

User Guide document for FRDM-KL25Z Evaluation board, Keil-MDK5 software development environment and mbed development platform

Course Instructor: Dr.Otmane Ait Mohamed Prepared by: MohammedHossein Askari Hemmat COEN 6711 Concordia University

Winter 2014



Note: Before attaching the FRDM-KL25Z board to your computer, please read the P&E OpenSDA Firmware section on this document.

Contents

1	Introduction	1
2	Step1: Keil MDK5 Installation process	1
3	Step2: P&E OpenSDA Firmware	3
4	Step3: Attaching FRDM-KL25Z board to PC	3
5	Step4: Blinky Project 5.1 Create a μVision Project 5.2 Compiling and Programming 5.3 Debugging	4 4 6 8
6	mbed development platform:6.1Getting started:6.2Make an mbed project:	9 9 9
7	You can find more information on creating an mbed program in here.	10
8	Useful Links:	11

1 Introduction

This document guides you through the installation process of the following softwares and drivers on your PC:

- KEIL MDK5
- P&E OpenSDA Firmware

Also at the end of this document, you can find a thorough description of the Blinky project. Starting from building a simple project and adding the necessary libraries/drivers to the project to compiling the source code and debugging the project while it is executing on the FRDM-KL25Z platform.

You can find more information about the FRDM-KL25Z platform on the Freescal website. Also, you can find many useful information regarding this platform on mbed and element14 websites.



Figure 1: FRDM-KL25Z board pin description

2 Step1: Keil MDK5 Installation process

Keil MDK5 is a complete software development environment for almost all of the ARM based microcontrollers. MDK-ARM V5 is the latest version which were released in 2013. The installation process is pretty straight forward:

First you need to download the latest version of MDK-ARM V5 from KEIL website. Before downloading, you will be asked to fill out the installation form.



Figure 2: KEIL Product page

After you downloaded the file, you can run the mdk501.exe by double clicking on it. The program will install on your computer after several minutes. When you have successfully installed the program, the Pack Installer will pop-up where you can add the CMSIS library and some useful examples specially written for your specific platform to your program.

	' 1	ARM			
🙆 Pack Installer					
File Packs Window Help					
€ Board: All	- Categ	ory: All v Installed Examples			
Packs Examples					
Pack	Action	Description			
Keil::EFM32TGxxx_DFP	🚸 Install	Energy Micro EFM32 Tiny Gecko Series Device Support			
Keil::EFM32WGxxx_DFP	🚸 Install	Energy Micro EFM32 Wonder Gecko Series Device Support			
Keil::EFM32ZGxxx_DFP	🚸 Install	Energy Micro EFM32 Zero Gecko Series Device Support, Drivers and Examples			
Keil::FM3Basic_DFP	🚸 Install	Spansion FM3 Basic Series Device Support			
Keil::FM3HighPerformanc	🚸 Install	Spansion FM3 High Performance Series Device Support and Examples			
Keil::FM3LowPower_DFP	🚸 Install	Spansion FM3 Low Power Series Device Support			
Keil::FM3UltraLowLeak_DFP	🚸 Install	Spansion FM3 Ultra Low Leak Series Device Support			
Keil::FM4_DFP	🚸 Install	Spansion FM4 Series Device Support			
Keil::Kinetis_K10_DFP	🚸 Install	eescale Semiconductor Kinetis K10 Series Device Support			
Keil::Kinetis_K20_DFP	🚸 Install	eescale Semiconductor Kinetis K20 Series Device Support and Examples			
Keil::Kinetis_K30_DFP	🚸 Install	Freescale Semiconductor Kinetis K30 Series Device Support			
Keil::Kinetis_K40_DFP	🚸 Install	Freescale Semiconductor Kinetis K40 Series Device Support			
Keil::Kinetis_K50_DFP	🚸 Install	Freescale Semiconductor Kinetis K50 Series Device Support			
Keil::Kinetis_K60_DFP	🔅 Install	Freescale Semiconductor Kinetis K60 Series Device Support and Examples			
Keil::Kinetis_K70_DFP	🚸 Install	Freescale Semiconductor Kinetis K70 Series Device Support			
Keil::Kinetis KExx DEP	🕸 Install	Freescale Semiconductor Kinetis KExx Series Device Support			
Keil::Kinetis_KLxx_DFP	🚸 Install	Freescale Semiconductor Kinetis KLxx Series Device Support and Examples			
Keil::Kinetis_KMxx_DFP	Varian 1	Do one			
Keil::LPC1700_DFP	First Relea	ase version of KL Device Family Pack. t and Examples			
Keil::LPC1800_DFP	🚸 Install	NXP LPC1800 Series Device Support, Drivers and Examples			
Keil::LPC800_DFP	Install	NXP LPC800 Series Device Support			
Keil::MDK-Middleware	🔶 Up to date	Keil MDK-ARM Professional Middleware for ARM Cortex-M based devices			
Keil::SAM3_DFP	🚸 Install	Atmel SAM3 Series Device Support and Examples			

