



КАФЕДРА ПРОЕКТИРОВАНИЯ  
ИНФОРМАЦИОННО-  
КОМПЬЮТЕРНЫХ СИСТЕМ



# Генерация и обработка аудио потока

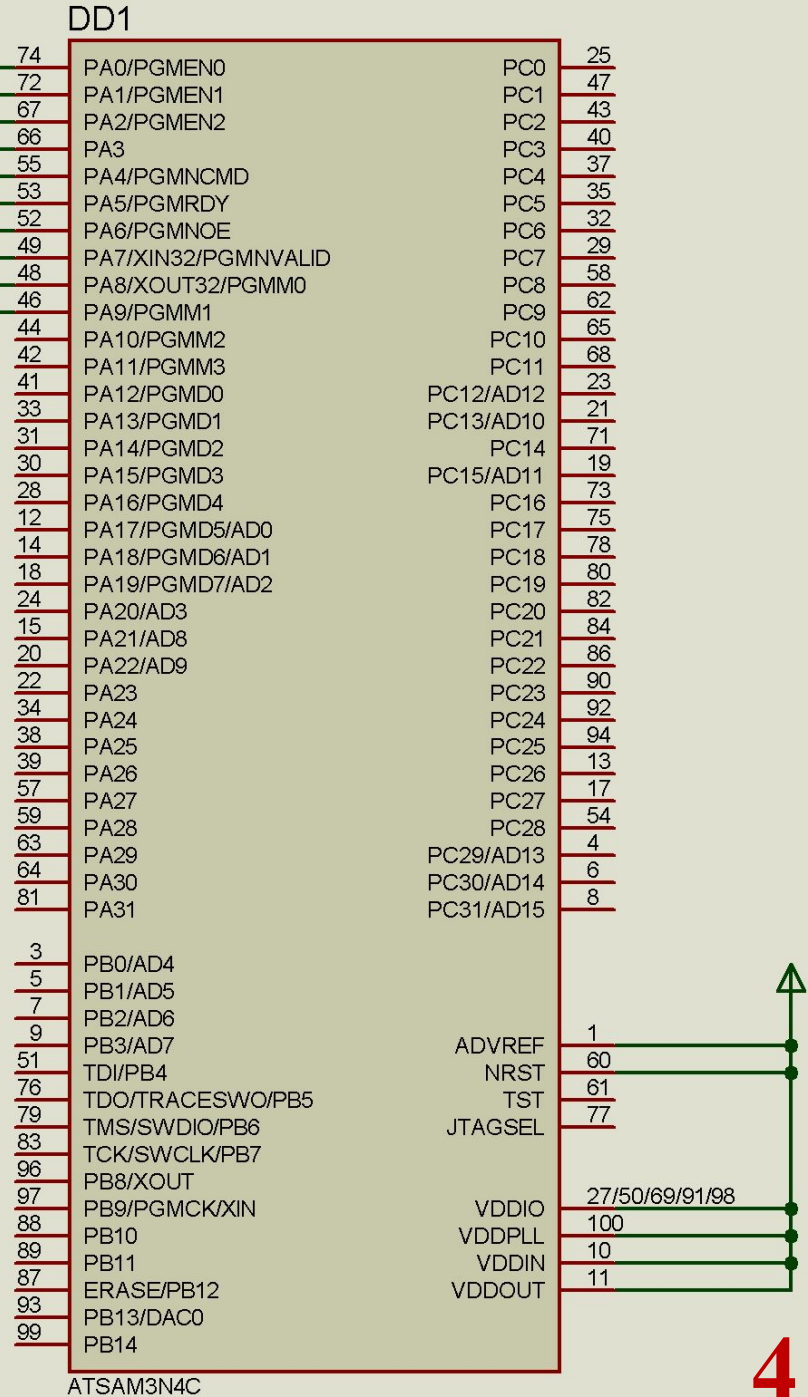
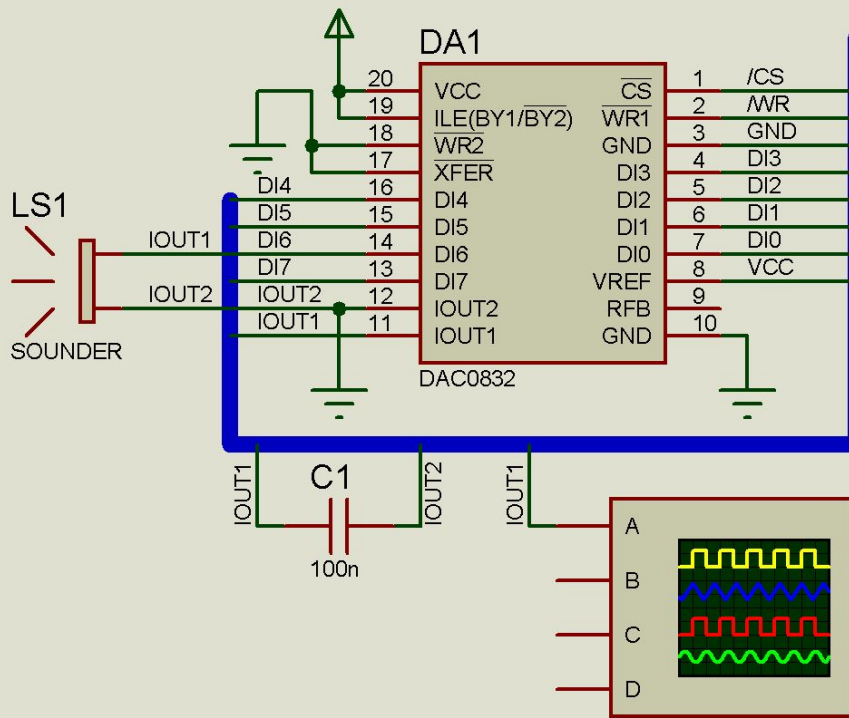
**Преподаватель:  
Олег Чеславович Ролич  
К.Т.Н., доцент**



