

Module 2: RT-Thread Device Driver – LED Control

Learning Objectives

- Understand PSOC-E84 boot sequence
- Create and compile LED blinking project
- Understand GPIO driver framework
- Control LEDs via device driver APIs

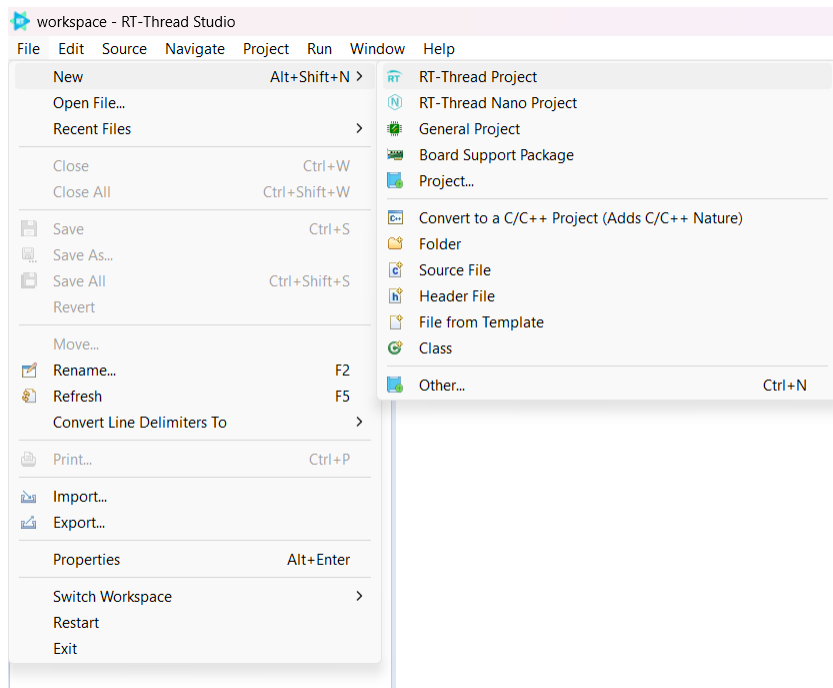
2.1 Boot Sequence

Edge E84 boot order:

M33_S → M33_NS → M55_NS

2.2 Creating the LED Project

1. Open RT-Thread Studio → File → New → RT-Thread Project



2. Select **Based on Board** Select **Based on Development Board**
3. Board: **PSOC_E84-EDGI-TALK** | BSP: **1.2.1**
4. Type: **Example Project** → **Edgi_Talk_M33_Blink_LED**

New Project

Create RT-Thread Project



Input project name, choose RT-Thread version and select one Board and choose template or example.

Project name: Edgi_Talk_M33_Blink_LED

☒ Use default location

Location: D:\RT-ThreadStudio\workspace\Edgi_Talk_M33_Blink_LED

Browse...

☐ Base On MCU ☒ Base On Board

Board : PSOC_E84-EDGI-TALK

BSP : 1.2.1

Type : Example

Example : Edgi_Talk_M33_Blink_LED

RT-Thread : 5.0.2

Adapter : KitProg3

Port : SWD

?

