

АМПЛИТУДНО-ЧАСТОТНОЕ УПРАВЛЕНИЕ АСИНХРОННЫМ ЭЛЕКТРОДВИГАТЕЛЕМ ¹

*С.В. Каверин, к.ф.-м.н., доцент, И.А. Каверина, к.ф.-м.н., доцент
ОАНО ВПО «Волжский университет имени В.Н.Татищева»
г. Тольятти, Россия*

Для некоторых, относительно простых задач управления электроприводом можно использовать неспециализированные микроконтроллеры, к которым разработчик привык, и которые свободно распространяются на рынке. В работе [1] изложены основные требования к микроконтроллерам и описан круг современных задач, где применение специализированных микросхем является наиболее разумным решением. За счет высокой выносливости, надежности, низкой стоимости и высокого к.п.д. асинхронные электродвигатели используются во многих промышленных приложениях. Однако недостатком асинхронных двигателей является работа только на номинальной скорости при подключении к сети. Это является причиной необходимости использования преобразователей частоты для регулировки частоты вращения этих электродвигателей. Наиболее популярным алгоритмом управления асинхронным трехфазным электродвигателем (АТД) является алгоритм с поддержанием постоянства отношения напряжение/частота ($V/f=\text{const}$) [2]. В настоящей статье мы описываем концептуальную возможность программного решения простой задачи управления трехфазным асинхронным двигателем (инвертора) на основе недорогого однокристального микроконтроллера ATmega128.

Микроконтроллер ATmega128 содержит два 16-разрядных таймер-счетчика (таймеры 1 и 3), которые предназначены для точного задания временных интервалов и генерации прямоугольных импульсов. Реализацию ШИМ генератора осуществляем аппаратно. В таймере 3 у нас целых три синхронизированных канала, которые инициализируются при помощи вызова функции `init_TC3()`.

// Код инициализации таймер счетчика 3

¹ Статья опубликована в материалах IX международной научно-практической конференции. Татищевские чтения: актуальные проблемы науки и практики. - Тольятти: Издательство Волжского университета им. В.Н. Татищева, 2012

```

void init_TC3() {
    TCCR3B |=(1<<CS30); // Тактировать от основного генератора
    SREG|=(1<<7); // Глобальное разрешение прерываний.
    ETIMSK|=(1<<TOIE3); // Разрешить прерывание по переполнению ТЗ.
    // Число тактов таймера до переполнения = 0x2710
    TCNT3H=0xFF-0x27; TCNT3L=0xFF-0x10;
    // Выбор режима – быстрый ШИМ
    TCCR3A|=(1<<WGM30)|(1<<WGM31);
    TCCR3B|=(1<<WGM32)|(1<<WGM33);
    // Генерировать ШИМ на выводах OC3x без инверсии выхода
    TCCR3A|=(1<<COM3A1)|(1<<COM3B1)|(1<<COM3B1);
} // Конец инициализации ТЗ

// Обработчик прерывания по переполнению ТЗ.
// необходимо подключить библиотеку #include <avr/signal.h>
SIGNAL (SIG_OVERFLOW3)
{ // Периодическая инициализация счётчика
    TCNT3H=0xFF-0x27; TCNT3L=0xFF-0x10+3;
}

```

Разрабатываемая система управления работает от трехфазной сети 380В и позволяет плавно менять обороты двигателя. При выборе микросхемы управления полным 3-фазным мостом следует обратить внимание на наиболее интересные изделия в гамме драйверов IR. Это микросхемы серий IR213* на напряжение 600В и IR223* на напряжение 1200В. Драйверы в такой конфигурации довольно редко встречаются у других производителей. Для управления транзисторами в схемах, рассчитанных на питание от выпрямленного напряжения 380В, International Rectifier выпускает драйверы серии IR22**.

В качестве силовых ключей можно выбрать транзисторы IRGB4045DPBF или им подобные.

