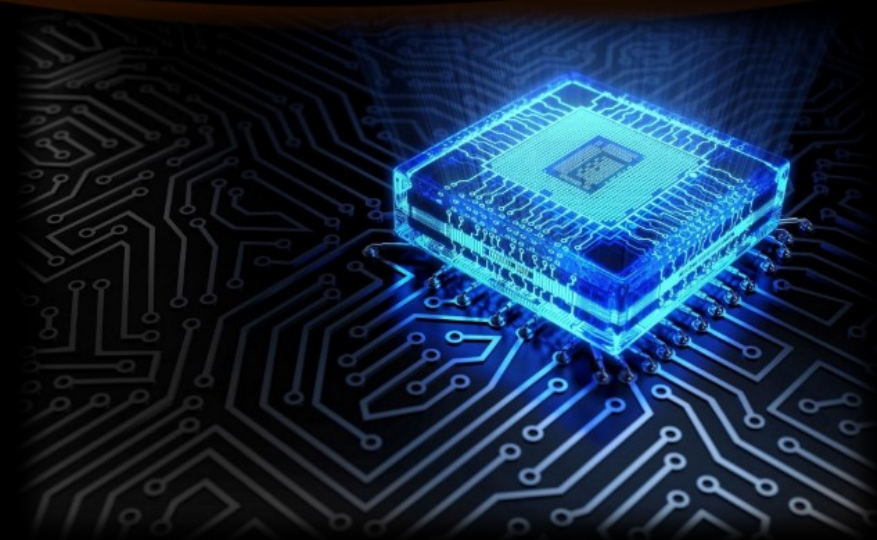
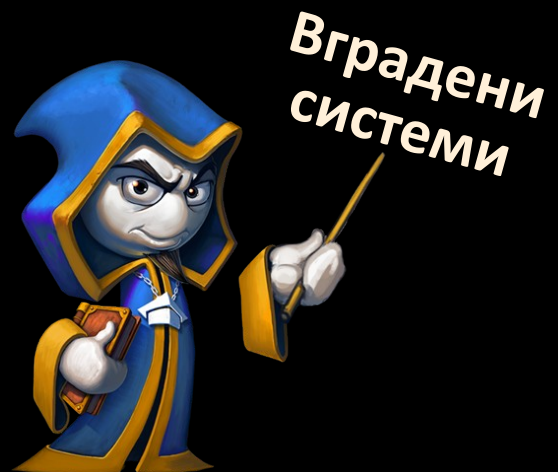


