

Laborator 12

Servomotor și Buzzer

Obiective

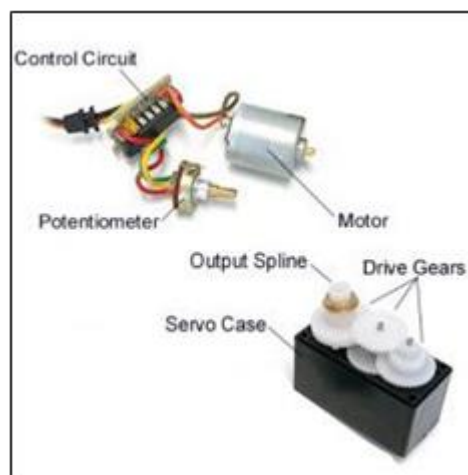
- Prezentarea servomotorului
- Înțelegerea principiului de funcționare a unui servomotor
- Controlul SW a unui servomotor
- Controlul unui Buzzer

Cuprins

Obiective.....	1
Cuprins.....	1
Servomotorul.....	2
Principiul de funcționare	2
Cum putem controla servomotorul.....	2
Cum se calculează factorul de umplere al semnalului de comandă PWM	4
Buzzerul	5
Aplicații	5
Scopul laboratorului	5

Servomotorul

Servomotorul este un motor electric special, de curent continuu, având ca scop deplasarea într-un timp prescris al unui sistem mecanic (sarcina) de-a lungul unei traiectorii date, realizând totodată și poziționarea acestuia la sfârșitul cursei cu o anumită precizie.



Principiul de funcționare

Pentru a înțelege principiul de funcționare al unui servomotor, trebuie să aruncăm o privire în interior:

Elementele principale ale unui servomotor:

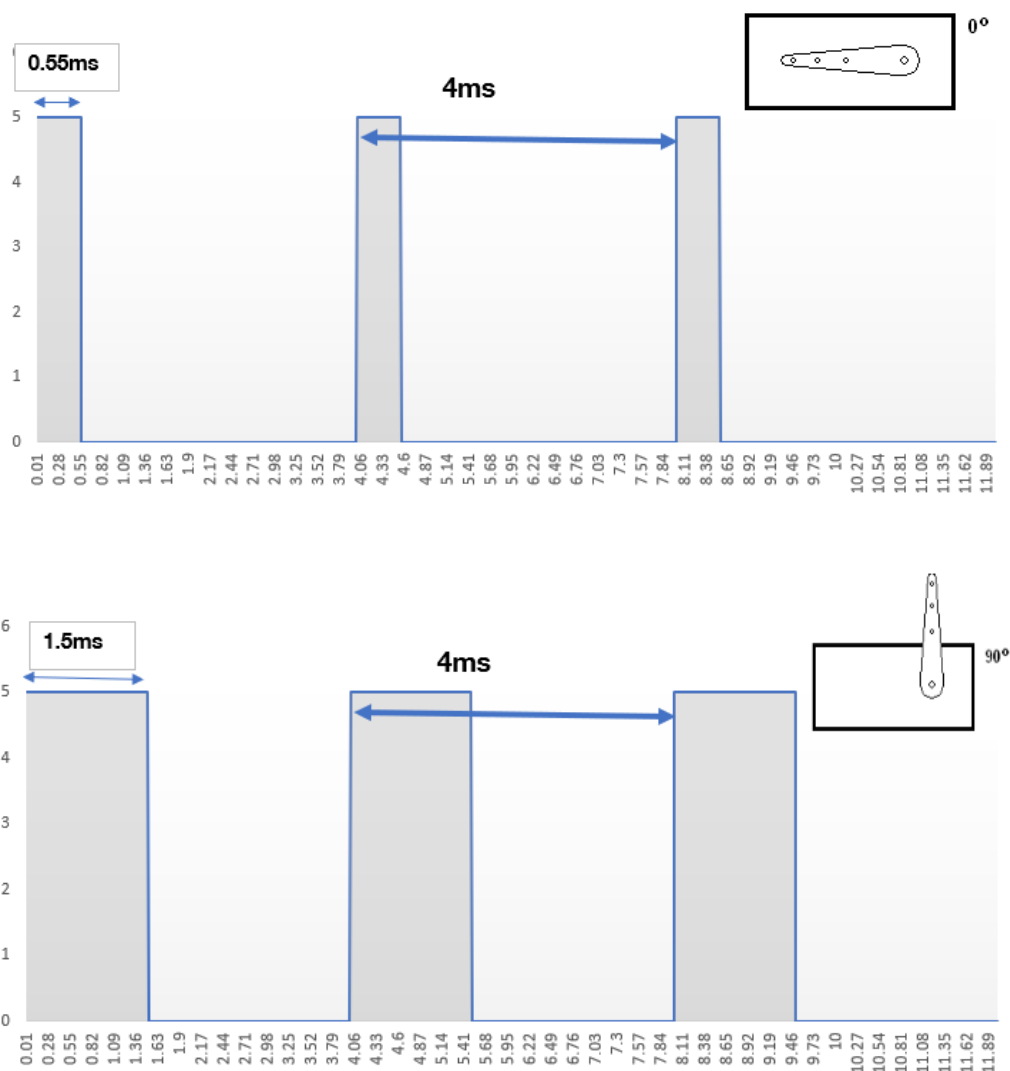
- Motor de curent continuu
- Circuit de control
- Potențiomtru

