### **APPLICATION NOTE**

### UT32M0R500

Creating a Project in the Keil IDE UT32M0R500 32-bit Arm<sup>™</sup> Cortex<sup>®</sup> M0+ Microcontroller

> 8/15/2018 Version #: 1.1.0

Version #: 1.1.0

8/15/2018

#### **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	Q\$30

#### **1.0 Overview**

This document details the process of creating a **UT32MOR500**-based embedded software project using the **Keil ARM** development tools. 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. Download **UT32M0R500\_API\_vx\_x\_zip** from <u>www.frontgrade.com/hirel.</u> 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.

	v Examples v nellowor		0 @
Organize  New folder	24	~	:== • 🔮
✓ □ Libraries ▷ □ Documents ▷ ▲ Music	Name	No items match	Date modified
<ul> <li>Pictures</li> <li>Videos</li> </ul>			
📲 Homegroup	E		
🖻 💒 Local Disk			
~	÷ 4	III	
File name: helloworld			-
Save as type: Project Files (*.uvproj	; *.uvprojx)		

Figure 1: Project Setup



Version #: 1.1.0

8/15/2018

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

evice				
	Software Packs		•	
Vendor:	ARM			
Device:	ARMCMOP			
Toolset:	ARM			
Search:		_		
	,		Description:	
	ARM ARM Cortex M0 ARM Cortex M0 plus ARMCM0P	-	The Cortex-M0+ processor is an entry-level 32-bit ARM Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including: - simple, easy-to-use programmers model - highly efficient ultra-low power operation - excellent code density.	*
	ARM Cortex M23		- deterministic, high-performance interrupt handling	
÷	ARM Cortex M3		- upward compatibility with the rest of the Cortex-M processor family.	
÷.	ARM Cortex M33			
±.	ARM Cortex M4			
+	🕸 ARM Cortex M7			
1000	ARM SCOOD	-		

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.

