Start a New Project with Keil MDK-ARM Version 5 and Atmel SAMD21

This tutorial is intended for starting a new project to develop software with Atmel SAMD21 Xplained Pro board (with the device ATSAMD21J18A) using Keil Microcontroller Development Kit for ARM (MDK-ARM) version 5.21a.



µVision V5.21.1.0 Copyright (C) 2016 ARM Ltd and ARM Germany GmbH. All rights reserved.

Create a New Project in Keil MDK v5.21

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

🗈 💕 🖬 🥥 석		New µVision Project		8 🚓 🖬 • 🔦
0 II II 0 I		New Multi-Project Workspace		
Project		Open Project Close Project		
		Export Manage)	
		Select Device for Target		
		Remove Item		
	1	Options	Alt+F7	
		Clean Targets		
		Build Target	F7	
		Rebuild all target files		
		Batch Build		
		Translate	Ctrl+F7	
		Stop build		

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.

🔣 Create New Project				×
\leftarrow \rightarrow \checkmark \uparrow 🗎 \diamond This PC \diamond Documents \diamond	~ U	Search Documents		P
Organize • New folder				?
Public Create a new folder.	Date modified	Туре	Size	^
This PC	12/23/2016 9:03 A	File folder		
Desktop				
 Documents Downloads 				
Music				
Videos				~
ت Ti106649001 (ش ۲ ۲ ۲) File <u>n</u> ame:				> ~
Save as type: Project Files (*.uvproj; *.uvprojx)		<u>O</u> pen	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.

- → ✓ ↑ 📳 > This PC > Documents >	v ℃	Search Documents		P
Organize 🝷 New folder			**** •	0
Public Name This PC	Date modified	Туре	Size	^
Documents				
Downloads				
🐌 Music				
Pictures				
📱 Videos				
🐛 TI10664900J (C:)				1 K
blinky	12/23/2016 9:01 A	File folder		~
v <				>
File name:				~
Save as type: Project Files (*,uyproj: *,uyprojx)				~

5. While in the project folder, type "blinky" in the File name field and click "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.

· · · · · · · · · · · · · · · · · ·		h hlinter	0
	✓ O Searc	n biinky	þ
Organize 🔻 New folder			• ?
^ Name	^	Date modified	Туре
	No items match your se	arch.	
neDrive			
🥏 Documents			
osic 😸 Music			
🥏 Pictures			
🧓 Public			
s This PC			
📘 Desktop			
Desktop			
Desktop			3
Desktop Documents File name: blinky			

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 "ATSAMD21J18A". 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 unique 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 "OK" button.

evice	Software Packs	•
Vendor: Device: Toolset Search:	Atmel ATSAMD21J18A ARM [d21j1	_
■ [●] At	mel SAM D Series SAM D21 ATSAMD21J15A	Description: The Atmel SMART SAM D ARM Cortex-M0+ based microcontroller (MCU) series builds on decades of innovation and experience in embedded Flash microcontroller technology. It not only sets a new benchmark for flexibility and ease-of-use but also combines the performance and energy efficiency of an ARM Cortex-M0+ based MCU with an optimized architecture and peripheral set. The Atmel I SMART SAM D SAM Cortex-M0+ desay and the set of the
	ATSAMD21115B ATSAMD21116A ATSAMD21116B ATSAMD21119A	A rich set of peripherals, flexibility and ease-of-use combined with low power consumption make the Atmel SAM D21 ARM Cortex-M0+ based microcontroller that set of peripherals, flexibility and ease-of-use combined with low power consumption make the Atmel SAM D21 ARM Cortex-M0+ based microcontroller series

7. Next, the Project Wizard will help you manage the run-time environment. Expand the Software Component selections, check "CMSIS>CORE" and "Device>Startup" then click "OK" button.

