

Conception de Système embarqué sur FPGA

Hervé MATHIAS

C2N

UNIVERSITÉ PARIS SUD

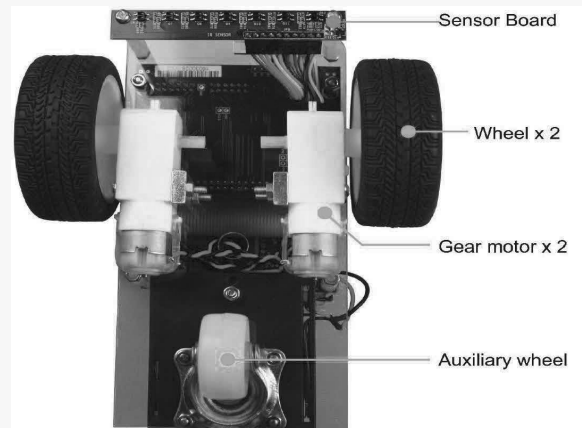
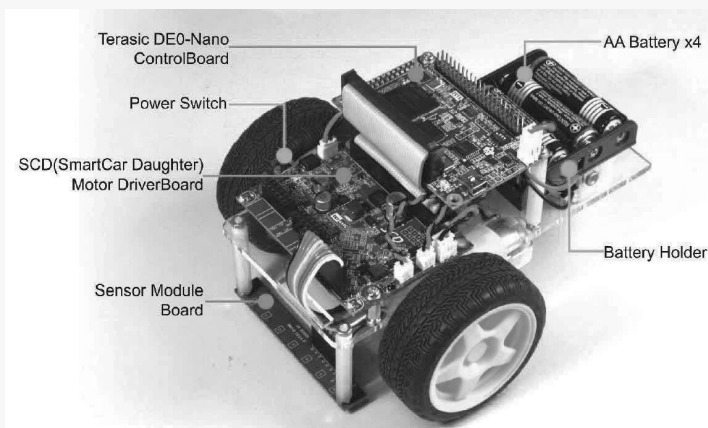
hervemathias@universite-paris-saclay.fr

Conception d'un système embarqué de commande de robot

1. Présentation du Robot
2. Description des différents composants
3. Parcours de labyrinthe
4. Système à concevoir

Conception d'un système de commande de Robot

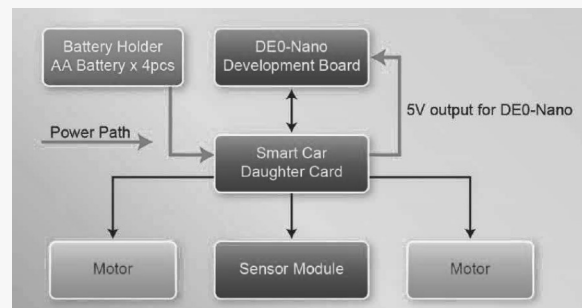
Présentation du Robot



3 cartes :

- DE0 Nano (FPGA)
- Carte Robot
- Carte Capteurs

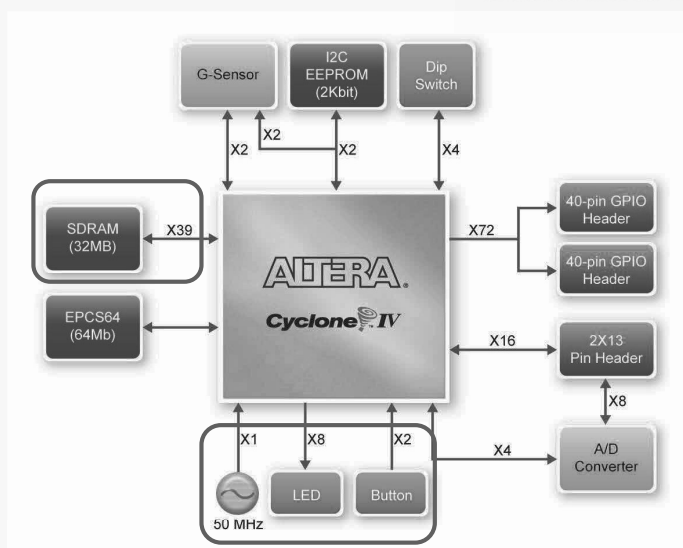
Deux roues motrices commandées par des moteurs à courant continu indépendants



Conception d'un système de commande de Robot

Présentation du Robot

Carte DE0 Nano



Contient un FPGA (Cyclone IV EP4CE22F17C6N) associé à plusieurs périphériques : SDRAM (32 Mo), LEDs, Boutons poussoirs ...

