**Лабораторная работа №9. Таймеры AVR Atmega 328P.**

**Формирование импульсных сигналов**

**Цель:** изучение особенностей применения таймеров в микропроцессорных системах.

**Оборудование на 1 рабочее место:**

Компьютер с ОС Windows 1шт

Отладочная плата МК AVR 1 шт.

Программатор USBISP 1 шт.

Логический анализатор 1 шт.

**Используемое ПО**:

Интегрированная среда разработки Atmel Studio 7.0 (или AVR Studio 4.19)

Программа для загрузки программного кода в микроконтроллер AVRDUDEPROG

Программа для анализа временных диаграмм Saleae Logic

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

См. содержание теоретической части практической работы №8.

**Сервоприводы**

Благодаря ШИМ можно управлять работой различных приводов. Сервопривод – это вид привода, который может точно управлять параметрами движения, а именно повернуть свой вал на определенный угол или поддерживать непрерывное вращение. Сервоприводы обладают стабильной работой, высокой устойчивостью к помехам, малыми габаритами и широким диапазоном контроля скорости. Важными особенностями сервоприводов являются способность увеличивать мощность и обеспечение обратной связи.

Включая или выключая обычный электрический двигатель, можно сформировать вращательное движение и заставить двигаться колеса или другие предметы, прикрепленные к валу. Движение будет непрерывным, но для того, чтобы понять, на какой угол повернулся вал или сколько оборотов он сделал, потребуется устанавливать дополнительные внешние элементы. Сервопривод уже содержит все необходимое для получения информации о текущих параметрах вращения и может самостоятельно выключаться, когда вал повернется на необходимый угол.

Важным отличием серводвигателя от шагового двигателя является возможность работать с большими ускорениями и при переменной нагрузке. Также серводвигатели обладают более высокой мощностью. Шаговые двигатели не обладают обратной связью, поэтому может наблюдаться эффект потери шагов, в серводвигателях потери шагов исключены – все нарушения будут зафиксированы и исправлены. При всех этих явных преимуществах серводвигатели являются более дорогостоящими приборами, чем шаговые, обладают более сложной системой подключения и управления и требуют более квалифицированного обслуживания. Важно отметить, что шаговые двигатели и сервоприводами не являются прямыми конкурентами – каждое из этих устройств занимает свою определенную сферу применения с чным периодом.

**Сервопривод SG90**

Сервопривод SG90 является одним из самых простых устройств такого типа. Широко используется в робототехнике и учебных задачах.

**Особенности:** Диапазон вращения: 180°; Напряжение питания: 4.8…6В; Крутящий момент: 1.8 кг.см при 4.8 В; Скорость вращения: 60° за 0.12 сек при 4.8 В; Внутренний интерфейс: аналоговый.

Особенности управления: для управления сервоприводом необходимо поддерживать ШИМ с периодом 20мс. Управление поворотом задаётся длинной импульса в диапазоне от 1 до 2мс.

**ПРАКТИЧЕСКАЯ ЧАСТЬ**

**Задание 1.** Выполните анализ представленных программ, загрузите их поочередно в микроконтроллер. Комментарии к коду представлены в тексте программы. Выполните подключение согласно рисунку 9.1, проанализируйте работу с помощью программы Logic. Определите параметры сигнала по временной диаграмме из программы Logic и путем расчета параметров таймера. Сравните результаты.

Код на языке Assembler

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | .include "m328Pdef.inc" ;библиотека с логическими именами портов ввода/вывода  .def tmp = r16 ;задание логического имени tmp - регистру r16  start: ;метка входу в программу  ldi tmp, 0x40 ;настройка PORTD6 (OC0A) на вывод информации  out DDRD, tmp ;  ldi tmp, 0x83 ;настройка режим PWM с использованием OC0A (не инверсный)  out TCCR0A,tmp ;  ldi tmp, 0x0A ; сравнение со значением 0x0a, длина «1» 10 отсчётов, длина «0» 246 отсчётов  out OCR0A,tmp ;  ldi tmp, 0x05 ;запуск таймера, предделитель clk/1024  out TCCR0B,tmp ;  p: ;метка для организации бесконечного цикла  jmp p ;безусловный переход к метке p |