S Vallant		Version	Description	
			Cortex Microcontroller Software Interface Components	
		5.0.0	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M	
		1.4.6	CMSIS-DSP Library for Cortex-M, SC000, and SC300	
		1.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
		2.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
			Unified Device Drivers compliant to CMSIS-Driver Specifications	
			ARM Compiler Software Extensions	
-			Startup, System Setup	
()		1.0.1	System Startup for Atmel SAM D21 Devices	
MDK-Pro	~	6.8.0	File Access on various storage devices	
MDK-Pro	~	5.32.2	User Interface on graphical LCD displays	
MDK-Pro	~	7.2.0	IPv4/IPv6 Networking using Ethernet or Serial protocols	
MDK-Pro	~	6.8.0	USB Communication with various device classes	
	De	scription		
	MDK-Pro MDK-Pro MDK-Pro MDK-Pro	Image: Wide and the second	Image: Constraint of the second sec	Cortex Microcontroller Software Interface Components S0.0 CMSIS-CORE for Cortex-M. SC000. SC300. ARMv8-M 1.4.6 CMSIS-SCORE for Cortex-M. SC000. and SC300 1.0 CMSIS-RIDS API for Cortex-M. SC000. and SC300 2.0 CMSIS-RIDS API for Cortex-M. SC000. and SC300 3.0 CMSIS-RIDS API for Cortex-M. SC000. and SC300 4.0 CMSIS-RIDS API for Cortex-M. SC000. and SC300 5.0 System Startup for Atmel SAM D21 Devices MDK-Pro S32.2 User Interface on graphical LCD displays MDK-Pro 68.0 USB Communication with various device classes Description Description

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 CMSIS and startup codes for the specific device were created in the project. Click "New" button for a new text file in the editing window.

C:\Users\ \Documents\blinky\blinky.uvprojx - μVision	- 0	×
<u>File E</u> dit <u>V</u> iew <u>P</u> roject Fl <u>a</u> sh <u>D</u> ebug Pe <u>ri</u> pherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp		
□ 2 日 2 本 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	🗟 🕫 🍳 🔹 🔗 🔗 💼 🔹 🔦	
🚺 New (Ctrl+N) 🔤 🕺 🐇 📥 着 🚸 🐡 🏟		
pro Create an empty document 🛛 🖡 🔟		
Project: blinky		
🖻 💭 Target 1		
Source Group 1		
CMSIS		
startup SAMD21 c (Startup)		
svstem samd21 c (Startun)		
E Project Books D Functions D Templates		
Build Output		ą 🗵
		~
		Y
		>
Create an empty document	ULINK2/ME Cortex Debugger	

9. Copy the code below into the new text file window. This is a simple program to blink the LED0 on the SAMD21 Xplained Pro board.

```
// Toggle LED0 on SAMD21 Xplained Pro at 1 Hz.
// LED0 is connected to PB30.
#include "samd21.h"
void delayMs(int n);
int main (void) {
   REG PORT DIR1 |= 0x40000000; /* make PB30 output */
   /* toggle LED continuously at 1Hz */
   while(1) {
       REG PORT OUT1 &= ~0x40000000; /* turn on LED0 */
       delayMs(500);
       REG_PORT_OUT1 |= 0x40000000;  /* turn off LED0 */
       delayMs(500);
    }
}
/* millisecond delay based on 1 MHz system clock */
void delayMs(int n) {
   int i;
   for (; n > 0; n--)
       for (i = 0; i < 199; i++)
           __asm("nop");
}
```

10. Click "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 "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'...".

C:\Users\ \Documents\blinky.vprojx - µVision	- 🗆	×
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
□ 🐸 🛃 🖉 ǎ 🖻 🏝 🍠 🕫 / ← → 陰 陰 陰 陰 信 譯 /// // // // // // // // // // // //	1 - 3	
🔗 🖾 🕮 🥔 🗮 🛱 Target 1 🔤 🕺 📥 🗟 🖘 🐡 🎒		
Project 4 🖬 🗋 blinky.c		▼ ×
Project: blinky 1// Toggle LED0 on SAMD21 Xplained Pro at 1 Hz. 2// LED0 is connected to PB30. Source Council		^
CM: Options for Group 'Source Group 1' Alt+F7 "Sama21.1"		
Add New Item to Group 'Source Group 1' Add Existing Files to Group 'Source Group 1' Add Existing Files to Group 'Source Group 1'		
Remove Group 'Source Group 1' and its Files (vold) { Rebuild all target files ORT_DIR1 = 0x40000000; /* make	PB30 outpu	.t */
Build Target F7 ggle LED continuously at 1Hz */ ▲ Manage Project Items {(1) { LEG PORT OUT1 ≤= ~0x40000000; /* turn	on LED0 */	
Show Indude File Dependencies ielayMs (500); 13 REG_PORT_OUT1 = 0x40000000; /* turn 16 delayMs (500);	off LED0 *	/
17 } 18 } 19		J
E Project GBooks O Functions 0, Templates	.1_ ↓/	>
Build Output		4 ×
		< >
<		>
Add Existing Files to current Project Group ULINK2/ME C	Cortex Debugger	