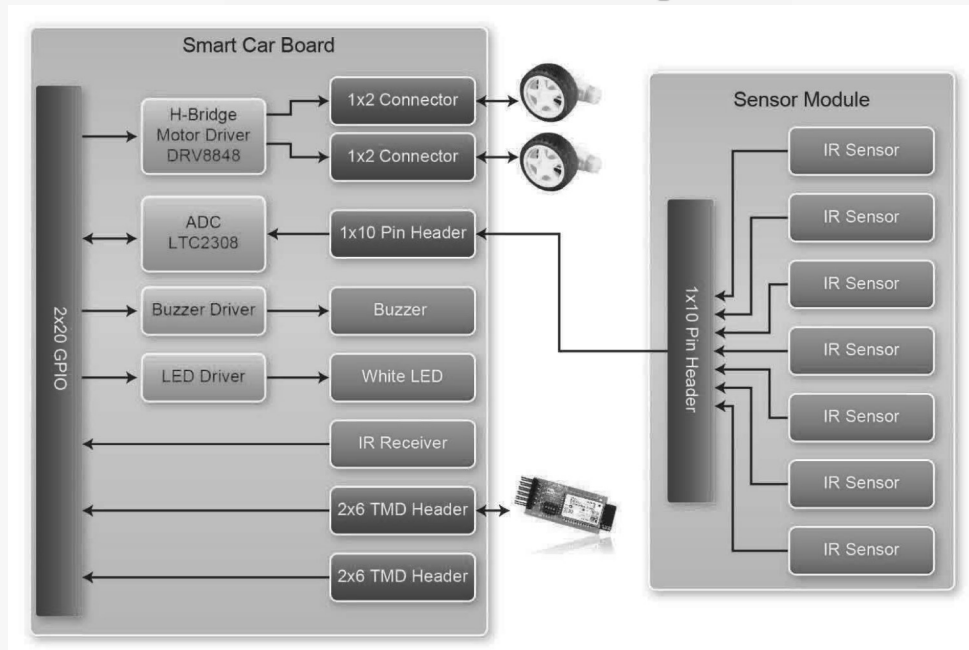
Resources	EP4CE22
Logic elements (LEs)	22,320
Embedded memory (Kbits)	594
Embedded 18 × 18 multipliers	66
General-purpose PLLs	4
Global Clock Networks	20
User I/O Banks	8
Maximum user I/O ⁽¹⁾	153

Possibilité d'implanter un système embarqué à base de processeur NIOS au sein du FPGA via l'outil Qsys

Conception d'un système de commande de Robot

Présentation du Robot

Carte Robot + Carte Capteurs



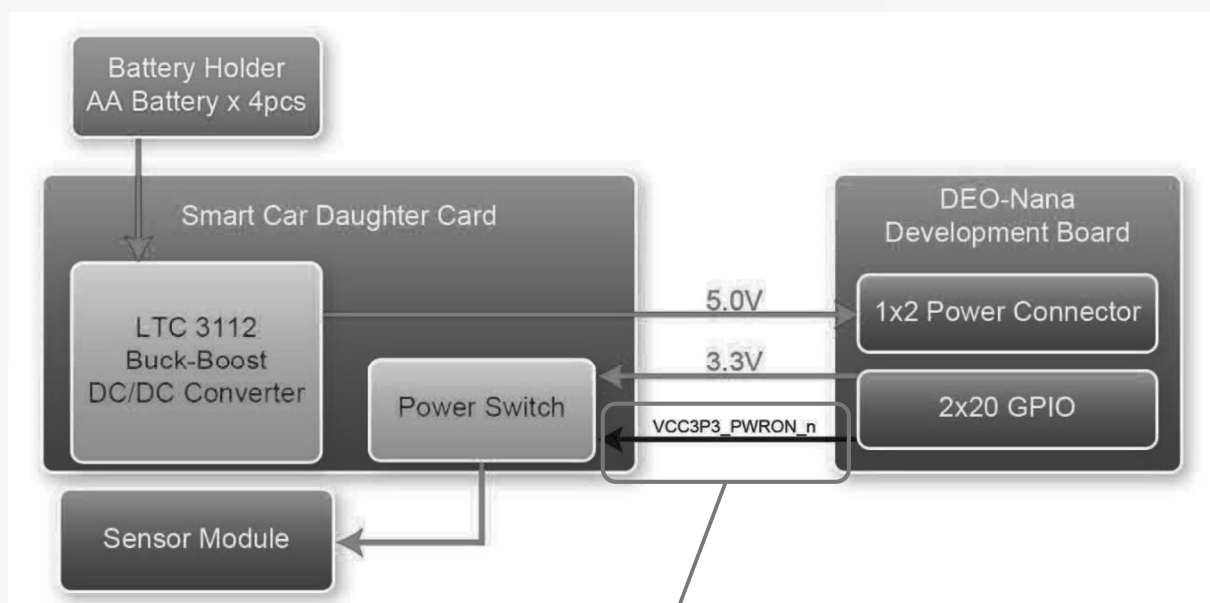
Carte Robot : Buzzer + 2 LEDs + ADC + ponts en H pour commander les moteurs + Récepteur Infrarouge (télécommande)

