Micro-Controller Experiment

Week8

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. Step to merge.
2. PWM_Buzzer Project.
3. ADC Project.
4. Merge both project.
5. Homework 8-1.
6. Homework 8-2.

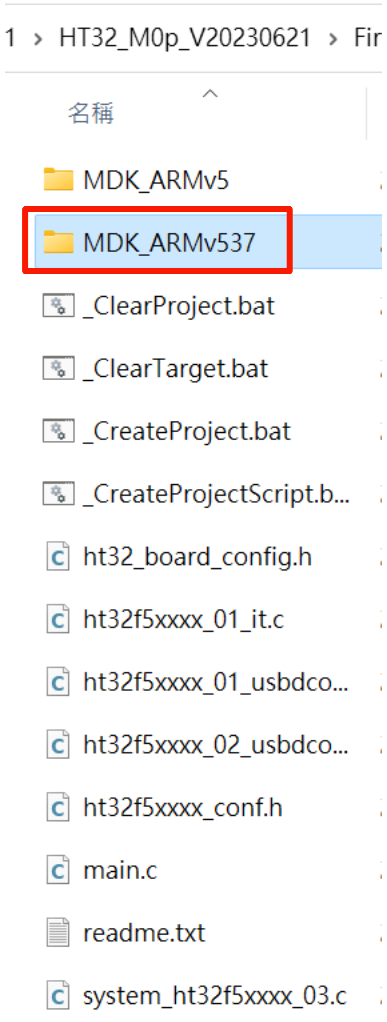
Steps

Steps

1. Confirm that the respective projects can be executed.
2. Understand the functions that will be used in the projects.
3. Decide which project is being merged into which project.
4. Edit code in main to implement functionality.
5. Add the required functions and files.

PWM_Buzzer Project

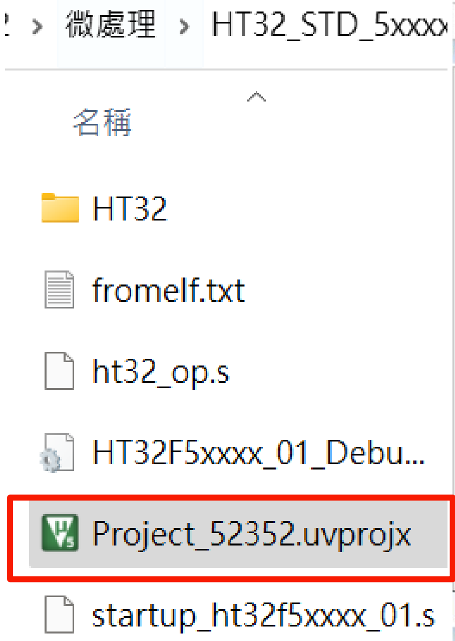
Launch project



1 › HT32_M0p_V20230621 › Fir

名稱

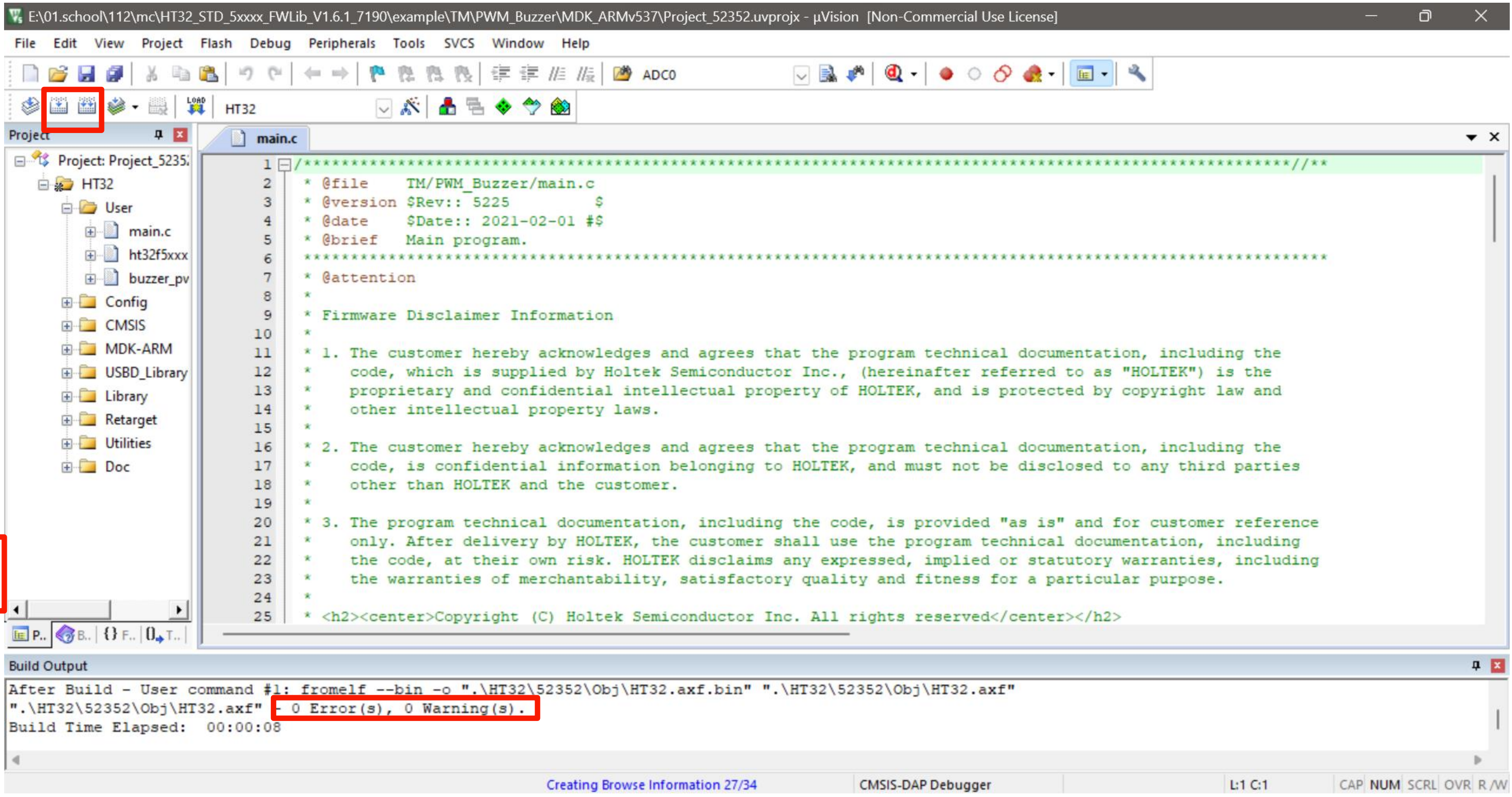
- MDK_ARMv5
- MDK_ARMv537
- _ClearProject.bat
- _ClearTarget.bat
- _CreateProject.bat
- _CreateProjectScript.b...
- ht32_board_config.h
- ht32f5xxxx_01_it.c
- ht32f5xxxx_01_usbdco...
- ht32f5xxxx_02_usbdco...
- ht32f5xxxx_conf.h
- main.c
- readme.txt
- system_ht32f5xxxx_03.c



› 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\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>
```

```
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   L:1 C:1   CAP NUM SCRL OVR R/W

Functions

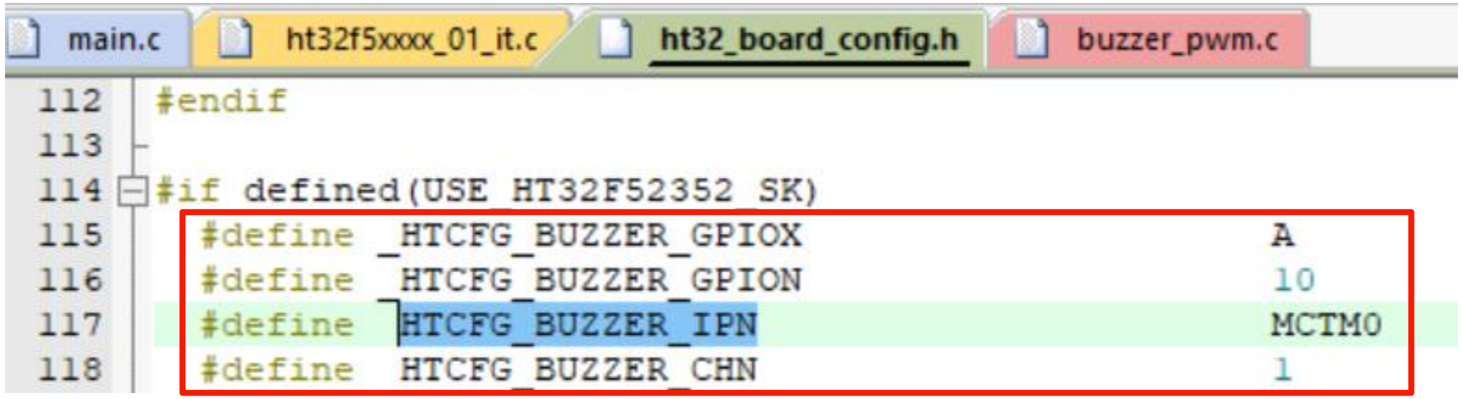
- Buzzer_Fun1():Bee 4 times, 3 kHz, active 50 ms, inactive 50 ms
- Buzzer_Fun2():Bee 2 times, 800 Hz, active 1000 ms, inactive 500 ms
- Buzzer_PlayTable():Bee 1 times, gBee_Scale[i] Hz, active 250 ms, inactive 250 ms

