Micro-Controller Experiment

Week4

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

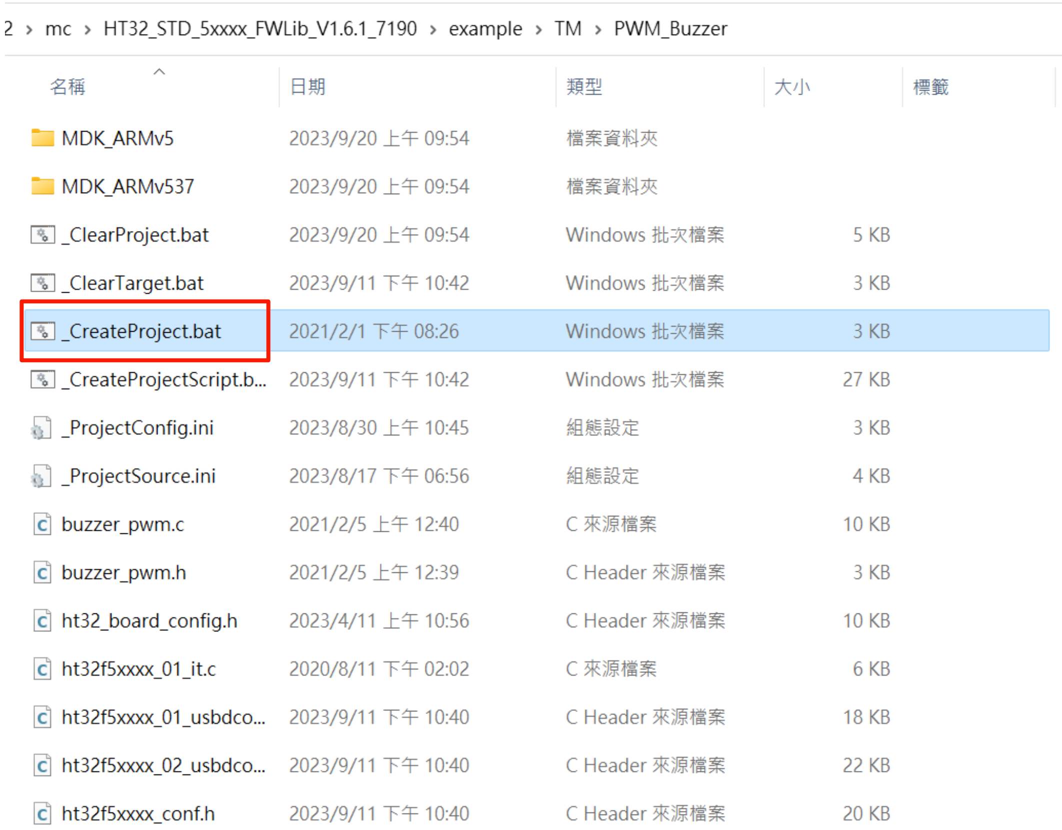
1. Timer Project.
2. Homework 4-1.
3. Homework 4-2.
4. Homework 4-3.

Timer Project

1. Execute “_CreatProject”

1. Go to “~/HT32_STD_5xxxx_FWLib_V1.5.1_7084/example/TM/PWM_Buzzer”.

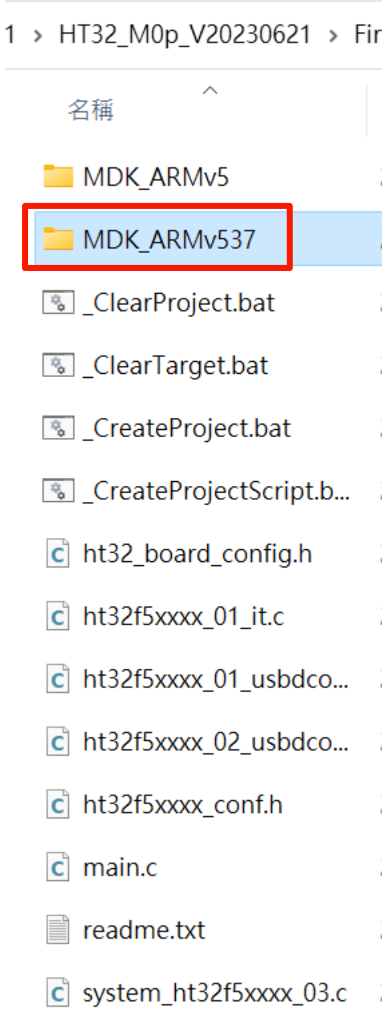
2. Double click “_CreateProject.bat”.



2 › mc › HT32_STD_5xxxx_FWLib_V1.6.1_7190 › example › TM › PWM_Buzzer

名稱	日期	類型	大小	標籤
MDK_ARMv5	2023/9/20 上午 09:54	檔案資料夾		
MDK_ARMv537	2023/9/20 上午 09:54	檔案資料夾		
_ClearProject.bat	2023/9/20 上午 09:54	Windows 批次檔案	5 KB	
_ClearTarget.bat	2023/9/11 下午 10:42	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...	2023/9/11 下午 10:42	Windows 批次檔案	27 KB	
_ProjectConfig.ini	2023/8/30 上午 10:45	組態設定	3 KB	
_ProjectSource.ini	2023/8/17 下午 06:56	組態設定	4 KB	
buzzer_pwm.c	2021/2/5 上午 12:40	C 來源檔案	10 KB	
buzzer_pwm.h	2021/2/5 上午 12:39	C Header 來源檔案	3 KB	
ht32_board_config.h	2023/4/11 上午 10:56	C Header 來源檔案	10 KB	
ht32f5xxxx_01_it.c	2020/8/11 下午 02:02	C 來源檔案	6 KB	
ht32f5xxxx_01_usbdco...	2023/9/11 下午 10:40	C Header 來源檔案	18 KB	
ht32f5xxxx_02_usbdco...	2023/9/11 下午 10:40	C Header 來源檔案	22 KB	
ht32f5xxxx_conf.h	2023/9/11 下午 10:40	C Header 來源檔案	20 KB	

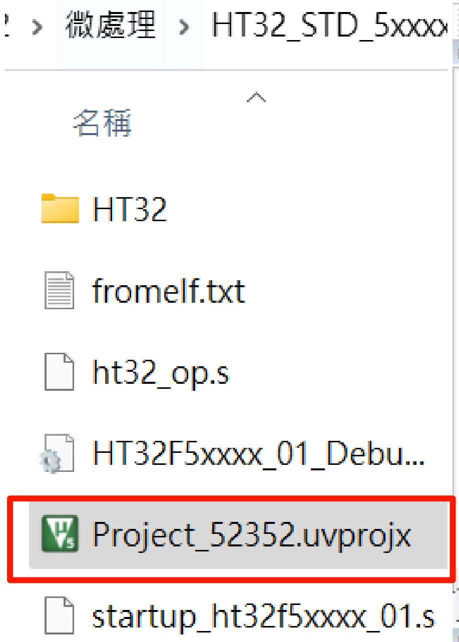
2. Launch project



1 › HT32_M0p_V20230621 › Fir

名稱

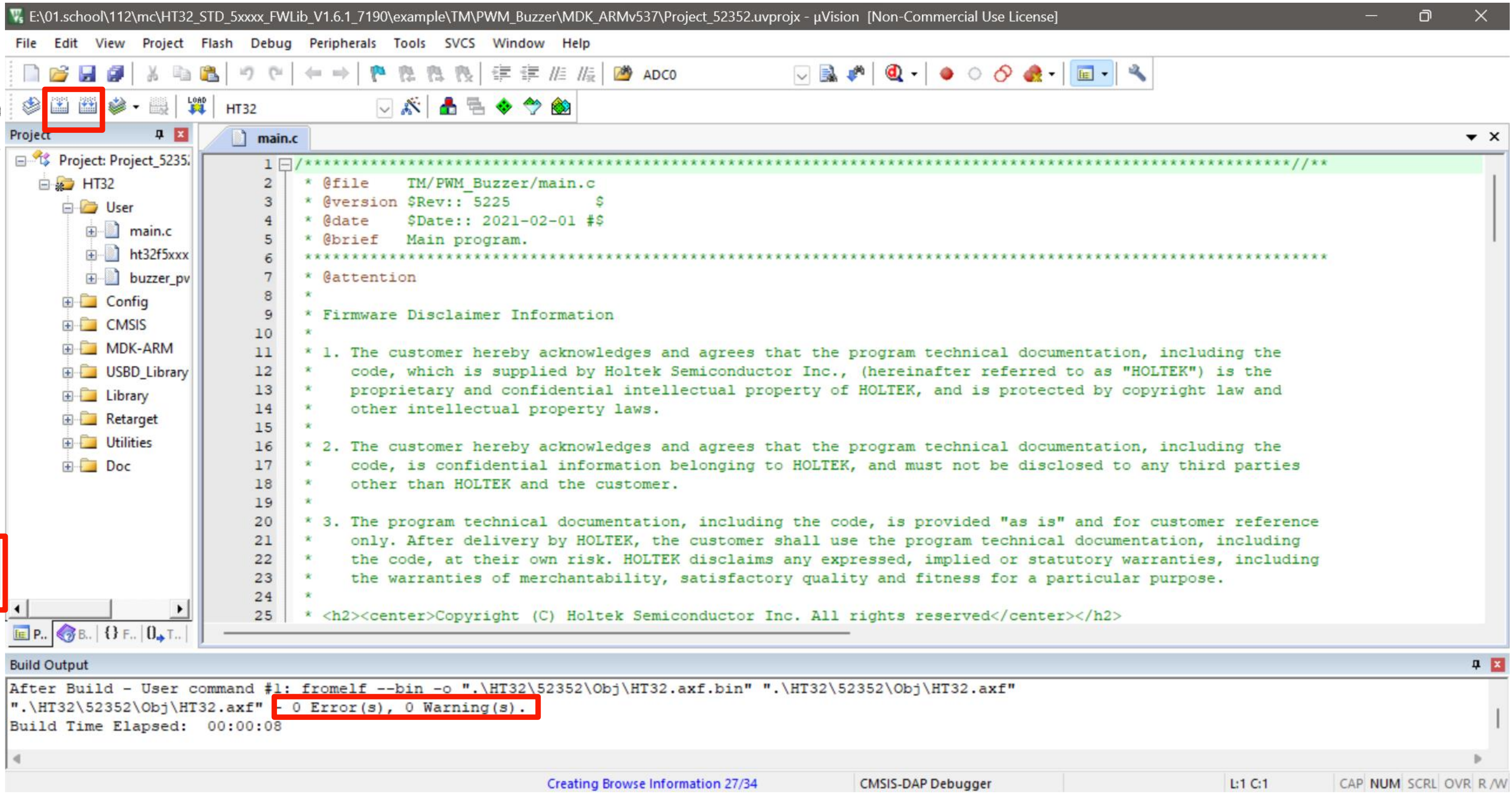
- MDK_ARMv5
- MDK_ARMv537
- _ClearProject.bat
- _ClearTarget.bat
- _CreateProject.bat
- _CreateProjectScript.b...
- ht32_board_config.h
- ht32f5xxxx_01_it.c
- ht32f5xxxx_01_usbdco...
- ht32f5xxxx_02_usbdco...
- ht32f5xxxx_conf.h
- main.c
- readme.txt
- system_ht32f5xxxx_03.c



› 微處理 › HT32_STD_5xxx

名稱

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



E:\01.school\112\mc\HT32_STD_5xxxx_FWLib_V1.6.1_7190\example\TM\PWM_Buzzer\MDK_ARMv537\Project_52352.uvprojx - µVision [Non-Commercial Use License]

```
/*********************************************************************************************//**
 * @file    TM/PWM_Buzzer/main.c
 * @version $Rev:: 5225         $
 * @date    $Date:: 2021-02-01 #$
 * @brief   Main program.
 *************************************************************************************************
 * @attention
 *
 * Firmware Disclaimer Information
 *
 * 1. The customer hereby acknowledges and agrees that the program technical documentation, including the
 *    code, which is supplied by Holtek Semiconductor Inc., (hereinafter referred to as "HOLTEK") is the
 *    proprietary and confidential intellectual property of HOLTEK, and is protected by copyright law and
 *    other intellectual property laws.
 *
 * 2. The customer hereby acknowledges and agrees that the program technical documentation, including the
 *    code, is confidential information belonging to HOLTEK, and must not be disclosed to any third parties
 *    other than HOLTEK and the customer.
 *
 * 3. The program technical documentation, including the code, is provided "as is" and for customer reference
 *    only. After delivery by HOLTEK, the customer shall use the program technical documentation, including
 *    the code, at their own risk. HOLTEK disclaims any expressed, implied or statutory warranties, including
 *    the warranties of merchantability, satisfactory quality and fitness for a particular purpose.
 *
 * <h2><center>Copyright (C) Holtek Semiconductor Inc. All rights reserved</center></h2>
```

