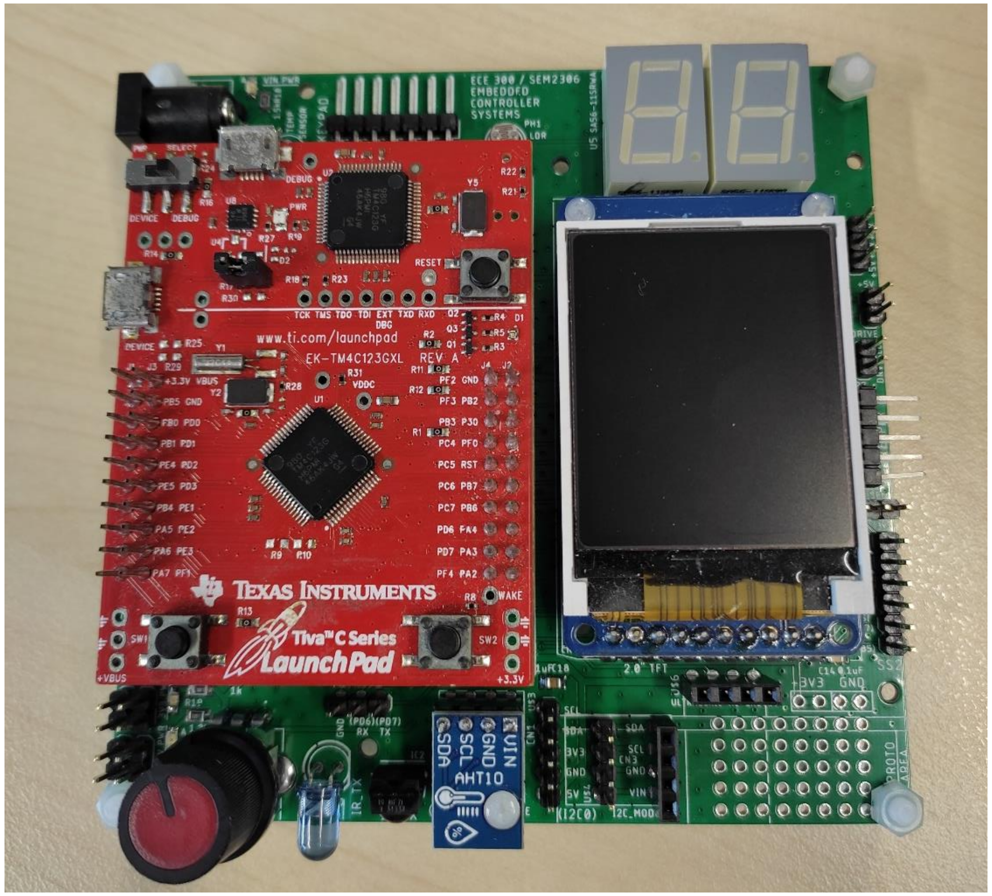
Lab 1: KEIL µVision, SysTick Timer & PLL

CS397

Week 1 Lab Session

- Lab components & equipment.
  - Tiva Baseboard & components:
    - Tiva LaunchPad
    - LCD display (mounted on baseboard)
    - Micro USB cable
  - Digilent Analog Discovery 2.
- Items to be returned at end of module.
- Keep items (and boxes) together & safely.

Development Boards



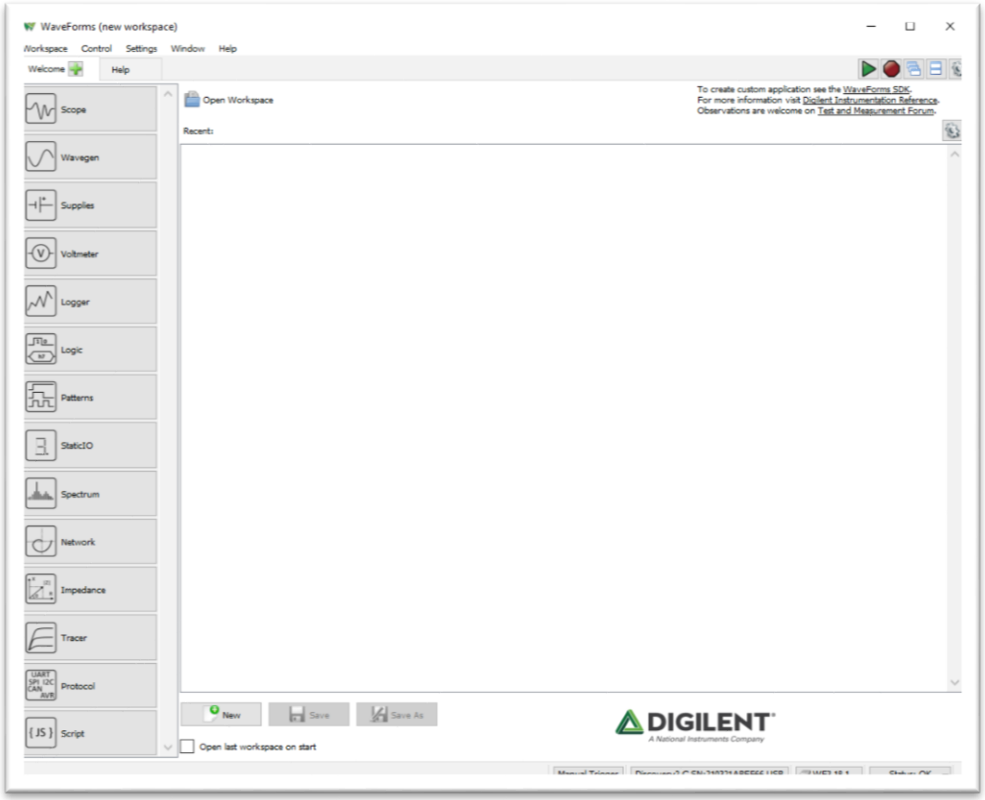
Tiva Baseboard (mounted)

Digilent Analog Discovery

CS397

Digilent Analog Discovery 2

- **Analog Discovery 2** is a USB-based tool that can function as an oscilloscope, logic analyzer, power supply.
- More details at Digilent website: https://digilent.com/shop/analog-discovery-2-100ms-s-usb-oscilloscope-logic-analyzer-and-variable-power-supply/.
- Download & install Waveforms software from Moodle.



To create custom application see the WaveForms SDK.
For more information visit Digilent Instrumentation Reference.
Observations are welcome on Test and Measurement Forum.

