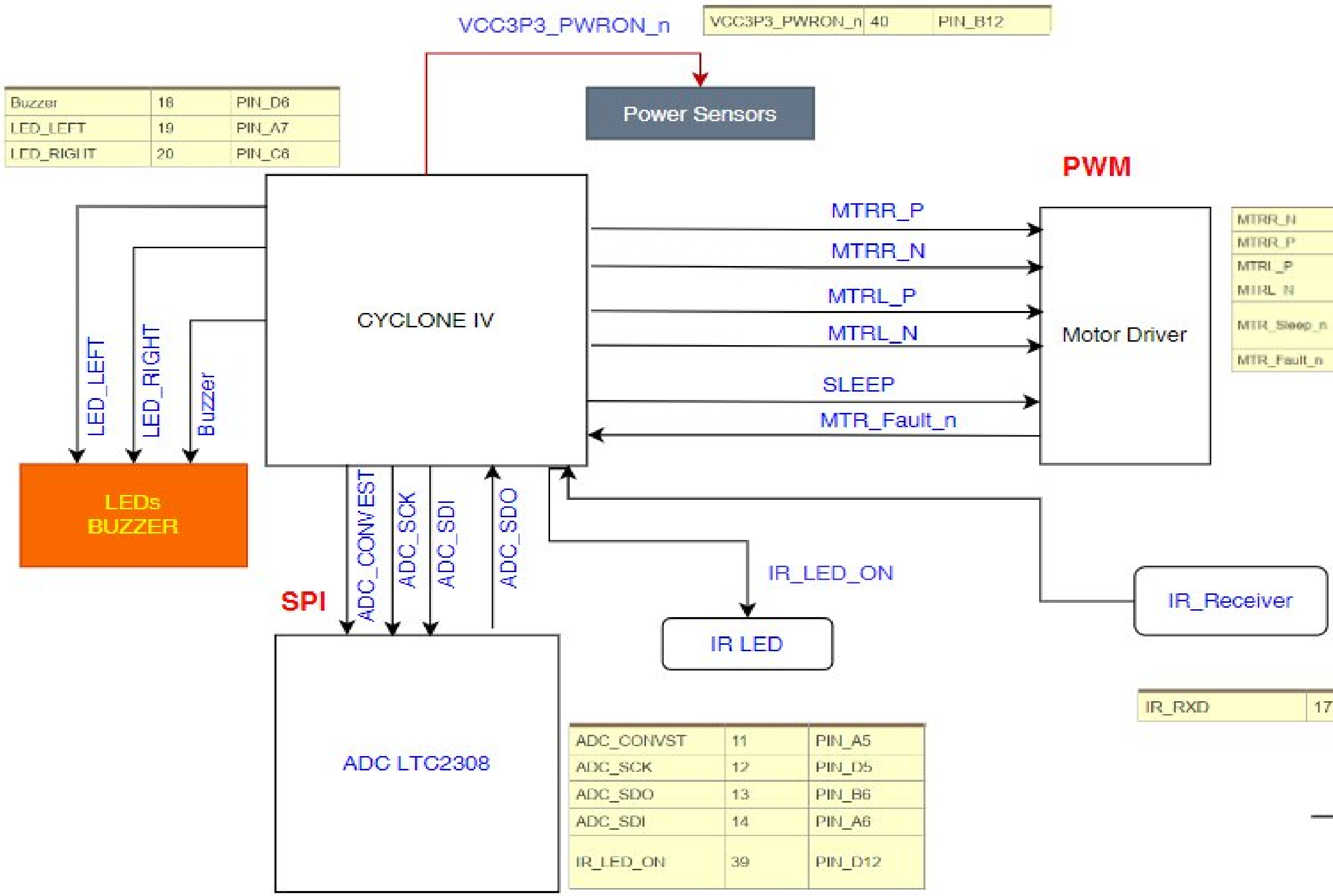
TP A-Cute Car

Guide



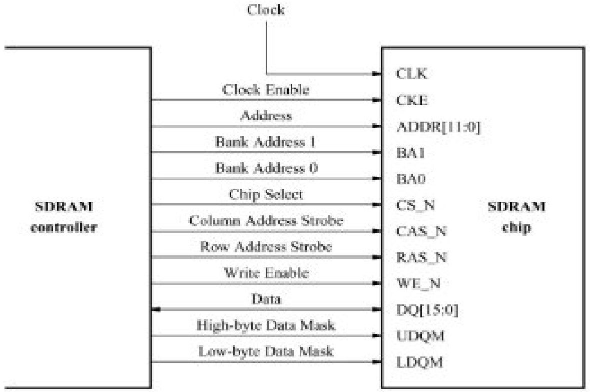
VCC3P3_PWRON_n	40	PIN_B12

Buzzer	18	PIN_D6
LED_LEFT	19	PIN_A7
LED_RIGHT	20	PIN_C8

MTRR_N	5	PIN_A2
MTRR_P	6	PIN_A3
MTRL_P	7	PIN_B3
MTRL_N	8	PIN_B4
MTR_Sleep_n	9	PIN_A4
MTR_Fault_n	10	PIN_B5

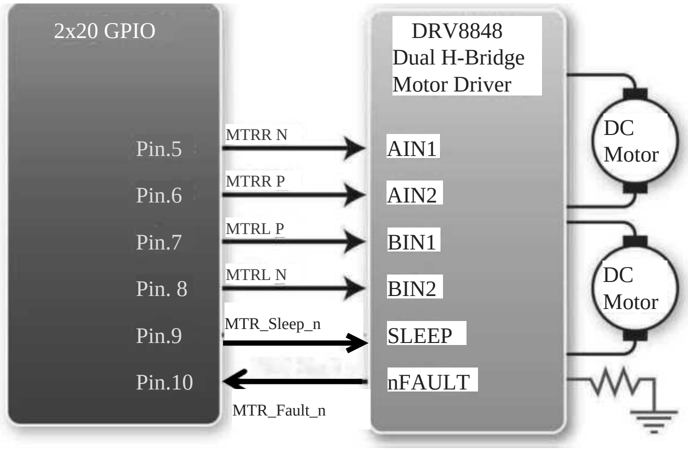
ADC_CONVST	11	PIN_A5
ADC_SCK	12	PIN_D5
ADC_SDO	13	PIN_B6
ADC_SDI	14	PIN_A6
IR_LED_ON	39	PIN_D12

