Программная реализация импульсной модуляции сигналов регулирования

Олег Вальпа

Приведено описание программного способа импульсной модуляции сигналов для автоматических систем регулирования на примере программ, разработанных автором статьи.

Дополнительные материалы к этой статье можно скачать, перейдя по ссылке в QR-коде

Введение

Импульсная модуляция является одним из самых распространённых способов регулирования в автоматических системах. Поскольку импульсная модуляция применяется для ключевых режимов регулирования, она позволяет получить высокий коэффициент полезного действия и часто является единственно возможным способом регулирования. Это относится, например, к регуляторам мощных токовых установок или нагревательного оборудования. Многие микропроцессоры имеют встроенные аппаратные средства широтно-импульсной модуляции (ШИМ). Но в случае нехватки количества таких средств или их полном отсутствии можно воспользоваться программным способом импульсной модуляции. Данный способ позволяет сформировать не только необходимое количество каналов регулирования, но и задать вид модуляции.

Широтно-импульсная модуляция

Рассмотрим несколько способов программной реализации импульсной модуляции регулирующего сигнала на основе конкретных примеров программ. Одним из самых распространённых видов модуляции является широтно-импульсная модуляция. Для её реализации программным путём потребуется один таймер для формирования регулярных отсчётов времени и несколько регистров. Пример такой программы на языке С с подробными комментариями для восьмиканального регулятора приведён в листинге 1.

Листинг 1. Программа широтно-импульсной модуляции

```
#include <iostream> // Подключение библиотеки ввода-вывода
#include <string> // Подключение строковой библиотеки
#include <math.h> // Подключение математической библиотеки
unsigned short n, k; // Индексные регистры программы
unsigned short pwm_c=0; // Регистр счётчика импульсов от 0 до 100%
unsigned short pwm_r[8]; // Регистры регуляторы
unsigned short pwm_do[]={0,0,0,0,0,0,0,0}; // Регистры выходных сигналов
// Функция обработчика прерывания таймера
int irq_tim()
{
unsigned short i; // Индексный регистр прерывания ШИМ
for(i=0;i<=7;i++) // Организация
```