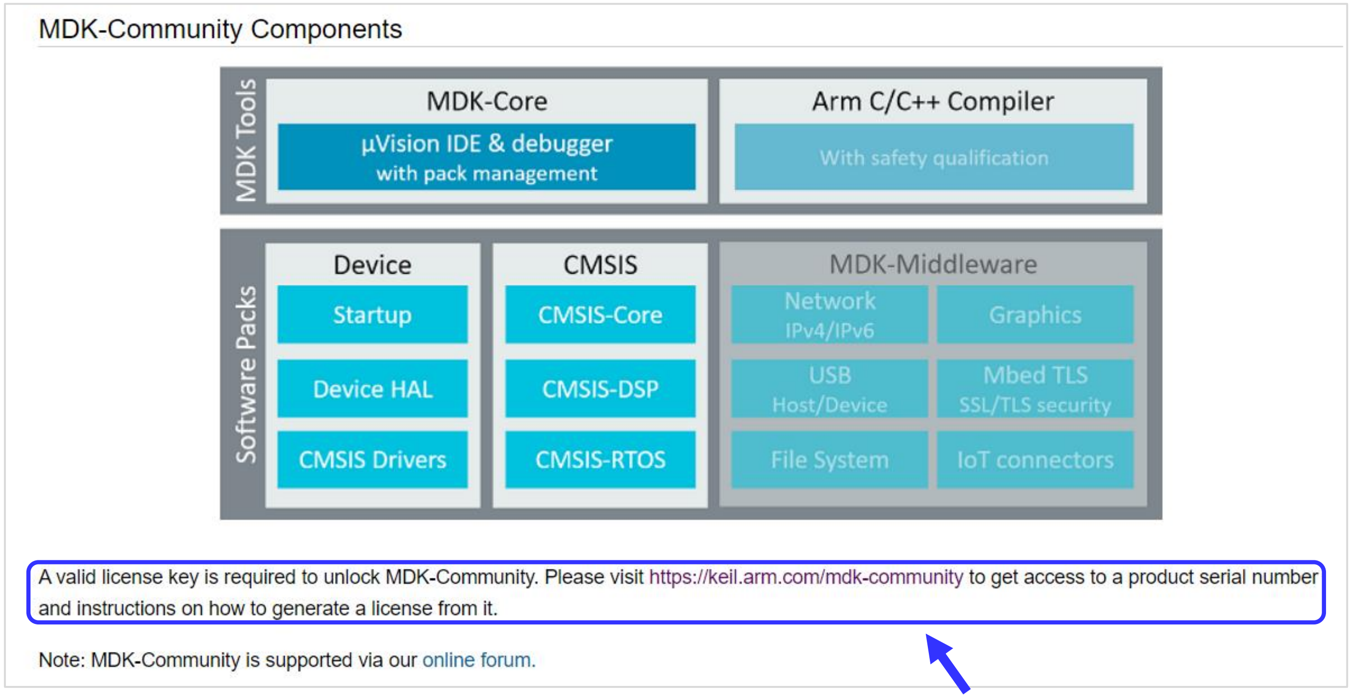
[Waveforms Software]

KEIL µVision Installation

CS397

KEIL Community Edition

- KEIL Community edition is a full-feature IDE for non-commercial use (education, hobby, ..). There is no compiled code size restriction as compared to the Lite version.
- Download link: https://www2.keil.com/mdk5/editions/community.
- Register for an account to acquire valid license key: https://www2.keil.com/mdk5/editions/community.


MDK-Community Components

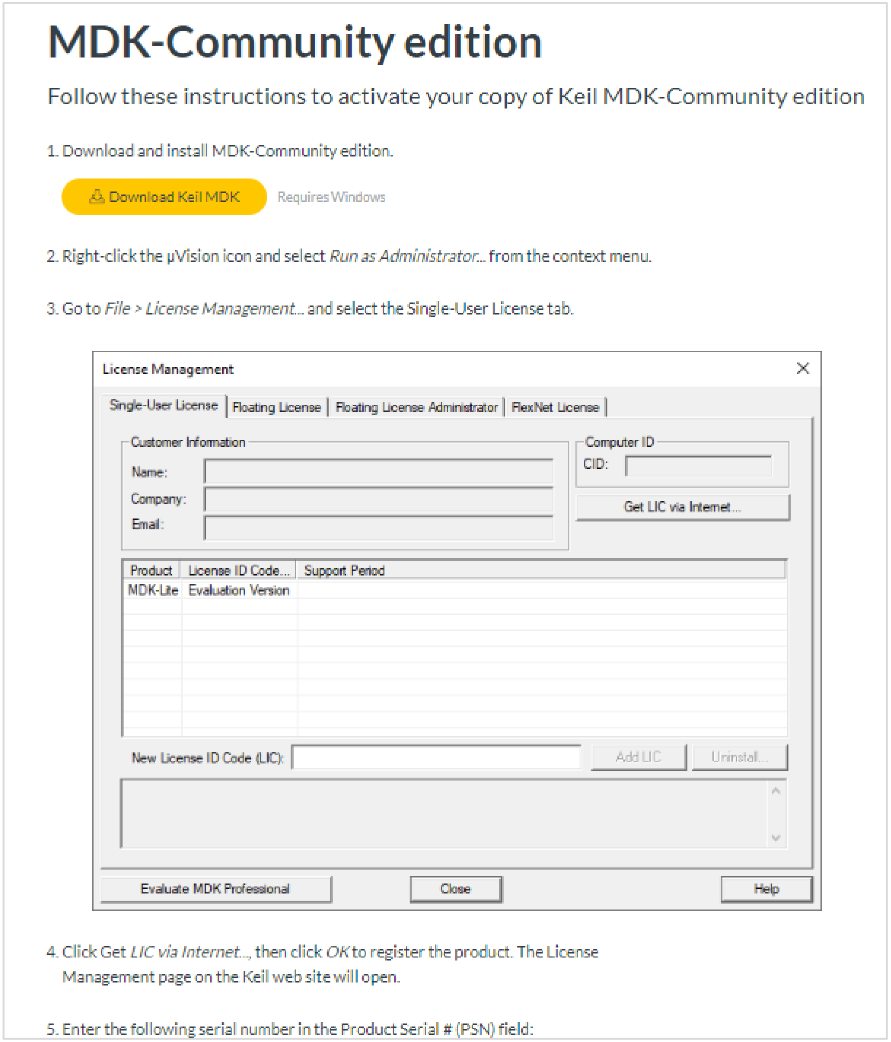
A valid license key is required to unlock MDK-Community. Please visit https://keil.arm.com/mdk-community to get access to a product serial number and instructions on how to generate a license from it.

Note: MDK-Community is supported via our online forum.

Download KEIL Community Edition

Link to download KEIL Community edition (requires account): https://www.keil.arm.com/mdk-community/.

Current version is **v5.36**.



MDK-Community edition

Follow these instructions to activate your copy of Keil MDK-Community edition

1. Download and install MDK-Community edition.

Download Keil MDK   Requires Windows

2. Right-click the µVision icon and select *Run as Administrator...* from the context menu.

3. Go to *File > License Management...* and select the Single-User License tab.

License Management

Single-User License | Floating License | Floating License Administrator | FlexNet License

Customer Information
Name:
Company:
Email:

Computer ID
CID:

Get LIC via Internet...

Product	License ID Code...	Support Period
MDK-Lite	Evaluation Version	

New License ID Code (LIC):   Add LIC   Uninstall...

Evaluate MDK Professional   Close   Help

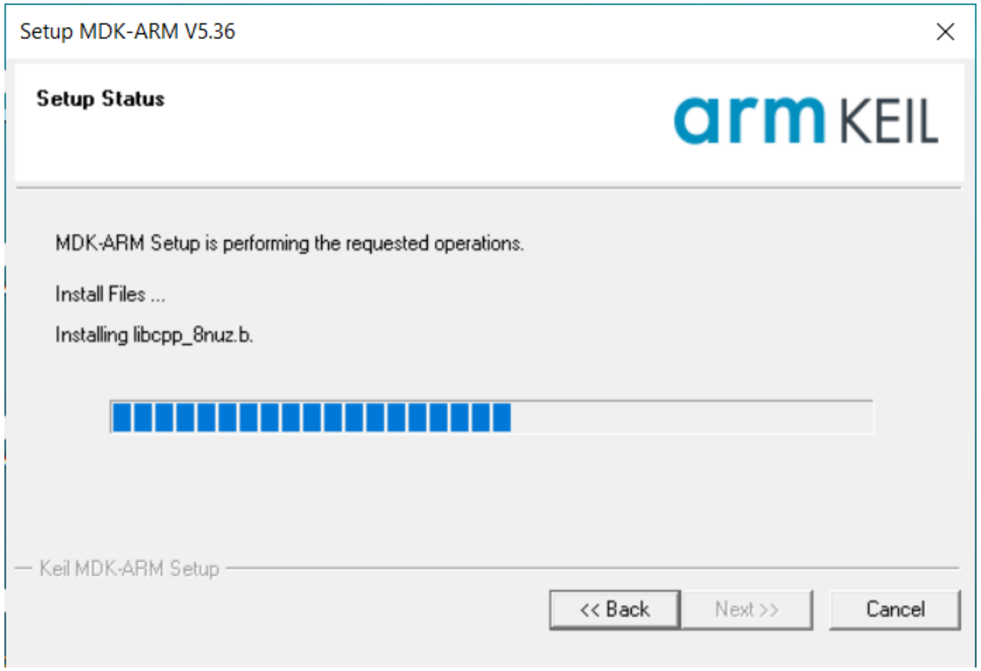
4. Click Get *LIC via Internet...*, then click *OK* to register the product. The License Management page on the Keil web site will open.