Функциональная схема сопряжения микроконтроллера ATmega128 с трехфазным мостовым драйвером IR2237 приведена на рисунке 1. Здесь подразумевается, что ШИМ-сигналы формируются аппаратно таймером 3 и результат подается на ножки PE3, PE4 и PE4. Логические элементы «AND», подают этот сигнал на соответствующую ножку драйвера силовых ключей, тем самым обеспечивают включение и отключение ключей. Таким образом, изменение частоты вращения ротора асинхронного трехфазного двигателя

достигается изменением частоты коммутации верхних и нижних ключей с одновременным изменением питающего напряжения $V_{пит}$ так, что соотношение между амплитудой и частотой напряжения оставалось постоянным. Изменение амплитуды $V_{пит}$ обеспечивается с помощью ШИМ.

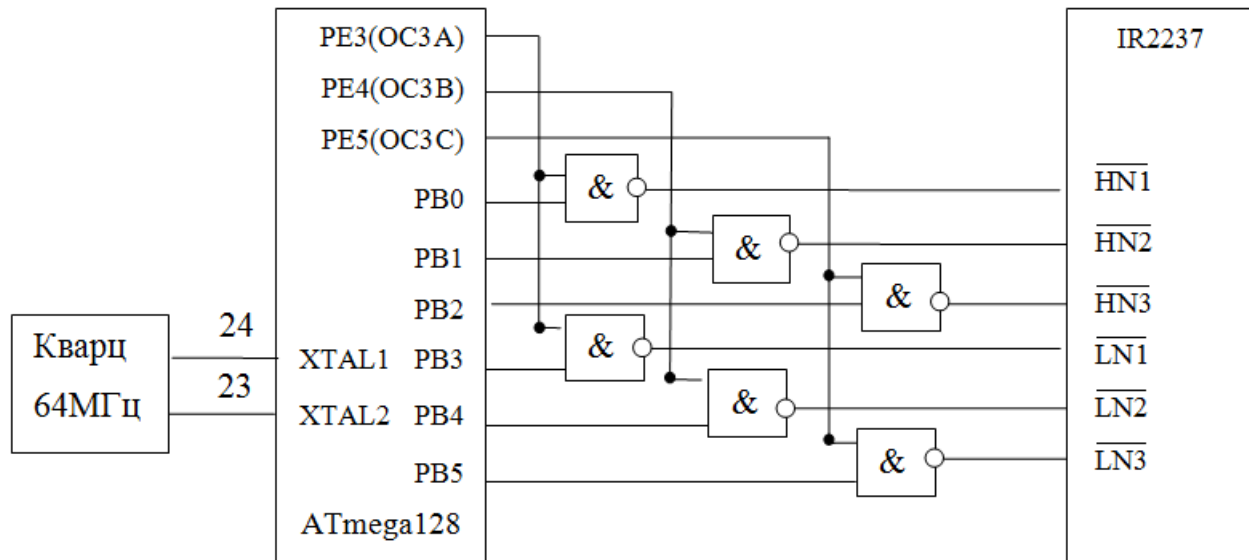


Рисунок 1 – Схема сопряжения ATmega128 с драйвером IR2237

Для управления АД необходимо сформировать вращающееся электромагнитное поле, управляя набором ключей К1-К6 – транзисторов IRGB (рисунок 2), преобразующих постоянное напряжение $V_{пит}$ в трехфазное напряжение переменного тока (рисунок 3). Изменение частоты вращения ротора АД достигается изменением частоты коммутации ключей К1-К6 силового блока с одновременным изменением $V_{пит}$ так, чтобы соотношение между амплитудой и частотой напряжения оставалось постоянным. Изменение амплитуды $V_{пит}$ обеспечивается с помощью ШИМ. В диаграмме 3 показаны уровни напряжений между началами обмоток.

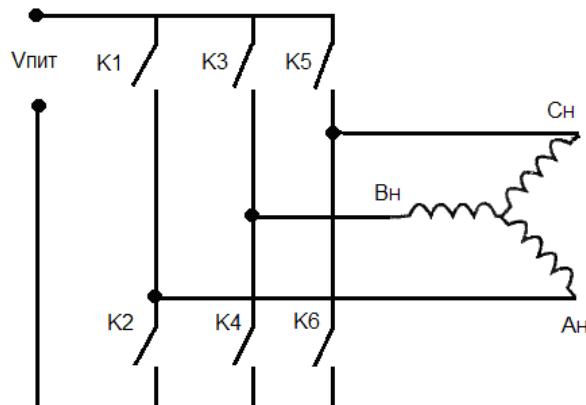


Рисунок 2 – Структура работы блока силовых ключей

Коды коммутации в режиме работы по часовой стрелке содержит триады: 101–100–110–010–011–001. Реверсное движение осуществляется в обратном порядке.

