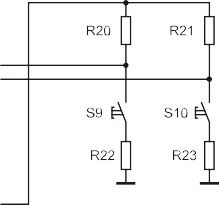
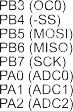
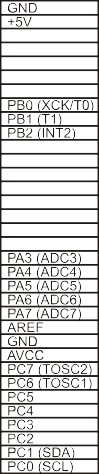
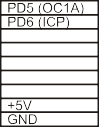
Лабораторная работа № 8



# Использование таймеров-счетчиков в режимах

**Fast PWM и Phase Correct PWM**

Цель работы:

1. Изучение принципа работы внутренних таймеров-счетчиков микро- контроллера ATmega8535.
2. Программирование таймеров-счетчиков для режимов Fast PWM

и Phase Correct PWM .

Оборудование и программное обеспечение:

1. Лабораторный макет MK8535 в составе: модуль базовый МБ8535 +

модуль сменный МС01.

1. Кабель соединительный для программирования МК.
2. Персональный компьютер.
3. Интегрированная среда программирования CodeVisionAVR.

# Часть 1. Исследование режима Fast PWM

**Подготовка к работе**

Рассмотрим и выполним пример, в котором будет использован таймер-счетчик T0 в режиме Fast PWM. Выход таймера-счетчика соеди- нён с выводом OC0 микроконтроллера (альтернативная функция вывода 3 порта B), к которому подключен светодиод VD9 (см. рис. 55).

Поскольку в этом режиме осуществляется широтно-импульсная модуляция на выводе OC0, то мы можем наблюдать изменение яркости светодиода VD9 при изменении значения OCR0. (Здесь нужно пони- мать, что светодиод – безынерционный излучатель, и ощущение яркости возникает вследствие инерционности нашего зрения, в зависимости от ширины импульсов тока). Для изменения значения OCR0 будем исполь- зовать кнопки S9 и S10. Пусть по нажатию кнопки S9 значение OCR0 будет увеличиваться, и, соответственно, будет расти ширина импульсов на выходе OC0 и яркость светодиода S9. По нажатию кнопки S10 OCR0 будет уменьшаться, а яркость светодиода – падать.

Импульсы с выхода OC0 будем наблюдать также при помощи осциллографа.

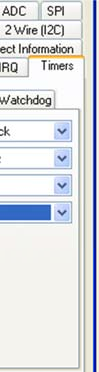
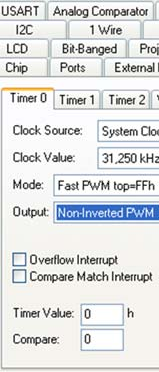
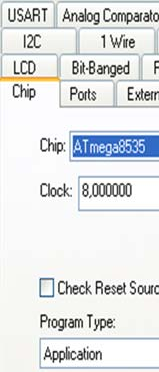
Текущее значение OCR0 будем выводить на светодиодный инди- катор VD1…VD8, подключенный к порту A (см. рис. 55).

Рис. 55. Структурная схема лабораторного макета. Для модуля МС01 показана только та часть схемы, которая используется в данной работе

Создадим проект в **CodeVisionAVR** и зададим нужный режим ра- боты МК. Порт A настроить на вывод. Линию 3 порта B настроить на вывод, чтобы разрешить прохождение сигнала от таймера на вывод OC0. Порт D настроить на ввод.

На вкладке **Timers** окна **CodeWizardAVR** (рис. 56) выберем **Timer0**, источник тактовых сигналов – тактовые импульсы МК (System Clock), тактовая частота таймера-счетчика (Clock Value) – 31,250 кГц (это 8 Мгц/256), режим работы таймера-счетчика (Mode) – Fast PWM, режим выхода OC0 – неинвертированный ШИМ (Non-inverted PWM).