5. Enter the following serial number in the Product Serial # (PSN) field:

Install KEIL Community Edition

Install KEIL MDK.

Current version is **v5.36**.


Setup MDK-ARM V5.36

Setup Status

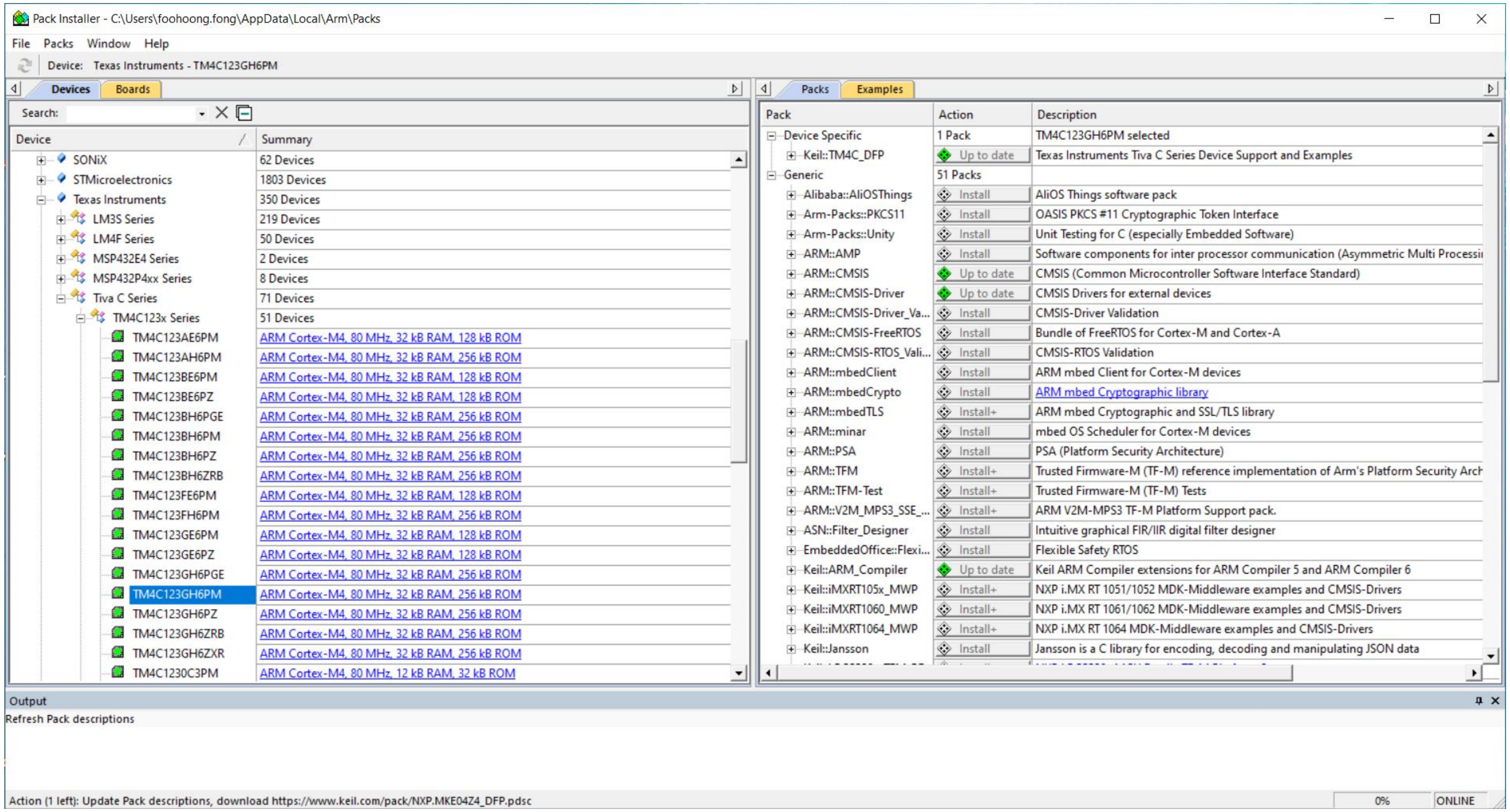
MDK-ARM Setup is performing the requested operations.

Install Files ...

Installing libcpp_8nuz.b.

Update Pack Installer

In Pack Installer, select the microcontroller we are using:
Texas Instruments, Tiva C series, TM4C123GH6PM.



Pack Installer - C:\Users\foohoong.fong\AppData\Local\Arm\Packs

File Packs Window Help

Device: Texas Instruments - TM4C123GH6PM

Devices | Boards

Search:

Device	Summary
SONiX	62 Devices
STMicroelectronics	1803 Devices
Texas Instruments	350 Devices
LM3S Series	219 Devices
LM4F Series	50 Devices
MSP432E4 Series	2 Devices
MSP432P4xx Series	8 Devices
Tiva C Series	71 Devices
TM4C123x Series	51 Devices
TM4C123AE6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 128 kB ROM
TM4C123AH6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123BE6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 128 kB ROM
TM4C123BE6PZ	ARM Cortex-M4, 80 MHz, 32 kB RAM, 128 kB ROM
TM4C123BH6PGE	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123BH6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123BH6PZ	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123BH6ZRB	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123FE6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 128 kB ROM
TM4C123FH6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123GE6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 128 kB ROM
TM4C123GE6PZ	ARM Cortex-M4, 80 MHz, 32 kB RAM, 128 kB ROM
TM4C123GH6PGE	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123GH6PM	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123GH6PZ	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123GH6ZRB	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C123GH6ZXR	ARM Cortex-M4, 80 MHz, 32 kB RAM, 256 kB ROM
TM4C1230C3PM	ARM Cortex-M4, 80 MHz, 12 kB RAM, 32 kB ROM

