## Start A New Project with Keil MDK-ARM Version 5 for STM32F030

This tutorial is intended for starting a new project to develop software with STM32FG030R8 Nucleo-64 board using Keil Microcontroller Development Kit for ARM (MDK-ARM) version 5.34.



## Create a New Project in Keil MDK v5.34

- 1. Launch Keil uVision IDE by double clicking on the icon
- 2. From the menu, select Project>New uVision Project...

<u>E</u> dit <u>V</u> iew	Proj	ect Fl <u>a</u> sh <u>D</u> ebug Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp	
💕 🗟 🕬	<	New µ <u>V</u> ision Project	🎽 الم 🖉 🖉 الم 🖉 الم
		New Multi-Project Workspace	
d		Open Project	
		Close Project	
		Import	•
		Export	•
		Manage	•
		Select Device for Target	
		Remo <u>v</u> e Item	
	ŝ	Options Alt+F7	
		Clean <u>T</u> argets	
		Build Target F7	·
		<u>R</u> ebuild all target files	
	4	Batch Build	
		Batch Setup	
	٨	Tr <u>a</u> nslate Ctrl+F7	·
		Stop b <u>u</u> ild	
		$\underline{1} \ C: \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	
roject 😽 Book Output	s   {}	Func  0 <sub>4</sub> Temp	
			>

3. In the "Create New Project" dialog box, click "New folder". Browse to a folder location where you would like to create your project folder.

- → × ↑ 🛱 > T	his PC > Documents >			✓ Č	ocuments
Organize 🗸 New fold	der				
p2_x	Create a new folder.	Date modified	Туре	Size	
screenshots	Adobe	4/28/2021 4:27 PM	File folder		
OneDrive	Custom Office Templates	4/28/2021 4:45 PM	File folder		
- onebine	Downloaded Installations	4/28/2021 9:39 PM	File folder		
🖳 This PC					
> 🧊 3D Objects					
> 📃 Desktop					
> 🖆 Documents					
> 🕂 Downloads					
> 🁌 Music					
> 📰 Pictures					
> 📑 Videos					
> 🏪 Windows (C:)					
> 👝 Data (D:)					
> 👝 Local Disk (E:)					
VI					
File name:					~
Save as type: Proje	ect Files (*.uvproi: *.uvproix)				~
				Cauc	Cancel

4. Type in the folder name, for example, "Blinky", then click the "Open" button. This will create a folder named "Blinky" to hold all the files for the new project. It will also take you inside that folder.

Create New Project						×
· •	This PC → Documents →			ب ق	Search Documents	
Organize 👻 New f	older					•
p2_x	^ Name	Date modified	Туре	Size		
screenshots	Adobe	4/28/2021 4:27 PM	File folder			
	Custom Office Templates	4/28/2021 4:45 PM	File folder			
	Downloaded Installations	4/28/2021 9:39 PM	File folder			
💻 This PC	Blinky	5/25/2021 8:46 PM	File folder			
🇊 3D Objects						
📃 Desktop						
🖶 Documents						
👆 Downloads						
👌 Music						
Pictures						
Videos						
Lindows (C:)						
Data (D:)						
Local Disk (E:)						
- · · ·	~					
File name:						~
Save as type: Pr	niect Files (*.uvproj: *.uvprojx)					~
	·····					
N Hide Folders					Open Cano	el
						>

5. While in the project folder, type "Blinky" in the File name field and click the "Save" button. This creates a project with the project named "Blinky". Although we used the same name for the project folder and the project, they do not have to be identical.

→ ~ ↑ 📙	> This PC > Documer	nts > Blinky			V O Search	Blinky	
rganize 👻 New	folder					💷 🔹 🔮	)
	↑ Name	^	Date modified No items matcl	Type n your search.	Size		
Data (D:) Local Disk (E:)	Blinky						~
Save as <u>t</u> ype: I	Project Files (*.uvproj; *.	uvprojx)			Save	Cancel	<u> </u>

6. The Project Wizard will prompt you to select the device type for the project target. You may drill down the device family tree to find "STM32F030R8". If you installed a large number of supported devices, it may take a while to find the device in the family trees. In that case, type in a substring of the device name in the "Search" field and the matching devices will appear in the window below. Click select the device type then click the "OK" button.

