FRONTGRADE

APPLICATION NOTE

UT32M0R500

Creating a Project in the Keil IDE UT32M0R500
32-bit Arm™ Cortex® M0+ Microcontroller V6

7/24/2025
Version #: 2.0.0

Table 1: Cross Reference of Applicable Products

Product Name	Manufacturer Part Number	SMD #	Device Type	Internal Pic Number
Arm Cortex M0+	UT32M0R500	5962-17212	Project Setup	QS30

1.0 Overview

This document details the process of creating a **UT32M0R500**-based embedded software project using the **Keil ARM Compiler 6**, which uses LLVM compiler. LLVM compiler produces efficient code generation, better diagnostics, and faster development. The ARM Compiler 6 is partially compatible with previous ARM Compiler Versions (5 and before), so this application note guides how to port your application source code, compiler settings and make best use of new compiler optimizations and diagnosis facilities. For the purposes of this document, we will create a project named **helloworld** and configure the **Keil** tools to include all the source modules required for a successful build. Using this template, the user should be able to create projects using (a) their preferred application source directory structures and (b) the directory structure for the **Keil**-supplied files.

2.0 Creating a design project with Keil uVision IDE

1. Submit request to download **SDK_2_0_0.zip** from https://www.frontgrade.com/product/ut32m0r500-0 under Software Download Request form. Once the download has completed, unzip the files. Create a directory of your choice for the **helloworld** project.
2. Launch Keil uVision 
3. From the Project menu, select New uVision Project….
4. Under the directory of choice, specify the project name as **helloworld** and click **Save,** see Figure 1.