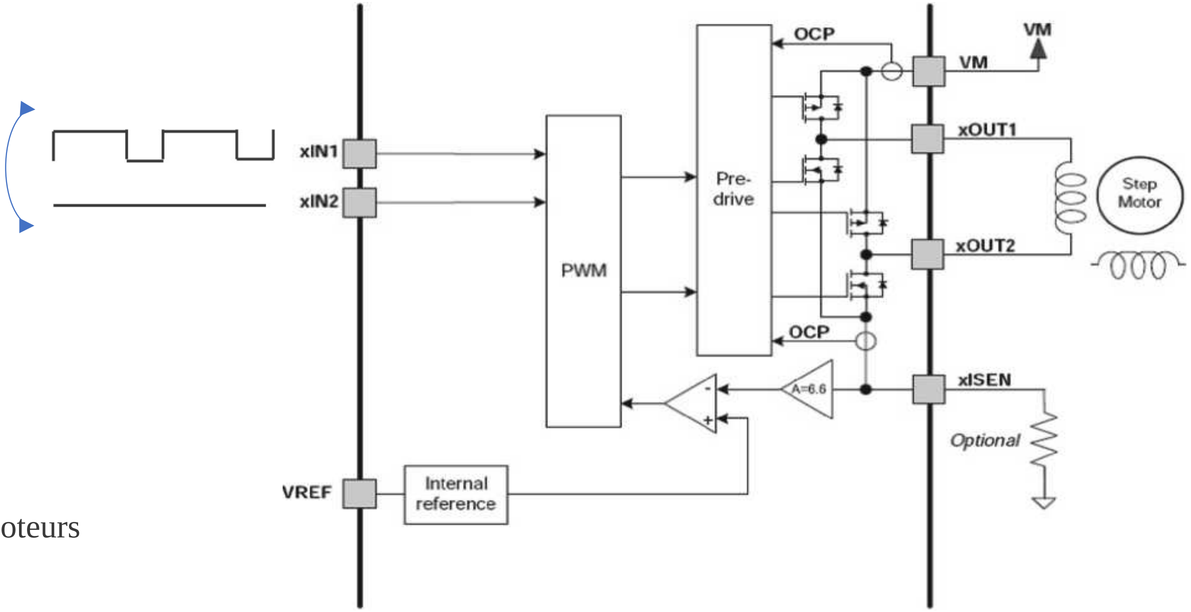
IR_RXD	17	PIN_B7



Pilotage Moteur

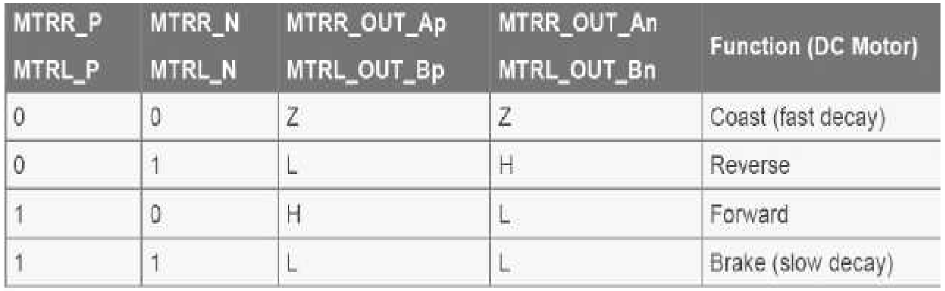
Circuit de commande des moteurs Ponts en H - DRV8848





Signal **MTR_Sleep_n** à positionner à 1 pour mettre en route les moteurs

Signaux **MTRx_N** et **MTRx_P** à générer à l'aide d'un générateur PWM



MTRR_P	MTRR_N	MTRR_OUT_Ap	MTRR_OUT_An	Function (DC Motor)
MTRL_P	MTRL_N	MTRL_OUT_Bp	MTRL_OUT_Bn	
0	0	Z	Z	Coast (fast decay)
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	L	L	Brake (slow decay)

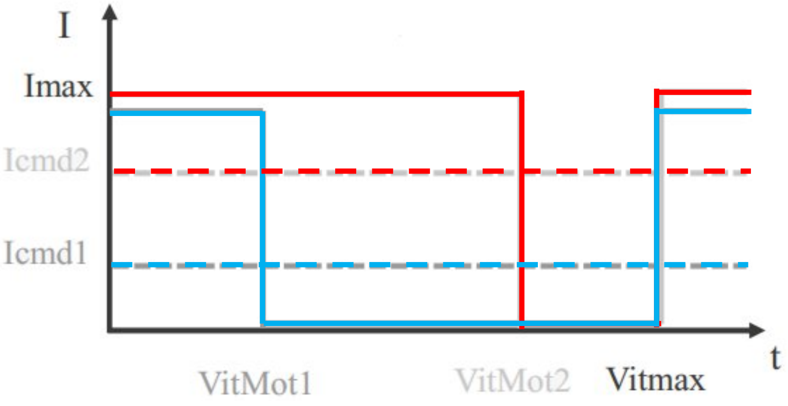
Driver moteur génère les signaux de commande du pont en H à partir des signaux PWM en entrée

Contrôle vitesse moteurs par Modulation à Largeur d’impulsion

Commande d’un moteur

Go	Dir	VitMot(12bits)

Go = 1 : Marche Dir = 1 : Arrière
Go = 0 : Arrêt Dir = 0 : Avant



Icmd = Imax * (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

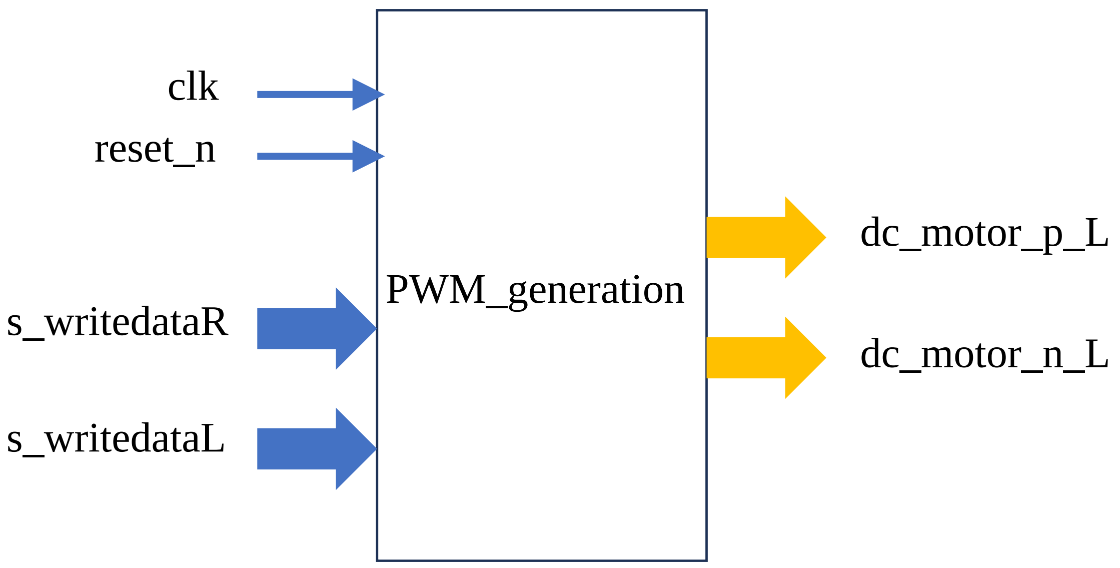
Si VitMotD > VitMotG
Robot tourne vers la gauche

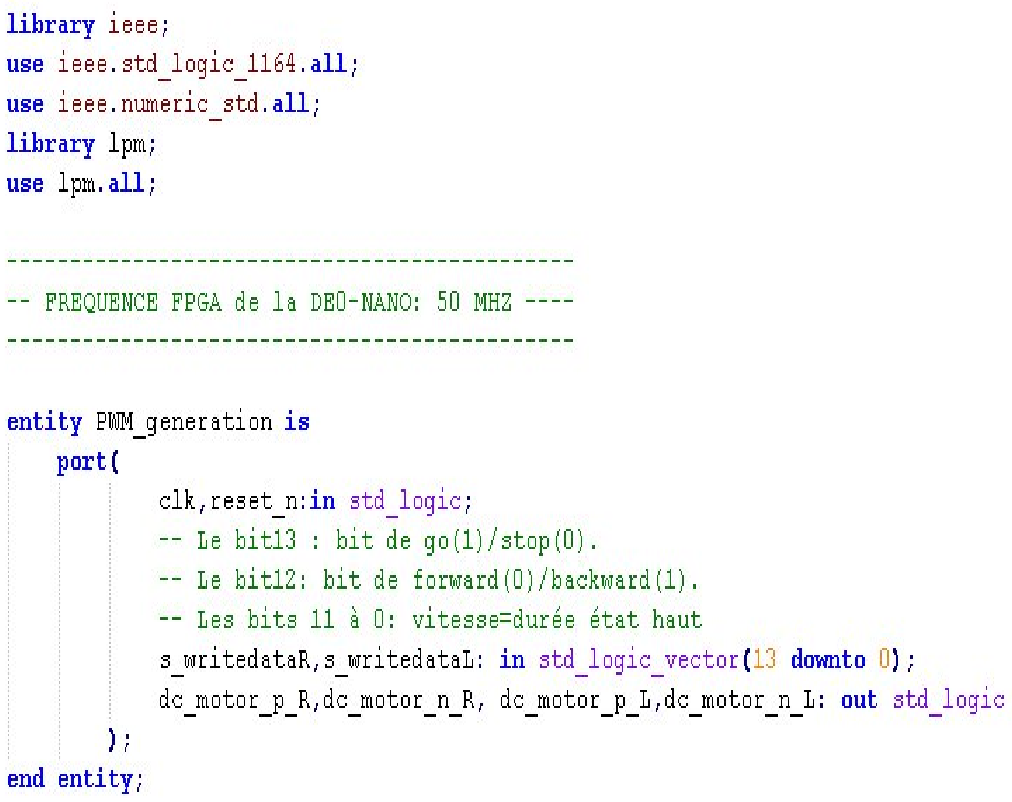
Si VitMotD < VitMotG
Robot tourne vers la droite

