Micro-Controller Experiment

Week2

Teacher: 廖裕評 Yu-Ping Liao

TA: 陳大荃 Da-chuan Chen, 陳恩妮 En-ni Chen

Class Rules

1. No drink besides water.
2. Bring a laptop and breadboard if needed.
3. Ask us TAs to sign and borrow development boards. Do not sign or ask others to sign for you without TAs’ permission.
4. Arriving 10 minutes after the bell rings will be regarded as absent.
5. If you damage any borrowed equipment, you have to pay for it.

Homework Rules

1. Includes: A. Class content, B. Class exercise, C. Homework (screenshot or video)
2. Editing software: MS PowerPoint
3. File format: PDF
4. Filename: "date_group_studentID_name.pdf", like "0916_第1組_11028XXX_陳OO.pdf"
5. The homework deadline is 23:59 of the day before the next class. If you are late, then your grade will be deducted.

Contact

If you encounter any problems with this class, please get in touch with us with the following

E-mails:

1. Teacher, Prof. Yu-Ping Liao 廖裕評： lyp@cycu.org.tw
2. TA, Da-chuan Chen 陳大荃： dachuan516@gmail.com
3. TA, En-ni Chen 陳恩妮： anna7125867@gmail.com

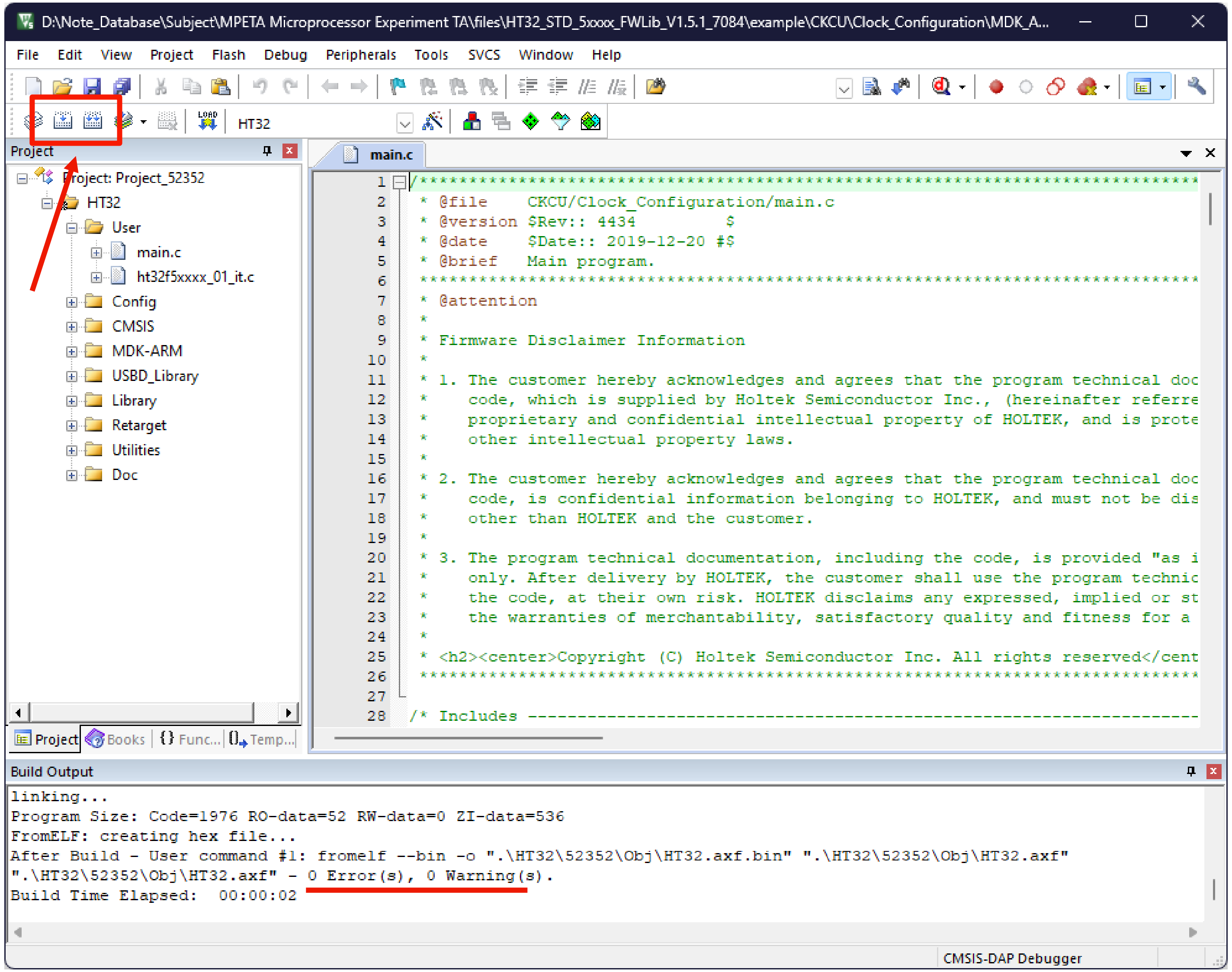
Or visit 篤信 Lab353 for further questions.

Outline of the Week

1. Execute the CKCU project.
2. Homework W2-1.
3. GPIO Introduction.
4. GPIO Project.
5. Homework W2-2.
6. Homework W2-4 Bonus Question.
7. Homework W2-5 Bonus Question.

Execute the CKCU Project

1. Compile



D:\Note_Database\Subject\MPETA Microprocessor Experiment TA\files\HT32_STD_5xxxx_FWLib_V1.5.1_7084\example\CKCU\Clock_Configuration\MDK_A...
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
HT32
Project
Project: Project_52352
HT32
User
main.c
ht32f5xxxx_01_it.c
Config
CMSIS
MDK-ARM
USBD_Library
Library
Retarget
Utilities
Doc
Project
Books
Func...
Temp...
main.c
Build Output
CMSIS-DAP Debugger

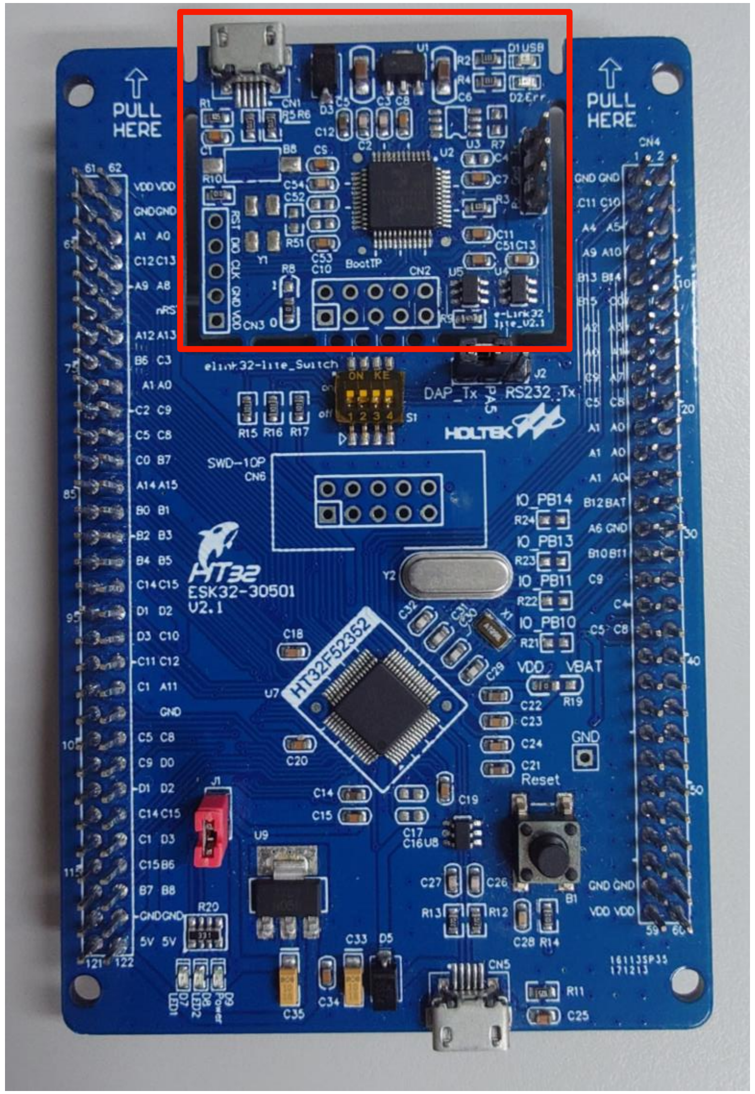
```
/*****************************************************************************
 * @file    CKCU/Clock_Configuration/main.c
 * @version $Rev:: 4434         $
 * @date    $Date:: 2019-12-20 #$
 * @brief   Main program.
 *****************************************************************************
 * @attention
 *
 * Firmware Disclaimer Information
 *
 * 1. The customer hereby acknowledges and agrees that the program technical doc
 *    code, which is supplied by Holtek Semiconductor Inc., (hereinafter referre
 *    proprietary and confidential intellectual property of HOLTEK, and is prote
 *    other intellectual property laws.
 *
 * 2. The customer hereby acknowledges and agrees that the program technical doc
 *    code, is confidential information belonging to HOLTEK, and must not be dis
 *    other than HOLTEK and the customer.
 *
 * 3. The program technical documentation, including the code, is provided "as i
 *    only. After delivery by HOLTEK, the customer shall use the program technic
 *    the code, at their own risk. HOLTEK disclaims any expressed, implied or st
 *    the warranties of merchantability, satisfactory quality and fitness for a
 *
 * <h2><center>Copyright (C) Holtek Semiconductor Inc. All rights reserved</cent
 *****************************************************************************

/* Includes ------------------------------------------------------------------
```

```
linking...
Program Size: Code=1976 RO-data=52 RW-data=0 ZI-data=536
FromELF: creating hex file...
After Build - User command #1: fromelf --bin -o ".\HT32\52352\Obj\HT32.axf.bin" ".\HT32\52352\Obj\HT32.axf"
".\HT32\52352\Obj\HT32.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:02
```

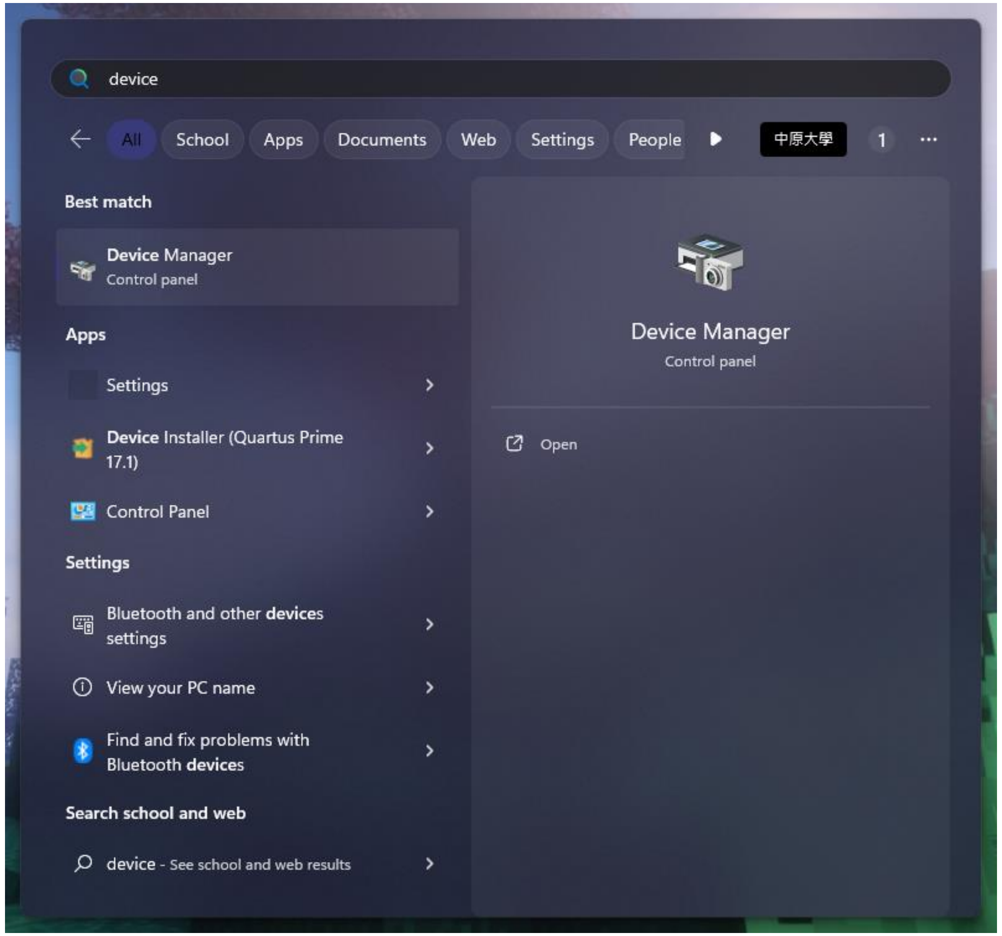
2. Connect development board

- Boxed area is the Debugger/Programmer of this board. Responsible for communication between HT32F52352 and your laptop.
- Rest area are the MCU HT32F52352 and its support circuitry.
- Connect USB Micro-B side of the transmission wire to the boxed area, USB Type-A to your laptop.