< Back

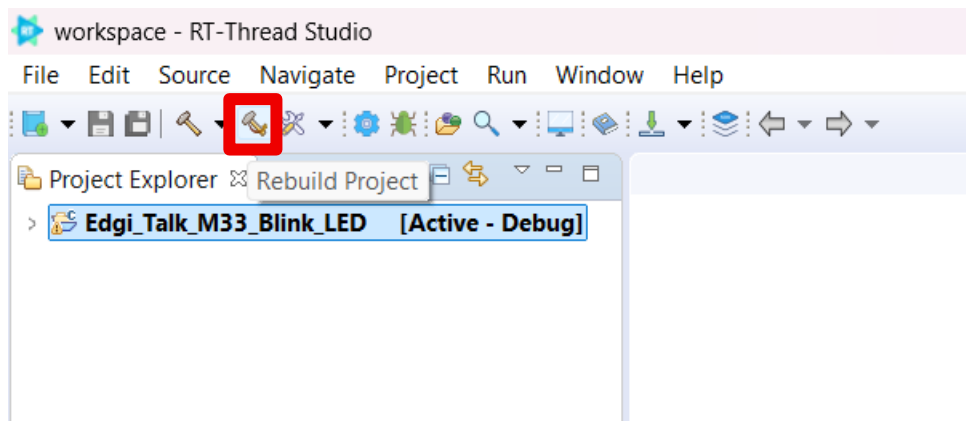
Next >

Finish

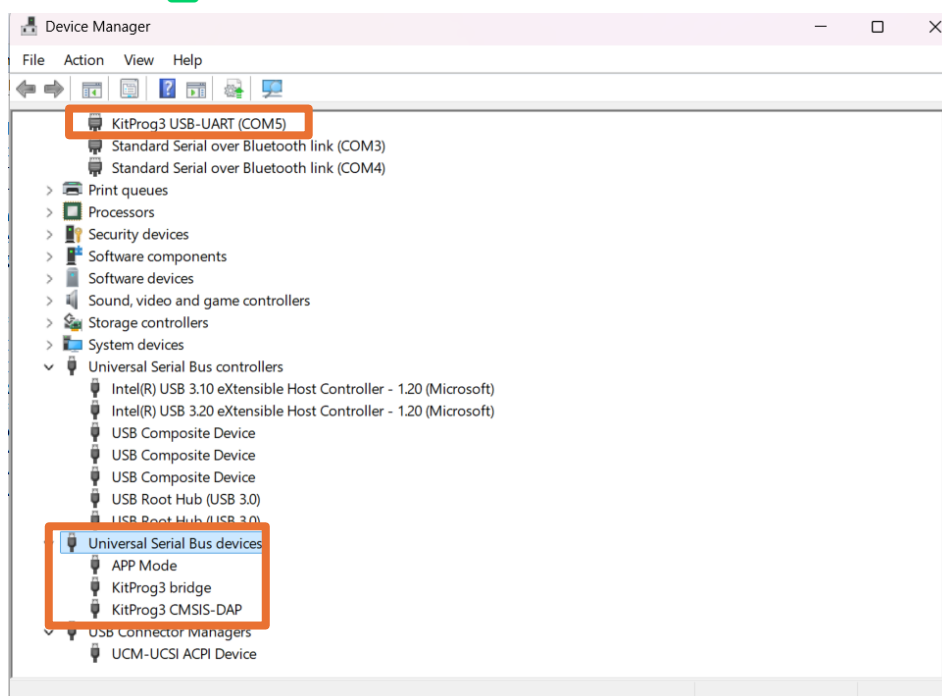
Cancel

2.3 Building and Flashing

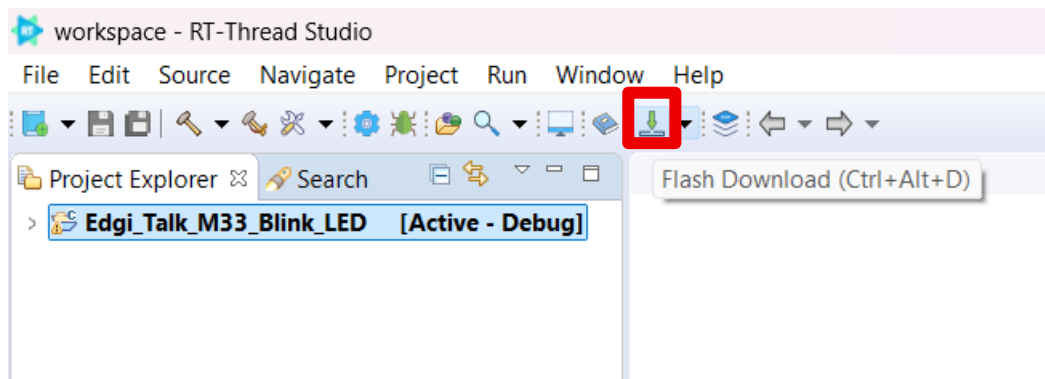
- **Compile:** Click Hammer icon



- Use USB Cable plug in to DAP port and connect to laptop/pc
 - Open **Device Manager** and make sure they are all connected:
1. **KitProg3 USB-UART (COM5)** ✓
 2. **KitProg3 bridge** ✓
 3. **KitProg3 CMSIS-DAP** ✓
 4. **APP Mode** ✓