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

Look <u>i</u> n:	Blinky	▼ = 🔁 →	
Name	^	Date modified	Туре
Listings		12/23/2016 9:05 AM	File fo
Objects		12/23/2016 9:05 AM	File fo
RTE		12/23/2016 9:05 AM	File fo
🔛 blinky.c)	12/23/2016 9:11 AM	C File
<			,
ile <u>n</u> ame:	blinky.c		bb
S. 11 (1)			100

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

Project	4 🛛	blinky.c			▼ ×
Project: blinky Farget 1 Source Group 1 blinky.c CMSIS Ovice		1// T 2// L 3 4 #inc 5 6 void 7	oggle LEDO on SAMD21 Xplained Pr EDO is connected to PB30. lude "samd21.h" delayMs(int n);	o at 1 Hz.	^
startup_SAMI	D21.s (Startup) 21.c (Startup)	8 int 9 10 11 12	<pre>main (void) { REG_PORT_DIR1 = 0x40000000; /* toggle LED continuously at 1H while(1) {</pre>	/* make PB30 output z */	*/
		13 14 15	<pre>REG_PORT_OUT1 &= ~0x40000000 delayMs(500); REG_PORT_OUT1 = 0x40000000;</pre>	; /* turn on LED0 */ /* turn off LED0 */	
•		16 17 18 } 19	delayMs(500); }		~
Project 🚷 Books {} Func	tions 0. Templates <	00 (*	2112a a-1 ba 1 km-	11- + /	>

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



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



- 17. 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 Atmel SAMD21 Xplained Pro board is connected to the computer with the USB cable.
- 18. Click "Options for Target..." button.



19. Select Debug tab. Click the radio button before "Use:". Click the pull-down menu on the right and select "CMSIS-DAP Debugger".

 Use Simulato Limit Speed to 	r <u>with restrictions</u> Se	ettings	ULINK2 ULINK2 ULINK	2/ME Cortex Debugger 2/ME Cortex Debugger Pro Cortex Debugger	Settings
Load Applica nitialization File:	tion at Startup 🔽 Run to main	0 V Londo Initializatio	CMSIS J-LINK Models ST-Link PEMicr	-DAP Debugger / J TRACE Contex s Cortex-M Debugger k Debugger to Debugger	main()
Restore Debug Breakpoir Watch Wi Memory D	Session Settings Its 🔽 Toolbox ndows & Performance Analyzer Display 🔽 System Viewer	Restore IV Bre IV Wa IV Me	Stellari SiLabs Altera E tch Win	s ICDI UDA Debugger Blaster Cortex Debugger dows isplay IV System V	v
CPU DLL:	Parameter:	Driver DLL	:	Parameter:	
SARMCM3.DLL		SARMCM	3.DLL		
Dialog DLL:	Parameter:	Dialog DLI	L:	Parameter:	
DARMCM1.DLL	-pCM0+	TARMCM	1.DLL	-pCM0+	

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

C Use Simulato	with restrictions Se	ttings OJse:	ULINK2 ULINK2	2/ME Cortex Debugger Settings
✓ Load Applicat Initialization File: —Restore Debug ↓ Breakpoir ↓ Watch Wir ↓ Memory D	ion at Startup Run to main() Session Settings ts Foolbox adows & Performance Analyzer isplay System Viewer) Initializatio	DLINK CMSIS- J-LINK Models ST-Link PEMicr NULink Stellaris SiLabs Altera E tch Wint mory Di	PAP Debugger DAP Debugger i Cortex M Debugger c Debugger s ICDI UDA Debugger slaster Cortex Debugger v v Splay ✓ System Viewer
CPUDII	Parameter	Driver DI I	-	Parameter
SARMCM3.DLL	1	SARMCM	3.DLL	
Dialog DLL:	Parameter:	Dialog DLI	-	Parameter:
DARMCM1.DLL	-pCM0+	TARMCM	1.DLL	-pCM0+
	Manage Com	ponent Viewer Descrip	tion File	· · · · · · · · · · · · · · · · · · ·

21. In the next dialog box, pull down the Port: and select "SW". The Atmel SAMD21 Xplained Pro board uses "Serial Wire" debug interface, not JTAG. If the board is properly connected to the computer, the debug interface information should be filled in the dialog box automatically. Click "OK" button to

close it then click "OK" button to close the Options dialog box.