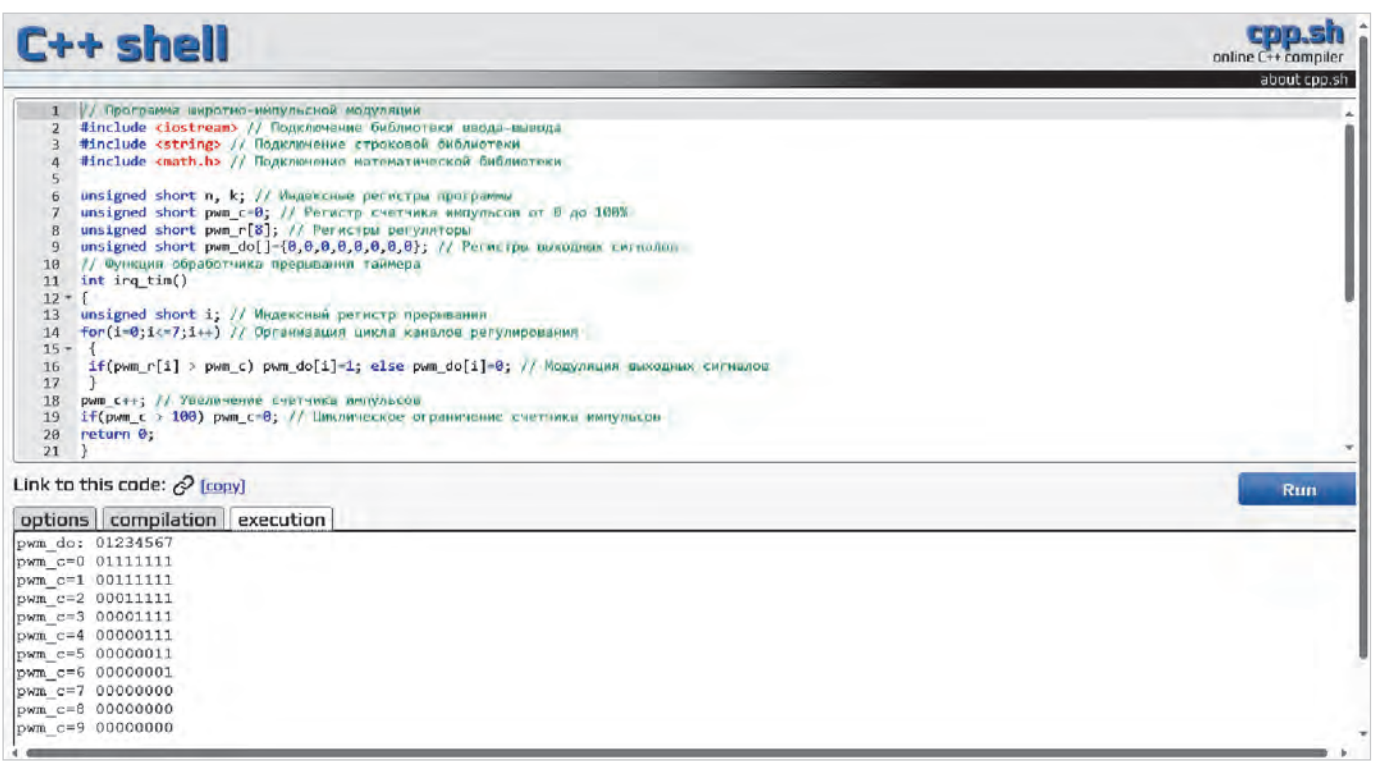

Рис. 1. Окно транслятора

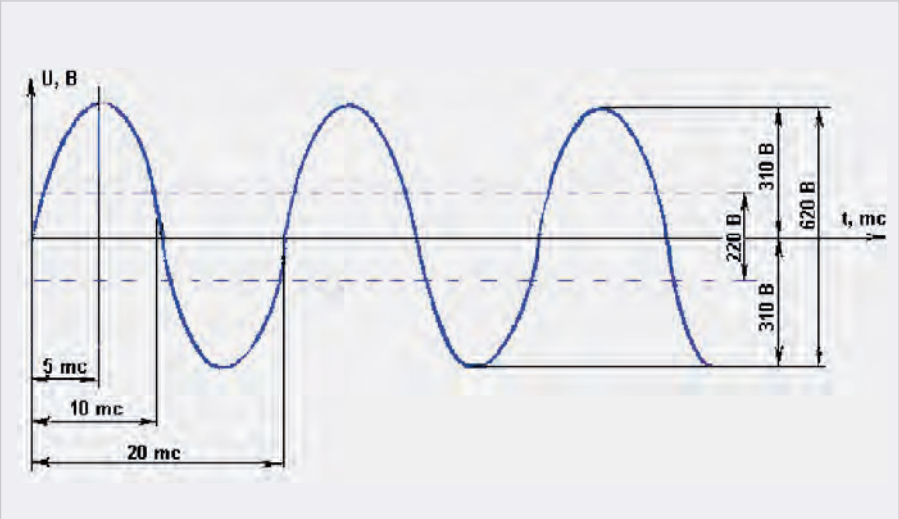

Рис. 2. График напряжения промышленной сети

```
pwm_do: 01234567
pwm_c=0 00000000
pwm_c=1 01111000
pwm_c=2 01100111
pwm_c=3 01001001
pwm_c=4 01010110
pwm_c=5 00101010
pwm_c=6 00101101
pwm_c=7 00110010
pwm_c=8 00011100
pwm_c=9 00000011
```

