



TP 05 : Simulation et implémentation d'un programme VHDL pour commander un moteur a courant continue à vitesse variable

Thème : Moteur CC, vitesse variable



Commande d'un moteur à courant continu à vitesse constante

Objectifs pédagogiques

- Comprendre le principe de commande d'un moteur à courant continu (MCC) à l'aide d'un driver en pont en H.
- Écrire un module VHDL qui pilote le moteur dans les deux sens de rotation à l'aide de boutons poussoirs.
- Mettre en œuvre la liaison entre le FPGA/CPLD et le driver de puissance.

Partie théorique

♦ Moteur à courant continu

- Le MCC convertit l'énergie électrique en énergie mécanique (rotation).
- Le **sens de rotation** dépend de la polarité appliquée à ses bornes.
- Pour inverser le sens, on inverse la polarité → d'où l'utilisation d'un **pont en H**.

♦ Driver (pont en H)

- Le driver (ex. **L293D**, **L298N**) permet de commander le moteur avec de faibles signaux logiques issus du FPGA.
- Sortie logiques reliées au FPGA/CPLD :
 - **IN1 / IN2** → déterminent le sens de rotation.
 - L'arrêt est obtenu en mettant les deux entrées à 0..
- Sorties de puissance : reliées directement aux bornes du moteur.

♦ Description du système

- Entrées du FPGA/CPLD
 - BP1 (Monter / sens horaire)
 - BP2 (Descendre / sens antihoraire)
 - BP1 + BP2 simultanés → Arrêt

♦ Table de vérité

BP1	BP2	IN1	IN2	Action moteur
0	0	0	0	Arrêt
0	1	0	1	Marche (cense droite)

1	0	1	0	Marche (cense gauche)
1	1	1	1	Arrêt

Travail demandé

- **Écrire un module VHDL** qui :
Lit les entrées BP1 (bouton 1), BP2 (bouton 2) qui contrôleront le sens de rotation du moteur.
Génère les sorties IN1 (LED 1), IN2 (LED 2) (selon la table de vérité) qui normalement seront connectés aux entrées du Driver
- Génère PIN planner
- **Simuler le code pour vérifier la logique.**
- **Implémenter sur FPGA/CPLD** et relier les sorties aux entrées du driver.
- **Tester sur le moteur réel** avec les boutons poussoirs.

Commande d'un moteur à courant continu avec variation de vitesse (PWM)

Objectifs pédagogiques

- Comprendre le principe de la **variation de vitesse** d'un moteur à courant continu par **modulation de largeur d'impulsion (PWM)**.
- Écrire un **module VHDL** qui génère un signal PWM.
- Associer ce signal PWM aux entrées du **driver en pont en H** pour commander à la fois le **sens de rotation** et la **vitesse**.
- Mettre en œuvre la liaison entre FPGA/CPLD et driver de puissance.

Partie théorique

- ◆ Moteur à courant continu
 - La vitesse du moteur est proportionnelle à la **tension moyenne appliquée**.
 - Pour réduire la vitesse, on ne baisse pas directement la tension d'alimentation, mais on applique une **PWM**.
- ◆ PWM (Pulse Width Modulation)
 - Signal carré de fréquence fixe, Pour un moteur à courant continu, la fréquence PWM adéquate se situe généralement entre 20 kHz et 25 kHz. Cela permet d'éviter les bruits audibles (supérieurs à 20 kHz) et d'assurer une commande fluide sans pertes excessives.
 - La **largeur de l'impulsion (duty cycle)** détermine la tension moyenne :
 - 25 % → moteur lent
 - 50 % → vitesse moyenne
 - 75 % → moteur rapide
 - 100 % → vitesse maximale
- ◆ Driver (ex. L293D, L298N)
 - Entrées logiques : **IN1, IN2** → sens de rotation.

- Entrée **EN** (Enable) → reçoit le signal PWM pour moduler la vitesse.
- Sorties → reliées aux bornes du moteur.
- ♦ Description du system
 - BP1 : marche avant (horaire).
 - BP2 : marche arrière (antihoraire).
 - BP1 + BP2 : arrêt.
 - Potentiomètre ou switchs : réglage du rapport cyclique PWM (donc de la vitesse).

Travail demandé

- Commande du moteur
Associer IN1, IN2 aux boutons poussoirs (selon la table de vérité).
Relier pwm_out à l'entrée **EN** du driver.
- **Écrire un module VHDL** qui :
Générateur PWM en VHDL (fréquence fixe, rapport cyclique variable par un switch à 2 bits) Switch (1..2) qui fonctionne selon le tableau suivant :

Switch 1	Switch 2	Cycle PWM
0	0	25 %
0	1	50 %
1	0	75 %
1	1	100 %

Lit les entrées BP1 (bouton 1), BP2 (bouton 2) et
Génère les sorties IN1 (LED 1), IN2 (LED 2) (selon la table de vérité tableau 1)
- Intégrer ce signal PWM dans la commande du moteur via l'entrée **EN** (Buzzer) du driver.
- Génère PIN planner
- Simuler le système pour vérifier la variation de vitesse avec le PWM varie bien avec la valeur de duty ainsi que changement de sens de rotation selon BP1/BP2
- Programmer le CPLD/FPGA et tester sur le moteur réel.

