

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

01/06/2023

Interface VGA

Plan de Validation (Intermédiaire)

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

Atmani Hicham & Kamal Kherchouch
FORMATION SAFRAN CHEZ AJC

Date	Version	Remarque	Auteur
05/06/2023	A0	Création du Document de plan de validation pour la solution intermédiaire	AHI & KKH
12/06/2023	A1	Ajout de l'analyse hardware	AHI & KKH
13/06/2023	A1	Ajout du module Pattern_VGA Retrait des résultats de simulation pour les placer dans le rapport de validation version A0.	AHI & KKH
15/06/2023	A2	Ajout des objectifs de validation (analyse, test et démonstration)	AHI & KKH

Table des matières

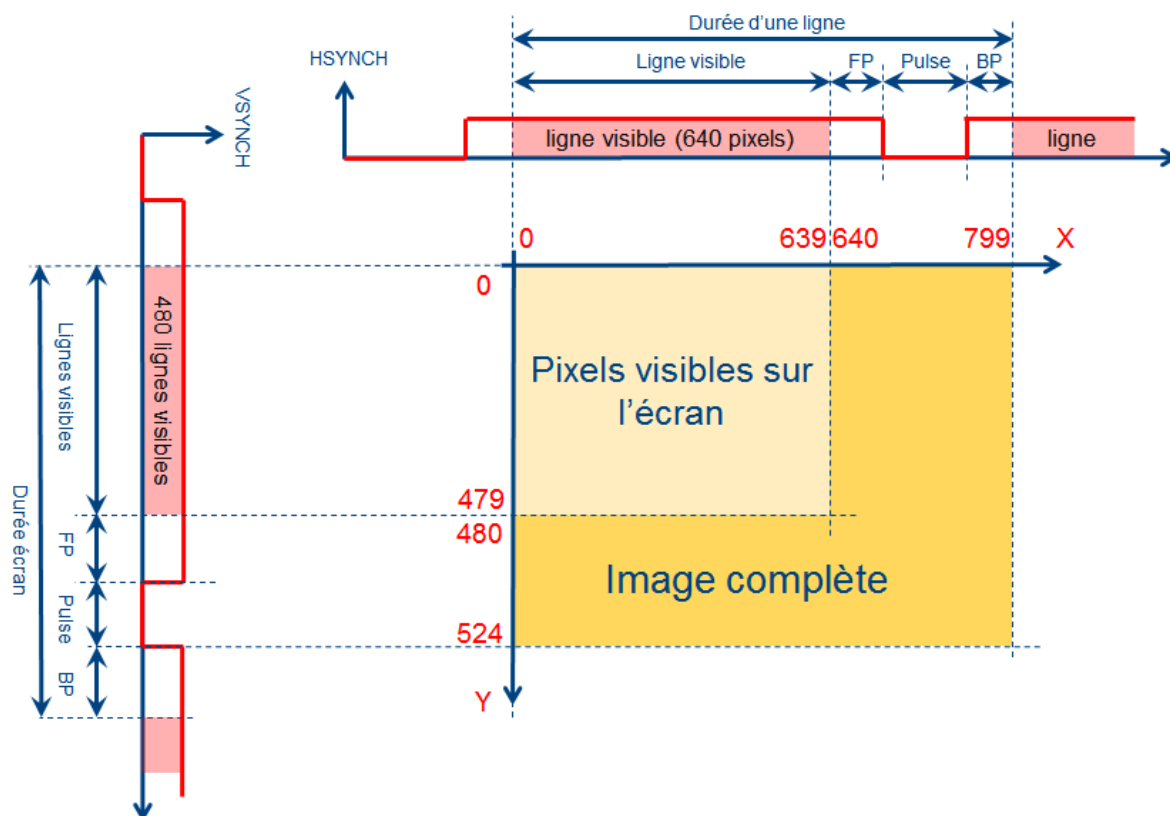
1.	Présentation de la norme VGA et de la solution intermédiaire	3
1.1.	Norme VGA.....	3
1.2.	Analyse Hardware	4
1.2.1.	Carte Cora Z7	4
1.2.2.	Carte Pmod	5
1.3.	Objectif de la solution intermédiaire	6
2.	Modules.....	7
2.1.	Module d'horloge : Phase-Locked Loop (PLL)	7
2.2.	Module de synchronisation (Synchro)	7
2.3.	Module de réalisation d'image (Pattern_VGA)	8
3.	Réalisation de la solution Intermédiaire	9
4.	Plan de validation de la solution intermédiaire	10
4.1.	Analyse de la version intermédiaire du projet	11
4.1.1.	Validation du module de synchronisation.....	11
4.1.2.	Validation de l'ajout du module PLL.....	11
4.1.3.	Validation de l'ajout du module Pattern_VGA.....	12
4.2.	Test de la version intermédiaire du projet.....	12
4.2.1.	Signal H_SYNC	12
4.2.2.	Signal V_SYNC	13
4.3.	Démonstration de la solution intermédiaire	13