main.c

- Buzzer_Init(0):Initialization Setting
- Buzzer_Start()
- Buzzer_IsFinish()

buzzer_pwm.c

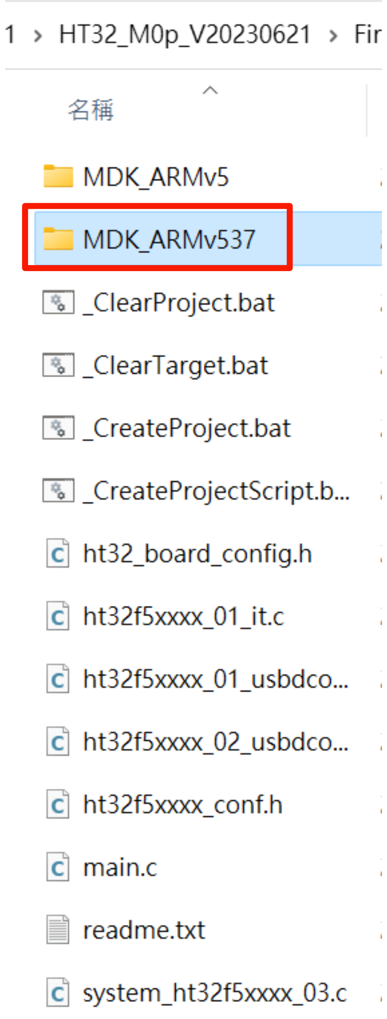
ht32_board_config.h



```
#endif

#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
```

ADC Project

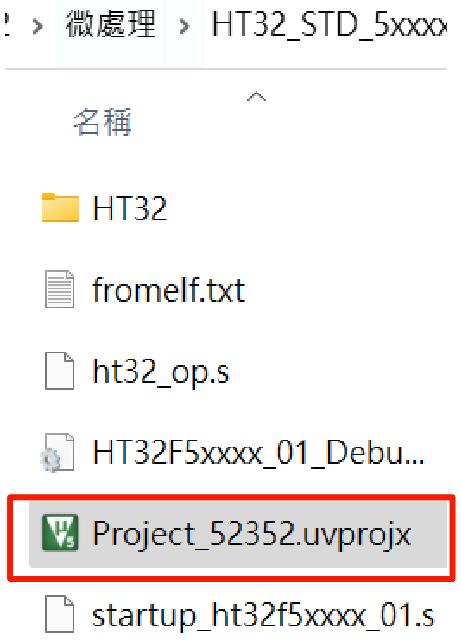
Launch project



1 › HT32_M0p_V20230621 › Fir

名稱

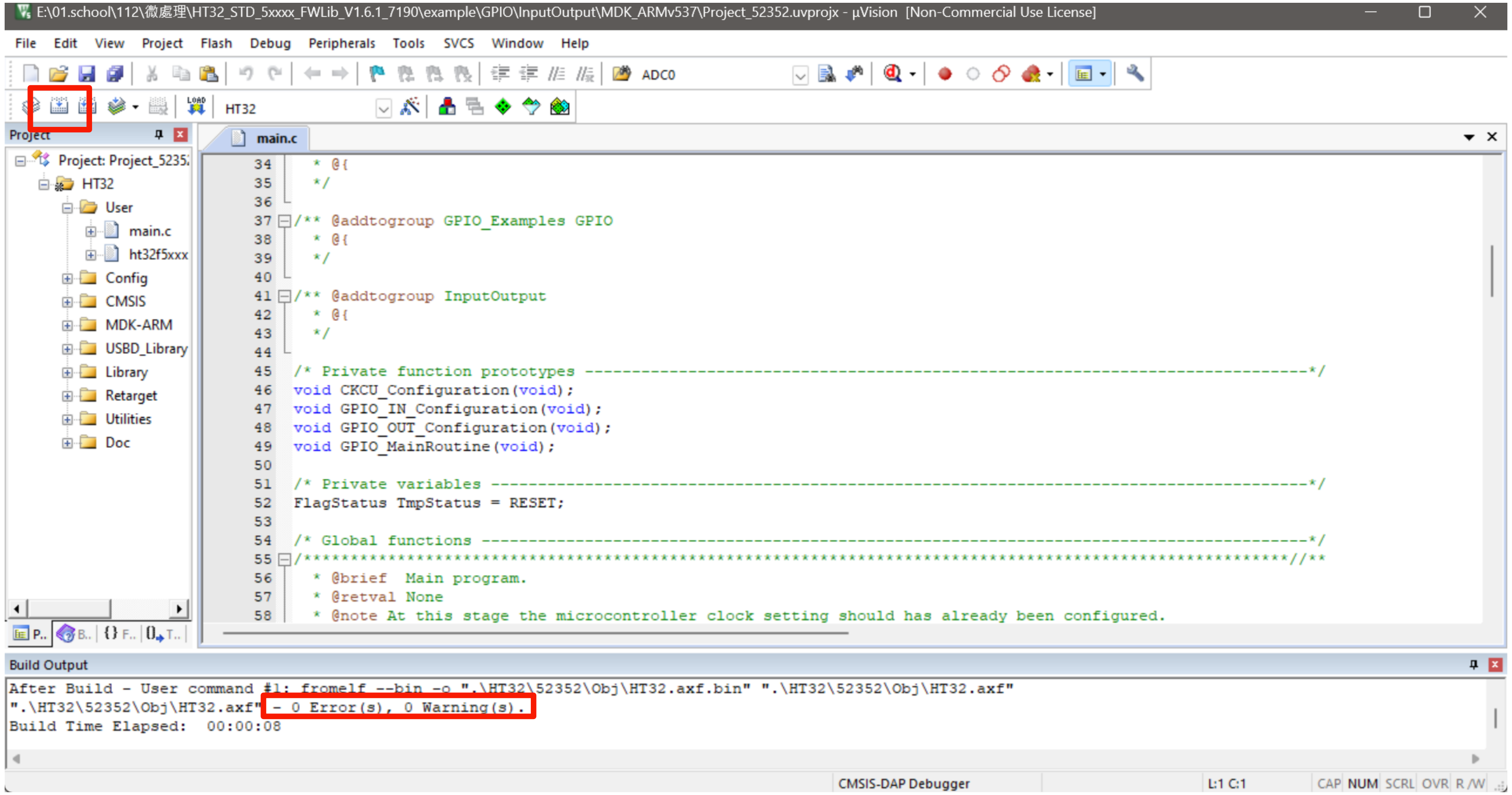
- MDK_ARMv5
- MDK_ARMv537
- _ClearProject.bat
- _ClearTarget.bat
- _CreateProject.bat
- _CreateProjectScript.b...
- ht32_board_config.h
- ht32f5xxxx_01_it.c
- ht32f5xxxx_01_usbdco...
- ht32f5xxxx_02_usbdco...
- ht32f5xxxx_conf.h
- main.c
- readme.txt
- system_ht32f5xxxx_03.c



› 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

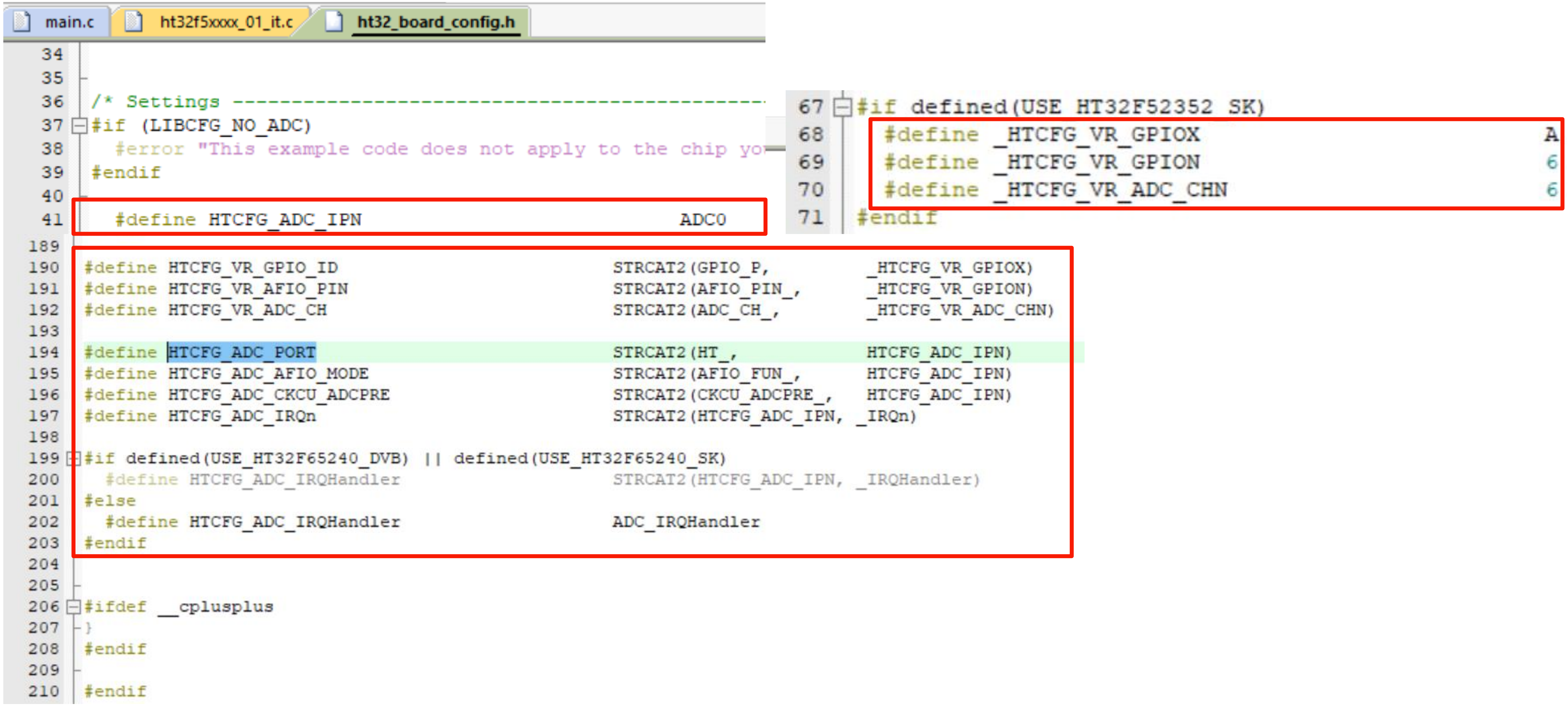
```
After Build - User command #1: fromelf --bin -o ".\HT32\52352\Obj\HT32.axf.bin" ".\HT32\52352\Obj\HT32.axf"
".\HT32\52352\Obj\HT32.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:08
```