PWM

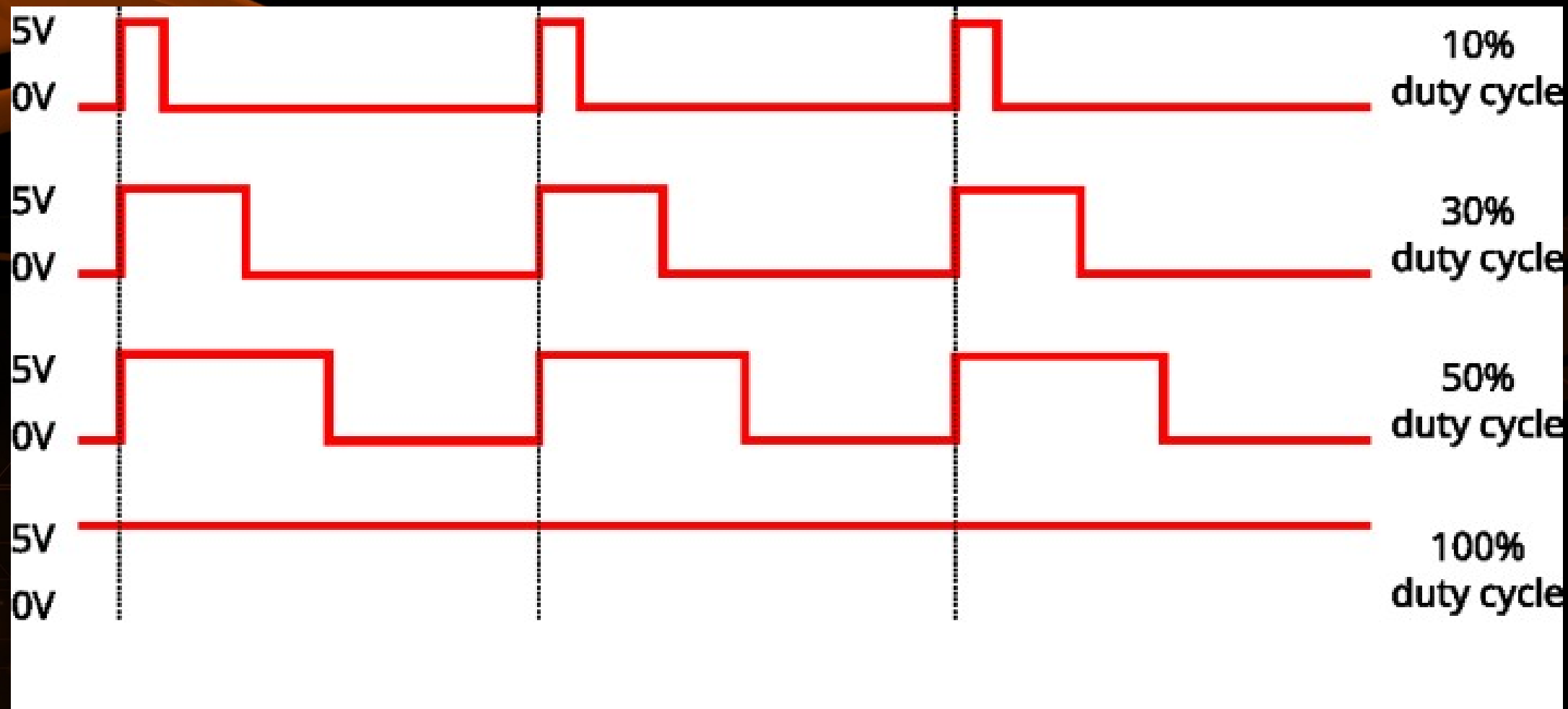


Инж. Венцеслав Кочанов

Pulse Width Modulation (PWM)

Модулацията на ширината на импулса е техника на модулация, използвана за кодиране на съобщение в пулсиращ сигнал. Въпреки че това е модулационна техника за кодиране на информация за предаване, тя се използва широко за контролиране на яркостта на светодиода и скоростта на двигателя.

ШИМ сигналът се състои от два основни компонента, които определят неговото поведение: честота и работен цикъл. Честотата определя колко бързо ШИМ завършва един цикъл (т.е. 1000 Hz честота ще бъде 1000 цикъла в секунда). Работният цикъл описва количеството време на работа като процент от общото време на цикъл.



Фиг.1

В цифровата електроника, когато сигналът е на 5V, ние наричаме това ON, а когато сигналът е на 0V, наричаме това OFF. От изображението по-горе първият сигнал е показан като 10% работен цикъл, което означава, че времето му за ВКЛЮЧВАНЕ е 10% от цикъла, а времето за ИЗКЛЮЧВАНЕ е 90% от цикъла. Вторият сигнал е 30% работен цикъл, тъй като времето му за ВКЛЮЧВАНЕ е 30% от цикъла, а времето за ИЗКЛЮЧВАНЕ е 70% от цикъла. 100% работен цикъл означава, че сигналът е ВКЛЮЧЕН през цялото време. Чрез промяна на работния цикъл на сигнала, можем да контролираме процента на времето на ВКЛЮЧВАНЕ за дадена верига, следователно индиректно да контролираме, например, яркостта на светодиода. 100% коефициент на запълване ще включва LED с пълна яркост, а 50% коефициент на запълване ще включва LED с половин яркост.

ОСНОВНИ СВОЙСТВА НА ШИМ

Широчинно-импулсната модулация или PWM е техника за подаване на електрическа енергия към товар, който има сравнително бавен отговор. Захранващият сигнал се състои от поредица от импулси на напрежение, така че ширината на отделните импулси контролира ефективното ниво на напрежение към товара. Както AC, така и DC сигнали може да се симулира с PWM. В тези бележки ще опишем използването на ШИМ на Arduino за управление на светодиоди и постояннотокови двигатели.

