

# MAPS-K64 用户指南

## 1 引言

本文详细描述了如何搭建MAPS-K64的软件开发环境，如何构建和运行MAPS-K64软件包中提供的演示程序。

## 2 概述

### 2.1 MAPS-K64软件包

MAPS-K64 软件包，提供了对飞思卡尔Kinetis K64内核以及外设的软件支持。MAPS-K64软件包中包括了设备的硬件抽象层(HAL)，建立在HAL之上的设备驱动和演示程序。演示程序演示了设备驱动和HAL的用法和MAPS-K64的主要功能。此外该软件包中还包括了最新的RTOS内核，USB协议栈和其他软件模块用于扩展和增强MAPS-K64的功能。

## 目录

|     |                              |   |
|-----|------------------------------|---|
| 1   | 引言 .....                     | 1 |
| 2   | 概述 .....                     | 1 |
| 2.1 | MAPS-K64 软件包 .....           | 1 |
| 2.2 | 硬件平台 .....                   | 2 |
| 2.3 | 支持的工具链 .....                 | 2 |
| 3   | 构建和运行演示程序 .....              | 2 |
| 3.1 | IAR Embedded Workbench ..... | 2 |
| 3.2 | KEIL uVision IDE .....       | 7 |

## 2.2 硬件平台

MAPS-K64和MAPS -Dock.

## 2.3 支持的工具链

1. IAR Embedded Workbench v7.20
2. Keil uVision IDE v5.11

## 3 构建和运行演示程序

本章描述了用IAR Embedded Workbench和Keil uVision IDE构建，运行和调试MAPS -K64软件包中提供的演示程序的流程。

以下以MAPS -K64软件包中的hello\_world演示程序为例。

### 3.1 IAR Embedded Workbench

#### 3.1.1 构建演示程序

演示程序的工作区文件位于：

```
<install_dir>/demos/<demo_name>/<compiler>/<board_name>/<demo_name>.eww
```

以hello\_world程序为例，IAR工作区文件位于：

```
<install_dir>/demos/hello_world/iar/mapsk64/hello_world.eww
```

由于演示程序需要使用MAPS的设备驱动库来调用HAL和外设驱动，所以在构建演示程序之前，需要先构建MAPS的设备驱动库ksdk\_platform\_lib.a，设备驱动库的工程文件已经添加进了演示程序的工作区文件中。