Functions & File

- ADC_Configuration():
- ADC_Cmd(): Enable ADC
- ADC_SoftwareStartConvCmd(): Software trigger to start ADC conversion

→ main.c

ht32_board_config.h

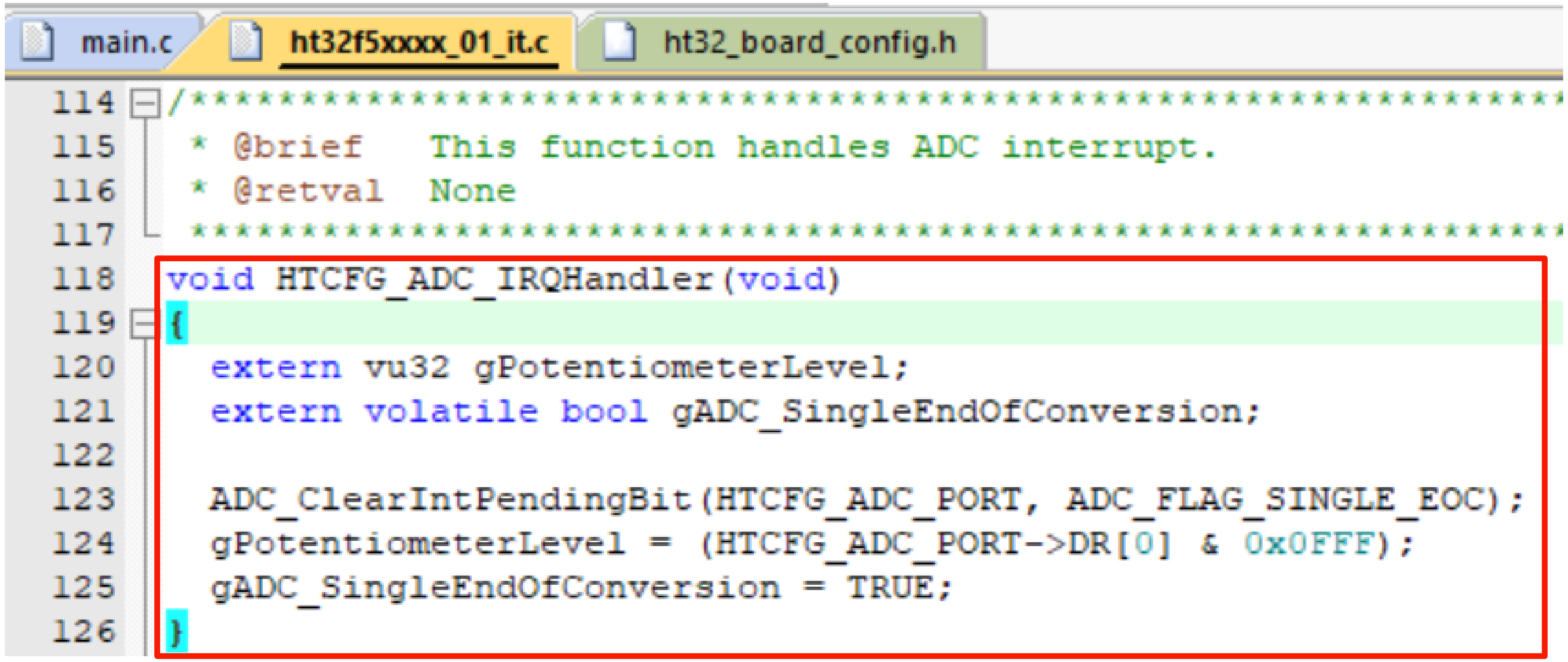


```c


/* Settings -----------------------------------------------
#if (LIBCFG_NO_ADC)
  #error "This example code does not apply to the chip yo
#endif

  #define HTCFG_ADC_IPN                                  ADC0
```

```c
#if defined(USE_HT32F52352_SK)
  #define _HTCFG_VR_GPIOX                                    A
  #define _HTCFG_VR_GPION                                    6
  #define _HTCFG_VR_ADC_CHN                                  6
#endif
```

```c

#define HTCFG_VR_GPIO_ID                   STRCAT2(GPIO_P,        _HTCFG_VR_GPIOX)
#define HTCFG_VR_AFIO_PIN                  STRCAT2(AFIO_PIN_,     _HTCFG_VR_GPION)
#define HTCFG_VR_ADC_CH                    STRCAT2(ADC_CH_,       _HTCFG_VR_ADC_CHN)

#define HTCFG_ADC_PORT                     STRCAT2(HT_,           HTCFG_ADC_IPN)
#define HTCFG_ADC_AFIO_MODE                STRCAT2(AFIO_FUN_,     HTCFG_ADC_IPN)
#define HTCFG_ADC_CKCU_ADCPRE              STRCAT2(CKCU_ADCPRE_,  HTCFG_ADC_IPN)
#define HTCFG_ADC_IRQn                     STRCAT2(HTCFG_ADC_IPN, _IRQn)

#if defined(USE_HT32F65240_DVB) || defined(USE_HT32F65240_SK)
  #define HTCFG_ADC_IRQHandler             STRCAT2(HTCFG_ADC_IPN, _IRQHandler)
#else
  #define HTCFG_ADC_IRQHandler             ADC_IRQHandler
#endif


#ifdef __cplusplus
}
#endif

#endif
```

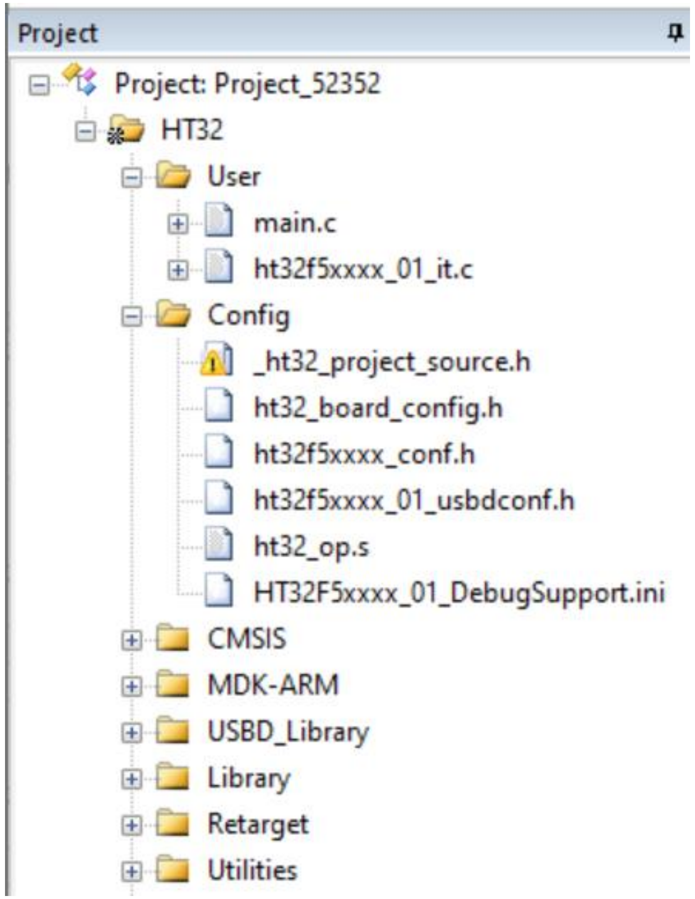
ht32f5xxxx_01_it.c



```c
/*****************************************************************************************
  * @brief   This function handles ADC interrupt.
  * @retval  None
  ****************************************************************************************
void HTCFG_ADC_IRQHandler(void)
{
  extern vu32 gPotentiometerLevel;
  extern volatile bool gADC_SingleEndOfConversion;

  ADC_ClearIntPendingBit(HTCFG_ADC_PORT, ADC_FLAG_SINGLE_EOC);
  gPotentiometerLevel = (HTCFG_ADC_PORT->DR[0] & 0x0FFF);
  gADC_SingleEndOfConversion = TRUE;
}
```