1. Освоить принципы генерации аудио потока
2. Изучить форматы представления, хранения и передачи аудио данных
3. Изучить алгоритмы взаимного преобразования PDM-формата данных в PCM-формат



Согласно заданным принципиальной схеме и управляющей программе, на базе микроконтроллера ATSAM3N4C, цифро-аналогового преобразователя DAC0832 и генератора SOUNDER звука построить модель устройства генерации звуковых колебаний на базе PDM-фильтрации цифровых аудио данных



КАФЕДРА ПРОЕКТИРОВАНИЯ  
ИНФОРМАЦИОННО-  
КОМПЬЮТЕРНЫХ СИСТЕМ

# Схема в Proteus



Новый

проект

«*MMVP\_LW4\_ATSAM3N4C\_SOUNDER*»  
рекомендуется создавать на базе одного  
из предыдущих, ранее отлаженных,  
например, на основе  
«*MMVP\_LW3\_ATSAM3N4C\_CAMERA\_LCD\_1*»  
путём его копирования через буфер  
обмена. Лишние файлы рекомендуется  
удалить в соответствии со следующим  
рисунком



MMVP\_LW4\_ATSAM3N4C\_SOUNDER

Includes

cm3

main.c

newlib\_stubs.c

startup\_sam3n.c

flash.ld

makefile

MMVP\_LW4\_ATSAM3N4C\_SOUNDER.pdsprj

PDM+ArithmeticFT.xmcd

sam3n\_series.pdf

*Модель устройства  
генерации звуковых  
колебаний на базе PDM-  
фильтрации цифровых  
аудио данных*

**Примерный состав проекта  
MMVP\_LW4\_ATSAM3N4C\_SOUNDER**

# Модель устройства генерации звуковых колебаний



**Edit Component**

Part Reference:  Hidden:

Part Value:  Hidden:

Element:

Program File:   Hide All

Crystal Frequency:  Hide All

PCB Package:   Hide All

Advanced Properties:

Disassemble Binary Code   Hide All

Other Properties:

Exclude from Simulation  Attach hierarchy module  
 Exclude from PCB Layout  Hide common pins  
 Exclude from Bill of Materials  Edit all properties as text

