Procedure to create a new project using the Keil IDE ver 4

- 0) Create a folder (directory) for your project
- 1) Download file keilstartup.zip from the class webpage to the project folder and unzip it.
- 2) Start the Keil IDE software. If it starts up with an open project, close the project.
- Create a new uVision project. On the project tab select New uVision Project: Project > New uVision Project
- A Create New Project window will open.
 Navigate so that the directory you created is shown in the Save In box at top of window..
 Enter the name you wish for the project in the File Name field at the bottom.
 Click Save.
- 5) A Select Device window will open with a long list of company names:

Select Device for Target 'Target 1'	
CPU Vendor: <unknown> Device: <unknown> Toolset: <unknown> Data base Description:</unknown></unknown></unknown>	
Actel Analog Devices ARM Atmel AustriaMicroSystems Cirrus Logic Cypress Dialog Semiconductor Ember Energy Micro Freescale Semiconductor Fujitsu Semiconductor Generic Currus Logic Currus Logic Curru	
OK Cancel	Help

Scroll down to STMicroelectronics, click on the + symbol and then scroll down to select the STM32F100RB microcontroller:

Select Device for Target 'Target 1'
CPU Vendor: STMicroelectronics Device: STM32F100RB Toolset: ARM Data base Description: ARM 32-bit Cortex-M3 Microcontroller, 24MHz, 128kB Flash, 8kB SRAM, PLL, Embedded Internal RC 8MHz and 40kHz, Real-Time Clock, Power Saving Modes, JTAG and SWD, 7-channel DMA, Nested Interrupt Controller, STM32F100CB STM32F100R6 STM32F100R6 STM32F100R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32F10R8 STM32
OK Cancel Help

Note that in the description window appears a summary of the resources that come with this microcontroller. Click OK.

6) A prompt to add a start up file will appear:

µVision	
?	Copy 'startup_stm32f10x_md_vl.s' to Project Folder and Add File to Project ?
	Yes No

Click Yes.

7) If you have a source file you have begun writing, such as a modified Blinky.c file, copy that file to the directory where the new project is but rename the file to match the name you have given to your new project. NOTE: project names and files must not have spaces in them.

8) On the left of the screen is a project box like this that shows the files associated with your project:



Click on the + just left the word Target 1 to get this:



Right click on the line Source Group 1 Left click on Add Existing Files to Group 'Source Group 1'

Another window will open showing files in the project. Select your source file (.c file) and click Add. Select the **system_stm32f10x.c** file and click Add Close the window. 9) On the Flash tab select Configure Flash tools (**Flash > Configure Flash Tools**). A window like this should open (it is possible that it will open with a different tab active. Click the left tab which is labeled Device to get this screen). STM32F100RB should be shown.

Coptions for Target 'STM32F1	IORB Flash'	
Vendor: STMicroelectronics Device: STM32F100KB Flash Vendor: STMicroelectronics Device: STM32F100RB Toolset: ARM STM32F078CB STM32F078CB STM32F078VB STM32F00C4 STM32F100C4 STM32F100C6 STM32F100C8 STM32F100C8 STM32F100C8 STM32F100C8 STM32F100C8 STM32F100C8 STM32F100R4 STM32F100R4 STM32F100R4 STM32F100R8 STM32F10R8 STM32F100R8 STM32F10R8 STM32		
STM32F100RC		
	OK Cancel Defaults Help	

Clicking the Target tab give this (default values should be ok):

🛚 Options for Target 'STM32F100RB Flash'			
Device Target Output Listing User C/C++ Asm Linker Debug Utilities			
STMicroelectronics STM32F100RB			
Xtal (MHz): 8.0	Code Generation		
Operating system: None	Use Cross-Module Optimization		
System-Viewer File (.Sfr):	Use MicroLIB 🗖 Big Endian		
C:\Keil4\ARM\SFD\ST\STM32F1xx\STM32F100xx.sfr			
Use Custom SVD File			
Read/Only Memory Areas	Read/Write Memory Areas		
default off-chip Start Size Startup	default off-chip Start Size Nolnit		
□ R0M1: C	□ BAM1: □ □		
□ R0M2: □ ○	RAM2:		
ROM3:	RAM3:		
on-chip	on-chip		
▼ IROM1: 0x8000000 0x20000 •	▼ IRAM1: 0x20000000 0x2000 □		
IROM2:	IBAM2:		
OK Can	cel Defaults Help		