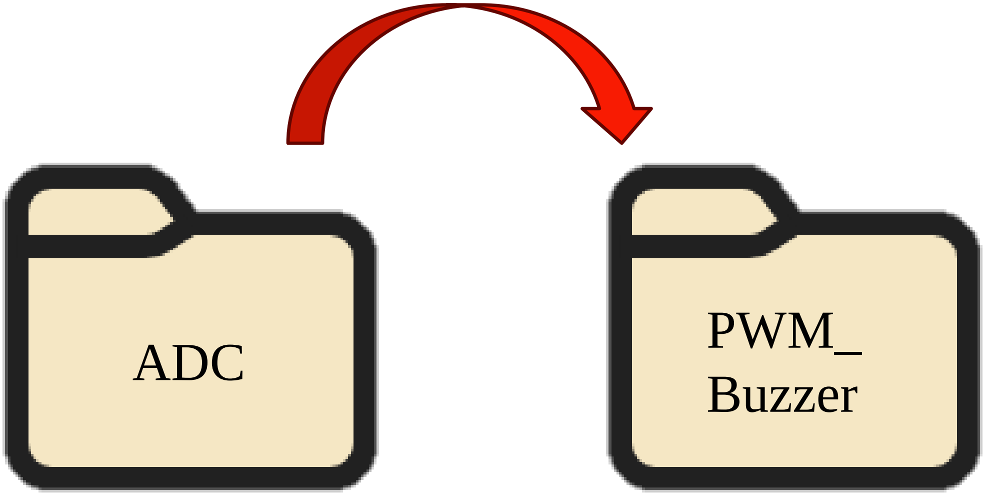
Merge Projects

ADC Project & PWM_Buzzer Project



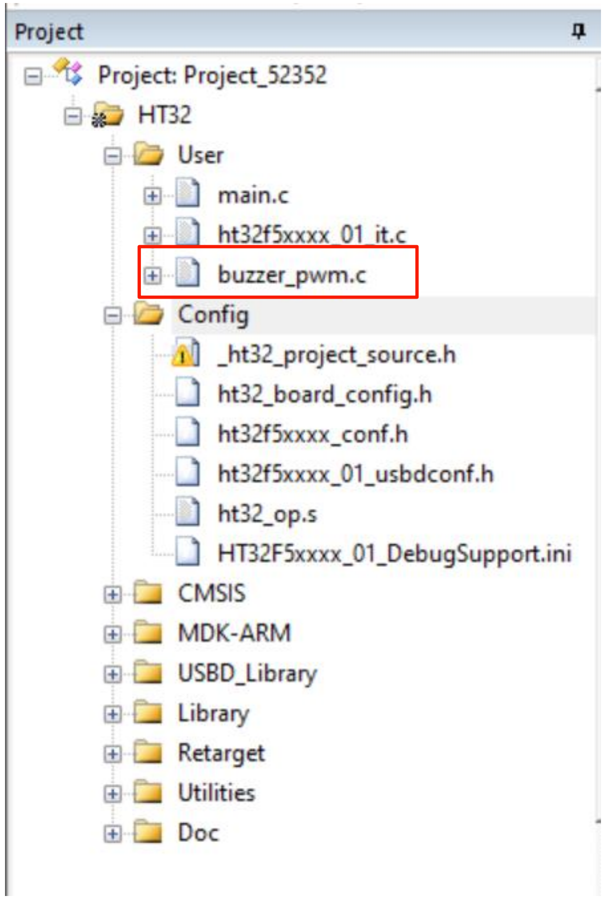
Project

- Project: Project_52352
  - HT32
    - User
      - main.c
      - ht32f5xxxx_01_it.c
    - Config
      - _ht32_project_source.h
      - ht32_board_config.h
      - ht32f5xxxx_conf.h
      - ht32f5xxxx_01_usbdconf.h
      - ht32_op.s
      - HT32F5xxxx_01_DebugSupport.ini
    - CMSIS
    - MDK-ARM
    - USBD_Library
    - Library
    - Retarget
    - Utilities



ADC

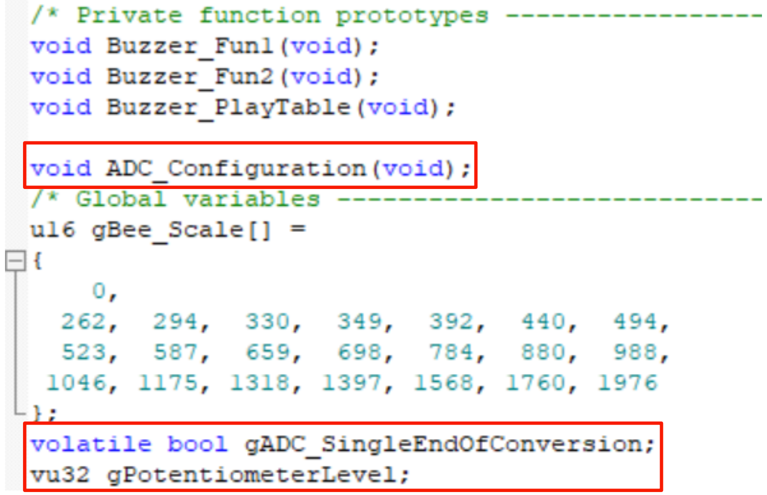
PWM_
Buzzer



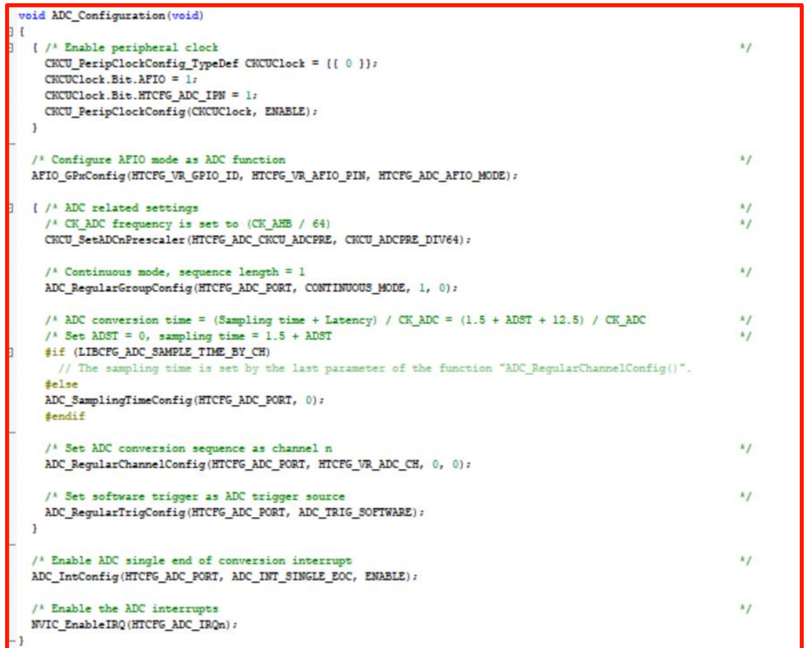
Project

- Project: Project_52352
  - HT32
    - User
      - main.c
      - ht32f5xxxx_01_it.c
      - buzzer_pwm.c
    - Config
      - _ht32_project_source.h
      - ht32_board_config.h
      - ht32f5xxxx_conf.h
      - ht32f5xxxx_01_usbdconf.h
      - ht32_op.s
      - HT32F5xxxx_01_DebugSupport.ini
    - CMSIS
    - MDK-ARM
    - USBD_Library
    - Library
    - Retarget
    - Utilities
    - Doc