Рис. 3. Результат работы программы частотно-импульсной модуляции

```
цикла каналов регулирования
 {
 if(pwm_r[i] > pwm_c) pwm_
do[i]=1; else pwm_do[i]=0; //
Модуляция выходных сигналов
 }
pwm_c++; // Увеличение счетчика
импульсов
if(pwm_c > 100) pwm_c=0; //
Циклическое ограничение счётчика
импульсов
return 0;
}
// Главная функция программы
int main()
{
 // Код основной программы
 for(k=0;k<=7;k++) pwm_r[k]=k;
// Задать значения регуляторов
ШИМ для всех каналов от 0 до 7 %
 std::cout << "pwm_do: 01234567"
<< '\n';
 for(n=0;n<=9;n++) //
Организовать цикл для
отображения ШИМ выходов
 {
 std::cout << "pwm_c=" << pwm_c
<< ' ';
 irq_tim(); // Эмуляция
прерывания таймера
 for(k=0;k<=7;k++) std::cout
<< pwm_do[k]; // Вывод каналов
регулирования
 std::cout << '\n';
 }
return 0;
}
```

Для проверки работы программы и её редактирования можно использовать онлайн-редактор с транслятором языка С, например, C++Shell [1]. Окно этого транслятора показано на рис. 1.

В первое поле редактирования необходимо скопировать приведённую программу и нажать программную кнопку «Run». Программа автоматически транслируется, и результат её работы отобразится в окне вывода. Из результата видно значение всех восьми выходных сигналов для десяти значений внутреннего счётчика импульсов программы. Как и следовало ожидать, произошло заполнение сигналов регулирования единичными значениями в соответствии с их предварительно заданными значениями от 0 до 7 процентов.

Параметры таймера в приведённой программе должны определяться характером регулируемого устройства. Например, для систем нагрева от промышленной сети переменного тока с синусоидальной частотой 50 Гц длительность активной фазы нагрева составляет 10 мс, как показано на рис. 2.

Не рекомендуется задавать период таймера меньше 10 мс, поскольку при этом будет происходить коммутация нагрузки на активном участке напряжения, а не при переходе через ноль. Это приводит к формированию мощных высоковольтных импульсных выбросов напряжения в питающую сеть и электромагнитных импульсов, способных нарушить работу приборов, находящихся рядом. К таким приборам относятся радио- и телевизионные приёмники, процессорные системы и медицинское оборудование, сбой которого опасен для жизни пациентов.

Большинство твердотельных реле для переключения больших токов имеют в своём составе схему коммутации при переходе через ноль, которая исключает формирование вредных и мощных импульсов перенапряжения.

Частотно-импульсная модуляция

Для формирования регулирующего ШИМ-сигнала в диапазоне от 0 до 100% минимальный период таймера составит: T = 10 мс × 100 = 1 с. Получается, что для 50% сигнала ШИМ в течение половины секунды нагреватель будет включён на полную мощность, а вторую половину секунды полностью отключён. Такой режим нагрузки промышленной сети не всегда допустим. Например, для маломощных генераторов питающей сети это приводит либо к выходу этих генераторов из строя, либо к их нестабильной работе.

В таком случае рекомендуется применять частотно-импульсную модуляцию (ЧИМ) регулирующего сигнала. Основное отличие данного вида модуляции состоит в том, что заполнение всего периода регулирующего сигнала происходит равномерно. Поэтому для 50% сигнала ЧИМ время включения и отключения нагревателя будет чередоваться каждые 10 мс. Таким образом, нагрузка на питающую сеть будет более сбалансированной.

Для формирования ЧИМ-сигнала удобно использовать функцию синуса, задавая её период в соответствии с процентным значением и формируя импульсный выходной сигнал длительностью 10 мс при переходе функции через ноль.

Предыдущую программу легко преобразовать в программу частотно-импульсной модуляции путём замены функции обработчика прерывания таймера. Код такой функции приведён в листинге 2.