**// OC0 output: Non-Inverted PWM TCCR0 = 0x6C;**



**TCNT0 = 0x00;**

**OCR0 = PORTA = 10; // Начальное значение OCR0**

**while (1)**

**{**

**// ждем нажатия кнопок S9 или S10 while ((PIND.2 != 0) && (PIND.3 != 0));**

**if (PIND.2 == 0) // если нажата кнопка S9,**

**{ // увеличение OCR0 - расширение импульса delay\_ms(20); // задержка против «дребезга» if (OCR0 <= 245) PORTA = OCR0 += 10;**

**while (PIND.2 == 0); // ждать отпускания S9 delay\_ms(20); // задержка против «дребезга»**

а) б)

Рис. 56. Выбор тактовой частоты (а) и настройка таймера-счетчика T0 (б)

Оценим частоту сигнала на выходе OC0 при таких настройках:

**}**

**if (PIND.3 == 0) //если нажата кнопка S10**

**{ // уменьшение OCR0 - сужение импульса delay\_ms(20); // задержка против «дребезга» if (OCR0 >= 10) PORTA = OCR0 -= 10;**

**while (PIND.3 == 0); // ждать отпускания S10**

*f*  *fclk* \_ *I* /*O* 

*OC N*  256

8000000 Гц

256  256

 31250 Гц  122,1 Гц.

256

# delay\_ms(20); // задержка против «дребезга»

**}**

**}**

К заготовке текста программы, сформированной **CodeWizardAVR }**

для МК ATmega8535, добавим строки, реализующие ввод с кнопок. Остальные строки, не существенные для данного примера, здесь опущены:

Поскольку таймер – это аппаратный узел, то после запуска про- граммы и установки режимов он продолжает работать самостоятельно,

# #in #in

**voi**

**{**

**O**

**a**

# lude <mega8535.h> lude <delay.h>

**main(void)**

**c**

**c**

**d**

**a**

независимо от программы. Задача программы здесь – запись новых зна- чений в регистр OCR0. На каждое нажатие кнопки значение OCR0 из- меняется с шагом 10. Видно также, что в программе предприняты меры против «дребезга контактов» кнопок.

**P RTA=0x00; DDRA=0xFF; // P RTB=0x00; DDRB=0x08; // P RTD=0x00; DDRD=0x00; //**

**O**

**O**

**Port A initi Port B initi Port D initi**

**lization lization lization**

**/ Timer/Counter 0 initialization**

**/**

**/**

**a**

**/ Clock source: System Clock**

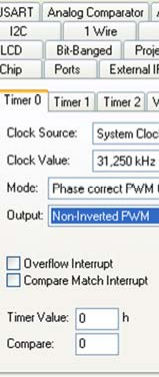
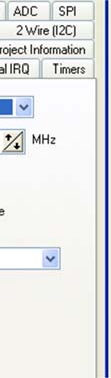
**/**

**/ Clock value: 31,250 kHz**

**/**

**/ Mode: Fast PWM top=FFh**

**Задание**



1. Создать проект и реализовать рассмотренный выше пример программы.
2. При помощи осциллографа наблюдать сигнал на выходе OC0. Какова частота импульсов и почему?
3. Наблюдать при помощи осциллографа изменение ширины импульсов на выходе OC0. Изменяется ли при этом период импульсов?
4. Изменить программу так, чтобы выход ОС0 работал в режиме

«Inverted PWM». Как изменился сигнал на выходе?

1. Изменить программу так, чтобы яркость светодиода циклически изме- нялась по линейному закону, затем – по экспоненциальному закону (чтобы скомпенсировать эффект пресыщения зрения).

# Часть 2. Исследование режима Phase Correct PWM

**Подготовка к работе**

Выполним пример, аналогичный части 1, с той лишь разницей, что таймер-счетчик T0 будет использован в режиме Phase Correct PWM.

Создадим проект в **CodeVisionAVR** и зададим нужный режим работы МК. Порт A настроить на вывод. Линию 3 порта B настроить на вывод, чтобы разрешить прохождение сигнала от таймера на вывод OC0. Порт D настроить на ввод.

На вкладке **Timers** окна **CodeWizardAVR** выберем **Timer0**, источ- ник тактовых сигналов – тактовые импульсы МК (System Clock), тактовая частота таймера-счетчика (Clock Value) – 31,250 кГц (это 8 Мгц/256), режим работы таймера-счетчика (Mode) – Phase Correct PWM, режим вы- хода OC0 – неинвертированный ШИМ (Non-inverted PWM) (см. рис. 57).