- **Flash:** Click Download icon



- **Serial Terminal LOG Output**

```

Problems Tasks Console Properties Terminal
Edgi_Talk_M33_Blink_LED (Jolin)
Microsoft Windows [Version 10.0.26200.8655]
(c) Microsoft Corporation. All rights reserved.

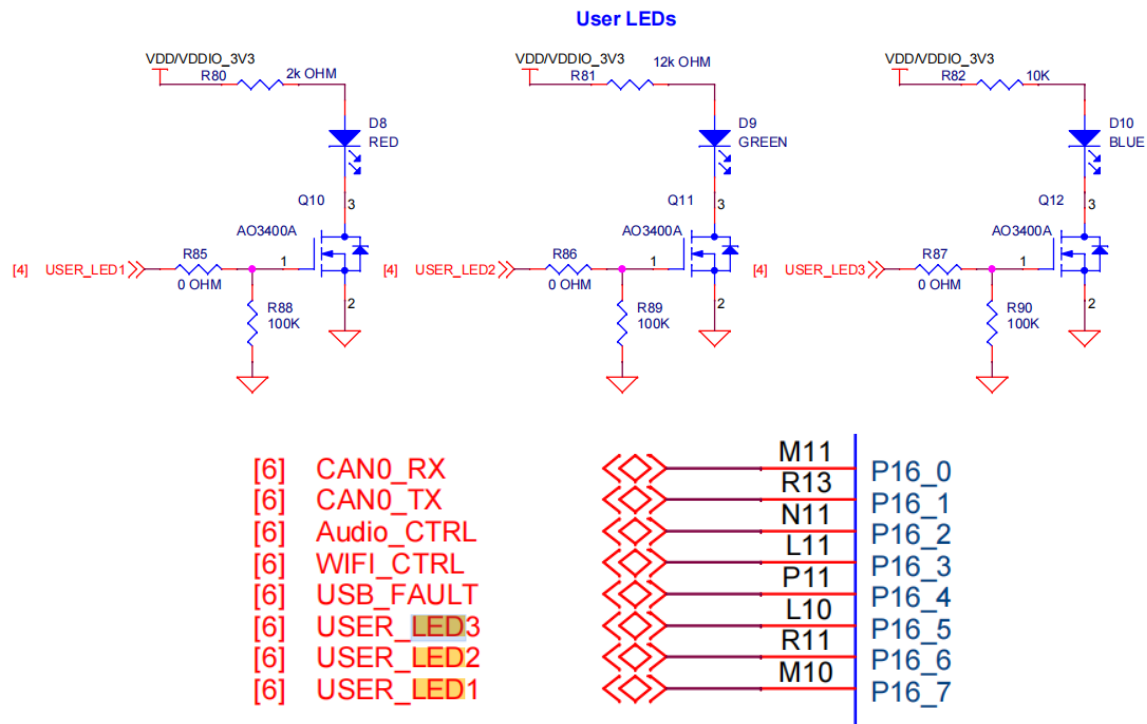
D:\RT-ThreadStudio\workspace\Edgi_Talk_M33_Blink_LED>D:/RT-ThreadStudio/plugins/org.rt-thread.studio.serial.putty_1.0.5/putty/plink.exe -serial COM5 -sercfg 115200,8,n,1,N
O C@+[2J+[]H***** PSOC Edge MCU: CM33 Secure Mode*****
PSRAM Cache is Enabled
PSRAM init successful
***** PSOC Edge MCU: CM33 Secure Mode Exit*****

\ | /
- RT - Thread Operating System
/ | \ 5.0.2 build Jul 6 2026 11:23:42
2006 - 2022 Copyright by RT-Thread team
Hello RT-Thread
This core is cortex-m33
msh >help
RT-Thread shell commands:
list          - list objects
version       - show RT - Thread version information
clear         - clear the terminal screen
free          - Show the memory usage in the system.
ps            - List threads in the system.
help          - RT - Thread shell help.
pin           - pin [option]
reboot        - Reboot System
poweroff      - The software enables the system to shut down. Simply press the button to restart it.
msh >

