

Achizitii de Date

Laboratorul 1: Pulse Width Modulation

Dan Novischi

1. Introducere

Scopul acestui laborator îl reprezintă utilizarea tehnicii de modularea a semnalelor în latime (en. Pulse Width Modulation) pentru a varia independent viteza unor motoare de curent continuu folosind Arduino și TinkerCAD.

2. Modularea Semnalelor în Latime / Pulse Width Modulation

PWM (Pulse Width Modulation) este o tehnică 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 în LOW, de exemplu 5V - 0V). Raportul dintre perioada de timp corespunzătoare valorii ON și perioada totală dintr-un ciclu ON-OFF se numește factor de umplere (duty cycle) și reprezintă, în medie, tensiunea pe care o va primi dispozitivul electronic. Astfel, se pot controla circuite analogice din domeniul digital. În cazul unui motor, căruia i se aplică un semnal PWM cu factor de umplere de 0%, viteza de rotație a acestuia va fi egală cu 0 rpm. Un factor de umplere de 100% va duce la o turație maximă a acestuia.

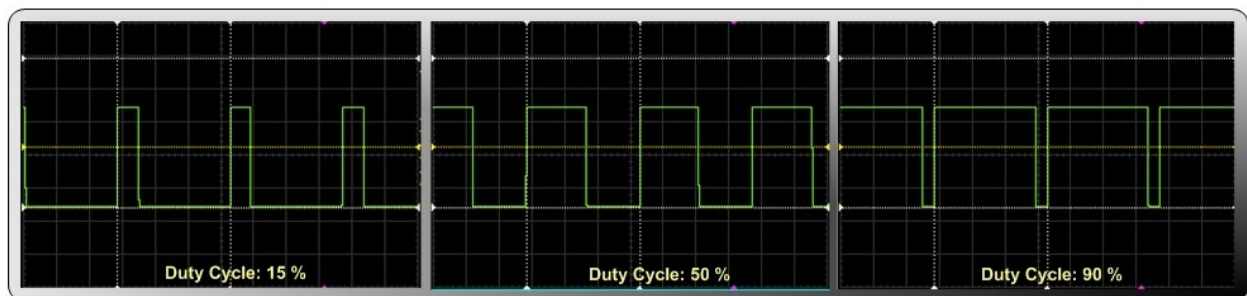
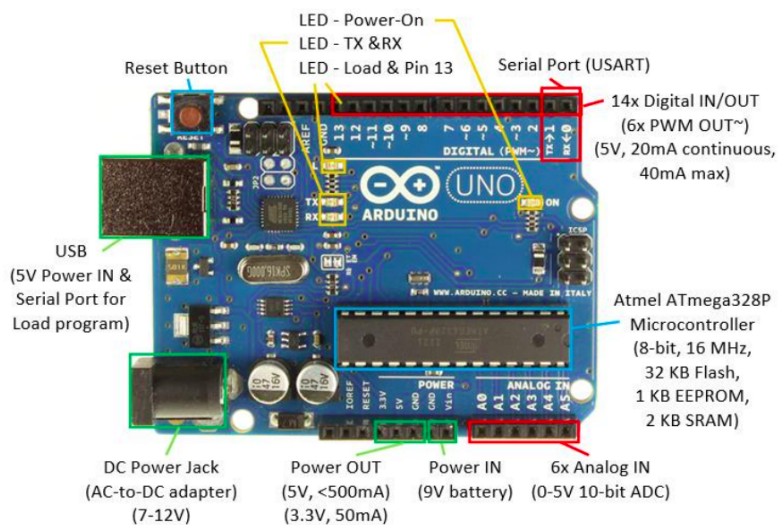