ANNEX

#-----GLOBAL-----#

set_location_assignment	PIN_12	-to	clk
set_location_assignment	PIN_41	-to	rst_n

#-----LED-----#

set_location_assignment	PIN_67	-to	led[1]
set_location_assignment	PIN_66	-to	led[2]
set_location_assignment	PIN_61	-to	led[3]
set_location_assignment	PIN_58	-to	led[4]
set_location_assignment	PIN_57	-to	led[5]
set_location_assignment	PIN_56	-to	led[6]
set_location_assignment	PIN_55	-to	led[7]
set_location_assignment	PIN_54	-to	led[8]

#-----Bouton-----#

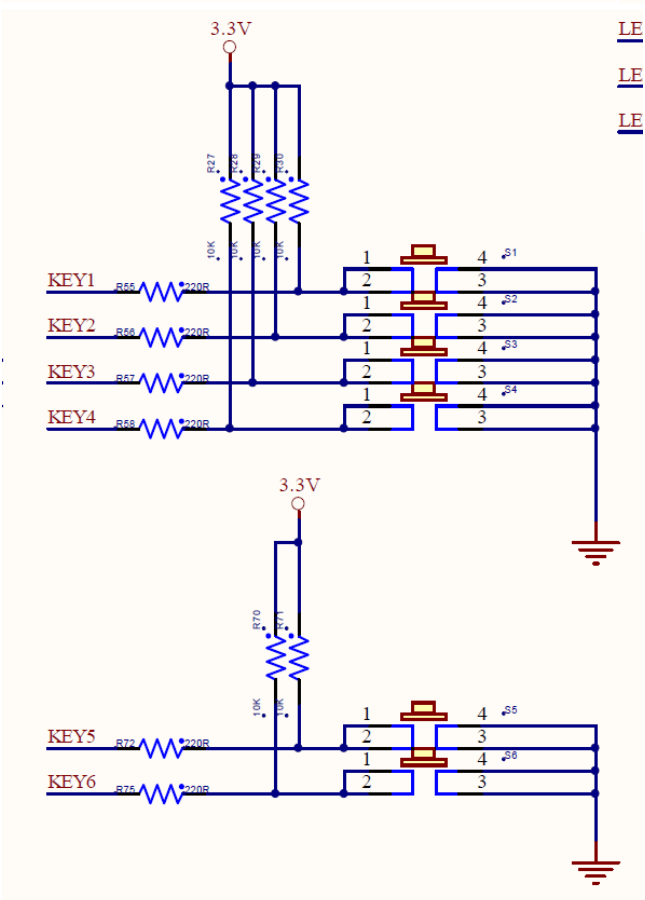
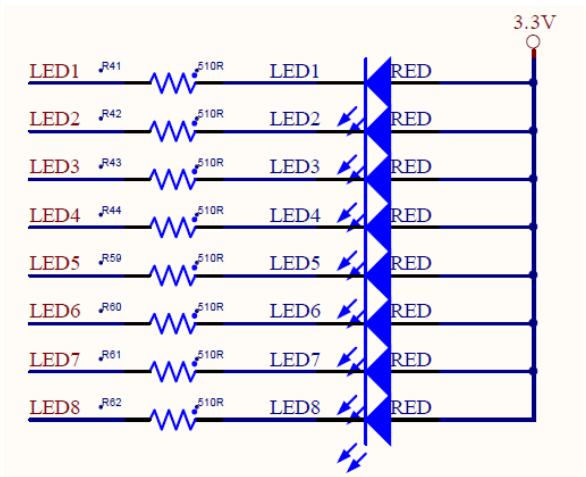
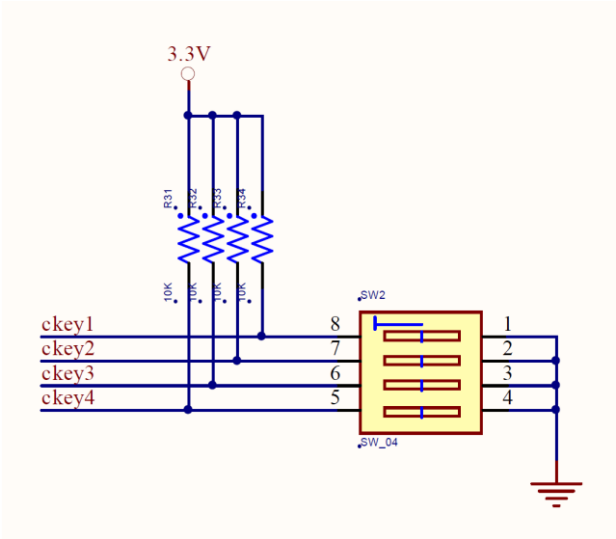
set_location_assignment	PIN_49	-to	key[1]
set_location_assignment	PIN_48	-to	key[2]
set_location_assignment	PIN_47	-to	key[3]
set_location_assignment	PIN_44	-to	key[4]
set_location_assignment	PIN_43	-to	key[5]
set_location_assignment	PIN_42	-to	key[6]

#-----Switch-----#

set_location_assignment	PIN_53	-to	ckey[1]
set_location_assignment	PIN_52	-to	ckey[2]
set_location_assignment	PIN_51	-to	ckey[3]
set_location_assignment	PIN_50	-to	ckey[4]

#-----Buzzer-----#

set_location_assignment	PIN_73	-to	beep
-------------------------	--------	-----	------



LE
LE
LE