CMSIS       Cortex Microcontroller Software Interface Components         CORE       ✓       S.0.1       CMSIS-CORE for Cortex-M. SC000, SC300, ARM/8-M         DSP       □       1.5.1       CMSIS-CORE for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       1.5.1       CMSIS-SPE Library for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       CMSIS-RTOS API for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       CMSIS-RTOS API for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       CMSIS-RTOS API for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       CMSIS-RTOS API for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       CMSIS-RTOS API for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       CMSIS-RTOS API for Cortex-M. SC000, and SC300         Image: RTOS (API)       □       2.1.0       Complex for Data Schange         Image: RTOS (API)       □       2.0.0       Complex for Data Schange         Image: RTOS (API)       □       1.0.1       System Components for Data Schange         Image: RTOS (API)       Image: RTOS (API)       Image: RTOS (API)       Image: RTOS         Image: RTOS (API)	omponent S	el. Variant		Version	Description	
• CORE           • CORE           • S0.1         CMSIS-CORE for Cortex-M. SC000, SC300, ARM/6-M             • DSP           1.5.1         CMSIS-CORE for Cortex-M. SC000, and SC300             • RTOS (API)           1.0.0         CMSIS-RTOS API for Cortex-M. SC000, and SC300             • RTOS (API)           21.0         CMSIS-RTOS API for Cortex-M. SC000, and SC300             • RTOS (API)           21.0         CMSIS-RTOS API for Cortex-M. SC000, and SC300             • CMSIS Driver           21.0         CMSIS-RTOS API for Cortex-M. SC000, and SC300             • CMSIS Driver           CMSIS-RTOS API for Cortex-M. SC000, and SC300             • CMSIS Driver           CMSIS-RTOS API for Cortex-M. SC000, and SC300             • CMSIS Driver           CMSIS-RTOS API for Cortex-M. SC000, and SC300             • Compiler           ARM Compiler           Compiler Extensions for ARM Compiler 5 and ARM Compiler 5             • Data Exchange           Software Components for Data Exchange           Software Components for Data Exchange             • Derice           Startup           Software Components for Data Exchange             • Stantup           Software Software Component	SIS				Cortex Microcontroller Software Interface Components	
	CORE			5.0.1	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M	
Image: Provide and Provided Provid	DSP			1.5.1	CMSIS-DSP Library for Cortex-M, SC000, and SC300	
B ← RTOS2 (API)     CMSIS-RTOS API for Cortex-M, SC000, and SC300     Unified Device Drivers compliant to CMSIS-Driver Specifications     Compiler     ARM Compiler     ARM Compiler     12.0     Complere Extensions for ARM Compiler 5 and ARM Compiler 6     Software Components for Data Exchange     Device     Startup     Startup     Startup     V     10.1     System and Startup for Generic ARM Cortex-M0+ device     Startup     Software Components for Data Exchange     Software Components for Data Exchange     Startup     Startup     Startup     V     10.1     System and Startup for Generic ARM Cortex-M0+ device     Software Component     Vogitach ffRSTL Safety Software Component     Graphics     NDK-Pro     S306     User Interface on graphical LCD displays     Nork     NWP     1.4.1     Network Midleware     Oryx Embedded Mii.     1.72     Mideleware package(CycloneTCP, CycloneSSL and CycloneCrypto)     Nicrium     10.0     Micrium Real Time Kernel	RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
♦ CMSIS Driver         Unified Device Drivers compliant to CMSIS-Driver Specifications           ♦ Compiler         ARM Compiler         12.0         Compiler Extensions for ARM Compiler 5 and ARM Compiler 6           ♦ Data Exchange         Software Components for Data Exchange         Software Components for Data Exchange           ● Device         Startup         Interventional Startup For Generic ARM Contex-M0+ device           ● Istartup         ✓         1.0.1         System and Startup For Generic ARM Contex-M0+ device           ● File System         MDK-Pro         6.9.8         File Access on various storage devices           ● File System         MDK-Pro         5.36.6         User Interface on graphical LCD displays           ● Network         IMP         1.4.1         Network Indeleware Budded Mini         1.7.2           ● Ony: Embedded Middleware         Ony: Embedded Mini         1.0.0         Micrium Real Time Kernel	RTOS2 (API)			2.1.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
	SIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications	
◆ Data Exchange     Image: Software Components for Data Exchange       ◆ Device     Startup. System Setup       ◆ Startup     Image: Display Startup       ◆ File System     MDK-Pro       ◆ File System     69.8       ● Graphics     MDK-Pro       ◆ Graphics     MDK-Pro       ◆ Startup     536.6       User Interface on graphical LCD displays       ◆ Network     IwIP       ◆ Ony: Embedded Middleware     Ony: Embedded Middleware       ◆ Startup     1.2       ◆ RTOS     Micrium       1.0.0     Micrium Real Time Kernel	npiler	ARM Compile	er	1.2.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6	
◆ Device     Startup. System Setup       ✓ Stortup     1.0.1     System and Startup for Generic ARM Cortex-M0+ device       ◆ File System     MDK-Pro     6.9.8     File Access on various storage devices       ◆ Functional Safety     Yogitech fRS1L Safety Software Component       ◆ Rework     MDK-Pro     5.36.6     User Interface on graphical LCD displays       ◆ Network     WIP     1.4.1     Network MIdleware package(CycloneTCP, CycloneSSL and CycloneCrypto)       ◆ RTOS     Micrium     1.0.0     Micrium Real Time Kernel	a Exchange	and the state of the state of the			Software Components for Data Exchange	
Startup     10.1     System and Startup for Generic ARM Cortex-M0+ device       File System     MDK-Pro     59.8     File Access on various storage devices       Functional Safety     Yogitech RSTL Safety Software Component       Graphics     MDK-Pro     536.6     User Interface on graphical LCD displays       Network     IvIP     14.1     Network Indeleware package(CycloneTCP, CycloneSSL and CycloneCrypto)       RTOS     Micrium     1.0.0     Micrium Real Time Kernel	ice				Startup, System Setup	
◆ File System         MDK-Pro         ► 69.8         File Access on various storage devices           ◆ Functional Safety         Yogitech fRSTL Safety Software Component           ◆ Graphics         MDK-Pro         ► 536.6         User Interface on graphical LCD displays           ◆ Network         IMIP         1.4.1         Network IwIP Bundle           ◆ Oryx Embedded Middleware         Oryx Embedded Mi         1.7.2         Middleware package(CycloneTCP, CycloneSSL and CycloneCrypto)           ◆ RTOS         Micrium         1.0.0         Micrium Real Time Kernel	Startup 🔽	i i i i i i i i i i i i i i i i i i i		1.0.1	System and Startup for Generic ARM Cortex-M0+ device	
	System	MDK-Pro	•	6.9.8	File Access on various storage devices	
Graphics     MDK-Pro     536.6     User Interface on graphical LCD displays       Network     IMIP     1.4.1     Network IwIP Bundle       Oryx Embedded Middleware     Oryx Embedded Mi     1.7.2     Middleware package(CycloneTCP, CycloneSSL and CycloneCrypto)       RTOS     Micrium     1.0.0     Micrium Real Time Kernel	ctional Safety	and the balance of a			Yogitech fRSTL Safety Software Component	
Network     Network     Network     Network     Oryx Embedded Middleware     Oryx Embedded Mi     17.2     Middleware package(CycloneTCP, CycloneSSL and CycloneCrypto)     RTOS     Micrium     10.0     Micrium Real Time Kernel	phics	MDK-Pro	•	5.36.6	User Interface on graphical LCD displays	
Oryx Embedded Middleware     Oryx Embedded Mi     17.2     Middleware package(CycloneTCP, CycloneSSL and CycloneCrypto)     RTOS     Micrium     10.0     Micrium Real Time Kernel	work	IwIP	-	1.4.1	Network IwIP Bundle	
RTOS         Micrium         1.0.0         Micrium Real Time Kernel           Security	x Embedded Middleware	Oryx Embedd	ed Mi	1.7.2	Middleware package(CycloneTCP, CycloneSSL and CycloneCrypto)	
🔹 Security	)S	Micrium		1.0.0	Micrium Real Time Kernel	
	urity					
VSB MDK-Pro 🔽 6.11.0 USB Communication with various device classes	1	MDK-Pro		6.11.0	USB Communication with various device classes	
} ♦ mbed	ed					