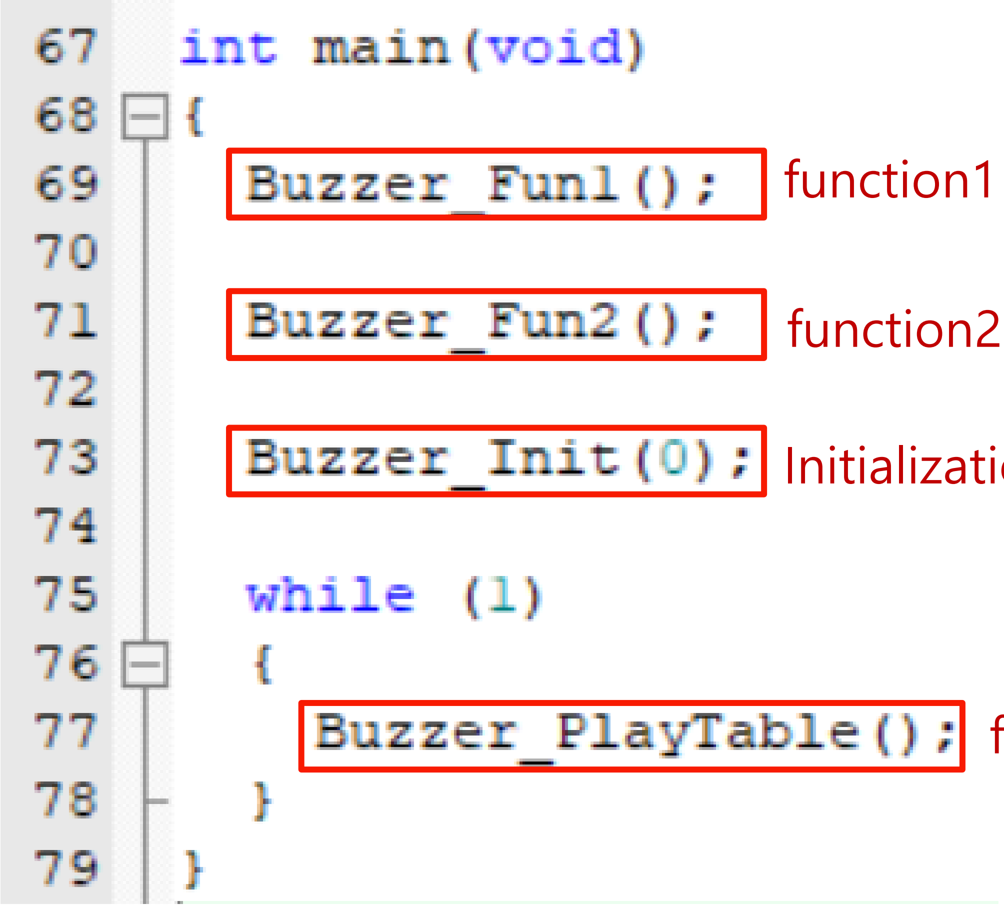
Build Output

```
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
```

Creating Browse Information 27/34

CMSIS-DAP Debugger

main



```
int main(void)
{
    Buzzer_Fun1();

    Buzzer_Fun2();

    Buzzer_Init(0);

    while (1)
     {
        Buzzer_PlayTable();
     }
}
```

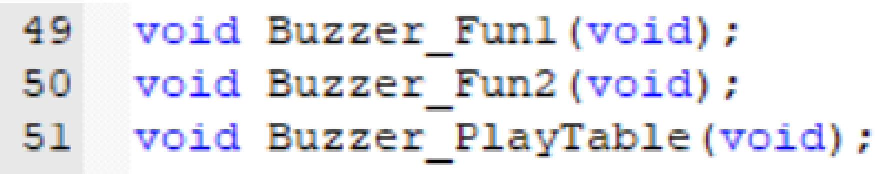
function1

function2

Initialization Setting

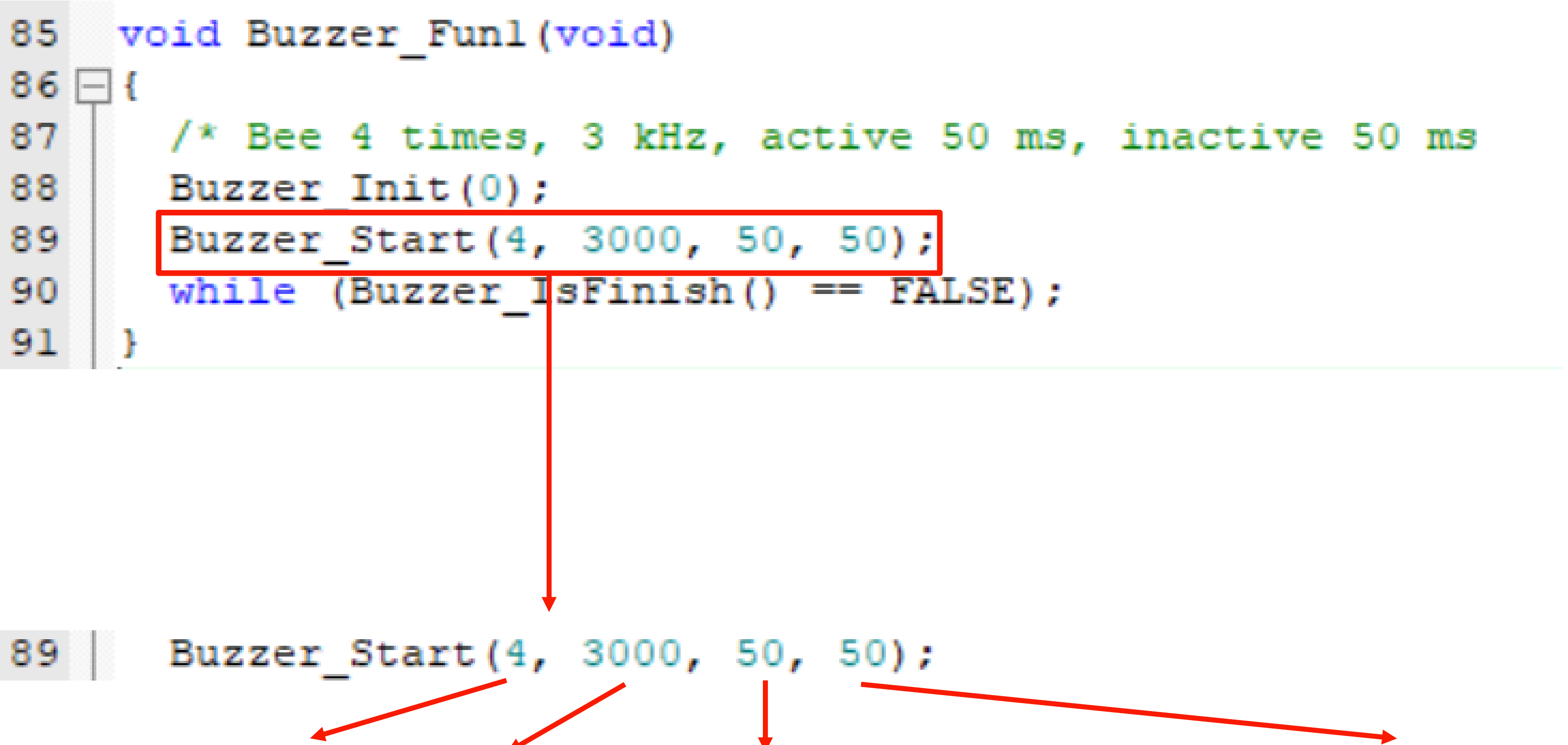
function3

Declare functions in the beginning



```
void Buzzer_Fun1(void);
void Buzzer_Fun2(void);
void Buzzer_PlayTable(void);
```

funciton1

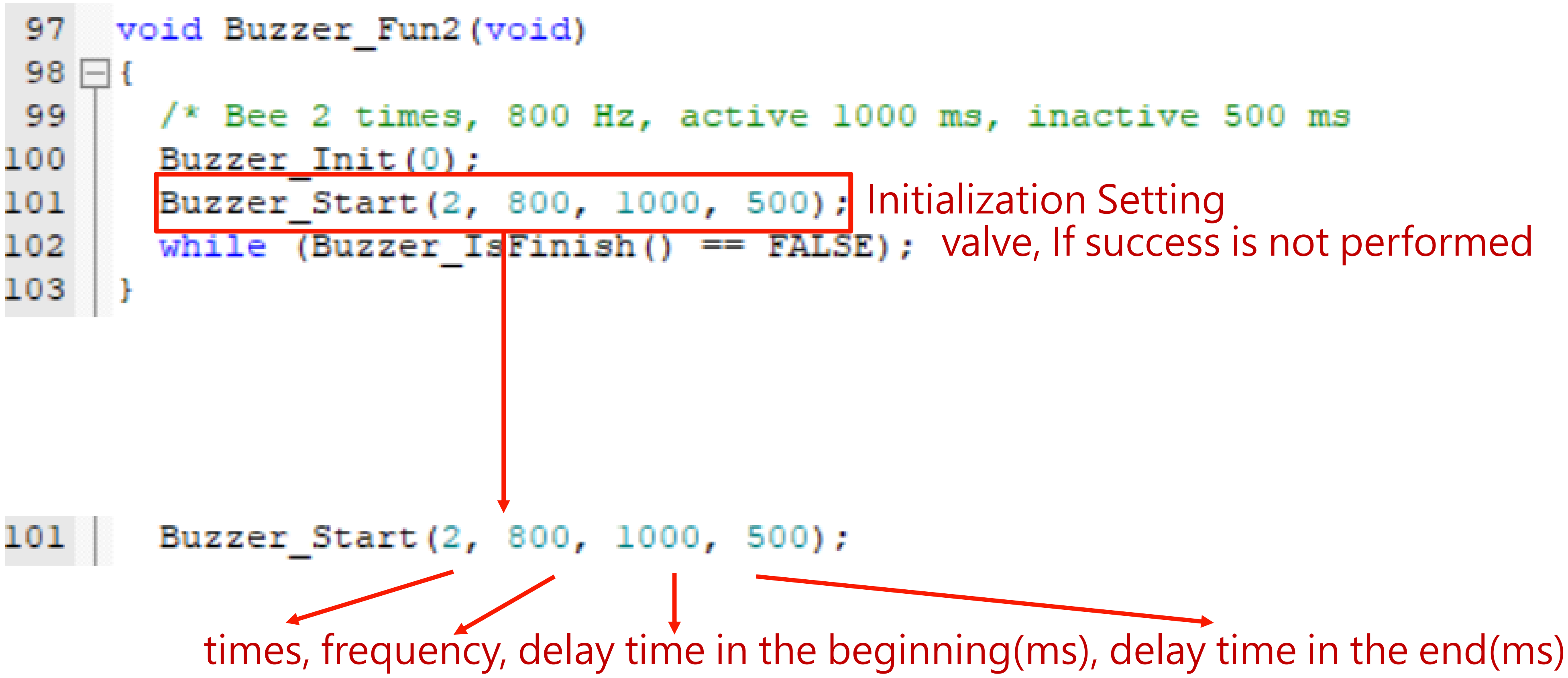


```
void Buzzer_Fun1(void)
{
    /* Bee 4 times, 3 kHz, active 50 ms, inactive 50 ms
    Buzzer_Init(0);
    Buzzer_Start(4, 3000, 50, 50);
    while (Buzzer_IsFinish() == FALSE);
}
```

```
    Buzzer_Start(4, 3000, 50, 50);
```