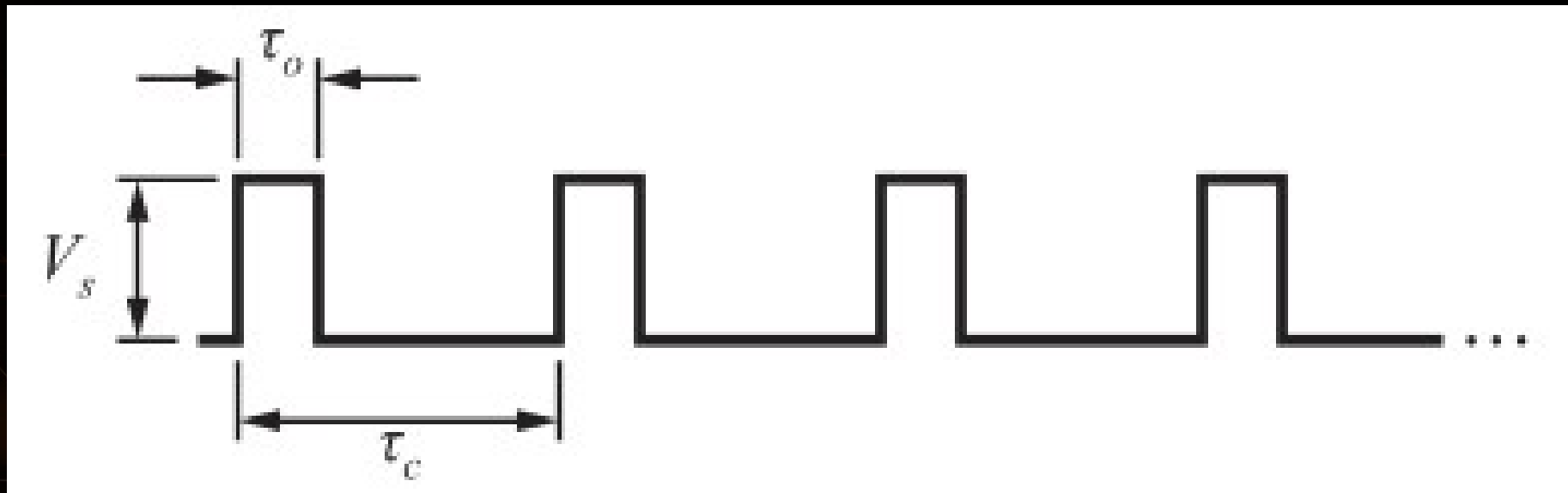
ШИМ импулсната поредица действа като DC сигнал, когато устройствата, които приемат сигнала, имат електромеханично време за реакция, което е по-бавно от честотата на импулсите.

За постояннотоков двигател съхранението на енергия в намотките на двигателя ефективно изглажда енергийните изблици, доставени от входните импулси, така че двигателят да получава по-малка или по-голяма входна електрическа мощност в зависимост от ширината на импулсите. За светодиода, използването на ШИМ кара светлината да се включва и изключва при честота, отколкото очите ни могат да открият. Ние просто възприемаме светлината като по-ярка или по-слаба в зависимост от ширината на импулсите в изхода на ШИМ.

Фигура 1 показва сигнал за напрежение, съставен от импулси с продължителност t_{on} , които се повтарят на всеки t_s единици време. Изходът на ШИМ канал е или V_s волта по време на импулса, или нула волта в противен случай. Ако този сигнал се подаде като вход към устройство, което има време за реакция много по-голямо от t_s , устройството ще изпита сигнала като приблизително DC вход с ефективно напрежение от

$$V_{\text{eff}} = V_s \frac{\tau_o}{\tau_s}$$

Съотношението τ_o/τ_s се нарича работен цикъл на правоъгълните импулси. Ефективното постоянно напрежение, подадено към товара, се контролира чрез регулиране на работния цикъл.



Използване на ШИМ на Arduino

Arduino Uno има 14 цифрови входно/изходни (I/O) пина1. Конвенционалната, т.е. не ШИМ, работа на цифровите I/O щифтове се управлява с функциите `pinMode`, `digitalRead` и `digitalWrite`.

Функцията `pinMode` се използва за конфигуриране на щифт като вход или изход. Когато цифров I/O щифт е конфигуриран като вход, `digitalRead` чете състоянието на щифта, което ще бъде HIGH или LOW. В скица на Arduino HIGH е предварително дефинирана константа, която се оценява като „истина“ в условен израз и е еквивалентна на числена стойност от 1. Електрически, стойност HIGH означава, че напрежението на пина е близо до 5 V. По същия начин, константата LOW се интерпретира като „невярно“ в условни изрази, числено е еквивалентна на 0, а електрически напрежението на щифта е 0. Когато цифров I/O щифт е конфигуриран за изход, `digitalWrite` се използва за задаване на напрежението на щифта на HIGH или НИСКО.

На Arduino Uno, PWM изходът е възможен на цифрови I/O пинове 3, 5, 6, 9, 10 и 11. На тези цифрове функцията `analogWrite` се използва за задаване на работния цикъл на PWM импулсна поредица, която работи при приблизително 500 Hz². Така, при честота $f_c = 500 \text{ Hz}$, периодът е $t_c = 1/f_c \sim 2 \text{ ms}$. Както при конвенционалните цифрови I/O, функцията `pinMode` трябва да бъде извикана първо, за да конфигурира цифровия щифт за изход.