CMSIS-DAP - JTAG/SW Adapter	SW Device			
EDBG CMSIS-DAP	IDCODE	Device Name		Move
Serial No: ATML213002180000	SWDIC 0x0BC11477	ARM CoreSight SW-DP		Up
Firmware Version: U3. 19.0 TA3	Automatic Detection Manual Configuration	ID CODE:		
	Add Delete	Update	AP:	0x00
Connect & Reset Options		Cache Options	Download Options	
Connect Normal 💌 Re	set Autodetect	Cache <u>C</u> ode	└── Verify Code Down └── Download to <u>F</u> las	load h

22. The selected debug interface should appear at the lower right corner. Click on the "Start/Stop Debug Session" button to launch the debugger.

C:\Users\ \Documents\blinky\blinky.uvproj	x - μVision	- 0	×
File Edit View Project Flash Debug Peripherals	Tools SVCS Window Help		
	陰 读 評 //// //// 🖄 stack 🛛 🔽 🛤 🖉 🍭	● ○ ⊘ @ = ◄	
🔮 🕮 🥔 🔜 🙀 Target 1 🛛 🗸 🔊	📩 🗟 🗇 🐡 🏙	Start/Stop Debug Session (Ctrl+F5)	
Project 📮 🗵	blinky.c	Enter or leave a debug session	• ×
Project: blinky Target 1 Source Group 1 CMSIS CMSIS Device Startup_SAMD21.s (Startup) System_samd21.c (Startup) Project @ Books {} Functions (0, Templates)	<pre>1 // Toggle LED0 on SAMD21 Xplained 2 // LED0 is connected to PB30. 3 4 #include "samd21.h" 5 6 void delayMs(int n); 7 8 int main (void) { 9 REG_PORT_DIR1 = 0x40000000; 10 11 /* toggle LED continuously at 12 while(1) { 13 REG_PORT_OUT1 &= ~0x40000 14 delayMs(500); 15 REG_PORT_OUT1 = 0x400000 16 delayMs(500); </pre>	<pre>Pro at 1 Hz. /* make PB30 output * Hz */ 000; /* turn on LED0 */ 00; /* turn off LED0 */</pre>	*/
Build Output			a 🗖
<pre>*** Using Compiler 'V5.06 update 3 (build 3) Build target 'Target 1' assembling startup SAMD21s compiling system_samd21.c compiling blinky.c linking Program Size: Code=432 RO-data=212 RW-data='' .\Objects\blinky.axf" - 0 Error(s), 0 Warn: Build Time Elapsed: 00:00:12</pre>	00)', folder: 'C:\Keil_v5.21a\ARM\ARMCC\Bin' 4 ZI-data=612 ng(s).		<
<			>
Enter or leave a debug session		CMSIS-DAP Debugger	

23. The programmer progress bar appears at the bottom of the window. When done, the message should appear in the Build Output pane. If you are using the evaluation version of the Keil uVision, a warning of the code size limitation will pop up after the target device is programmed. Click "OK" to

close the size warning message dialog box.

C:\Users\ \Documents\blinky\blinky.uvproj	ix - µVision — 🗌	\times
😵 🖾 🔛 😻 🚔 🚑 Target 1 🛛 🗹 🔊		
Project 🛛 🗘 🖬	blinky.c	▼ ×
 Project: blinky Farget 1 Source Group 1 blinky.c CMSIS Device startup_SAMD21.s (Startup) system_samd21.c (Startup) 	<pre>1// Toggle LED0 on SAMD21 Xplained Pro at 1 Hz. 2// LED0 is connected to PB30. 3 4 #include "samd21.h" 5 #Vision X 000; /* make PB30 output 1y at 1Hz */ 40000000; /* turn on LED0 */ 0K 000000; /* turn off LED0 */</pre>	*/
Project OBooks Functions D_ Templates	<	>
Build Output		4 ×
<pre>linking Program Size: Code=432 RO-data=212 RW-data=4 ".\Objects\blinky.axf" - 0 Error(s), 0 Warni Build Time Elapsed: 00:00:12 Load C:\\Users\\Shujen\\DocUmerca\\Blinky\\ Erase Done. Verify OK. Flash Load finished at 09:15:34 </pre>	4 ZI-data=612 Ing(s). Objecta\\blinky.axf"	~
	CMSIS-DAP Debuaaer	

24. The uVision IDE should change to the debug perspective. Click "Run" button and the program will start running and the LED0 blinks.