1. Présentation de la norme VGA et de la solution intermédiaire

1.1. Norme VGA

La norme VGA 640 x 480 pixels (avec 16 couleurs) à 60 Hz impose une horloge de 25.175 MHz par pixel avec un rafraichissement vertical de 31.46875 KHz.

La norme VGA est représentée via l'image ci-dessous :



On y retrouve notre partie visible de l'image (640 x 480 pixels) plus des pixels supplémentaires qui nous permettrons d'avoir une marge lors de la synchronisation de l'image.

Les pixels sont composés de la manière suivante (selon la norme VGA):

Nom	Horizontal	Vertical
Image visible	640	480
Front Porch	16	10
Back Porch	48	33
Largeur de synchronisation	96	2
Nb total de pixel	800	525

On sait donc, de source sûre, que notre carte Cora Z7, nous fournit une horloge cadencée à 125 MHz, et qu'il va falloir prendre en compte cette horloge pour notre projet.

1.2.2. Carte Pmod

L'objectif de cette partie est de comprendre le fonctionnement de la carte Pmod.

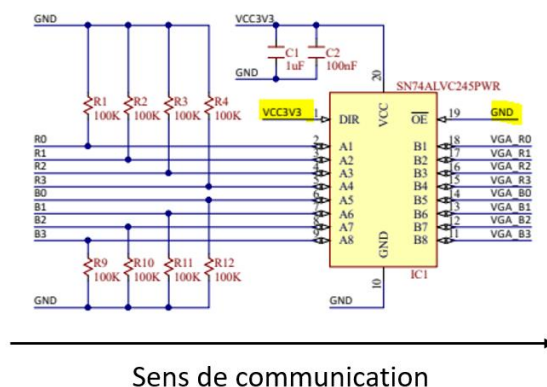
D'après la datasheet, on peut voir que la carte est composée de :

- Deux connecteurs Pmod
- Deux composants transceivers
- Une partie de conversion numérique / analogique
- Un connecteur VGA

Les deux connecteurs Pmod permettent de localiser les signaux qui vont communiquer avec la carte Cora z7. L'emplacement des signaux sur les connecteurs Pmod seront décrit plus tard dans le fichier. (cf. Plan de validation de la solution intermédiaire)

D'après la datasheet des transceivers SN74AVC4T245 on peut voir que les deux composants sont configuré pour avoir une communication allant du port A vers le port B, dans notre cas de la carte Cora Z7 vers notre écran où sera projeter notre image.