Si DirMotG = 0 et DirMotD = 1

Si VitMotD = VitMotG Robot fait
Robot fait demi-tour sur place

Module VHDL générateur PWM

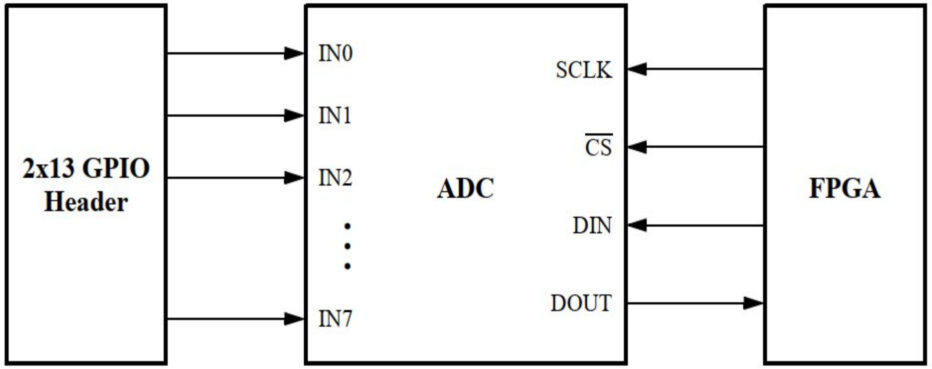



```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
library lpm;
use lpm.all;

---------------------------------------------
-- FREQUENCE FPGA de la DE0-NANO: 50 MHZ ----
---------------------------------------------

entity PWM_generation is
    port(
            clk,reset_n:in std_logic;
            -- Le bit13 : bit de go(1)/stop(0).
            -- Le bit12: bit de forward(0)/backward(1).
            -- Les bits 11 à 0: vitesse=durée état haut
            s_writedataR,s_writedataL: in std_logic_vector(13 downto 0);
            dc_motor_p_R,dc_motor_n_R, dc_motor_p_L,dc_motor_n_L: out std_logic
        );
end entity;
```

ADC de la DE0-NANO : ADC128S002



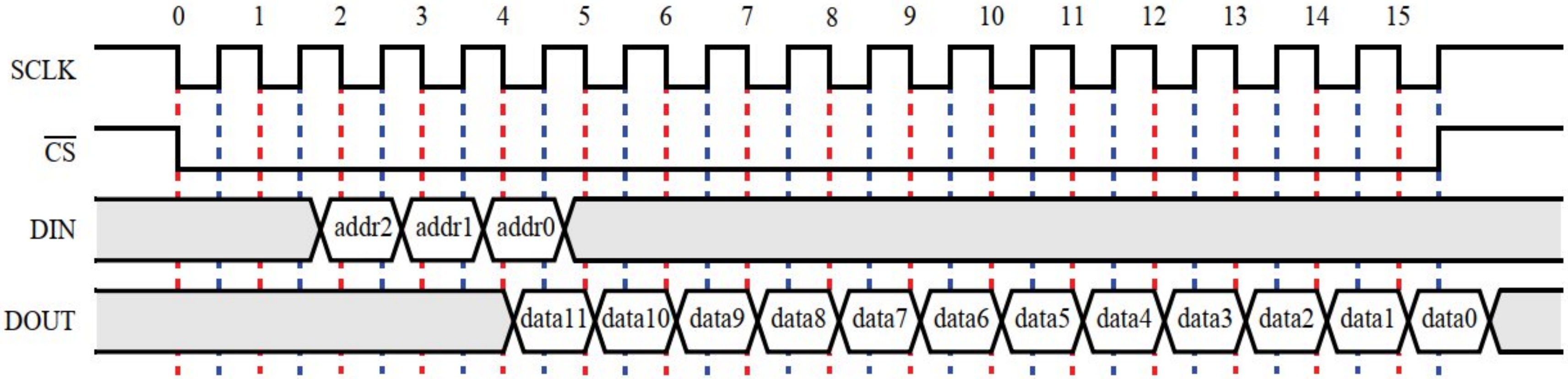
Signals to and from the ADC

8 analog input pins IN0 through IN7. ADC reads and convert the signal on one of these eight input channels

SCLK serves as a device clock
CS select the ADC chip.
The DIN and DOUT wires are used for transferring addresses and data between the two chips.
DIN is mapped to the ADC_SADDR. Provide the address of the **next channel requested for conversion**.
The address is 3 bits in length and is sent to the ADC serially at a rate of 1 bit per SCLK cycle.

DOUT is mapped to the ADC_SDAT and used to send the digital value (12 bits long) of the converted signal.
Sent to the FPGA in a serial manner at a rate of 1 bit per SCLK cycle

Timing



Timing requirements for the ADC

A réaliser

A- On se propose d’étudier les deux modules :

- l’IP Core fourni par Altera interfaçant l’ADC ADC128S002 au bus Avalon (cartes DE0-nano) (module 1)
- le bloc VHDL capteurs_sol_seuil (module 2) pilotant l’ADC LTC2803

1- Implémenter et mettre en oeuvre le module 1 (Acquérir deux signaux 0 et 3.3 v)

2- Transformer le module 2 en IP core pour qu’il soit interfacé au bus Avalon (Acquérir deux signaux 0 et 3.3 v).

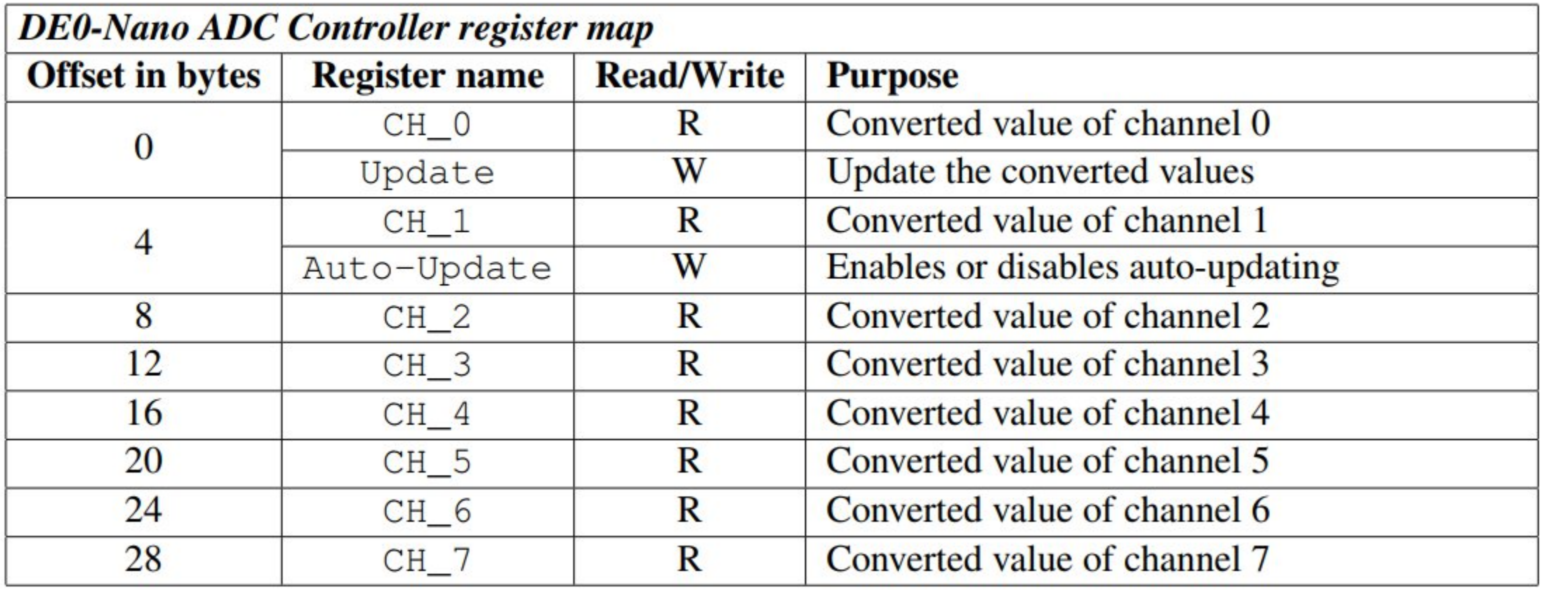
3- En utilisant l’analyse logique, retrouver les chronogrammes de fonctionnement de :