## Настройки микроконтроллера

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Проект «*MMVP\_LW4\_ATSAM3N4C\_SOUNDER*»

Файлы

*newlib\_stubs.c* и *flash.ld*

остаются без изменений по  
сравнению с проектом  
«*MMVP\_LW3\_ATSAM3N4C\_CAMERA\_LCD\_1*»



# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
#include "sam3n4c.h"  
#include "pmc.h"  
#include "pio.h"  
#include "board.h"  
#define WINLEN    32 // Количество бит  
// в подвижном окне для преобразования PDM в PCM  
volatile int32_t ITM_RxBuffer;  
// DAC commands  
// PA8 <---> /CS  
#define SOUND_CS    {PIO_PA8, (AT91S_PIO *) PIOA,  
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}  
// PA9 <---> /WR  
#define SOUND_WR    {PIO_PA9, (AT91S_PIO *) PIOA,  
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
// DAC data bus
// PA0 <---> DI0
#define SOUND_DI0    {PIO_PA0, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
// PA1 <---> DI1
#define SOUND_DI1    {PIO_PA1, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
// PA2 <---> DI2
#define SOUND_DI2    {PIO_PA2, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
// PA3 <---> DI3
#define SOUND_DI3    {PIO_PA3, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
// PA4 <---> DI4
#define SOUND_DI4    {PIO_PA4, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
// PA5 <---> DI5
#define SOUND_DI5    {PIO_PA5, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
// PA6 <---> DI6
#define SOUND_DI6    {PIO_PA6, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
// PA7 <---> DI7
#define SOUND_DI7    {PIO_PA7, (AT91S_PIO *) PIOA,
AT91C_ID_PIOA, PIO_OUTPUT_0, PIO_DEFAULT}
```

# *Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных*

*Файл main.c*

```
// Sound DAC data bus pins
Pin SoundBusPins[] = {
    SOUND_DI0,
    SOUND_DI1,
    SOUND_DI2,
    SOUND_DI3,
    SOUND_DI4,
    SOUND_DI5,
    SOUND_DI6,
    SOUND_DI7,
};
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
// Sound DAC control pins
Pin SoundCtrlPins[] = {
    SOUND_CS,
    SOUND_WR,
};

uint8_t wave[] = { 127, 255, 222, 218, 170, 68, 16, 0,
    0, 1, 4, 74, 171, 111, 127, 255,
};

const uint16_t n = sizeof(wave) / sizeof(uint8_t);
const uint8_t bytesLen = WINLEN / 8 / sizeof(uint8_t);
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
/* Массив, определяющий пропорции плотности битов */  
uint8_t onesQuantity[] = {  
    0, /* количество единичных битов в числе 0 */  
    1, /* количество единичных битов в числе 1 */  
    1, /* количество единичных битов в числе 2 */  
    2, /* количество единичных битов в числе 3 */  
    1, /* количество единичных битов в числе 4 */  
    2, /* количество единичных битов в числе 5 */  
    2, /* количество единичных битов в числе 6 */  
    3, /* количество единичных битов в числе 7 */
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
1, /* количество единичных битов в числе 8 */  
2, /* количество единичных битов в числе 9 */  
 2, /* количество единичных битов в числе 10 */  
 3, /* количество единичных битов в числе 11 */  
 2, /* количество единичных битов в числе 12 */  
 3, /* количество единичных битов в числе 13 */  
 3, /* количество единичных битов в числе 14 */  
 4, /* количество единичных битов в числе 15 */  
};
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
void TC0_Handler ( void );
```

```
int main ( void ) {
```

```
    // Разрешение тактирования порта PIOA
```

```
    PMC_EnablePeripheral ( ID_PIOA );
```

```
    PIO_Configure ( SoundBusPins, PIO_LISTSIZE (  
SoundBusPins ) );
```

```
    PIO_Configure ( SoundCtrlPins, PIO_LISTSIZE (  
SoundCtrlPins ) );
```



# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

// Инициализация таймера TC0