Figure 3: Pack Installer

After installing your desired libraries and Example package from Pack Installer window, you can then run the μ Vision program. You can find how to make a simple project on the μ Vision User's Guide

3 Step2: P&E OpenSDA Firmware

P&E OpenSDA Firmware provides the drivers for programming, debugging and communicating with FRDM-KL25Z platform. Without this Firmware you are not able to program your device. You can find the latest version of this software on the P&E micro website . You need to download and install the PEDrivers_install.exe under the Windows USB Drivers box.

The FRDM-KL25Z platform provides you an onboard debugging capability. On the other hand, OpenSDA provides you a UART communication channel with the onboard processor(KL25Z). So P&E OpenSDA Firmware will install not only the programming/debugging driver but also it will create OpenSDA-CDC serial port on your computer. This driver will add a virtual COM port on your computer when ever you attach the board to one of your USB ports.

4 Step3: Attaching FRDM-KL25Z board to PC

So far, we have installed the software development environment and P&E OpenSDA Firmware. At this point, we are all set to attach the board to the computer. Windows will automatically find and install all the proper driver for the board.

Driver Software Installation	_	23
Installing device driver software		
USB Composite Device PEMicro/Freescale - CDC Serial Port PEMicro/Freescale Debug App	Ready to use Searching Windows Update Searching Windows Update	
Skip obtaining driver software from Windo	ws Update might take a while.	
		Close

Figure 4: Driver software Installation

Since there is a pre-installed bootloader on the FRDM-KL25Z platform, after the first installation of the board, windows will detect FRDM-KL25Z as a storage device. So a new storage device will be added to your PC. You can drop your executable codes into this drive and OpenSDA will program the board. Since we are using the KEIL MDK5 as our programming environment, it will be more convenient to program the board using KEIL MDK5 as well. For this purpose, we need to change the pre-installed bootloader. First you need to unplug the board from the USB port and then while you are pressing the reset button re-plug the board to one of your USB ports. This time a Bootloader drive will appear on your computer.

C-12 1 C + 1000	and Rest countries and	- 0 - X
Compute	ter > • • • •	h Computer 🖉
Organize 🔻 System p	properties Uninstall or change a program Map network drive Open Control Panel	5 × 🗊 🔞
☆ Favorites	Hard Disk Drives (3)	
📰 Desktop 👔 Downloads 💔 Dropbox	Lacd Disk (C) New Volume (D) New Volume (E) 5.1 68 free of 97.5 68 S 120 68 free of 270 68 97.5 68 68	
🔛 Recent Places	Devices with Removable Storage (2)	
Call Libraries Courments J Music S Pictures Videos Videos	BOTLOADER (G)	
🕵 Computer		
🚢 Local Disk (C:)		
New Volume (D:)		
BOOTLOADER (G:)		
年 Network		

Figure 5: Driver software Installation

Now you just need to drag and drop the DEBUG-APP_Pemicro_v108.SDA on this folder so the next time you attach the board, you will be able to program it via KEIL MDK5.

5 Step4: Blinky Project

5.1 Create a μ Vision Project

In this section you can find how to make a simple project in Keil-MDK. In Keil-MDK Select Project> New μ Vision project. Find MKL25ZL128xxx4 in the target device and click OK.