  - l’ADC LTC 2803
  - l’ADC ADC128S002

B – Comment peut-on caractériser la valeur du seuil à fixer pour le module 2

C- Transformer tous les modules fournis sous forme d’IP Core. Déduire un système exclusivement fait d’IP Core pouvant réaliser un suivi de ligne.

ADC Controller



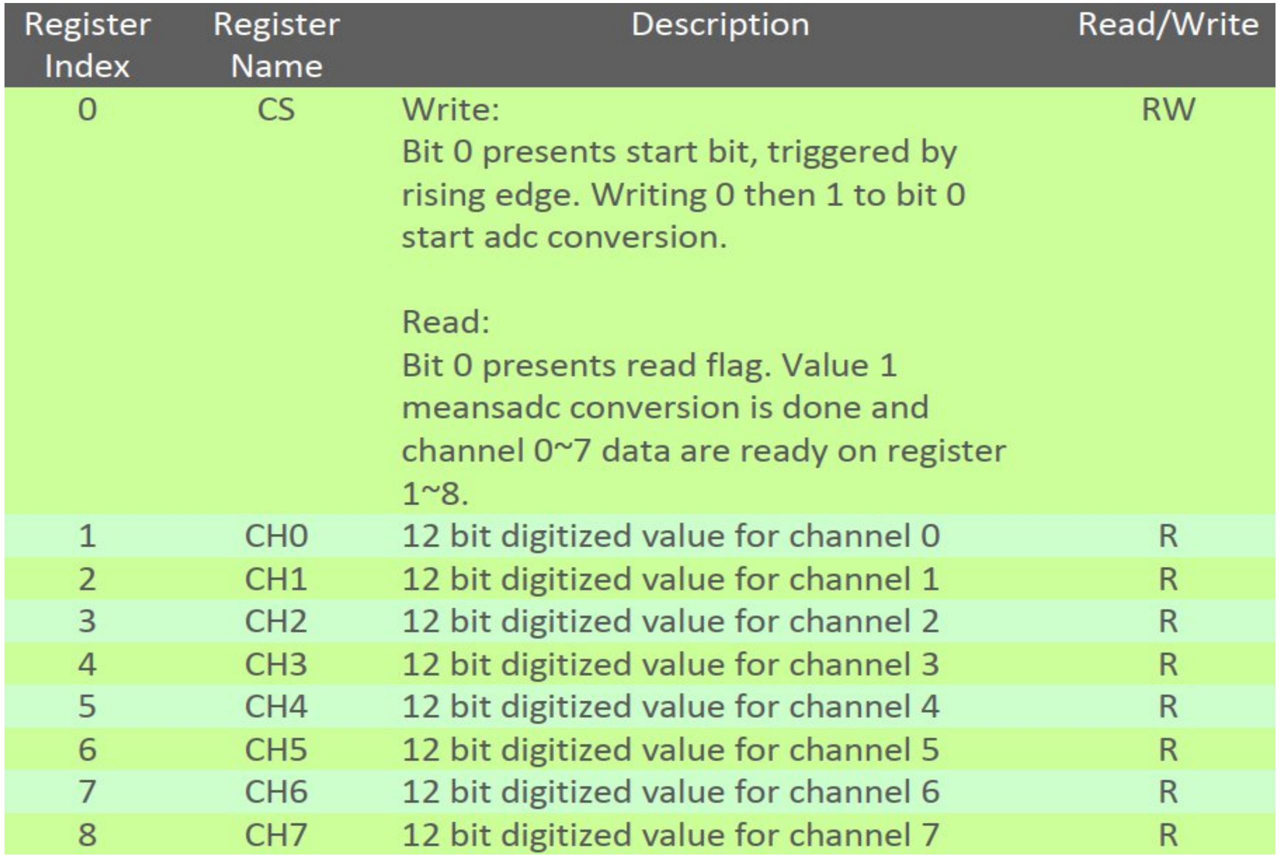
DE0-Nano ADC Controller register map

Offset in bytes	Register name	Read/Write	Purpose
0	CH_0	R	Converted value of channel 0
	Update	W	Update the converted values
4	CH_1	R	Converted value of channel 1
	Auto-Update	W	Enables or disables auto-updating
8	CH_2	R	Converted value of channel 2
12	CH_3	R	Converted value of channel 3
16	CH_4	R	Converted value of channel 4
20	CH_5	R	Converted value of channel 5
24	CH_6	R	Converted value of channel 6
28	CH_7	R	Converted value of channel 7

The Update register is used to initiate a conversion operation. Performing a write to this register will update all channels in use, with the new values becoming available once the entire conversion process has finished. If reads to the channel registers are attempted while a conversion is taking place, then the wait_request signal will be raised, causing the processor to stall until the update has finished.

The Auto-Update register is initially loaded with a zero value. The auto-update allows the system to automatically begin a second update cycle after the first finishes. As result, channel values can be accessed during an update cycle, and it is user’s responsibility to ensure the values used are up-to-date. Writing a ‘1’ to this register enables auto-update, while writing a ‘0’ disables it.

Registres ADC



Register Index	Register Name	Description	Read/Write
0	CS	Write: Bit 0 presents start bit, triggered by rising edge. Writing 0 then 1 to bit 0 start adc conversion. Read: Bit 0 presents read flag. Value 1 meansadc conversion is done and channel 0~7 data are ready on register 1~8.	RW
1	CH0	12 bit digitized value for channel 0	R
2	CH1	12 bit digitized value for channel 1	R
3	CH2	12 bit digitized value for channel 2	R
4	CH3	12 bit digitized value for channel 3	R
5	CH4	12 bit digitized value for channel 4	R
6	CH5	12 bit digitized value for channel 5	R
7	CH6	12 bit digitized value for channel 6	R
8	CH7	12 bit digitized value for channel 7	R