times, frequency, delay time in the beginning(ms), delay time in the end(ms)

funciton2



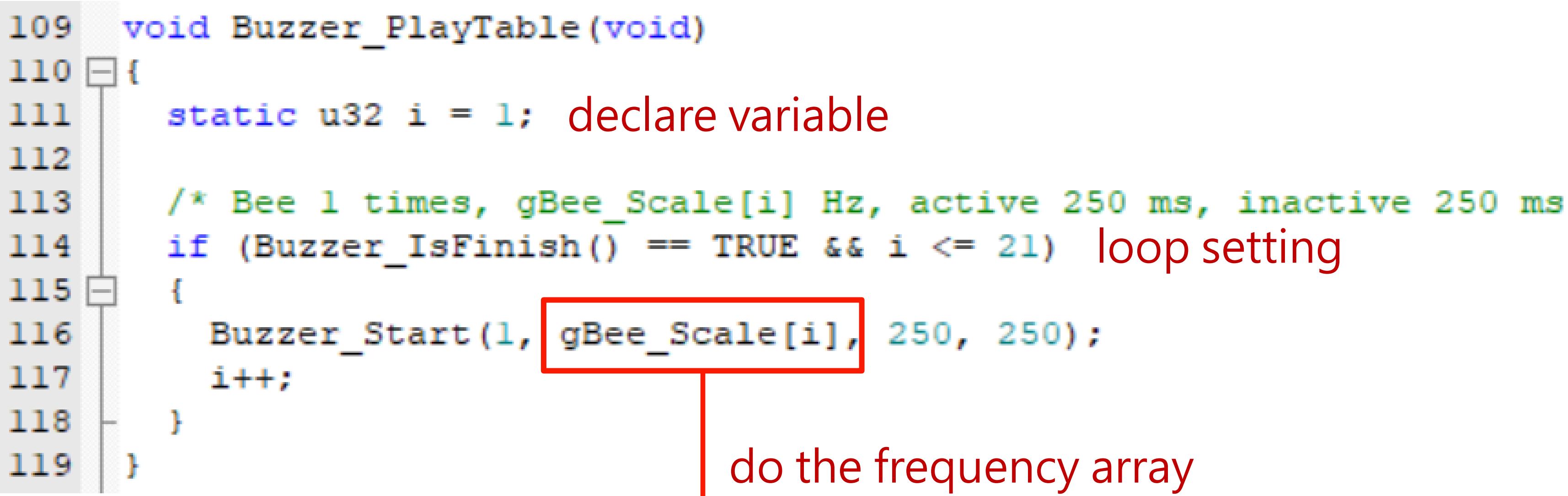
```
void Buzzer_Fun2(void)
{
    /* Bee 2 times, 800 Hz, active 1000 ms, inactive 500 ms
    Buzzer_Init(0);
    Buzzer_Start(2, 800, 1000, 500);
    while (Buzzer_IsFinish() == FALSE);
}
```

Initialization Setting valve, If success is not performed

```
    Buzzer_Start(2, 800, 1000, 500);
```

times, frequency, delay time in the beginning(ms), delay time in the end(ms)

function3

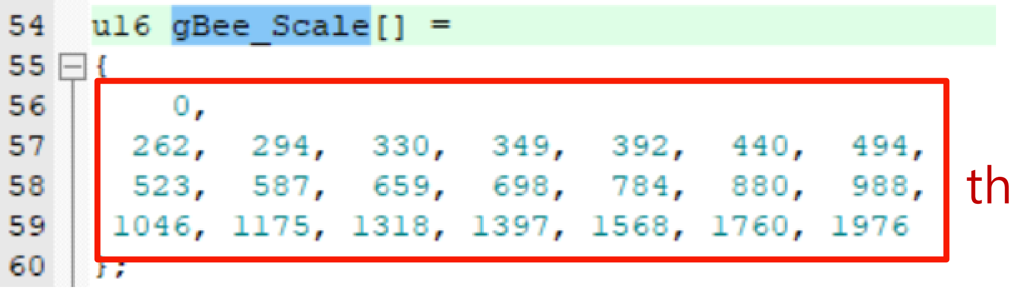


```
void Buzzer_PlayTable(void)
{
  static u32 i = 1;

  /* Bee 1 times, gBee_Scale[i] Hz, active 250 ms, inactive 250 ms
  if (Buzzer_IsFinish() == TRUE && i <= 21)
  {
    Buzzer_Start(1, gBee_Scale[i], 250, 250);
    i++;
  }
}
```

declare variable

loop setting

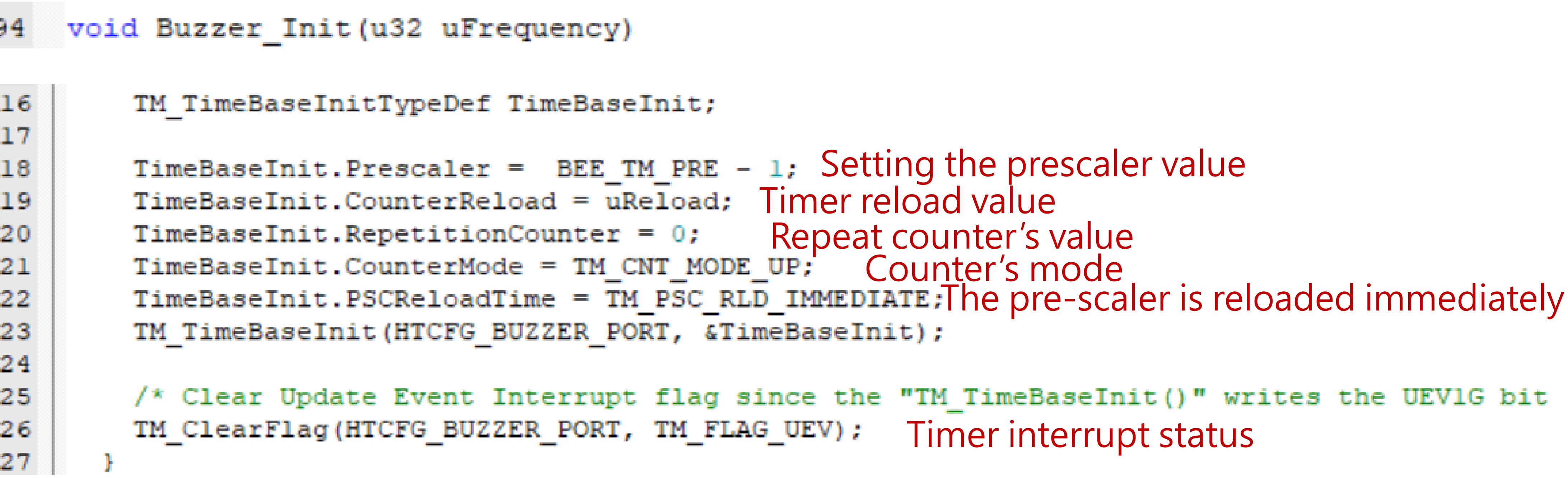
do the frequency array



```
u16 gBee_Scale[] =
{
    0,
  262,  294,  330,  349,  392,  440,  494,
  523,  587,  659,  698,  784,  880,  988,
 1046, 1175, 1318, 1397, 1568, 1760, 1976
};
```

the frequency array

Initialization Setting



```
void Buzzer_Init(u32 uFrequency)

    TM_TimeBaseInitTypeDef TimeBaseInit;

    TimeBaseInit.Prescaler =  BEE_TM_PRE - 1;
    TimeBaseInit.CounterReload = uReload;
    TimeBaseInit.RepetitionCounter = 0;
    TimeBaseInit.CounterMode = TM_CNT_MODE_UP;
    TimeBaseInit.PSCReloadTime = TM_PSC_RLD_IMMEDIATE;
    TM_TimeBaseInit(HTCFG_BUZZER_PORT, &TimeBaseInit);

    /* Clear Update Event Interrupt flag since the "TM_TimeBaseInit()" writes the UEV1G bit
    TM_ClearFlag(HTCFG_BUZZER_PORT, TM_FLAG_UEV);
  }
```

Setting the prescaler value

Timer reload value

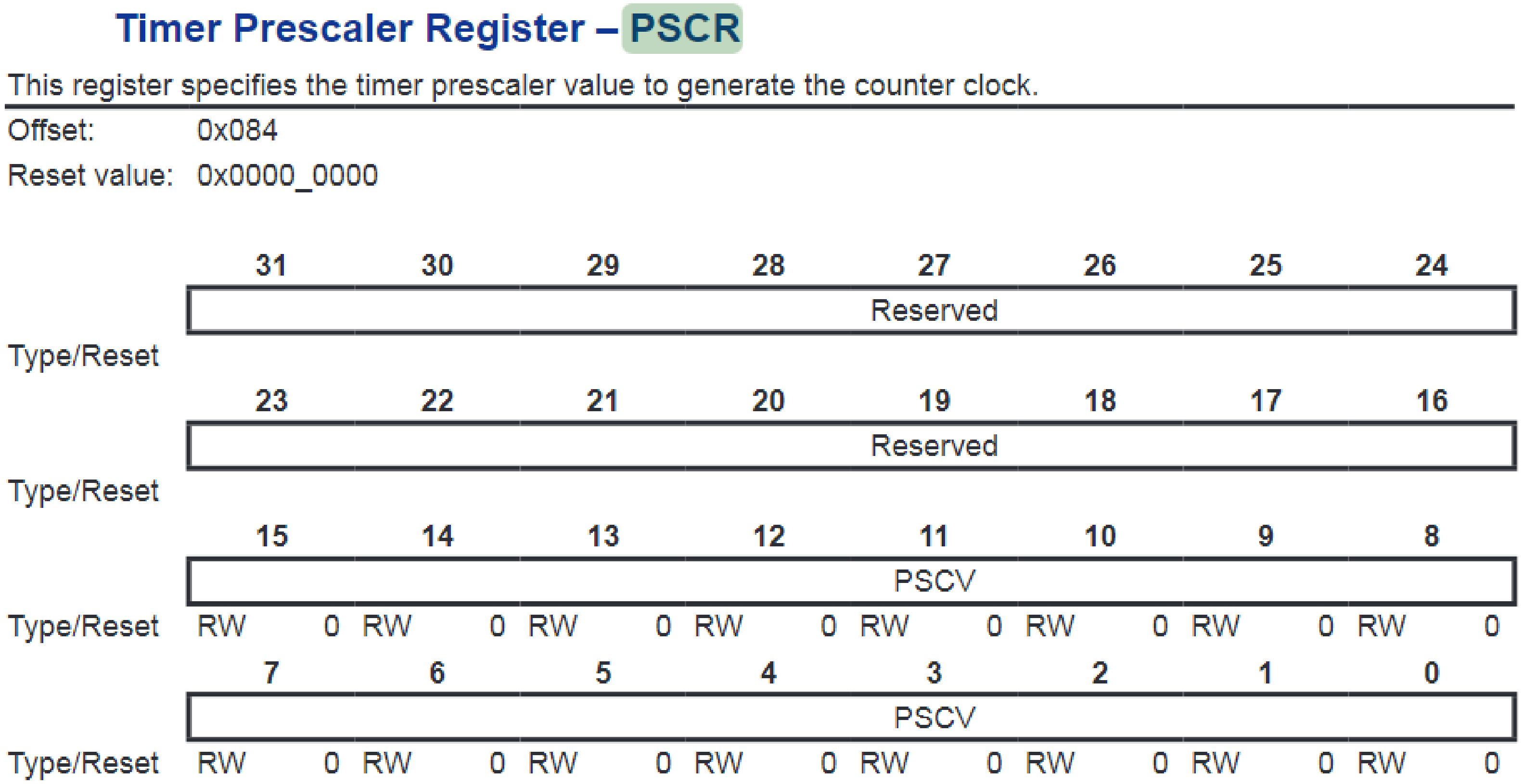
Repeat counter’s value

Counter’s mode

The pre-scaler is reloaded immediately after the trigger

Timer interrupt status

Timer Prescaler Resgister



Timer Prescaler Register – PSCR

This register specifies the timer prescaler value to generate the counter clock.

