成。

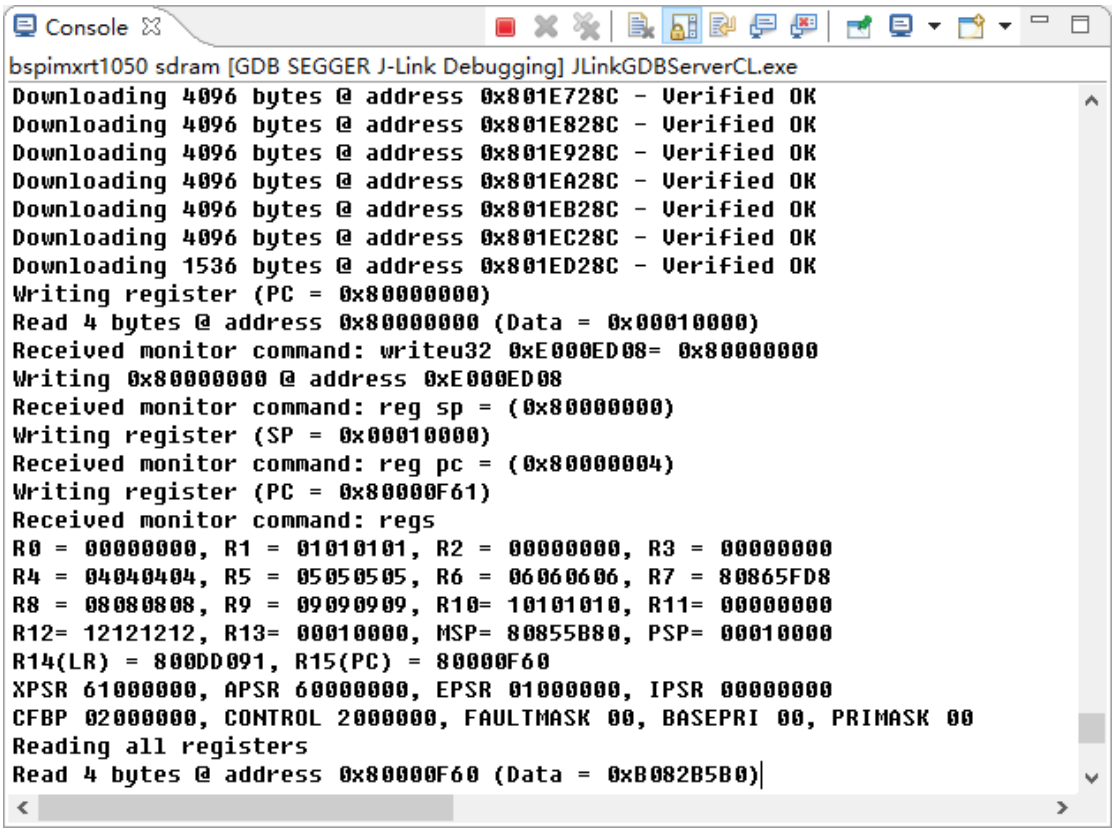

图 4.6 Console 窗口

3. 导入 Extension 工程的.elf 文件。方法是在 Debugger Console 窗口中输入“add-symbol-file D:/project/imxrt1050/extExample/Debug/extExample.elf 0x81800000”。“add-symbol-file”为 GDB 命令，第一个参数为要导入的.elf 文件，第二个参数为对应文件的链接地址。

