

SISTEM DE GENERARE A IMPULSURILOR PWM PENTRU CONTROLUL MOTORULUI BLDC

Victor POPA, Vadim PLATON

Universitatea Tehnică a Republicii Moldova

Abstract: În lucrarea este descris sistemul digital elaborat pentru generare a impulsurilor PWM, care permite schimbarea factorului de umplere, pentru controlul turațiilor și direcției de rotație a rotorului motorului BLDC. În acest scop ca element de bază a procesării și generării impulsurilor a fost ales controlerul Atmega32 și interfața de programare Atmel Studio. În procesul dezvoltării acestui sistem s-a studiat teoria de bază a conducerii și structurii unui motor BLDC, a driverului ESC de asemenea și teoriei impulsurilor PWM, după care a fost posibilă dezvoltarea funcțională a sistemului în dependență de necesitățile utilizatorului ca de exemplu controlul turațiilor de pe calculator prin interfața UART.

Cuvinte cheie: PWM, sistem, motor, BLDC, USART, ATmega32, factor de umplere

Introducere

Motoarele BLDC în ultimii ani sunt utilizați din ce în ce mai des în diferite dispozitive electrice începând de la drone de jucării până la mașini electrice (electrocar), acestea au un șir de avantaje esențiale care determină popularitatea înaltă a acestora. Diferența constructivă constă în eliminarea periilor de carbon care în motoarele simple de curent continuu exercită funcția de transmitere a curentului electric de la sursa de curent pe bobinele motorului, acest fapt și determină dezavantajul principal al motoarelor cu pierii, deoarece cu timpul aceste perii se deteriorează și disiparea temperaturii crește, în rezultat crește și consumul de curent, scăzând capacitatea de lucru a motorului. Avantajele motoarelor BLDC sunt:

- Creșterea duratei de exploatare, creșterea vitezei și puterii.
- Dimensiunile mici.
- Inexistența periilor de carbon.

Problema de bază în implementarea sistemelor de control pentru motoarele BLDC este necesitatea unui sistem integrat (embedded systems) de control al schimbării polarității fazelor cu o viteză foarte mare pentru crearea unui câmp magnetic rotativ și ca rezultat mișcarea rotorului motorului. Cu dezvoltarea științei a fost posibilă crearea driverurilor pentru implementarea algoritmilor de control, însă aceste dispozitive au nevoie de un semnal PWM (Pulse Width Modulation), care trebuie să fie generat cu o anumită frecvență și factorul de umplere.

Se propune de a elabora un sistem de generare a impulsurilor PWM pe baza controlerului Atmega32 cu posibilitatea variației factorului de umplere de la calculator prin interfața UART pentru controlul turațiilor și direcției acestora.

PWM (Pulse Width Modulation) este o modalitate folosită pentru a varia în mod controlat tensiunea dată unui dispozitiv electronic. Această metodă schimbă foarte rapid tensiunea oferită dispozitivului respectiv din ON în OFF și invers (trecuri rapide din HIGH (5V de exemplu) în LOW (0V)). Perioada de timp corespunzătoare valorii ON dintr-un ciclu ON-OFF se numește factor de umplere (*duty cycle*) și reprezintă, în medie, ce tensiune va primi dispozitivul electronic, astfel, se pot controla circuite analogice din domeniul digital.

Factorul de umplere se exprimă în procente și reprezintă cât la sută din perioada unui semnal acesta va fi pe nivelul ON. În Figura 1 se pot observa semnale PWM cu factori de umplere diferiți. Astfel, valoarea factorului de umplere (D) se poate deduce din formula:

$$D = \frac{t_{on}}{t_{on} + t_{off}} * 100\% = \frac{\text{pulse width}}{\text{period}} * 100\% \quad (1)$$

Astfel, tensiunea medie care ajunge la dispozitiv este dată de relația: $D * V_{cc}$.