Скелетът на скица в листинг 1 показва основните компоненти на кода за използване на ШИМ на Arduino. Две целочислени променливи, `PWM_out_pin` и `PWM_out_level` се използват за указване на щифта

съответно число и изходно ниво. Във функцията за настройка изразът `pinMode(PWM_out_pin, OUTPUT);` конфигурира `PWM_out_pin` за изход. Символът `OUTPUT` е константа, дефинирана от инструментите за програмиране на Arduino (IDE или интегрирана среда за разработка).

Декларирането на `analogWrite(PWM_out_pin, PWM_out_level);` настройва щифта `PWM_out_pin` на стойността на работния цикъл на ШИМ и следователно на ефективното напрежение съгласно уравнение (1). Параметърът, който задава работния цикъл, е 8-битово цяло число без знак, т.е. стойност в диапазона $0 \leq PWM_out_level \leq 255$. Цифровото изходно напрежение на Arduino Uno е или 0 V, или 5 V. По този начин, в уравнение (1), $V_s = 5\text{ V}$. Изходното ниво на PWM, определено с `analogWrite`, е 8-битова стойност, която съответства на диапазон на ефективно напрежение от 0 до 5 V. Фигура 2 изобразява връзките между изходните параметри на ШИМ.

```
int PWM_out_pin = 9; // Must be one of 3, 5, 6, 9, 10, or 11
// for Arduino Uno
void setup() {
  pinMode(PWM_out_pin, OUTPUT);
}
void loop() {
  byte PWM_out_level;
  PWM_out_level = ... // Code logic to set output level
  analogWrite( PWM_out_pin, PWM_out_level);
}
```

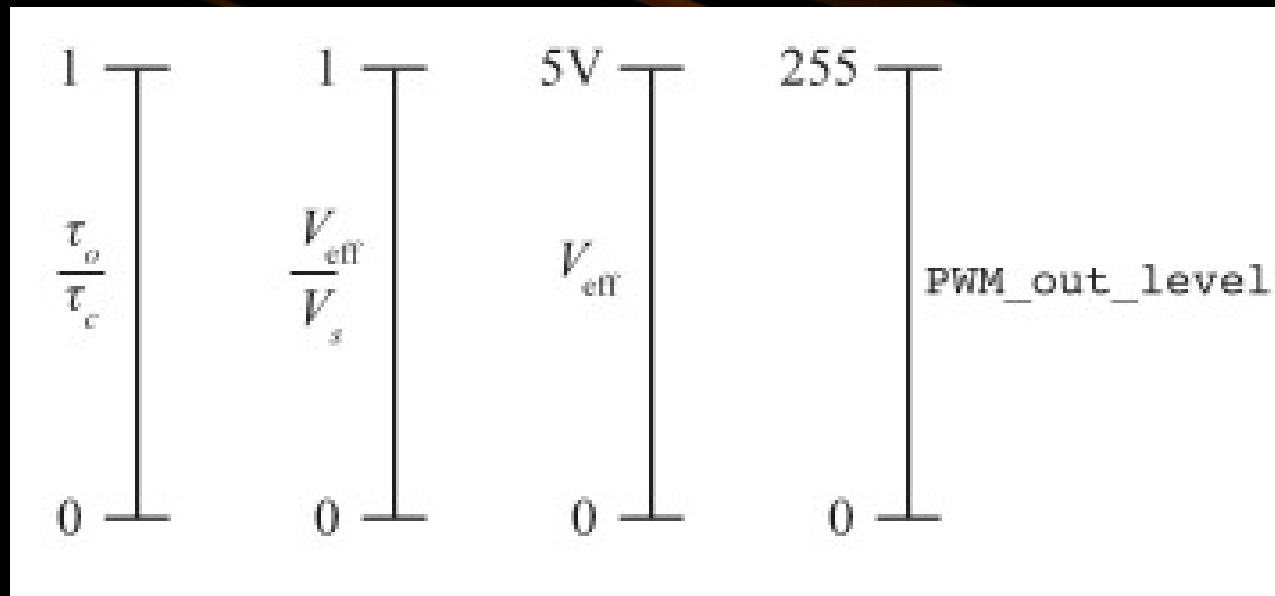
Списък 1: Скелет на скица на Arduino за демонстриране на използването на `pinMode` и `analogWrite` при контролиране на PWM изхода.