```

2.4 LED Circuit Principle

LED	Component	Color	Control
USER_LED1	D8	Red	GPIO HIGH = ON
USER_LED2	D9	Green	GPIO LOW = OFF
USER_LED3	D10	Blue	MOSFET-driven



2.5 LED Code Example

```
#include <rtthread.h>
#include <rtdevice.h>
#include <board.h>

#define LED_PIN GET_PIN(16, 6)

#define LED_PIN_B          GET_PIN(16, 5)

int main(void)
{
    rt_pin_mode(LED_PIN, PIN_MODE_OUTPUT);

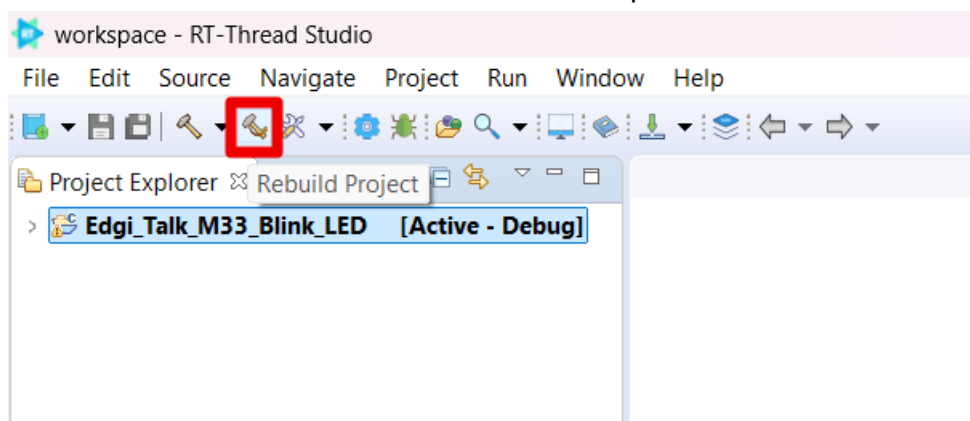
    rt_kprintf("Hello RT-Thread\r\n");
    rt_kprintf("This core is cortex-m33\r\n");
    rt_pin_mode(LED_PIN_B, PIN_MODE_OUTPUT);
    while (1)
    {
        rt_pin_write(LED_PIN_B, PIN_HIGH);
        rt_thread_mdelay(500);
        rt_pin_write(LED_PIN_B, PIN_LOW);
        rt_thread_mdelay(500);
    }

    Return 0;
}

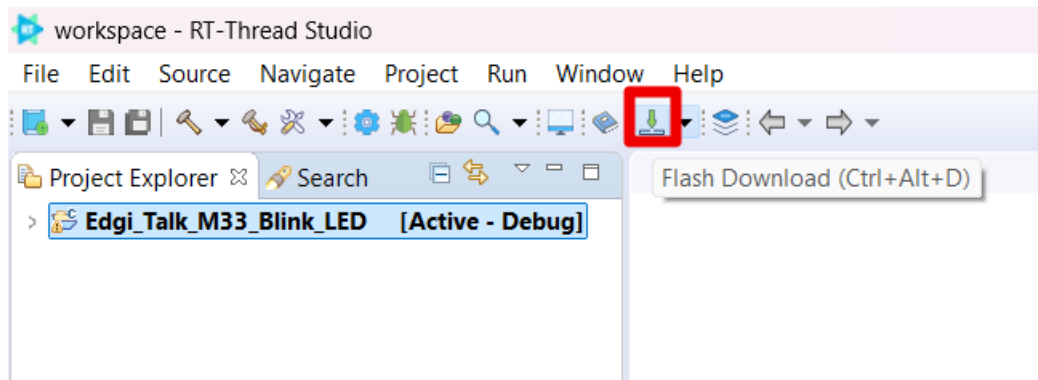
}

}
return 0;
}
```

- Click the hammer button in the Studio to compile



- For firmware flashing: click the download button in the Studio to flash the firmware.



2.6 Expected Result

Blue LED blinks at **500ms intervals**.

