Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Архітектура комп'ютерів 3. Мікропроцесорні системи**

**Воркшоп 5**

«Communication: I2C»

Виконав:

студент групи ІО-23

Корбут М. Я.

Залікова книжка №2313

Перевірив

Каплунов А.В.

Київ - 2025

**Воркшоп 5**

**Тема:** «Communication: I2C»

**Мета:** зрозуміти як працюють I2C інтерфейс та як його налаштовувати.

**Хід роботи:**

У процесі роботи було активовано та налаштовано послідовний інтерфейс I2C для керування зовнішнім ЦАП CS43L22 та його вбудованим генератором тонів. Також використовувался інтерфейси DMA та I2S3. З програмного боку використовувалися спеціальні функції для ініціалізації модуля та передачі йому команд для програвання мелодії. Також використовувалася спеціальна функція для створення світломузики за допомогою вбудованих світлодіодів. Відповідний фрагмент коду з файлу main.c:

typedef enum

{

  C4,

  C5,

  D5,

  E5,

  F5,

  G5,

  A5,

  B5,

  C6,

  D6,

  E6,

  F6,

  G6,

  A6,

  B6,

  C7,

  MAX\_VALUE

} soundToneType;

static const uint8\_t soundToneReg[16]={   //list of all possible register values

    0x09,

    0x19,

    0x29,

    0x39,

    0x49,

    0x59,

    0x69,

    0x79,

    0x89,

    0x99,

    0xA9,

    0xB9,

    0xC9,

    0xD9,

    0xE9,

    0xF9

};

/\* USER CODE BEGIN PD \*/

#define CS43L22\_I2C\_ADDRESS   0x94

#define I2C\_TIMEOUT       10

/\* USER CODE END PD \*/

/\* USER CODE BEGIN PV \*/

int16\_t dataI2S[100] = {0};

/\* USER CODE END PV \*/

void CS43L22\_Init(void)

{

  // Enable chip

  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);

  //

  // Initialization

  //

  uint8\_t TxBuffer[2];

  TxBuffer[0] = 0x0D;

  TxBuffer[1] = 0x01;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x00;

  TxBuffer[1] = 0x99;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x47;

  TxBuffer[1] = 0x80;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x32;

  TxBuffer[1] = 0xFF;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x32;

  TxBuffer[1] = 0x7F;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x00;

  TxBuffer[1] = 0x00;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x04;

  TxBuffer[1] = 0xAF;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x0D;

  TxBuffer[1] = 0x70;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x05;

  TxBuffer[1] = 0x81;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x06;

  TxBuffer[1] = 0x07;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x0A;

  TxBuffer[1] = 0x00;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x27;

  TxBuffer[1] = 0x00;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x1A;

  TxBuffer[1] = 0x0A;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x1B;

  TxBuffer[1] = 0x0A;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x1F;

  TxBuffer[1] = 0x0F;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  TxBuffer[0] = 0x02;

  TxBuffer[1] = 0x9E;

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

}

void CS43L22\_Beep(soundToneType pitch, uint32\_t duration\_ms)

{

  uint8\_t TxBuffer[2];

  // Set volume and off time

  TxBuffer[0] = 0x1D;   // Register address

  TxBuffer[1] = 0x00;   // Value (volume and off time)

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  // Set sound frequency

  TxBuffer[0] = 0x1C;   // Register address

  if (pitch < MAX\_VALUE) {

    TxBuffer[1] = soundToneReg[pitch]; //Set pitch from the list

  }

  else{

    TxBuffer[1] = soundToneReg[15]; //Set the highest available

  }

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  // Enable continuous mode (SOUND STARTED)

  TxBuffer[0] = 0x1E;   // Register address

  TxBuffer[1] = 0xC0;   // Value (beep and tone configuration)

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

  // Playing...

  HAL\_Delay(duration\_ms);

  // Disable continuous mode (SOUND STOPED)

  TxBuffer[0] = 0x1E;   // Register address

  TxBuffer[1] = 0x00;   // Value (beep and tone configuration)

  HAL\_I2C\_Master\_Transmit(&hi2c1, CS43L22\_I2C\_ADDRESS, (uint8\_t\*) &TxBuffer, 2, I2C\_TIMEOUT);

}

void lightMusic(soundToneType note){

  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12 | GPIO\_PIN\_13 | GPIO\_PIN\_14 | GPIO\_PIN\_15, GPIO\_PIN\_RESET);  // disables all LEDs

  switch(note){ // picks LEDs to light based on the note selected

  case C4:

  case C5:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);

    break;

  case D5:

  case D6:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);

    break;

  case E5:

  case E6:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);

    break;

  case F5:

  case F6:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);

    break;

  case G5:

  case G6:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12 | GPIO\_PIN\_13, GPIO\_PIN\_SET);

    break;

  case A5:

  case A6:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13 | GPIO\_PIN\_14, GPIO\_PIN\_SET);

    break;

  case B5:

  case B6:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14 | GPIO\_PIN\_15, GPIO\_PIN\_SET);

    break;

  case C6:

  case C7:

  default:

    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_15 | GPIO\_PIN\_12, GPIO\_PIN\_SET);

    break;

  }

}

/\* USER CODE END 0 \*/

/\* USER CODE BEGIN WHILE \*/

  HAL\_Delay(1000);  //delay before starting

  int i = 1;

  soundToneType melody[] = {

    A5, A5, A5, F5, C6, A5, F5, C6, A5,

      E6, E6, E6, F6, C6, G5, F5, C6,

      A5, F5, C6, A5, F5, C6, A5,

      E6, E6, E6, F6, C6, G5, F5, C6,

      // remaining unused notes for full coverage

      B5, D5, B6, D6, G6, A6, B6, C7, C4};

  uint32\_t noteDurations[] = {

      500, 500, 500, 350, 150, 500, 350, 150, 1000,

      500, 350, 150, 500, 350, 150, 500, 1000,

      500, 500, 350, 150, 500, 350, 150,

      500, 350, 150, 500, 350, 150, 500, 1000,

      // Fast run through all remaining notes

      200, 200, 200, 200, 200, 200, 200, 500,

      400};

  int melodyNoteCount = sizeof(melody) / sizeof(melody[0]);

  while (i<4) { //repeat the melody 3 times

      for(int j=0; j<melodyNoteCount; ++j){ //for each note

          CS43L22\_Beep(melody[j], noteDurations[j]);  //Play the note

          lightMusic(melody[j]);  //activate LEDs

          HAL\_Delay(50);  //wait before the next note

      }

      HAL\_Delay(1000);  //delay between melodies

      i++;

  }

  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12 | GPIO\_PIN\_13 | GPIO\_PIN\_14 | GPIO\_PIN\_15, GPIO\_PIN\_RESET);  // disables all LEDs

  while (1) {}

  /\* USER CODE END 3 \*/

Зауважте, що коментарі написано ангілйською, адже вони писалися на віддаленій машині, де я не знайшов можливості змінити розкладку.

**Результати роботи програми:**

Після запуску, програма тричі відтворює мелодію з “зоряних війн”, супроводжуючи її світломузикою.

**Репозиторій**

Код було завантажено до репозиторію GitHub. Переглянути його можна за [посиланням](https://github.com/Misha1tigr/ak3-labs-korbut/tree/main/Workshop_5).