Device Software Packs	<u> </u>	
Vendor: STMicroelectronics Device: STM32F030R8Tx Toolset: ARM Search f030		
•**       \$TM32F030         •**       \$TM32F030C6         •**       \$TM32F030C8         •**       \$TM32F030CC         •**       \$TM32F030F4         •**       \$TM32F030F4         •**       \$TM32F030K6         •**       \$TM32F030R8         •**       \$TM32F030R8         •**       \$TM32F030R8         •**       \$TM32F030R8	Desgription: STMicroelectronics' STM32F0 series delivers 32-bit performance while providing the essentials of the STM32 family, particularly for cost- sensitive applications. STM32 F0 MCUs combine real-time performance, low-power operation, and the advanced architecture and peripherals associated with the STM32 platform available. This series of products is highly competitive in traditional 8-bit and 16-bit markets and eliminates the need to manage different architectures and the associated development overhead. Typical applications include application control and user interfaces, handheld equipment, A/V receivers and digtal TV, PC peripherals, gaming and GPS platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems, video intercoms, and HVACs - SRAM with HW parity checking	<
	OK Cancel Help	_

If the device "STM32F030R8" could not be found, you need to install its Software Pack. Please refer to the Keil uVision IDE installation.

7. Next, the Project Wizard will help you manage the run-time environment. Expand the Software Component selections, check "CMSIS>CORE", select "Standalone" in the pull-down of Device

Software Component	Sel.	Variant		Version	Description	
Board Support		STM32F030-Disco	ve	1.1.0	STMicroelectronics STM32F030-Discovery Board	
					Cortex Microcontroller Software Interface Components	
CORE				5.4.0	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M, ARMv8.1-M	
DSP	$\mathbf{\nabla}$	Source	$\sim$	1.8.0	CMSIS-DSP Library for Cortex-M, SC000, and SC300	
NN Lib				1.3.0	CMSIS-NN Neural Network Library	
🗉 🚸 RTOS (API)				1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
🖽 🚸 RTOS2 (API)				2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
🕬 🚸 CMSIS Driver					Unified Device Drivers compliant to CMSIS-Driver Specifications	
🕬 Compiler		ARM Compiler		1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6	
🛛 💠 Device		Standalone	$\sim$	1.00	All HAL and LL peripheral APIs are selectable as individual components.	
Startup				1.0.0	System Startup for STMicroelectronics	
🗉 🚸 STM32Cube HAL	$\sim$					
🗉 💠 STM32Cube LL						
🗉 💠 File System		MDK-Plus	$\sim$	6.13.8	File Access on various storage devices	
Graphics		MDK-Plus	$\sim$	6.10.8	User Interface on graphical LCD displays	
🛛 💠 Graphics Display						
🗈 💠 Network		MDK-Plus	$\sim$	7.14.0	IPv4 Networking using Ethernet or Serial protocols	
🗄 🚸 USB		MDK-Plus	$\sim$	6.14.1	USB Communication with various device classes	
1						•
alidation Output		Description				

category then check "Device>Startup". Click the "OK" button to close the window.

8. You should see a project with a target created in the Project pane. If you click the + signs to open the target, you would see the startup codes for the specific device that were created in the project.