Листинг 2. Функция обработчика прерывания таймера ЧИМ

```
int irq_tim()
{
unsigned short i; // Индексный регистр прерывания
unsigned short b[]={0,0,0,0,0,0,0,0}; // Переменные прерывания
float f[8], PI=3.14159265;
for(i=0;i<=7;i++) // Организация цикла каналов регулирования
 {
 f[i]=sin((2.0*PI*pwm_r[i]*pwm_c)/100.0);
 if(f[i]>0 and b[i]==0) {pwm_do[i]=1; b[i]=1;} else {pwm_do[i]=0;} // Модуляция выходных сигналов
 if(f[i]<0) b[i]=0;
 }
pwm_c++; // Увеличение счетчика импульсов
if(pwm_c > 100) pwm_c=0;
// Циклическое ограничение счётчика импульсов
return 0;
}
```

Для наглядности результата необходимо также заменить в основной программе строку «pwm_r[k]=k;» на строку «pwm_r[k]=k*10;», увеличив тем самым заданные значения регуляторов ШИМ для всех каналов в 10 раз.

Теперь, после трансляции и выполнения новой программы, получится результат, представленный на рис. 3.

Как видно из результата, например, для пятого канала с заданным значением 50% выходной сигнал чередуется значениями 0 и 1. Суммарное количество единиц, соответствующих включению выходного ключа канала, будет равно 50 из 100 частей. Таким образом, регулирующий сигнал обеспечивает равномерную нагрузку на промышленную питающую сеть во время работы.

Заключение

Приведённые выше программы можно скачать с сайта редакции [2] и модернизировать их для реализации конкретной задачи автоматического регулирования.

Импульсные регулирующие сигналы легко преобразуются в аналоговые регулирующие сигналы при необходимости. Для этого они подключаются к RC-фильтру низких частот, на выходе которого будет сформирован аналоговый сигнал с амплитудой от 0 до физического значения напряжения логической единицы, обычно до 5 вольт. Таким образом, можно получить необходимое количество аналоговых сигналов регулирования без использования цифро-аналоговых преобразователей.

Это ещё одно из преимуществ программного способа модуляции, обеспечивающего сокращение аппаратных затрат.

Литература

1. URL: https://cpp.sh.
2. URL: https://cta.ru.

РЫНОК

«Совтест АТЕ» модернизирует производственные мощности

С целью повышения производительности и улучшения качества очистки электронных узлов на производственных мощностях ООО «Совтест АТЕ» завершена модернизация участка отмывки. Введена в эксплуатацию современная автоматическая установка струйной отмывки печатных плат SEIM DE ION 6052 с внешней системой деионизации воды.

Ранее используемая установка отмывки плат рамочного типа не обеспечивала необходимую производительность и была сложна в эксплуатации из-за долгого и трудоёмкого процесса установки плат в рамки. Новая система SEIM DE ION 6052, выполненная в формате корзинной установки, позволяет значительно упростить процесс: печатные платы можно быстро расположить в двухуровневых корзинах, вмещающих большое количество плат различных размеров и форм. Для фиксации изделий не требуются дополнительные приспособления: металлические направляющие с силиконовой оболочкой надёжно удерживают платы, предотвращая их повреждения.

Особенностью новой установки является уникальная конструкция распылительных сопел, устраняющая теневые зоны в рабочей камере, что обеспечивает равномерное и высококачественное отмывание.

Процесс полностью автоматизирован и включает несколько этапов:

- отмывка в отмывочной жидкости;
- ополаскивание в деионизированной воде с контролем уровня проводимости;
- сушка горячим воздухом.

SEIM DE ION 6052 позволяет использовать любые отмывочные жидкости на водной основе, предназначенные для струйной отмывки печатных плат. Установка оснащена системой автоматической подготовки отмывочной жидкости и контролем уровня проводимости воды для ополаскивания. Все параметры можно легко настроить посредством сенсорного дисплея с интуитивно понятным интерфейсом.

Такие установки уже успешно используются у наших клиентов и зарекомендовали себя как надёжное решение для различных условий производства.

https://sovtest-ate.ru
info@sovtest-ate.ru

На правах рекламы