**Аппаратные платформы встраиваемых систем**

Отчет по лабораторной работе №4

На тему **«Пьезоизлучатель (Buzzer)»**

Работу выполнили:

Студенты гр. 13541/1 Онищенко Д.И.

Шаменов А.А.

Преподаватель: Васильев А.Е.

Содержание

[Цель работы: 3](#_Toc11065571)

[Подготовка к работе: 3](#_Toc11065572)

[Теоретическая информация: 3](#_Toc11065573)

[Ход работы: 5](#_Toc11065574)

[Вывод 10](#_Toc11065575)

# Цель работы:

Получение навыков работы с пьезоизлучателем (buzzer) МК STM32.

# Подготовка к работе:

1. Подготовить проект в IARWE, согласно документу IAR Project for IAR SK Board
2. Ознакомиться со схемой платы IAR SK (STM32F407ZG-board-schematic.pdf)
3. Ознакомиться с документацией МК STM32F407 (STM32F4xx\_RM.pdf)

# Теоретическая информация:

**Пьезоизлучатель.**

Пьезоэлектрический излучатель, пьезоизлучатель – электроакустическое устройство, способное воспроизводить звук, либо излучать ультразвук, благодаря обратному пьезоэлектрическому эффекту (т.е. эффекту возникновения поляризации диэлектрика под действием механических напряжений).

На плате STM32 расположен пьезоизлучатель PB-1221PQ, приведем его характеристики в нижеследующей таблице

Таблица 1 – Характеристики пьезоизлучателя PB-1221PQ

|  |  |
| --- | --- |
| Частота колебаний (Гц) | 2048 |
| Рабочее напряжение (В) | 1.25 ~ 2.5 |
| Номинальное напряжение (В) | 1.5 |
| Максимальное потребление (mA) | 18 / ном. напр. |
| Уровень звукового давления (dB/min) | 8 / ном. напр. |
| Сопротивление катушки (Ω) | 42±5 |
| Рабочая температура, 0C | -20~+70 |
| Температура хранения, 0C | -30~+80 |

Схема подключения пьезоизлучателя к плате показана ниже. Отметим, что он подключен от источника 3.3В, управляющий сигнал - BUZ:

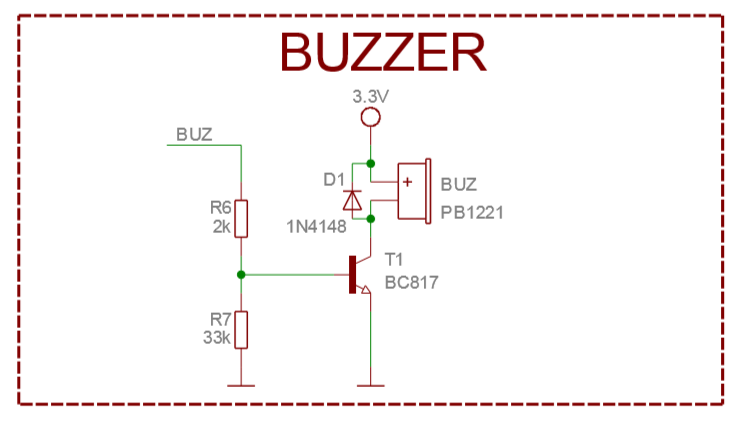


Рисунок 1 – Схема подключения пьезоизлучателя

# Ход работы:

**Задание состоит из следующих пунктов:**

1. Изучить работу с пьезоизлучателем. Необходимо написать программу, в ней задать такие конфигурации ШИМа и таймера, чтобы пьезоизлучатель воспроизводил звук путем колебания звуковых волн. Модифицировать программу.

2. При нажатии кнопки WKUP\_BTN необходимо, чтобы пьезоизлучатель воспроизводил звук, в противном случае звук не должен воспроизводится.