Select Device for Target 'Target 1'	x
CPU Vendor: Freescale Semiconductor Device: MKL25Z128xx4 Toolset: ARM Search: Description:	
Core features - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ core (up to 48MHz CPU Clock) - 32bit ARM Cortex-M0+ corte	* E
OK Cancel He	elp

In the new window, check the startup option under the Device tab and Core option under the CMSIS tab and then select OK.

oftware Component	Sel.	Variant	Version	Description	
🔶 CMSIS				Cortex Microcontroller Software Interface Components	
CORE	1		3.20.0	CMSIS-CORE for Cortex-M, SC000, and SC300	
OSP			1.4.1	CMSIS-DSP Library for Cortex-M, SC000, and SC300	
🗉 💠 RTOS (API)			1.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300	
🔶 Device				Startup, System Setup	
Ø Startup	T		1.0.0	System Startup for Freescale KLxx Series	
Drivers				Unified Device Drivers	
💠 File System		MDK-Pro	5.0.4	File Access on various storage devices	
Graphics		MDK-Pro	5.22.1	User Interface on graphical LCD displays	
Network		MDK-Pro	5.0.4	IP Networking using Ethernet or Serial protocols	
🔶 USB		MDK-Pro	5.0.4	USB Communication with various device classes	
lidation Output		Descript	ion		
lidation Output		Descript	ion		
fidation Output		Descript	ion		

Now we need to write a C code to control the RGB LED on the board. Create a new file by selecting New under the File tab. Now save the file blinky with .C extension. Make sure that you have added the .c extension at the end of your file name.

		Project • •	Search Project	5
Organize New for	older			≡ - 0
Desktop Downloads Dropbox Cropbox Cro	Name RTE Blinky.build_Jog.htm Blinky.uvguit.HOSSEIN Blinky.uvguit.HOSSEIN Blinky.uvgptx Blinky.uvoptx Blinky.uvoptx Blinky.uvoptx	Date modified 12/24/2013 8:40 PM 12/24/2013 8:41 PM 12/24/2013 8:36 PM 12/24/2013 8:40 PM 12/24/2013 8:36 PM 12/24/2013 8:40 PM 12/24/2013 8:40 PM 12/24/2013 8:40 PM	Type File folder Firefox HTML Doc HOSSEIN File HOSSEIN File UVOPT File UVOPT File µVision4 Project uVision9 Project	Size 1 KB 68 KB 68 KB 5 KB 7 KB 0 KB 17 KB
 Videos Homegroup File name: Save as type: 	inby.c Files (*.*)			

This C file will be later compiled so we need to add it to our project:



Now you should be able to find your C file under the Source Group 1 tab. Copy and paste the following code into blinky.c. Make sure that you have Delay.h file in your project folder.

```
#include <MKL25Z4.H>
#include "Delay.h"
const uint32_t led_mask = 1UL << 1;</pre>
/*-----
 MAIN function
                                                                        ----*/
 *-----
int main (void) {
 SystemCoreClockUpdate();
                                           /* Get Core Clock Frequency */
 SysTick_Config(SystemCoreClock/1000); /* Generate interrupt each 1 ms */
 SIM->SCGC5 |= (1UL << 12);
                                   /* Enable Clock to Port D */
 PORTD \rightarrow PCR[1] = (1UL \ll 8);
                                                /* Pin PTD1 is GPIO */
 FPTD->PDOR = led_mask;
                                /* switch Blue LED off */
 FPTD->PDDR = led_mask;
                                /* enable PTD1 as Output */
 while(1) {
              FPTD->PCOR = led_mask;
   Delay(500);
              FPTD->PDOR = led_mask;
   Delay(500);
 }
}
```

5.2 Compiling and Programming

In order to compile your code, simply click on "Rebuild all the target files" under the Project Tab. You should be able to compile the code without any errors or warnings.

20 21 E Project	m
Build Output	
Rebuild target 'Target 1' compiling blinky.c assembling startup_MKL2524.s compiling system_MKL2524.c linking .\Blinky.axf" - 0 Error(s), 0 Warning(s).	ata=1124

This C code still cannot be run on the microcontroller. We need to generate the Intel HEX file out of this code:

D:\Keil\Freedom Board\Blinky\Project\Bl	inky.uvproj - µVision			
File Edit View Project Flash Debu	g Peripherals Tools SVCS Window H	lelp		
🗋 🗋 😹 🕼 🕹 🛍 🖉 🤊 🤊	🖛 🔿 🕐 🎘 🎘 🎘 🎼 🛱 🖉	//# 🖄 💌 🖬 🐨 🍕 💷 🔍		
🗇 🕮 🕮 🧼 📇 🛛 🌠 🛛 Target 1	🖌 🔶 🕾 🖡 💸 🔁			
Project 📮 🖬	in blinky			▼ ×
Early Traget I Cover Group 1 Cover	<pre>is notuce SML2554 # 2 come uint22 t iet i 3 /*</pre>	ask = 1UL << 1; Coptions for Target Target I' Device Target Outor Analysis User C.C++ Aem Linker Debug Utilities Select Folder for Objects. Name of Executable: Binky P Create Executable: 'Binky 2 Debug Homation 3 P Jonate HEX Rie P Drove HEX Rie P Create Library: 'JBinky Jib	Ceate Bath Fie	,
Rebuild target 'Target 1' compling blinky.c assembling startup_MKLS54.s. compling system_MKLS54.s. linking Program Size: Code=1376 RO-da ".\Blinky.axf" - 0 Error(s),	 ta=224 RN-data=4 ZI-data=1124 O Marning(s).	4 ox Garcel Defaults	Heb	^
Server List				r
			PEMicro Debugger	L:10 C:1 CAP NUM SCRL OVR R/W

We need to set the OpenSDA as the programming and debugging tool. In the same window under the Debug tab select PEMicro Debugger and then select OK.

😨 Options for Target 'Target 1'	×
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities
C Use Simulator Settings	PEMicro Debugger Settings
	Stellaris ICDI
Initialization File:	Initializatic ULINK / J-TRACE Cortex Debugger
Edit	NULink Debugger ≡ Edit
Restore Debug Session Settings	Restore CMSIS-DAP Debugger
Watch Windows & Performance Analyzer	Watch Vilhoows
	Memory Display
CPU DLL: Parameter:	Driver DLL: Parameter:
SARMCM3.DLL	SARMCM3.DLL
Distan Di Lu Brannatan	Dista Di L. Domenton
DARMCM1.DLL PCM0+	TARMCM1.DLL pCM0+
3 ОК Сап	cel Defaults Help

Now we just need to program the board with the generated hex file. Under the Flash tab, select Download option. You need to specify the target device that you are going to program. Select KL25Z128M4 from the "Choose Device" window and then click OK.

Choose Device	
KL 05Z32M4	
KL02Z8M4	- 1
KL02Z16M4	
KL02Z32M4	
KL14Z32M4	
KL14Z64M4	
KL15Z32M4	
KL15Z64M4	
KL15Z128M4	
KL16Z32M4	
KL16264M4	
KL162128M4	
KL162256M4	
KL24Z32M4	
KL24204M4	
KL25252M4	=
KI 257128M4	
KL26Z32M4	
KL26Z64M4	
KL26Z128M4	
KL26Z256M4	
KL34Z64M4	
KL36Z64M4	
KL36Z128M4	
KL362256M4	
KL46Z128M4	
KL46ZZ36M4	
KM147C4M5	
KM147128M5	
KM32764M5	
KM33Z64M5	
1	*
V UN 👗 Lancel	

On the opened window (PEMICRO Connection Manager) make sure that you have selected "OpenSDA Embedded Tower Debug USB-Port" from the Interface box. Also make sure that you have selected "Use SWD reduced pin protocol for communications" and then hit the Connect button in order to program the board. Press reset button on the board to let the program execute.