Offset: 0x084

Reset value: 0x0000_0000

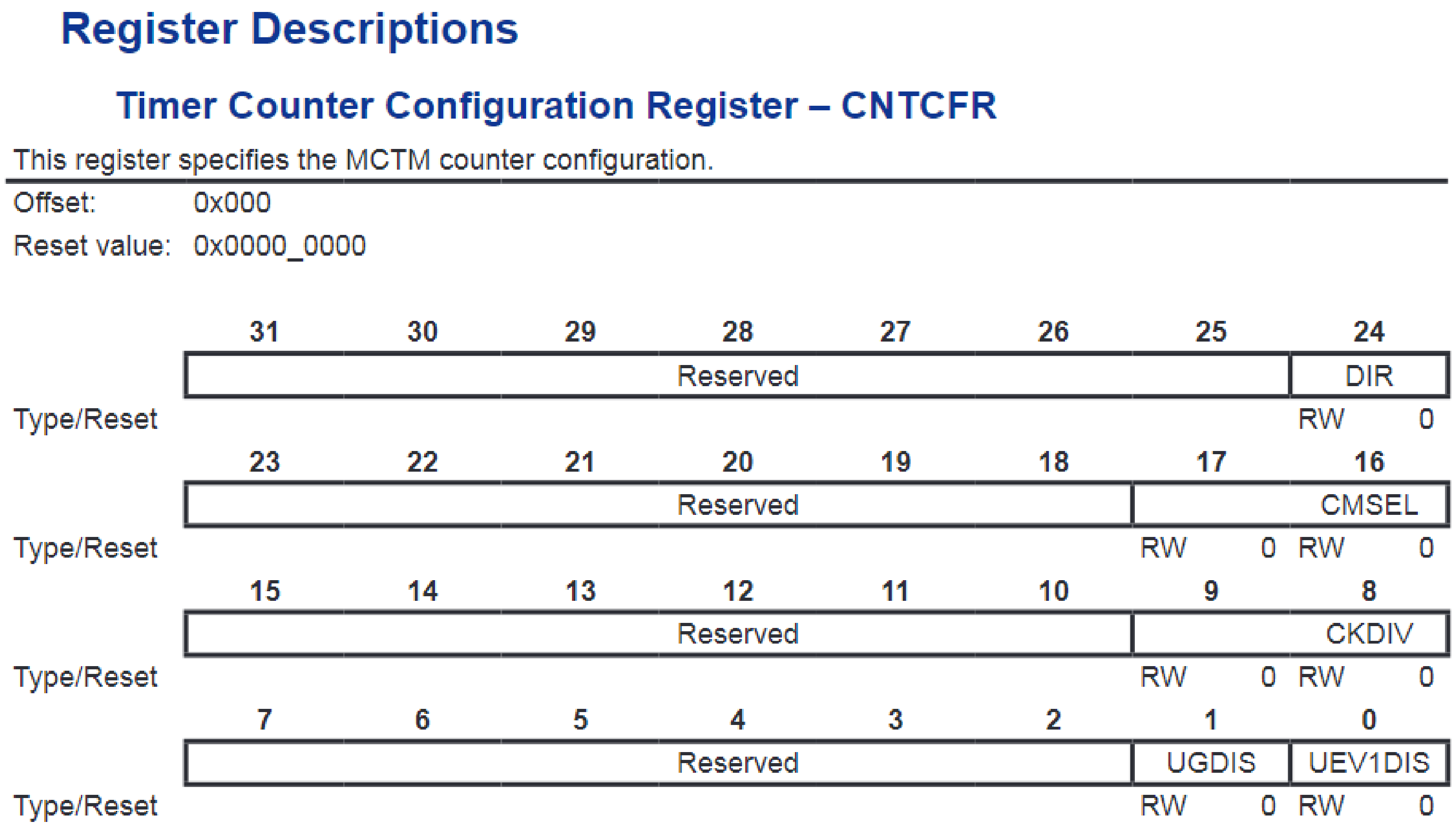
	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								

	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								

	15	14	13	12	11	10	9	8
	PSCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

	7	6	5	4	3	2	1	0
	PSCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Timer Counter Configuration Register



Register Descriptions

Timer Counter Configuration Register – CNTCFR

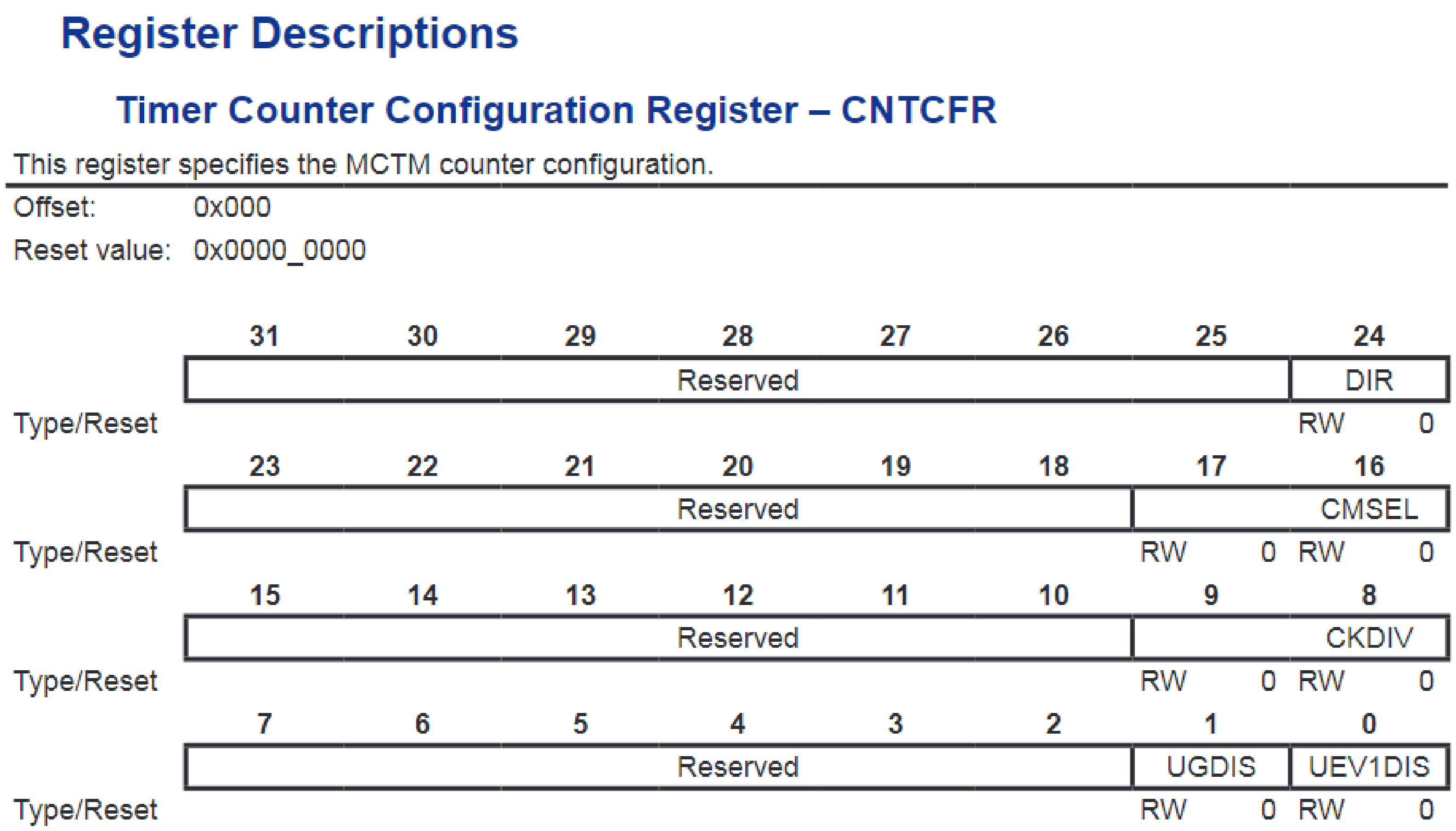
This register specifies the MCTM counter configuration.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							DIR
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved						CMSEL	
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved						CKDIV	
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved						UGDIS	UEV1DIS
Type/Reset							RW 0	RW 0

Timer Counter Configuration Register



Register Descriptions

Timer Counter Configuration Register – CNTCFR

This register specifies the MCTM counter configuration.

Offset: 0x000

Reset value: 0x0000_0000

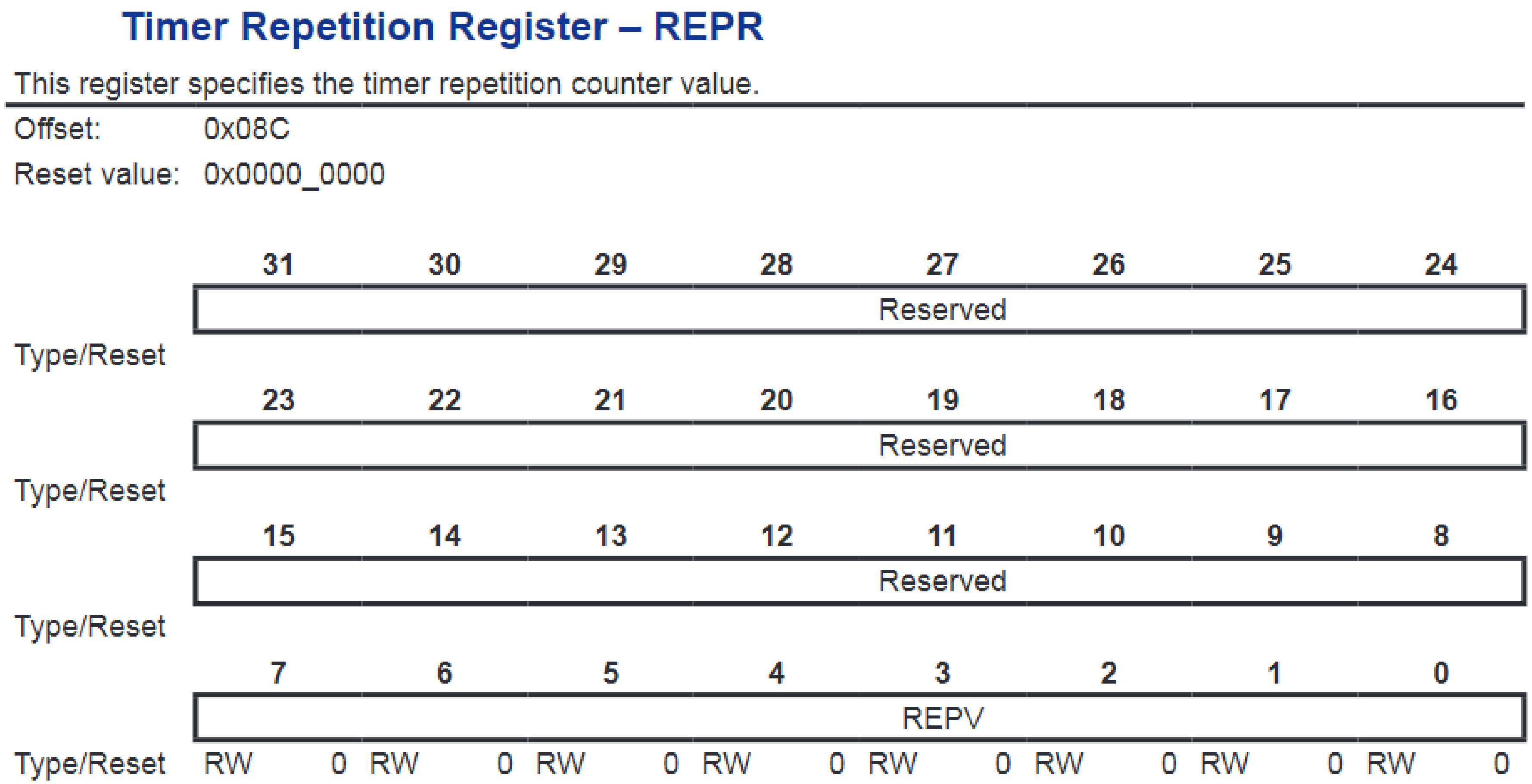
	31	30	29	28	27	26	25	24
	Reserved							DIR
Type/Reset								RW 0