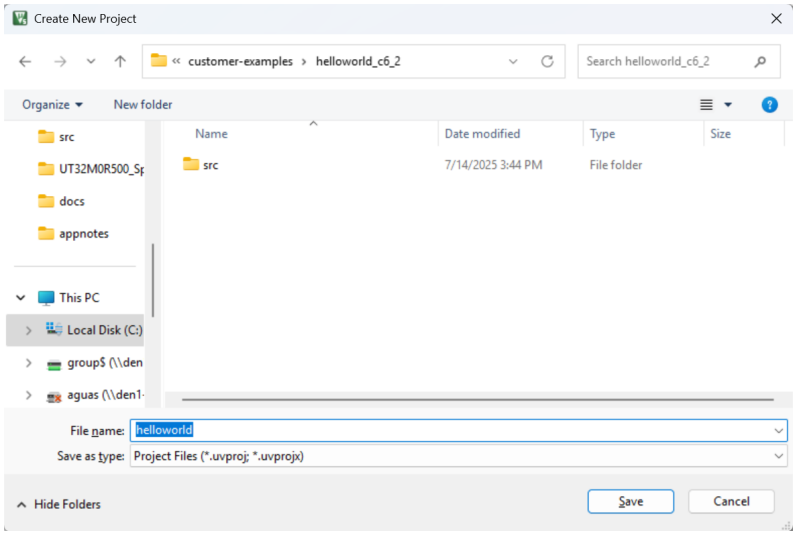


Figure 1: Project Setup

5. Select **Device** and click **OK,** see Figure 2.

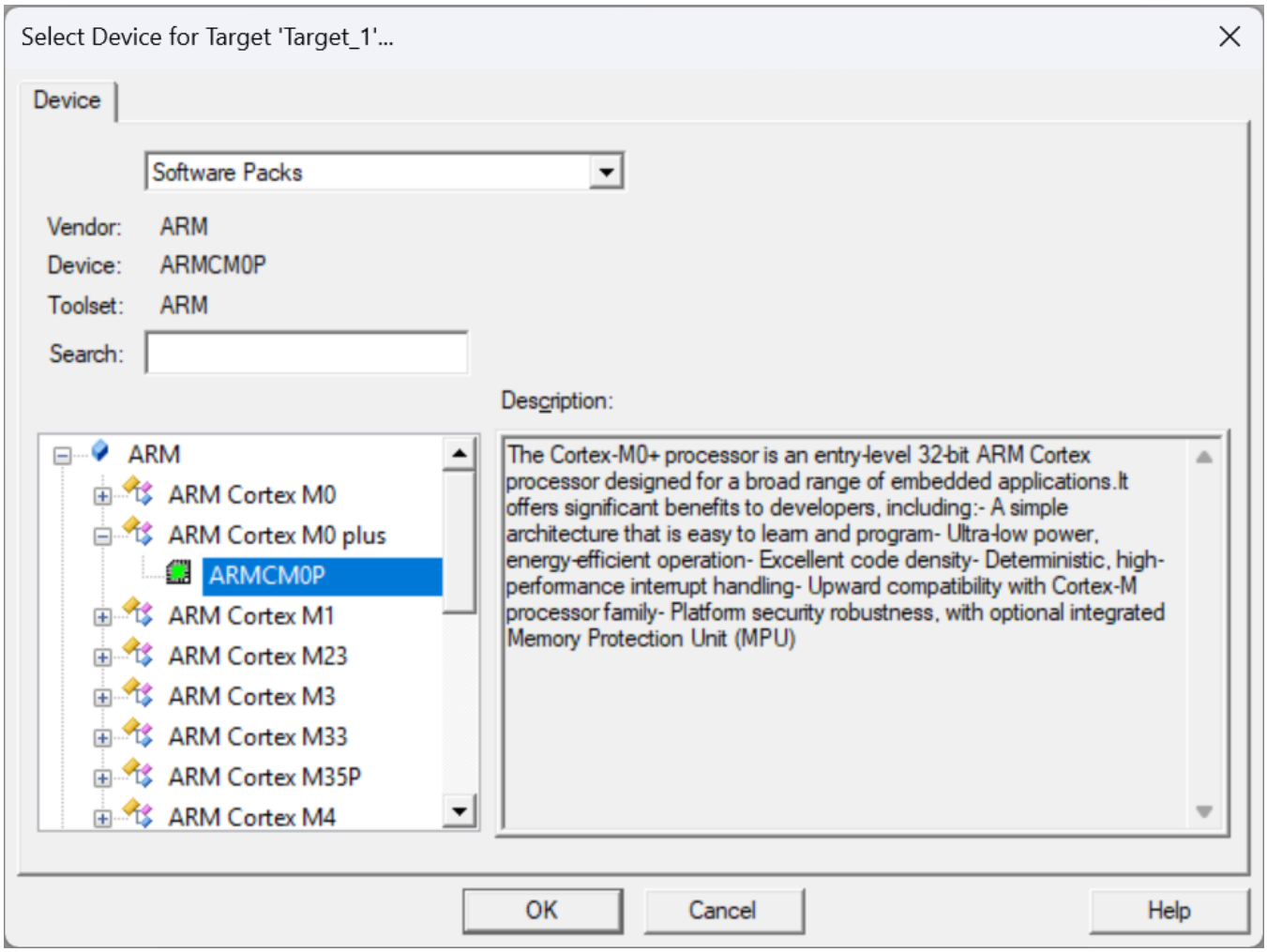


Figure 2: Select Device

6. Click the **Manage Run-Time Environment** symbol  and under **Software Component,** select the appropriate components and click **OK,** see Figure 3.

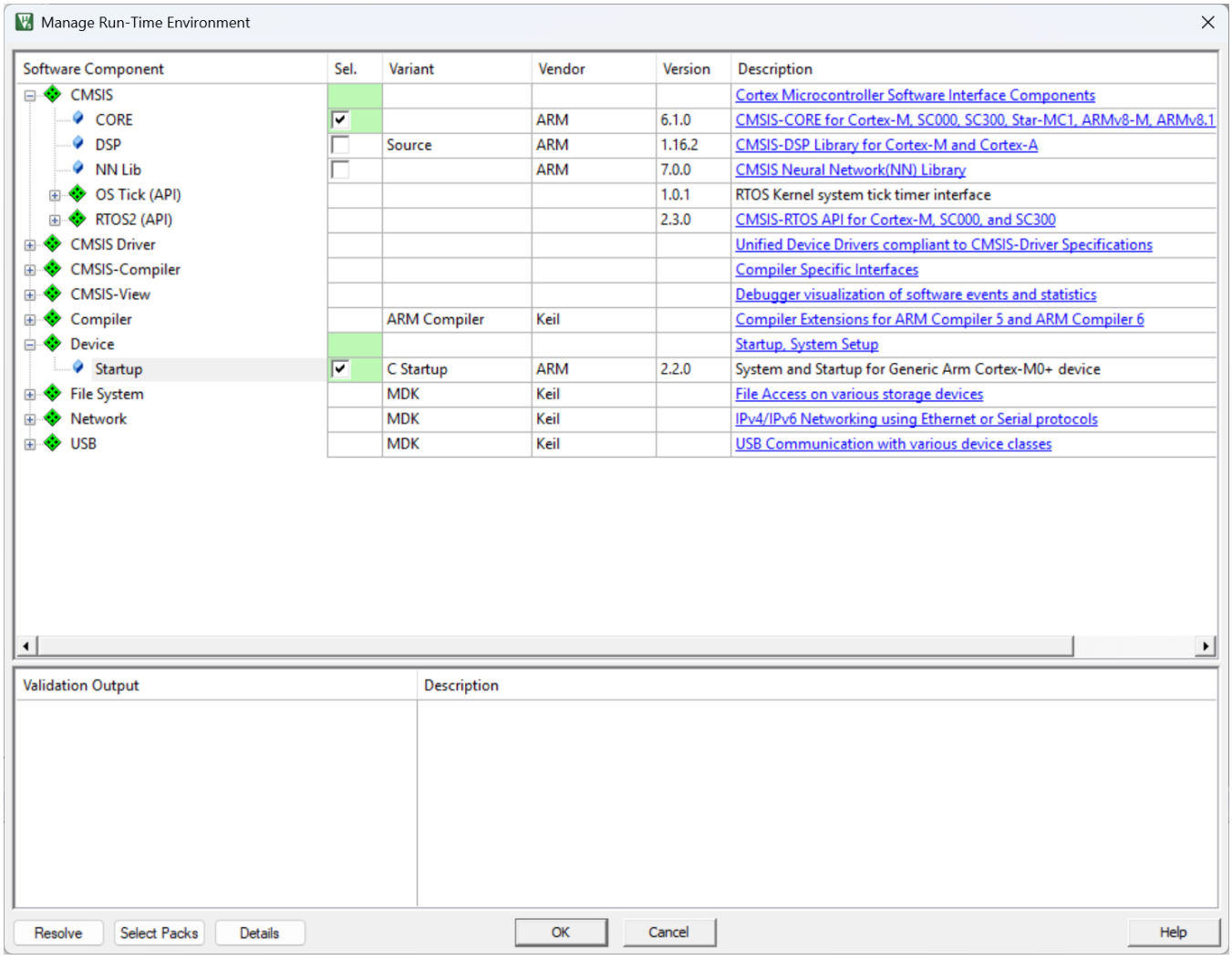
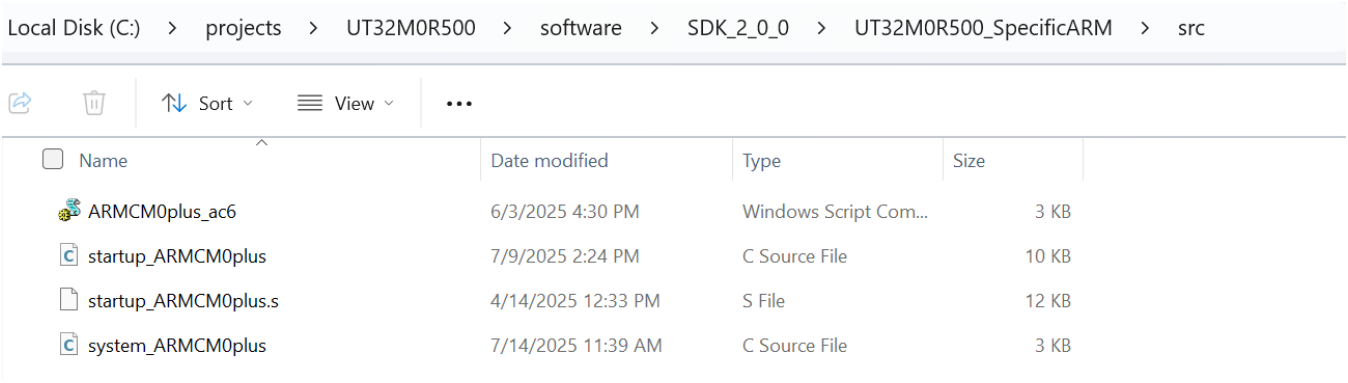