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help Registers Currently used: 648 Bytes (14) File Core Project Flash Debug Peripherals Tools SVCS Window Help Registers File File File File File File File File	C:\Users\ \Documents\blinky\blinky.uvprojx - µVision		\Box ×
Registers	File Edit View Project Flash Debug Peripherals Tools SVCS Window Help		
Registers Disassembly 4 Registers Start code execution 0x0000021E 0000 MOVS r0,r0 9 EEG_PORT_DIR1 = 0x40000000; /* make FB30 output */ 10: iii: /* toggle LED continuously at HE */ 0x00000224 0x00000226 v v R4 0x0000024 v v v R6 0x0000000 Startup_SAMD21s v v R7 0x0000000 8 int main (void) { v v R10 0x0000000 8 int main (void) { v v v R10 0x0000000 8 int main (void) { v v v R11 0x0000000 8 int main (void) { v v v R12 0x0000000 V 9 REG_PORT_DIR1]= 0x40000000; /* make PB30 output */ v R13 REG_PORT_OUT1 is continuously at Hz */ v v v R14 (R) 0x0000000 v is REG_PORT_OUT1 is continuously at Hz */ v R14 v call Stack + Locals 1	□ ○ □ 2 2 2 3 3 3 3 3 7 (← →) 1 2 1 3 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2	ି 🔗 🌲 🔲 🔍	
Register Start code securition 0x0000021E 0000 MOVS r0,r0 Start code securition 9: REG_PORT_DIRL = 0x40000000; /* make PB30 output */ 10: 11: /* toggle LED continuously at 11x */ R2 0x00000284 0x00000000 R4 0x00000000 VDR r0,r6 R6 0x00000000 VDR r0,r6 R6 0x0000000 VDR r0,r6 R7 0x0000000 VDR r0,r6 R7 0x0000000 VDR r0,r6 R6 0x0000000 VDR r0,r6 R7 0x0000000 VDR r0,r6 R11 0x0000000 VDR r0,r6 R12 0x0000000 VDR r0,r6 R14 (LR) 0x0000000 V V R14 (LR) 0x0000000 V V V V V	Registers Run (F5) Disassembly		д 🗴
R2 0.20000083 R3 0.20000083 R4 0.20000084 R5 0.20000000 R6 0.00000000 R7 0.00000000 R8 0.200000074 R4 0.00000004 R6 0.0000000074 R10 0.000000074 R11 0.000000074 R12 0.000000074 R11 0.000000074 R11 0.000000074 R11 0.000000074 R12 0.000000074 R13 (SP) 0.020000068 R14 0.00000004 R14 0.000000064 R14 0.000000064 R14 0.000000064 R14 0.000000064 R14 0.0000000064 R14 0.000000000 R14 0.00000000 R2 Call Stack + Locals R2 Registers *** Restricted Version with 32768 R2 Call Stack + Locals * m	Register Start code execution 0x0000021E 0000 MOVS r0,r0 r0,r0 9: REG_PORT_DIR1 = 0x40000000; /* make PB30 output */ 10: 11: /* toggle LED continuously at Hz */ R0 0x20000068 11: /* toggle LED continuously at Hz */ 12: * 6x00000260	(^
R3 020000058 R4 0x00000004 R5 0x20000004 R6 0x00000000 R7 0x00000000 R8 0xD0FFEFF7 R9 0x200000074 R10 0x000000074 R11 0x000000074 R11 0x000000074 R11 0x000000074 R11 0x000000074 R11 0x000000074 R12 0x1Becomotion R14 0x00000000 R14 0x0000000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x000000000 R15 REG_PORT_OUT1 R14 0x01000000 R15 REG_PORT_OUT1 R14 0x010000000 R15 REG_PORT_OUT1 R14 0x010000000 R15 REG_PORT_OUT1 R14 0x010000000 R15 REG_PORT_OUT1 R16			```
R3 00000000 R7 0x0000000 R8 0x0000000 R9 0x20000074 R10 0x00000004 R10 0x00000024 R11 /* toggle LED continuously at 1Hz */ R12 0xFBFFDFF7 R13(SP) 0x20000264 R14 0x0000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x00000000 R14 0x000000000000 R14 0x00000000000000000000000000000000000	R3 0x2000068		+ ×
Command Image: Call Stack + Locals Image: Call Stack + Locals Image: Call Stack + Locals **** Restricted Version with 32768 Byte Code Size Limit Image: Call Stack + Locals Type **** Currently used: 648 Bytes (1%) Image: Call Stack + Locals Type <	R6 0x0000000 R7 0x0000000 R8 0xDFFBFF7 R9 0x20000074 R10 0x0000024 R11 0x0000024 R12 0xFFBFF77 R13(SP) 0x20000268 R14(R) 0x0000204 R15(CpC) 0x0000020 Project Registers	330 output */ n LEDO */ Ef LEDO */	^ ~ >
*** Restricted Version with 32768 Byte Code Size Limit *** Currently used: 648 Bytes (1%) Name Location Type • m 0x000000 int f0 • SSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet Cull Stack + Locals Memory 1	Command		д 🗵
Abstand preakulasite breakting	<pre>*** Restricted Version with 32768 Byte Code Size Limit *** Currently used: 648 Bytes (1%) </pre> Name Location Type m 0x000000 int f() Source Size Limit (1)		
MNN-LIAR LIAPUID	ASSIGN BREAKDISADLE BREAKLINDLE BREAKLIND BREAKLIST BREAKSET	CMSIS-DAP Debugger	+1:0.00