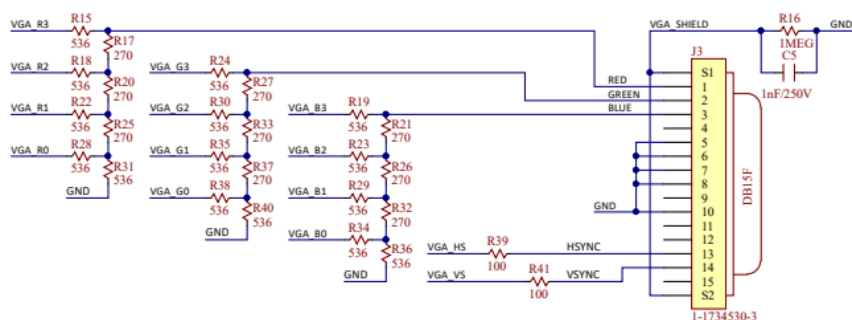
CONTROL INPUTS		OUTPUT CIRCUITS		OPERATION
OE	DIR	A PORT	B PORT	
L	L	Enabled	Hi-Z	B data to A bus
L	H	Hi-Z	Enabled	A data to B bus
H	X	Hi-Z	Hi-Z	Isolation



Pour finir, les signaux sortant des transceivers vont se retrouver dans un bloc de pont diviseur de tension afin de permettre une conversion des numérique en analogique (4 vers 1) puis sortie à l'aide du connecteur VGA.

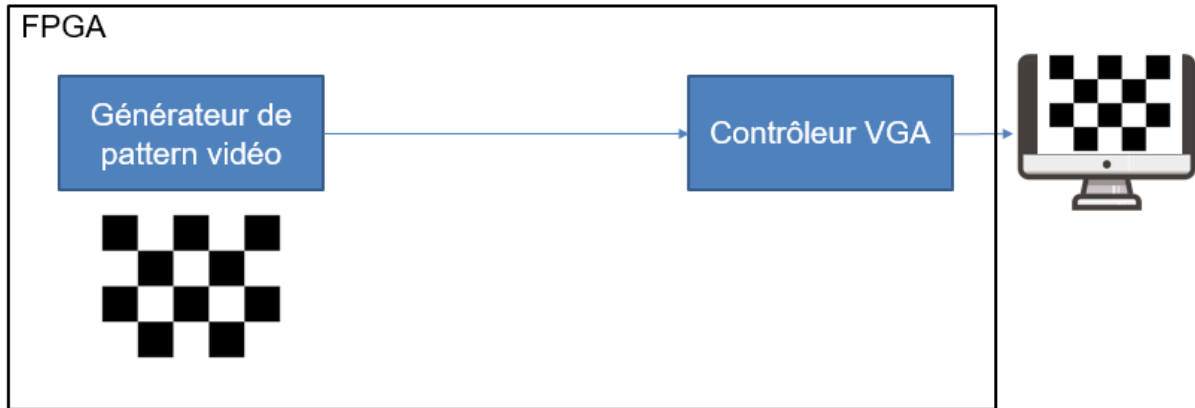
On passe donc de 4 signaux d'intensité par couleurs à un signal qui sera variable entre 0V et 3.3V. (0V étant pour la couleur Noir et 3.3V pour la couleur Blanc.)

Le niveau de tension du signal de couleur permettra de définir l'intensité de la couleur.



1.3. Objectif de la solution intermédiaire

L'objectif de la solution intermédiaire est de générer un « pattern » vidéo (type damier noir et blanc) et de projeter ce dernier sur un écran à l'aide d'un contrôleur VGA (Cf. image ci-dessous).



Remarque : Le pattern que nous allons réaliser est un damier de couleur noir et blanc de cinq colonnes pour 4 lignes. Ce damier aura l'avantage de :

- Nous permettre de valider l'utilisation de l'intégralité des couleurs (car pour avoir du blanc, il faut utiliser 100% de l'intensité des couleurs rouge, vert et bleu)
- Le damier nous permettra de valider la gestion des lignes et des colonnes car ça nous permettra de valider qu'un carré n'est dévier et que l'on a donc bien des lignes horizontales et verticales
- Le damier noir et blanc nous permettre de mieux visualiser la mise en place de notre filtre Gaussien avec les nuances entre les hautes et les basses fréquences

2. Modules

Description des modules que nous aurons besoin de réaliser et de valider unitairement avant de les intégrer à notre solution intermédiaire finale.