ADC DE0 NANO

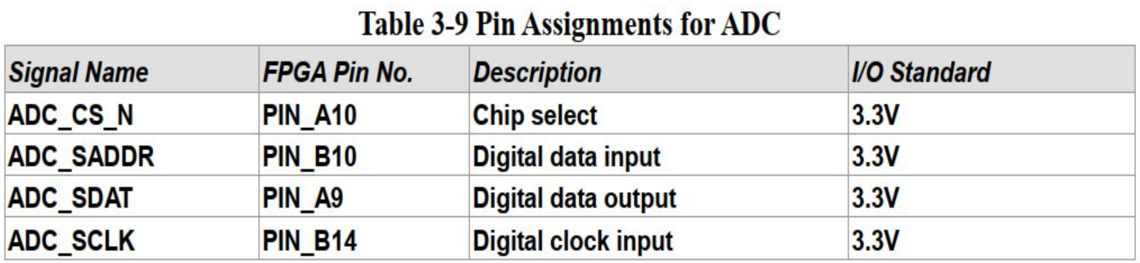
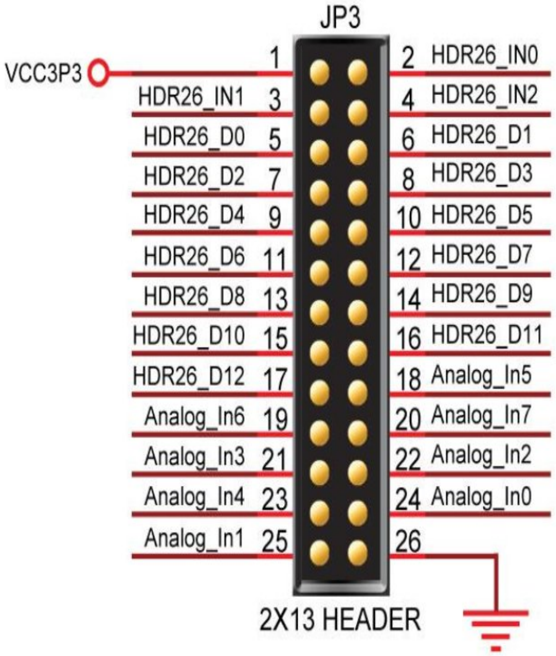
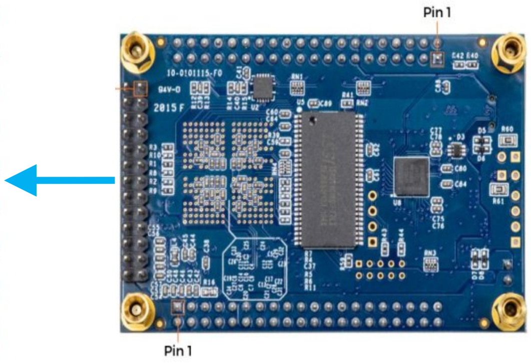
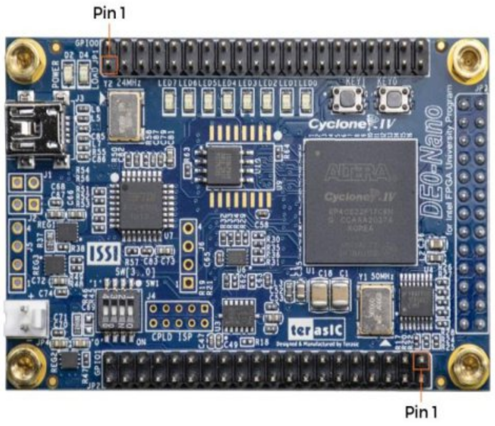


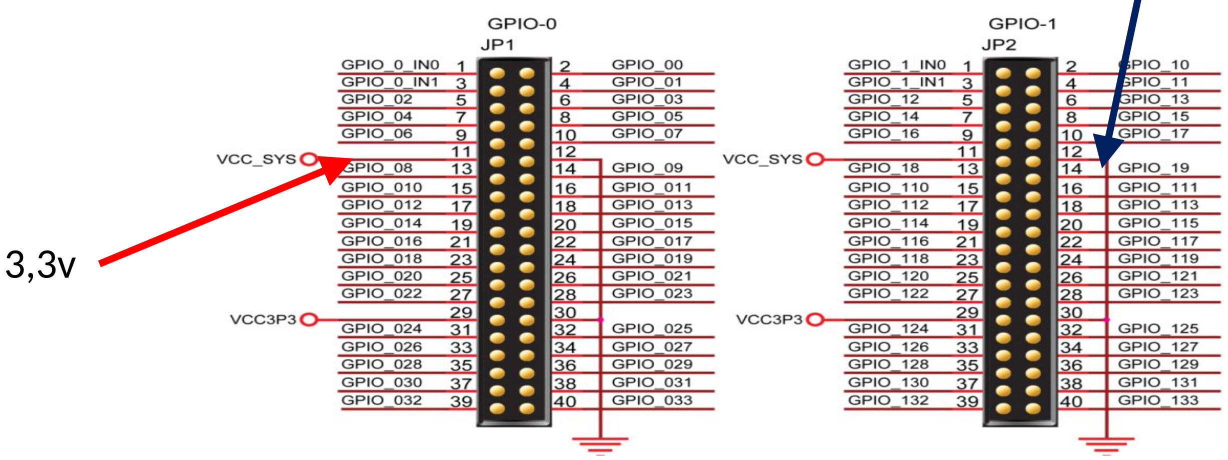
Table 3-9 Pin Assignments for ADC

Signal Name	FPGA Pin No.	Description	I/O Standard
ADC_CS_N	PIN_A10	Chip select	3.3V
ADC_SADDR	PIN_B10	Digital data input	3.3V
ADC_SDAT	PIN_A9	Digital data output	3.3V
ADC_SCLK	PIN_B14	Digital clock input	3.3V

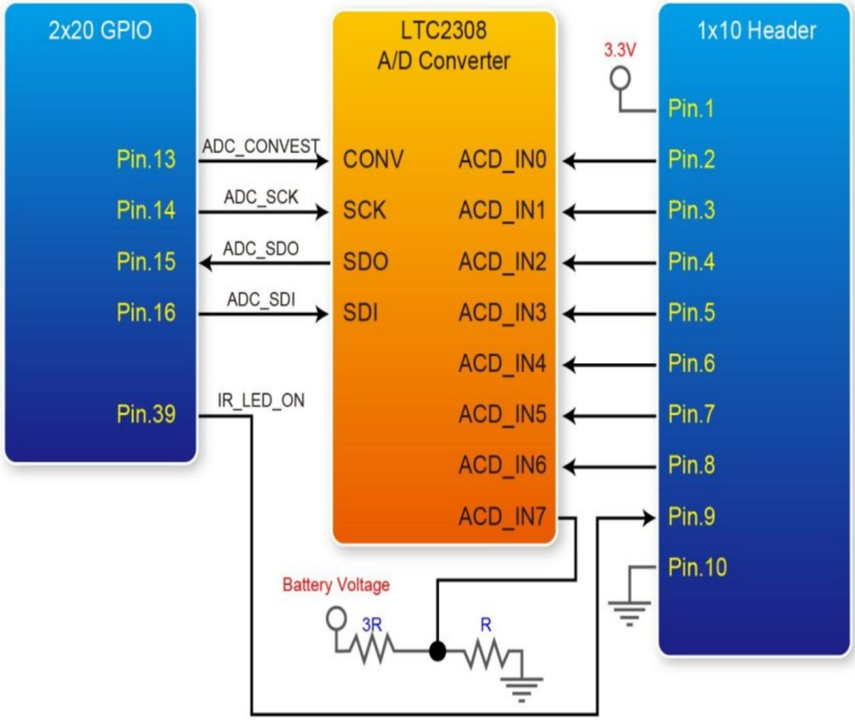

JP3
VCC3P3
1 / 2 HDR26_IN0
HDR26_IN1 3 / 4 HDR26_IN2
HDR26_D0 5 / 6 HDR26_D1
HDR26_D2 7 / 8 HDR26_D3
HDR26_D4 9 / 10 HDR26_D5
HDR26_D6 11 / 12 HDR26_D7
HDR26_D8 13 / 14 HDR26_D9
HDR26_D10 15 / 16 HDR26_D11
HDR26_D12 17 / 18 Analog_In5
Analog_In6 19 / 20 Analog_In7
Analog_In3 21 / 22 Analog_In2
Analog_In4 23 / 24 Analog_In0
Analog_In1 25 / 26
2X13 HEADER