**Практическая часть:**

Продемонстрируем последовательность действий для задания №1**.**

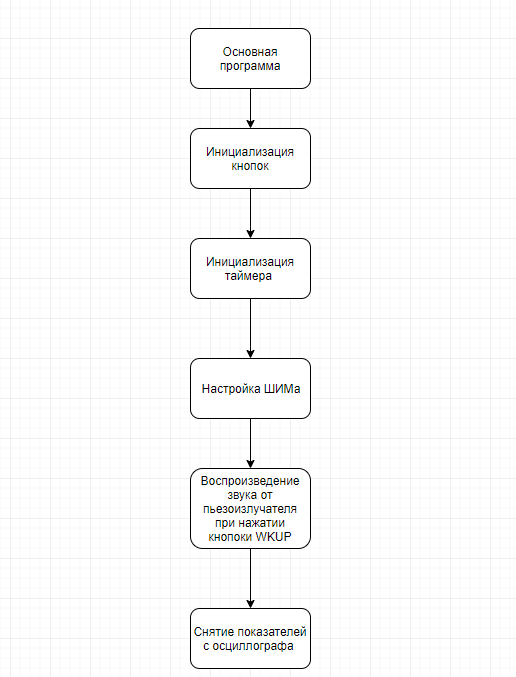


Рисунок 2 – Схема последовательности действии для первого задания

Проинициализируем кнопки и светодиоды:

|  |
| --- |
| 1. void BTN\_init (void) { 2. /\* FOR User Button \*/ 3. RCC\_AHB1PeriphClockCmd(BUTTON\_USER\_RCC, ENABLE); 4. GPIO\_InitStruct.GPIO\_Pin = BUTTON\_USER\_PIN; 5. GPIO\_InitStruct.GPIO\_Mode = GPIO\_Mode\_IN; 6. GPIO\_Init(BUTTON\_USER\_PORT, &GPIO\_InitStruct); 7. /\* FOR WKUP \*/ 8. RCC\_AHB1PeriphClockCmd(BUTTON\_WKUP\_RCC, ENABLE); 9. GPIO\_InitStruct.GPIO\_Pin = BUTTON\_WKUP\_PIN; 10. GPIO\_InitStruct.GPIO\_Mode = GPIO\_Mode\_IN; 11. GPIO\_Init(BUTTON\_WKUP\_PORT, &GPIO\_InitStruct); 12. } |

Далее проинициализируем таймер и настроим ШИМ:

|  |
| --- |
| 1. GPIO\_InitTypeDef GPIO\_InitStructure; 2. TIM\_TimeBaseInitTypeDef TIM\_TimeBaseStructure; 3. TIM\_OCInitTypeDef TIM\_OCInitStructure; 4. RCC\_ClocksTypeDef RCC\_Clocks; 5. int pwm\_state = 0; 6. /\*!< At this stage the microcontroller clock setting is al 7. ready configured, 8. this is done through SystemInit() function which is called 9. from startup 10. file (startup\_stm32f4xx.s) before to branch to application 11. main. 12. To reconfigure the default setting of SystemInit() function, 13. refer to 14. system\_stm32f4xx.c file 15. \*/ 16. ENTR\_CRT\_SECTION(); 17. /\* SysTick Config\*/ 18. if(SysTick\_Config(SystemCoreClock/10000)) 19. { 20. /\* Capture error \*/ 21. while (1); 22. } 23. EXT\_CRT\_SECTION(); 24. //GLCD init 25. GLCD\_PowerUpInit((pInt8U)IAR\_Logo.pPicStream); 26. /\*Turn on Backlight\*/ 27. GLCD\_Backlight(BACKLIGHT\_OFF); 28. /\*TAMPER button init\*/ 29. STM\_EVAL\_PBInit(BUTTON\_TAMPER,BUTTON\_MODE\_GPIO); 30. /\* Enable the BUZZER\_GPIO Clock \*/ 31. RCC\_AHB1PeriphClockCmd(BUZZER\_GPIO\_CLK, ENABLE); 32. /\* Configure Buzzer pin\*/ 33. GPIO\_InitStructure.GPIO\_Pin = BUZZER\_GPIO\_PIN; 34. GPIO\_InitStructure.GPIO\_OType = GPIO\_OType\_PP; 35. GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AF; 36. GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_NOPULL; 37. GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz; 38. GPIO\_Init(BUZZER\_GPIO\_PORT, &GPIO\_InitStructure); 39. /\*Select AF\*/ 40. GPIO\_PinAFConfig(BUZZER\_GPIO\_PORT,BUZZER\_PIN\_SOURCE, 41. BUZZER\_PIN\_AF); 42. /\* Buzzer Timer enable\*/ 43. RCC\_APB2PeriphClockCmd(RCC\_APB2PERIPH\_BUZZER\_TIM,ENABLE); 44. RCC\_APB2PeriphResetCmd(RCC\_APB2PERIPH\_BUZZER\_TIM,DISABLE); 45. RCC\_APB2PeriphClockCmd(RCC\_APB2PERIPH\_BUZZER\_TIM, ENABLE); 46. RCC\_GetClocksFreq(&RCC\_Clocks); 47. /\*Time base configuration\*/ 48. TIM\_TimeBaseStructure.TIM\_Period = 0xFF; 49. /\*8 bit resolution\*/ 50. TIM\_TimeBaseStructure.TIM\_Prescaler = 51. RCC\_Clocks.HCLK\_Frequency/(256\*BUZZER\_FREQ); /\*2kHz PWM 52. periotd\*/ 53. TIM\_TimeBaseStructure.TIM\_CounterMode = TIM\_CounterMode\_Up; 54. TIM\_TimeBaseStructure.TIM\_ClockDivision = 0; 55. TIM\_TimeBaseStructure.TIM\_RepetitionCounter = 0; 56. TIM\_TimeBaseInit(BUZZER\_TIM,&TIM\_TimeBaseStructure); 57. /\*Output Compare init\*/ 58. TIM\_OCInitStructure.TIM\_OCMode = TIM\_OCMode\_PWM1; 59. TIM\_OCInitStructure.TIM\_OutputState = 60. TIM\_OutputState\_Disable; 61. TIM\_OCInitStructure.TIM\_OutputNState = 62. TIM\_OutputNState\_Disable; 63. /\*50%\*/ 64. TIM\_OCInitStructure.TIM\_Pulse = 0x80; 65. TIM\_OCInitStructure.TIM\_OCPolarity = TIM\_OCPolarity\_High; 66. TIM\_OCInitStructure.TIM\_OCNPolarity = TIM\_OCNPolarity\_High; 67. TIM\_OCInitStructure.TIM\_OCIdleState = TIM\_OCIdleState\_Reset; 68. TIM\_OCInitStructure.TIM\_OCNIdleState = 69. TIM\_OCIdleState\_Reset; 70. TIM\_OC3Init(BUZZER\_TIM,&TIM\_OCInitStructure); 71. TIM\_ARRPreloadConfig(BUZZER\_TIM,ENABLE); 72. /\*Timer counter enable\*/ 73. TIM\_Cmd(BUZZER\_TIM,ENABLE); 74. /\*Enables the TIM peripheral Main Outputs\*/ 75. TIM\_CtrlPWMOutputs(BUZZER\_TIM,ENABLE); |

В бесконечном цикле считываем кнопку WKUP\_BTN, проверяем, нажата ли она, и, если нажата, воспроизводим звук при помощи функции TIM\_CCxCmd():

|  |
| --- |
| 1. while(1) 2. { 3. if (Bit\_RESET != STM\_EVAL\_PBGetState(BUTTON\_WAKEUP)) 4. { 5. if(0 == pwm\_state) 6. { 7. pwm\_state = 1; 8. /\*Enabel PWM Output\*/ 9. TIM\_CCxCmd(BUZZER\_TIM,TIM\_Channel\_3,TIM\_CCx\_Enable); 10. } 11. } 12. else 13. { 14. if(1 == pwm\_state) 15. { 16. pwm\_state = 0; 17. /\*Delay\*/ 18. TIM\_CCxCmd(BUZZER\_TIM,TIM\_Channel\_3,TIM\_CCx\_Disable); 19. } 20. } 21. } |

Снимем показания с осциллографа:

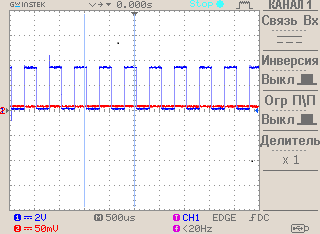


Рисунок 3 – Показания осциллографа при нажатой кнопке

## Вывод

В данной лабораторной работе были приобретены практические навыки работы с пьезоизлучателем. На практике реализовали воспроизведение звука при помощи пьезоизлучателя. Конфигурация таймера и настройки ШИМа имеют некоторые особенности: для работы с пьезоизлучателем необходимо использовать первый таймер и третий канал. Их грамотная конфигурация и является основной особенностью работы с пьезоизлучателем.