Micro-B



3. Check connection



Best match

Device Manager
Control panel

Apps

Settings

Device Installer (Quartus Prime 17.1)

Control Panel

Settings

Bluetooth and other devices settings

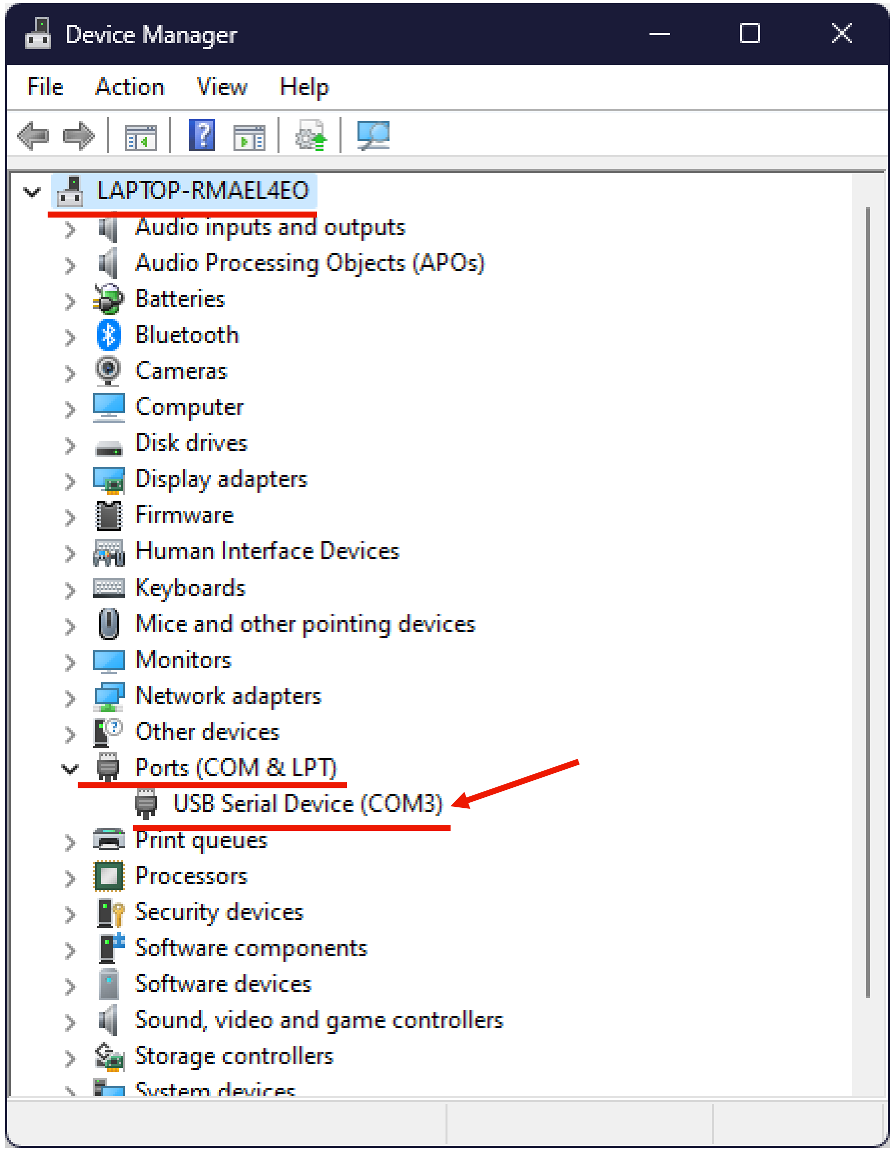
View your PC name

Find and fix problems with Bluetooth devices

Search school and web

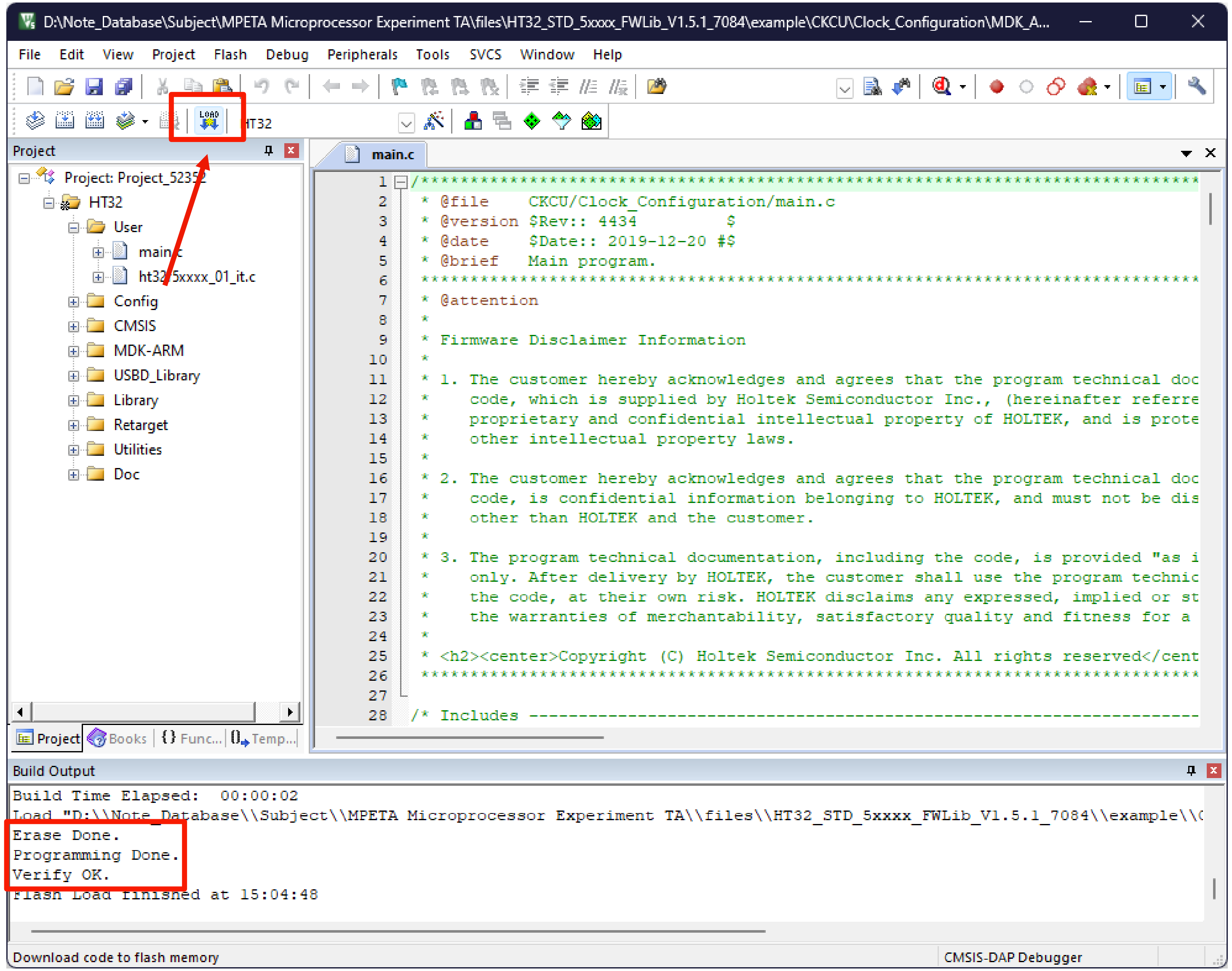
device - See school and web results

Device Manager
Control panel



LAPTOP-RMAEL4EO
- Audio inputs and outputs
- Audio Processing Objects (APOs)
- Batteries
- Bluetooth
- Cameras
- Computer
- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Ports (COM & LPT)
  - USB Serial Device (COM3)
- Print queues
- Processors
- Security devices
- Software components
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices

4. Flash compiled code to board

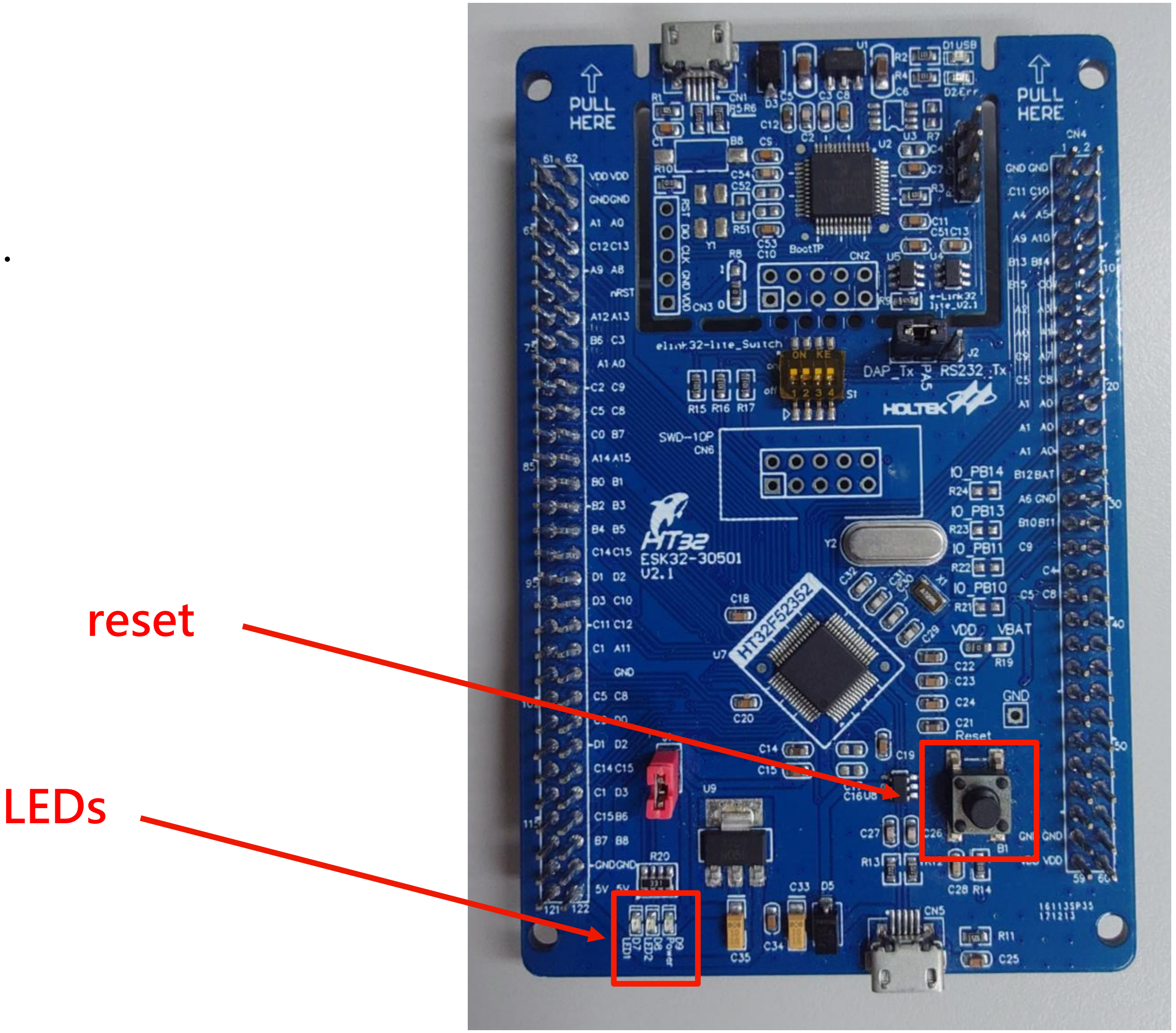


```
/*********************************************************************************************
 * @file    CKCU/Clock_Configuration/main.c
 * @version $Rev:: 4434         $
 * @date    $Date:: 2019-12-20 #$
 * @brief   Main program.
 *********************************************************************************************
 * @attention
 *
 * Firmware Disclaimer Information
 *
 * 1. The customer hereby acknowledges and agrees that the program technical doc
 *    code, which is supplied by Holtek Semiconductor Inc., (hereinafter referre
 *    proprietary and confidential intellectual property of HOLTEK, and is prote
 *    other intellectual property laws.
 *
 * 2. The customer hereby acknowledges and agrees that the program technical doc
 *    code, is confidential information belonging to HOLTEK, and must not be dis
 *    other than HOLTEK and the customer.
 *
 * 3. The program technical documentation, including the code, is provided "as i
 *    only. After delivery by HOLTEK, the customer shall use the program technic
 *    the code, at their own risk. HOLTEK disclaims any expressed, implied or st
 *    the warranties of merchantability, satisfactory quality and fitness for a
 *
 * <h2><center>Copyright (C) Holtek Semiconductor Inc. All rights reserved</cent
 *********************************************************************************************

/* Includes ------------------------------------------------------------------------------
```

```
Build Time Elapsed:  00:00:02
Load "D:\\Note_Database\\Subject\\MPETA Microprocessor Experiment TA\\files\\HT32_STD_5xxxx_FWLib_V1.5.1_7084\\example\\(
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 15:04:48
```

5. Observe result

1. Press the “reset” button.
2. Make sure “LED1” and “LED2” are flickering.



Exercise W2-1

https://github.com/CYCU-AIoT-System-Lab/Microcontroller-Experiment/blob/main/w2/CKCU-Clock_Configuration-Experiment_Steps.md

Modify clock speed of CKCU example

- Objective 1: Change the LED flickering frequency and compare the before and after differences.
- Objective 2: Find the corresponding pins connected with LED1~3.
- Hint: Use “F12” to look for the main function's parameter controlling the clock speed.

GPIO Introduction

What is GPIO?

A general-purpose input/output (GPIO) is an uncommitted digital signal pin on an integrated circuit or electronic circuit (e.g. MCUs/MPUs) board which may be used as an input or output, or both, and is controllable by software.

GPIOs have no predefined purpose and are unused by default. If used, the purpose and behavior of a GPIO is defined and implemented by the designer of higher assembly-level circuitry: the circuit board designer in the case of integrated circuit GPIOs, or system integrator in the case of board-level GPIOs.