C:\Users\chens\Documents\Blinky\Blinky.uvprojx - µVision [Non-Commercial Use License] -		×
Edit <u>V</u> iew <u>P</u> roject Fl <u>a</u> sh <u>D</u> ebug Peripherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp		
〕 😂 🛃 😹 ங 🛍   り や   ← ⇒   や な ね ね   卓 幸 非 ル   🖄 🖉 📃 🔍 🔜 🖉 🍭 🔹   ♦ ○ 🔗 🍓	•	2
🗋 New (Ctrl+N) 💦 👔 🗟 🗟 😓 🗇 🏟		
Create an empty document 🛛 🚛 🔟		
* Project: Blinky		
🗄 🚂 Target 1		
- Source Group 1		
startup_strip=20000005 (Startup)		
		_
Project Books   O Functions   O <sub>p</sub> Templates		
Project Books   1) Functions   0., Templates   d Output		4 ×
Project Books   () Functions   (), Templates   d Output		<del>р</del> 🗙
Project Books   O Functions   O <sub>9</sub> Templates   d Output		4 ×
Project Books   O Functions   O <sub>0</sub> Templates   I Output		<u></u>
Project Books   0 Functions   0, Templates   d Output		<u>ф X</u>
Project G Functions   0, Templates   d Output		<u>Ф</u> 🗙
Project GBooks   () Functions   (), Templates   d Output		4 × ∧ ∧ >

Click the "New" button for a new text file in the editing window.

9. Copy the code below into the new text file window. This is a simple program to blink the LED LD2 on the STM32F030 Nucleo-64 board.

```
/* use delay loop, 1 sec on 1 sec off
* default 8 MHz clock
* LD2 connects to PA5
*/
#include "stm32f0xx.h"
void delayMs(int n);
int main(void) {
   GPIOA->MODER \mid = 0 \times 00000400;
                            /* set pin to output mode */
   while(1) {
      GPIOA->BSRR = 0x00000020; /* turn on LED */
      delayMs(1000);
      GPIOA->BSRR = 0x00200000; /* turn off LED */
      delayMs(1000);
   }
}
/* 8 MHz SYSCLK */
void delayMs(int n) {
   int i;
   for (; n > 0; n--)
      for (i = 0; i < 1142; i++);
}
```

10. Click the "Save" button to save the file with the code. By default, the file will be saved in the project folder.



11. Give the file name "Blinky.c" and click the "Save" button. The editor is context-sensitive. Once you make it a C source file, the file content will change the color.



12. Once the source file is saved, it needs to be associated with the project. Right click on "Source Group 1" and select "Add Existing Files to Group 'Source Group 1'...".

Image: CrUSers Chens Documents Blinky, Blinky, Wprojx - µVision [Non-Commercial Use License]       -       L         File       Edit       View       Project       File       Solution       Solution         Image: Ima	x x x
<pre>iile Edit View Project Flash Debug Peripherals Tools SVCS Window Help</pre>	<b>►</b> • <b>▲</b>
○       ○	▼ ×
Set Target 1 Ski & Sk	▼ ×
oject	▼ X
% Project: Blinky       1 /* use delay loop, 1 sec on 1 sec off         2 * default 8 MHz clock         3 * LD2 connects to PA5         • CN       CN         • De       Add New Item to Group 'Source Group 1'         • Add New Item to Group 'Source Group 1'       avHs (int n);         • Remove Group 'Source Group 1'       evHs (int n);         • Rebuild all target files       (void) {	^
Target 1     2 * default 8 MHz clock     Source Group Source Group 1:     Add New Item to Group 'Source Group 1:     Add New Item to Group 'Source Group 1:     Remove Group Source G	
Some cross to PAS	
<pre>     CN * Control of corp found shop fo</pre>	
Add New Item to Group 'Source Group 1' Add Existing Files to Group 'Source Group 1' Remove Group Source Group 1 and its Files (void) { PAHBENR  = 0x00020000; /* enable GPIOA clock */	
Add Existing Files to Group 'Source Group 1'       All Existing Files to Group 'Source Group 1'         Remove Group Source Group 1 and its Files       (void) {         PAHBENR  = 0x00020000;       /* enable GPIOA clock */	
Remove Group Source Group 1 and its Files (void) { Rebuild all target files Rebuild all target f	
Rebuild all target files	
Build Target F7 a->MODER 4= ~0x00000000; /* clear pin mode */	
A->MODER  = 0x00000400; /* set pin to output mode */	
e (1) {	
Show Include File Dependencies GPTOA->SBRR = 0x00000020; /* turn on LED */	
1/ delayms (1000); 18 GPIOA->BSRF = 0x00200000; /* turn off LED */	
19 delayMs(1000);	
20 }	
22	
23 /* 8 MHz SYSCLK */	~
<b>Project</b>	>
ild Output	ф 🗵
	^
	>
d Existing Files to current Project Group ULINK2/ME Cortex Debugger	