Figure 3: Software Components

7. Under the folder where the project was created, browse to **RTE\Device\ARMCM0P** and replace **startup_ARMCM0plus.c**, **system_ARMCM0plus.c**, and **ARMCM0plus_ac6.sct** with the files from **SDK_2_0_0\UT32M0SpecificARM\src\.**

NOTE: Files under SDK_2_0_0\UT32M0SpecificARM\src are specific startup files for Frontgrade' UT32M0R500.



8. Under the folder where the project was created, create a **src** folder for the **.c** files. In the **Project,** double-click **Source Group 1** and rename it to **hello_src**.
9. Right-click on **hello_src** and click on **Add New Item to Group 'hello_src'**…. Add a new **C** source file, **hello_test.c** and copy the source code from Code 1.

```
#include <stdio.h>
#include "UT32M0R500.h"
#include "ut32m0_uart.h"

UART_TypeDef *UART0 = (UART_TypeDef *) UART0_BASE;
UART_InitTypeDef UART_InitStruct;
uint32_t ActualBaudRate;

int main (void){
        UART_StructInit (&UART_InitStruct);
        ActualBaudRate=UART_Init (UART0, &UART_InitStruct);
        UART_Cmd (UART0, ENABLE, ENABLE);

        for(;;){
                printf("Hello World!!!\r\n");
        }
}
```

Code 1: Hello World Source Code

10. Right-click on **Target1** and select **Add Group…** to create groups for source and include files from Frontgrade's Standard Peripheral Library, <your working dir>\StdPeriphLib\src and <your working dir>\StdPeriphLib\inc. Add sources and include files to their respective directories, see Figure 4.