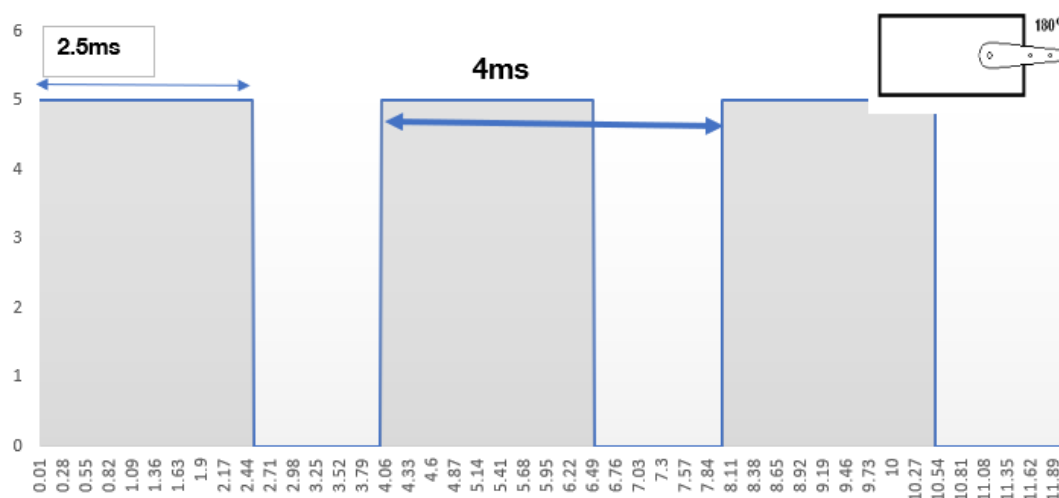
Odată cu mișcarea axului, rezistența potențiometrului se modifică, respectiv se poate afla poziția exactă a acestuia. Circuitul de control primește comanda sub forma unui impuls PWM și cunoscând poziția curentă, comandă motorul de curent continuu într-o direcție sau în alta, până se ajunge la poziția dorită.

Cum putem controla servomotorul

După cum am menționat anterior, servomotorul este comandat prin intermediul unui impuls PWM. Acesta trebuie să aibă o perioadă de 4ms (modulul PWM al microcontrolerului a fost deja inițializat cu o perioadă de 4ms).

Variind factorul de umplere (duty cycle) se poate controla poziția acului:





Cum se calculează factorul de umplere al semnalului de comandă PWM

$$dutyCycle = \frac{T_{on}}{Period} * 100$$

Așadar avem:

- $0.55 / 4 * 100 = 13.75\%$
- $1.5 / 4 * 100 = 37.5\%$
- $2.5 / 4 * 100 = 62.5\%$

Controlul PWM se realizează cu ajutorul funcțiilor:

```
esp_err_t pwm_init(uint32_t period, uint32_t *duties, uint8_t channel_num, const
uint32_t *pin_num);
```

În care:

- period reprezintă perioada dată în us
- duties este un vector ce conține toți factorii de umplere inițiali pentru fiecare canal PWM, dați în us (nu în %)
- channel_num reprezintă numărul de canale configurate
- pin_num este un vector ce conține numerele pinilor corespunzătoare canalelor pwm configurate

```
esp_err_t pwm_set_duty(uint8_t channel_num, uint32_t duty);
```

 funcție ce setează factorul de umplere

În care:

- channel_num reprezintă numărul canalului pentru care se va schimba factorul de umplere
- duty reprezintă factorul de umplere dat în us

Buzzerul

Un buzzer poate fi de două feluri: **Activ și Pasiv**.

Buzzer-ul **Activ** conține un oscilator propriu, ceea ce înseamnă că este necesară doar alimentarea sa cu 3-5V pentru a produce un sunet. Oscilatorul va alterna tensiunea ON și OFF foarte rapid, producând un ton ce poate fi variat cu un PWM.

Buzzer-ul **Pasiv** nu conține un oscilator propriu, iar în cazul alimentării DC, sunetul produs va fi sub forma unui click deoarece diaphragm-ul se comută în limita superioară sau inferioară și ramane acolo. Dacă este aplicat un semnal PWM el poate produce un sunet deoarece PWM-ul alternează alimentarea asemănător unui oscilator.

Aplicații

Funcție	Parametrii	Descriere
SERVO_vChangeAngle	u32ServoAngle	Funcția modifică unghiul servomotorului la 0, 90 sau 180 de grade.
BUZZER_vChangeDutyCycle	u32BuzzerDutyCycle	Funcția modifică factorul de umplere al sunetului produs de buzzer.

Scopul laboratorului

1. Creați funcția BUZZER_vChangeDutyCycle;
2. Din aplicația web, folosind butonul de claxon (Honk), porniți și opriți buzzerul;
Nota: Verificarea cererii venita de la aplicația web se face la 500ms.
3. Creați funcția SERVO_vChangeAngle;
4. Din aplicația web, folosind butonul de deschidere a portbagaj (Open Trunk), deschideți și închideți portbagajul;
Nota: Verificarea cererii venita de la aplicația web se face la 500ms.