13. Locate the file Blinky.c, click select it then click the "Add" button. Click the "Close" button to

🐻 Add Files to (	Group 'Source Group 1'		×
Look in: 📙 Blinl	ky	• 🗕 🖆 📰 •	
Name	^	Date modified	
BebugConfi	9	5/25/2021 8:50 PM	
Listings		5/25/2021 8:51 PM	
Objects		5/25/2021 8:51 PM	
RTE		5/25/2021 8:51 PM	
Blinky.c		5/25/2021 8:57 PM	
<			>
File <u>n</u> ame: Blir	iky.c	Add	
Files of type: C	Source file (*.c)	✓ Close	

proceed.

14. After the file is added to the project, it will appear under "Source Group 1".

		~
C:\Users\chens\Documents\Blinky\Blinky.uvprojx - µVision [Non-Commercial Use License]	- ⊔	X
Eile Edit View Project Flash Debug Peripherals Iools SVCS Window Help		1
□ 🗃 🖬 🖉   ǎ □ №   * ♥ ♥   ← →   陀 惊 微 微   译 译 // // // 🔯 🛛 🛛 😡 🕸 *   @ •   ● ○ 🔗	👧 🛛 📔	- 🔦
🕸 🕮 🕮 🗸 🖼 🛱 Target 1 🛛 🔽 🔊 🐁 🗣 🗇 🌚		
Project 4 🖬 🗋 Blinky.c		▼ ×
Project: Blinky 1 /* use delay loop, 1 sec on 1 sec off		^
□ 🔁 💭 Target 1 2 * default 8 MHz clock		
Source Group 1     3 * LD2 connects to PA5		
Blinky.c 5 finclude "stm32f0xx.h"		
CMSIS 6		
Device 7 void delayMs (int n);		
startup_stm32f030x8.s (Startup)		
System_stm32f0xx.c (Startup) 10 RCC->AHBENR  = 0x00020000; /* enable GPIOA clock */		
11		
12 GPIOA->MODER &= ~0x00000C00; /* clear pin mode */		
13 GPIOA->MODER  = 0x00000400; /* set pin to output mode *	/	
17 while(1) (		
16 GPIOA->BSRR = 0x00000020; /* turn on LED */		
17 delayMs(1000);		
18 GPIOA->BSR = 0x00200000; /* turn off LED */ 10 deladd (0000); /* turn off LED */		
20 }		
21)		
22		
▲ 23 /* 8 MHz SYSCLK */		$\checkmark$
E Project Sooks   1) Functions   0, Templates     <		>
Build Output		<b>Д</b> 🛛
		~
		$\checkmark$
<		>
ULINK2/ME Cortex De	bugger	

15. Before we build the project, we need to select the compiler used. Click the "Options for Target..." button to open up a dialog box.



16. In the dialog box, select the "Target" tab, in the Code Generation area, pick "Use default compiler version 5" from the pull-down menu. Click "OK" to close the dialog box.
In the older versions of Keil uVision IDE, the default compiler was version 5 and the programs in this book were tested using the default version 5 compiler. The recent versions of Keil uVision IDE changed the default compiler to version 6 and came with a new set of default compiler settings. Some of the programs in this book may not work properly with the new settings in version 6. Once you select the version 5 compiler, your computer should remember your choice.