Packs | Examples

Pack	Action	Description
Device Specific	1 Pack	TM4C123GH6PM selected
Keil::TM4C_DFP	Up to date	Texas Instruments Tiva C Series Device Support and Examples
Generic	51 Packs	
Alibaba::AliOSThings	Install	AliOS Things software pack
Arm-Packs::PKCS11	Install	OASIS PKCS #11 Cryptographic Token Interface
Arm-Packs::Unity	Install	Unit Testing for C (especially Embedded Software)
ARM::AMP	Install	Software components for inter processor communication (Asymmetric Multi Processi
ARM::CMSIS	Up to date	CMSIS (Common Microcontroller Software Interface Standard)
ARM::CMSIS-Driver	Up to date	CMSIS Drivers for external devices
ARM::CMSIS-Driver_Va...	Install	CMSIS-Driver Validation
ARM::CMSIS-FreeRTOS	Install	Bundle of FreeRTOS for Cortex-M and Cortex-A
ARM::CMSIS-RTOS_Vali...	Install	CMSIS-RTOS Validation
ARM::mbedClient	Install	ARM mbed Client for Cortex-M devices
ARM::mbedCrypto	Install	ARM mbed Cryptographic library
ARM::mbedTLS	Install+	ARM mbed Cryptographic and SSL/TLS library
ARM::minar	Install	mbed OS Scheduler for Cortex-M devices
ARM::PSA	Install	PSA (Platform Security Architecture)
ARM::TFM	Install+	Trusted Firmware-M (TF-M) reference implementation of Arm's Platform Security Arch
ARM::TFM-Test	Install+	Trusted Firmware-M (TF-M) Tests
ARM::V2M_MPS3_SSE_...	Install+	ARM V2M-MPS3 TF-M Platform Support pack.
ASN::Filter_Designer	Install	Intuitive graphical FIR/IIR digital filter designer
EmbeddedOffice::Flexi...	Install	Flexible Safety RTOS
Keil::ARM_Compiler	Up to date	Keil ARM Compiler extensions for ARM Compiler 5 and ARM Compiler 6
Keil::iMXRT105x_MWP	Install+	NXP i.MX RT 1051/1052 MDK-Middleware examples and CMSIS-Drivers
Keil::iMXRT1060_MWP	Install+	NXP i.MX RT 1061/1062 MDK-Middleware examples and CMSIS-Drivers
Keil::iMXRT1064_MWP	Install+	NXP i.MX RT 1064 MDK-Middleware examples and CMSIS-Drivers
Keil::Jansson	Install	Jansson is a C library for encoding, decoding and manipulating JSON data

Output

Refresh Pack descriptions

Action (1 left): Update Pack descriptions, download https://www.keil.com/pack/NXP.MKE04Z4_DFP.pdsc 0% ONLINE

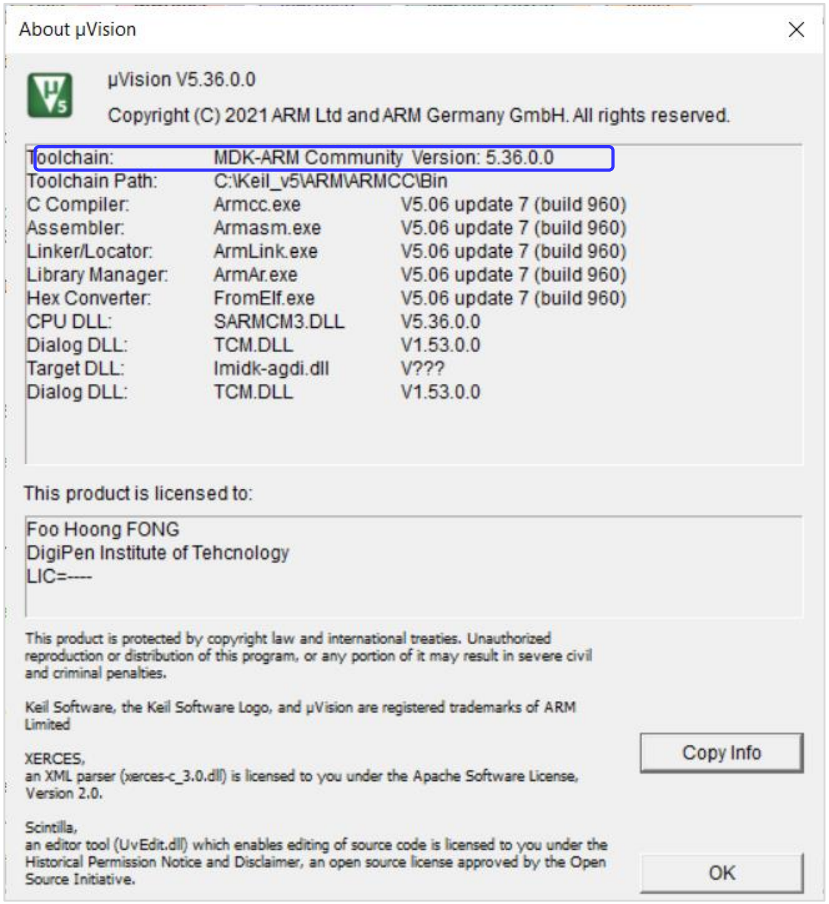
Update to Community License

The default would be a Lite edition. We want to change it to a Community edition.

Follow the instructions at https://www.keil.arm.com/mdk-community/ to add the new license number.

If all steps were completed correctly, you should have update to the **Community edition**.

Go to: 'Help' -> 'About µVison' to check your SW version.



About µVision

µVision V5.36.0.0

Copyright (C) 2021 ARM Ltd and ARM Germany GmbH. All rights reserved.

```
Toolchain:         MDK-ARM Community  Version: 5.36.0.0
Toolchain Path:    C:\Keil_v5\ARM\ARMCC\Bin
C Compiler:        Armcc.exe          V5.06 update 7 (build 960)
Assembler:         Armasm.exe         V5.06 update 7 (build 960)
Linker/Locator:    ArmLink.exe        V5.06 update 7 (build 960)
Library Manager:   ArmAr.exe          V5.06 update 7 (build 960)
Hex Converter:     FromElf.exe        V5.06 update 7 (build 960)
CPU DLL:           SARMCM3.DLL        V5.36.0.0
Dialog DLL:        TCM.DLL            V1.53.0.0
Target DLL:        lmidk-agdi.dll     V???
Dialog DLL:        TCM.DLL            V1.53.0.0
```

This product is licensed to:

Foo Hoong FONG
DigiPen Institute of Tehcnology
LIC=----

This product is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it may result in severe civil and criminal penalties.

Keil Software, the Keil Software Logo, and µVision are registered trademarks of ARM Limited

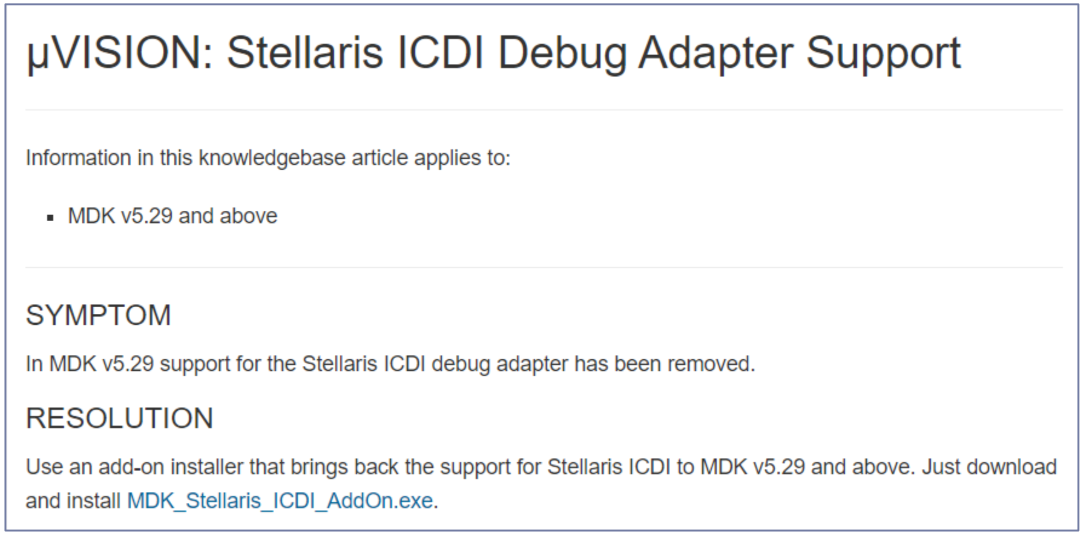
XERCES, an XML parser (xerces-c_3.0.dll) is licensed to you under the Apache Software License, Version 2.0.

Scintilla, an editor tool (UvEdit.dll) which enables editing of source code is licensed to you under the Historical Permission Notice and Disclaimer, an open source license approved by the Open Source Initiative.

If using µVision v5.29 or later ..