Количествата на фигура 2 са линейно свързани. По този начин:

$$\text{PWM_out_level} = 255 \times \frac{T_o}{T_c} = 255 \times \frac{V_{\text{eff}}}{V_s}$$

Да не забравяме, че вторият вход за analogWrite е 8-битова стойност без знак, така че PWM_out_level трябва да бъде тип байтова променлива. Тъй като $V_s = 5 \text{ V}$ винаги, за рутинни изчисления можем да използваме простата формула:

$$\text{PWM_out_level} = \frac{255}{5} \times V_{\text{eff}}$$



Фигура 2: Връзки на мащабиране за PWM параметри. Изходното ниво на PWM е 8-битовата стойност, използвана като втори параметър към функцията `analogWrite`

Следователно, за да се осигури ефективно напрежение от 3 V, използваме:

$$\text{PWM_out_level} = \frac{255}{5} \times 3 = 153$$

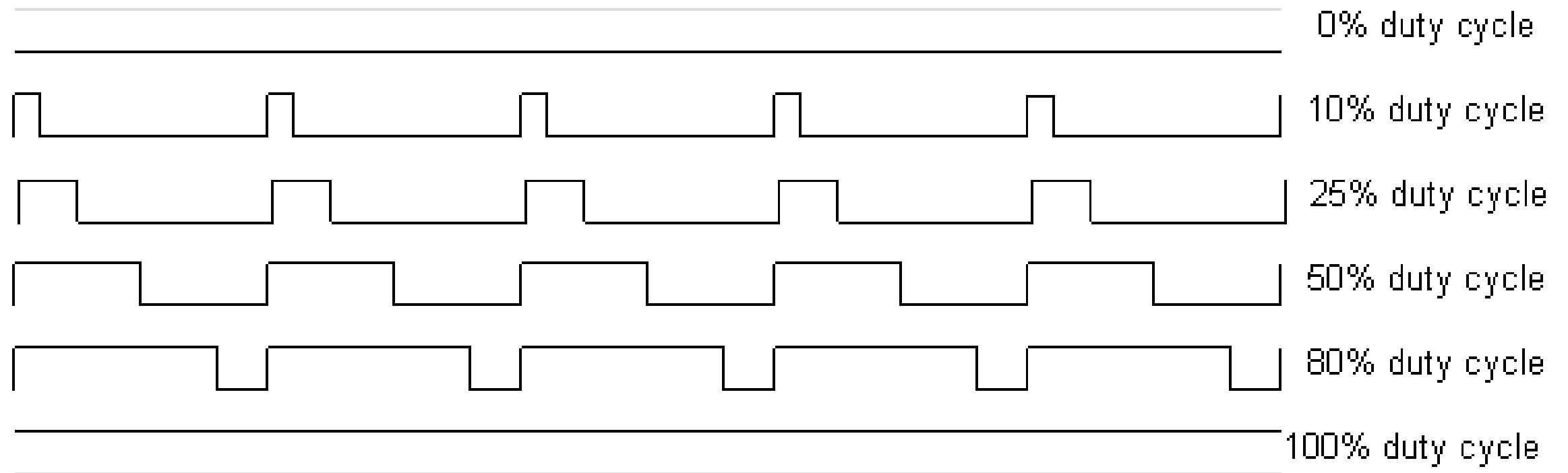
Използване на ATmega PWM регистри директно

Чипът ATmega168P/328P има три PWM таймера, контролиращи 6 PWM изхода. Чрез директно манипулиране на регистрите на таймера на чипа можете да получите повече контрол, отколкото осигурява функцията `analogWrite`.

Листът с данни на AVR ATmega328P предоставя подробно описание на таймерите с ШИМ, но листът с данни може да бъде труден за разбиране, поради многото различни режими на управление и изход на таймерите.

Обобщение

Широчинно-импулсната модулация или PWM е техника за получаване на аналогови резултати с цифрови средства. Цифровото управление се използва за създаване на квадратна вълна, сигнал, който се включва и изключва. Този модел на включване и изключване може да симулира напрежения между пълния V_{cc} на платката (напр. 5 V на UNO, 3,3 V на MKR платка) и изключен (0 волта) чрез промяна на частта от времето, което сигналът изразходва спрямо времето, което сигналът изразходва. Продължителността на "навреме" се нарича ширина на импулса. За да получите различни аналогови стойности, вие променяте или модулирате тази ширина на импулса. Ако повториме този модел на включване и изключване достатъчно бързо със светодиода например, резултатът е сякаш сигналът е стабилно напрежение между 0 и V_{cc} , контролиращо яркостта на светодиода.



Вградени системи



Въпроси?