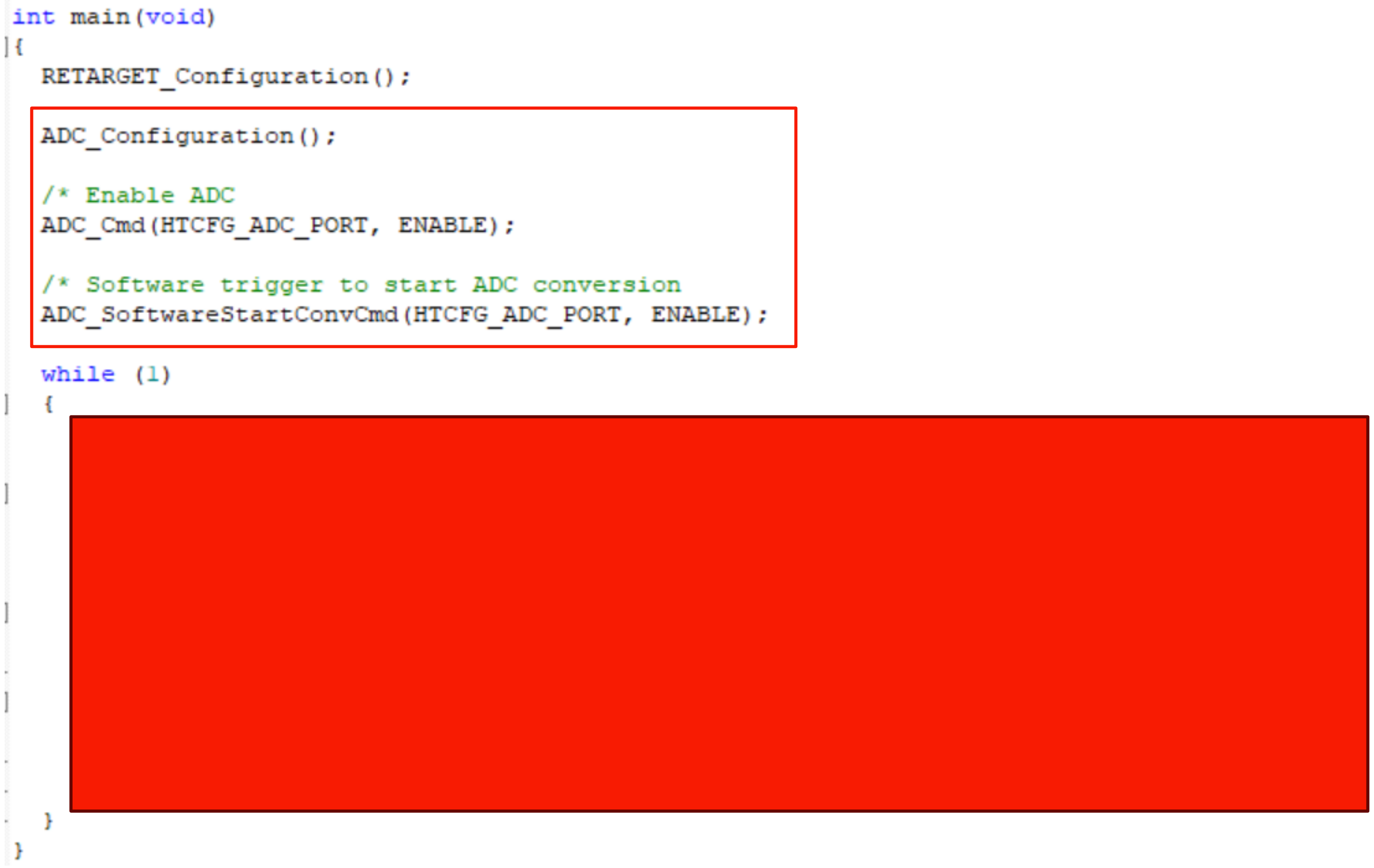
main.c



```c
/* Private function prototypes -----------------
void Buzzer_Fun1(void);
void Buzzer_Fun2(void);
void Buzzer_PlayTable(void);

void ADC_Configuration(void);
/* Global variables ---------------------------
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
};
volatile bool gADC_SingleEndOfConversion;
vu32 gPotentiometerLevel;
```



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

  /* Configure AFIO mode as ADC function                                        */
  AFIO_GPxConfig(HTCFG_VR_GPIO_ID, HTCFG_VR_AFIO_PIN, HTCFG_ADC_AFIO_MODE);

  { /* ADC related settings                                                     */
    /* CK_ADC frequency is set to (CK_AHB / 64)                                 */
    CKCU_SetADCnPrescaler(HTCFG_ADC_CKCU_ADCPRE, CKCU_ADCPRE_DIV64);

    /* Continuous mode, sequence length = 1                                     */
    ADC_RegularGroupConfig(HTCFG_ADC_PORT, CONTINUOUS_MODE, 1, 0);

    /* ADC conversion time = (Sampling time + Latency) / CK_ADC = (1.5 + ADST + 12.5) / CK_ADC  */
    /* Set ADST = 0, sampling time = 1.5 + ADST                                 */
    #if (LIBCFG_ADC_SAMPLE_TIME_BY_CH)
      // The sampling time is set by the last parameter of the function "ADC_RegularChannelConfig()".
    #else
    ADC_SamplingTimeConfig(HTCFG_ADC_PORT, 0);
    #endif

    /* Set ADC conversion sequence as channel n                                 */
    ADC_RegularChannelConfig(HTCFG_ADC_PORT, HTCFG_VR_ADC_CH, 0, 0);

    /* Set software trigger as ADC trigger source                               */
    ADC_RegularTrigConfig(HTCFG_ADC_PORT, ADC_TRIG_SOFTWARE);
  }

  /* Enable ADC single end of conversion interrupt                              */
  ADC_IntConfig(HTCFG_ADC_PORT, ADC_INT_SINGLE_EOC, ENABLE);

  /* Enable the ADC interrupts                                                  */
  NVIC_EnableIRQ(HTCFG_ADC_IRQn);
}
```

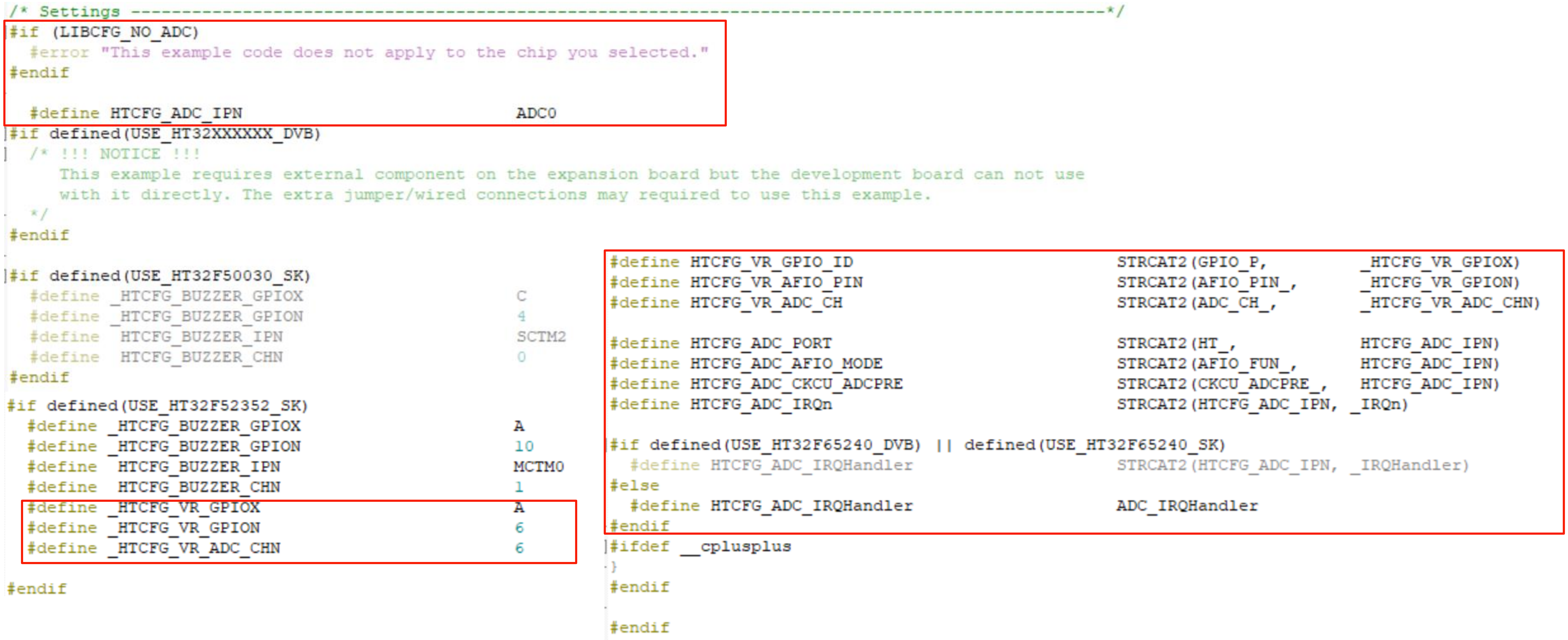


```c
int main(void)
{
  RETARGET_Configuration();

  ADC_Configuration();

  /* Enable ADC
  ADC_Cmd(HTCFG_ADC_PORT, ENABLE);

  /* Software trigger to start ADC conversion
  ADC_SoftwareStartConvCmd(HTCFG_ADC_PORT, ENABLE);

  while (1)
  {
```

[redacted]

```c
  }
}
```

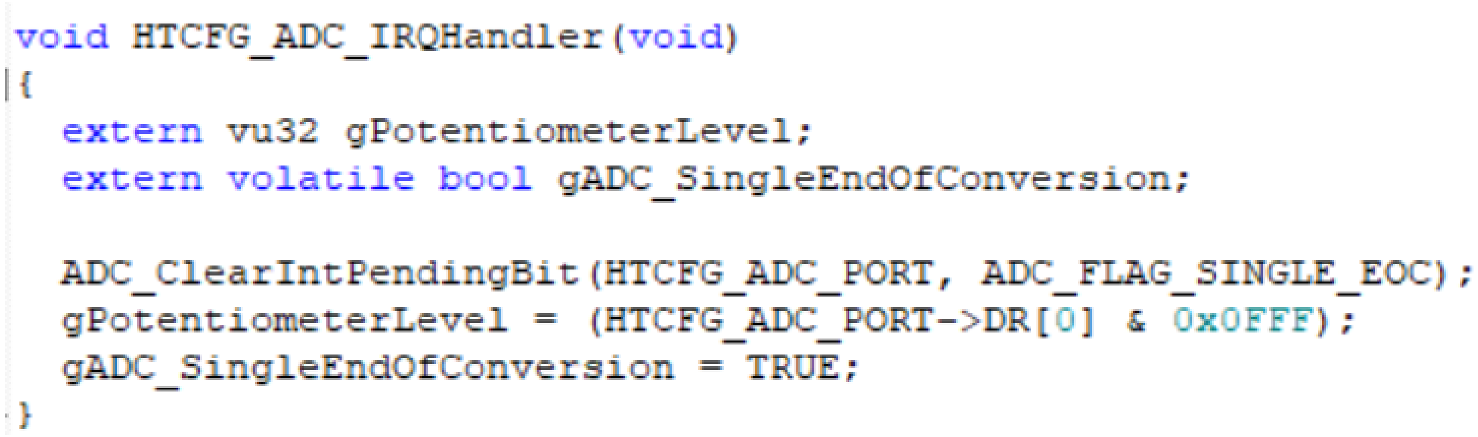
ht32_board_config.h



```
/* Settings ------------------------------------------------------------------------------------------*/
#if (LIBCFG_NO_ADC)
  #error "This example code does not apply to the chip you selected."
#endif