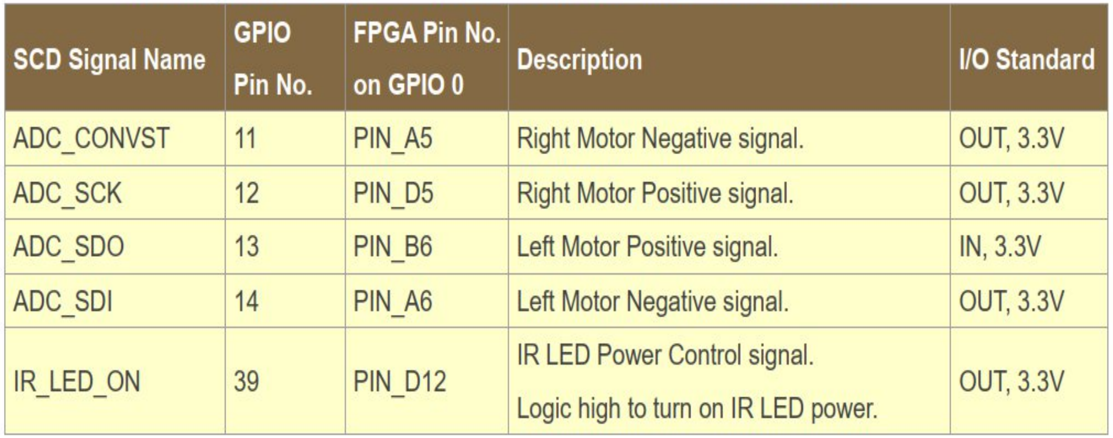



3,3v
GPIO-0
JP1
GPIO_0_IN0 1 / 2 GPIO_00
GPIO_0_IN1 3 / 4 GPIO_01
GPIO_02 5 / 6 GPIO_03
GPIO_04 7 / 8 GPIO_05
GPIO_06 9 / 10 GPIO_07
VCC_SYS 11 / 12
GPIO_08 13 / 14 GPIO_09
GPIO_010 15 / 16 GPIO_011
GPIO_012 17 / 18 GPIO_013
GPIO_014 19 / 20 GPIO_015
GPIO_016 21 / 22 GPIO_017
GPIO_018 23 / 24 GPIO_019
GPIO_020 25 / 26 GPIO_021
GPIO_022 27 / 28 GPIO_023
VCC3P3 29 / 30
GPIO_024 31 / 32 GPIO_025
GPIO_026 33 / 34 GPIO_027
GPIO_028 35 / 36 GPIO_029
GPIO_030 37 / 38 GPIO_031
GPIO_032 39 / 40 GPIO_033
GPIO-1
JP2
GPIO_1_IN0 1 / 2 GPIO_10
GPIO_1_IN1 3 / 4 GPIO_11
GPIO_12 5 / 6 GPIO_13
GPIO_14 7 / 8 GPIO_15
GPIO_16 9 / 10 GPIO_17
VCC_SYS 11 / 12
GPIO_18 13 / 14 GPIO_19
GPIO_110 15 / 16 GPIO_111
GPIO_112 17 / 18 GPIO_113
GPIO_114 19 / 20 GPIO_115
GPIO_116 21 / 22 GPIO_117
GPIO_118 23 / 24 GPIO_119
GPIO_120 25 / 26 GPIO_121
GPIO_122 27 / 28 GPIO_123
VCC3P3 29 / 30
GPIO_124 31 / 32 GPIO_125
GPIO_126 33 / 34 GPIO_127
GPIO_128 35 / 36 GPIO_129
GPIO_130 37 / 38 GPIO_131
GPIO_132 39 / 40 GPIO_133

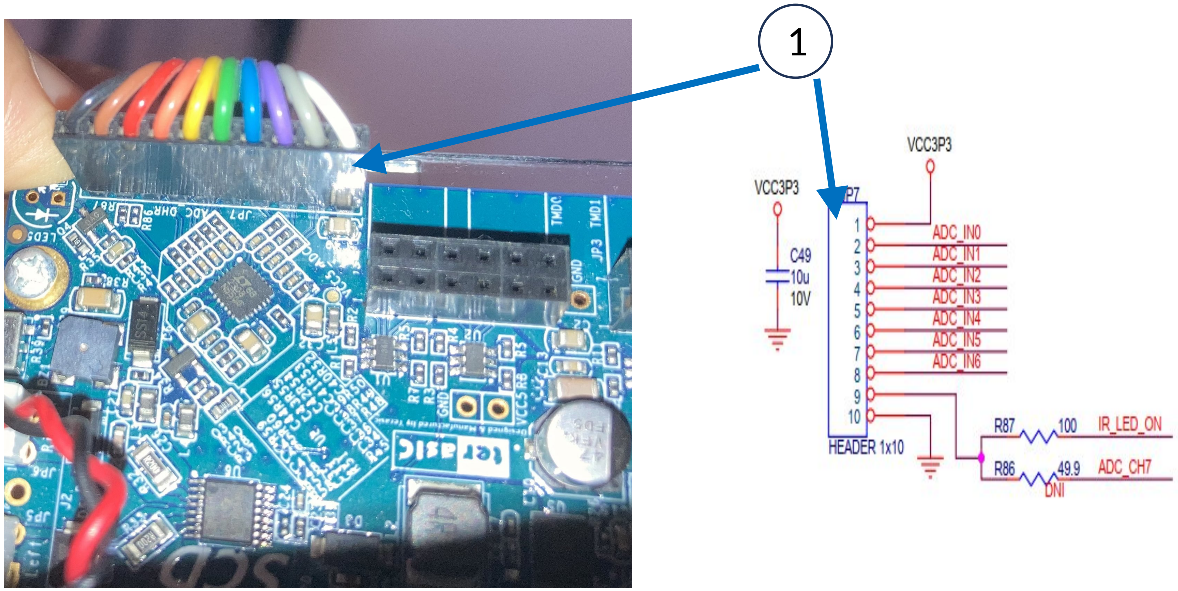
ADC A-Cute Car



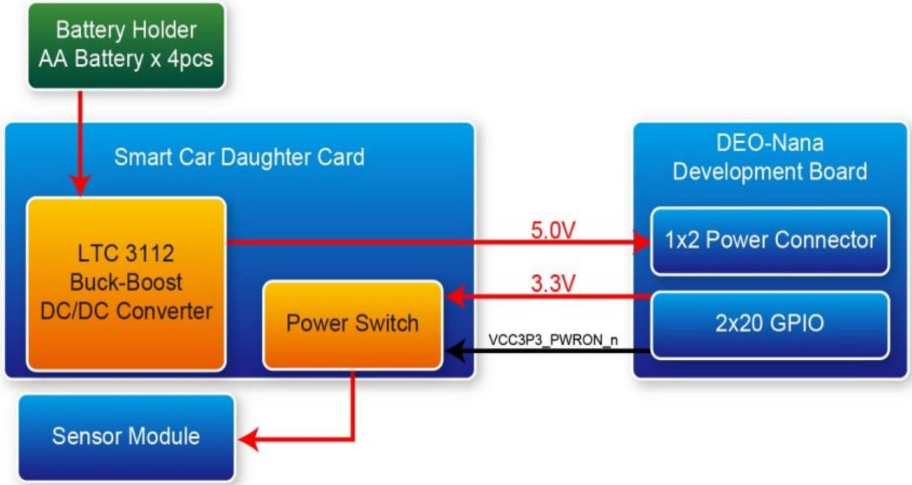
Pin assignment of ADC 4-Wire SPI interface and Power control



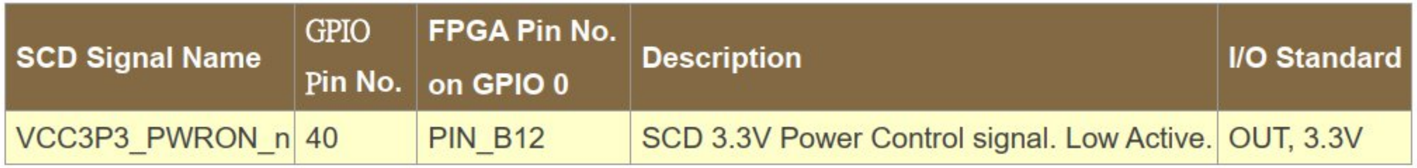
SCD Signal Name	GPIO Pin No.	FPGA Pin No. on GPIO 0	Description	I/O Standard
ADC_CONVST	11	PIN_A5	Right Motor Negative signal.	OUT, 3.3V
ADC_SCK	12	PIN_D5	Right Motor Positive signal.	OUT, 3.3V
ADC_SDO	13	PIN_B6	Left Motor Positive signal.	IN, 3.3V
ADC_SDI	14	PIN_A6	Left Motor Negative signal.	OUT, 3.3V
IR_LED_ON	39	PIN_D12	IR LED Power Control signal. Logic high to turn on IR LED power.	OUT, 3.3V