Microelectronics STM3	2F030R8Tx		-Code ( ARM	Generation Compiler:	Use defau	lt compiler vers	sion 6 💌
perating system: None	Xtal (MHz):	12.0		•	Use defau Use defau V6.16 V5.06 upd	t compiler vere t compiler vers ate 7 (build 96	<del>iion S</del> sion 5 0)
TM32F0x0.svd			J U:	se MicroLli	BI	lig Endian	
Read/Only Memory Area	as		-Read/	Write Merr	nory Areas		
Jefault off-chip	Start Size	Startup	default	off-chip	Start	Size	Nolnit
ROM1:		С		RAM1:			
ROM2:		c		RAM2:			
ROM3:	<u> </u>	- c		RAM3:		[	
on-chip	1			on-chip	1		
₩ IROM1: 0x800	00000 0x10000	•		IRAM1:	0x20000000	0x2000	
IROM2:		0		IRAM2:			

17. Click the "Build" button to build the project.

C:\Users\chens\Documents\Blinky\Blinky.uvprojx	- µVision [Non-Commercial Use License] — [	) X
	₱ ゐ ゐ ゐ   淳 淳 // / / / @ ●	<b>.</b> •
Project		
Build (F7)	l binky.c	• •
Target 1	2 * default 8 MHz clock	Â
Source Group 1	3 * LD2 connects to PA5	
Blinky.c	4 */	
CMSIS	5 #include "stm32f0xx.h"	
🗆 💠 Device	7 void delayMs(int n);	
startup_stm32f030x8.s (Startup)	8	
system_stm32f0xx.c (Startup)	9 int main(void) {	
	10 RCC->AHBLNK  = 0X00020000; /* ENABLE GPICA CLOCK */	
	12 GPIOA->MODER &= ~0x00000C00; /* clear pin mode */	- 18
	<pre>13 GPIOA-&gt;MODER  = 0x00000400; /* set pin to output mode */</pre>	- 18
		- 18
	15 while (1) { 16 GPIOA->BSRR = 0x00000020; /* turn on LED */	- 11
	17 delayMs(1000);	- 11
	<pre>18 GPIOA-&gt;BSRR = 0x00200000; /* turn off LED */</pre>	- 11
	19 delayMs(1000);	
	21 }	
	22	
	23 /* 8 MHz SYSCLK */	¥
Project or Books   {} Functions   0, Templates	<<	>
uild Output		<b>4</b>
		~

18. You should get a clean build with this project with 0 Error and 0 Warning.



19. Once the project is built, we need to configure the debug interface so that the program may be downloaded to the target. Make sure the STM32F030 Nucleo-64 board is connected to the computer with the USB cable. The ST-LINK debugger on the Nucleo board is MBED compatible. It will appear as a USB thumb drive as below.

ine share view Drive	10013	
$\rightarrow$ $\checkmark$ $\uparrow$ $\blacksquare$ $\rightarrow$ This $\rightarrow$ NODE	✓ Č	
<ul> <li>OneDrive</li> <li>This PC</li> </ul>		
👕 3D Objects		
E Desktop	DETAILS.TXT MBED.HTM	
Documents		
Downloads		
Music		
Pictures		
Videos		
Data (D:)		
Local Disk (F:)		
NODE_F030R8 (F:)		
NODE_F030R8 (F:)		
💣 Network		
		8== -

- 20. Click the "Options for Target..." button again as in Step 15.
- 21. Select Debug tab. Select "ST-Link Debugger" from the pull-down menu. Click the radio button before "Use:".

C Use Simulator	with restrictions Settings	s lee: ULINK2/ME Costey Debugger V Settings	1
Limit Speed to I	Real-Time	ULINK2/ME Contex Debugger	
		ULINK Pro Cortex Debugger	_
Load Applicatio	n at Startup 🔽 Run to main()	CMSIS-DAP Debugger p main()	
Initialization File:		Initializatid J-LINK / J-TRACE Cortex	
	Edit	Edit	
Restore Debug S	ession Settings	Restore Remicro Debugger	_
Breakpoints	Toolbox	I anicio Debugger	
Watch Wine	dows & Performance Analyzer	Altera Blaster Cortex Debugger V Watch vvingows	
Memory Dis	play Vistem Viewer	Memory Display Viewer	
CPU DLL: F	Parameter:	Driver DLL: Parameter:	
SARMCM3.DLL	-REMAP	SARMCM3.DLL	_
Dialog DLL: F	arameter:	Dialog DLL: Parameter:	
DARMCM1.DLL	рСМ0	TARMCM1.DLL PCM0	-
Warm if outdate	d Even table is loaded	Wam if autidated Executable is leaded	
	d Executable is loaded		
	Manage Componen	nt Viewer Description Files	