Datasheet P27

GPIO available pins

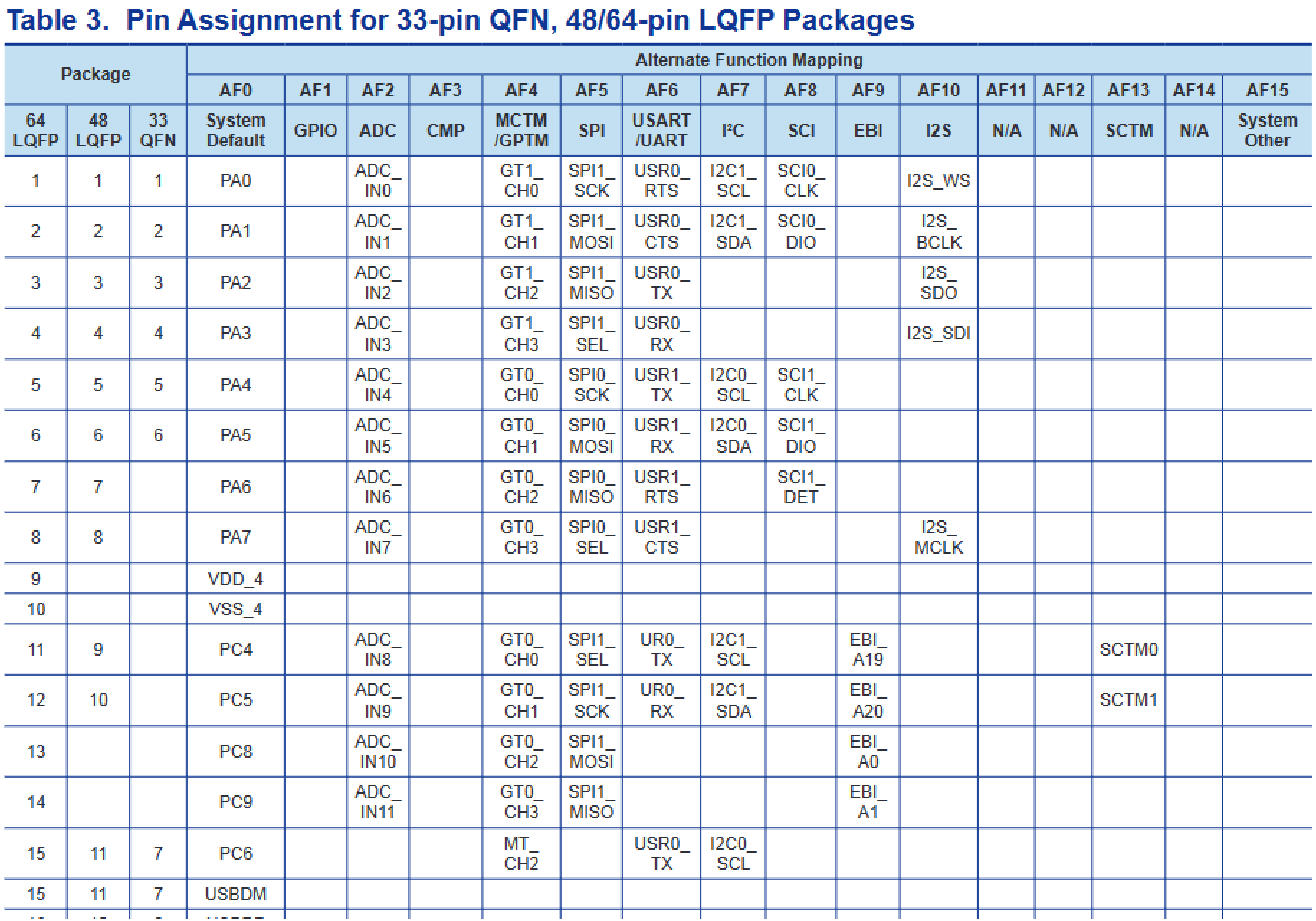


Table 3. Pin Assignment for 33-pin QFN, 48/64-pin LQFP Packages

Package			Alternate Function Mapping															
			AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
64 LQFP	48 LQFP	33 QFN	System Default	GPIO	ADC	CMP	MCTM /GPTM	SPI	USART /UART	I²C	SCI	EBI	I2S	N/A	N/A	SCTM	N/A	System Other
1	1	1	PA0		ADC_IN0		GT1_CH0	SPI1_SCK	USR0_RTS	I2C1_SCL	SCI0_CLK		I2S_WS					
2	2	2	PA1		ADC_IN1		GT1_CH1	SPI1_MOSI	USR0_CTS	I2C1_SDA	SCI0_DIO		I2S_BCLK					
3	3	3	PA2		ADC_IN2		GT1_CH2	SPI1_MISO	USR0_TX				I2S_SDO					
4	4	4	PA3		ADC_IN3		GT1_CH3	SPI1_SEL	USR0_RX				I2S_SDI					
5	5	5	PA4		ADC_IN4		GT0_CH0	SPI0_SCK	USR1_TX	I2C0_SCL	SCI1_CLK							
6	6	6	PA5		ADC_IN5		GT0_CH1	SPI0_MOSI	USR1_RX	I2C0_SDA	SCI1_DIO							
7	7		PA6		ADC_IN6		GT0_CH2	SPI0_MISO	USR1_RTS		SCI1_DET							
8	8		PA7		ADC_IN7		GT0_CH3	SPI0_SEL	USR1_CTS				I2S_MCLK					
9			VDD_4															
10			VSS_4															
11	9		PC4		ADC_IN8		GT0_CH0	SPI1_SEL	UR0_TX	I2C1_SCL		EBI_A19				SCTM0		
12	10		PC5		ADC_IN9		GT0_CH1	SPI1_SCK	UR0_RX	I2C1_SDA		EBI_A20				SCTM1		
13			PC8		ADC_IN10		GT0_CH2	SPI1_MOSI				EBI_A0						
14			PC9		ADC_IN11		GT0_CH3	SPI1_MISO				EBI_A1						
15	11	7	PC6				MT_CH2		USR0_TX	I2C0_SCL								
15	11	7	USBDM															

*The GPIO pins that are not labeled on GPIO column can also be used for GPIO functions.

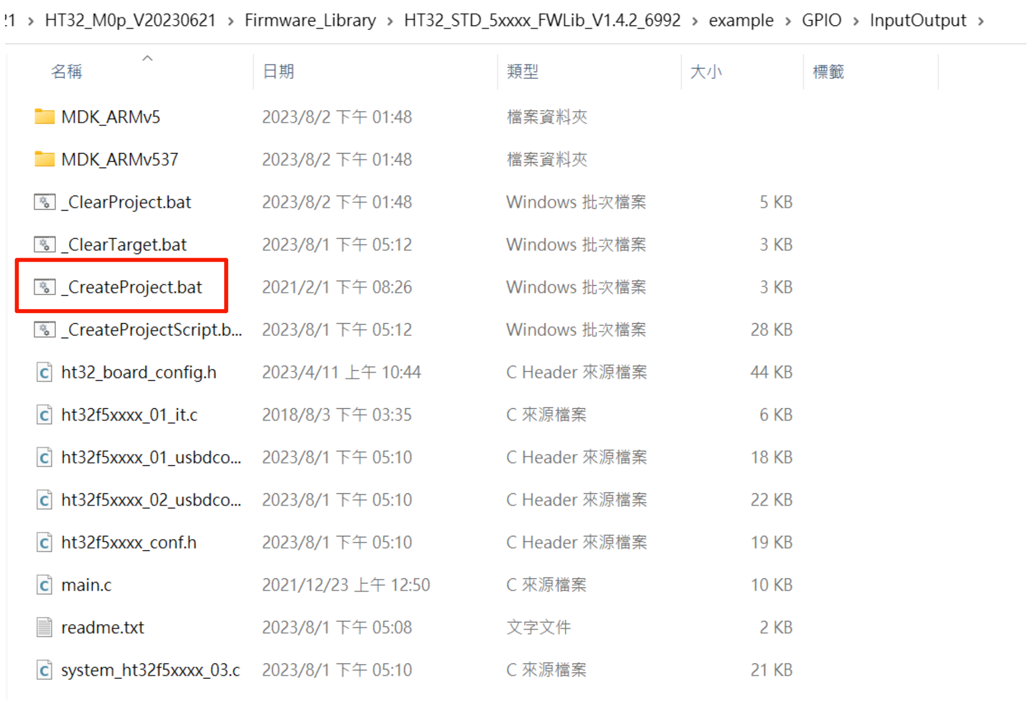
GPIO Project

1. Execute "_CreatProject"

1. Go to "~/HT32_STD_5xxxx_FWLib_V1.5.1_7084/example/GPIO/InputOutput".

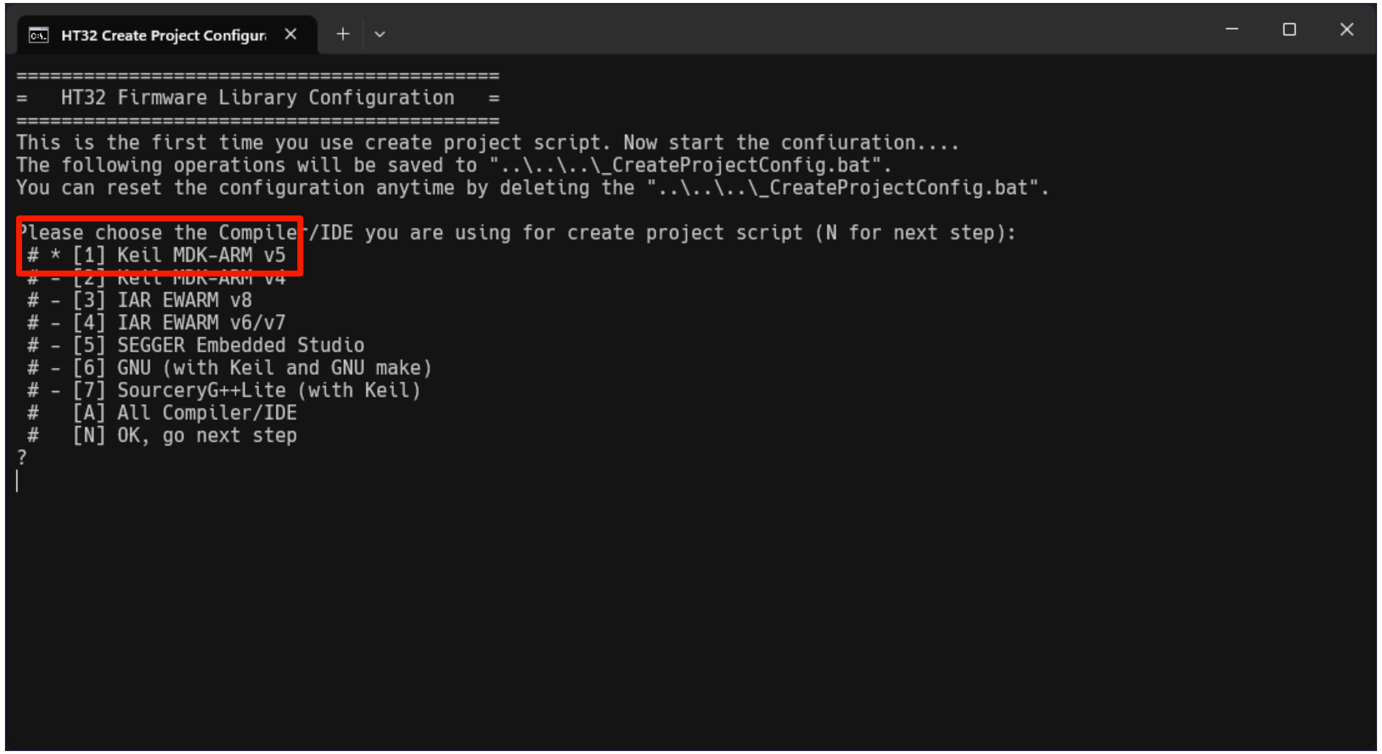
2. Double click "_CreateProject.bat".

3. Type "1" and "N" after prompt appears.



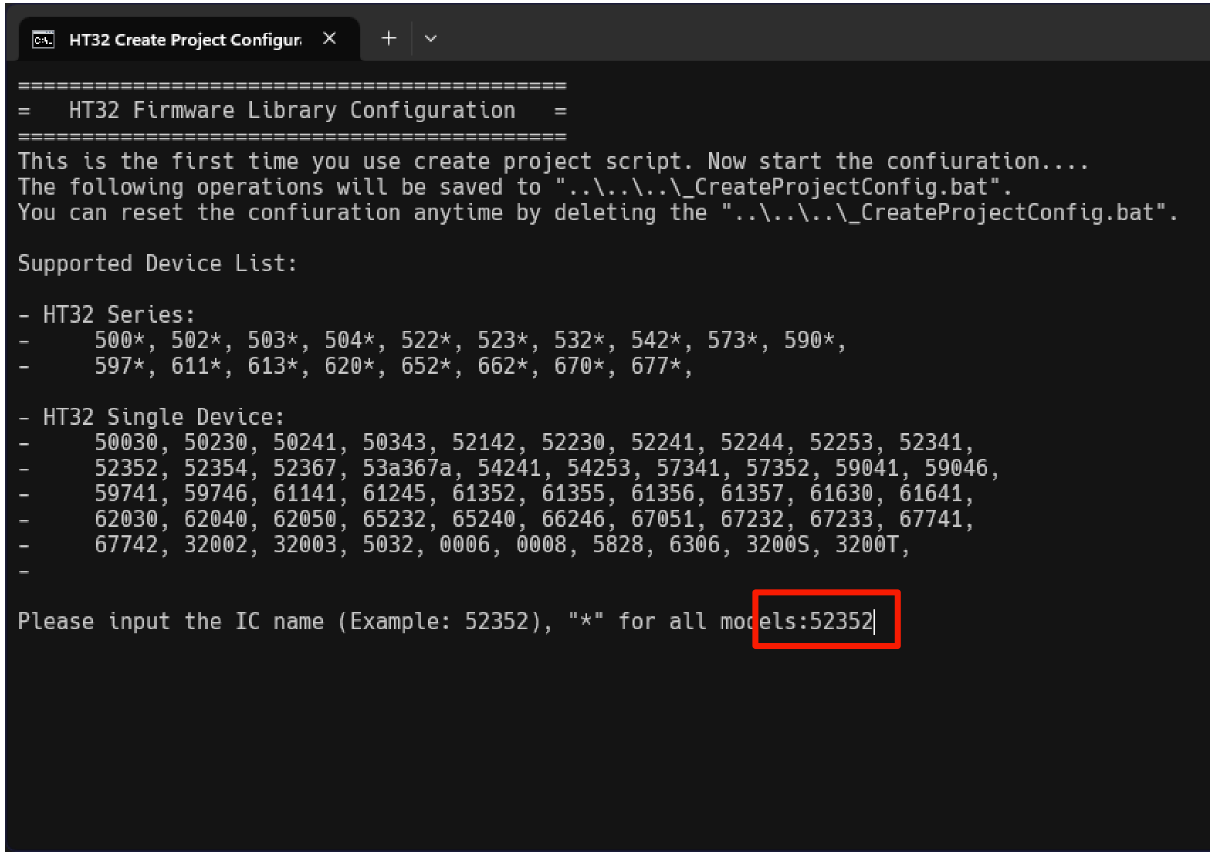
:1 › HT32_M0p_V20230621 › Firmware_Library › HT32_STD_5xxxx_FWLib_V1.4.2_6992 › example › GPIO › InputOutput ›

名稱	日期	類型	大小	標籤
MDK_ARMv5	2023/8/2 下午 01:48	檔案資料夾		
MDK_ARMv537	2023/8/2 下午 01:48	檔案資料夾		
_ClearProject.bat	2023/8/2 下午 01:48	Windows 批次檔案	5 KB	
_ClearTarget.bat	2023/8/1 下午 05:12	Windows 批次檔案	3 KB	
_CreateProject.bat	2021/2/1 下午 08:26	Windows 批次檔案	3 KB	
_CreateProjectScript.b...	2023/8/1 下午 05:12	Windows 批次檔案	28 KB	
ht32_board_config.h	2023/4/11 上午 10:44	C Header 來源檔案	44 KB	
ht32f5xxxx_01_it.c	2018/8/3 下午 03:35	C 來源檔案	6 KB	
ht32f5xxxx_01_usbdco...	2023/8/1 下午 05:10	C Header 來源檔案	18 KB	
ht32f5xxxx_02_usbdco...	2023/8/1 下午 05:10	C Header 來源檔案	22 KB	
ht32f5xxxx_conf.h	2023/8/1 下午 05:10	C Header 來源檔案	19 KB	
main.c	2021/12/23 上午 12:50	C 來源檔案	10 KB	
readme.txt	2023/8/1 下午 05:08	文字文件	2 KB	
system_ht32f5xxxx_03.c	2023/8/1 下午 05:10	C 來源檔案	21 KB	