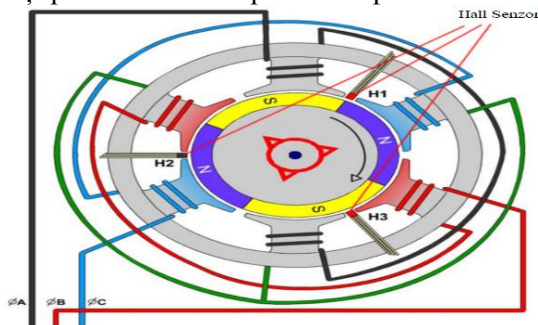
Fig. 1. Semnal PWM cu diferiți factori de umplere.

Modularea folosește variația factorului de umplere a unei forme de undă dreptunghiulară pentru a genera la ieșire o tensiune analogică. Considerând o formă de undă dreptunghiulară $f(t)$ cu o valoare minimă y_{min} și o valoare maximă y_{max} și factorul de umplere D (ca în figura figura 1) valoarea medie a formei de undă e dată de relația:

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt. \quad (2)$$

1. Modelarea Sistemului

Pentru a controla turațiile motorului experimental, este necesar de a genera un semnal PWM cu următoarele caracteristici: Frecvența 50Hz, $T=20ms$, Factorul de umplere să varieze între 5% și 10% unde procentajul reprezintă viteza și direcția rotorului după cum urmează reprezentat mai jos. Structura motorului BLDC și parametrii de umplere sunt prezentate în figura 2.



2ms = 2000 μ = 10% max speed right

1.5ms = 1500 μ = 7,5% STOP

1ms = 1000 μ = 5% max speed left

Fig. 2. Structura motorului BLDC și parametrii de umplere.

Pentru generarea semnalului PWM de referință sunt folosite posibilitățile tehnice ale controlerului Atmega32, în special Timer1 16 biți. Acest timer poate funcționa în una din cele trei stări posibile precum: Fast PWM, Phase Correct PWM, Phase and Frequency Correct PWM. Fiecare din acest mod are avantajele și modalitatea sa de numărare. Se propune de a folosi modul Fast PWM, reprezentat în figura 3. În tabelul 1 este prezentat modul de configurare PWM. Acest timer are un set de registre fig.4, care trebuie să fie configurați corespunzător caracteristicilor semnalului dorit.

Prima problemă care necesită a fi rezolvată este faptul că controlerul Atmega32 are frecvența de 16Mhz, ceea ce nu corespunde cu necesitatea de generarea unui PWM cu frecvența de 50Hz. Pentru aceasta se alege un prescaler potrivit din tabelul 3 oferit de producătorul microcontrolerului, prescaler=8.

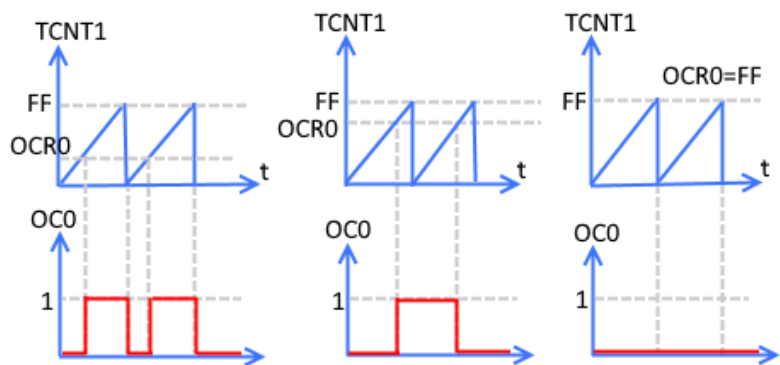


Fig. 3. Modul de comparare a valorilor în registre TCNT1 și OCR1

$$\frac{F_{CPU}}{8} = \frac{16000000Hz}{8} = 2000000Hz. \quad (3)$$

Pentru ca timer1 să poată genera impulsuri la PortB3 este necesar ca acesta să compare două valori, OCR1(Output Compare Register) și TCNT1(Timer Counter), exemplu de o relație de cauzalitate dintre variația valorilor în registrul OCR1 și registrul TCNT1 este reprezentat în fig. 3.