	23	22	21	20	19	18	17	16
	Reserved						CMSEL	
Type/Reset							RW 0	RW 0

	15	14	13	12	11	10	9	8
	Reserved						CKDIV	
Type/Reset							RW 0	RW 0

	7	6	5	4	3	2	1	0
	Reserved						UGDIS	UEV1DIS
Type/Reset							RW 0	RW 0

Timer Repetition Register



Timer Repetition Register – REPR

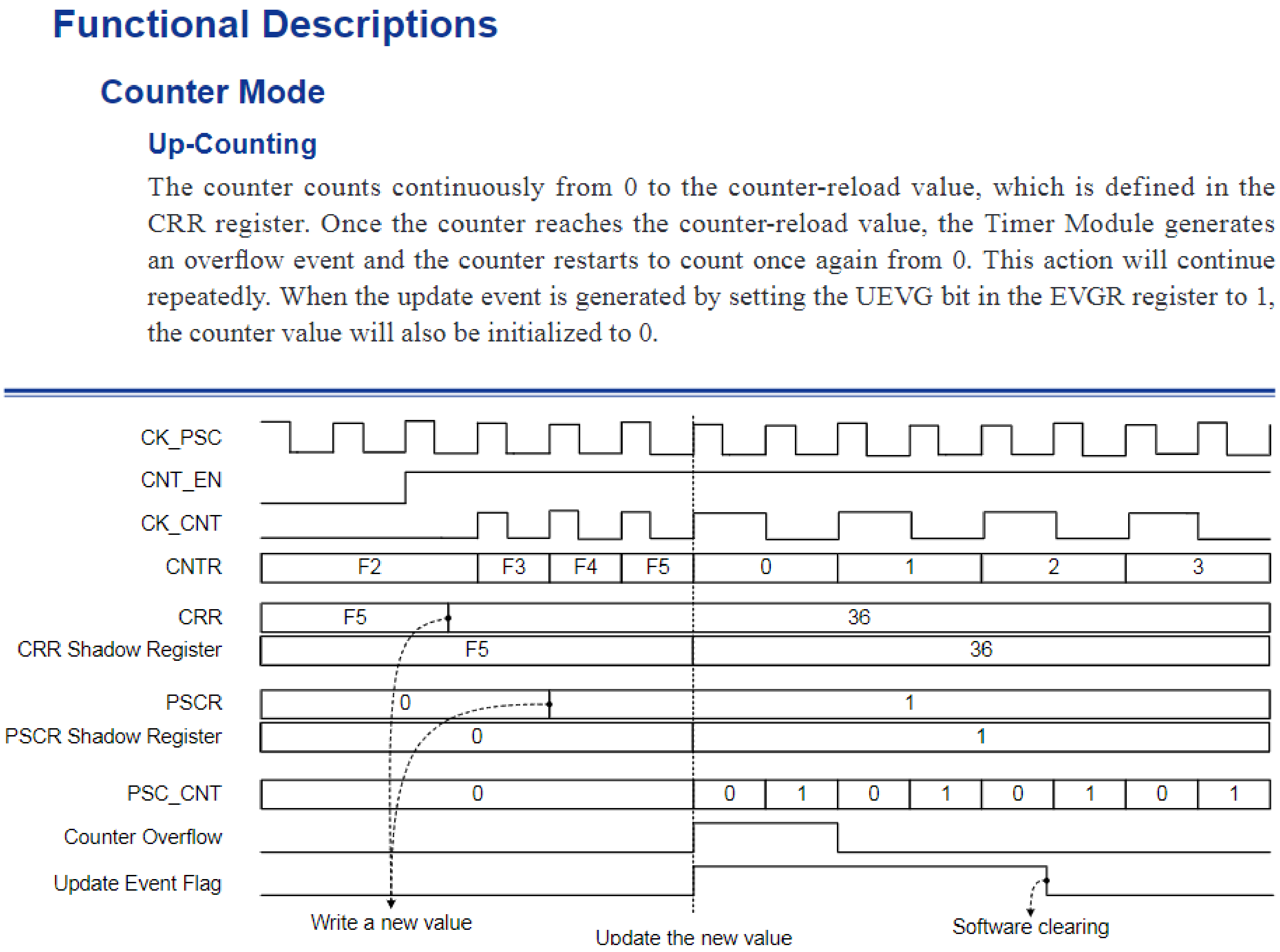
This register specifies the timer repetition counter value.

Offset: 0x08C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	REPV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Counter mode



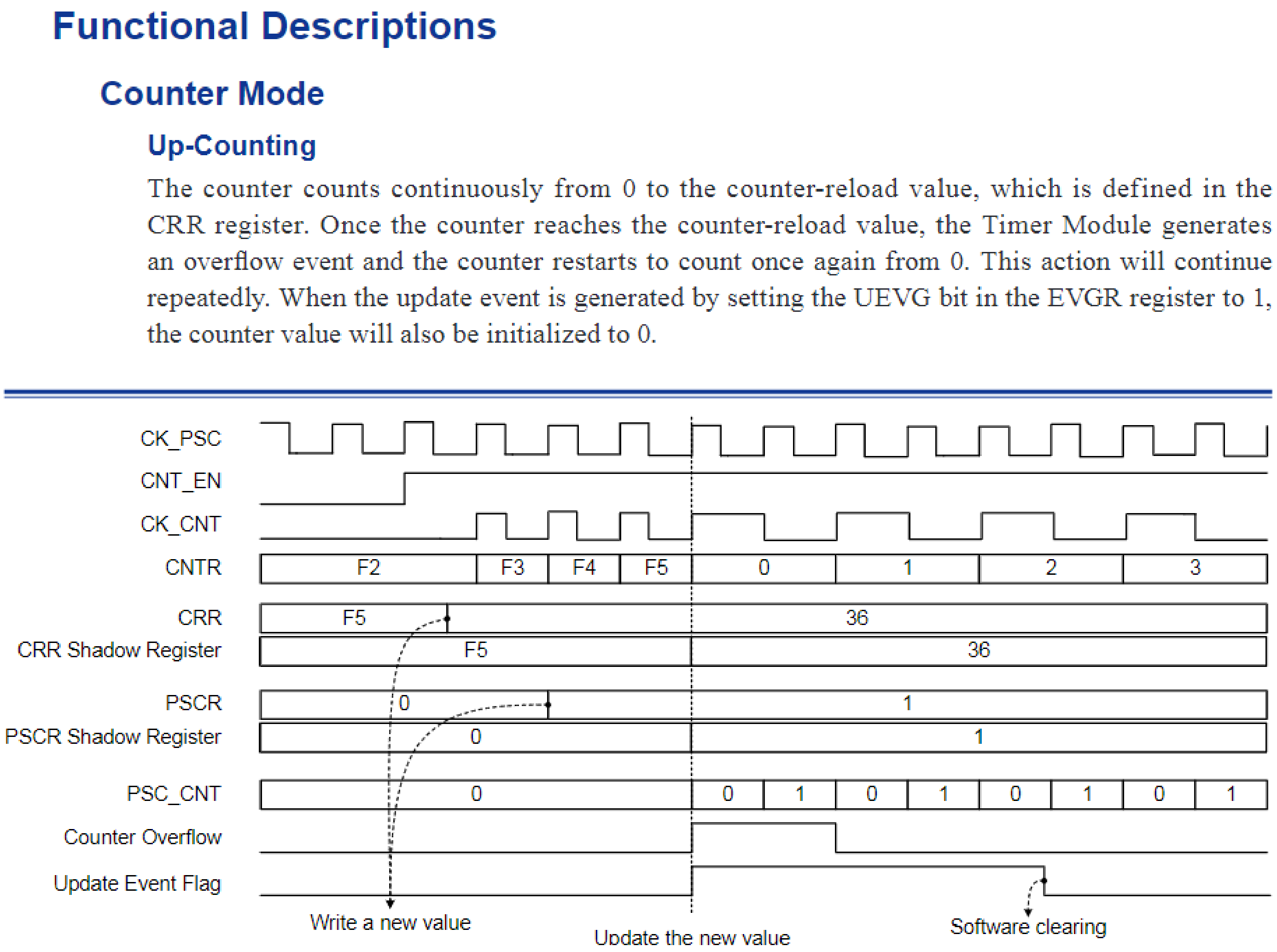
Functional Descriptions

Counter Mode

Up-Counting

The counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register. Once the counter reaches the counter-reload value, the Timer Module generates an overflow event and the counter restarts to count once again from 0. This action will continue repeatedly. When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.

Counter mode



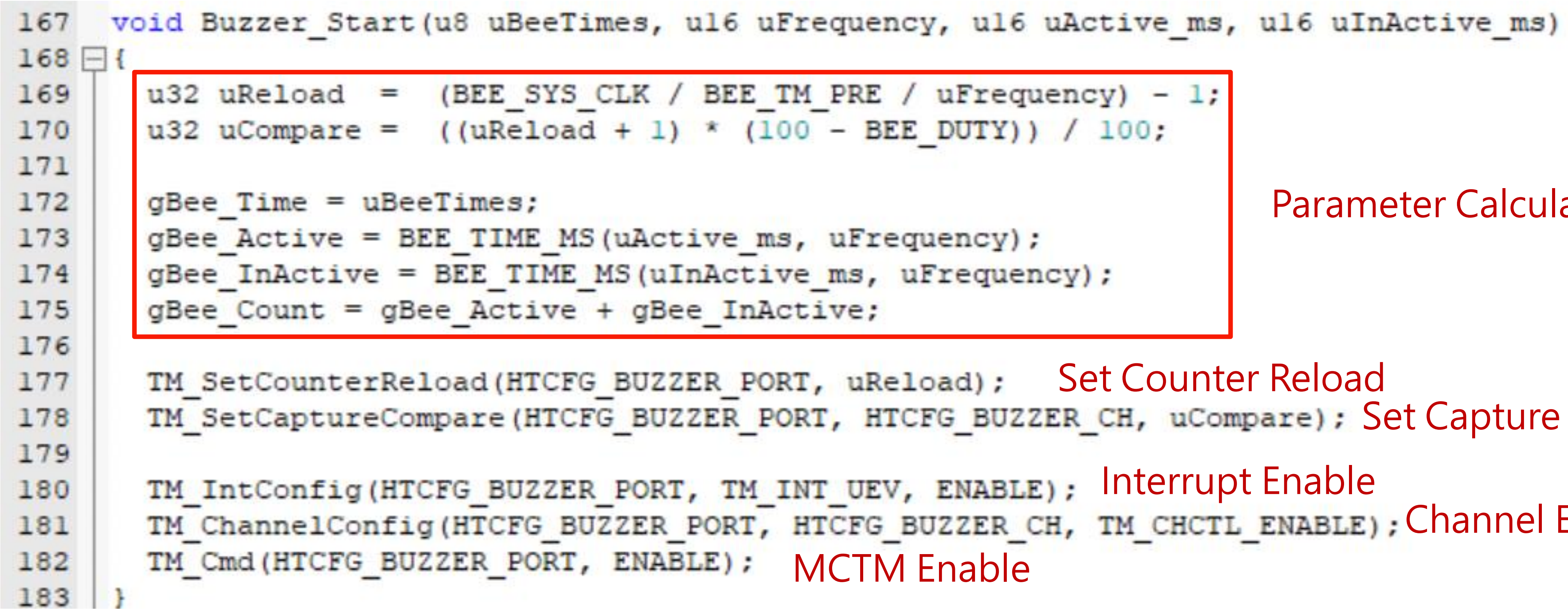
Functional Descriptions

Counter Mode

Up-Counting

The counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register. Once the counter reaches the counter-reload value, the Timer Module generates an overflow event and the counter restarts to count once again from 0. This action will continue repeatedly. When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.