Alimentation capteur



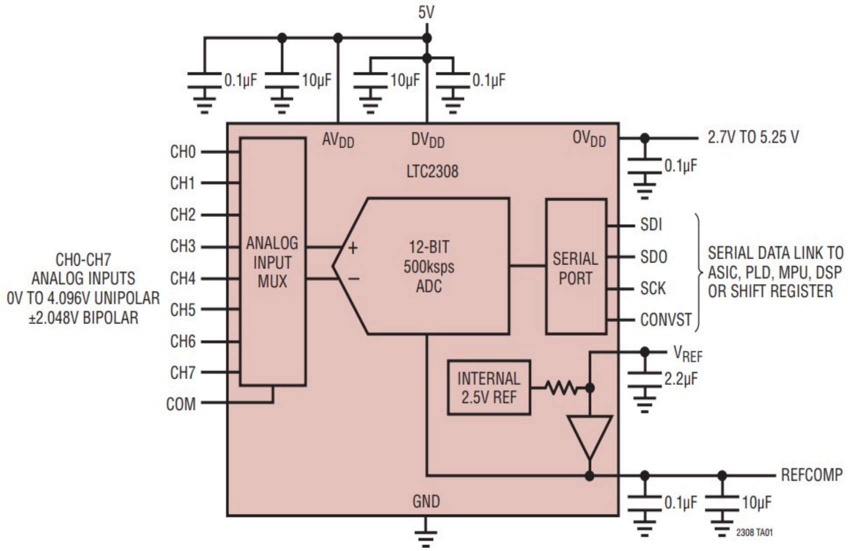
Pin Assignment of 3.3V Power Control



SCD Signal Name	GPIO Pin No.	FPGA Pin No. on GPIO 0	Description	I/O Standard
VCC3P3_PWRON_n	40	PIN_B12	SCD 3.3V Power Control signal. Low Active.	OUT, 3.3V

LTC 2308

Possibilité de convertir des signaux référencés à la masse ou différentiels



Configuration des caneaux

S/D	0/S	S1	SO	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								+	-

Rôle des 6 bits SDI

S/D	0/S	S1	SO	UNI	SLP

S/D = SINGLE-ENDED/DIFFERENTIAL BIT

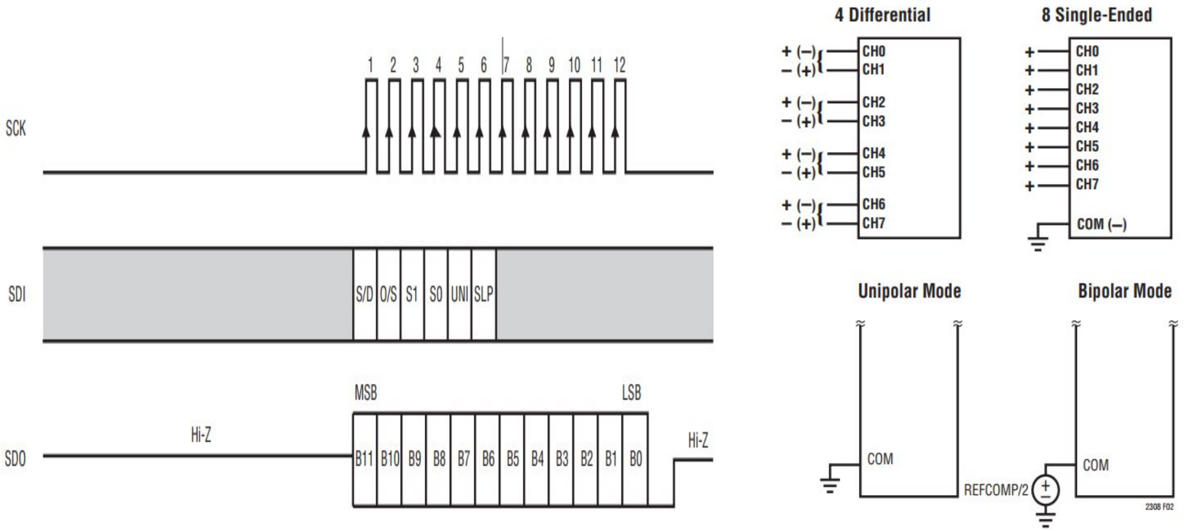
0/S = ODD/SIGN BIT

S1 = ADDRESS SELECT BIT 1

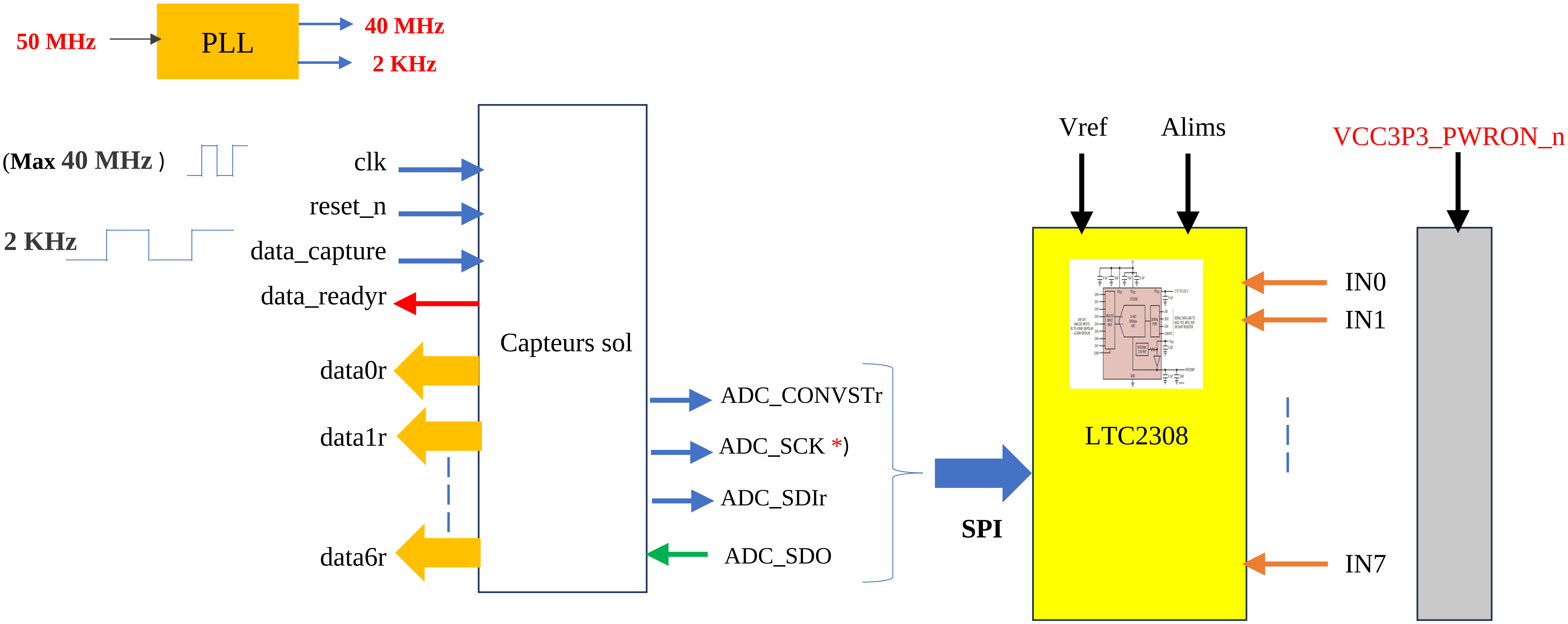
SO = ADDRESS SELECT BIT 0

UNI = UNIPOLAR/BÏPÔLÂR BIT

SLP = SLEEP MODE BIT



Module VHDL capteurs sol



50 MHz

PLL

40 MHz

2 KHz

(**Max** **40 MHz**)