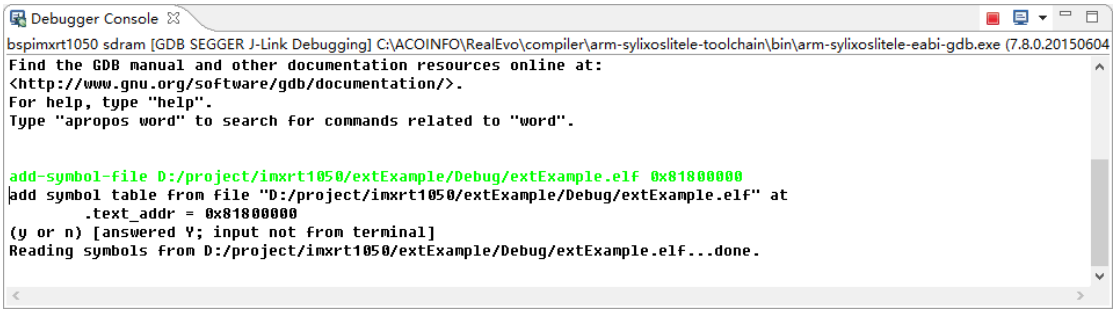

图 4.7 Debugger Console 窗口

4. 在 Extension 工程源文件打一个断点，然后全速仿真，加载运行 Extension 工程，仿真会在断点处停止。

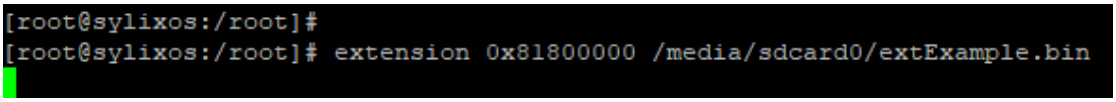

图 4.8 加载运行 Extension 工程

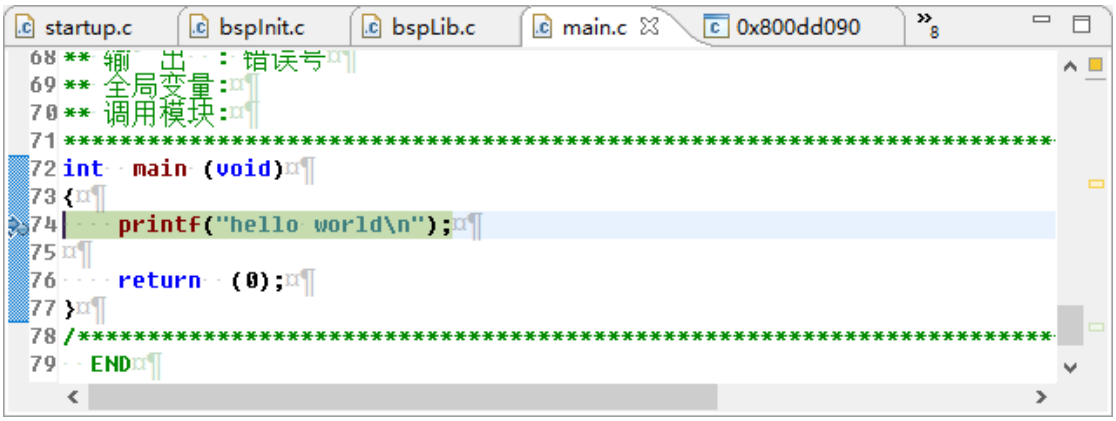

图 4.9 仿真在 Extension 源码处停止

第5章 参考资料

如果还想进一步学习研究 SylixOS，可以参考下面一些具体的资源：

i.MX-RT1050 官网页面
https://www.nxp.com/cn/products/processors-and-microcontrollers/applications-processors/i.mx-applications-processors/i.mx-rt-series/i.mx-rt1050-crossover-processor-with-arm-cortex-m7-core:i.MX-RT1050 NXP
对 i.MX-RT1050 芯片提供的各种资源。

i.MX RT1050 EVK 官网页面
https://www.nxp.com/cn/products/processors-and-microcontrollers/applications-processors/i.mx-applications-processors/i.mx-rt-series/i.mx-rt1050-evaluation-kit:MIMXRT1050-EVK NXP
对 i.MX RT1050 EVK 开发板提供的各种资源。

SylixOS IDE 免费申请页面
http://www.acoinfo.com/html/experience.php
在这里填写表格，可以免费申请体验版 IDE。申请提交后，翼辉信息工作人员会主动联系您，向您发放 IDE 下载链接及注册码。

翼辉信息官网
http://www.acoinfo.com/
里面有翼辉信息及其产品的相关介绍，下载栏还有各种文档手册可看。

官方的 QQ 群
SylixOS 技术讨论（32537017）
方便用户和爱好者实时讨论 SylixOS 相关技术问题。但比较深入些的技术讨论还是建议在官方论坛进行，这样可以保留讨论历史，便于遇到同样问题的人查阅，避免了相同问题重复讲解。

官方微信公众号

官方微信公众号 acoinfo
公众号上可以申请 IDE，查看公司介绍、产品介绍、联系方式，可以获取最新技术动态。

SylixOS 官网
http://www.sylixos.com/
这里可以下载到 SylixOS 的源码。

SylixOS 官方 git 仓库
http://git.sylixos.com/cgit/

这里可以下载到最新的支持库、bsp 或其他组件源码。

SylixOS 官方 wiki
http://wiki.sylixos.com/index.php/%E9%A6%96%E9%A1%B5
这里有一些教程和博客文章。

SylixOS 官方论坛
http://bbs.sylixos.com/forum.php
可以在这里学习和讨论 SylixOS 开发中的诸多问题。

官方文档
以下是学习 SylixOS 必需研读的官方文档，在官网或论坛都可下载到。
《翼辉信息宣传册电子版 V3.2》
《RealEvo 软件注册步骤》
《RealEvo-IDE 快速入门》
《RealEvo-IDE 使用手册》
《RealEvo-Simulator 使用手册》
《RealEvo-QtSylixOS 使用手册》
《SylixOS shell 用户手册》
《SylixOS 应用开发手册》
《SylixOS 设备驱动程序开发》
《SylixOS 开发 i.MX-RT1050 入门手册》

销售与服务网络

北京翼辉信息技术有限公司

地址：北京市海淀区中关村翠湖科技园 12 号楼

电话：010-56082458

传真：010-56082457

邮箱：acoinfo@acoinfo.com

南京翼辉信息技术有限公司

地址：南京市雨花台区软件大道 180 号大数据产业基地 7 幢 6 楼

电话：025-83127300

传真：025-83127399

邮箱：nanjing@acoinfo.com

翼辉信息官网
www. acoinfo.com

SylixOS社区
www. sylixos.com

翼辉信息公众号
acoinfo

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