Оценим частоту сигнала на выходе OC0 при таких настройках:

а) б)

Рис. 57. Выбор тактовой частоты (а) и настройка таймера-счетчика T0 (б)

# Задание

1. Создать проект для режима Phase Correct PWM и реализовать про- граммы, аналогичные части 1.
2. При помощи осциллографа наблюдать сигнал на выходе OC0. Какова частота импульсов и почему?
3. Наблюдать при помощи осциллографа изменение ширины импульсов на выходе OC0. Изменяется ли при этом период импульсов?
4. Изменить программу так, чтобы выход ОС0 работал в режиме

«Inverted PWM». Как изменился сигнал на выходе?

*fOC*

  *fclk* \_ *I* /*O*  8000000 Гц  31250 Гц  61 Гц.

*N*  512 256  512 512

# РАБОТА С МОДУЛЕМ 7-СЕГМЕНТНОГО ЖИДКОКРИСТАЛЛИЧЕСКОГО ИНДИКАТОРА

Внешние устройства, выполненные в виде законченного функци- онального модуля с собственным интерфейсом, могут быть подключены к выводам портов МК. Для обеспечения обмена данными придется вы- полнить правила, предписываемые интерфейсом модуля, т.е. программ- ным способом сформировать на выводах порта МК нужные комбинации сигналов в соответствии с временными диаграммами.

# Жидкокристаллический модуль МТ10Т7-7

Жидкокристаллический модуль МТ10Т7-7 состоит из БИС кон- троллера и ЖК-панели. Модуль может отображать 10 знакомест (цифр с точкой). Внешний вид приведен на рис. 58.

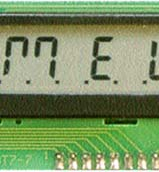


Рис. 58. Внешний вид модуля МТ10Т7-7

Любой сегмент любого знакоместа можно включать и выключать независимо от остальных сегментов. Структурная схема модуля пред- ставлена на рис. 59. Регистры данных в БИС делятся на две тетрады: SGx(L) и SGx(H). Запись данных в знакоместо производится за два так-

резистора на вывод V0 (рис. 59). *R*внеш = 0 – максимальная контраст- ность, *R*внеш = ∞ (нет резистора) – минимальная контрастность.

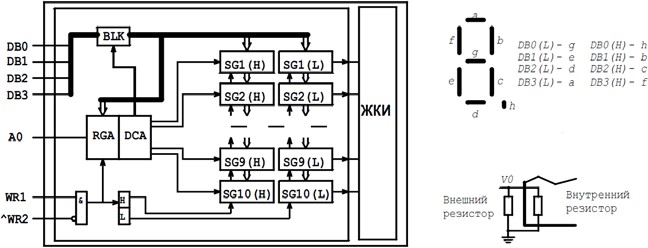


Рис. 59. Структурная схема модуля МТ10Т7-7

# Описание интерфейса модуля МТ10Т7-7

Сначала на шине выставляется адрес необходимого знакоместа, который фиксируется в регистре адреса при низком уровне на входе А0 соответствующим сигналом на входе WRx (см. табл. 20). Входы WR1 и ^WR2 защелкивают информацию, стоящую на шине, во внутренних регистрах статического типа. Внутри БИС эти входы объединены по схеме WR1 & ^WR2. Таким образом, информация запишется только при WR1 = 1 и ^WR2 = 0 одновременно. Такое решение позволяет осу- ществлять функцию CS (выбор кристалла) при большом количестве мо- дулей на шине, или если на шине имеются другие устройства.

*Таблица 20*

Сигналы интерфейса МТ10Т7-7

та: сначала в младшую тетраду, затем в старшую. Младшая тетрада

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сигнал | Запись  в регистр адреса | | Запись  в регистр данных | | Хранение | |
| A0 | 0 | | 1 | | x | |
| ^WR2 | 0 | 1 | 0 | 1 | 1 | x |
| WR1 | 1 | 0 | 1 | 0 | x | 0 |
| DB0..3 | Адрес | | Данные | | x | |

отвечает за сегменты *g*, *e*, *d*, *a*, а старшая – за сегменты *h*, *b*, *c*, *f*. Запись

«1» вызывает высвечивание соответствующего сегмента, запись «0»

вызывает его гашение.