Код на языке C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #include <avr/io.h>//библиотека с логическими именами портов ввода/вывода  int main(void) //главная процедура, точка входа в программу  {  DDRD = 0x40; //настройка PORTD6 (OC0A) на вывод  //Настройка TC0 в режиме ШИМ на выводе OC0A (не инверсный режим)  TCCR0A = (1<<COM0A1)|(0<<COM0A0)|(1<<WGM01)|(1<<WGM00);  //сравнение со значением 0x0a, длина «1» 10 отсчётов, длина «0» 246 отсчётов  OCR0A = 0x0A;  //запуск таймера с частотой Fclk/1024  TCCR0B = (1<<CS02)|(0<<CS01)|(1<<CS00);  while (1) //бесконечный цикл  {  }} |



Рисунок 9.1 Подключение логического анализатора для заданий 1,2

**Задание 2.** В задании 1 измените режим работы таймера на ШИМ с коррекцией фазы и выполните проверку работы с помощью логического анализатора. Подключение выполните согласно рисунку 9.1. Зафиксируйте результаты в отчете в виде снимка экрана. Отметьте, как изменился сигнал после смены режима.

**Задание 3.** Для формирования частотной модуляции необходимо обеспечить изменение выхода OCxA или OCxB на противоположное значение, через заданный интервал времени. Для этого можно в режиме СТС (или при необходимости других) настроить изменение выхода при совпадении с регистрами сравнения OCRxA (или OCRxB) с помощью битов COMxA[1:0] (COMxB[1:0]).

Соответствие выводов таймера представлено в таблице 9.1

Таблица 9.1 – выводы таймеров

|  |  |  |
| --- | --- | --- |
| Вывод таймера | Порт | Вывод отладочной платы |
| OC0A | PD6 | 6 |
| OC0B | PD5 | 5 |
| OC1A | PB1 | 9 |
| OC1B | PB2 | 10 |
| OC2A | PB3 | 11 |
| OC2B | PD3 | 3 |

Разработайте программу, используя проекты из заданий 1,2 текущей работы и задания из практической работы 8. Программа должна обеспечивать формирование частотной модуляции по следующему правилу:

1КГц – 2КГц – 3КГц – 1КГц – 2КГц – 3КГц – …

Интервал на поддержание частоты 250мс.

Выполните сборку программы. Загрузите программу в контроллер. Проанализируйте работу контроллера с использованием логического анализатора.

**Задание 4.** Разработайте программу для управления сервоприводом SG90. Программа должна обеспечивать поворот привода по следующему правилу.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Порядок | п-ц-л | п-ц-л | п-л-ц | п-л-ц | п-ц-л | п-ц-л | п-л-ц | п-л-ц | л-ц-п | л-п-ц |
| Задержка,мс | 700 | 700 | 700 | 700 | 1500 | 1500 | 1500 | 1500 | 1000 | 1000 |
| Timer | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 |

П – крайнее правое состояние

Ц – центральное состояние

Л – крайнее левое состояние

После завершения указанной последовательности состояний, привод возвращается в начальное состояние и цикл повторяется.

Задержка – время пребывания в конкретном состоянии (Timer1)

Выполните сборку программы. Загрузите программу в контроллер. Проанализируйте работу контроллера с использованием логического анализатора. Проанализируйте работу устройства с помощью сервопривода, при наличии.

**Задание 5.** Управление сервоприводом по UART

Измените программу из задания 5 таким образом, чтобы сервопривод управлялся по интерфейсу UART. Из терминала на контроллер должно поступать значение угла поворота от 0 до 180 градусов, МК при этом должен обеспечивать соответствующий поворот привода. При получении значений, выходящих за диапазон, привод должен сохранять текущее состояние.

Выполните сборку программы. Загрузите программу в контроллер. Проанализируйте работу контроллера с использованием логического анализатора. Проанализируйте работу устройства с помощью сервопривода, при наличии.