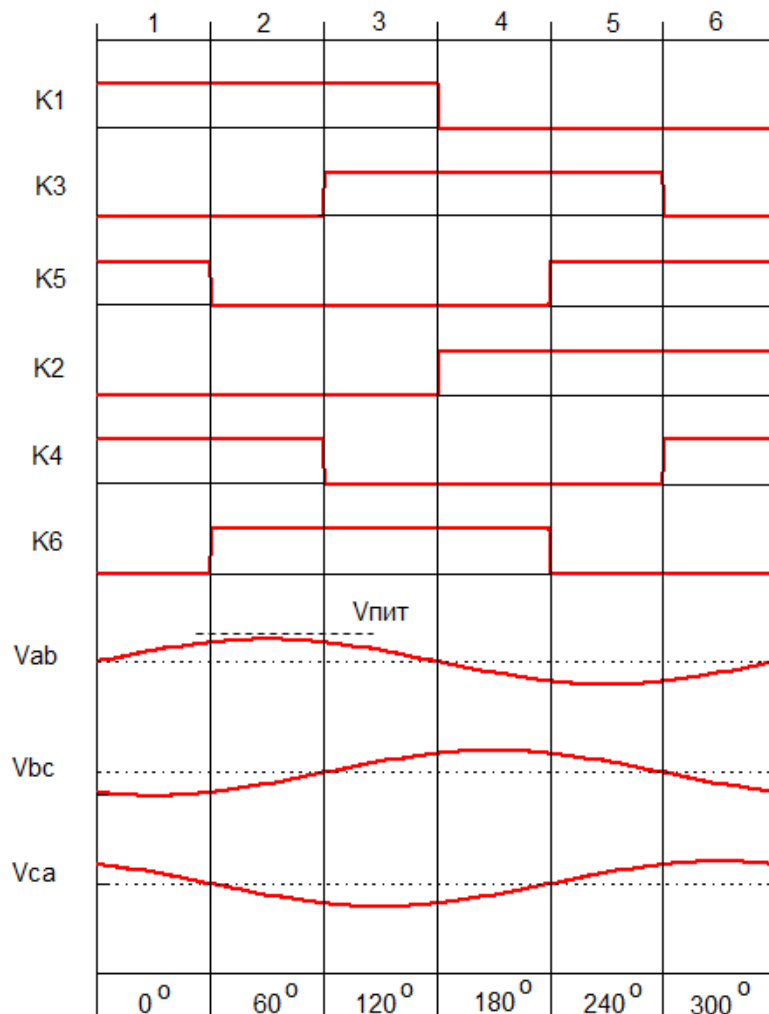


Рисунок 3 – Временные диаграммы работы блока ключей

Фрагмент функции, которая генерирует переключение силовых ключей:

```
void switch_kluch() {
/* коммутации в режиме работы по часовой стрелке */
if ((PINB.0==1)&& (PINB.1==0)&& (PINB.2==1)) { // был код 101
    PORTB |= (1<<0)|(1<<4)|(1<<5); // запущен код 100
    PORTB &= ~(1<<1)&~(1<<2)&~ (1<<3);
}
if ((PINB.0==1)&& (PINB.1==0)&& (PINB.2==0)) { // был код 100
    PORTB |= (1<<0)|(1<<1)|(1<<5); // запущен код 110
    PORTB &= ~(1<<2)&~(1<<3)&~ (1<<4);
}
..... // и т.д.
```

}

Одним из способов решения задачи формирования с помощью инвертора трехфазной синусоидальной системы напряжений со сдвигом по фазе 120 градусов на обмотках статора является использование таблицы синусов. В этом случае частота статора определяет три дискретных времени интеграторов, которые вычисляют мгновенные значения фаз для каждого статорного напряжения:

$$\begin{aligned}\theta_1[k] &= \theta_1[k-1] + \omega_s[k] \cdot T_s, \\ \theta_2[k] &= \theta_2[k-1] + \omega_s[k] \cdot T_s, \\ \theta_3[k] &= \theta_3[k-1] + \omega_s[k] \cdot T_s,\end{aligned}$$

где $\theta_1[0]=0$, $\theta_2[0]=-2\pi/3$, $\theta_3[0]=-4\pi/3$ а T_s – период дискретизации.