2.1. Module d'horloge : Phase-Locked Loop (PLL)

Nous devons réaliser une horloge cadencée à la fréquence de fonctionnement de la norme VGA soit 25.175 MHz à partir de notre horloge d'entrée, cadencée à 125 MHz. Pour ce faire nous allons créer un module PLL.



Pour cela nous aurons les signaux suivants :

En entrée

- Clk : signal d'horloge de 125 MHz.
- Reset : signal permettant de réinitialiser notre module.

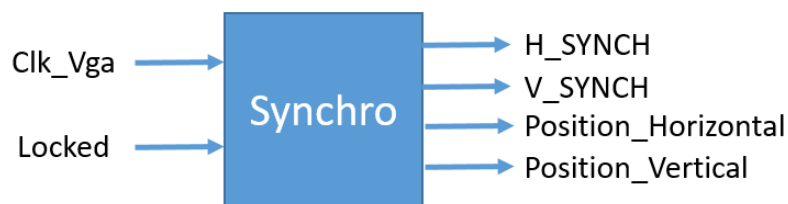
En Sortie

- Clk_Vga : signal d'horloge cadencé à 25.175 MHz.
- Locked : signal qui passe un '1' lorsque la clock de sortie est validée à la consigne demandée.

Remarque : pour la suite de notre projet nous utiliserons le signal Locked comme signal de reset des autres modules. Cela nous permettra de toujours travailler avec une horloge VGA stabilisée à 25.175MHz.

2.2. Module de synchronisation (Synchro)

Le module de synchronisation va nous permettre de réaliser le dimensionnement de notre afficheur, en fonction des informations données dans la partie Norme VGA.



Pour cela nous aurons les signaux suivants :

En entrée

- Clk_Vga : signal d'horloge de 25.175 MHz.
- Locked : signal permettant de réinitialiser notre module (Reset).

En Sortie

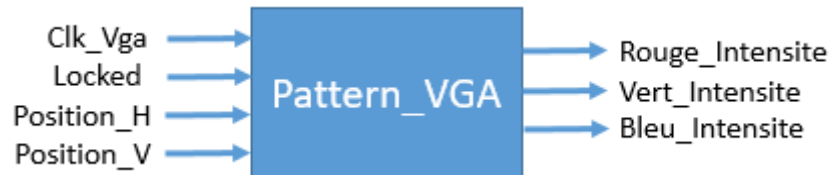
- H_SYNC : signal de synchronisation des lignes
- V_SYNC : signal de synchronisation des colonnes
- Position_Horizontal : compteur de pixel sur l'axe horizontal
- Position_Vertical : compteur de pixel sur l'axe vertical

2.3. Module de réalisation d'image (Pattern_VGA)

Le module pattern_VGA va nous permettre de :

- Réaliser notre damier sur la partie visible de notre image
- Forcée la couleur noire sur la partie invisible de notre image

Pour ce faire, nous allons créer le module suivant :



Pour cela nous aurons les signaux suivants :