```
TC_Configure ( TC0, TC_CMR_WAVE |  
TC_CMR_WAVSEL_UP_RC |  
TC_CMR_TCCLKS_TIMER_CLOCK1 );  
TC_Start ( TC0 );
```

```
TC0->TC_CHANNEL[0].TC_RC = 0x0040;
```

// Разрешение прерывания

// по переполнению таймера TC0

```
TC0->TC_CHANNEL[0].TC_IER = TC_IER_CPCS;
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
// Включение обработки прерываний таймера TC0
```

```
NVIC_SetPriorityGrouping ( 0 );
```

```
NVIC_EnableIRQ ( TC0_IRQn );
```

```
for(;;) {
```

```
}
```

```
return 0;
```

```
}
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
void TC0_Handler ( void ) {  
/* Преобразование PDM в PCM с заданным окном длиной WINLEN бит */  
    uint16_t i;  
    uint8_t bit = wave[0] & 0x80; // Выделение старшего бита в wave[0]  
/* Циклический сдвиг всего массива wave[] влево  
   * (в сторону старшего бита) */  
    for ( i = 0; i < n - 1; i++ ) {  
        wave[i] <<= 1;  
        if (wave[i+1] & 0x80) {  
            wave[i] |= 1; } }  
    wave[n-1] <<= 1;  
    if (bit) { wave[n-1] |= 1; }  
/* ----- */
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
/* Усреднение количества единичных битов в окне,  
 * начиная от старшего бита wave[0] длиной WINLEN */  
uint16_t onesQ = 0;  
for ( i = 0; i < bytesLen; i++ ) {  
    onesQ += onesQuantity[(wave[i] >> 4) & 0x0F];  
    onesQ += onesQuantity[wave[i] & 0x0F];  
}  
if (onesQ) {  
    onesQ <<= 8; // *= 256 (ибо ЦАП 8-разрядный)  
    onesQ--;  
    onesQ /= WINLEN; }  
}
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

Файл *main.c*

```
SoundBusPins[0].pio->PIO_SODR = onesQ & 0xFF;
SoundBusPins[0].pio->PIO_CODR = (~onesQ) & 0xFF;
    PIO_Set ( SoundCtrlPins + 1 );
    PIO_Clear ( SoundCtrlPins + 1 );
    /* ----- */

    NVIC_ClearPendingIRQ ( TC0_IRQn );
}
```

# Программа генерации звуковых колебаний на базе PDM-фильтрации цифровых данных

## Отличительные строки файла *startup\_sam3n.c*

```
static void TimingDelay_Decrement ( void );
```

```
void IntDefaultHandler ( void );
```

```
void TC0_Handler ( void );
```

```
/* Exception Table */
```

```
__attribute__ ((section(".vectors")))
```

```
IntFunc exception_table[] = {
```

```
/* Configure Initial Stack Pointer, using linker-generated symbols */
```

```
(IntFunc) (&_estack),
```

```
Reset_Handler,
```

```
IntDefaultHandler,
```

```
.....
```

```
IntDefaultHandler, /* 22 Reserved */
```

```
TC0_Handler, /* 23 Timer Counter 0 */
```

```
IntDefaultHandler, /* 24 Timer Counter 1 */
```

*Программа генерации звуковых колебаний  
на базе PDM-фильтрации цифровых данных*

## Отличительные строки make-файла

# Target file name (without extension)

**TARGET**=MMVP\_LW4\_ATSAM3N4C\_SOUNDER

.....

**SOURCE**=startup\_sam3n.c \

main.c \

newlib\_stubs.c \

\$(CM3\_SPL\_DIR)/src/pmcs.c \

\$(CM3\_SPL\_DIR)/src/pio.c \

\$(CM3\_SPL\_DIR)/src/tc.c \

\$(CM3\_SPL\_DIR)/src/nvic.c

*Файл **makefile***



КАФЕДРА ПРОЕКТИРОВАНИЯ  
ИНФОРМАЦИОННО-  
КОМПЬЮТЕРНЫХ СИСТЕМ



*Коррекция проекта  
завершена*

*Удачного его  
построения и проверки  
работы!*