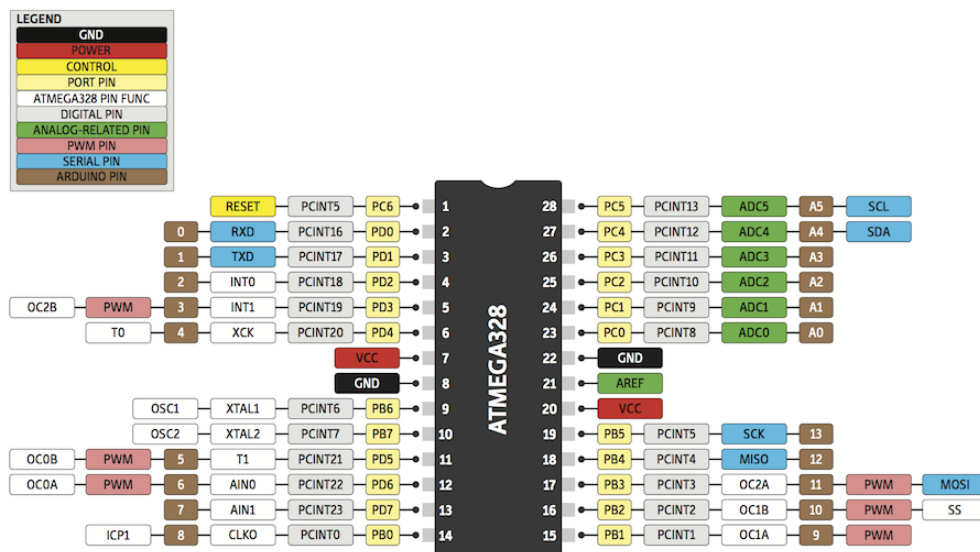
Figura 1: Forme de unda PWM la diferiti factori de umplere (duty cycle)

3. Arduino si Interfata PWM

Placa Arduino UNO, prezentata in Figura 2a, este o placă de dezvoltare open-source realizată pe baza microcontrolerului ATmega328P.



(a) Structura Placii.



Din punct de vedere al software-ului structura generala a unei aplicatii integrate pentru Arduino se bazeaza pe functiile `void setup(void)`; si `void loop(void)`; avand urmatoarea forma:

```
int main(void){
    setup();
    while(1){
        loop();
    }
}
```

Astfel, functia `void setup(void)`; va contine intotdeauna codul pentru setarea modului de functionare a diferitelor periferice si/sau functii ale pinilor, in timp ce functia `void loop(void)`; va contine intotdeauna logica aplicatiei.

Pentru utilizarea facilitatilor de PWM ale placii Arduino, [API-ul](#) ne pune la dispozitie trei functii, anume:

- `pinMode()` – care controleaza functia unui anumit pin (ex: INPUT, OUTPUT)
- `analogWrite()` – care controleaza factorul de umplere al modulului de PWM relativ la forma de unda generata pe pinii specifici acestuia de pe placa Arduino.
- `map()` – scaleaza un numar de la un interval la alt interval.

4. Cerinte

Cerinta 1 Utilizand TinckerCAD familiarizati-va cu circuitul si scheletul de cod pentru acest laborator: <https://www.tinkercad.com/things/bBao5PX0QXR>

Cerinta 2 Pornind de la circuitul dat legati un al doilea motor la integratul L293D.

Cerinta 3 In scheletul de cod asociat simularii, completati setarile necesare utilizarii pinilor 10, 11, 9, 3 in modul de OUTPUT pentru PWM.

Cerinta 4 Implementati functia `void setPWM(Motor m, Direction d, int duty_cycle)`; pentru a misca un anumit motor, intr-o anumita directie cu factorul de umplere `duty_cycle` primit ca parametru.

Cerinta 5 In functia `loop()`; implementati o simulare pentru accelerarea celor doua motoare, in directia inainte in 10 pasi pe durata unei secunde.

Cerinta 6 In functia `loop()`; adaugati o simulare pentru decelerarea celor doua motoare, in directia inainte in 10 pasi pe durata unei secunde.

Cerinta 7 In functia `loop()`; adaugati o simulare similara cerintelor 5 si 6 dar pentru directia inapoi.