Если одно из этих значений становится больше 2π , то для поддержания области значений в диапазоне от 0 до 2π из результирующего значения вычитается 2π . Таблица синусов используется для вычисления трех напряжений, которые необходимо приложить к статору:

$$\begin{aligned}V_a[x] &= V_{sm}(\omega_s[k]) \cdot \sin(\theta_1[k]) \\ V_b[x] &= V_{sm}(\omega_s[k]) \cdot \sin(\theta_2[k]) \\ V_c[x] &= V_{sm}(\omega_s[k]) \cdot \sin(\theta_3[k])\end{aligned}\tag{1}$$

где $V_{sm}(\omega_s[k])$ – амплитуда напряжения статора, определенная по принципу постоянства отношения напряжение-частота.

Обычное ШИМ-управление подразумевает использование таблицы синусов для всех значений аргумента от 0 до 2π . Используя свойства тригонометрических функций, можно сократить размер таблицы преобразования. Наиболее простым и эффективным способом является использование таблицы преобразования со значениями синусов в диапазоне только от 0 до $\pi/3$, т.к.

$$\begin{aligned}\sin(q) &= \sin(q - \pi/3) + \sin(2\pi/3 - q) \text{ для } q \text{ между } \pi/3 \text{ и } 2\pi/3; \\ \sin(q) &= \sin(\pi - q) \text{ для } q \text{ между } 2\pi/3 \text{ и } \pi; \\ \sin(q) &= -\sin(q - \pi) \text{ для } q \text{ между } \pi \text{ и } 4\pi/3; \\ \sin(q) &= -\sin(q - 4\pi/3) + \sin(5\pi/3 - q) \text{ для } q \text{ между } 4\pi/3 \text{ и } 5\pi/3; \\ \sin(q) &= -\sin(2\pi - q) \text{ для } q \text{ между } 5\pi/3 \text{ и } 2\pi.\end{aligned}$$

Программная реализация данных функциональных преобразований (1) приведена ниже, где используется массив синусов `tab_sin[120]` – 120 точек синуса в диапазоне от 0 до $\pi/2$ умноженное на число 127 и округленное.

```

//////// Вычисление изменения напряжений в цикле от 0 до  $2\pi$ 
unsigned int duty_cycle(unsigned int theta, unsigned int Vm) {
    signed char  sinus ;
    unsigned int Vmxsin ;
    if (theta <= 120)          // x in [0;pi/2], т.к. по 120 точкам таблицы
        sinus = tab_sin[theta];    // sin(x)
    else {
        if (theta <= 240)      // x in [pi/2;pi]
            sinus = tab_sin[240-theta];    // sin(pi-x)
        else {
            if (theta<=360)    // x in [pi;3*pi/2]
                sinus = tab_sin[theta-240];    // sin(x-pi)
            else                // x in [3*pi/2;2*pi]
                sinus = tab_sin[480-theta]; // sin(2*pi-x)
        }
    }
    Vmxsin = (sinus * Vm)/128 ;
    return ((unsigned int) (Vmxsin));
}

```

Управление скоростью вращения ротора осуществляется через задание частоты питающей двигатель сети f . При питании обмотки статора трёхфазным током создаётся вращающееся магнитное поле, синхронная частота вращения n [об/мин] которого связана с частотой сети f [Гц] соотношением:

$$n = \frac{60 \cdot f}{p}, \text{ где } p - \text{число пар магнитных полюсов обмотки статора.}$$

Считаем, что асинхронный двигатель имеет число полюсов $p = 4$. Так как при прохождении одной синусоиды происходит 6 переключений силовых ключей, то при частоте сети f число переключений в секунду: $N_{switch} = 6 \cdot p \cdot f = 24 \cdot f$. Синусоида от переключения до следующего переключения проходит 80 точек из таблицы синусов, обеспечивая при этом 80 ШИМ-сигналов. Таким образом, получаем, что в секунду количество широтно-импульсных модуляций равно $N_{pwm} = 80 \cdot 24 \cdot f = 1920 \cdot f$.