2 KHz

clk

reset_n

data_capture

data_readyr

data0r

data1r

data6r

Capteurs sol

ADC_CONVSTr

ADC_SCK *)

ADC_SDIr

ADC_SDO

SPI

Vref

Alims

VCC3P3_PWRON_n

LTC2308

IN0

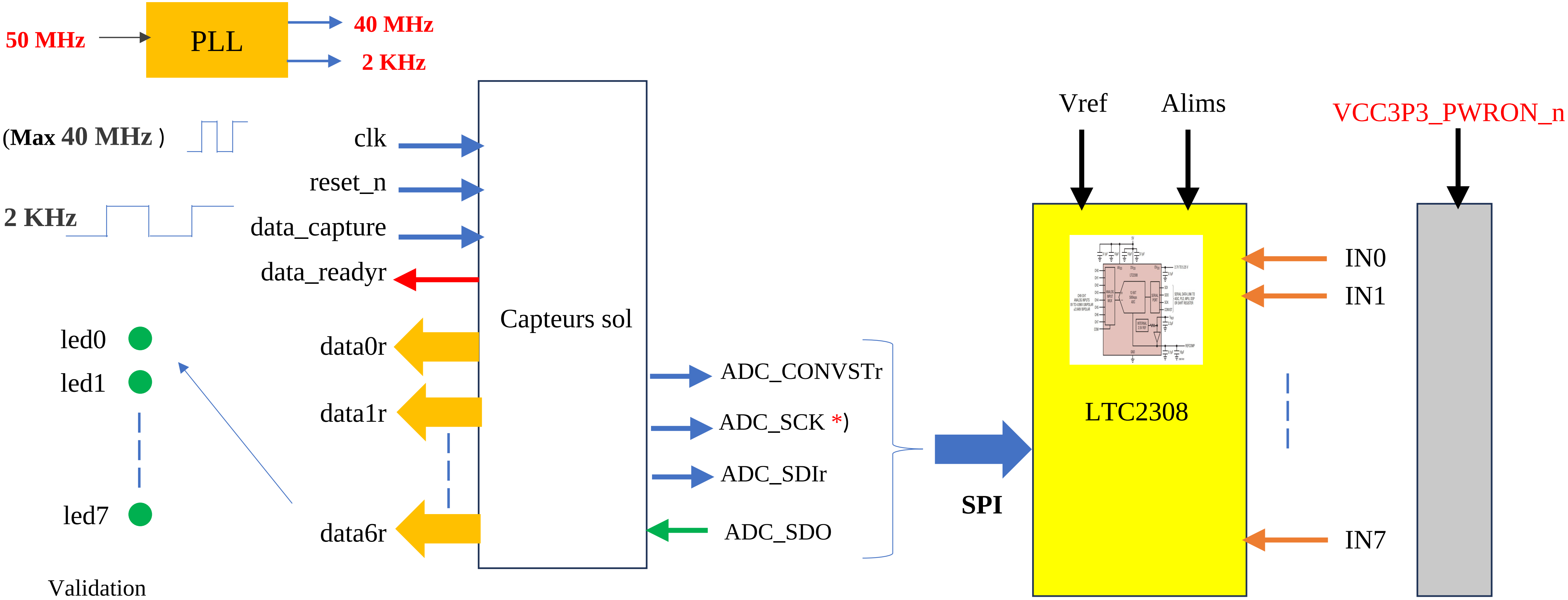
IN1

IN7

*(**Max** **40 MHz**)

ADC_SCK = $\overline{\overline{\text{clk}}}$

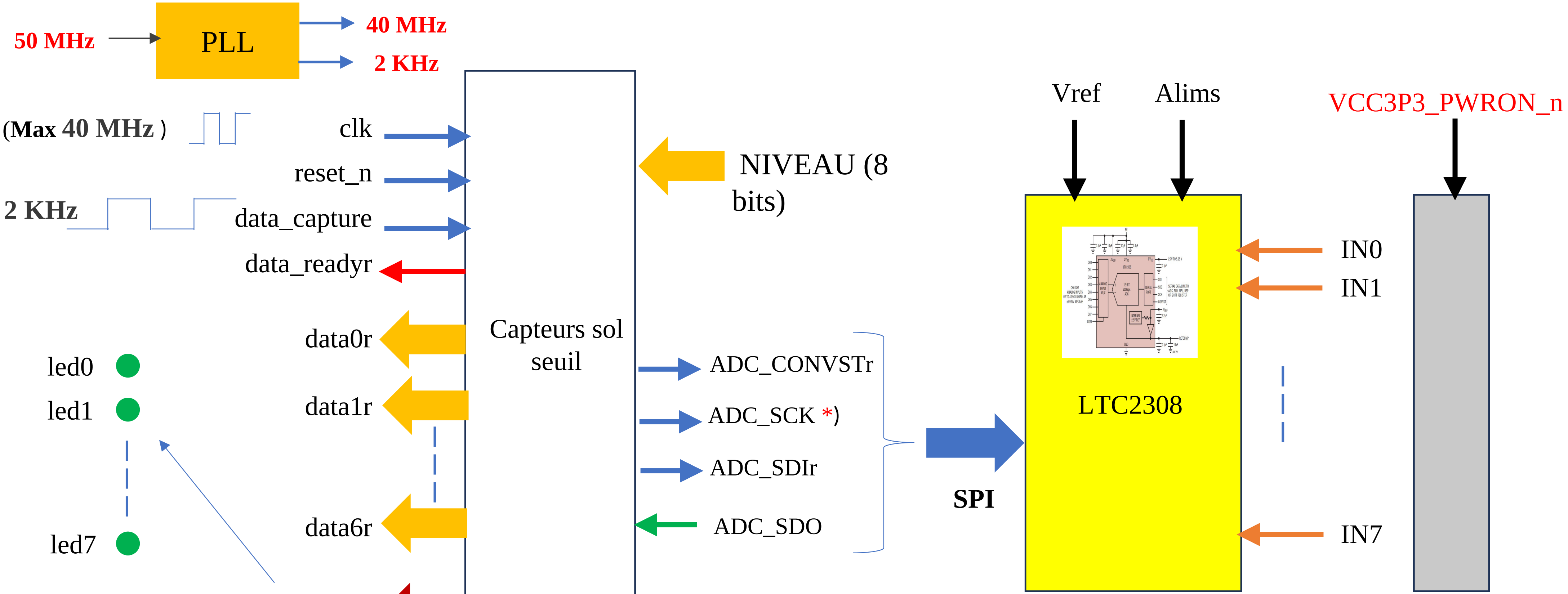
Module VHDL capteurs sol



Validation
- visuelle (Leds)
- analyse logique

*(Max 40 MHz)
ADC_SCK = $\overline{\overline{\text{clk}}}$

Module VHDL capteurs sol seuil



50 MHz

PLL

40 MHz

2 KHz

(**Max** 40 MHz)

clk

reset_n

2 KHz

data_capture

data_readyr

NIVEAU (8 bits)

Capteurs sol seuil

data0r

data1r

data6r

vect_capt (7 bits)

led0

led1

led7

ADC_CONVSTr

ADC_SCK *)

ADC_SDIr

ADC_SDO

SPI

Vref

Alims

VCC3P3_PWRON_n

LTC2308

IN0

IN1

IN7

Validation
- visuelle (Leds)
- analyse logique

*(**Max** 40 MHz)

ADC_SCK = $\overline{\overline{\text{clk}}}$