Clicking the Output tab should reveal this window. By default the name shown should be the name of your project (Blinky happened to be the project used to get this screen shot):

I Options for Target 'STM32F100RB Flash'	X
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Select Folder for Objects Name of Executable: Blinky	
 Create Executable: .\Flash\Blinky Debug Information 	🔲 Create Batch File
Create HEX File From Browse Information	
C Create Library: .\Flash\Blinky.lib	
OK Cancel Defaults	Help

The Listing tab has this (default selections should be ok):

W C	ptions for Target 'STM32	2F100RB Flash'		
De	vice Target Output Listing	User C/C++ Asm	Linker Debug Utilities	
	Select Folder for Listings	Page	Width: 79 📩 Page Length: 66 📩	
	 ✓ Assembler Listing: .\Flash*.lst ✓ Cross Reference 			
	C Compiler Listing: .\Flash*.txt C Preprocessor Listing: .\Flash*.i			
[☑ Linker Listing: .\Flash\Blin	ky.map		
	Memory Map	Symbols	🔽 Size Info	
	🔽 Callgraph	Cross Reference	✓ Totals Info	
			Unused Sections Info	
			Veneers Info	
		ОК С	ancel Defaults	Help

Here is the User tab contents (the default, blank, fields are ok):

🛚 Options for Ta	arget 'STM32F100RB Flash'	×
Device Target 0	Dutput Listing User C/C++ Asm Linker Debug Utilities	
Run User Programs Before Compilation of a C/C++ File		
🔲 Run #1:	🗖 DOS16	
	Stop Build/Rebuild #1 on Exit Code: Not Specified	
□ Run #2:	🗖 DOS16	
	Stop Build/Rebuild #2 on Exit Code: Not Specified	
- Run User Progra	ams Before Build/Rebuild	
🔲 Run #1:	🗖 DOS16	
🔲 Run #2:		
Run User Programs After Build/Rebuild		
🗖 Run #1:	🗖 DOS16	
🔲 Run #2:	🗖 DOS16	
Beep When Complete Start Debugging		
OK Cancel Defaults Help		

The C/C++ tab does need your attenton. In the Define box the processor being used must be defined. Enter $STM32F10X_MD_VL$.

🖲 Options for Target 'Target 1'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Preprocessor Symbols Define: Undefine:	
Language / Code Generation Strict ANSI C Optimization: Level 0 (-00) Optimize for Time Split Load and Store Multiple Split Load and Store Multiple One ELF Section per Function Read-Write Position Independent Include Paths Misc Controls Compiler ccpu Cortex-M3-D_EVAL -g -00apcs=interwork 1 C: Keil4\ARM\RV31\INC	Warnings: <unspecified></unspecified> Thumb Mode No Auto Includes
OK Cancel Defaults	Help

Below is the window opened when Asm is selected (default values are ok):

🕱 Options for Target 'STM32F100RB Flash'	×	
Device Target Output Listing User C/C++ Asm Linker Debug Utilities		
Conditional Assembly Control Symbols		
Define:		
Undefine:		
Language / Code Generation		
Split Load and Store Multiple		
Read-Only Position Independent		
Read-Write Position Independent		
Thumb Mode		
No Warnings No Auto Includes		
Include		
Paths J		
Controls		
Assembler		
control I C:\Keil4\ARM\RV31\INC		
string		
OK Cancel Defaults Help		

And the Linker tab (default is ok):

Coptions for Target 'STM32F100RB	Flash'
Device Target Output Listing User 0	C/C++ Asm Linker Debug Utilities
 Use Memory Layout from Target Dialog Make RW Sections Position Independent Make RO Sections Position Independent Don't Search Standard Libraries Report 'might fail' Conditions as Errors 	dent R/O Base: 0x08000000 dent R/W Base 0x20000000 disable Warnings:
Scatter File	Edit
Misc controls	
Linkercpu Cortex-M3 *.o controlstrictscatter ''.\Flash\Blinky string	.sct"
OK	Cancel Defaults Help