- In version 5.29 of KEIL µVison and later, support for the Stellaris ICDI debug adapter has been removed.
- Download the Stellaris ICDI add-on (http://www.keil.com/support/docs/4196.htm) and install after KEIL µVision installation.
- A version of the utility is also available on Moodle:' 'MDK_Stellaris_ICDI_AddOn.exe'.

Do this step if ICDI link could not be established.



µVISION: Stellaris ICDI Debug Adapter Support

Information in this knowledgebase article applies to:

- MDK v5.29 and above

SYMPTOM

In MDK v5.29 support for the Stellaris ICDI debug adapter has been removed.

RESOLUTION

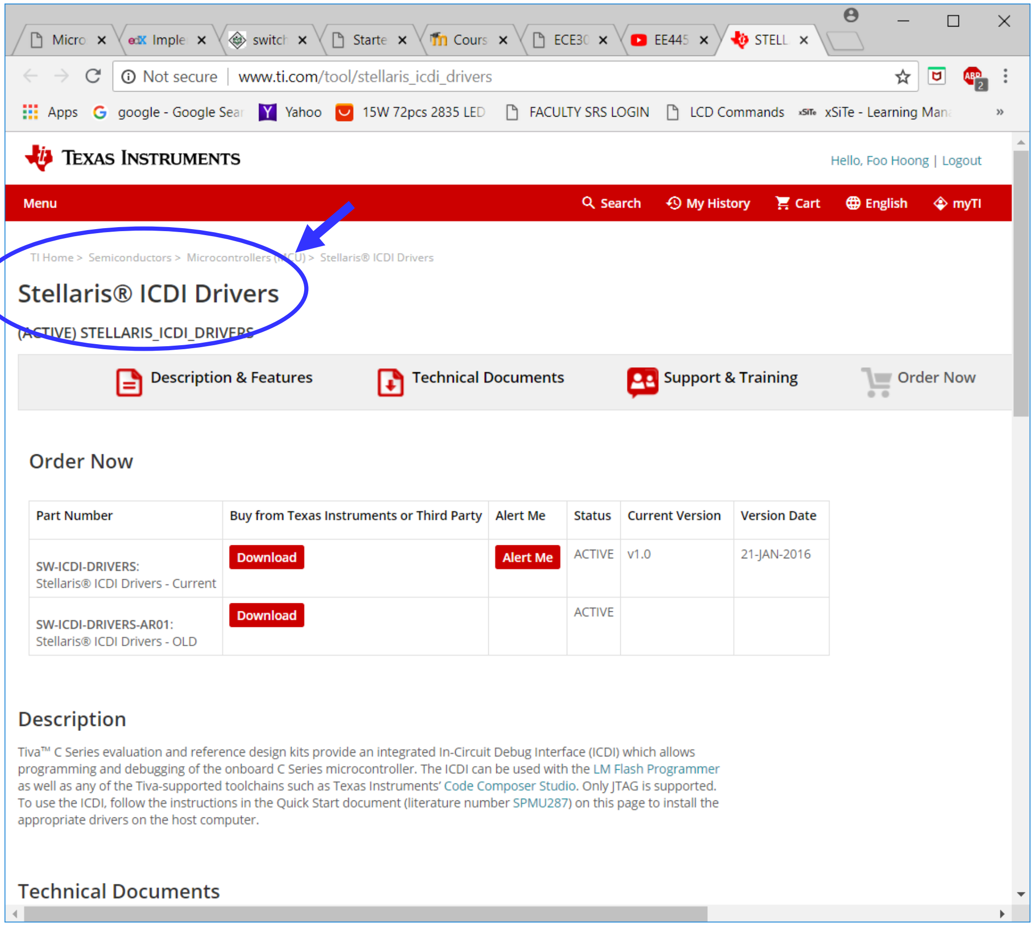
Use an add-on installer that brings back the support for Stellaris ICDI to MDK v5.29 and above. Just download and install MDK_Stellaris_ICDI_AddOn.exe.

Stellaris Driver

CS397

Stellaris Driver Installation

- Driver at can be downloaded from http://www.ti.com/tool/stellaris_icdi_drivers.
  - A version is also available at Moodle: ‘spmc016a.zip’
- Reference document: *Stellaris Driver Installation (spmu287b.pdf).*
  - A version of the document is available at Moodle.



Stellaris® ICDI Drivers

(ACTIVE) STELLARIS_ICDI_DRIVERS

Order Now

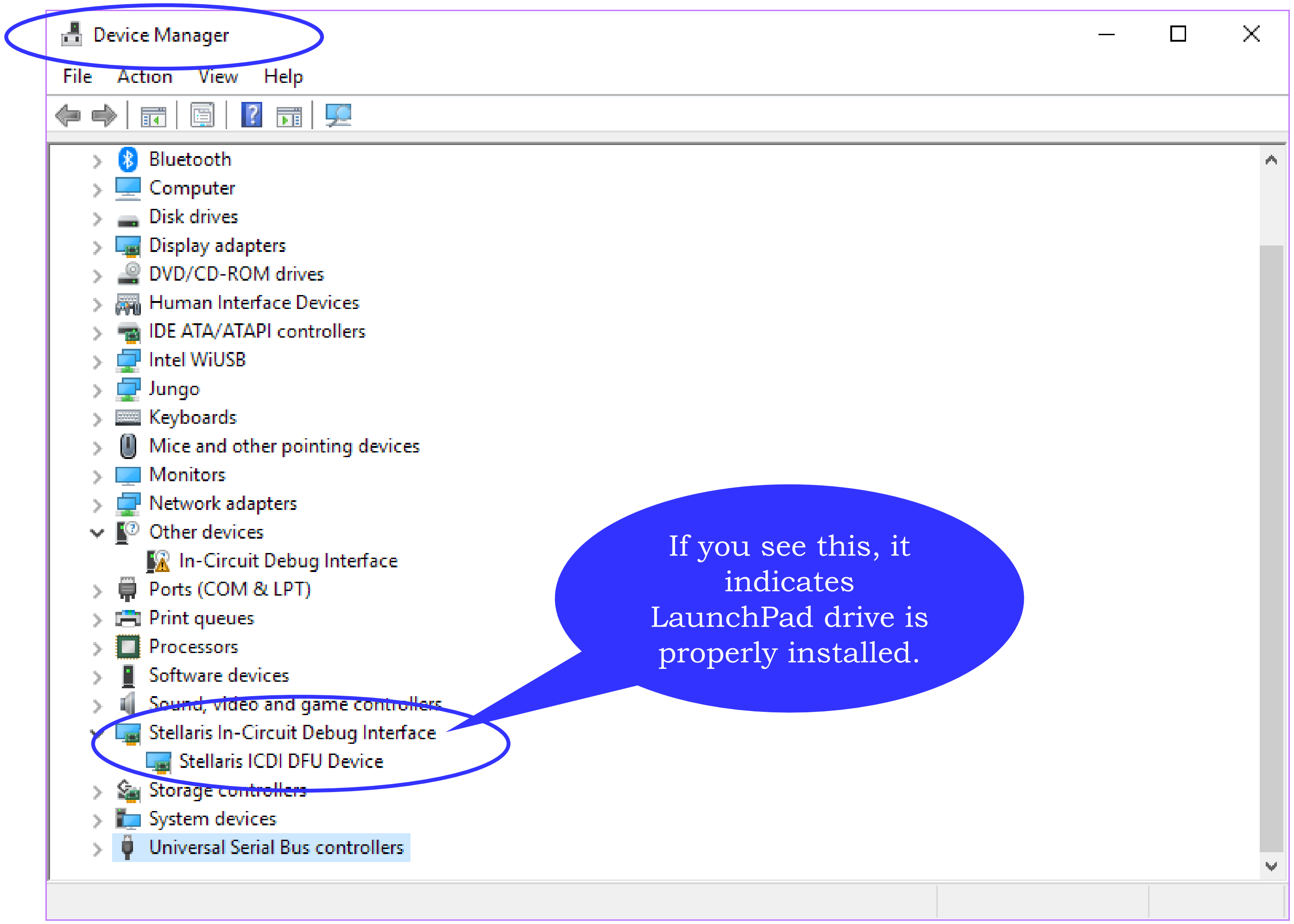
Part Number	Buy from Texas Instruments or Third Party	Alert Me	Status	Current Version	Version Date
SW-ICDI-DRIVERS: Stellaris® ICDI Drivers - Current	Download	Alert Me	ACTIVE	v1.0	21-JAN-2016
SW-ICDI-DRIVERS-AR01: Stellaris® ICDI Drivers - OLD	Download		ACTIVE		

