

Projeto de um ESC para os robôs de combate

April 26, 2019



Um ESC - *Electronic Speed Control* - é o circuito que recebe os comandos do receptor de rádio e aciona os motores.

A ideia é implementá-lo com um MSP430G2211 ou MSP430G2231, que são microcontroladores de 16 bits, em encapsulamentos de 14 pinos, feitos para aplicações bem simples. A principal restrição destes microcontroladores é o fato de terem apenas 1 timer, com 2 registradores de comparação/captura, o que significa que boa parte do processamento será obrigatoriamente feita por software.

O sistema tem como entradas:

1. Os sinais do radioreceptor - avanço/recuo (AR) e direita/esquerda (DE).
2. Uma fração da tensão da bateria.
3. Uma medida da corrente pelos transistores de potência.
4. Talvez um sinal de temperatura, dos MOSFETs drivers dos motores.

As saídas do sistema são:

1. Os sinais de comando dos motores esquerdo e direito - dois por motor.
2. Comandos de um led RGB.

1 Descrição das entradas

O rádioreceptor envia os sinais no formato padrão para servomotores de aeromodelos: um trem de pulsos com período de aproximadamente 20 ms e largura de pulso variando entre 1 e 2 ms. O sinal é modulado justamente na largura de pulso, sendo o valor mínimo com 1 ms e o máximo com 2 ms. No caso do canal de avanço/recuo, por exemplo, um valor de aproximadamente 1,5 ms é o

neutro (nem avanço nem recuo), sinais menores que 1,5 ms indicam recuo até a velocidade máxima de recuo com um sinal de 1 ms e sinais maiores indicam avanço, até o máximo de 2 ms. O canal de direita/esquerda tem funcionamento equivalente. Faremos duas variáveis **chARval** e **chDEval** para cada canal, que ficará com valores entre -255 e 255, sendo 0 o valor no ponto neutro.

Estes tempos podem variar com cada radioreceptor, com cada transmissor e até com a calibração de cada transmissor. Além disso, o clock do microcontrolador não é muito confiável (numa implementação sem cristal), logo é necessário ter um procedimento de calibração dos valores máximo, mínimo e neutro de cada canal.

É importante que caso os sinais do radioreceptor não estejam chegando que os motores fiquem parados e que tal situação seja indicada pelo led, logo devemos ter um tempo máximo **MAX_COMM_DELAY**, dentro do qual esperamos ter um sinal de comando em ambos canais.

A fração da tensão da bateria e, se implementado, a temperatura detectar situações em que não se deve operar (baixa tensão ou alta temperatura) e causam o desligamento dos motores.

2 Acionamento dos motores

O valor de cada motor **MEval** e **MDval**, também definido entre -255 a +255, é definido a partir dos valores dos canais de acordo com as seguintes condições:

- **chARval** = 0 \rightarrow **MDval** = -**chDEval** e **MEval** = **chDEval**
- **chDEval** = 0 \rightarrow **MDval** = **MEval** = **chMEval**
- $|\mathbf{chARval}| = |\mathbf{chDEval}|$, **chDEval** > 0 \rightarrow **MDval** = 0, **MEval** = **chDEval**
- $|\mathbf{chARval}| = |\mathbf{chDEval}|$, **chDEval** < 0 \rightarrow **MEval** = 0, **MDval** = **chDEval**

Não seguindo exatamente tal procedimento, mas podemos implementar como:

$$\begin{aligned}\mathbf{MEval} &= \mathbf{chARval} - \mathbf{chDEval} \\ \mathbf{MDval} &= \mathbf{chARval} + \mathbf{chDEval}\end{aligned}$$

Os motores são controlados através de pontes H com dois sinais. Para acionarmos os usaremos a técnica de manter um lado da ponte H em 0 e acionar o outro lado com um sinal PWM ou delta-sigma, logo um sinal de cada motor será de avanço (**MDA** e **MEA**) e outro de recuo (**MDR** e **MER**). Considerando que cada sinal tem um mínimo de 0 e um máximo de 255, os valores do pwm

de cada motor são definidos por:

$$\begin{aligned}\mathbf{MDA} &= \begin{cases} \mathbf{MDval}, & \text{se } \mathbf{MDval} > 0 \\ 0, & \text{caso contrário} \end{cases} \\ \mathbf{MDR} &= \begin{cases} -\mathbf{MDval}, & \text{se } \mathbf{MDval} < 0 \\ 0, & \text{caso contrário} \end{cases} \\ \mathbf{MEA} &= \begin{cases} \mathbf{MEval}, & \text{se } \mathbf{MEval} > 0 \\ 0, & \text{caso contrário} \end{cases} \\ \mathbf{MER} &= \begin{cases} -\mathbf{MEval}, & \text{se } \mathbf{MEval} < 0 \\ 0, & \text{caso contrário} \end{cases}\end{aligned}$$

3 Led de status

O led deve variar de cor de acordo com o status do sistema:

Verde – pleno funcionamento.

Amarelo – sem comunicação.

Vermelho (piscando?) -- bateria baixa.

Azul – calibração.

4 Procedimento de calibração das entradas