Контрастность индикатора зависит от напряжения питания модуля. Управление контрастностью производится подключением внешнего

Временные диаграммы записи приведены на рис. 60. При записи адреса знакоместа указатель тетрады сбрасывается в положение SGx(L). Запись данных производится в младшую тетраду при высоком уровне сигнала на входе А0 сигналом на входе WRx. По этому же сигналу ука- затель тетрады данных переключается в положение SGx(H), сохраняя при этом тот же адрес знакоместа. Данные в старшую тетраду SGx(H) записываются аналогично младшей тетраде SGx(L). После записи вто- рой тетрады содержимое регистра адреса инкрементируется и можно записывать данные в следующее знакоместо без записи адреса.

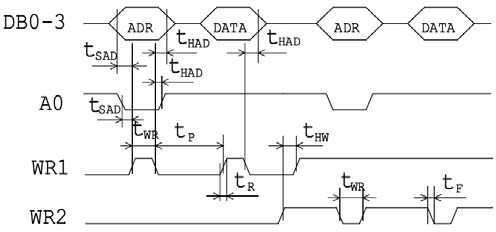
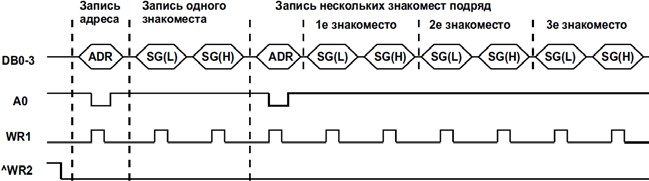


Рис. 61. Временные характеристики интерфейса МТ10Т7-7

*Таблица 22*

Динамические характеристики модуля

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Обозн. | min | max |
| Время предустановки, нс | tSAD | 0 | - |
| Время удержания, нс | tHAD | 100 | - |
| Длительность сигнала записи, нс | tWR | 100 | - |
| Пауза между WR, нс | tP | 200 | - |
| Задержка изменения сигналов WRx, нс | tHW | 50 | - |
| Время нарастания импульса, нс | tR | 0 | 50 |
| Время убывания импульса, нс | tF | 0 | 50 |

Рис. 60. Временные диаграммы записи

Адреса регистров модуля МТ10Т7-7 приведены в табл. 21. По ад- ресу 0Fh расположен триггер блокировки шины. Запись в него DB0 = 0 вызывает блокировку записи в БИС адресов и данных на 30 сигналов WRx. Разблокировка шины производится записью DB0 = 1 по адресу 0Fh.

*Таблица 21*

Адреса регистров модуля МТ10Т7-7

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № знакоместа | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  |
| Наименование | Sg1 | Sg2 | Sg3 | Sg4 | Sg5 | Sg6 | Sg7 | Sg8 | Sg9 | Sg10 | Блокировка |
| Адрес (hex) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0F |

После подачи питания содержимое регистров SGx не определено, поэтому при включении питания необходимо делать программную очистку регистров. Состояние триггера блокировки тоже не определено, поэтому перед началом вывода информации на индикатор необходимо произвести разблокировку шины. Разблокировка шины производится записью DB0 = 1 по адресу 0Fh.

Временные характеристики интерфейса МТ10Т7-7 приведены на рис. 61 и в табл. 22.

Назначение внешних выводов модуля МТ10Т7-7 описано в табл. 23.

*Таблица 23*

Назначение внешних выводов модуля МТ10Т7-7

|  |  |  |
| --- | --- | --- |
| № | Наименование | Назначение |
| 1 | A0 | Выбор адрес/данные: A0 = 0 – адрес, A0 = 1 – данные |
| 2 | ^WR2 | Запись в модуль. Активный уровень – низкий |
| 3 | WR1 | Запись в модуль. Активный уровень – высокий |
| 4 | DB3 | Шина адреса/данных |
| 5 | DB2 | Шина адреса/данных |
| 6 | DB1 | Шина адреса/данных |
| 7 | DB0 | Шина адреса/данных |
| 8 | GND | Общий |
| 9 | V0 | Управление контрастностью |
| 10 | +E | Питание модуля |