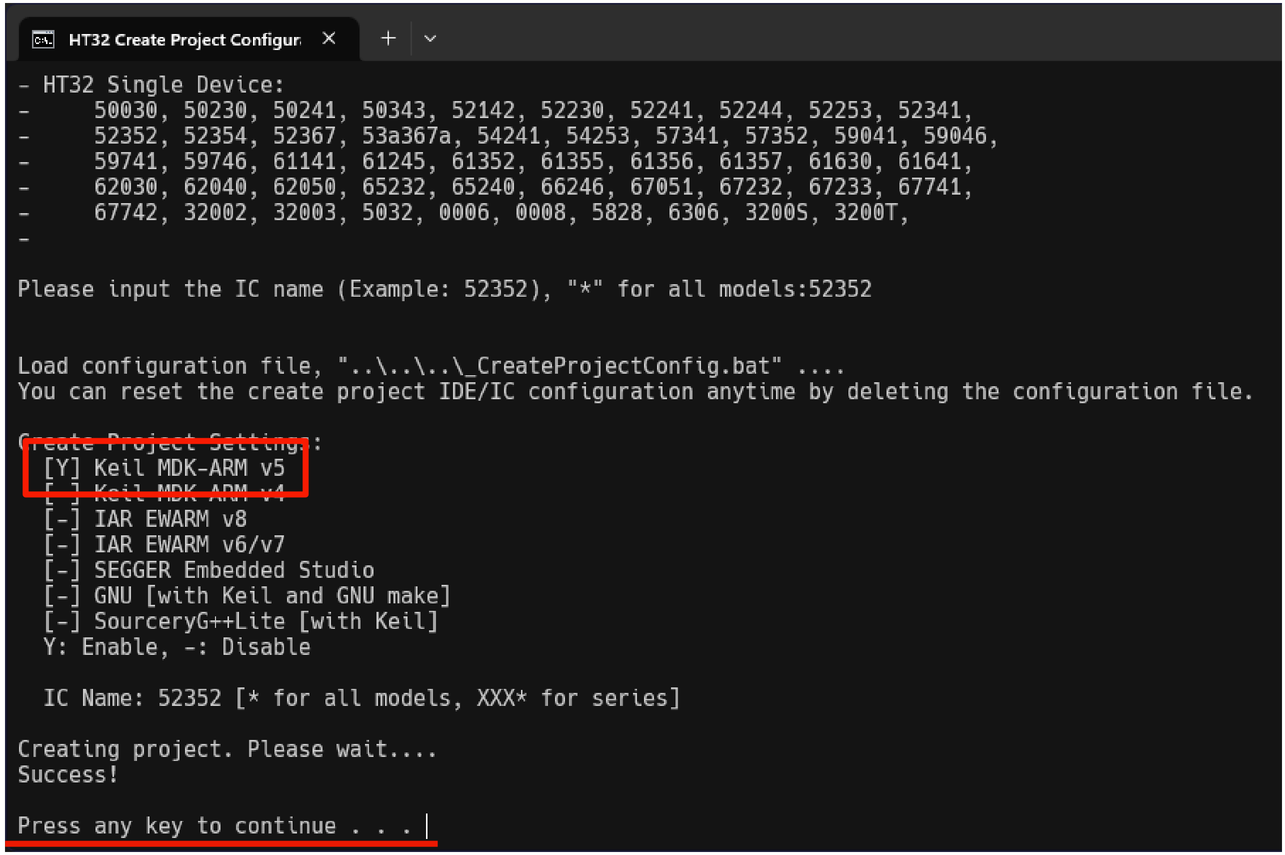


```
============================================
=   HT32 Firmware Library Configuration   =
============================================
This is the first time you use create project script. Now start the confiuration....
The following operations will be saved to "..\..\..\_CreateProjectConfig.bat".
You can reset the configuration anytime by deleting the "..\..\..\_CreateProjectConfig.bat".

Please choose the Compiler/IDE you are using for create project script (N for next step):
 # * [1] Keil MDK-ARM v5
 # - [2] Keil MDK-ARM v4
 # - [3] IAR EWARM v8
 # - [4] IAR EWARM v6/v7
 # - [5] SEGGER Embedded Studio
 # - [6] GNU (with Keil and GNU make)
 # - [7] SourceryG++Lite (with Keil)
 #   [A] All Compiler/IDE
 #   [N] OK, go next step
?
```

2. Type options

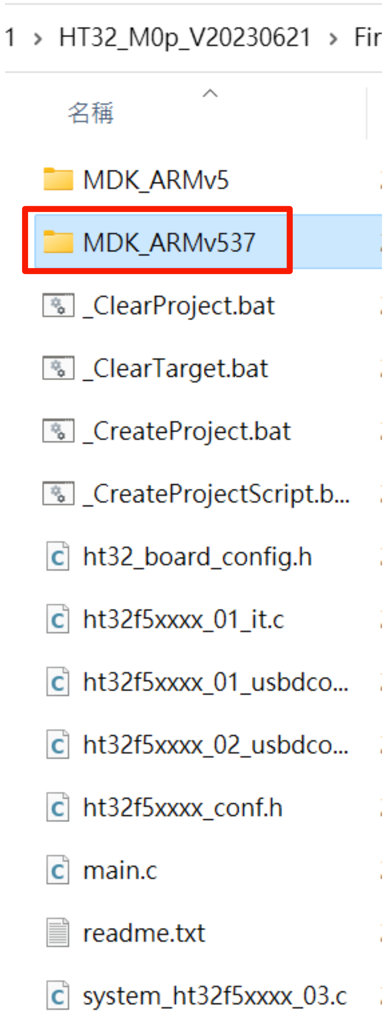


```
==========================================
=   HT32 Firmware Library Configuration   =
==========================================
This is the first time you use create project script. Now start the confiuration....
The following operations will be saved to "..\..\..\_CreateProjectConfig.bat".
You can reset the confiuration anytime by deleting the "..\..\..\_CreateProjectConfig.bat".

Supported Device List:

- HT32 Series:
-     500*, 502*, 503*, 504*, 522*, 523*, 532*, 542*, 573*, 590*,
-     597*, 611*, 613*, 620*, 652*, 662*, 670*, 677*,

- HT32 Single Device:
-     50030, 50230, 50241, 50343, 52142, 52230, 52241, 52244, 52253, 52341,
-     52352, 52354, 52367, 53a367a, 54241, 54253, 57341, 57352, 59041, 59046,
-     59741, 59746, 61141, 61245, 61352, 61355, 61356, 61357, 61630, 61641,
-     62030, 62040, 62050, 65232, 65240, 66246, 67051, 67232, 67233, 67741,
-     67742, 32002, 32003, 5032, 0006, 0008, 5828, 6306, 3200S, 3200T,
-

Please input the IC name (Example: 52352), "*" for all models:52352
```



```
- HT32 Single Device:
-     50030, 50230, 50241, 50343, 52142, 52230, 52241, 52244, 52253, 52341,
-     52352, 52354, 52367, 53a367a, 54241, 54253, 57341, 57352, 59041, 59046,
-     59741, 59746, 61141, 61245, 61352, 61355, 61356, 61357, 61630, 61641,
-     62030, 62040, 62050, 65232, 65240, 66246, 67051, 67232, 67233, 67741,
-     67742, 32002, 32003, 5032, 0006, 0008, 5828, 6306, 3200S, 3200T,
-

Please input the IC name (Example: 52352), "*" for all models:52352


Load configuration file, "..\..\..\_CreateProjectConfig.bat" ....
You can reset the create project IDE/IC configuration anytime by deleting the configuration file.

Create Project Settings:
  [Y] Keil MDK-ARM v5
  [ ] Keil MDK-ARM v4
  [-] IAR EWARM v8
  [-] IAR EWARM v6/v7
  [-] SEGGER Embedded Studio
  [-] GNU [with Keil and GNU make]
  [-] SourceryG++Lite [with Keil]
  Y: Enable, -: Disable

  IC Name: 52352 [* for all models, XXX* for series]

Creating project. Please wait....
Success!

Press any key to continue . . .
```

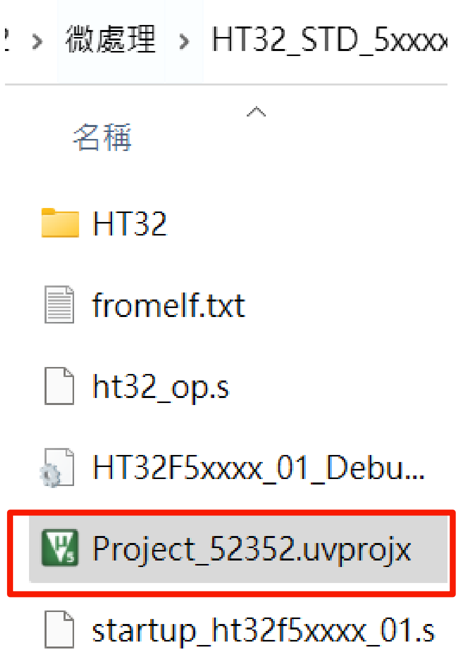
3. Launch project



1 › HT32_M0p_V20230621 › Fir

名稱

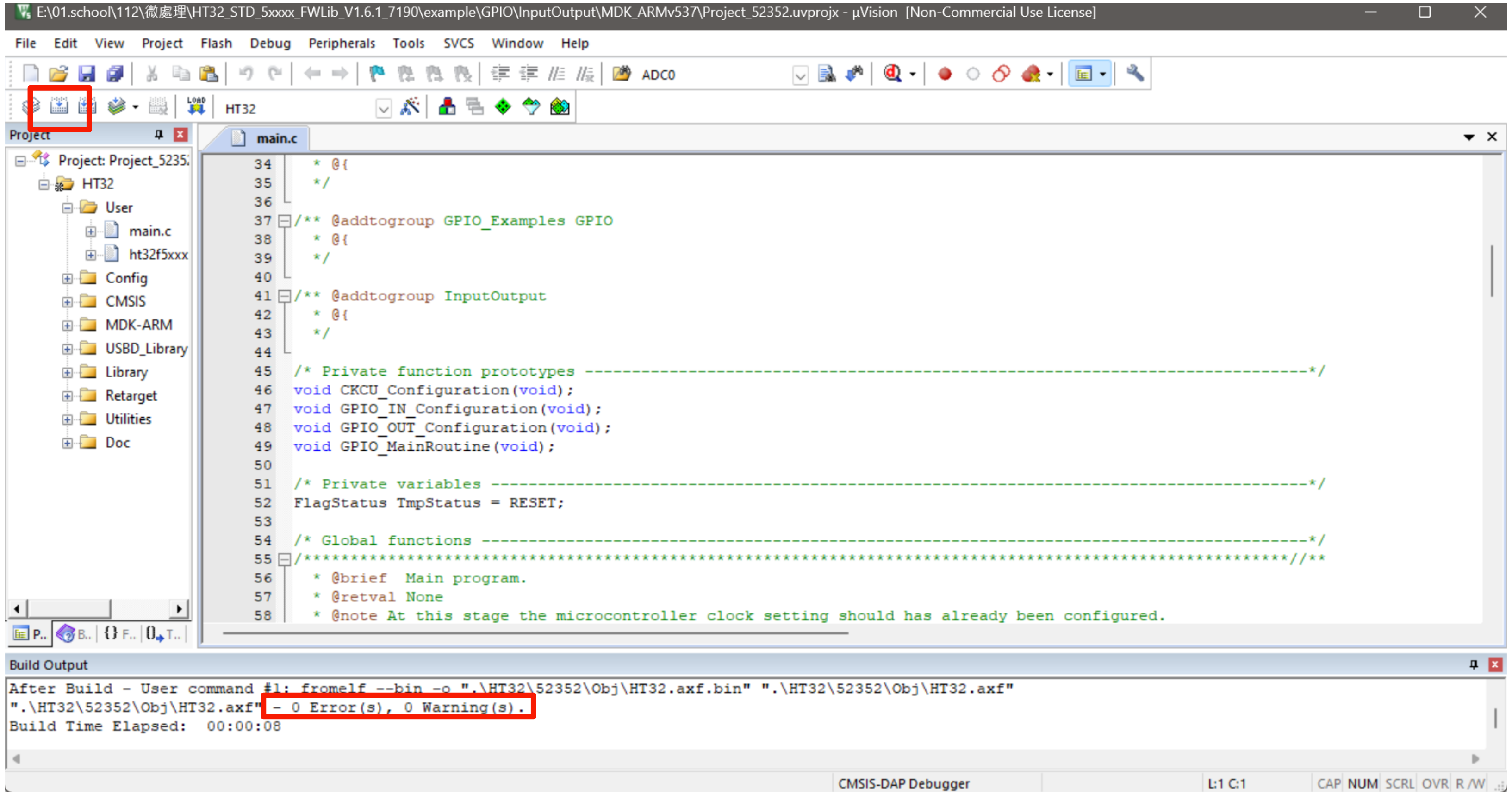
- MDK_ARMv5
- MDK_ARMv537
- _ClearProject.bat
- _ClearTarget.bat
- _CreateProject.bat
- _CreateProjectScript.b...
- ht32_board_config.h
- ht32f5xxxx_01_it.c
- ht32f5xxxx_01_usbdco...
- ht32f5xxxx_02_usbdco...
- ht32f5xxxx_conf.h
- main.c
- readme.txt
- system_ht32f5xxxx_03.c



› 微處理 › HT32_STD_5xxxx

名稱

- HT32
- fromelf.txt
- ht32_op.s
- HT32F5xxxx_01_Debu...
- Project_52352.uvprojx
- startup_ht32f5xxxx_01.s



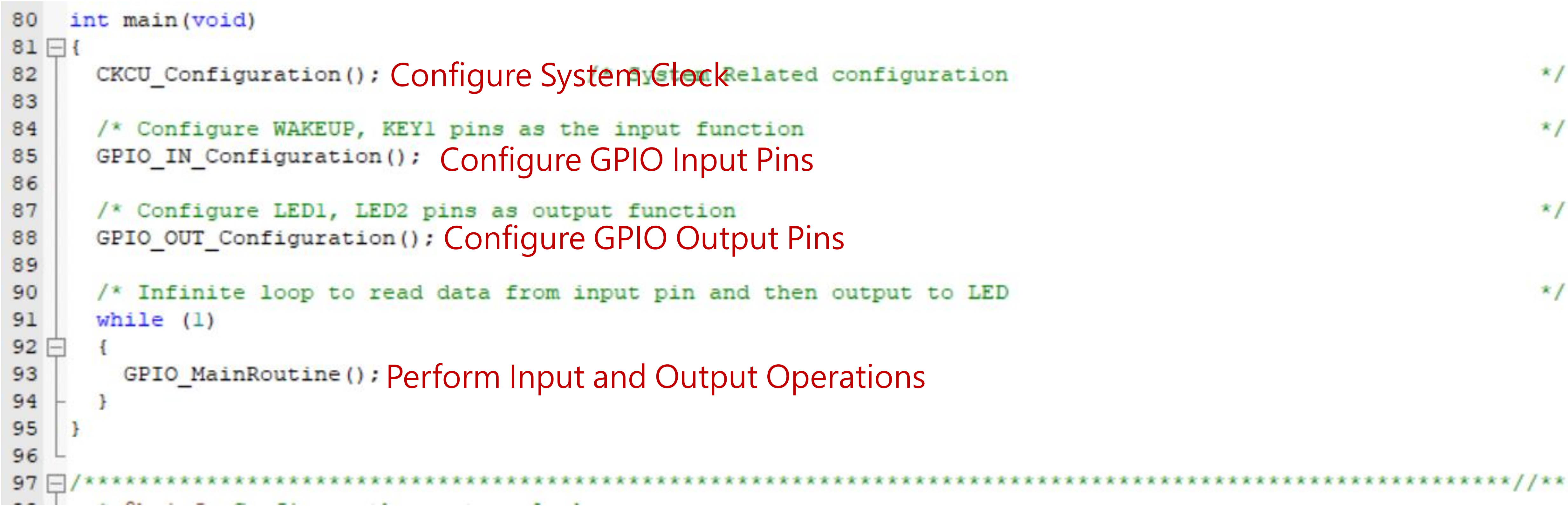
E:\01.school\112\微處理\HT32_STD_5xxxx_FWLib_V1.6.1_7190\example\GPIO\InputOutput\MDK_ARMv537\Project_52352.uvprojx - µVision [Non-Commercial Use License]

```
  * @{
  */

/** @addtogroup GPIO_Examples GPIO
  * @{
  */

/** @addtogroup InputOutput
  * @{
  */

/* Private function prototypes -----------------------------------------------------------------*/
void CKCU_Configuration(void);
void GPIO_IN_Configuration(void);
void GPIO_OUT_Configuration(void);
void GPIO_MainRoutine(void);

/* Private variables ---------------------------------------------------------------------------*/
FlagStatus TmpStatus = RESET;

/* Global functions ----------------------------------------------------------------------------*/
/*********************************************************************************************//**
  * @brief  Main program.
  * @retval None
  * @note At this stage the microcontroller clock setting should has already been configured.
```

