

Controllo di un motore brushless

Stefano Cattonar

Giugno 2019

Indice

1	Teoria	3
1.1	Algoritmo Six-Step Commutation	3
1.1.1	Descrizione dell'algoritmo	3
1.1.2	Implementazione dell'algoritmo	3
2	Pratica	4
2.1	Tabella di Commut.azione	4

Introduzione

La corrente elettrica passa in un filo di rame che avvolge a spirale un pezzo di ferro dolce chiamato rotore. Questo avvolgimento, crea un campo elettromagnetico al passaggio di corrente. Questo campo elettromagnetico è immerso in un altro campo magnetico creato dallo statore, che nel caso più semplice è costituito da una o più calamite, o elettrocalamite. Il rotore per induzione elettromagnetica inizia a girare, in quanto il "nord" del campo magnetico del rotore è attratto dal "sud" del campo magnetico dello statore e viceversa. Ogni mezzo giro la polarità viene invertita, in modo da dare continuità alla rotazione nel secondo mezzo giro ...e così via.

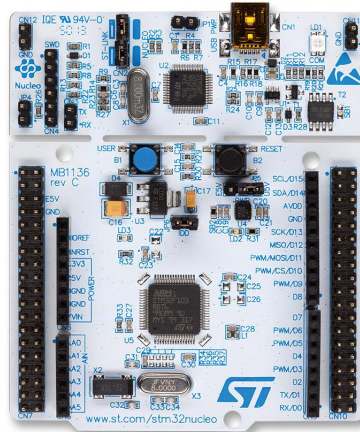
Durante la trasformazione, una modesta parte dell'energia viene dispersa in calore per l'effetto Joule. Il motore elettrico a seconda della sua tensione di alimentazione, e del suo comportamento, può essere un motore asincrono, motore sincrono, o un motore in corrente continua.

Formula della coppia in un motore elettrico:

$$T = \frac{V * w * k}{r} * k.$$

T = Momento Torcente
V = Voltaggio
w = Velocità di rotazione
k = Costante del motore
R = Resistenza

Per il corso di laboratorio ciberfisico abbiamo dovuto implementare il software di gestione di un motore brushless attraverso la programmazione a basso livello di una board della [STMicroelectronics](https://www.st.com/it/microelectronics). Nel nostro caso una ST-NucleoF401RE.



Capitolo 1

Teoria

1.1 Algoritmo Six-Step Commutation

1.1.1 Descrizione dell'algoritmo

L'algoritmo “*Six-Step Commutation*” è necessario per la corretta rotazione del motore. Come dice la parola inglese **commutation** presente nel nome dell'algoritmo, si tratta di commutare, cioè di eseguire la cummutazione dei tre circuiti elettrici presenti.

1.1.2 Implementazione dell'algoritmo

Per poter implementare l'algoritmo bisogna conoscere le seguenti cose:

- Conoscere il numero di poli magnetici del proprio motore
- Conoscere il numero di spire del proprio motore

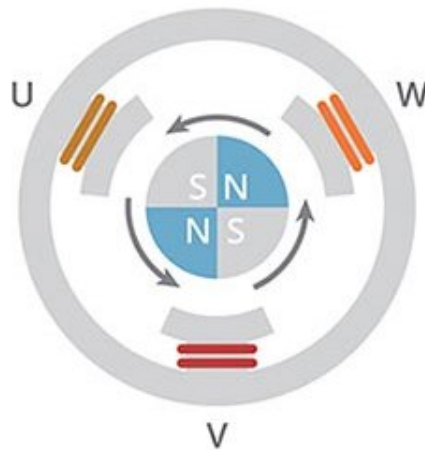
Dopodiché si può creare la tabella di commutazione. Nel prossimo capitolo mostrerò l'implementazione per il nostro motore.