打开演示程序的IAR工作区文件，在IAR工作区中选择设备驱动库工程ksdk\_platform\_lib，然后点击“Make”按钮：

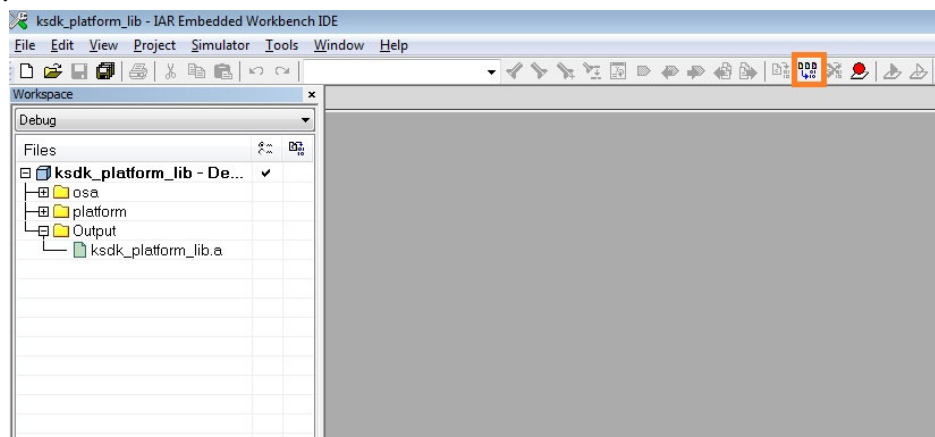


图1 构建设备驱动库

构建完成后，ksdk\_platform\_lib.a就生成在相应的文件夹中。

然后选择hello\_world工程，点击”make”按钮来构建演示程序：

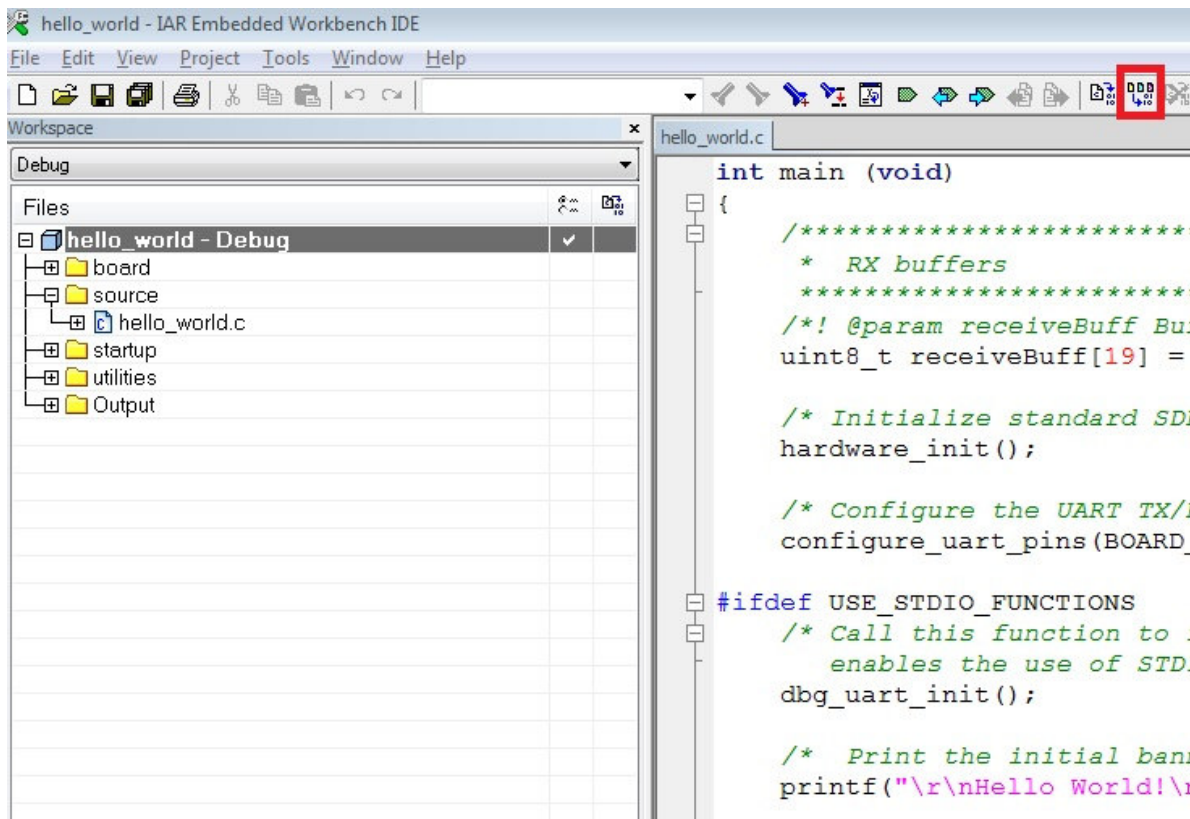


图2 构建hello\_world演示程序

构建完成后，IAR会在构建窗口显示相关的信息：

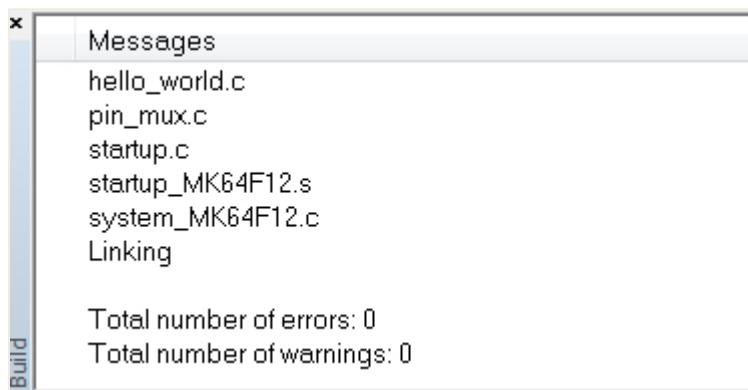


图3 构建hello\_world程序成功

### 3.1.2 运行演示程序

按照如下步骤在IAR中下载和运行演示程序：

1. 使用J-Link调试器。
2. 在电脑上打开一个终端程序，例如Putty，然后连接串口线到MAPS-Dock板的CN7端子，按下面的参数配置串口终端：
  - a. 波特率115200
  - b. 无校验
  - c. 8位数据位
  - d. 1位停止位