Figure 3: Software Components

Version #: 1.1.0

8/15/2018

- Under the folder where the project was created, browse to RTE\Device\ARMCMOP and replace startup\_ARMCMOplus.s and system\_ARMCMOplus.c with the files from UT32M0R500\_API\_vx\_x\_x\UT32M0SpecificARM\src\.
   NOTE: Files under UT32M0R500\_API\_vx\_x\_x\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

- 8/15/2018
- 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: 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

#### UT32M0R500 Creating a Project in the Keil IDE UT32M0R500 32-bit Arm<sup>™</sup> Cortex<sup>®</sup> M0+ Microcontroller

Version #: 1.1.0

8/15/2018

			Xtal (MHz):	90.0	Code C	Generation Compiler:	Use default	compiler versi	on _
Operating	g system:	None		-					
System V	iewer File:					se Cross-N	Aodule Optimiza	tion	
C:\proje	cts\UT32	MOR500\UT32	MOR500_Spec	aficARM	U D	se Micro L	IB T	Big Endian	
🔽 Use	Custom Fil	e							
-Read/	Only Memo	ny Areas			-Read/	Write Merr	nory Areas		
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	Nolnit
	ROM1:			C		RAM1:			
	ROM2:			- c		RAM2:			
	ROM3:		í –	- c		RAM3:		<u> </u>	П п
	on-chip					on-chip	1	1	
~	IROM1:	0x20000000	0x16000	e	•	IRAM1:	0x20016000	0x20000	
_	IDOUD		· · · · · ·			IRAM2-			

Figure 5: Target

Preprocessor Symbols		
Define: Undefine:		
Language / Code Generation		
Execute-only Code	Strict ANSI C	Wamings:
Optimization: Level 0 (-00)	Enum Container always int	All Warnings
Optimize for Time	Plain Char is Signed	🗖 Thumb Mode
Split Load and Store Multiple	Read-Only Position Independent	No Auto Includes
One ELF Section per Function	Read-Write Position Independent	C99 Mode
Include	DARMNinc:N.StdPeriphLibNinc:N.PrintfSupp DEVAL -ling -00 -apics=interworksplit_sect DARM/inc -lin//StdPeriphLib/inc -lin//PrintfS	ions -I Support -I/ConsoleAPI

Figure 6: C/C++ Include Paths

8/15/2018

Version #: 1.1.0

Options for Target 'Group 1'		x
Device Target Output Listing User	ar C/C++ Asm Linker Debug Utilities	
Conditional Assembly Control Symbols	S	
Define:		-
Undefine:		
Language / Code Generation		
Read-Only Position Independent     Read-Write Position Independent     Thumb Mode     No Warnings	Split Load and Store Multiple Execute-only Code No Auto Includes	
Include Paths Misc	cficARM\inc;\.\StdPeriphLib\inc	
Assembler control string	"EVAL SETA 1" -gapcs=intenwork -1\.\UT32M0R500_SpecificARM vinc	r
	OK Cancel Defaults He	lp