22. Verify that "Load Application at Startup" and "Run to main()" are checked. Click the "Settings" button.

evice Target Output Listing User C/C++ Asm	Linker Debug Utilities
C Use Simulator with restrictions Settings	
✓ Load Application at Startup       ✓ Run to main()         Initialization File:          Edit          Restore Debug Session Settings         ✓ Breakpoints       ✓ Toolbox         ✓ Watch Windows & Performance Analyzer         ✓ Memory Display       ✓ System Viewer	Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Initialization File:       Image: Construction of the startup       Image: Construction of the startup         Restore Debug Session Settings       Image: Construction of the startup       Image: Construction of the startup         Restore Debug Session Settings       Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup       Image: Construction of the startup         Image: Construction of the startup       Image: Construction of the startup
CPU DLL: Parameter:	Driver DLL: Parameter:
SARMCM3.DLL -REMAP	SARMCM3.DLL
, Dialog DLL: Parameter:	Dialog DLL: Parameter:
DARMCM1.DLL pCM0	TARMCM1.DLL PCM0
Wam if outdated Executable is loaded Manage Component	Warn if outdated Executable is loaded

23. In the next dialog box, If the board is properly connected to the computer, the debug interface information should be filled in the dialog box automatically. You should be able to see the Serial Number, the Target Com Port is "SW", and the SW Device should be found with an IDCODE. Click the "OK" button to close it then click the "OK" button to close the Options dialog box.

Debug Adapter Unit: <mark>ST-LINK/V2-1 ▼</mark> ☐ Shareable ST-Link	IDCODE         Device Name         Move           SWDIO         0x0BB11477         ARM Core Sight SW-DP (ARM Core         Up
Serial Number:           066DFF393731435243164612           /ersion: HW: [V2-1]         FW: [V2J37M26]           Image: Check version on start           Target Com           Port: SW           Clock           Req: 10 MHz           Selected: 0           MHz	Automatic Detection ID CODE:     Manual Configuration Device Name:     Add Delete Update IR len: AP: 0
Debug - Connect & Reset Options - Connect: Normal 💽 Reset: Autodete	Cache Options Download Options act Cache Code Verify Code Download ✓ Cache Memory Download to ∃ash

24. The selected debug interface (ST-Link) should appear at the lower right corner. Click on the "Start/Stop Debug Session" button to launch the debugger.

File Edit View Project Flash Debug Periph	and Table CVCC Window Unio	
	erais loois svCs window Help	
□ 😂 🖬 🖉 🕹 🛍 🖄 🗠 🔶	M M M   # # # # #   #   Ø	1 - 🔍
🧇 🏥 🎬 🗼 🕶 🔛 🙀 Target 1	🔍 🔊 🛔 🖶 🔶 🐡 🎒	(Ctrl+F5)
roject 🗜 🎽	Enter or leave a debug ses	sion
🖃 🍄 Project: Blinky	1/* use delay loop, 1 sec on 1 sec off	^
🖃 🚂 Target 1	2 * default 8 MHz clock	
Source Group 1	3 * LD2 connects to PA5	
Blinkv.c	4 */	
	5 #include "stm32f0xx.h"	
	7 void delavMs(int n):	
startup_stm32f030v8.s (Startup)	8	
startup_stribziosokos (startup)	9 int main(void) {	
system_stms2toxx.c (startup)	10 RCC->AHBENR  = 0x00020000; /* enable GPIOA clock */ 11	
	<pre>12 GPIOA-&gt;MODER &amp;= ~0x00000C00; /* clear pin mode */</pre>	
	13 GPIOA->MODER  = 0x00000400; /* set pin to output mode */	
	14 15 mbile (1) (	
	16 GPTOA->BSRR = 0x00000020; /* turn on LED */	
	17 delavMs (1000);	
	<pre>18 GPIOA-&gt;BSRR = 0x00200000; /* turn off LED */</pre>	
	19 delayMs(1000);	
	20 }	
	21 }	
	23 /* 8 MHz SYSCLK */	
🖬 Project 🎯 Books   🕄 Functions   🗓 Templates	<	>
uild Output		<b>4</b>
ssembling startup stm32f030x8.s		^
ompiling system_stm32f0xx.c		
compiling Blinky.c		
inking	2 deter 0 77 deter 1622	
<pre>.\Objects\Blinkv.axf" = 0 Frror(s)</pre>	0 Warning(s).	
uild Time Elapsed: 00:00:01		
-		>
nter or leave a debug session	ST-Link Debugger	