Build Output

```
After Build - User command #1: fromelf --bin -o ".\HT32\52352\Obj\HT32.axf.bin" ".\HT32\52352\Obj\HT32.axf"
".\HT32\52352\Obj\HT32.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:08
```

main.c



```c
int main(void)
{
  CKCU_Configuration();               /* System Related configuration                                  */

  /* Configure WAKEUP, KEY1 pins as the input function                                                 */
  GPIO_IN_Configuration();

  /* Configure LED1, LED2 pins as output function                                                      */
  GPIO_OUT_Configuration();

  /* Infinite loop to read data from input pin and then output to LED                                  */
  while (1)
  {
    GPIO_MainRoutine();
  }
}

/*******************************************************************************************************//**
```

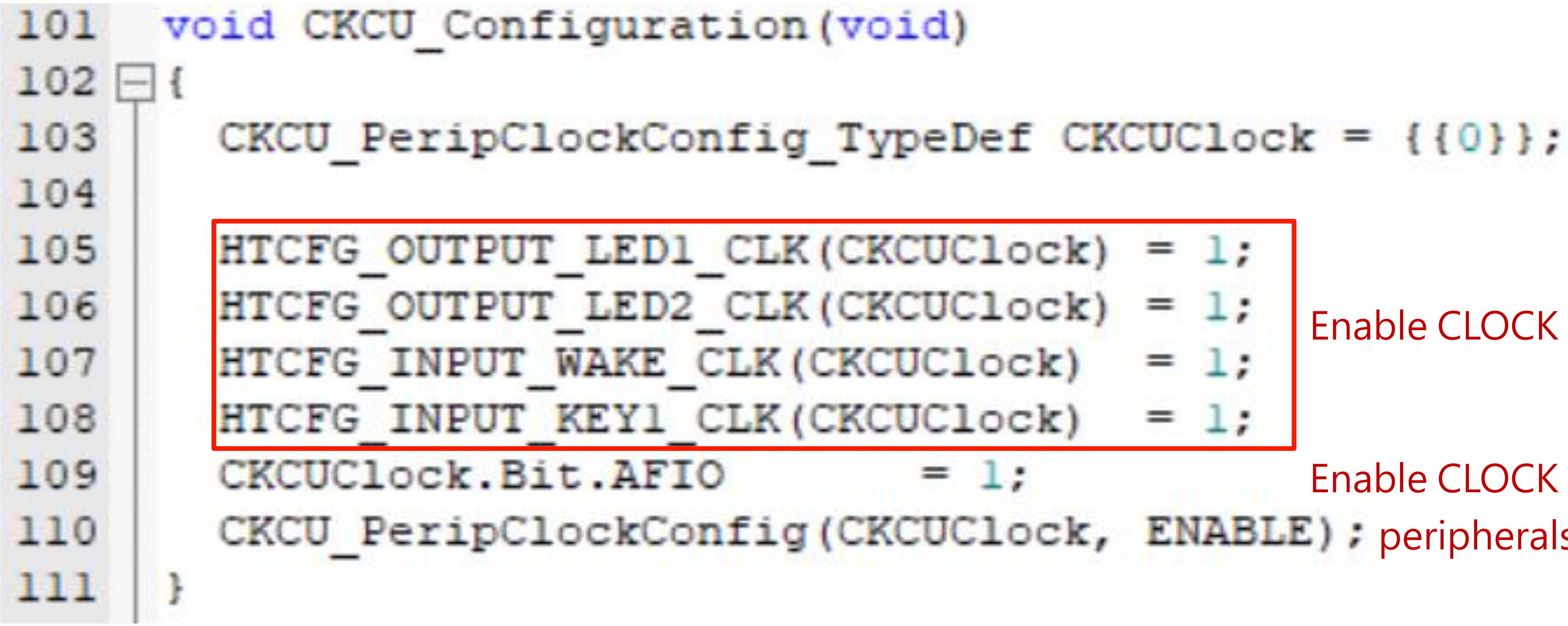
Configure System Clock

Configure GPIO Input Pins

Configure GPIO Output Pins

Perform Input and Output Operations

System clock



```
void CKCU_Configuration(void)
{
  CKCU_PeripClockConfig_TypeDef CKCUClock = {{0}};

  HTCFG_OUTPUT_LED1_CLK(CKCUClock)  = 1;
  HTCFG_OUTPUT_LED2_CLK(CKCUClock)  = 1;
  HTCFG_INPUT_WAKE_CLK(CKCUClock)   = 1;
  HTCFG_INPUT_KEY1_CLK(CKCUClock)   = 1;
  CKCUClock.Bit.AFIO            = 1;
  CKCU_PeripClockConfig(CKCUClock, ENABLE);
}
```

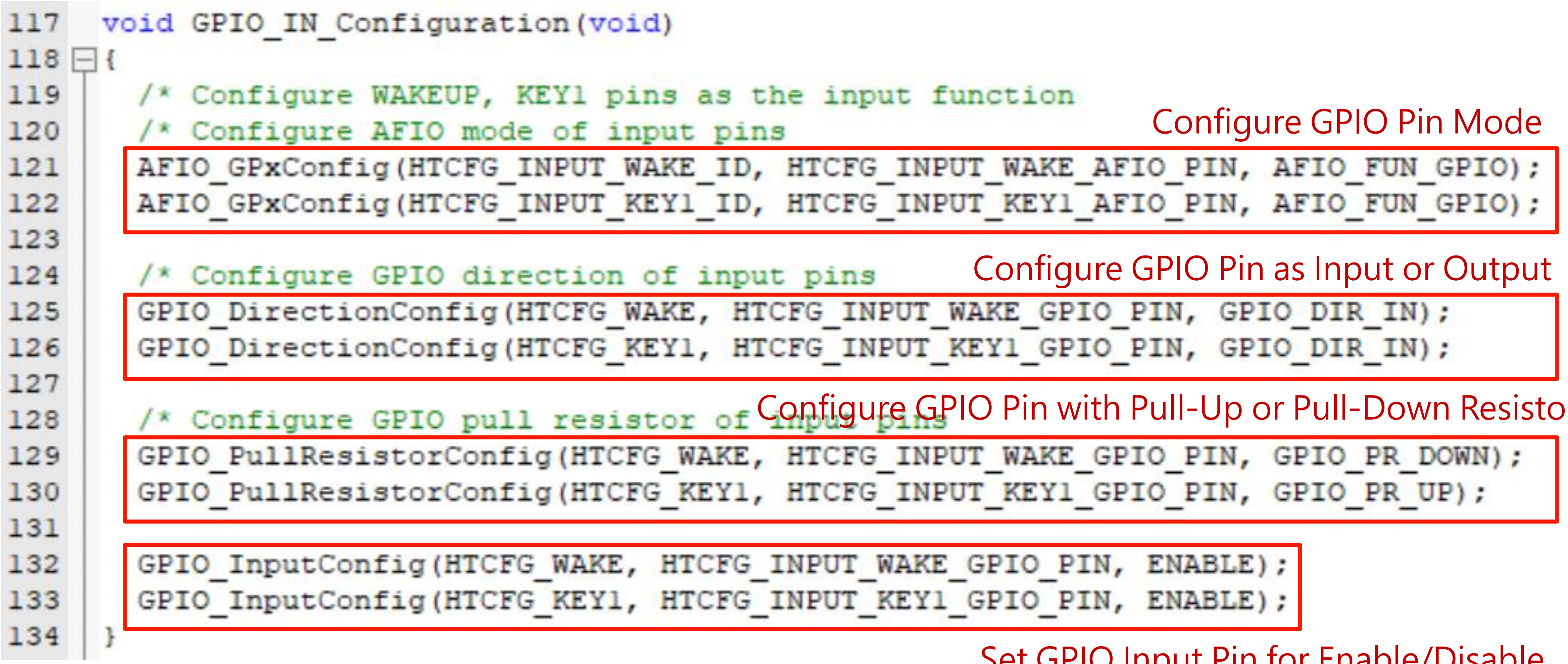
Enable CLOCK

Enable CLOCK Port

Enable CLOCK Multiplexing

peripherals clock

Input Configuration



```
void GPIO_IN_Configuration(void)
{
  /* Configure WAKEUP, KEY1 pins as the input function
  /* Configure AFIO mode of input pins
  AFIO_GPxConfig(HTCFG_INPUT_WAKE_ID, HTCFG_INPUT_WAKE_AFIO_PIN, AFIO_FUN_GPIO);
  AFIO_GPxConfig(HTCFG_INPUT_KEY1_ID, HTCFG_INPUT_KEY1_AFIO_PIN, AFIO_FUN_GPIO);

  /* Configure GPIO direction of input pins
  GPIO_DirectionConfig(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN, GPIO_DIR_IN);
  GPIO_DirectionConfig(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN, GPIO_DIR_IN);

  /* Configure GPIO pull resistor of input pins
  GPIO_PullResistorConfig(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN, GPIO_PR_DOWN);
  GPIO_PullResistorConfig(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN, GPIO_PR_UP);

  GPIO_InputConfig(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN, ENABLE);
  GPIO_InputConfig(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN, ENABLE);
}
```

Configure GPIO Pin Mode

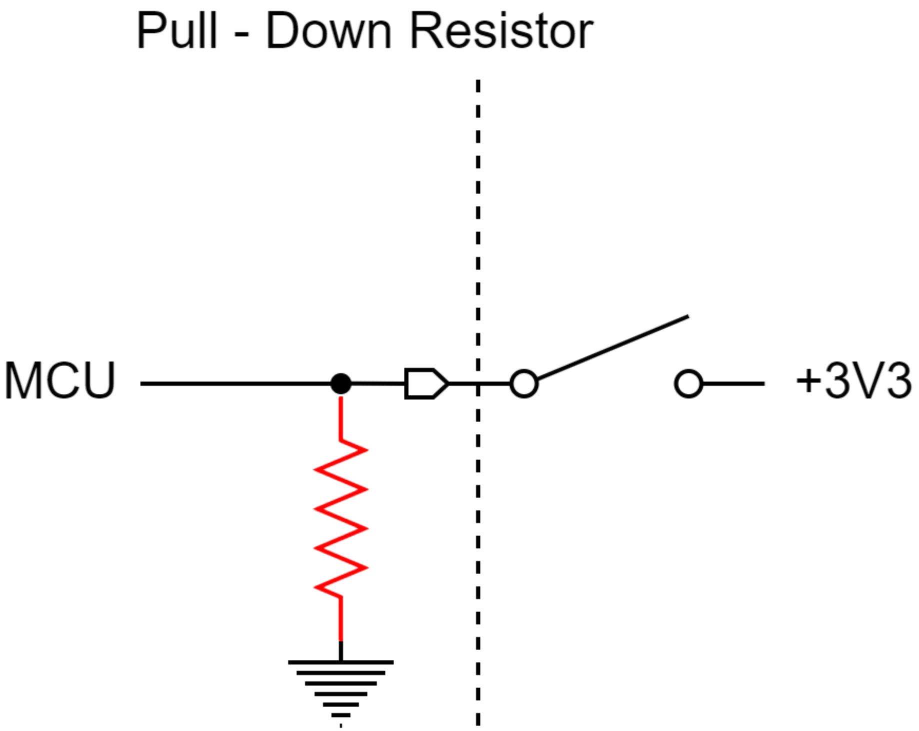
Configure GPIO Pin as Input or Output

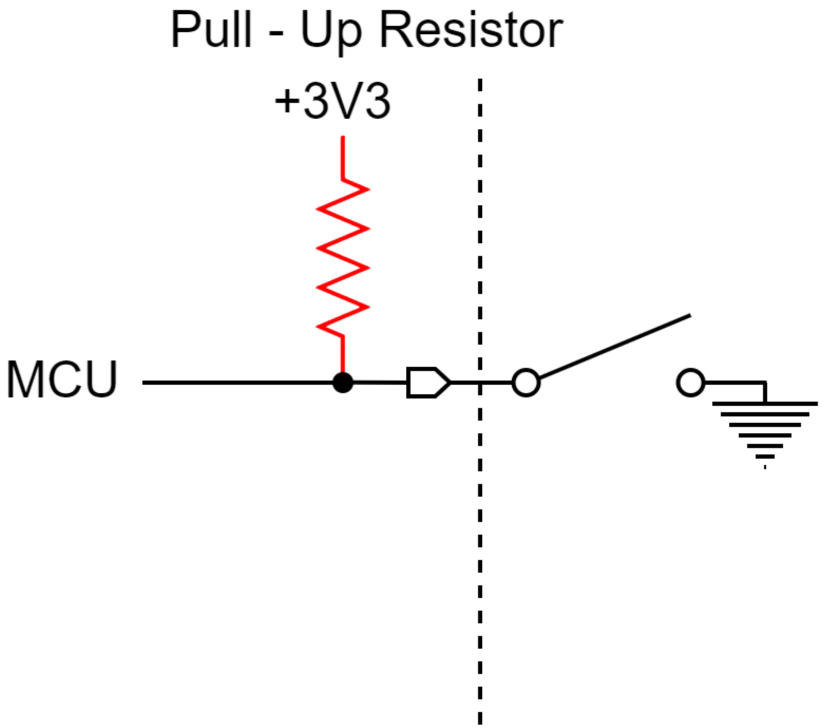
Configure GPIO Pin with Pull-Up or Pull-Down Resistor