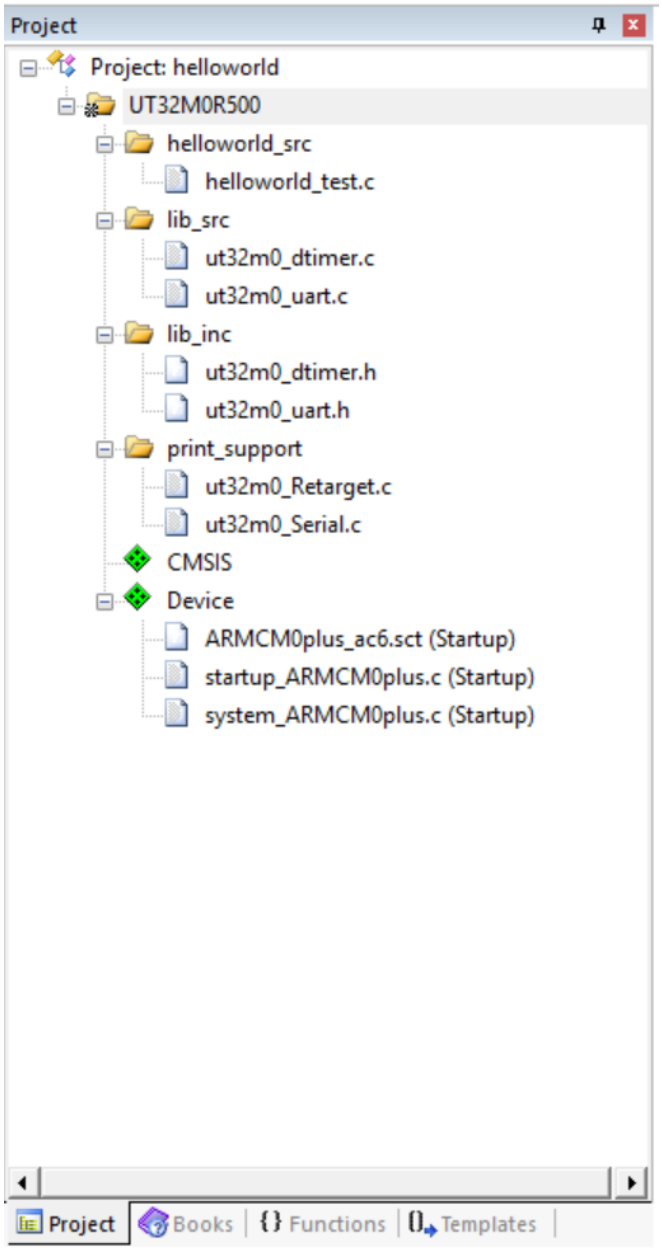


Figure 4: Add source and include files

11. Right-click on **Target1** and select **Options for Target 'Target 1'**…. see Figure 6-11 for basic settings—Change settings according to the particular project. For **C/C++** and **Asm** tabs, click ... and setup the compiler include paths; see Figure 6 and Figure 7.

    NOTE: ARM Compiler version 6 uses the scatter file for memory configurations.

    NOTE: the **System Viewer File** path in Figure 5 is:

    <your working dir>\UT32M0R500_SpecificARM\SVD\Wolverine_BasiCAN.SFR

    If your project requires the use of **PeliCAN,** set the path to:

    <your working dir>\UT32M0R500_SpecificARM\SVD\Wolverine_PeliCAN.SFR

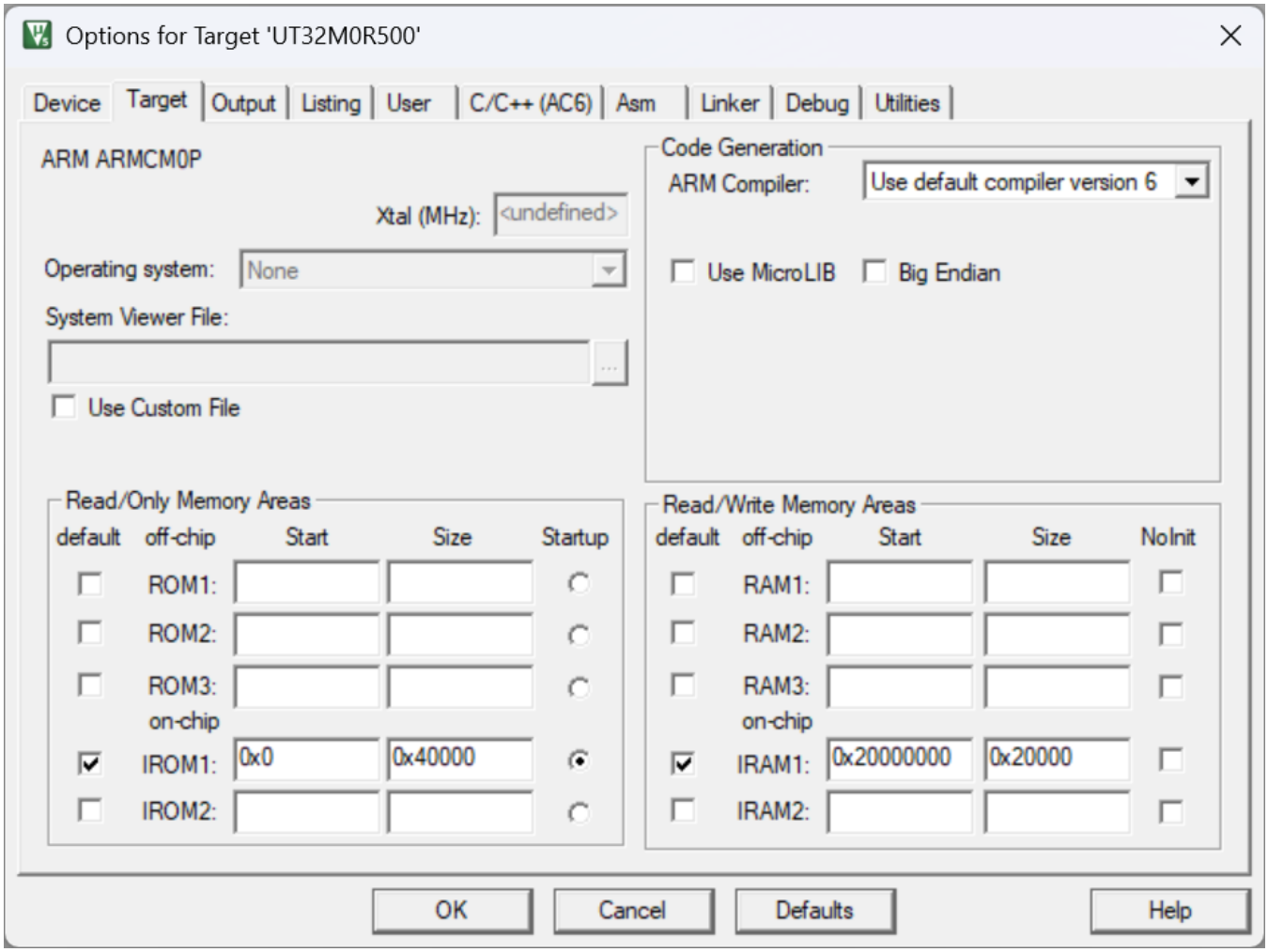


Figure 5: Target

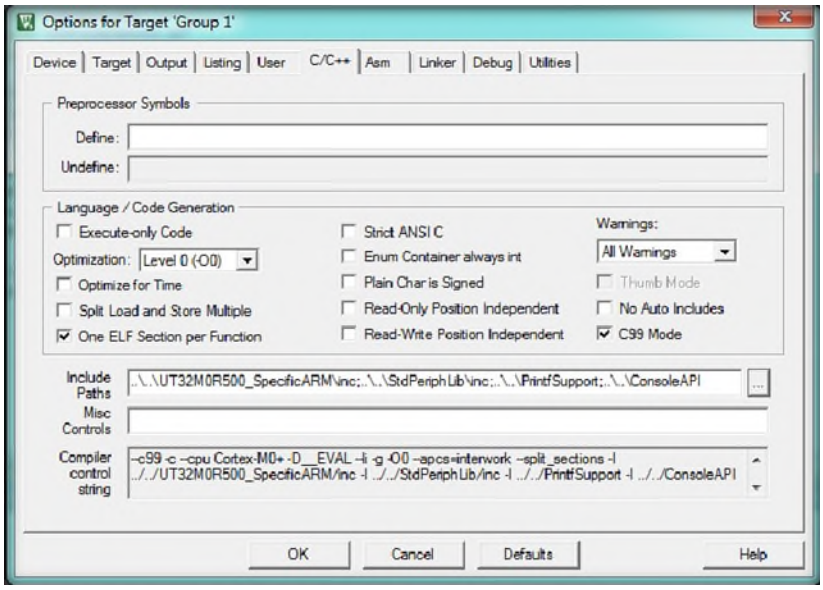


Figure 6: C/C++ Include Paths

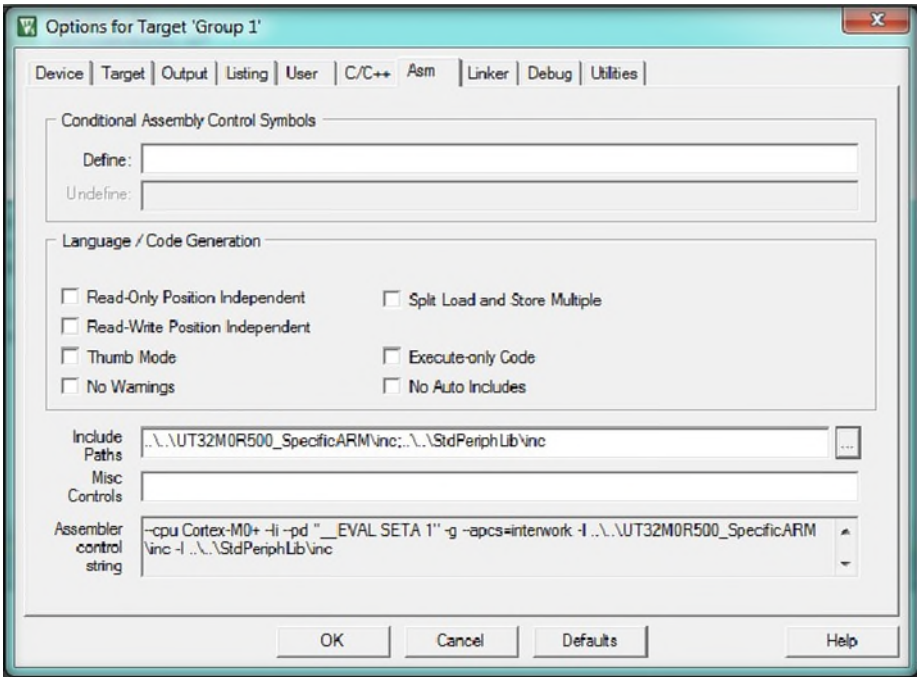


Figure 7: ASM Include Paths

NOTE: the scatter file is now used for Memory Layout, see picture below.