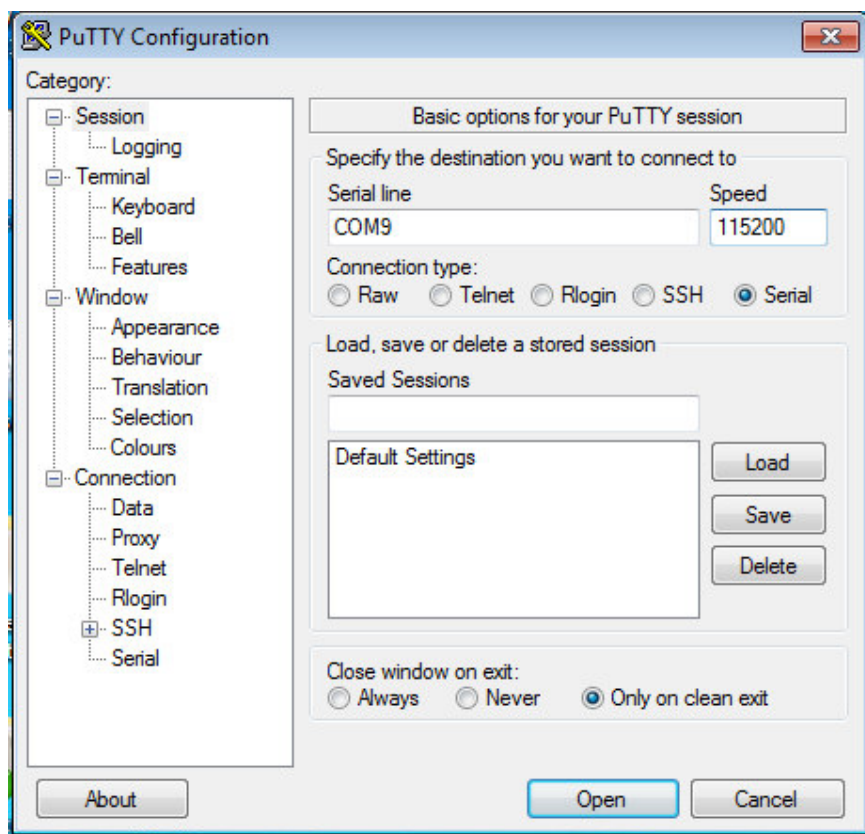


图4 终端（PuTTY）配置

### 3. 配置IAR的调试器

- 在debugger页面选择适合的调试器

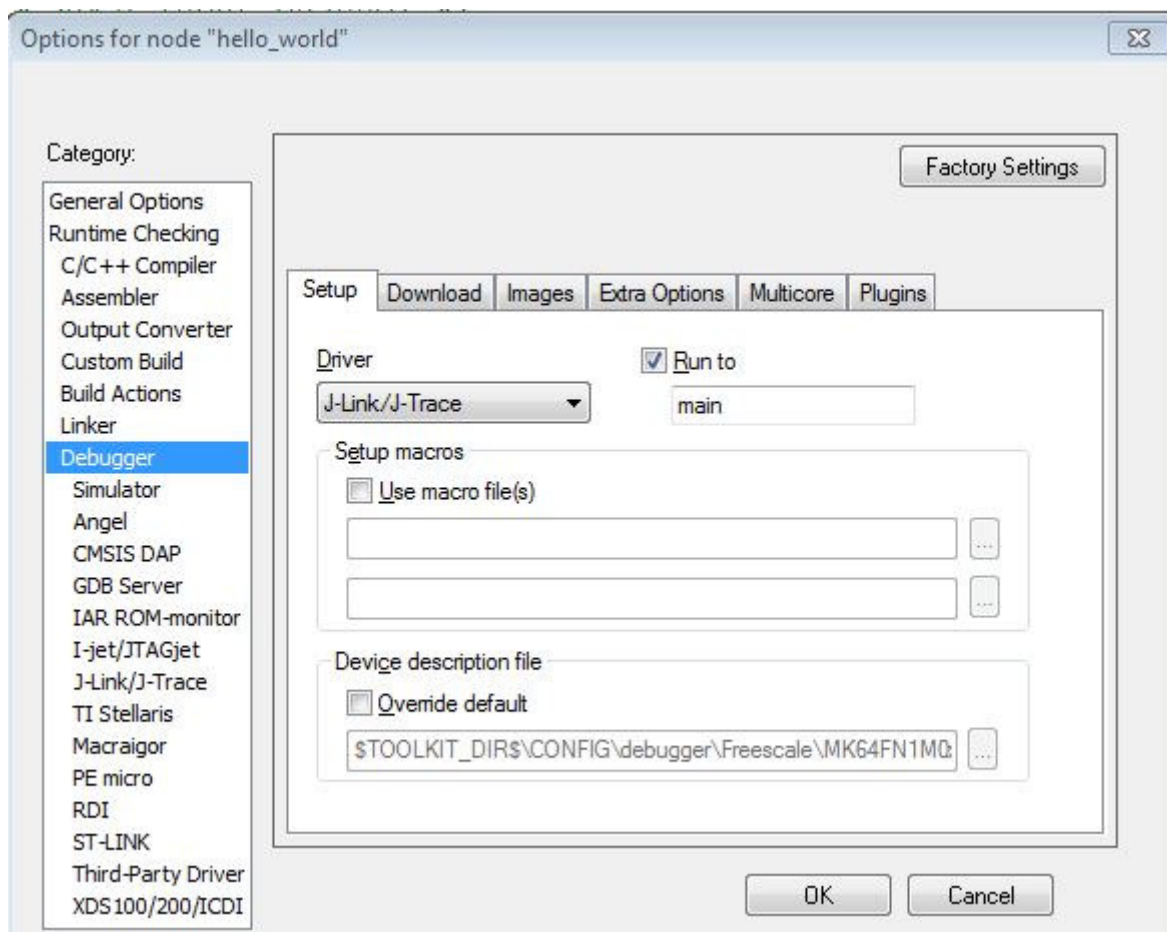


图5 MAPS-K64的debugger配置

- 在J-Link/J-Trace选择SWD作为debugger接口

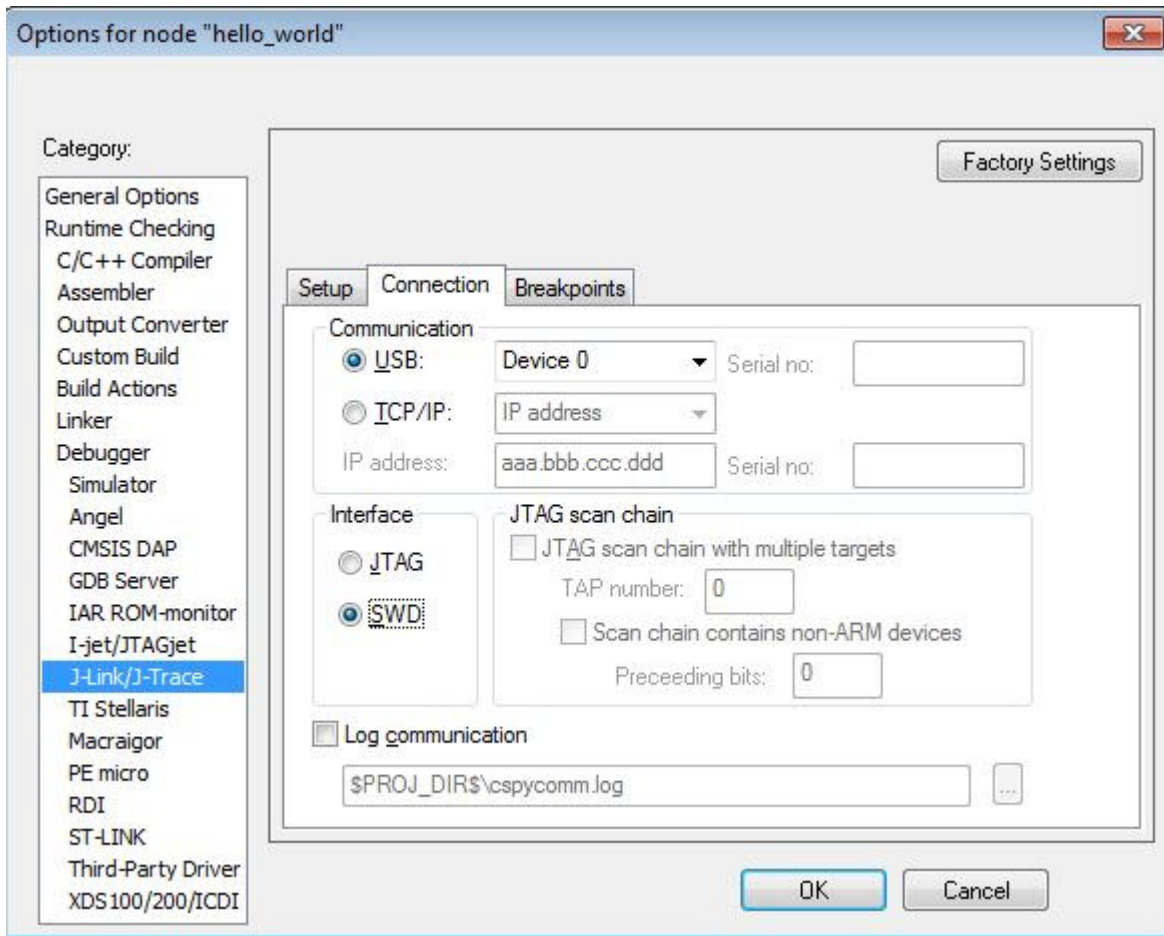


图6 MAPS-K64的debugger配置

4. 调试器成功连接后，点击“Download and Debug”按钮把程序下载到目标板中