The Debug tab opens this window. Note that the ST-Link Debugger must be chosen:

🛚 Options for Target 'STM32F100RB Flash'		
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities	
C Use Simulator Settings	 Use: ST-Link Debugger Settings 	
✓ Load Application at Startup ✓ Run to main() Initialization File:	✓ Load Application at Startup ✓ Run to main() Initialization File:	
Edit	Edit	
Restore Debug Session Settings Breakpoints Toolbox Watch Windows & Performance Analyzer Memory Display	Restore Debug Session Settings Breakpoints Watch Windows Memory Display	
CPU DLL: Parameter: SARMCM3.DLL	Driver DLL: Parameter: SARMCM3.DLL	
Dialog DLL: Parameter: DCM.DLL -pCM3	Dialog DLL: Parameter: TCM.DLL -pCM3	
OK Car	ncel Defaults Help	

And finally, the Utilities window. The ST-Link Debugger needs to be selected. Note the Settings button. Click Settings to open additional windows as shown on the next page.

🗷 Options for Target 'STM32F100RB Flash'							
Device Target Output Listing User C/C++ Asm Linker Debug Utilities							
Configure Flash Menu Command							
Use Target Driver for Flash Programming							
ST-Link Debugger 🚤 💽 Settings 🔽 Update Target before Debugging							
Init File: Edit							
Command: Arguments: Run Independent							
Configure Image File Processing (FCARM):							
Output File: Add Output File to Group:							
Startup							
Image Files Root Folder:							
OK Cancel Defaults Help							

Clicking Settings in the Utilities window and its Debug tab gives this. Note that Port must be set to SW (single wire). No embedded board was connected when this screen shot was taken and thus the No ST-LINK message. That should change when a board is connected.

Cortex-M Target Driver Setup						
Debug Trace Flash Download						
Debug Adapter	SW Device					
Unit:	Error					
Serial Number:	SWDIU No ST-LINK detected					
HW Version:	Down					
Firmware Version:	Automatic Detection ID CODE:					
> Port: SW -	C Manual Configuration Device Name:					
Max Clock: 1MHz 💌	Add Delete Update IR len:					
Debug	Cache Options Download Options					
Connect: Normal Rese	t Autodetect Cache Code Verify Code Download					
Reset after Connect	Cache Memory Download to Flash					
	OK Cancel Apply					

The trace tab. Default values should work.

Cortex-M Target Driver Setup	
Debug Trace Flash Download	
Core Clock: 10.000000 MHz Trace Port Serial Wire Output - UART/NRZ SWO Clock Prescaler: 5 Autodetect SWO Clock: 2.000000 MHz	Trace Enable Timestamps Imable Prescaler: PC Sampling Prescaler: 1024*16 ▼ Imable SLEEP: SLEEP: Sleep Control FOLD: Folded Instructions Image: Structure FOLD: Image: Structure Image: Structure Image: Structure Folded Image: Struct
ITM Stimulus Ports 31 P Enable: 0xFFFFFFFF Immodel Port Privilege: 0x00000008 Port Advanced settings Ignore packets with no SY Immodel Overwrite CYCCNT	ort 24 23 Port 16 15 Port 8 7 Port 0 Image: Comparison of the property of the p
	OK Cancel Apply

On the Flash Download tab, if there isn't an entry showing in the Programming Algorithm box, click on the Add button which opens another window, scroll down to find STM32F10x Med-density Flash, click on it to select, and then click Add..

Cortex-M Target Driver Setup				
Debug Trace Flash Download				
Download Function C Erase Full Chip E Erase Sectors C Do not Erase	 ✓ Program ✓ Verify ✓ Reset and F 	BAM for Start:	Algorithm 0x20000000 Size: 0x0800	
Programming Algorithm				
Description	Device Size	Device Type	Address Range	
STM32F10x Med-density Flash	i 128k	On-chip Flash	08000000H - 0801FFFFH	<
		Start:	Size:	
	Add	Remove		
			OK Cancel	Apply

- 10) On the Project tab, select Build Target (or use the button on a tool bar, or press F7). Your project should build unless there are errors in your source file.
- 11) Connect hardware, download, and test your program.