Carte Capteurs : 7 capteurs infrarouge

Conception d'un système de commande de Robot

Présentation du Robot

Gestion des alimentations



Signal **VCC3P3_PWRON_n** à positionner à 0 pour alimenter la carte robot

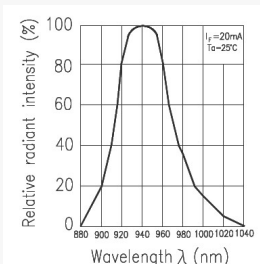
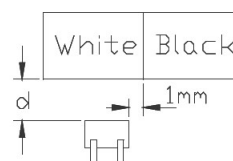
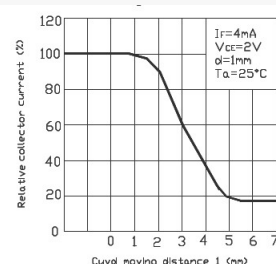
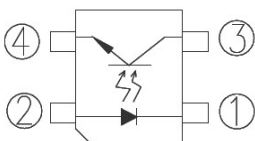
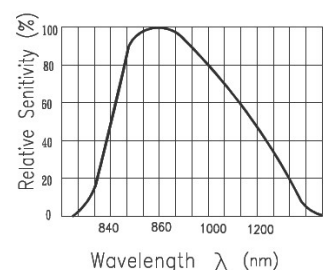
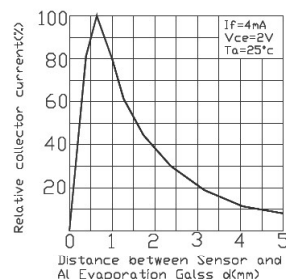
Conception d'un système embarqué de commande de robot

1. Présentation du Robot
2. Description des différents composants
3. Parcours de labyrinthe
4. Système à concevoir

Conception d'un système de commande de Robot

Description des composants

Capteurs IR



Phototransistor captant le signal IR réfléchi de la photodiode

Driver moteur génère les signaux de commande du pont en H à partir des signaux PWM en entrée

Il faut envoyer sur SDI la future configuration de conversion (6 bits) pendant que l'on récupère sur SDO les 12 bits de la conversion actuelle

Conception d'un système de commande de Robot

Description des composants

Convertisseur Analogique/Numérique LTC2308 – Liaison série

Rôle des 6 bits SDI

S/D	O/S	S1	S0	UNI	SLP
-----	-----	----	----	-----	-----

S/D = SINGLE-ENDED/DIFFERENTIAL BIT

O/S = ODD/SIGN BIT

S1 = ADDRESS SELECT BIT 1

S0 = ADDRESS SELECT BIT 0

UNI = UNIPOLAR/BIPOLAR BIT

SLP = SLEEP MODE BIT

Possibilité de convertir des signaux
référencés à la masse ou différentiels

Table 1. Channel Configuration

S/D	O/S	S1	S0	0	1	2	3	4	5	6	7	COM
0	0	0	0	+	-							
0	0	0	1			+	-					
0	0	1	0					+	-			
0	0	1	1							+	-	
0	1	0	0	-	+							
0	1	0	1			-	+					
0	1	1	0					-	+			
0	1	1	1							-	+	
1	0	0	0	+								-
1	0	0	1			+						-
1	0	1	0					+				-
1	0	1	1							+		-
1	1	0	0		+							-
1	1	0	1				+					-
1	1	1	0						+			-
1	1	1	1								+	-