Description

Tiva™ C Series evaluation and reference design kits provide an integrated In-Circuit Debug Interface (ICDI) which allows programming and debugging of the onboard C Series microcontroller. The ICDI can be used with the LM Flash Programmer as well as any of the Tiva-supported toolchains such as Texas Instruments' Code Composer Studio. Only JTAG is supported. To use the ICDI, follow the instructions in the Quick Start document (literature number SPMU287) on this page to install the appropriate drivers on the host computer.

Technical Documents

Stellaris Driver Installation



Bluetooth
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
IDE ATA/ATAPI controllers
Intel WiUSB
Jungo
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
In-Circuit Debug Interface
Ports (COM & LPT)
Print queues
Processors
Software devices
Sound, video and game controllers
Stellaris In-Circuit Debug Interface
Stellaris ICDI DFU Device
Storage controllers
System devices
Universal Serial Bus controllers

If you see this, it indicates LaunchPad drive is properly installed.

Test SW & LaunchPad Setup

- Download Lab 1 Program Template from Moodle.
- Re-build the program.
- Download the program to the Tiva LaunchPad.
- If your setup is correct, you should see the LED blinking **BLUE** on the Tiva LaunchPad.

Understand Program Template

- Program Walk-through:
    - main.c
    - Startup_TM4C123.c:
        - Vector table
    - Startup_TM4C123.s:
        - SystemCoreClockUpdate()
        - SystemInit()
    - BSP.c
    - Hal.c
        - Port_Init()

Programming the SysTick Timer

- Remove the function call `SysTick_Config()` in main.c.
- Write your own function **`Configure_SysTick(ticks)`**.
  - The function should initialise the SysTick timer using the SysTick Timer registers.
  - Parameter '**`ticks`**' denote the number of timer ticks between SysTick interrupts.
  - The SysTick timer registers are defined in header file 'NVIC.h'. Add this file to your program structure. You can place this file in the BSP directory.
  - Use the example from lecture notes as a guide.
- Verify that the processor is running at 80MHz, modify Lab 1 template program to perform the following:
  - SysTick Timer interrupt occurs at every **10ms**. Modify the SysTick interrupt handler as needed.
  - Program the LED to blink at **5 Hz**. The LED can blink in any colour (*your choice*).
  - **[SUBMIT 1]** Use the Digilent Analog Discovery to capture the output waveform at the GPIO pin for the LED. Verify that the frequency is 5Hz. Note that which GPIO pin to capture would depend on the colour.

SysTick Programming *(main.h)*

```
int main()
{
   BSPInit(); /* in BSP.c   */
   BOOL bToggle = TRUE;
   SystemCoreClockUpdate();

   /** NOTE for Lab 1:                                                 **/
   /** This program currently uses the CMSIS function to configure     **/
   /**   It is for testing your setup.                                  **/
   /** For LAB1, COMMENT OUT THIS LINE OF CODE &                        **/
   /**   replace it with your own function: Configure_SysTick(ticks)   **/
   /**   to provide system ticks for every 10 ms.                       **/
   /**   Use the lecture example as a guide.                            **/
   SysTick_Config( SystemCoreClock/1000 );
   for(;;)
   {
     if( FALSE != g_bSystemTick )  /* Check if flag is set by the SysTick Handler  */
     {
        /* Clear SysTick flag so we only processes it once       */
        g_bSystemTick = FALSE;
        /* Set LED to RED if toggle is TRUE(=1),                 */
        /* otherwise if toggle is FALSE(=0), the LED will be off */
        LED_RGB_SET( RGB_RED * bToggle );
        bToggle = !bToggle; /* Inverse toggle, so if 0 it becomes 1, 1 becomes 0 */
     }
   }
}
```

Remove function **`SysTick_Config()`**

SysTick Programming *(main.h)*

```
int main()
{
    BSPInit(); /* in BSP.c   */

    BOOL bToggle = TRUE;
    SystemCoreClockUpdate();

    /** TO DO FOR LAB 1: Initialize SysTick Timer                              **/
    /** Initialize SysTick Timer to trigger every 10 ms                        **/
    Configure_SysTick(ticks); /* your own function prototype, definitions    **/

    for(;;)
    {
      if( FALSE != g_bSystemTick ) /* Check if flag is set by SysTick Handler */
      {
         g_bSystemTick = FALSE;            /* Clear SysTick flag              */
         LED_RGB_SET( RGB_RED * bToggle );
         bToggle = !bToggle;       /* Inverse toggle                    */
      }
    }
}
```

Create your own function:
`Configure_SysTick (ticks)`

SysTick Timer Registers *(NVIC.h)*

- The NVIC register definitions are in a file ‘**NVIC.h**’. Use these definition for your programs to access the Systick Timer registers.

```
/******** NVIC registers (NVIC) *********************/
#define NVIC_ACTLR_R              (*((volatile uint32_t *)0xE000E008))
#define NVIC_ST_CTRL_R            (*((volatile uint32_t *)0xE000E010))
#define NVIC_ST_RELOAD_R          (*((volatile uint32_t *)0xE000E014))
#define NVIC_ST_CURRENT_R         (*((volatile uint32_t *)0xE000E018))
#define NVIC_EN0_R                (*((volatile uint32_t *)0xE000E100))
#define NVIC_EN1_R                (*((volatile uint32_t *)0xE000E104))
#define NVIC_EN2_R                (*((volatile uint32_t *)0xE000E108))
#define NVIC_EN3_R                (*((volatile uint32_t *)0xE000E10C))
#define NVIC_EN4_R                (*((volatile uint32_t *)0xE000E110))
#define NVIC_DIS0_R               (*((volatile uint32_t *)0xE000E180))


………
….
```

SysTick Timer Registers

More Practice on SysTick Timer:

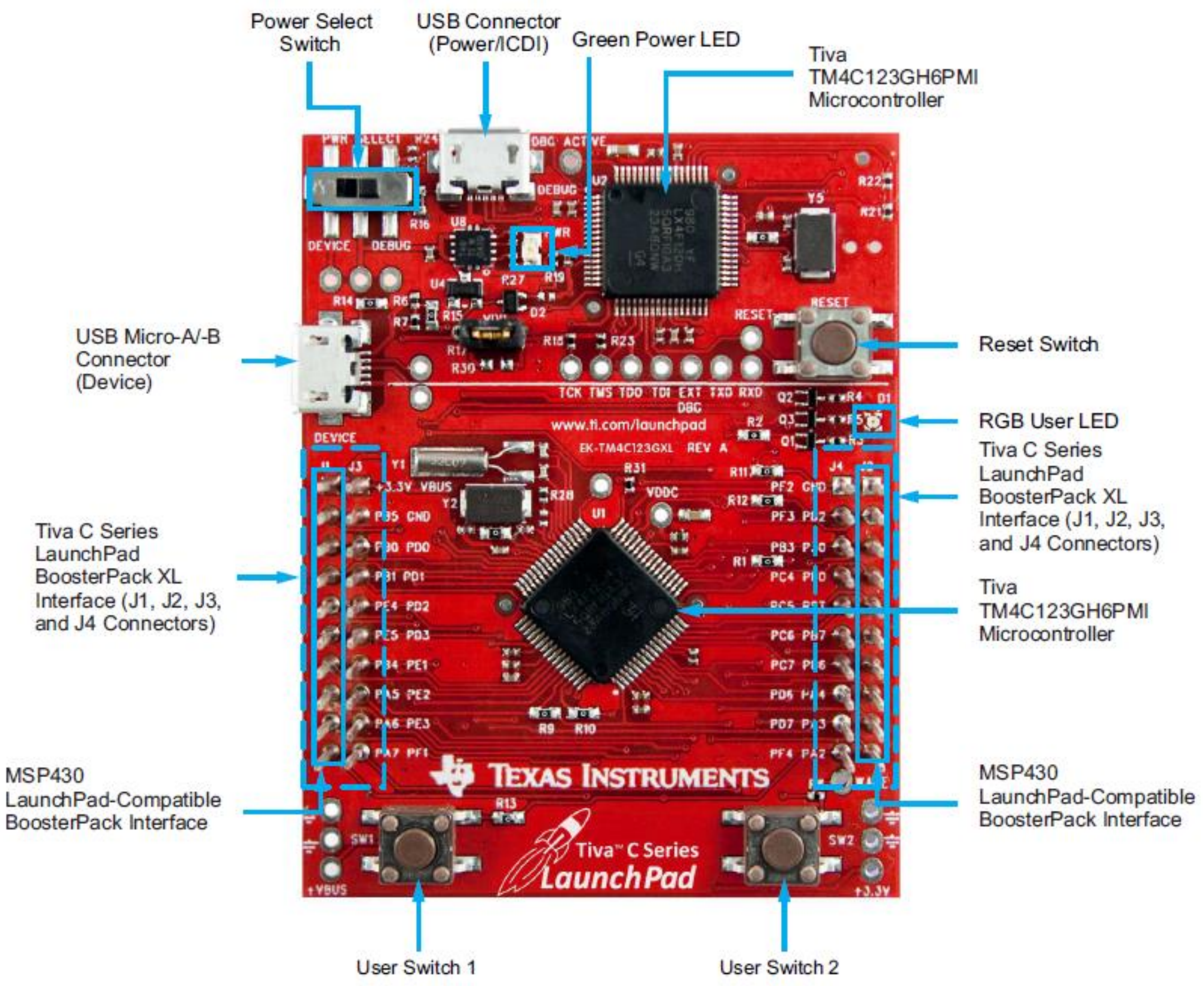
- Modify the program template to be able to perform the following:
  - Light up the colour-LED according to the timings in the table.
  - Let the LED colours run in sequence and indefinitely.
  - You may introduce **flag(s)** in the SysTick Timer Handler to help you to implement the timings for the LED colours.
- **[SUBMIT 2]** Submit your completed program as a zipped folder. Upload to Moodle.

Sequence	**Duration**	**LED Colour**
1	1.7s	**RED**
2	1.0s	**GREEN**
3	1.3s	**BLUE**
4	2.5s	**WHITE**

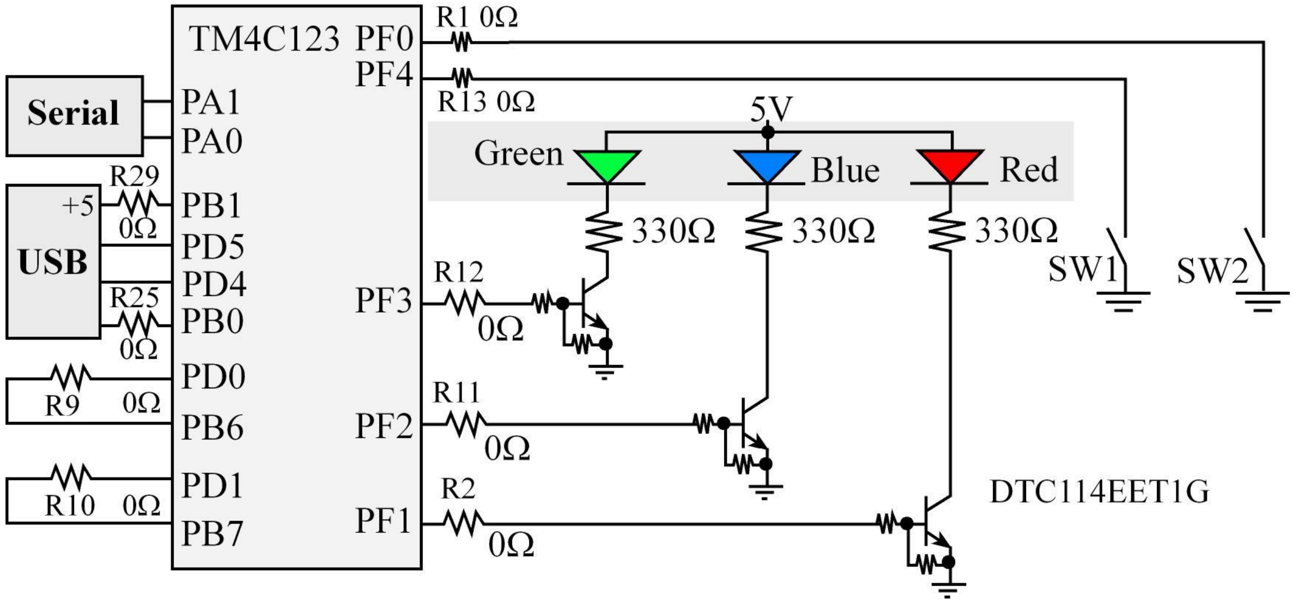
Tiva LaunchPad

For Reference

Tiva C Series TMC123G (LaunchPad)