25. The programmer progress bar appears at the bottom of the window. When done, the message "Running with Code Size Limit: 32K...<sup>1</sup>" should appear in the Command pane. The uVision IDE should change into debug perspective. The left pane displays the CPU registers. The right pane is split into Disassembly and source C code panes. The most commonly used debug buttons are at the beginning of the second row: Reset, Run, Stop, Step In (the function), Step over (the function), Step out (of the function), and Run to Cursor Line.

<sup>&</sup>lt;sup>1</sup> We are using a free non-commercial version, which has a 32K code size limit.

le Edit View Project Flash Debu	g Peripherals To	ools SVCS Window Help	•			
) 🖻 🛃 🧶 👗 🛍 🗳 🔍	🖛 🔿   🍋 🕴	14月1日 二日	R M	🗸 🗟 🥀 🙋 -	o 🔗 🚓 -	• 🛛 🖬 🔹 🖄
👬 🖪 🚳   የት የት የን 🔰 📑	> Q, II = 4	👌 💹 • 🔲 • 🖾 • 🎇 •	💷 + 🔢 + 🔀 -			
jisters 🔄 Run (F5) 🛛 📮 🗵	Disassembly					<b>4</b>
gister Start code execution	0x080001FA 10: 11:	.0000 DCW RCC->AHBENR  = 0:	0x0000 x00020000; /*	enable GPIOA clock */		^
R1 0x20000260	Cx080001FC	4815 LDR	r0,[pc,#84] ; @0	x08000254		
R2 0x20000260	0x08000200	2101 MOVS	r1, #0x01			
R3 0x20000260	0x08000202	0449 LSLS	r1,r1,#17			
B5 0x20000000	0~08000204	4308 0005	r0 r0 r1			Ň
R6 0x0000000						,
R7 0x0000000	Blinky.c	startup_stm32f030x8.s				<b>▼</b> ×
R8 0xFFFFFF R9 0xFFFFFFF R10 0x08000268	7 void 8 9 int	<pre>d delayMs(int n); main(void) {</pre>				^
R11 Ux08000268	10	RCC->AHBENR  = 0x0	0020000; /* en	able GPIOA clock */		
B13 (SP) 0x20000660	11					
R14 (LR) 0x0800012B	12	GPIOA->MODER &= ~0x	00000000; /* ci	ear pin mode ~/		
R15 (PC) 0x080001FC	14		, , ,	o pin oo ouopuo moue	·	
xPSR 0x01000000	15	while(1) {				
Banked	16	GPIOA->BSRR = 0	x00000020; /* tu	rn on LED */		
Jystem Viewal	17	delayMs(1000);				~
Project Registers	<	GPTOA-SHSRR = 0	¥00200000• /* †u	rn off LKD */		>
nmand	,	д 🗙	Call Stack + Locals			<b>4</b> X
nning with Code Size Limit:	32K	^	Name	Location/Value	Type	
ad "C:\\Users\\chens\\Docum	ents\\Blinky\	\Objects\\Blinky	e main	0~0000000	int f()	
			+ main	0.0000000	incit	
		~				
		>				
SIGN BreakDisable BreakEnabl	le BreakKill	BreakList BreakSet	Call Stack + Locals	Memory 1		

Click the "Run" button and the program will start running and the LED labeled LD2 blinks.

26. To stop the debug session and return to the build project perspective, click on the "Start/Stop Debug Session" button.



27. Congratulations! You have successfully finished the first programming project for the STM32F030 Nucleo-64 board using Keil MDK v5 uVision IDE!