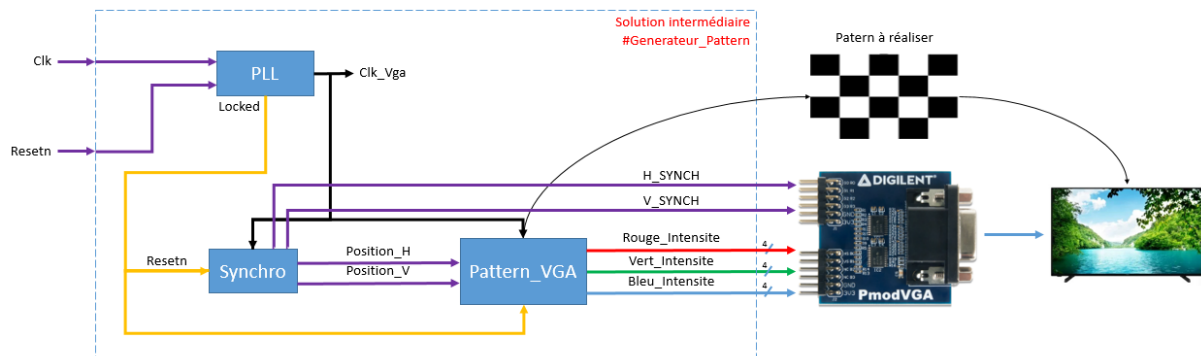
En entrée

- Clk_Vga : signal d'horloge de 25.175 MHz.
- Locked : signal permettant de réinitialiser notre module (Reset).
- Position_H : compteur de pixel sur l'axe horizontal
- Position_V : compteur de pixel sur l'axe vertical

En Sortie

- Rouge_Intensite : intensité de la couleur rouge sur 4 bits.
- Vert_Intensite : intensité de la couleur vert sur 4 bits.
- Bleu_Intensite : intensité de la couleur bleu sur 4 bits.

3. Réalisation de la solution Intermédiaire



La solution intermédiaire devra donner le résultat suivant :

Le module PLL nous délivre notre signal d'horloge permettant de faire fonctionner le VGA et un signal permettant de gérer nos reset système.

Le module synchro lui va nous fournir les signaux (H_Synch et V_Synch) qui iront directement sur la carte Pmod. Et les deux compteurs de pixel (horizontal et vertical).

La partie pattern_VGA va nous permettre de générer notre pattern vidéo souhaité. Ici, nous voulons créer un damier noir et blanc avec 5 colonnes et 4 lignes.

Cela implique que pour le format VGA (640 x 480 pixels) chaque rectangle du damier devra mesurer 128 x 120 pixels.

4. Plan de validation de la solution intermédiaire

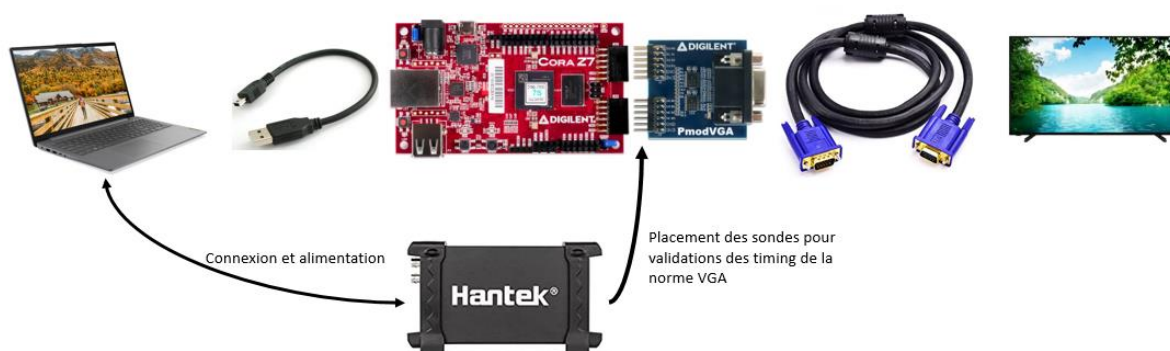
L'objectif de ce plan de validation est de proposer une solution afin de valider le bon fonctionnement de notre proposition de solution.

Si on reprend le tableau de décomposition des pixels de la norme VGA présenté dans la partie Norme VGA.