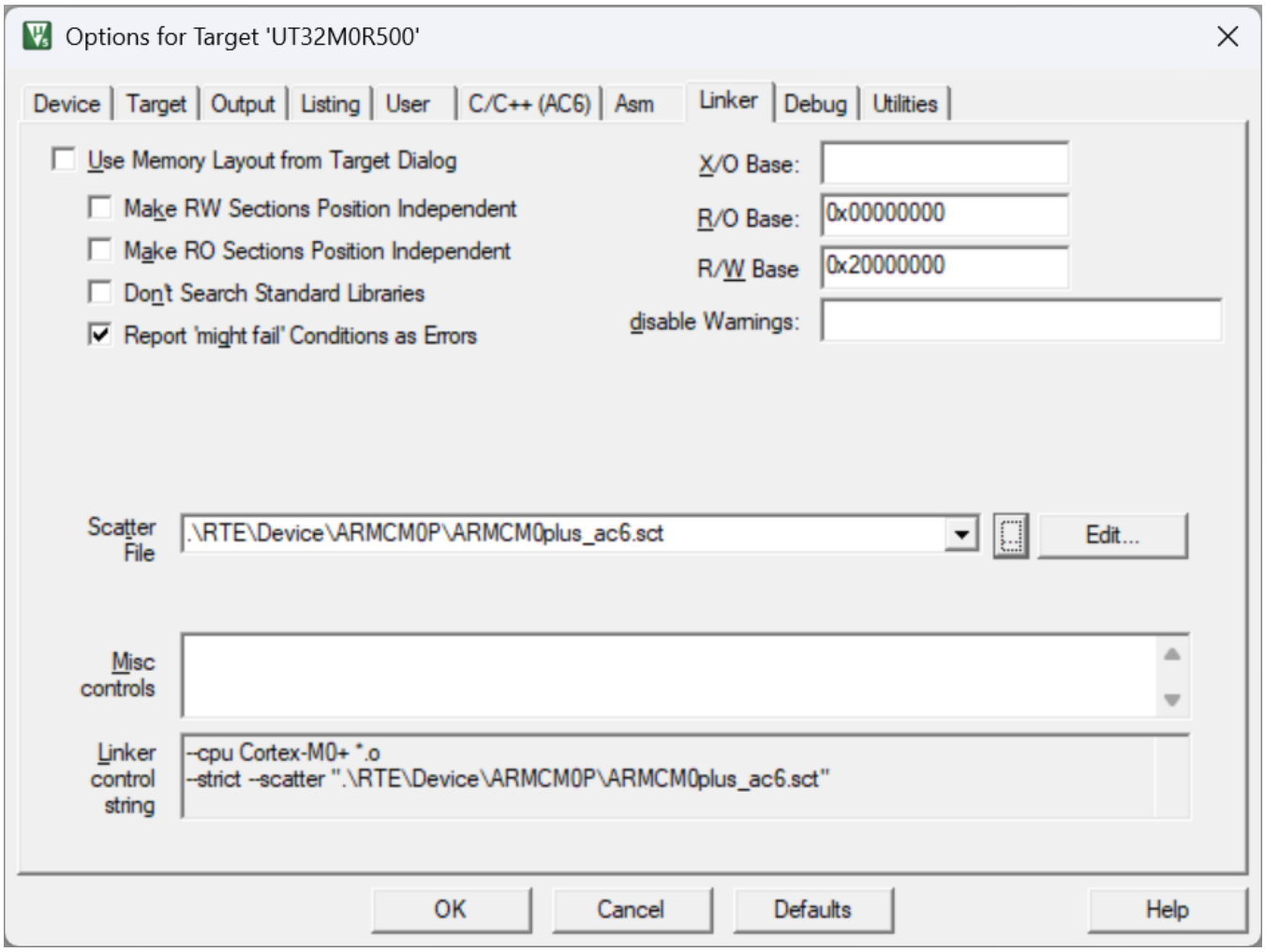


Figure 8: Linker

NOTE: the Initialization File path is: <your working dir>\UT32M0R500_SpecificARM\Wolv_SRAM_Debug.ini