Capitolo 2

Pratica

2.1 Tabella di Commut.azione

Il nostro brushless ha 2 poli magnetici come si può vedere dall'immagine sottostante:



La nostra tabella di commutazione quindi si presenta così:

U	V	W
+	-	off
-	off	-
off	+	-
-	+	off
-	off	+
off	-	+

Creata la tabella di commutazione è necessario capire il numero di volte in cui le sei fasi si ripetono in un giro completo.

```

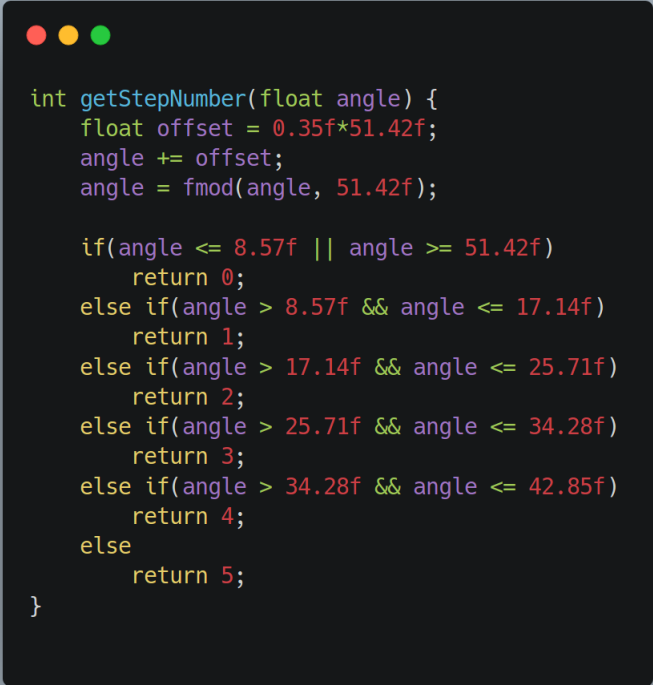
void switchPhase() {
  switch(step_number) {
    case 0:
      digitalWrite(pin_positive);
      digitalWrite(pin_negative);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      break;
    case 1:
      digitalWrite(pin_positive);
      digitalWrite(0);
      digitalWrite(pin_negative);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      break;
    case 2:
      digitalWrite(0);
      digitalWrite(pin_positive);
      digitalWrite(pin_negative);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      break;
    case 3:
      digitalWrite(pin_negative);
      digitalWrite(pin_positive);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      break;
    case 4:
      digitalWrite(pin_negative);
      digitalWrite(0);
      digitalWrite(pin_positive);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      break;
    case 5:
      digitalWrite(0);
      digitalWrite(pin_negative);
      digitalWrite(pin_positive);
      digitalWrite(0);
      digitalWrite(0);
      digitalWrite(0);
      break;
  }
}

```

Figura 2.1: Implementazione del codice della tabella

Nel nostro caso le sei fasi permettono di fare solo 51° di rotazione, quindi: $360/51 = 7$. Visto che il totale delle fasi fa 51° circa ogni fase equivale a $51/6 = 8,5^\circ$.

Una volta caricato questo codice C++ sulla board (insieme ad altro codice che sarà lasciato al lettore), il nostro motore è in grado di produrre una rotazione continua e quindi una coppia costante in una direzione.



```
int getStepNumber(float angle) {  
    float offset = 0.35f*51.42f;  
    angle += offset;  
    angle = fmod(angle, 51.42f);  
  
    if(angle <= 8.57f || angle >= 51.42f)  
        return 0;  
    else if(angle > 8.57f && angle <= 17.14f)  
        return 1;  
    else if(angle > 17.14f && angle <= 25.71f)  
        return 2;  
    else if(angle > 25.71f && angle <= 34.28f)  
        return 3;  
    else if(angle > 34.28f && angle <= 42.85f)  
        return 4;  
    else  
        return 5;  
}
```

Figura 2.2: Implementazione dell'algoritmo