  #define HTCFG_ADC_IPN                              ADC0
#if defined(USE_HT32XXXXXX_DVB)
  /* !!! NOTICE !!!
     This example requires external component on the expansion board but the development board can not use
     with it directly. The extra jumper/wired connections may required to use this example.
  */
#endif

#if defined(USE_HT32F50030_SK)
  #define _HTCFG_BUZZER_GPIOX                        C
  #define _HTCFG_BUZZER_GPION                        4
  #define  HTCFG_BUZZER_IPN                          SCTM2
  #define  HTCFG_BUZZER_CHN                          0
#endif

#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
  #define _HTCFG_VR_GPIOX                            A
  #define _HTCFG_VR_GPION                            6
  #define _HTCFG_VR_ADC_CHN                          6

#endif
```

```
#define HTCFG_VR_GPIO_ID                      STRCAT2(GPIO_P,          _HTCFG_VR_GPIOX)
#define HTCFG_VR_AFIO_PIN                     STRCAT2(AFIO_PIN_,       _HTCFG_VR_GPION)
#define HTCFG_VR_ADC_CH                       STRCAT2(ADC_CH_,         _HTCFG_VR_ADC_CHN)

#define HTCFG_ADC_PORT                        STRCAT2(HT_,             HTCFG_ADC_IPN)
#define HTCFG_ADC_AFIO_MODE                   STRCAT2(AFIO_FUN_,       HTCFG_ADC_IPN)
#define HTCFG_ADC_CKCU_ADCPRE                 STRCAT2(CKCU_ADCPRE_,    HTCFG_ADC_IPN)
#define HTCFG_ADC_IRQn                        STRCAT2(HTCFG_ADC_IPN,  _IRQn)

#if defined(USE_HT32F65240_DVB) || defined(USE_HT32F65240_SK)
  #define HTCFG_ADC_IRQHandler                STRCAT2(HTCFG_ADC_IPN,  _IRQHandler)
#else
  #define HTCFG_ADC_IRQHandler                ADC_IRQHandler
#endif
#ifdef __cplusplus
}
#endif

#endif
```

ht32f5xxxx_01_it.c



```
void HTCFG_ADC_IRQHandler(void)
{
  extern vu32 gPotentiometerLevel;
  extern volatile bool gADC_SingleEndOfConversion;