图7 Download and debug按钮

5. 下载完成后，IAR的调试器会停在main()函数的开始处：

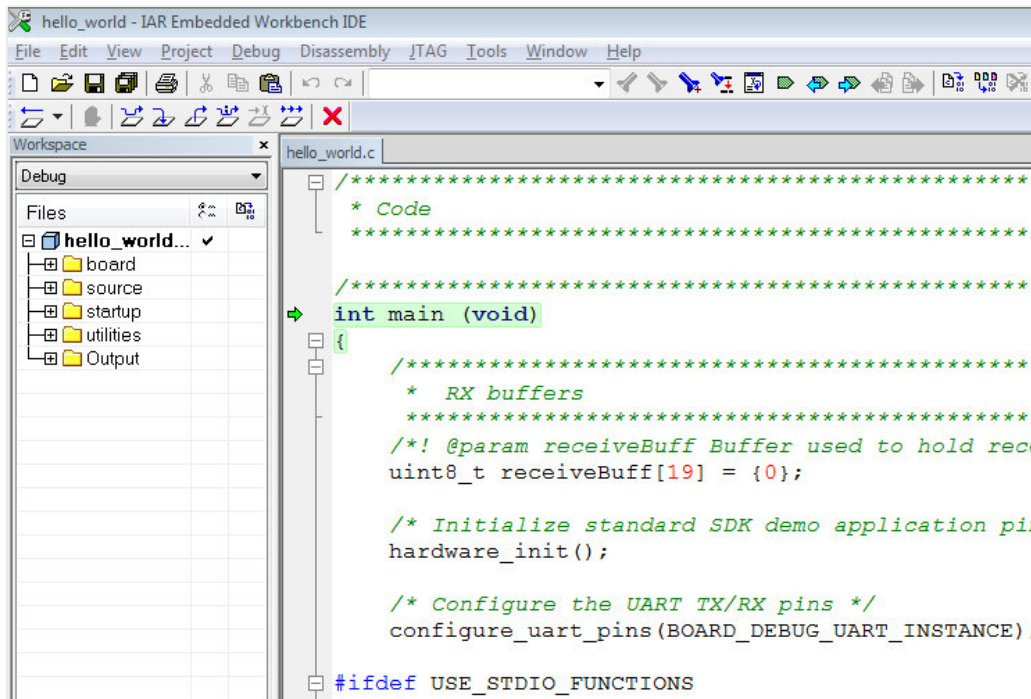


图8 debug时停在main()处

6. 点击“Go”按钮，程序继续运行



图9 Go按钮

7. Hello\_world程序在终端上显示如下信息，如果没有看到如下的信息，请检查串口终端设置和串口连接。

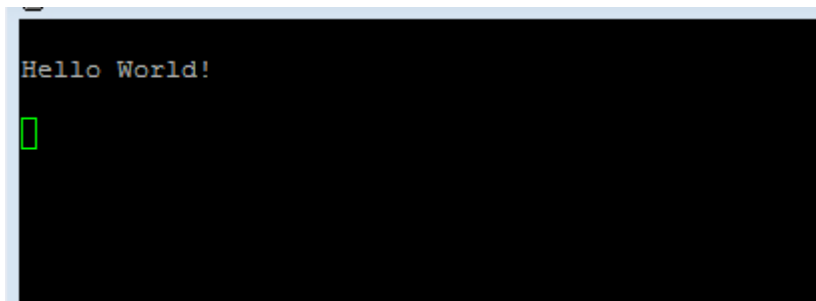


图10 Hello\_world程序运行结果

## 3.2 KEIL uVision IDE

### 3.2.1 构建演示程序

演示程序工作区文件位于：

<install\_dir>/demos/<demo\_name>/<compiler>/<board\_name>/<demo\_name>.uvmpw



以下以hello\_world演示程序为例，Keil工作区文件位于：

<install\_dir>/demos/hello\_world/uv4/maps64/hello\_world.uvmpw

由于演示程序需要使用MAPS的设备驱动库来调用HAL和外设驱动，在构建演示程序之前，需要先构建设备驱动库ksdk\_platform\_lib.lib，设备驱动库的工程文件已经添加进了工作区文件中。