Set GPIO Input Pin for Enable/Disable

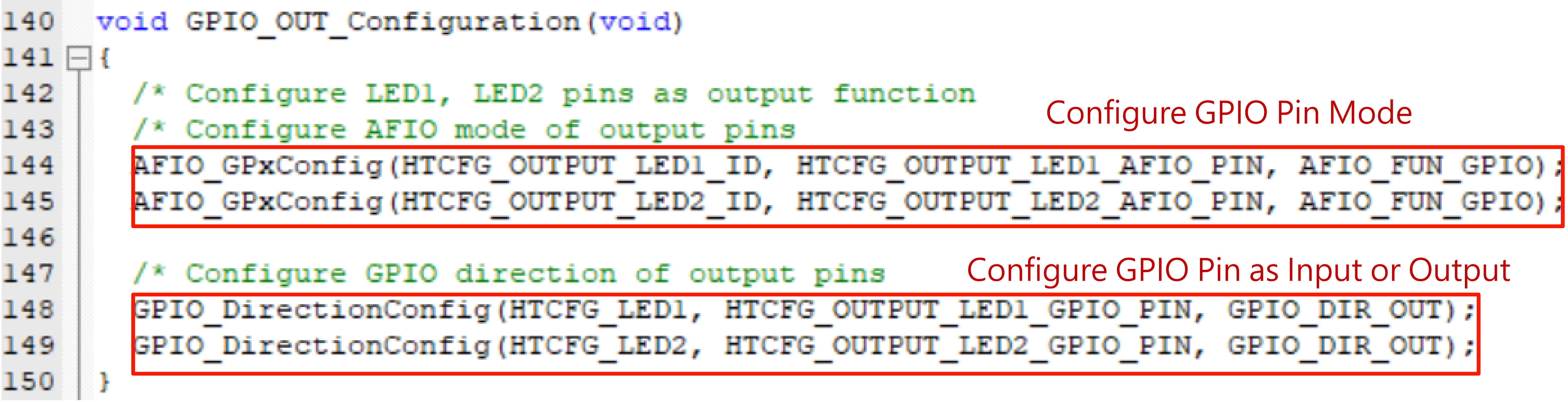
Pull-Up Resistor vs. Pull-Down Resistor

https://www.youtube.com/watch?v=k_GAuSONCqo&ab_channel=%E8%80%81%E6%98%8E





Output Configuration

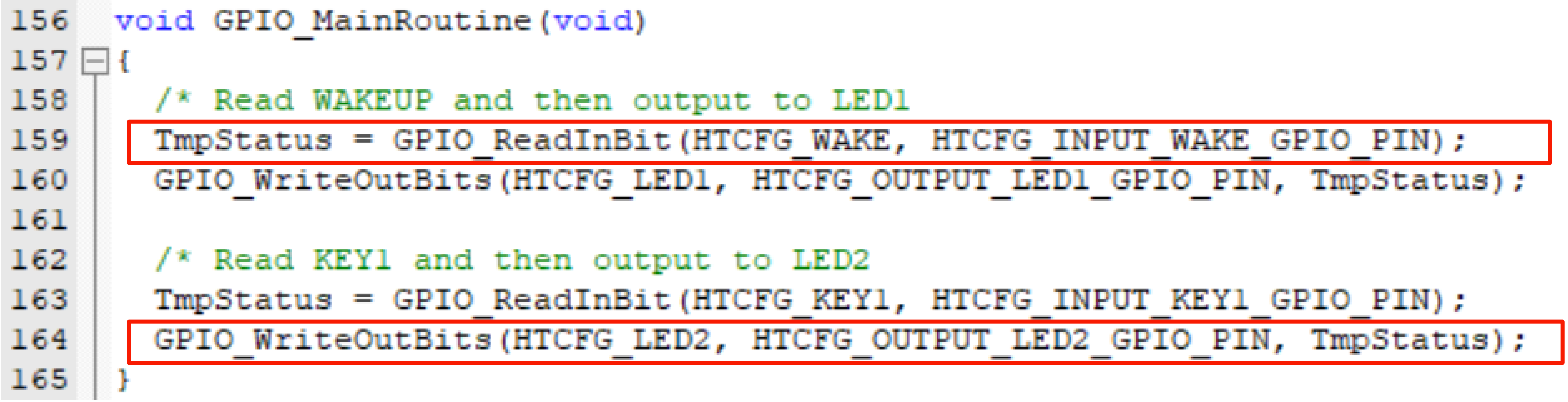


```
void GPIO_OUT_Configuration(void)
{
  /* Configure LED1, LED2 pins as output function
  /* Configure AFIO mode of output pins
  AFIO_GPxConfig(HTCFG_OUTPUT_LED1_ID, HTCFG_OUTPUT_LED1_AFIO_PIN, AFIO_FUN_GPIO);
  AFIO_GPxConfig(HTCFG_OUTPUT_LED2_ID, HTCFG_OUTPUT_LED2_AFIO_PIN, AFIO_FUN_GPIO);

  /* Configure GPIO direction of output pins
  GPIO_DirectionConfig(HTCFG_LED1, HTCFG_OUTPUT_LED1_GPIO_PIN, GPIO_DIR_OUT);
  GPIO_DirectionConfig(HTCFG_LED2, HTCFG_OUTPUT_LED2_GPIO_PIN, GPIO_DIR_OUT);
}
```

Configure GPIO Pin Mode

Configure GPIO Pin as Input or Output

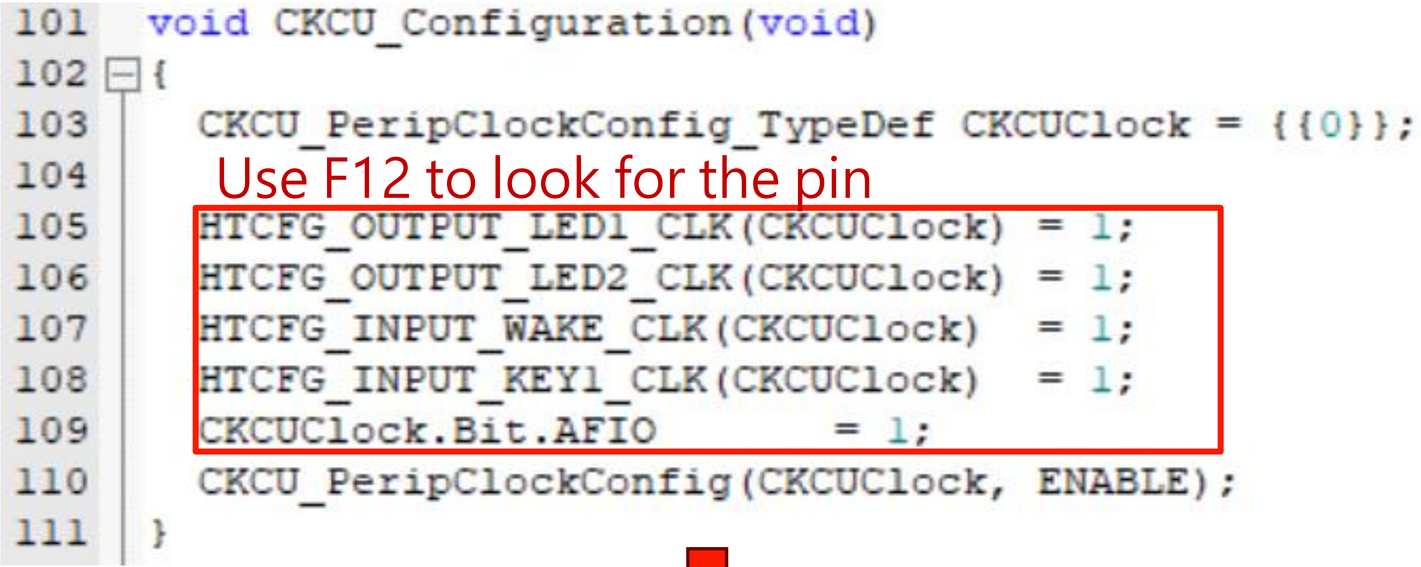
Subroutine Function



```
void GPIO_MainRoutine(void)
{
  /* Read WAKEUP and then output to LED1
  TmpStatus = GPIO_ReadInBit(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN);
  GPIO_WriteOutBits(HTCFG_LED1, HTCFG_OUTPUT_LED1_GPIO_PIN, TmpStatus);

  /* Read KEY1 and then output to LED2
  TmpStatus = GPIO_ReadInBit(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN);
  GPIO_WriteOutBits(HTCFG_LED2, HTCFG_OUTPUT_LED2_GPIO_PIN, TmpStatus);
}
```

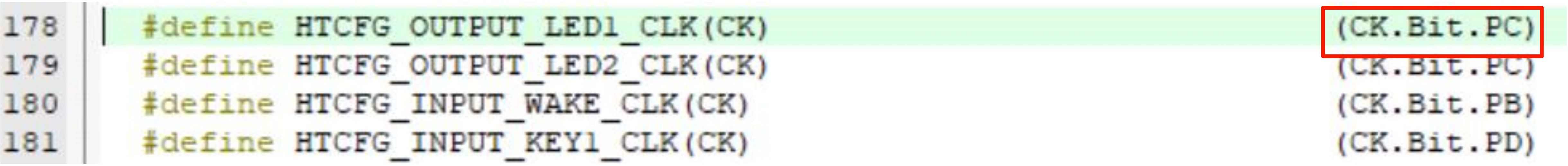
GPIO_ReadInBit（PA ~ PD、PIN）:Read Input Data from a Specified Pin
GPIO_WriteOutBits（PA ~ PD、PIN，SET/RESET ）:Write Output Data from a Specified Pin

Modify Definition Name



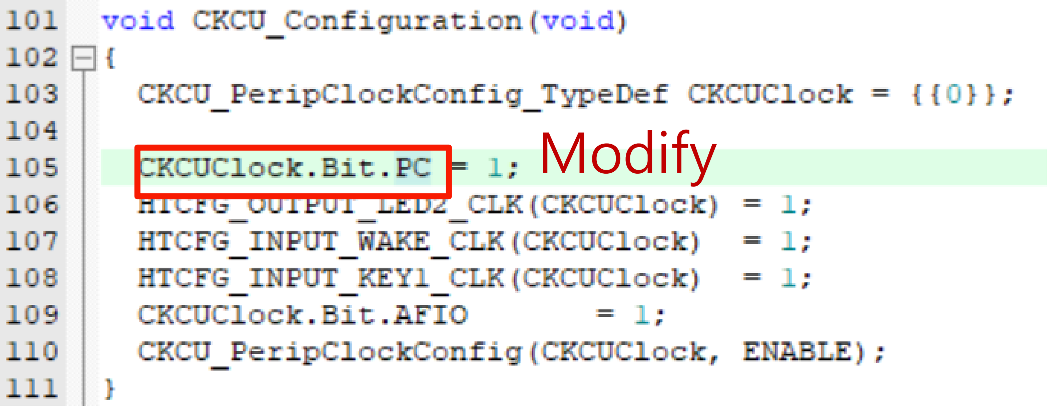
Use F12 to look for the pin

```
void CKCU_Configuration(void)
{
  CKCU_PeripClockConfig_TypeDef CKCUClock = {{0}};

  HTCFG_OUTPUT_LED1_CLK(CKCUClock)  = 1;
  HTCFG_OUTPUT_LED2_CLK(CKCUClock)  = 1;
  HTCFG_INPUT_WAKE_CLK(CKCUClock)   = 1;
  HTCFG_INPUT_KEY1_CLK(CKCUClock)   = 1;
  CKCUClock.Bit.AFIO             = 1;
  CKCU_PeripClockConfig(CKCUClock, ENABLE);
}
```



```
#define HTCFG_OUTPUT_LED1_CLK(CK)                                    (CK.Bit.PC)
#define HTCFG_OUTPUT_LED2_CLK(CK)                                    (CK.Bit.PC)
#define HTCFG_INPUT_WAKE_CLK(CK)                                     (CK.Bit.PB)
#define HTCFG_INPUT_KEY1_CLK(CK)                                     (CK.Bit.PD)
```