Tabel 1. Configurare modului PWM.

COM1A1 COM1B1	COM1A1 COM1B1	DESCRIPTION
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Mode 9,11,14,15 only: Enable OCR1A only (OC1B disconnected)
1	0	None-inverted mode (HIGH at bottom, LOW on Match)
1	1	Inverted mode (LOW at bottom, HIGH on Match)

Este necesar de a calcula valoarea de top a ICR1 pentru frecvența necesară a procesorului.

$$Val_{ICR1} = \frac{F_{cpu}}{8 \cdot 50Hz} = \frac{2000000}{50} = 40000. \quad (4)$$

Luând în considerare că numărătoarea se începe de la 1 avem 39999 de valori deci ICR1=39999.

În tabelul 2 este prezentată descrierea configurării registrelor TCCR1A și TCCR1B. Modul de funcționare a Timer1 este setat de registrele TCCR1A și TCCR1B. Bits COM1A1 (7) și COM1A0 (6) afectează semnalul care apare pe pinul OC1A (15 pini) (coincidența valorii registrului de numărare TCNT1 cu registrul de comparație OCR1A).

Tabel 2. Configurarea registrelor TCCR1A și TCCR1B.

	7 bit	6 bit	5 bit	4 bit	3 bit	2 bit	1 bit	0 bit
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS0

Bit-urile WGM13 (4), WGM12 (3) din registrul TCCR1B și bitii WGM11 (1), WGM10 (0) din registrul TCCR1A reglează modul de funcționare a timerului precum este reprezentat în fig. 4.

Având setați regiștrii și valorile din acestea se poate genera semnal PWM cu frecvența de 50Hz și factorul de umplere după necesitate calculând valoarea din OCR1 după relația de proporționalitate.

$$Val_{ICR1} = 100\%(\text{Factor de umplere } 100\%);$$

$$Val_{OCR1} = 5\%(\text{Factor de umplere } 5\%).$$

Modificând factorul de umplere dorit se modifică și valoarea din registrul OCR1 astfel la ieșirile PD4 a microcontrolerului se formează un semnal PWM cu frecvența de 50Hz și factorul de umplere dorit. Cu ajutorul parametrilor din figura 4, se setează modul de generare a semnalelor Fast PWM. [1]

Tabel 3. Configurarea prescalerului.

CS12	CS11	CS10	DESCRIPTION
0	0	0	Timer/Counter2 Disabled
0	0	1	No Prescaling
0	1	0	Clock / 8
0	1	1	Clock / 64
1	0	0	Clock / 256
1	0	1	Clock / 1024

MODE	WGM13	WGM12	WGM11	WGM10	DESCRIPTION	TOP
0	0	0	0	0	Normal	0xFFFF
1	0	0	0	1	PWM, Phase Corrected, 8bit	0x00FF
2	0	0	1	0	PWM, Phase Corrected, 9bit	0x01FF
3	0	0	1	1	PWM, Phase Corrected, 10bit	0x03FF
5	0	1	0	1	Fast PWM, 8bit	0x00FF
6	0	1	1	0	Fast PWM, 9bit	0x01FF
7	0	1	1	1	Fast PWM, 10bit	0x03FF
8	1	0	0	0	PWM, Phase and Frequency Corrected	ICR1
9	1	0	0	1	PWM, Phase and Frequency Corrected	OCR1A
10	1	0	1	0	PWM, Phase Correct	ICR1
11	1	0	1	1	PWM, Phase Correct	OCR1A
14	1	1	1	0	Fast PWM	ICR1
15	1	1	1	1	Fast PWM	OCR1A

Waveform Generator Mode bits (Abbreviated)

Fig. 4. Configurarea modului de generare a semnalului.