Figure 7: ASM Include Paths

Options for Target 'Target 1'	×
Device   Target   Output   Listing   User   C/C++   Asm	Linker Debug Utilities
Lise Memory Layout from Target Dialog     Ma <u>ke</u> RW Sections Position Independent     Make RO Sections Position Independent     Dont Search Standard Libraries     Report 'might fail' Conditions as Errors	X/O Base:
Scatter File	Edit
controls Linker control string -cpu Cortex-M0+ *.o -strict -scatter ".\Objects\gpio_test.sct"	*
ОКС	ancel Defaults Help

Figure 8: Linker

NOTE: the Initialization File path is: <your working dir>\UT32M0R500\_SpecificARM\Wolv\_SRAM\_Debug.ini

#### FRONTGRADE **APPLICATION NOTE**

### Creating a Project in the Keil IDE UT32M0R500 32-bit Arm<sup>™</sup> Cortex<sup>®</sup> M0+ Microcontroller

Version #: 1.1.0

Options for Tar	get 'Group 1'		X
Device Target	Output Listing User C/C++ Asm	Linker Debug	Utilities
C Use Simulator	with restrictions Settings of Real-Time	🔍 Use: ULINK	2/ME Cortex Debugger 💌 Settings
Load Applicat	ion at Startup 🔽 Run to main()	Load Application File:	tion at Startup 🔽 Run to main()
Restore Debug	Session Settings ts   Toolbox indows & Performance Analyzer lisplay  System Viewer	Restore Debug	Session Settings ts IV Toolbox indows isplay IV System Viewer
CPU DLL:	Parameter:	Driver DLL:	Parameter:
SARMCM3.DLL		SARMCM3.DLL	
Dialog DLL:	Parameter:	Dialog DLL:	Parameter:
DARMCM1.DLL	-pCM0+	TARMCM1.DLL	-pCM0+
	Manage Component V	iewer Description Fil	es
	OK Ca	ancel Defa	aults Help

Figure 9: Debugger

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

ULINK USB - JTAG/SW Adapter		evice Chain			
Serial No: 🗨		IDCODE	Device Name	IR len	Move
ULINK Version: ULINK2	TDO	Ox0BA01477	ARM CoreSight JTAG-DP	4	Up
Device Family: Cortex-M	TDI				Down
Firmware Version: V2.03	( Auto	omatic Detection	ID CODE:		-
SWJ Port: JTAG 👻	C Mar	nual Configuration	Device Name:		
Max Clock: 1MHz	Add	Delete	odate IR len:	AF	P: 0x00
- Debug - Connect & Reset Options			Cache Options - Dow	nload Options	s
Connect: Normal Res	et : Autodete	t T	Cache Code	Verify Code D	ownload
Reset after Connect			Cache Memory	Download to F	Flash
	Stop after Be	set			

Figure 10: Debugger Settings

8/15/2018

#### UT32M0R500 Creating a Project in the Keil IDE UT32M0R500 32-bit Arm<sup>™</sup> Cortex<sup>®</sup> M0+ Microcontroller

8/15/2018

Version #: 1.1.0

Cortex-M Target Driver Setup Debug Trace Flash Download					<b>.</b>
Download Function C Erase Full Chip C Erase Sectors C Do not Erase Programming Agorithm	Program     Verify     Reset and Ru	RAM for A Start: 0	Ngorithm x20000000	Size: Ox1000	
Description	Device Size	Device Type	Adidre	ess Range	
		Start:		Size:	
	Add	Remove			
	ОК	Cance	8		Help

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

Version #: 1.1.0

8/15/2018

#### **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	

#### **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 <b>datasheet is subject to change</b> . Specifications can be <b>TBD</b> and the part package and pinout are <b>not final</b> .
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.

**Frontgrade Technologies Proprietary Information** Frontgrade Technologies (Frontgrade or Company) reserves the right to make changes to any products and services described herein at any time without notice. Consult a Frontgrade sales representative to verify that the information contained herein is current before using the product described herein. Frontgrade does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by the Company; nor does the purchase, lease, or use of a product or service convey a license to any patents, rights, copyrights, trademark rights, or any other intellectual property rights of the Company or any third party.