  ADC_ClearIntPendingBit(HTCFG_ADC_PORT, ADC_FLAG_SINGLE_EOC);
  gPotentiometerLevel = (HTCFG_ADC_PORT->DR[0] & 0x0FFF);
  gADC_SingleEndOfConversion = TRUE;
}
```

Homework W8-1.

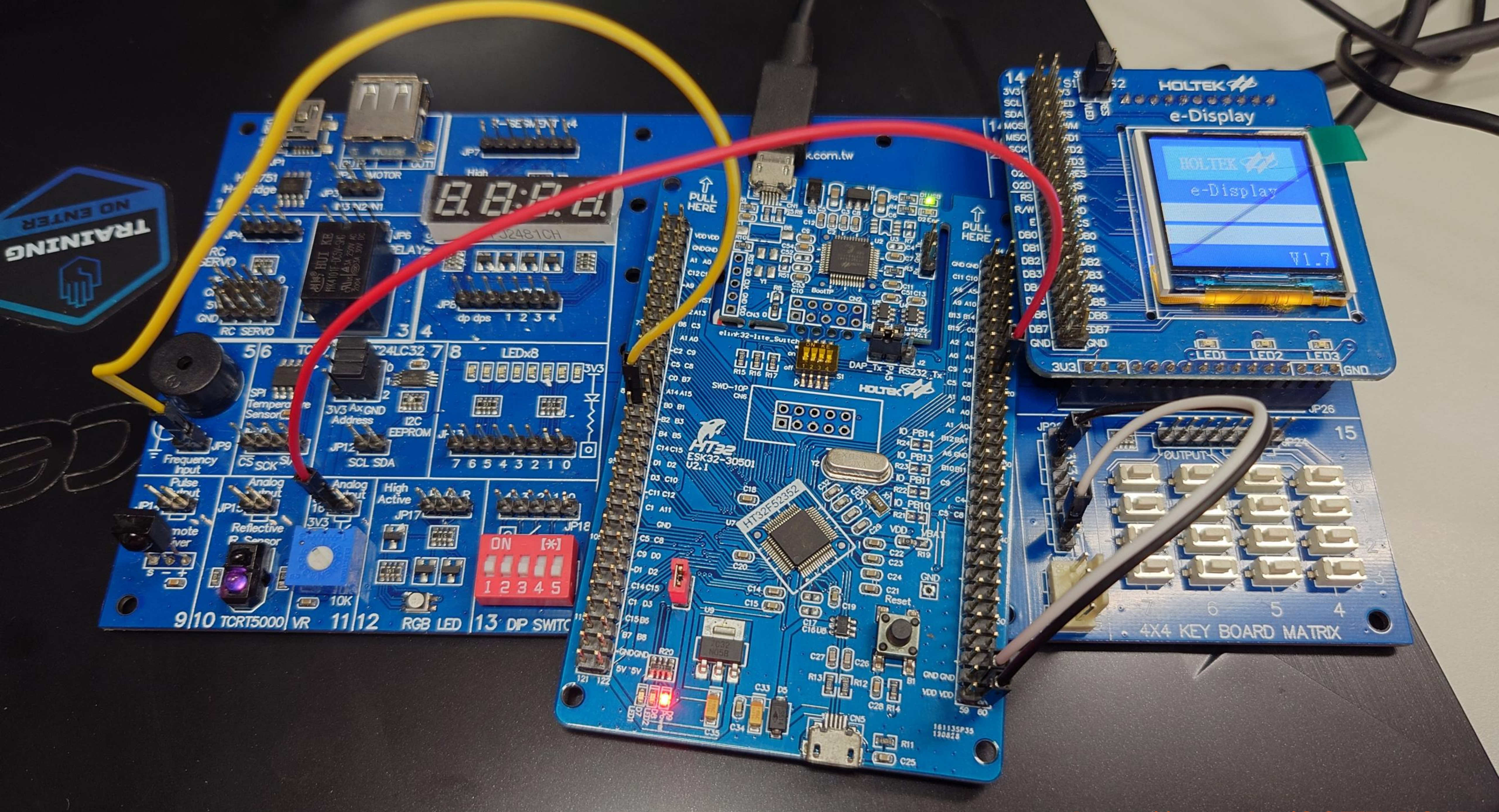
https://github.com/CYCU-AIoT-System-Lab/Microcontroller-Experiment/blob/main/w8/PWM_with_ADC-Experiment_Steps.md

Use a variable resistor to control the buzzer.

- Objective: Change the behavior of buzzer when gPotentiometerLevel>2000.
- Hint:

1. First, ensure that both projects are working.
2. Add the required functions and files.
3. Use **A7** for ADC, **B0** for Buzzer.

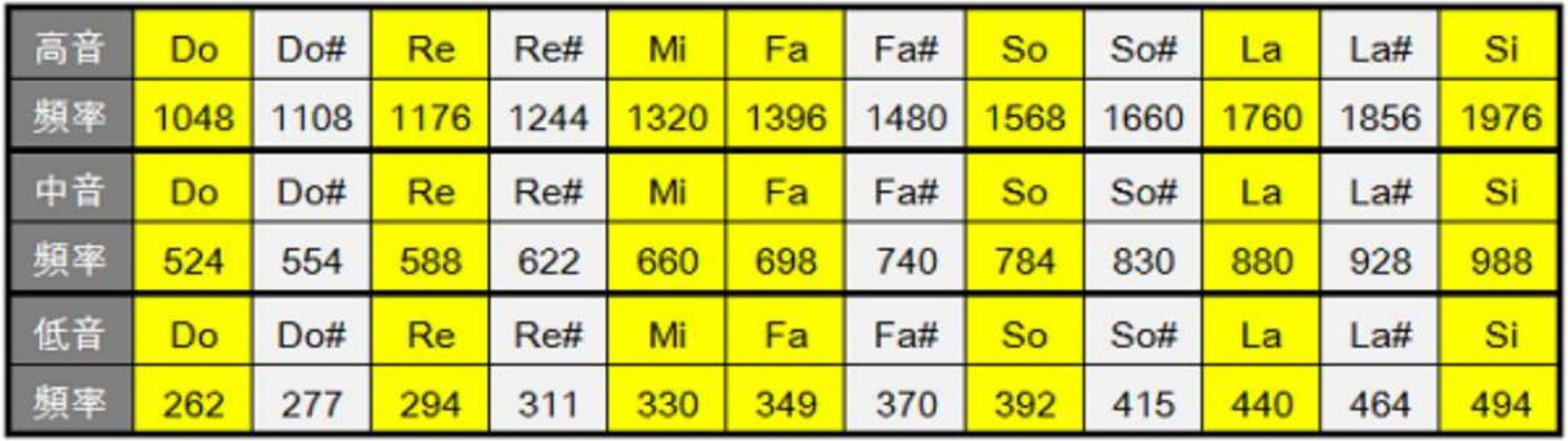
☆ PS. Please record.



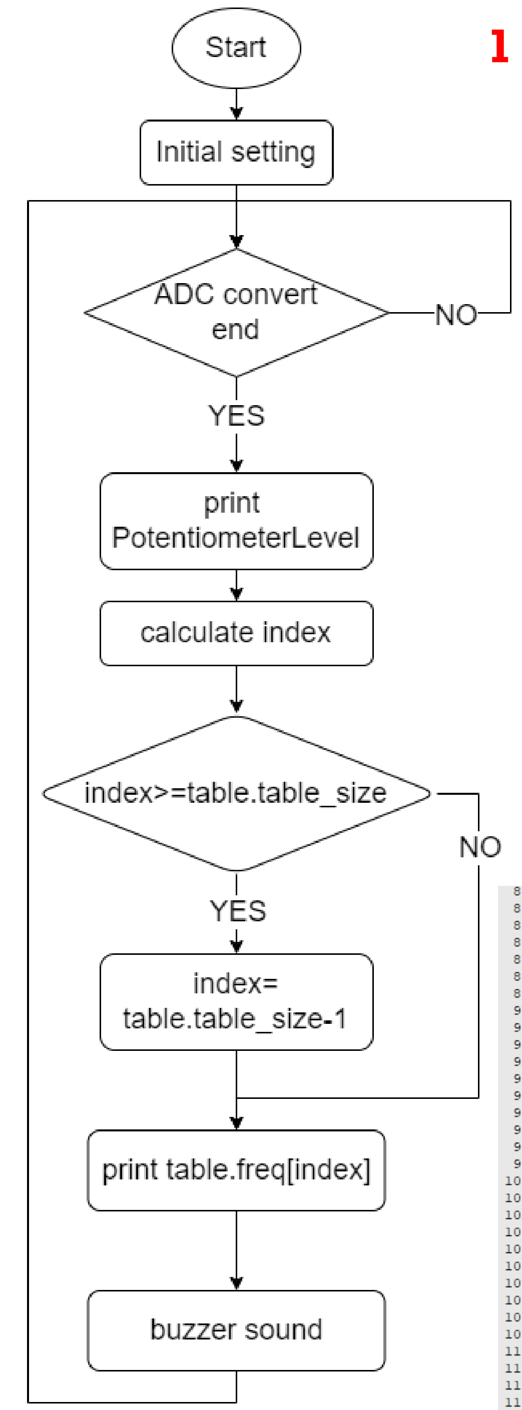
https://youtu.be/OQTJE1XG47o

Homework W8-2.

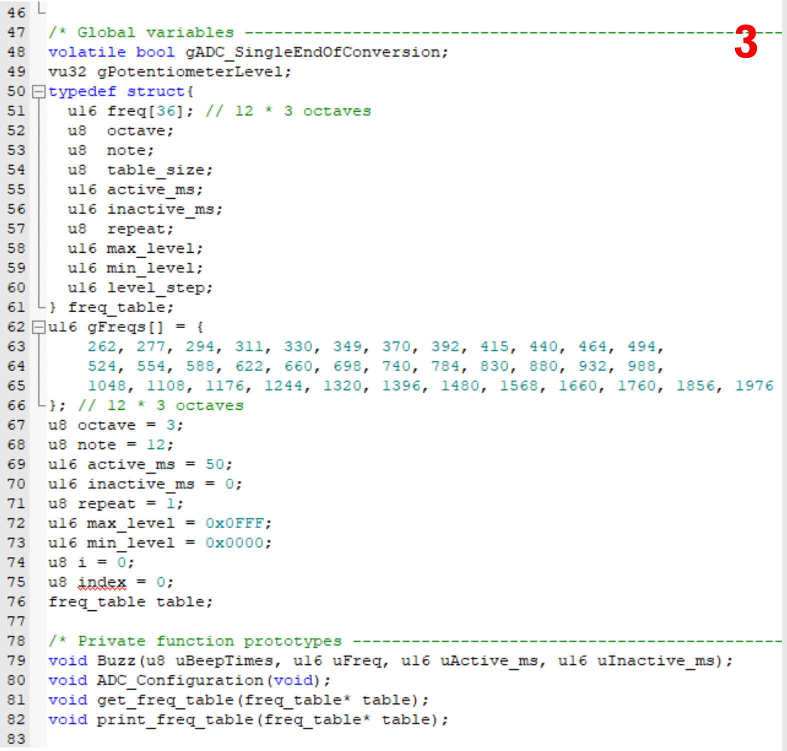
Scaled Frequency Table



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

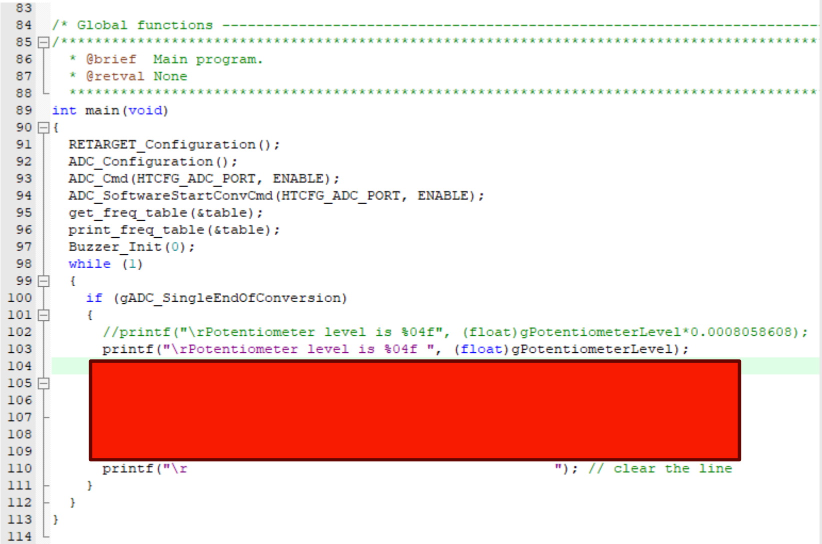


1



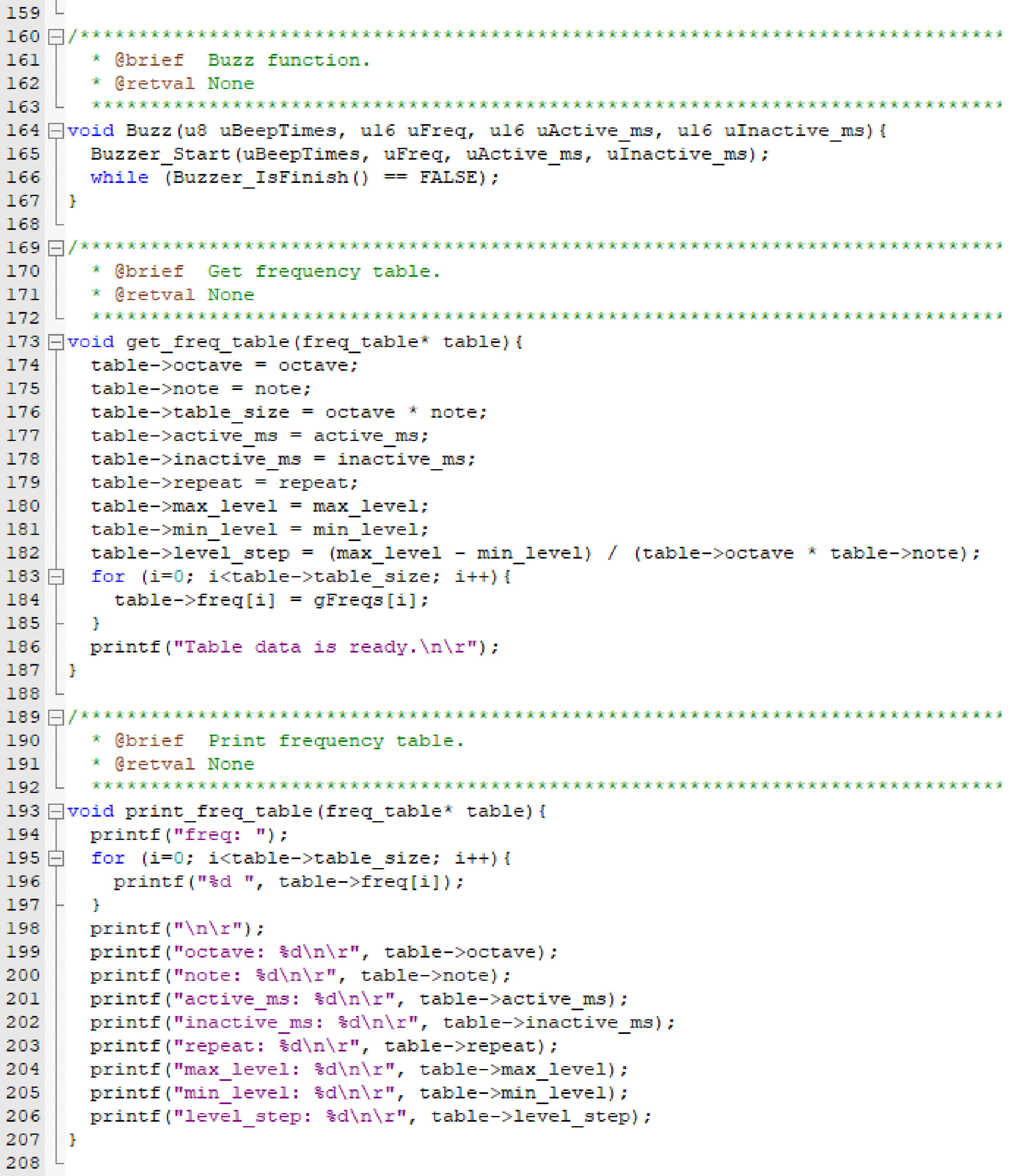
```c