Buzzer_Start



```c
void Buzzer_Start(u8 uBeeTimes, u16 uFrequency, u16 uActive_ms, u16 uInActive_ms)
{
  u32 uReload  =  (BEE_SYS_CLK / BEE_TM_PRE / uFrequency) - 1;
  u32 uCompare =  ((uReload + 1) * (100 - BEE_DUTY)) / 100;

  gBee_Time = uBeeTimes;
  gBee_Active = BEE_TIME_MS(uActive_ms, uFrequency);
  gBee_InActive = BEE_TIME_MS(uInActive_ms, uFrequency);
  gBee_Count = gBee_Active + gBee_InActive;

  TM_SetCounterReload(HTCFG_BUZZER_PORT, uReload);
  TM_SetCaptureCompare(HTCFG_BUZZER_PORT, HTCFG_BUZZER_CH, uCompare);

  TM_IntConfig(HTCFG_BUZZER_PORT, TM_INT_UEV, ENABLE);
  TM_ChannelConfig(HTCFG_BUZZER_PORT, HTCFG_BUZZER_CH, TM_CHCTL_ENABLE);
  TM_Cmd(HTCFG_BUZZER_PORT, ENABLE);
}
```

Parameter Calculation

Set Counter Reload

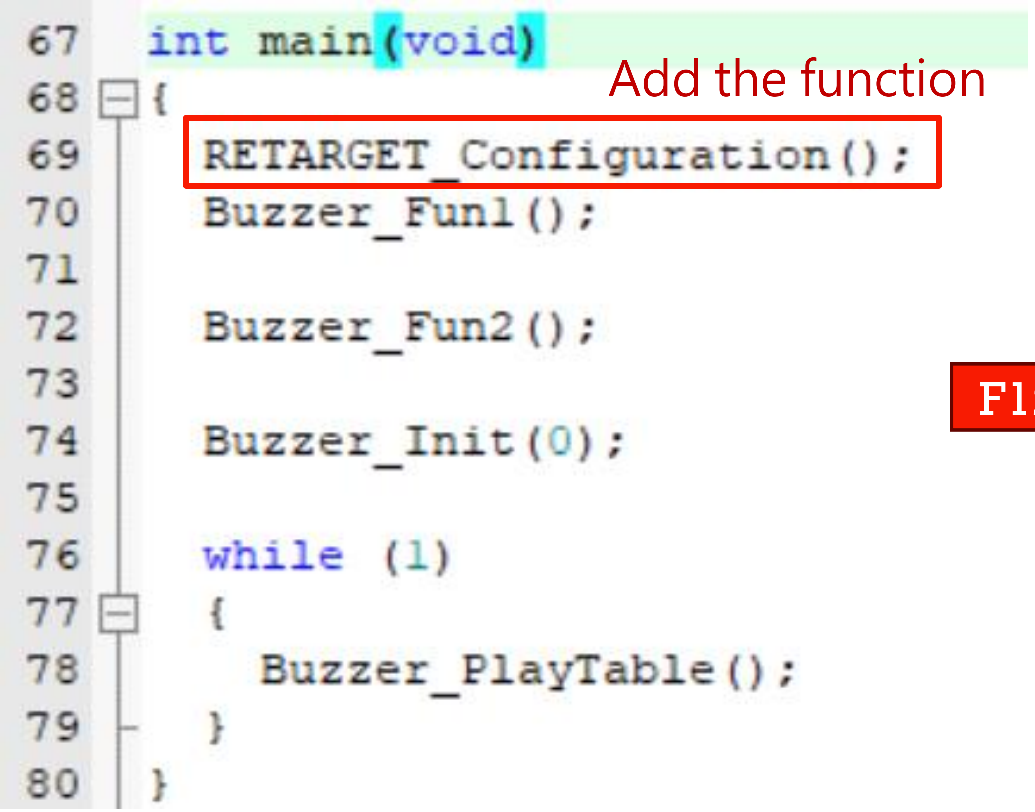
Set Capture Compare

Interrupt Enable

Channel Enable

MCTM Enable

Add USART to check value

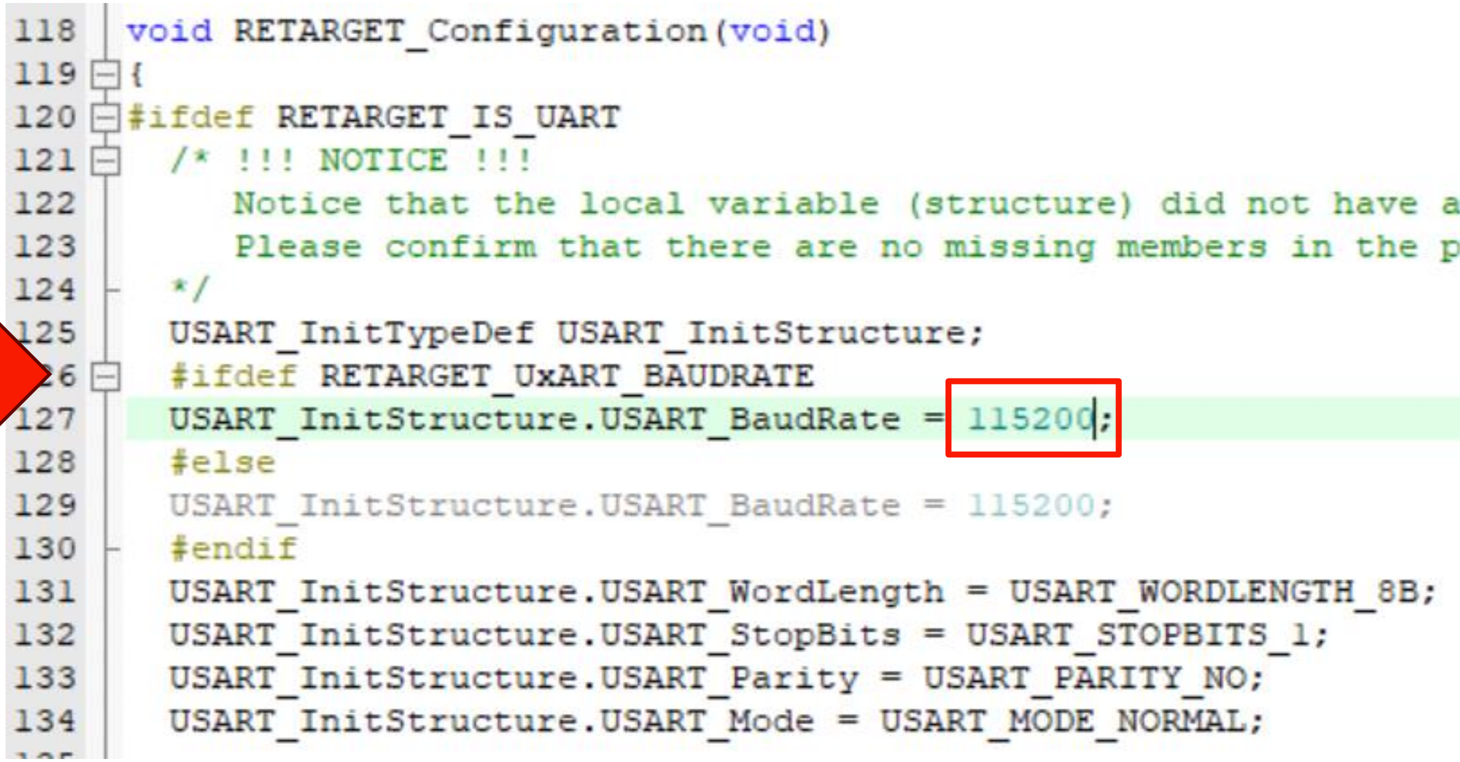


Add the function

```c
int main(void)
{
  RETARGET_Configuration();
  Buzzer_Fun1();

  Buzzer_Fun2();

  Buzzer_Init(0);

  while (1)
  {
    Buzzer_PlayTable();
  }
}
```

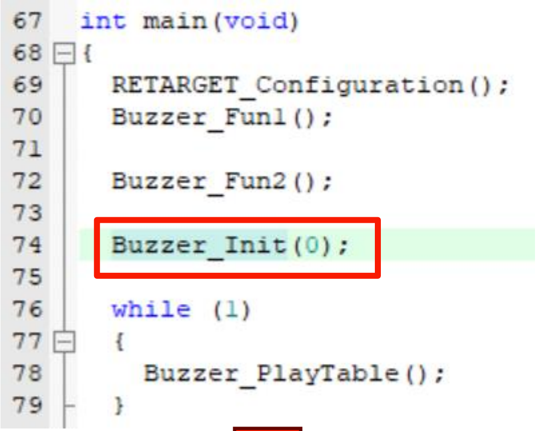
F12

Serial Port Setting



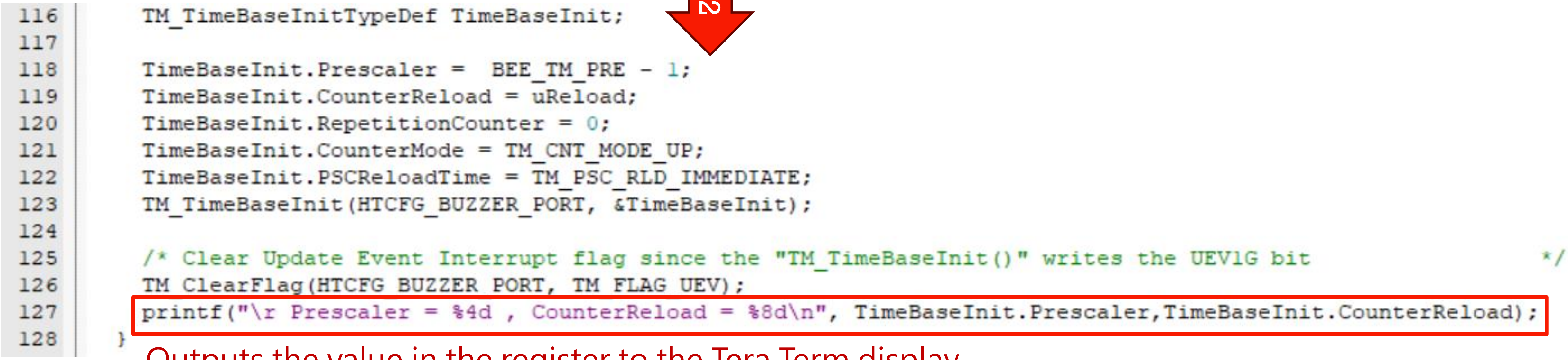
```c
void RETARGET_Configuration(void)
{
#ifdef RETARGET_IS_UART
  /* !!! NOTICE !!!
     Notice that the local variable (structure) did not have a
     Please confirm that there are no missing members in the p
  */
  USART_InitTypeDef USART_InitStructure;
  #ifdef RETARGET_UxART_BAUDRATE
  USART_InitStructure.USART_BaudRate = 115200;
  #else
  USART_InitStructure.USART_BaudRate = 115200;
  #endif
  USART_InitStructure.USART_WordLength = USART_WORDLENGTH_8B;
  USART_InitStructure.USART_StopBits = USART_STOPBITS_1;
  USART_InitStructure.USART_Parity = USART_PARITY_NO;
  USART_InitStructure.USART_Mode = USART_MODE_NORMAL;
```

Edit Buzzer_Init(0)



```
int main(void)
{
  RETARGET_Configuration();
  Buzzer_Fun1();

  Buzzer_Fun2();

  Buzzer_Init(0);

  while (1)
  {
    Buzzer_PlayTable();
  }
```