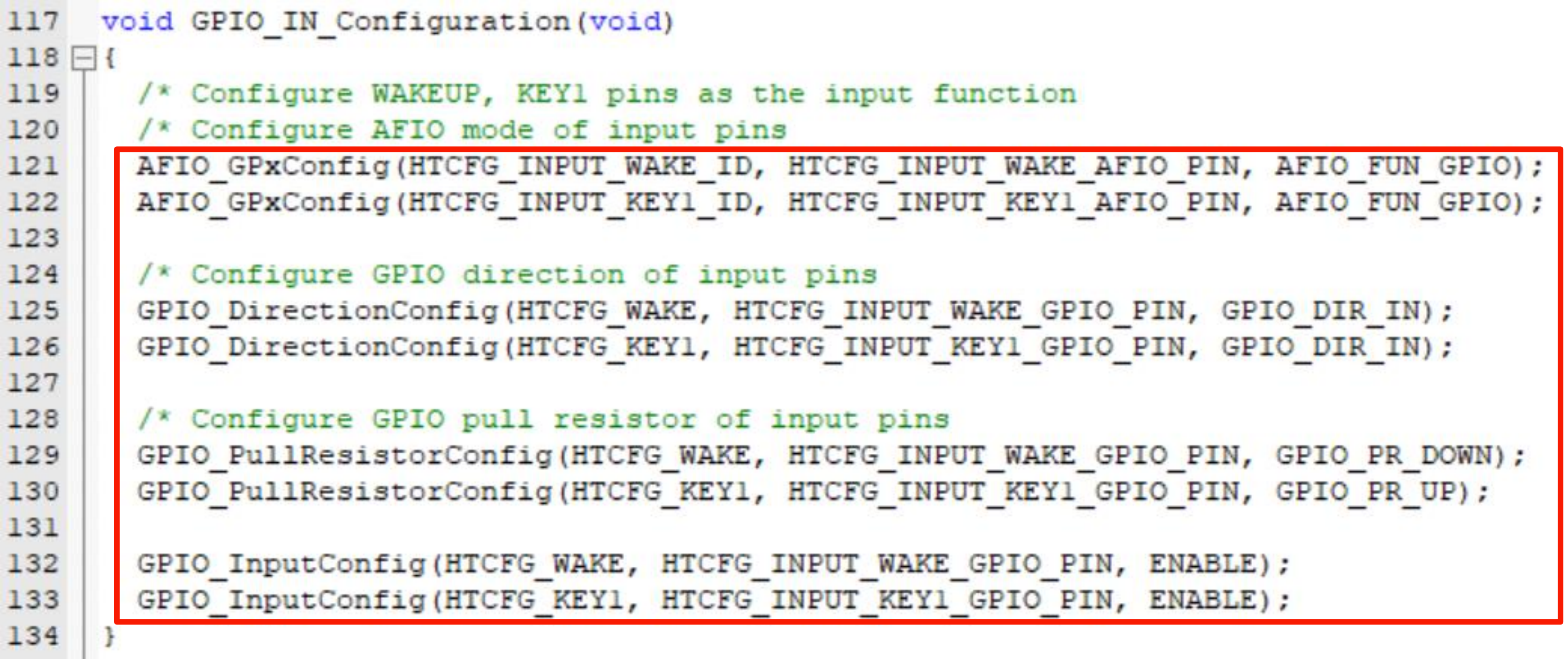
Copy



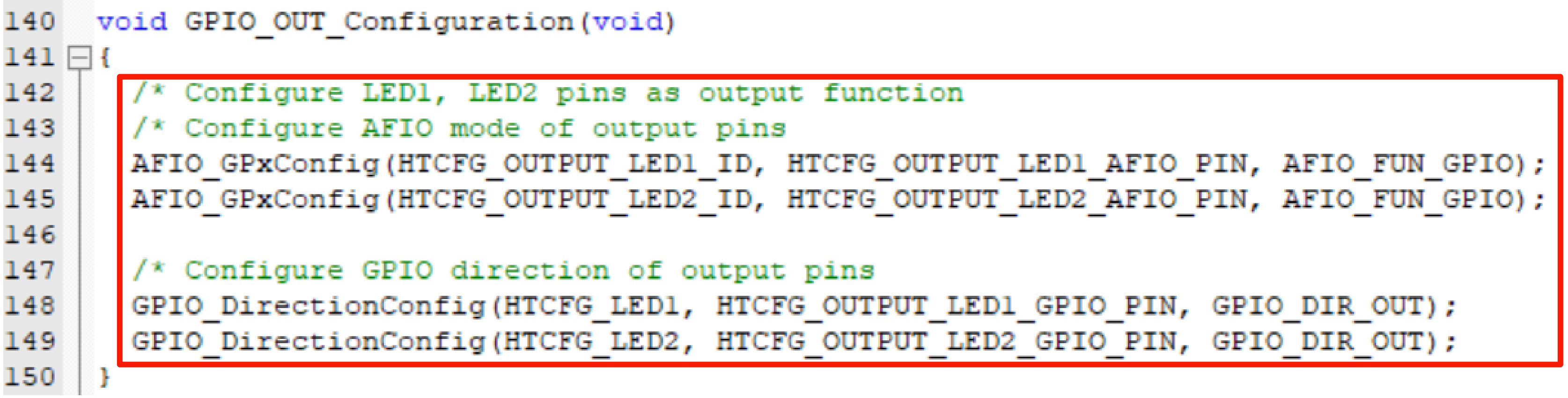
Modify

```
void CKCU_Configuration(void)
{
  CKCU_PeripClockConfig_TypeDef CKCUClock = {{0}};

  CKCUClock.Bit.PC = 1;
  HTCFG_OUTPUT_LED2_CLK(CKCUClock)  = 1;
  HTCFG_INPUT_WAKE_CLK(CKCUClock)   = 1;
  HTCFG_INPUT_KEY1_CLK(CKCUClock)   = 1;
  CKCUClock.Bit.AFIO          = 1;
  CKCU_PeripClockConfig(CKCUClock, ENABLE);
}
```

Also Modify Input and Output



```
void GPIO_IN_Configuration(void)
{
  /* Configure WAKEUP, KEY1 pins as the input function
  /* Configure AFIO mode of input pins
  AFIO_GPxConfig(HTCFG_INPUT_WAKE_ID, HTCFG_INPUT_WAKE_AFIO_PIN, AFIO_FUN_GPIO);
  AFIO_GPxConfig(HTCFG_INPUT_KEY1_ID, HTCFG_INPUT_KEY1_AFIO_PIN, AFIO_FUN_GPIO);

  /* Configure GPIO direction of input pins
  GPIO_DirectionConfig(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN, GPIO_DIR_IN);
  GPIO_DirectionConfig(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN, GPIO_DIR_IN);

  /* Configure GPIO pull resistor of input pins
  GPIO_PullResistorConfig(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN, GPIO_PR_DOWN);
  GPIO_PullResistorConfig(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN, GPIO_PR_UP);

  GPIO_InputConfig(HTCFG_WAKE, HTCFG_INPUT_WAKE_GPIO_PIN, ENABLE);
  GPIO_InputConfig(HTCFG_KEY1, HTCFG_INPUT_KEY1_GPIO_PIN, ENABLE);
}
```

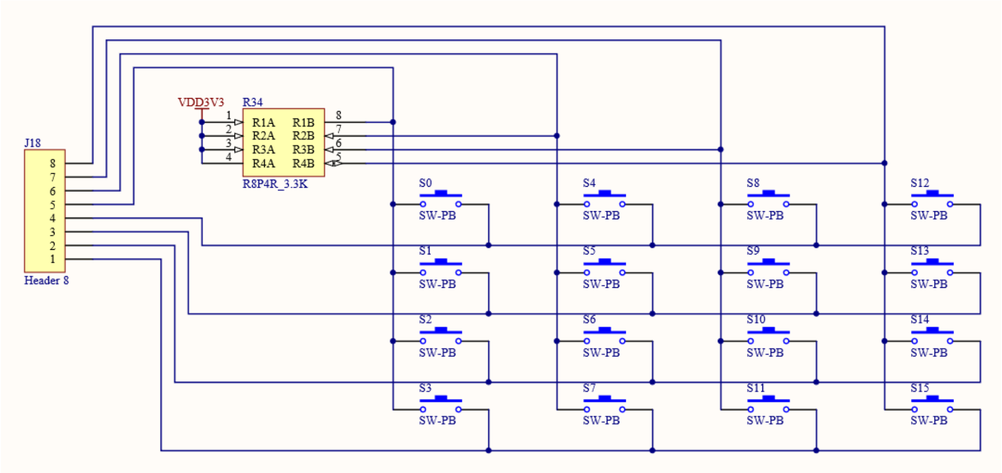


```
void GPIO_OUT_Configuration(void)
{
  /* Configure LED1, LED2 pins as output function
  /* Configure AFIO mode of output pins
  AFIO_GPxConfig(HTCFG_OUTPUT_LED1_ID, HTCFG_OUTPUT_LED1_AFIO_PIN, AFIO_FUN_GPIO);
  AFIO_GPxConfig(HTCFG_OUTPUT_LED2_ID, HTCFG_OUTPUT_LED2_AFIO_PIN, AFIO_FUN_GPIO);

  /* Configure GPIO direction of output pins
  GPIO_DirectionConfig(HTCFG_LED1, HTCFG_OUTPUT_LED1_GPIO_PIN, GPIO_DIR_OUT);
  GPIO_DirectionConfig(HTCFG_LED2, HTCFG_OUTPUT_LED2_GPIO_PIN, GPIO_DIR_OUT);
}
```

Homework W2-2.

https://github.com/CYCU-AIoT-System-Lab/Microcontroller-Experiment/blob/main/w2/GPIO-InputOutput-Experiment_Steps.md

Control LED by buttons



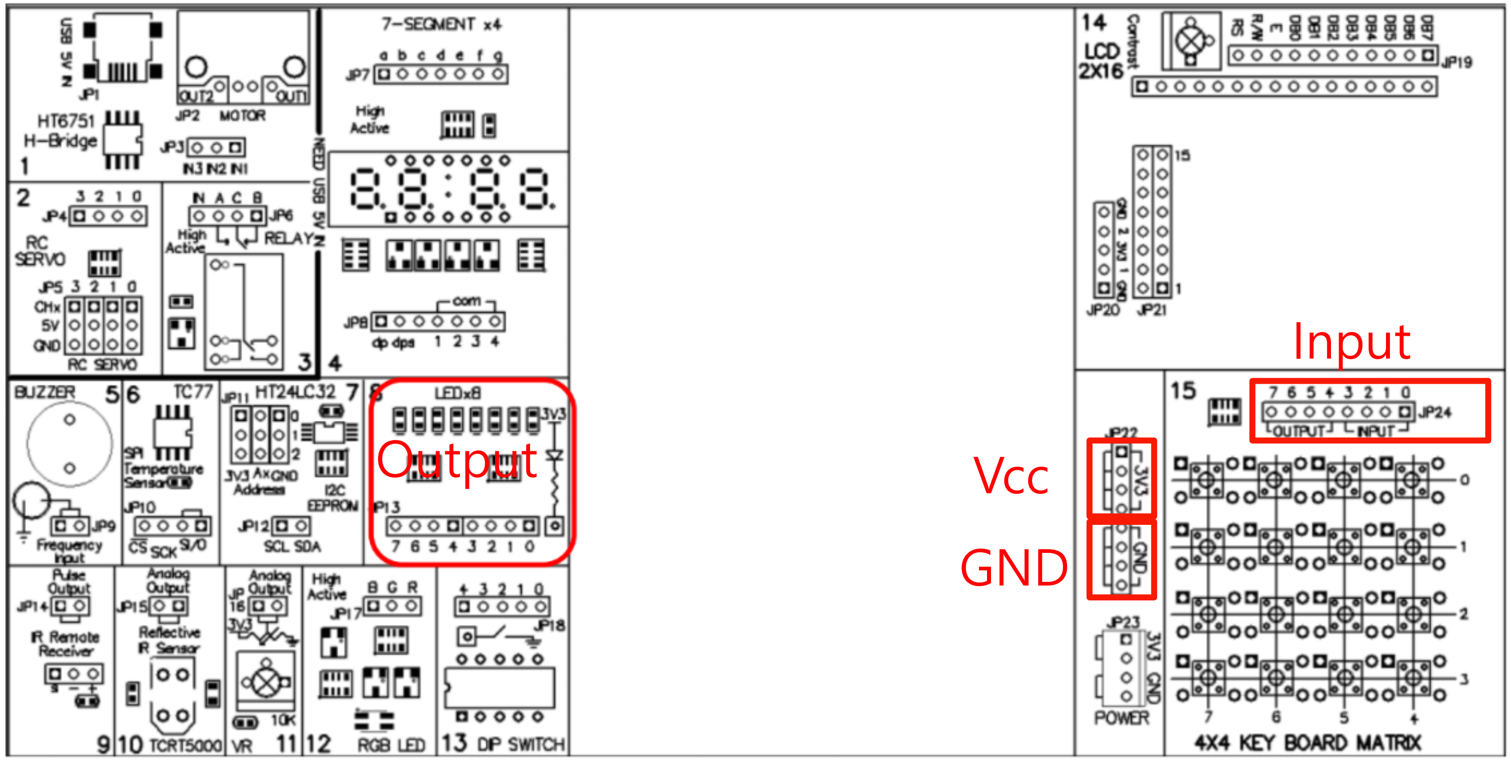
- Objective: Configure two buttons, one turns on LED1, another turns off LED2.
- Hint:

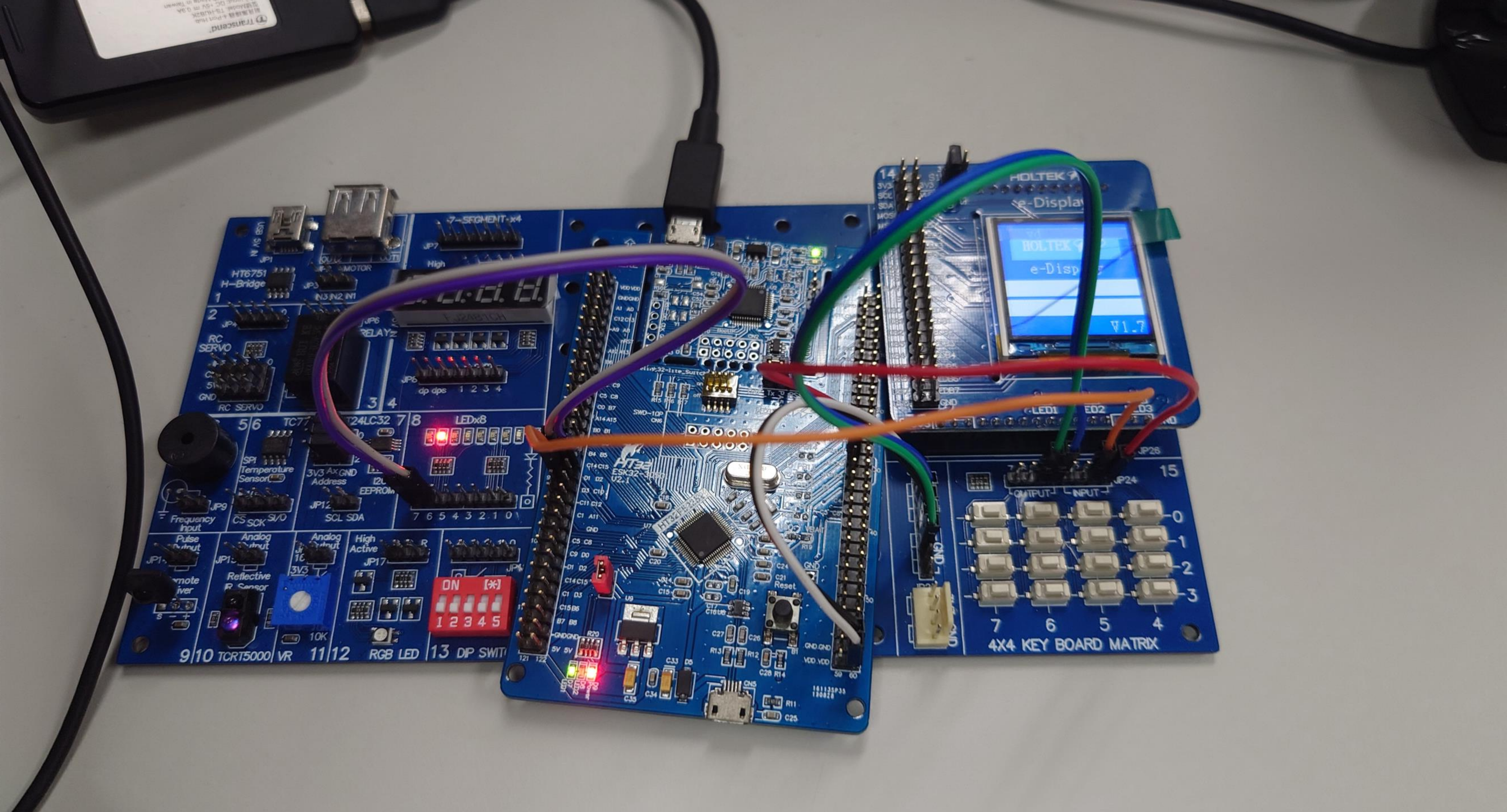
1. Use pin PC14, PC15 to connect to JP13.