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

C:\Users\	\Documents\blinky\blinky.uvprojx - µVision	- 0	×
File Edit View	Project Flash Debug Peripherals Tools SVCS Window Help		
🗋 💕 🛃 🕘	メ 🖦 🙇 🤊 や ← ⇒ や 改 改 改 律 準 /// //k 20 stack 🛛 🔽 象 🖉		
🏦 🗐 🕲		Q Start/Stop Debug Session (Ctrl+F5)	
Registers	4 🗵 Disassembly	Enter or leave a debug session	д×
Register ⊂ Core − R0 − R1 − R2 − R3 − R4 − R5 − R6 − R7 − R8 − R9 − R10 − R12 − R13 (SP)	Value 0x0000021E 0000 MOVS r0,r0 9: REG_PORT_DIR1 = 0x40000000; /* make FB30 or 0x2000068 0: 11: /* toggle LED continuously at 1Hz */ 0x2000068 0x20000068 0x00 r0,r60]; @ make FB30 or 0x20000068 0x00000204 decord DE r0.[pc,f60]; @ make FB30 or 0x20000068 0x00000204 decord DE r0.[pc,f60]; @ make FB30 or 0x20000068 0x00000204 decord DE r0.[pc,f60]; @ make FB30 or 0x00000004 0x00000204 decord DE r0.[pc,f60]; @ make FB30 or 0x00000000 6 void delayMs(int n); 0x0000000; % make FB50 or 0x00000000 6 void delayMs(int n); 0x0000000; /* make FB50 or 0x0000024 9 REG_PORT_DIR1 [= 0x40000000; /* make FB50 or 0x00000284 0 PREG_PORT_DIR1 [= 0x40000000; /* make FB50 or 0x00000284 1 /* toggle LED continuously at 1Hz */ 0x20000268 12 while(1) {	utput */	 ∧ → × ∧
R14 (LR) R15 (PC) R15 (PC) Project	6x00000147 13 All _ FORT_OUT1 = _ Oxf00000000, // * t 0x0000020 14 delayMs (500); 0x0100000 v 15 REG_PORT_OUT1 = 0x40000000; /* t	turn off LED0 */	~
Command	T I Call Stack + Locals		д 🗴
*** Restricted *** Currently	Version with 32768 Byte Code Size Limit used: 648 Bytes (1%)		
>			
ASSIGN BreakDi:	sable BreakEnable BreakKill BreakList BreakSet	mory 1	
Enter or leave a de	bug session	CMSIS-DAP Debugger t1:	0.00

26. Congratulations! You have successfully finished the first programming project for Atmel SAMD21 Xplained Pro board using Keil MDK v5 uVision IDE!