打开演示程序的Keil工作区文件，选择平台驱动库ksdk\_platform\_lib，然后点击“Build”按钮：

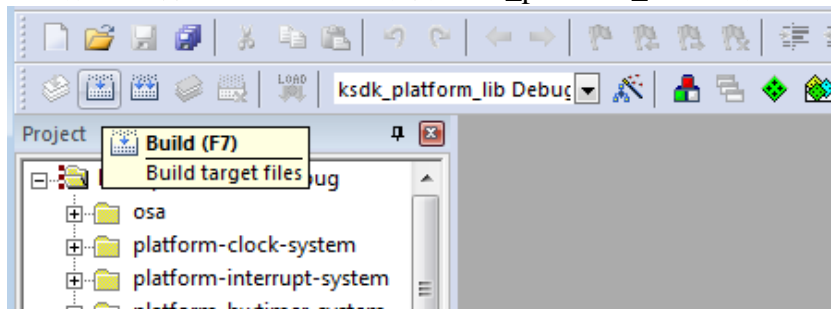


图11 构建平台驱动库

构建完成后，ksdk\_platform\_lib.lib就生成在相应的文件夹中。

然后选择演示程序工程，点击”build”按钮来构建演示程序

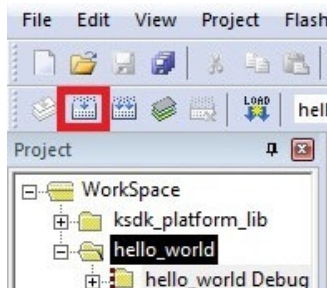


图12 构建演示程序

或者点击”rebuild”按钮重建演示程序

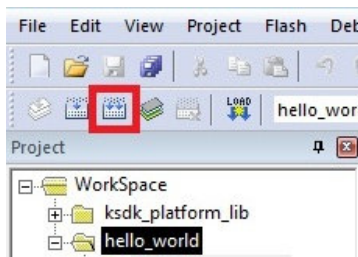


图13 Rebuild演示程序

构建完成后，Keil会在构建输出窗口显示如下信息：



```

Build Project 'hello_world' - Target 'hello_world Debug'
compiling fsl_misc_utilities.c...
compiling fsl_debug_console.c...
assembling startup_MK64F12.s...
compiling system_MK64F12.c...
compiling startup.c...
compiling hello_world.c...
compiling gpio_pins.c...
compiling pin_mux.c...
compiling hardware_init.c...
linking...
Program Size: Code=18008 RO-data=1192 RW-data=40 ZI-data=49344
"debug\hello_world.out" - 0 Error(s), 0 Warning(s).

```

图14 构建hello\_world程序成功

### 3.2.2 运行演示程序

1. 串口终端的配置同3.1.2
2. 配置Keil的debugger
  - a. 按快捷键“ALT+F7”或选择“Project > Options for Target <project\_name>,”调出Option窗口。
  - b. 在“Options for Target ...”对话框中,选择Debug页面, 确保simulator没有被选中且选择了适合的debug driver, 然后点击debug driver旁的“Settings”按钮, 配置debugger信息。