elected to displa and click OK. port and Interface Type	ay this dialog on s	tartup. Specify	communications
port and Interface Type			
ISB Multilisk USB Mult			
ISB Multilisk, USB Mul			Add LPT Port
556 Malainik, 056 Mal	tilink FX, or Embedded OS	Jtag	Refresh List
ARM Cable, Connected ISB Multilink, LISB Mult	via Parallel Port or BDM L tilink FX, or Embedded OS	ightning Itag	(Heln2)
Cyclone - Serial Port		and g	TIOPT
Cyclone - Ethernet Port Cyclone - USB Port			
FraceLink - USB Port	d.		
DpenSDA Embedded T	ower Debug - USB Port		1
			-
unication Speed			
Port wait states : IO DE	LAY ONT =	0	
-			
peed = (0) : Multilink	k = 1.00Mhz , Multilink FX	(= 25.00Mhz , OSJT	'AG = 0.25Mhz 💌
D reduced pin protocol	for communications	2	
d Pue Fraguanau (Far pr	rogramming)	2	
tect	rogrammingj		
ernal Bus frequency (FF	REQ) in Hz =	0 (Decimal)	
ns			
ter Reset and before co	mmunicating to target for	0 m	iilliseconds (decimal).
ol for Cyclone / TraceLi	ink / Multilink Universal F×	<	
power to target	Regulator Output Volt	age Power Dow	n Delay 250 mS
ff target upon software (exit 3V 🖵	Power U	p Delay 1000 mS
nact (Basat)	3 Hotevr		Abort
meccinasey	<u>II</u> U(Syl		
	yclone - Serial Port yclone - Serial Port yclone - USB Port (aceLink	yclone - Serial Port yclone - USB Port TraceLink - USB Port TraceLink - Litternet Fort TraceLink - Litternet Fort IpensDA Embedded Tower Debug - USB Port anication Speed Port wait states : IO_DELAY_ONT = ipeed = [(0) : Multilink = 1.00Mhz , Multilink PA BDM SPEED = 0 D reduced pin protocol for communications all Bus Frequency (FOR EQ) in Hz = main Bus frequency (FREQ) in Hz = 18 19 19 10 Cyclone / TraceLink / Multilink Universal PP power to target Frequency (Reset) 10 Cyclone / TraceLink / Multilink Universal PP power to target TraceLink / Multilink Universal PP TraceLink / Multilink / Multilink Universal PP TraceLink / Multilink / Multilink / Multilink Universal PP TraceLink / Multilink / Multili	Jyckone - Serial Port Jyckone - USB Port Jyckone - USB Port TraceLink - USB Port Inclusion Speed Port wait states : IO_DELAY_CNT = 0 Speed = [0] : Multilink = 1.00Mhz , Multilink FX = 25.00Mhz , OSJI D reduced pin protocol for communications 1 bus Frequency (FOF programming) tect emal Bus frequency (FREQ) in Hz = 0 for Cyclone / TraceLink / Multilink Universal FX- power to target Regulator Output Voltage Power Dow ft Arget upon software exit 3/ * Power U 3/ * Power U 3/ * Power U 3/ * Power U

5.3 Debugging

To debug your code while it is executing on the board press Ctrl+F5 or selet start/stop under the Debug tab.

D:\Keif\Freedom	n Board\Jed2\Jed2.uvpr	xojx - µVision	
File Edit View	Project Flash Del	ebug Peripherals Tools SVCS Window Help	
🗋 🐸 🖬 🥔	8 in 🔁 🤊	◇ ← ◇ 隆 熟 熟 律 準 准 振 🦉 📃 🔍 🖉 💽 ◆ ◇ 🔗 🍓 💷 🔦	
88 🗉 📀	0 0 0 0		
Registers	# 🖲	Dissistembly	a 💽
Register	Value	7: SystemCoreClockUpdate(); /* Get Core Clock Frequency */	^
Core		8: SvFick Confide SystemCoreLineRuppate (xC00000405): /* Generate interrupt each 1000 ms */	
	0x20000068	0x0000016C 210A MOVS r1,#0x0A	
R2	0x20000068	<pre>4 00000016E 4830 LDR r0.16c.#1921 ; M0x00000230 </pre>	E.
	0x00000640	Pi led2.c Pi startup MKL2524.s Pi system MKL2524.c	• ×
R5	0x20000004	1 #include <mkl2524.h></mkl2524.h>	
- R7	0x00000000	<pre>2 const uint32_t led_mask = 1UL << 1;</pre>	
RS	0xFFFFFEB	4 MAIN function	
- R10	0x00000640	5 **/	
R11	0x00000640	<pre>> 7 SystemCoreClockUpdate(); /* Get Core Clock Frequency */</pre>	
R12 R13 (SP)	0x20000468	8 SysTick Config(SystemCoreClock/10); / / Generate interrupt each 1000 ms */	
R14 (LR)	0x00000157	9 SIM->SUGUS [= [10L << 12]; /* Inable Clock to Part D */ 10 PORTD->PCR[] = [10L << 8]; /* Pin FTD1 is GPIO */	
RT5 (PC) €−xPSR	0x00000168 0x01000000	11 FFID->FDOR = led_mask; /* switch Blue LED off */	
Banked		12 FPID-PFDUK = led_mask; /* enable FID1 as Output */	
B System		14 while(1) (
Mode	Thread	15 FYID-SPCOR = led_mask; 16)	
Stack	Privileged MSP	17)	
		18	
Project Rec	aisters		b.
Command		Call Stack + Locals	a 🖬
		Name Location/Value Type	
Breakpoint En	acountered.	□ main 0x0000168 int f()	
(×	
>			
ASSIGN BreakD	Disable BreakEna	able BreakKill BreakList BreakAccess COVERAGE DEFINE DIR 🛛 🕼 Call Stack + Locats 🔚 Memory 1	

Use F11 and F5 to execute the program step by step. This program will blink LED

6 mbed development platform:

mbed development platform is an online development kit which supports many ARM based platforms from different vendors. Fortunately, FRDM is one of the well supported platform by mbed. Before going to the next section, make sure that you have created an account on the mbed website .