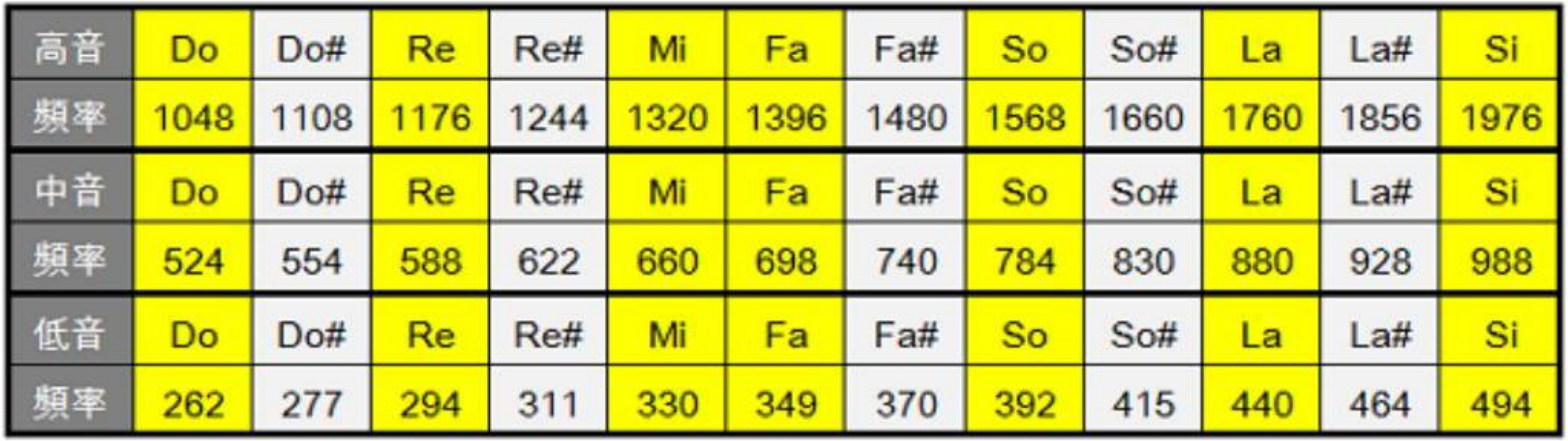
F12



```
    TM_TimeBaseInitTypeDef TimeBaseInit;

    TimeBaseInit.Prescaler =  BEE_TM_PRE - 1;
    TimeBaseInit.CounterReload = uReload;
    TimeBaseInit.RepetitionCounter = 0;
    TimeBaseInit.CounterMode = TM_CNT_MODE_UP;
    TimeBaseInit.PSCReloadTime = TM_PSC_RLD_IMMEDIATE;
    TM_TimeBaseInit(HTCFG_BUZZER_PORT, &TimeBaseInit);

    /* Clear Update Event Interrupt flag since the "TM_TimeBaseInit()" writes the UEV1G bit                    */
    TM_ClearFlag(HTCFG_BUZZER_PORT, TM_FLAG_UEV);
    printf("\r Prescaler = %4d , CounterReload = %8d\n", TimeBaseInit.Prescaler,TimeBaseInit.CounterReload);
}
```

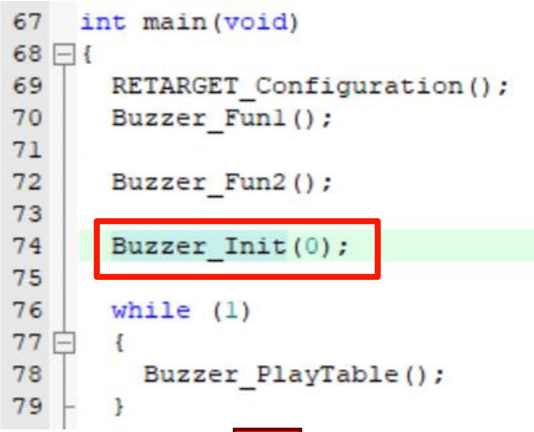
Outputs the value in the register to the Tera Term display.

Scale Frequency Table



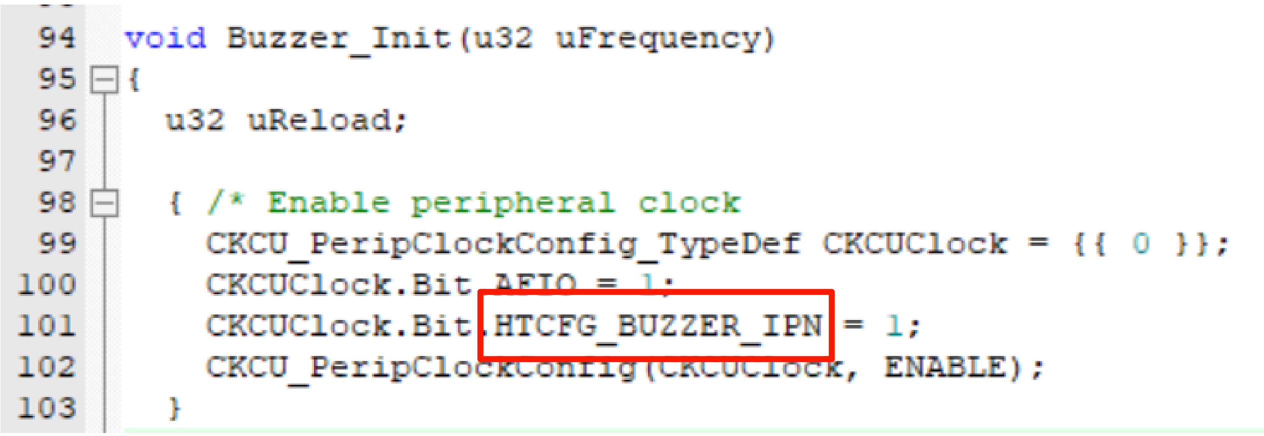
高音	Do	Do#	Re	Re#	Mi	Fa	Fa#	So	So#	La	La#	Si
頻率	1048	1108	1176	1244	1320	1396	1480	1568	1660	1760	1856	1976
中音	Do	Do#	Re	Re#	Mi	Fa	Fa#	So	So#	La	La#	Si
頻率	524	554	588	622	660	698	740	784	830	880	928	988
低音	Do	Do#	Re	Re#	Mi	Fa	Fa#	So	So#	La	La#	Si
頻率	262	277	294	311	330	349	370	392	415	440	464	494

Check the pin



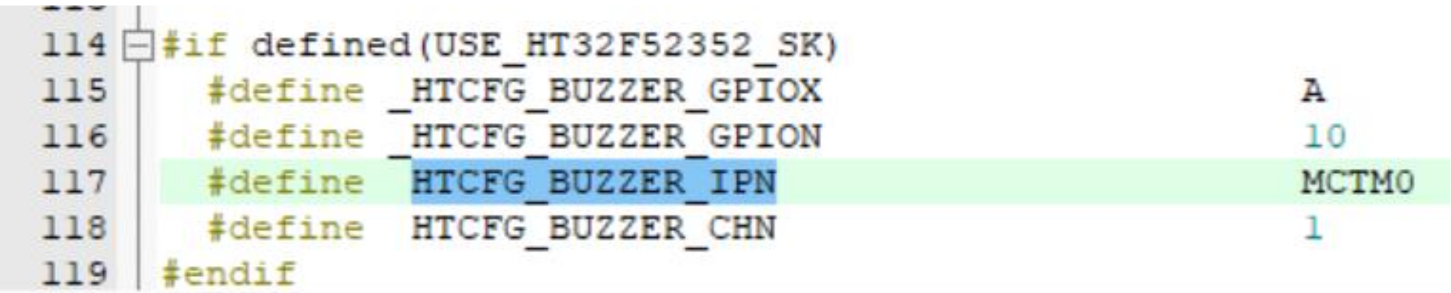
```
int main(void)
{
  RETARGET_Configuration();
  Buzzer_Fun1();

  Buzzer_Fun2();

  Buzzer_Init(0);

  while (1)
  {
    Buzzer_PlayTable();
  }
```

F12



```
void Buzzer_Init(u32 uFrequency)
{
  u32 uReload;

  { /* Enable peripheral clock
    CKCU_PeripClockConfig_TypeDef CKCUClock = {{ 0 }};
    CKCUClock.Bit.AFIO = 1;
    CKCUClock.Bit.HTCFG_BUZZER_IPN = 1;
    CKCU_PeripClockConfig(CKCUClock, ENABLE);
  }
```

F12



```
#if defined(USE_HT32F52352_SK)
  #define _HTCFG_BUZZER_GPIOX                    A
  #define _HTCFG_BUZZER_GPION                    10
  #define  HTCFG_BUZZER_IPN                      MCTM0
  #define  HTCFG_BUZZER_CHN                      1
#endif
```

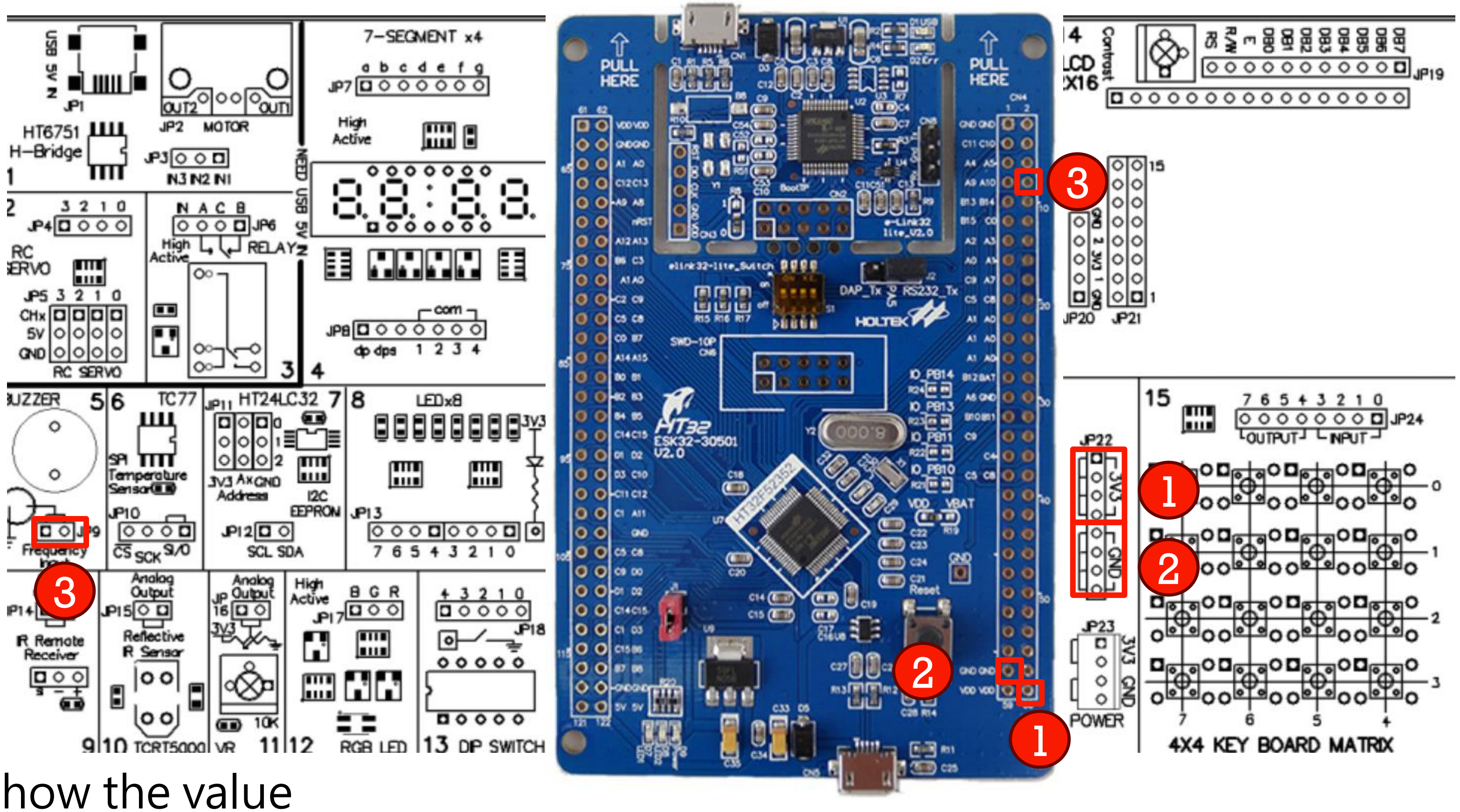
Homework W4-1.

https://github.com/CYCU-AIoT-System-Lab/Microcontroller-Experiment/blob/main/w4/TM-PWM_Buzzer-Experiment_Steps.md