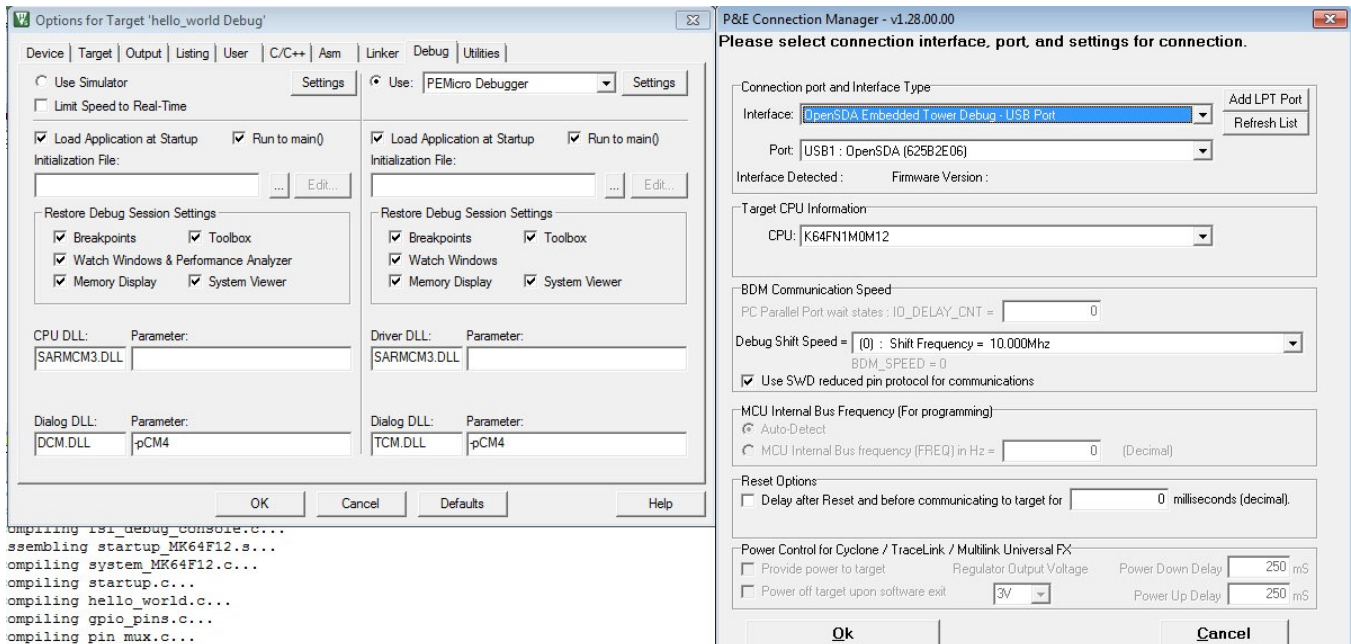


图15 MAPS-K64 Keil的debugger配置

3. 参照 3.1.2配置串口终端.
4. 调试器连接成功后, 点击 “Start/Stop Debug Session”按钮或者按快捷键CTRL+F5进入调试.



图16 开始调试

5. 程序下载成功后, Keil的调试器停在main()函数的开始处.

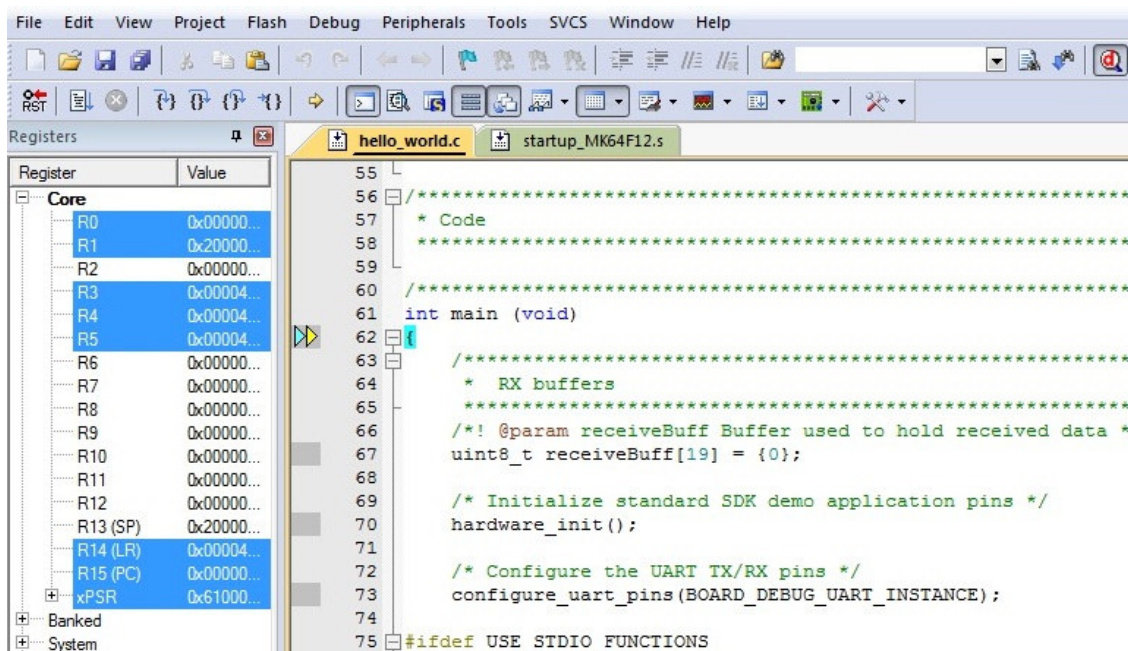


图17 开始调试时停在main()函数的开始处

6. 点击“Run”按钮或者按F5继续运行程序

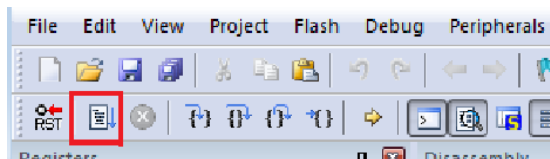


图18 运行按钮

7. 在Putty终端就可以看到hello\_world程序的运行了。如果没有, 请检查串口终端设置和串口连接。

#### How to Reach Us:

##### Home Page:

[freescale.com](http://freescale.com)

##### Web Support:

[freescale.com/support](http://freescale.com/support)

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions).

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Tower is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The ARM Powered Logo is a trademark of ARM Limited. ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2014 Freescale Semiconductor, Inc.