2. Each button in key board matrix is connect with both terminal to their corresponding numbered pin in JP24.

3. Use pin PD1, PB12 to connect to JP24 input section.

4. JP24 output section connect to 3V3 and GND.





https://youtu.be/K46_xkXOUrc

Homework W2-4 Bonus Question

Control LED light by the button

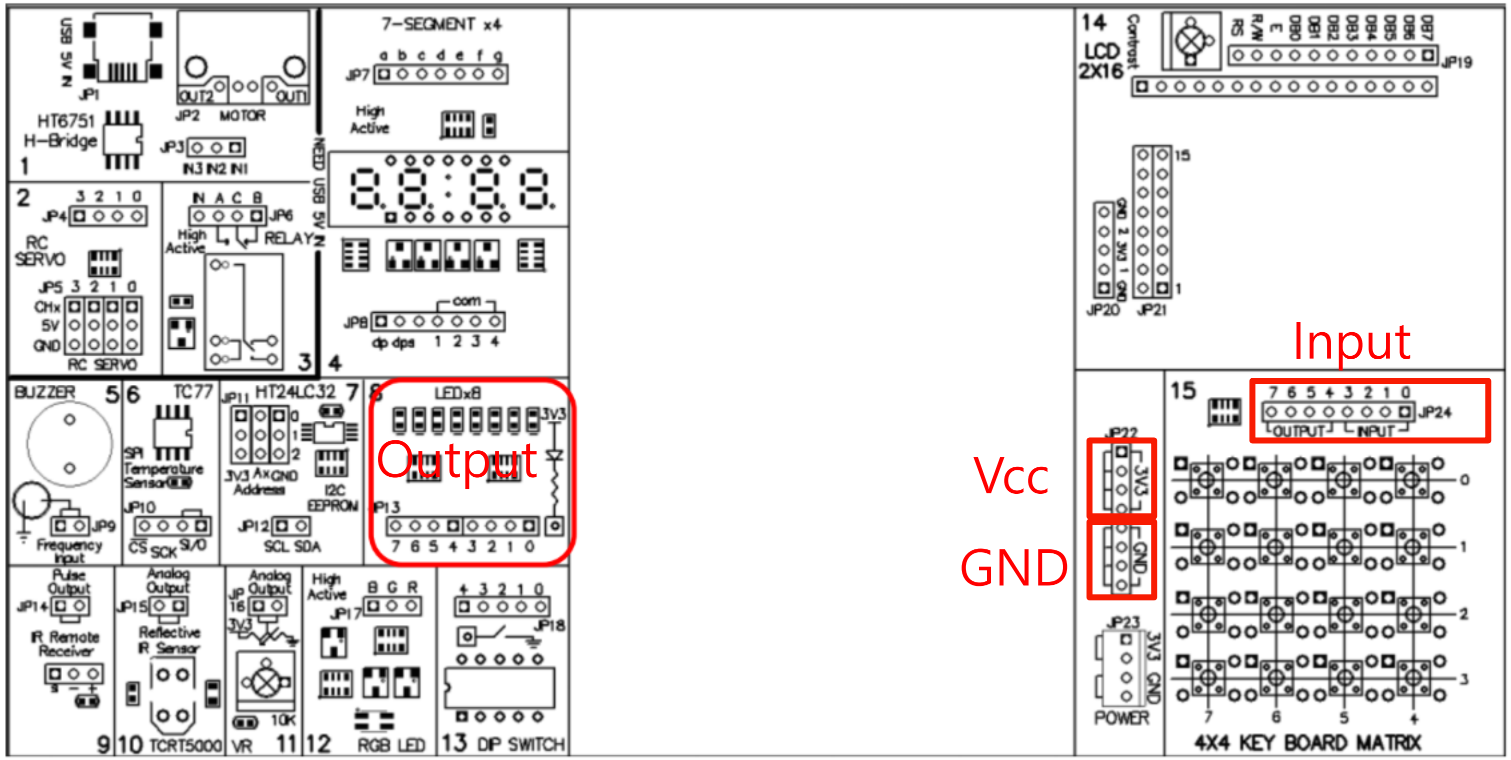
- Objective: LED flickers when the button connected with PA0 is pressed.
- Hint:

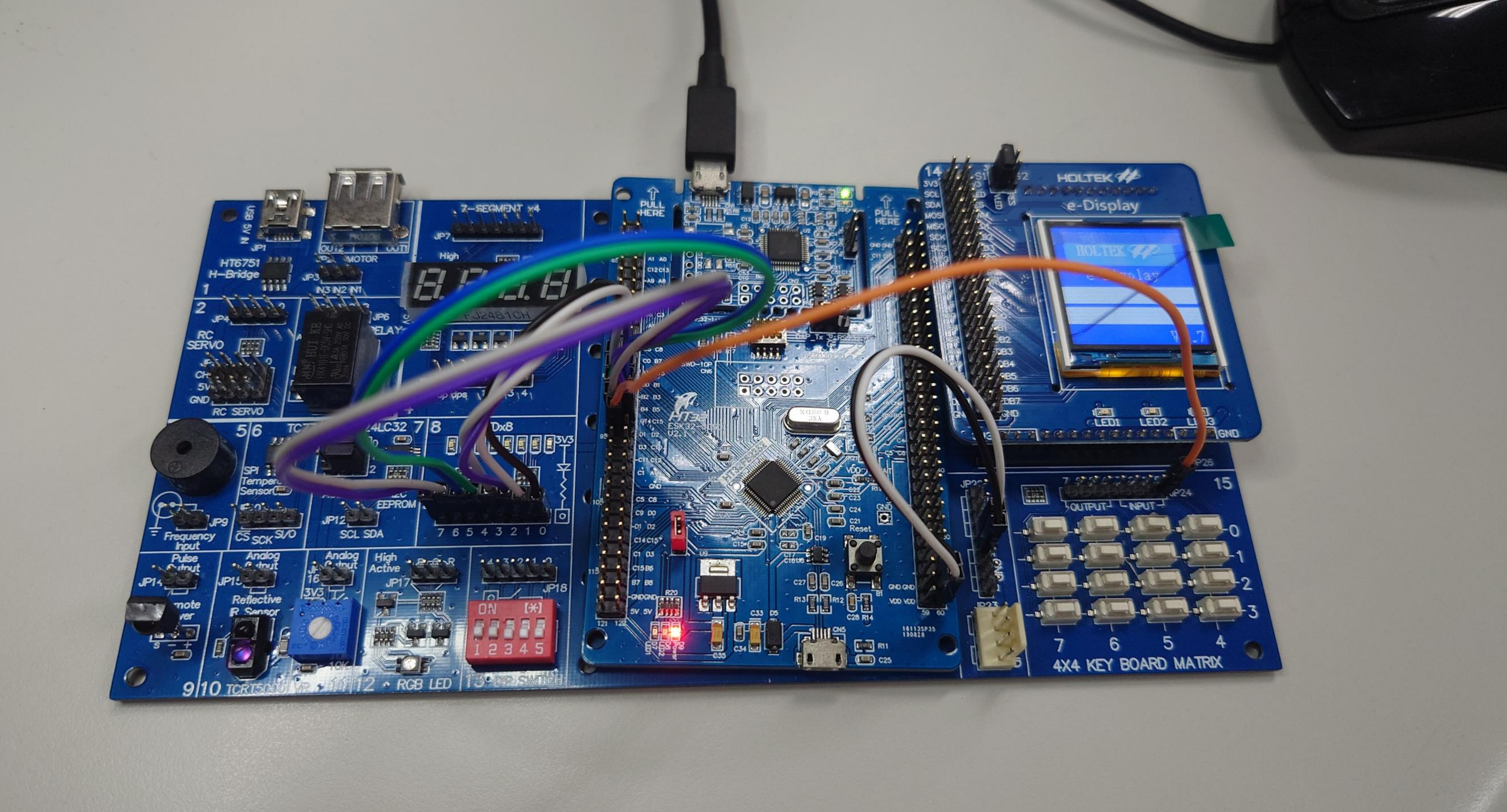
1. PA0 is set as an input.
2. Draw the wiring diagram and

explain what values should be

given to the LEDs

for them to be bright.





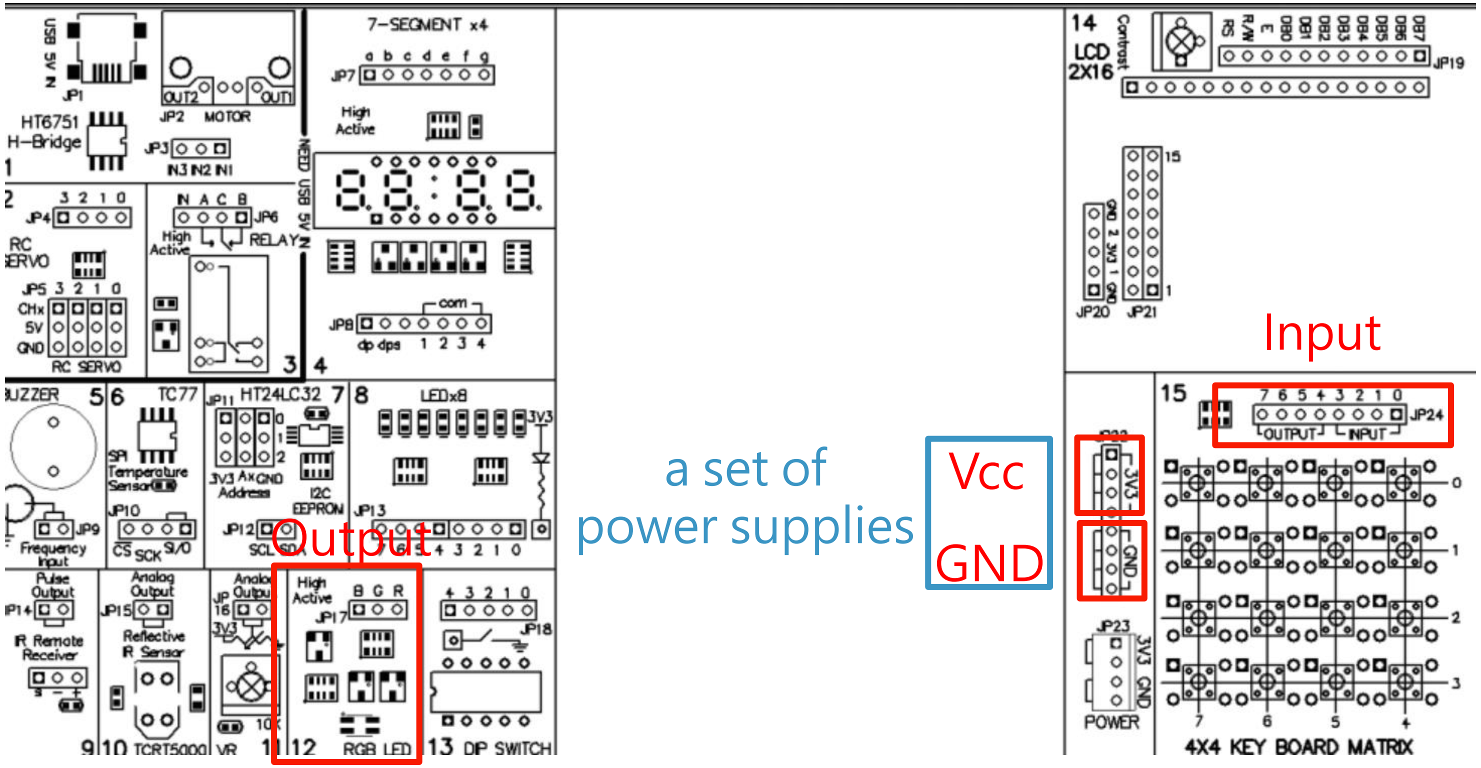
https://youtu.be/FwJsIexCTwQ

Homework 2-5 Bonus Question

Create an RGB marquee

- Objective: Control the RGB LED to illuminate repeatedly in the sequence of blue, green, red, blue-green, blue-red, green-red, white.
- Hint:

1. PA0、PA1、PC2 attach to B、G、R
2. Draw the wiring diagram and explain what values should be given to the RGBs for them to be bright.



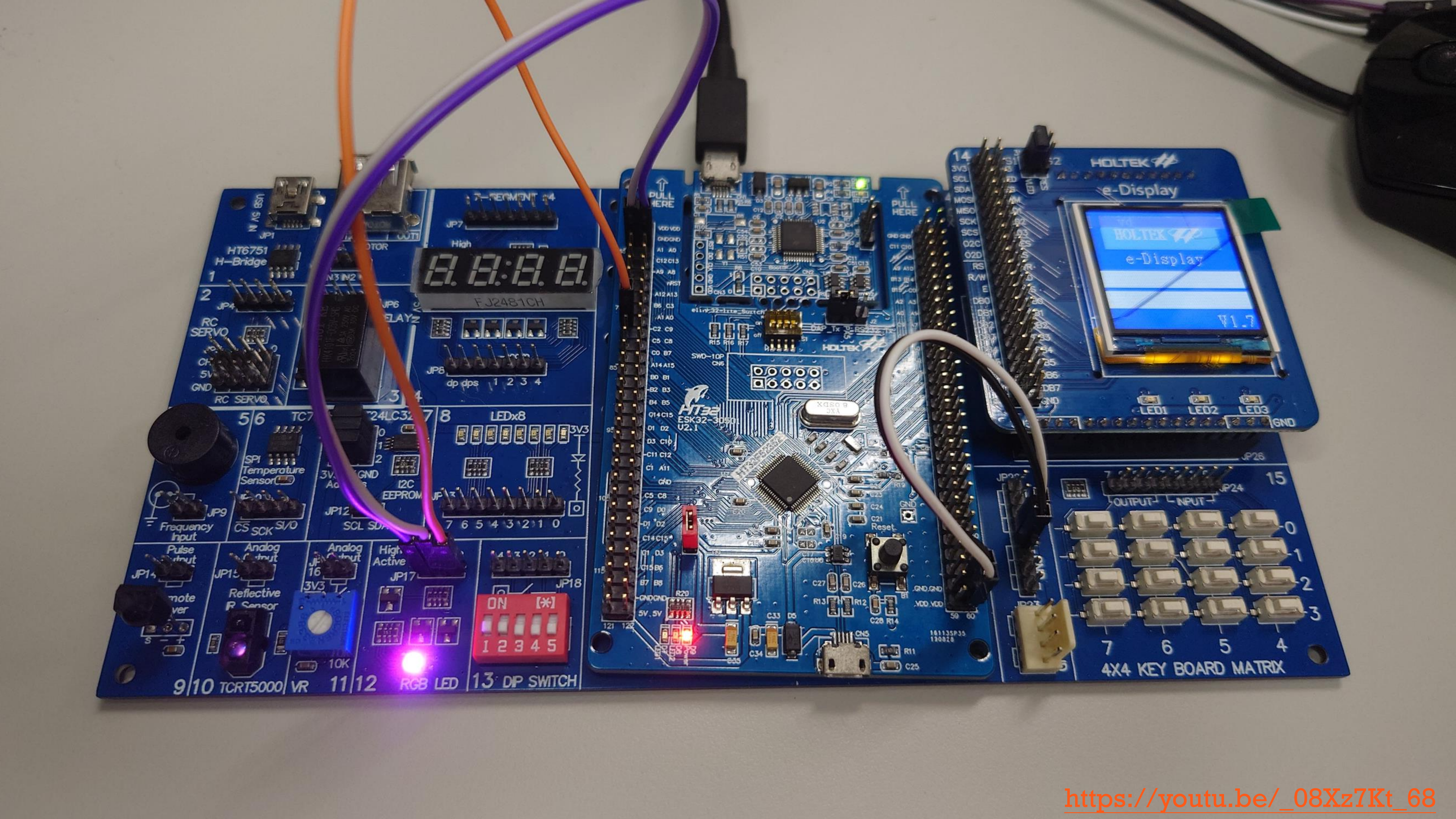
Input

Output

a set of power supplies

Vcc

GND



https://youtu.be/_08Xz7Kt_68

Class Dismissed