În figura 5 este prezentată o parte din listingul programului cu comentarii, în care se explică destinația fiecărui registru utilizat.

```

int main(void)
{
    uart_initialize();                /*Initialize uart
    UCSRB |= (1 << RXCIE);          /*ISR
    sei();                          /*ON Global Interrupt

    DDRD = (1 << DDD4);             /*Set PD4 as output
    TCCR1A = 0;                    /*Clean TCCR1B Register
    TCCR1B = 0;                    /*Clean TCCR1B Register
    TCCR1A |= ( 1<<COM1B1 |1<<COM1B0 |1<<WGM11); /*Set inverted mode
    TCCR1B |= ( 1<<WGM12 |1<<WGM13 |1<<CS11); /*Set Fast PWM mode and prescaler 8
    ICR1 = 39999;                  /*Set register TOP value
    init_bldc();
    ADC_Init();                    /*Initialize ADC */
    while(1)
    {
        ADC_Value = ADC_Read(1);   /*Read the status on Y-OUT pin using channel 0 */
        sprintf(str, "Y=%d\r\n", ADC_Value); /*Y = PA1
        write_uart_strg(str);      /*Transmit ADC_Val via UART to Desktop
        duty=map(ADC_Value,0,1023,2000,4000); /*Convert Pot position to duty
        OCR1B = ICR1 - duty;      /*Change OCR1B value to change duty cycle
    }
}

int ADC_Read(char channel)
{
    int ADC_value;
    ADMUX = (0x40) | (channel & 0x07); /* set input channel to read */
    ADCSRA |= (1<<ADSC);             /* start conversion */
    while((ADCSRA & (1<<ADIF))!= 0); /* monitor end of conversion interrupt flag */
    ADCSRA |= (1<<ADIF);            /* clear interrupt flag */
    ADC_value = (int)ADCL;          /* read lower byte */
    ADC_value = ADC_value + (int)ADCH*256; /* read higher 2 bits, Multiply */
    return ADC_value;              /* return digital value */
}

void ADC_Init()
{
    DDRA = 0x00;                   /* Make ADC port as input */
    ADCSRA = 0x87;                 /* Enable ADC, fr/128 */
    ADMUX = 0x40;                  /* Vref: Avcc, ADC channel: 0 */
}

```

Fig 5. Listingul programului.

Concluzii

În urma elaborării acestui sistem au fost implementate cu succes toate particularitățile tehnice și atinse scopurile propuse, a fost creat un sistem digital de generare a impulsurilor PWM cu posibilitatea de a schimba factorul de umplere pentru controlul turațiilor și direcția de rotație a rotorului motorului BLDC. Pentru realizarea acestui sistem pentru procesarea și generarea impulsurilor a fost ales controlerul Atmega32 și interfața de programare Atmel Studio. Proiectul dat poate fi utilizat în diferite aplicații tehnice unde este necesară generarea impulsurilor PWM; ca de exemplu pentru modularea și convertirea din semnal digital în semnal analogic sinusoidal (figura 6)

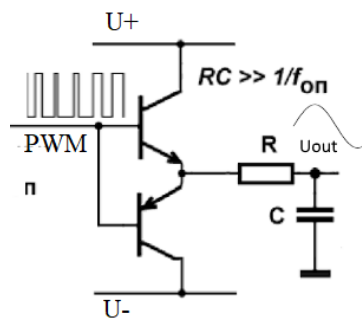


Fig .6 Generare semnalului sinusoidal prin PWM.

Bibliografie

1. Atmel@AVR@ 8-bit Microcontroller Atmega32-Datasheet. <http://ww1.microchip.com/downloads/en/devicedoc/doc2503.pdf> [Accesat 05.03.2019].
2. Satish Kumar Peddapelli *Pulse Width Modulation: Analysis and Performance in Multilevel Inverters* De Gruyter, 2017. — 210 p. — ISBN 978-3110468175.