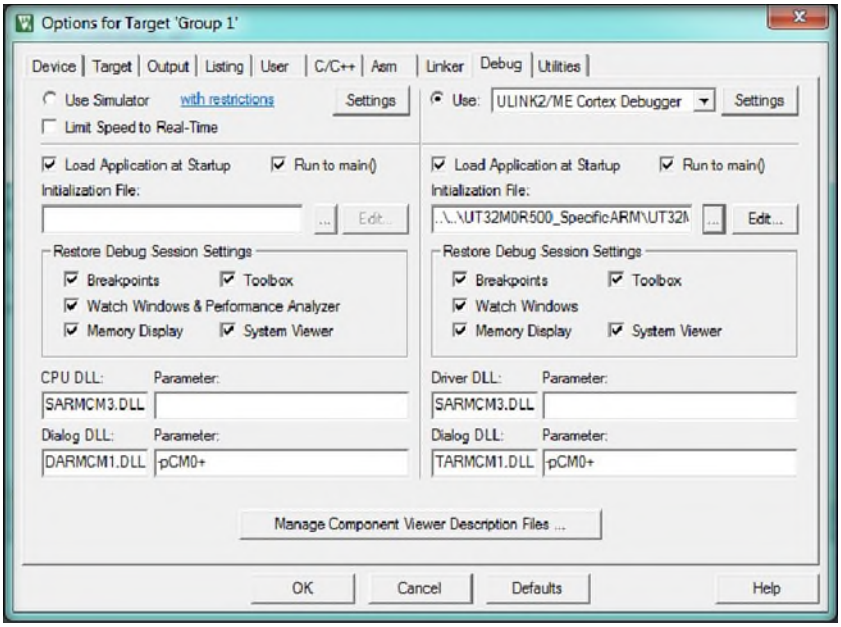


Figure 9: Debugger

NOTE: the serial for your **JTAG** pod will appear in the **Serial No:** box.

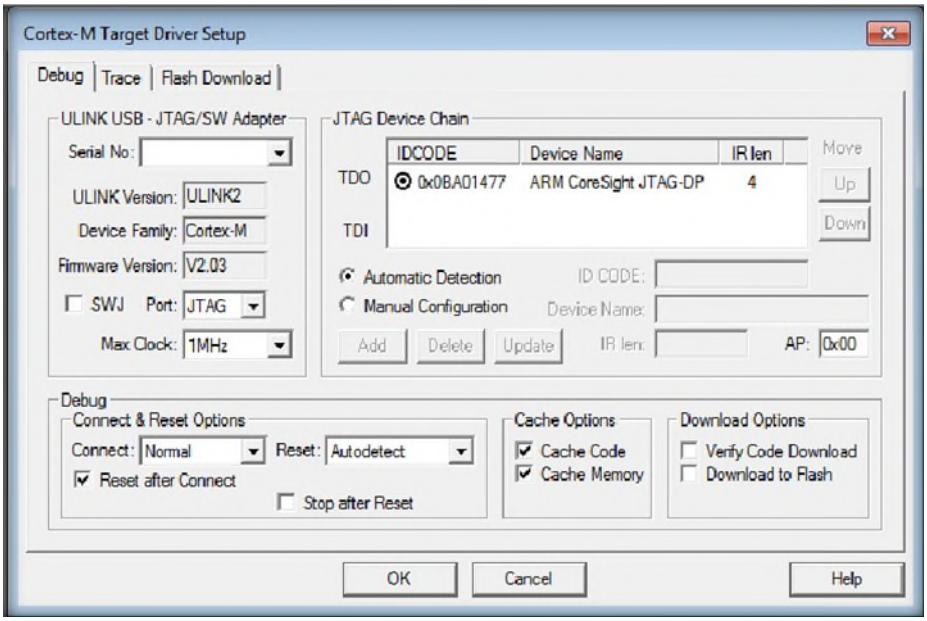


Figure 10: Debugger Settings

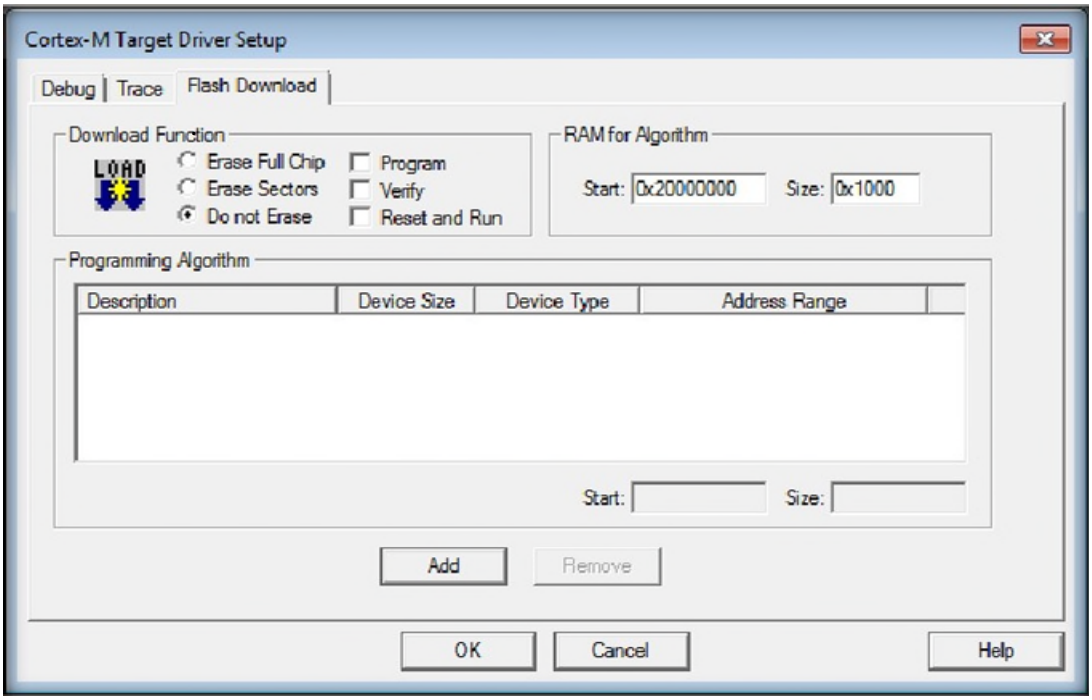


Figure 11: Flash Download

12. In the Project Explorer view, click on  and Build Project.
13. Start the debugger  and run  the application. Display the output using your favorite Terminal, see Figure 12.