6.1 Getting started:

To get started, you need to re-program the BOOTLOADER of the microcontroller in order to be able to program the board using the .bin files. You can download this BOOTLOADER from here. Using this new BOOTLOADER, re-boot the board according to the section four of this document. You can find more information on how to re-boot the board in here. If you re-boot the board properly, you should be able to see the "MBED" Folder on your computer after you re-plugged the board.



6.2 Make an mbed project:

Once you log into your mbed account, click on the Compiler button in order to use the mbed Compiler:

A Handbook Cookbook	Platforms Components Code	e Questions Forum	Dashboard Compiler			
mbed	Search mbed.org	Go	Hi, 💩			
Explore	Getting Started	Prototype	Production			
Development Platform	for Devices					
The mbed development platform is products based on ARM microcon	s the fastest way to create trollers.	C027-C20/U20/G35				
The project is being developed by contributions of the global mbed D	ARM, its Partners and the eveloper Community.		things starter kit			
Find out why you should base you powered product on the mbed plat	r next ARM microcontroller tform »	C027 mbed enabled Int	ernet of the second			

In the new window, right click on "My Programs" and select "New Program". In the "Create New Program" window, make sure that you have selected the "Blinky LED Hello World" option from the template menu. Provide a name on the Program Name box and click on OK:

mbed Compiler - Workspace Management								
🎦 New 👻 Disport 🗟 Save Al 🖽 Comple 🗸 @ Comml × 〇 Revisions 🗢 〜 為 後 🌾 Ш Help								
Program Workspace	Workspace Management							
My Programs	Manage your Program Workspace Workspace Choose which programs are open in your Program Workspace.							
Listing all programs in your Program Workspace								
	Name Tago Modified Description							
	Your Program Workspace is empty. You can import a program or greate a new one.							
	Create new program M Create new program for Freescale K1252 in your workspace. You can always change the platform of this program are not emplate M Image: Please specify program name and template Image: Please specify program name and template Image: Please specify program name and template Image: Program Name: Test Image: Please specify program name and template Image: Please specify program name and template Image: Program Name: Test Image: Please specify program to be created in your workspace. Image: Please specify program name and template Image: Test Image: Test Image: Test Image: Test Image: Test Imag							

Since you have selected a "Blinky LED Hello World" template for your project, the main code is already generated by the program and you don't need to add anything to the project. On the left hand side of the compiler, double click on the main.cpp file and review the generated code. To generate a .bin file from this project, you need to compile the code. Click on the "Compile" button:

mbed Compiler - /Test/main.cpp						
🎦 New 👻 🎦 Import 🛛 🗟 Save 📮	Save Al 🖉 Comple 🔽 🕭 Commit 👻 🕜 Revisions 🛛 🗠 🕼 🍓 🐁 🔛 Help				Freesci	ile KL25Z 🦔
Program Workspace	main.cpp x Compile and Download - Ctrl-D				[Expand
E ⊘ in yn frogans C II methogo R ⊘ nbed	<pre>1 function "model.h" 2 Digitalion myled(IDD1); 3 Digitalion myled(IDD1); 4 myled ";; 4 myled ";; 5 mild(0,7); 1 myled ";; 5 mild(0,7); 1 myled ";; 5 mild(0,7); 1 myled ";; 6 mild(0,7); 1 myled ";; 7 myled "; 7</pre>					× ×
	Compile output for program: Test			Errors: 0	Warnings: 0	Infos: 1
	Description	Error Number	Resource	In Folder	Locat	an
	♥ AAASSE		LOUIS LACIONS			
Ready.					INS	

The software will compile your code and if the compilation was successful, it will provide you a link to download the .bin file. Save the .bin file on the "MBED" folder on your computer and press the reset button on the board to allow the execution of your program on the microcontroller.

7 You can find more information on creating an mbed program in here.

8 Useful Links:

- Creating a μ vision Projects
- MDK-ARM5
- FRDM-KL25Z Product Page
- MDK5 Software Packs
- FRDM-KL25Z User's Manual