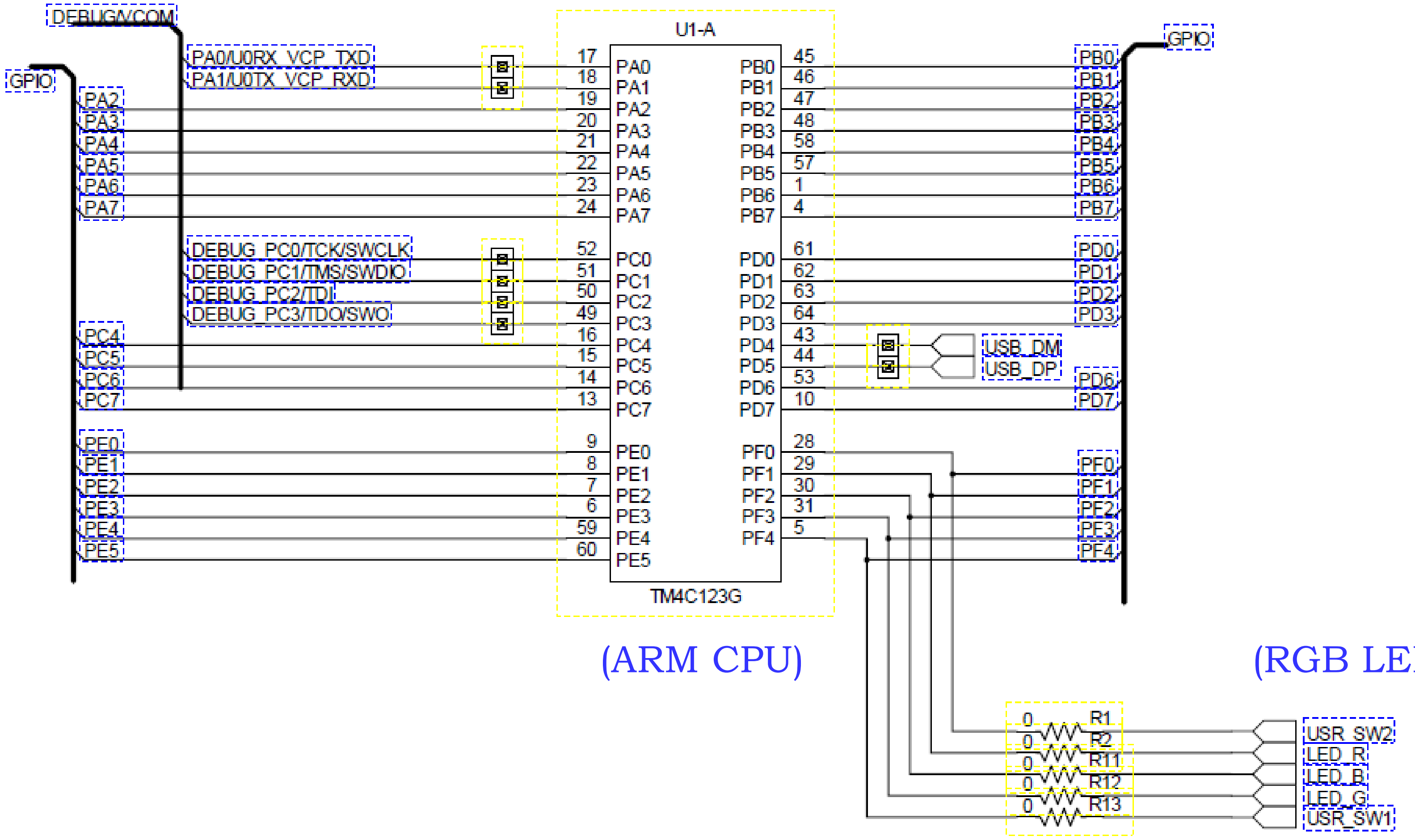

Power Select Switch
USB Connector (Power/ICDI)
Green Power LED
Tiva TM4C123GH6PMI Microcontroller
USB Micro-A/-B Connector (Device)
Reset Switch
RGB User LED
Tiva C Series LaunchPad BoosterPack XL Interface (J1, J2, J3, and J4 Connectors)
Tiva C Series LaunchPad BoosterPack XL Interface (J1, J2, J3, and J4 Connectors)
Tiva TM4C123GH6PMI Microcontroller
MSP430 LaunchPad-Compatible BoosterPack Interface
MSP430 LaunchPad-Compatible BoosterPack Interface
User Switch 1
User Switch 2

Tiva LaunchPad



GPIO Pin	Pin Function	USB Device
PF4	GPIO	SW1
PF0	GPIO	SW2
PF1	GPIO	RGB LED (Red)
PF2	GPIO	RGB LED (Blue)
PF3	GPIO	RGD LED (Green)

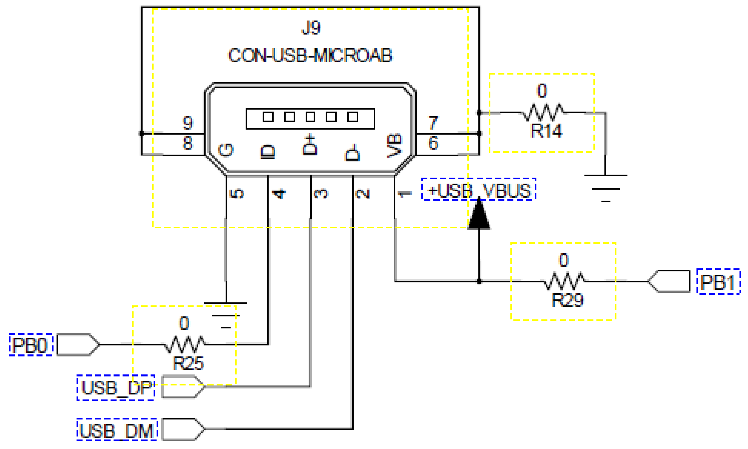
LaunchPad Schematics (partial)

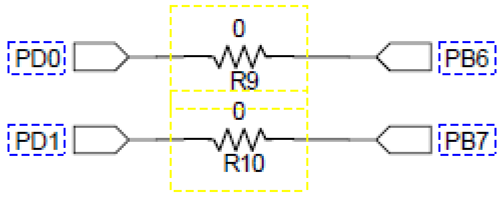


(ARM CPU)

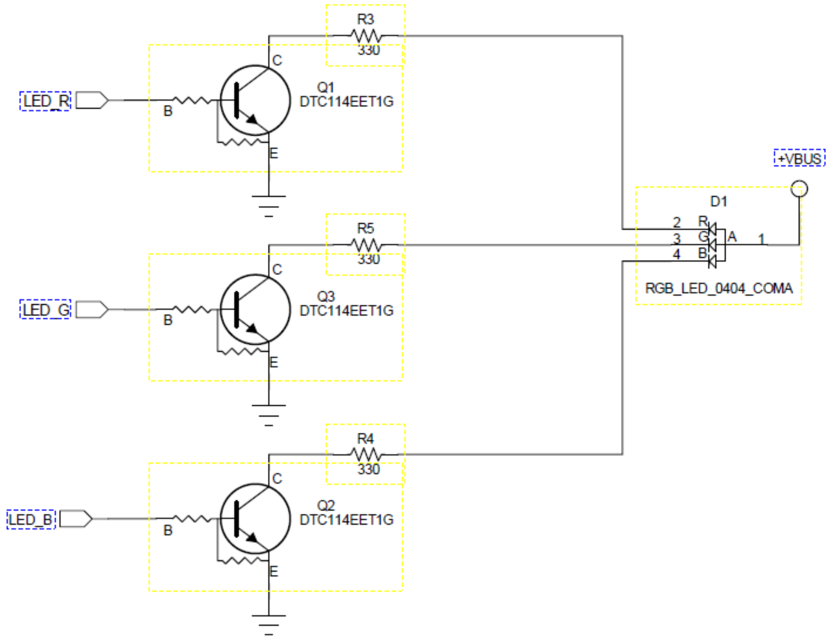
(RGB LED)

LaunchPad Schematics (partial)



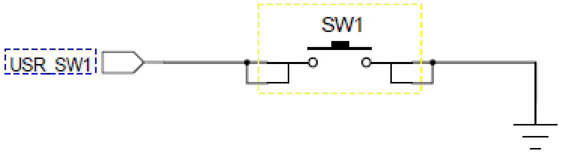


Tiva LaunchPad – SW1, SW2, LED

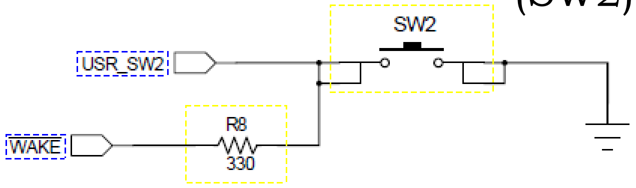


(RGB LED)

(SW1)



(SW2)



Source: Tiva C Series TM4C123G LaunchPad Evaluation Board – User Guide (spmu296.pdf)

Reference Documents

- Tiva C Series TM4C123G LaunchPad Evaluation Board – User Guide (‘*spmu296.pdf*’)
- Tiva TM4C123GH6PM Microcontroller – Data Sheet (‘*spms376e.pdf*’)

Both files can be downloaded from Moodle.

KEIL Debugger - Potential Issues

- Should you face issue where KEIL debugger exits almost immediately when DEBUG mode was entered, the following link can help: http://users.ece.utexas.edu/~valvano/Volume1/Window8KeilDebuggerFix.htm.