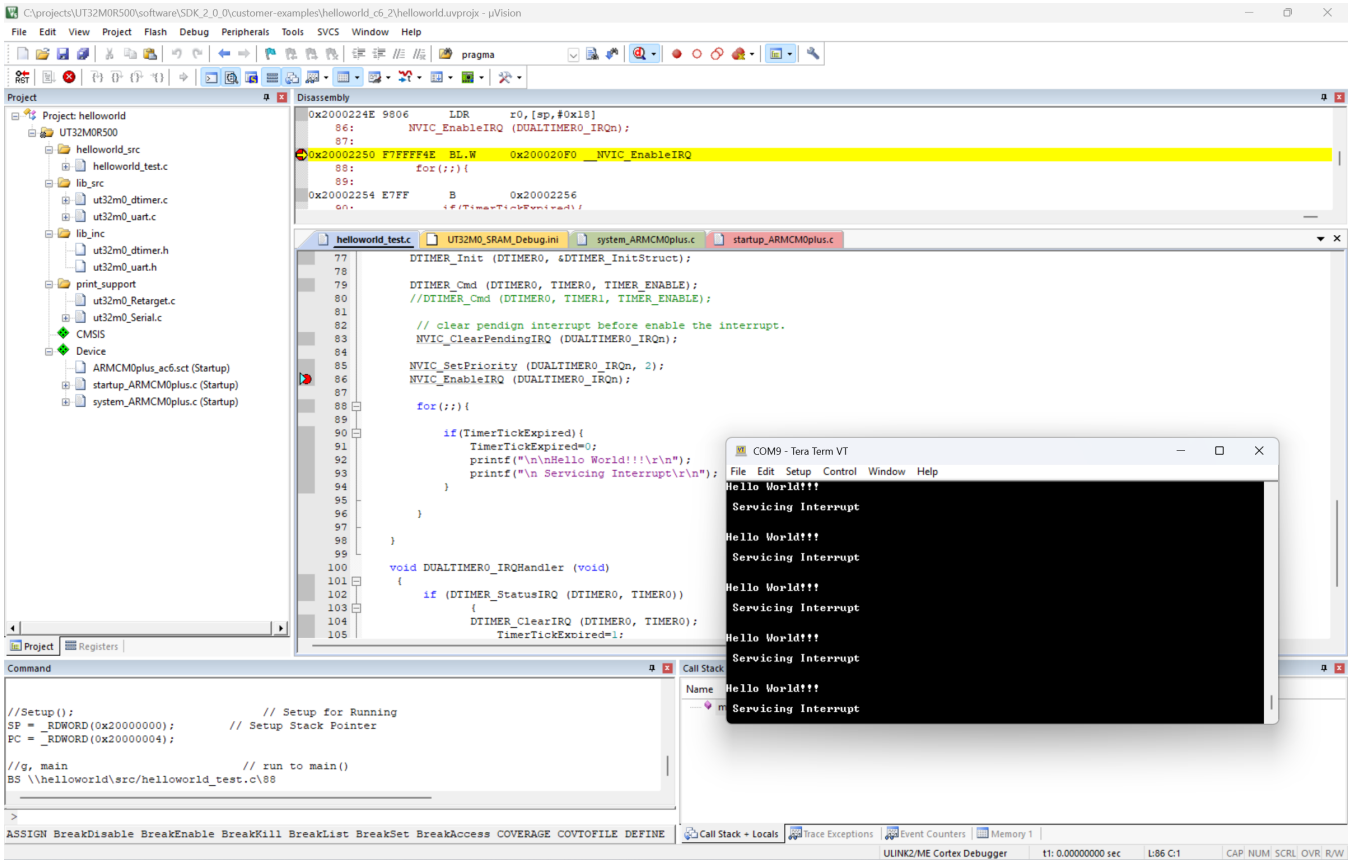


Figure 12: Hello World Display

3.0 Revision History

Date	Revision #	Author	Change Description	Page #
5/17	1.0.0	SW	Initial Release	
12/17	1.0.1	AW	Minor edits for directory names	
2/18	1.0.2	AW	Additional edits for directory names and dialog settings box	
8/15/18	1.1.0	JA	Second release	
7/24/2025	2.0.0	JA	Added instructions for creating a project for ARM Keil Compiler V6	

Datasheet Definitions

	Definition
Advanced Datasheet	Frontgrade reserves the right to make changes to any products and services described herein at any time without notice. The product is still in the development stage and the **datasheet is subject to change**. Specifications can be **TBD** and the part package and pinout are **not final.**
Preliminary Datasheet	Frontgrade reserves the right to make changes to any products and services described herein at any time without notice. The product is in the characterization stage and prototypes are available.
Datasheet	Product is in production and any changes to the product and services described herein will follow a formal customer notification process for form, fit or function changes.