/* Global variables -------------------------------------------------------
volatile bool gADC_SingleEndOfConversion;
vu32 gPotentiometerLevel;
typedef struct{
  u16 freq[36]; // 12 * 3 octaves
  u8  octave;
  u8  note;
  u8  table_size;
  u16 active_ms;
  u16 inactive_ms;
  u8  repeat;
  u16 max_level;
  u16 min_level;
  u16 level_step;
} freq_table;
u16 gFreqs[] = {
    262, 277, 294, 311, 330, 349, 370, 392, 415, 440, 464, 494,
    524, 554, 588, 622, 660, 698, 740, 784, 830, 880, 932, 988,
    1048, 1108, 1176, 1244, 1320, 1396, 1480, 1568, 1660, 1760, 1856, 1976
}; // 12 * 3 octaves
u8 octave = 3;
u8 note = 12;
u16 active_ms = 50;
u16 inactive_ms = 0;
u8 repeat = 1;
u16 max_level = 0x0FFF;
u16 min_level = 0x0000;
u8 i = 0;
u8 index = 0;
freq_table table;

/* Private function prototypes ---------------------------------------------
void Buzz(u8 uBeepTimes, u16 uFreq, u16 uActive_ms, u16 uInactive_ms);
void ADC_Configuration(void);
void get_freq_table(freq_table* table);
void print_freq_table(freq_table* table);

```

2



```c

/* Global functions --------------------------------------------------------
/*****************************************************************************
  * @brief  Main program.
  * @retval None
  ***************************************************************************
int main(void)
{
  RETARGET_Configuration();
  ADC_Configuration();
  ADC_Cmd(HTCFG_ADC_PORT, ENABLE);
  ADC_SoftwareStartConvCmd(HTCFG_ADC_PORT, ENABLE);
  get_freq_table(&table);
  print_freq_table(&table);
  Buzzer_Init(0);
  while (1)
  {
    if (gADC_SingleEndOfConversion)
    {
      //printf("\rPotentiometer level is %04f", (float)gPotentiometerLevel*0.0008058608);
      printf("\rPotentiometer level is %04f ", (float)gPotentiometerLevel);
      [redacted]
      printf("\r                                        "); // clear the line
    }
  }
}

```

3



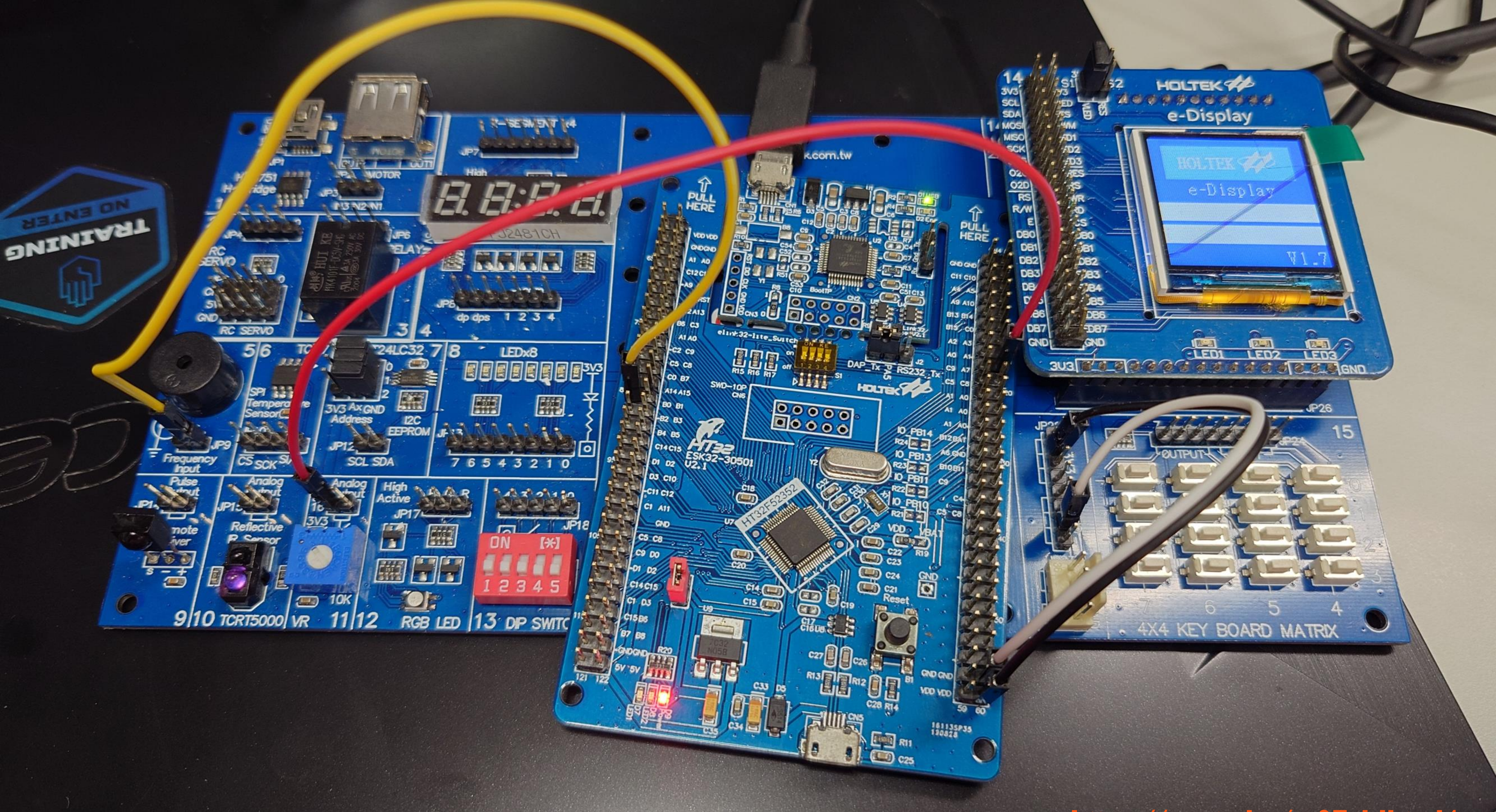
```c

/*****************************************************************************
  * @brief  Buzz function.
  * @retval None
  ***************************************************************************
void Buzz(u8 uBeepTimes, u16 uFreq, u16 uActive_ms, u16 uInactive_ms){
  Buzzer_Start(uBeepTimes, uFreq, uActive_ms, uInactive_ms);
  while (Buzzer_IsFinish() == FALSE);
}

/*****************************************************************************
  * @brief  Get frequency table.
  * @retval None
  ***************************************************************************
void get_freq_table(freq_table* table){
  table->octave = octave;
  table->note = note;
  table->table_size = octave * note;
  table->active_ms = active_ms;
  table->inactive_ms = inactive_ms;
  table->repeat = repeat;
  table->max_level = max_level;
  table->min_level = min_level;
  table->level_step = (max_level - min_level) / (table->octave * table->note);
  for (i=0; i<table->table_size; i++){
    table->freq[i] = gFreqs[i];
  }
  printf("Table data is ready.\n\r");
}

/*****************************************************************************
  * @brief  Print frequency table.
  * @retval None
  ***************************************************************************
void print_freq_table(freq_table* table){
  printf("freq: ");
  for (i=0; i<table->table_size; i++){
    printf("%d ", table->freq[i]);
  }
  printf("\n\r");
  printf("octave: %d\n\r", table->octave);
  printf("note: %d\n\r", table->note);
  printf("active_ms: %d\n\r", table->active_ms);
  printf("inactive_ms: %d\n\r", table->inactive_ms);
  printf("repeat: %d\n\r", table->repeat);
  printf("max_level: %d\n\r", table->max_level);
  printf("min_level: %d\n\r", table->min_level);
  printf("level_step: %d\n\r", table->level_step);
}

```

Use a variable resistor to control the buzzer.

- Objective: Make buzzer play all the note in P26 with the control of variable resistor.
- Hint:

1. Modify main.c as shown in P27.
2. Finish the code in while loop and execute it.
3. Use **A7** for ADC, **B0** for Buzzer.

☆ PS. Please record.



https://youtu.be/m9Z_blktml4

Class Dismissed