Conception d'un système embarqué de commande de robot

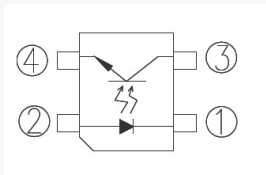
1. Présentation du Robot
2. Description des différents composants
3. Parcours de labyrinthe
4. Système à concevoir

Conception d'un système de commande de Robot

Parcours de labyrinthe : Position de la ligne

Exploitation des 7 capteurs Infra rouge

x7



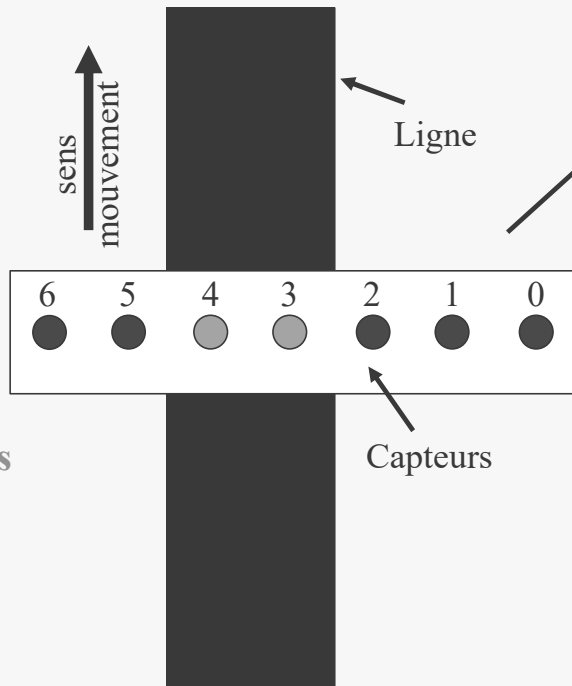
Phototransistor captant le signal IR réfléchi de la photodiode

Ligne en dessous

→ 1

Pas de ligne

→ 0



Calcul de la position

0 0 1 1 0 0 0

PDU

PPU

Position premier un : PPU = 3

Position dernier un : PDU = 4

$pos_ligne = PPU + PDU - 6 = 1$

Valeurs possibles : de -6 à +6

0 si ligne au milieu

Conception d'un système de commande de Robot

Parcours de labyrinthe : Commande des moteurs

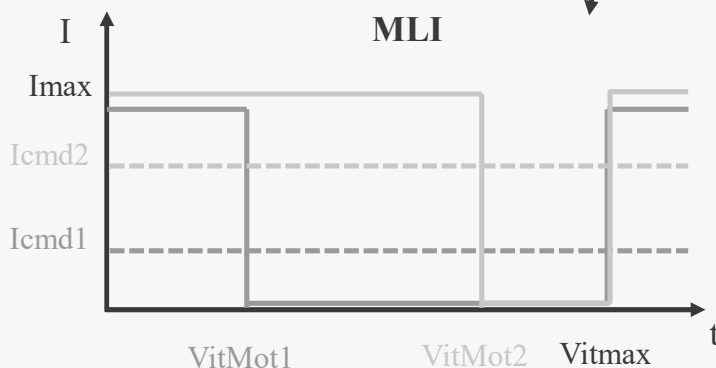
Contrôle vitesse moteurs par Modulation à Largeur d'Impulsion (MLI)

Commande d'un moteur



Go = 1 : marche
Go = 0 : arrêt

Dir = 1 : arrière
Dir = 0 : avant



$$Icmd = I_{max} * (VitMot/Vitmax)$$

La vitesse du moteur est proportionnelle à Icmd

Calcul de la vitesse

$$VitMotD = VitMoy + pos_ligne * K$$

$$VitMotG = VitMoy - pos_ligne * K$$

Si DirMotG = DirMotD = 0

Si VitMotD = VitMotG

Robot va tout droit

Si VitMotD > VitMotG

Robot tourne vers la gauche

Si VitMotD < VitMotG

Robot tourne vers la droite

Si DirMotG = 0 et DirMotD = 1

Si VitMotD = VitMotG

Robot fait demi-tour sur place

Conception d'un système embarqué de commande de robot

1. Présentation du Robot
2. Description des différents composants
3. Parcours de labyrinthe
4. Système à concevoir

Conception d'un système embarqué

Présentation du système de commande robot

Système pour aller-retours avec suivi de ligne

Le processeur NIOS donne des ordres à des automates pour le suivi de ligne ou les demi-tours