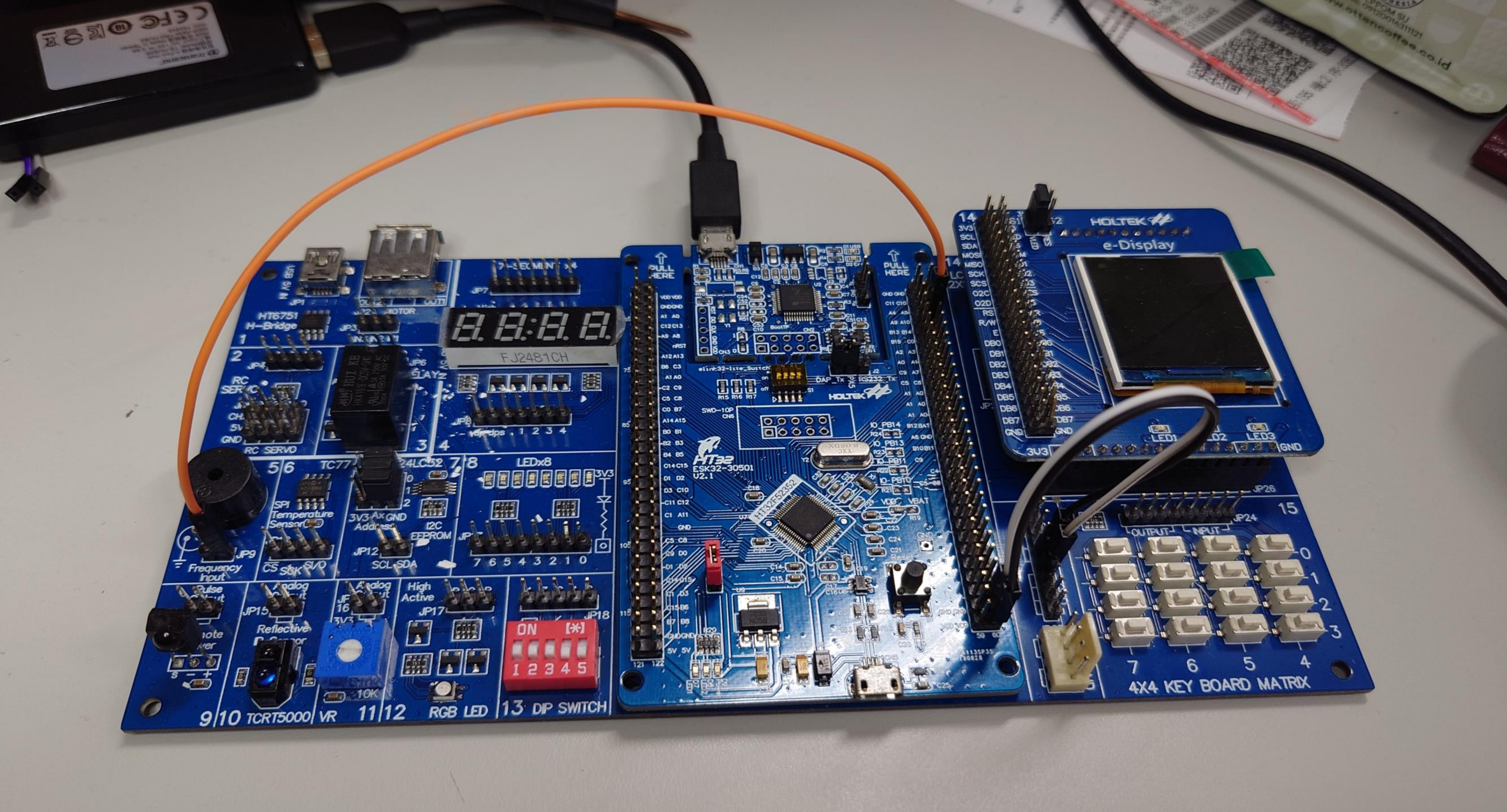
Execute the example

- Objective: Use timer to control buzzer and display in Tera Term.
- Hint:

1. Wiring



2. TERA TERM show the value



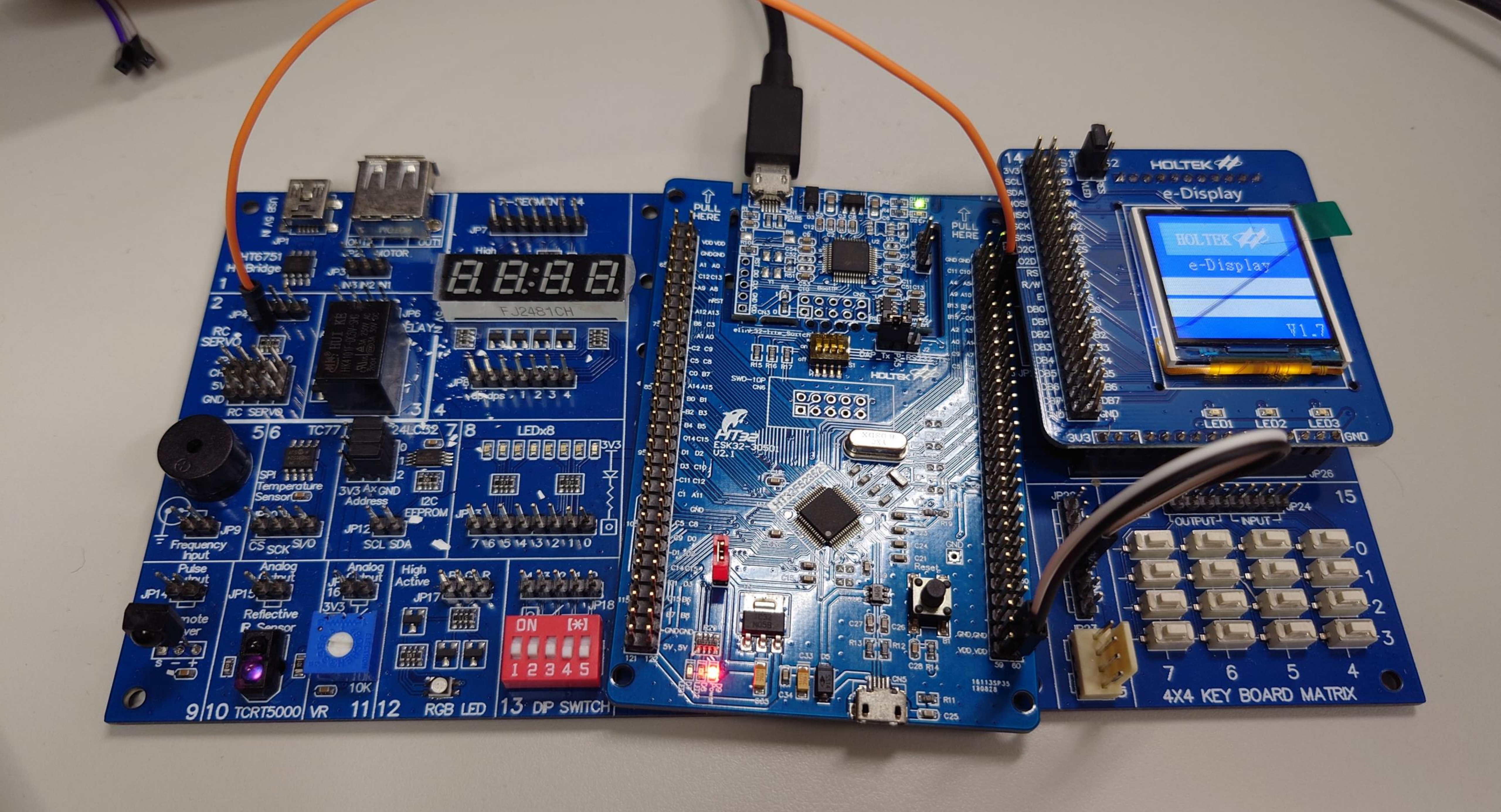
https://youtu.be/jCMfqXhKw2I

Homework W4-2

Play a part of Bee’s music

- Objective: Edit the frequency array to play a part of “Little Bee Song”.
- Hint:

1. Use pin **A10.**
2. Look for numbered musical notation of **"Little Bee Song"**.
3. Find the corresponding scale frequency.
4. Edit gBee_Scale.



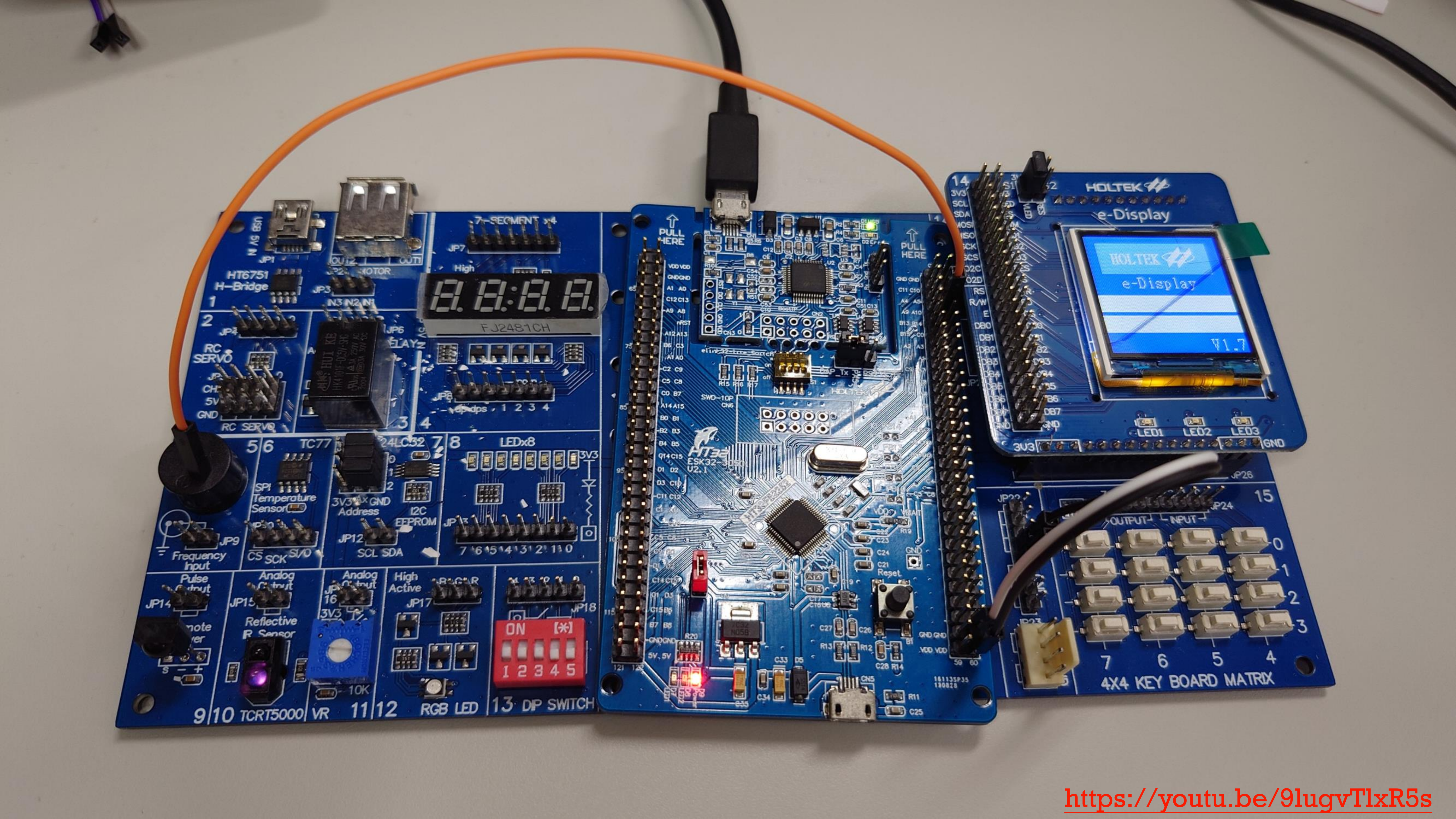
https://youtu.be/g1pHioxOzoQ

Homework W4-3

Play music faster.

- Objective: Edit delay time.
- Hint:

1. Use pin **A10.**
2. Edit Buzzer_PlayTable function



https://youtu.be/9lugvTlxR5s

Class Dismissed