Таймер 3 работает от внешнего кварцевого резонатора частотой 64 МГц, поэтому количество тактов для одного ШИМ:

$$N_{tact} = \frac{64\,000\,000}{1920 \cdot f} = \frac{33333}{f}. \quad (2)$$

Таймер счетчик 3, до переполнения (прерывания), отсчитывает такты от 0 до 0xFFFF, что составляет 65535 тактов. Чтобы привести количество тактов к нужному числу, рассчитанному по формуле (2), необходимо в регистр TCNT3 внести число $Begin = 65535 - N_{tact}$, тогда отсчет начнется не от 0, а от числа $Begin$. Регистр TCNT3 состоит из двух регистров TCNT3H и TCNT3L, поэтому имеем $TCNT3L = 0xFF - ((Begin \ll 8) \gg 8)$; $TCNT3H = 0xFF - (Begin \gg 8)$.

Чтобы обеспечить выход логической 1, соответствующей напряжению $V = 7 \cdot f$, положим в регистр OCR3x (x – это фазы А, В, С), число

$$OCR3x = TCNT3 + \frac{V_x}{V_{НОМ}} \cdot N_{tact} = TCNT3 + \frac{V_x}{380} \cdot N_{tact}. \quad (3)$$

Ориентировочная реализация данного подхода приводится ниже.

```
#define MAX_THETAx4 480 // число точек синуса от 0 до 2π
int f=0, V=0;          // заданная частота и напряжение V=7*f
int flagP=0, flagS=0;  // Флаги пуска и останова двигателя
unsigned int Begin;    // переменная для расчета начала регистра TCNT3
unsigned int X;        // переменная для промежуточных расчетов
int chet=0,            // счетчик переключений
// начальные значения фаз для каждого статорного напряжения
unsigned int theta1=0, theta2=160, theta3=320;
void main(void) {
    /* Блок инициализации периферии и портов (опущен) */
    init_TC3() // Инициализация Таймера 3
    while(flagP==0) {;} // Ожидания нажатия кнопки Пуск ( flagP=1)
    PORTB |= (1<<0)|(1<<2)|(1<<4); // Запуск двигателя
    PORTB &= ~(1<<1)&~(1<<3)&~(1<<5); // запущен 101010
    while(flagS==0) { // Цикл до нажатия кнопки Стоп (flagS=1)
        // Расчет параметров обычной ШИМ
        V=7*f;          // Расчет напряжения, f – частота
        Begin=65535 - 33333/f; // начало отсчета регистра TCNT3
```

```

// Расчет точек на синусоиде для каждой фазы A, B, C
if (theta1>=MAX_THETAx4) theta1 -= MAX_THETAx4;
if (theta2>=MAX_THETAx4) theta2 -= MAX_THETAx4;
if (theta3>=MAX_THETAx4) theta3 -= MAX_THETAx4;
Va= duty_cycle(theta1, V); // напряжение фазы A
Vb= duty_cycle(theta1, V); // напряжение фазы B
Vc= duty_cycle(theta1, V); // напряжение фазы C
}
}

/*****/
// Обработчик прерываний Таймера 3
SIGNAL (SIG_OVERFLOW3) {
    // Задание ШИМ для заданных напряжений фаз A, B, C
    X=Va*88/f;      // см. формулу (3)
    OCR3AL = TCNT3L + ((X<<8)>>8); OCR3AH = TCNT3L + (X>>8);
    X=Vb*88/f;      // см. формулу (3)
    OCR3BL = TCNT3L + ((X<<8)>>8); OCR3BH = TCNT3L + (X>>8);
    X=Vc*88/f;      // см. формулу (3)
    OCR3CL = TCNT3L + ((X<<8)>>8); OCR3CH = TCNT3L + (X>>8);
    if (chet==80); { // номер точки после последнего переключения
        chet= 0;
        switch_kluch(); // переключение силовых ключей
    }
    else chet= chet +1;
}

```

Библиографический список

1. В. Козаченко. Основные тенденции развития встроенных систем управления двигателями и требования к контроллерам. // CHIP NEWS. – 1999.-№1.-С.2-9.
2. Рекомендации по применению Atmel AVR494: Управление асинхронным электродвигателем переменного тока по принципу постоянства V/f и обычного ШИМ-управления. // www.gaw.ru