Sachant que la résolution d'un pixel est de 25.175 MHz soit 39,72 ns et que la fréquence de rafraichissement verticale est de 31.46875 KHz soit de 31,78 μ s (ce qui correspond à 480 lignes x 800 pixels), on obtient les timings suivant :

Nom	Horizontal (Pixel)	Horizontal (Timing μ s)	Vertical (Pixel)	Vertical (Timing ms)
Image visible	640	25.422	480	15.253
Front Porch	16	0.636	10	0.318
Back Porch	48	1.907	33	1.049
Largeur de synchronisation	96	3.813	2	0.064
Nb total de pixel	800	31.778	525	16.683

Pour valider ces différents timings, nous utiliserons un oscilloscope Hantek et des sondes que nous placerons sur les broches à valider :



Le pinout de la carte Pmod VGA est le suivant :

Header J1						Header J2					
Pin	Signal	Description	Pin	Signal	Description	Pin	Signal	Description	Pin	Signal	Description
1	R0	Red 0	7	B0	Blue 0	1	G0	Green 0	7	HS	Horizontal Sync
2	R1	Red 1	8	B1	Blue 1	2	G1	Green 1	8	VS	Vertical Sync
3	R2	Red 2	9	B2	Blue 2	3	G2	Green 2	9	NC	Not Connected
4	R3	Red 3	10	B3	Blue 3	4	G3	Green 3	10	NC	Not Connected
5	GND	Power Supply Ground	11	GND	Power Supply Ground	5	GND	Power Supply Ground	11	GND	Power Supply Ground
6	VCC3V3	Positive Power Supply	12	VCC3V3	Positive Power Supply	6	VCC3V3	Positive Power Supply	12	VCC3V3	Positive Power Supply

4.1. Analyse de la version intermédiaire du projet

Cette partie va permettre de faire l'analyse de la partie intermédiaire de notre projet. Pour ce faire, nous utiliserons la partie simulation du logiciel Vivado.

4.1.1. Validation du module de synchronisation

Cette partie va permettre de réaliser la validation en simulation du module synchronisation présenté dans la partie Module de synchronisation (Synchro).

Pour ce faire, nous devons commencer par réaliser le code VHDL du module, puis un test Bench qui permettra de faire fonctionner ce dernier.

Pour ce test bench, et comme la PLL n'est toujours pas mis en place, nous créerons directement une horloge cadencé à 25.175MHz (fréquence de fonctionnement du VGA).

La validation du module suivra les étapes suivantes :

- Validation du reset.
 - o Si le reset est à 1, le module ne doit pas fonctionner
 - o Si le reset est à 0, le module est en mode de fonctionnement
- Validation de la création de l'horloge du test Bench.
 - o L'horloge doit être cadencée à 25.175MHz.
- Validation du signal de synchronisation horizontal (H_SYNC).
 - o L'impulsion doit être de 3.8μs
 - o La période doit être de 31.778μs
- Validation du signal de synchronisation vertical (V_SYNC).
 - o L'impulsion doit être de 64ms
 - o La période doit être de 16.68ms

4.1.2. Validation de l'ajout du module PLL

Après avoir ajouté le module PLL à notre projet, nous modifieront notre test Bench pour avoir une horloge d'entrée cadencée à 125MHz (horloge qui sera plus tard fournis par la carte Cora Z7).

La validation suivra les étapes suivantes :

- Validation de la création de l'horloge du test Bench.
 - o L'horloge doit être cadencée à 125MHz
- Validation de l'horloge de sortie de la PLL.
 - o Le signal Locked doit être à 1 pour indiquer que l'horloge de sortie soit stabilisée
 - o L'horloge doit être cadencée à 25.175MHz

4.1.3. Validation de l'ajout du module Pattern_VGA

Une fois le module ajouté à notre projet, nous n'aurons pas besoin de modifier le test Bench pour réaliser la simulation.

La validation suivra les étapes suivantes :

- Valider que les sorties d'intensité de couleur sont à 0 lorsque l'on est dans les zones front porch, back porch et largeur de synchronisation.
- Visualiser la représentation d'une ligne avec 2 parties blanches
- Visualiser la représentation d'une ligne avec 3 parties blanches

4.2. Test de la version intermédiaire du projet

Le test de la version intermédiaire du projet passe par :

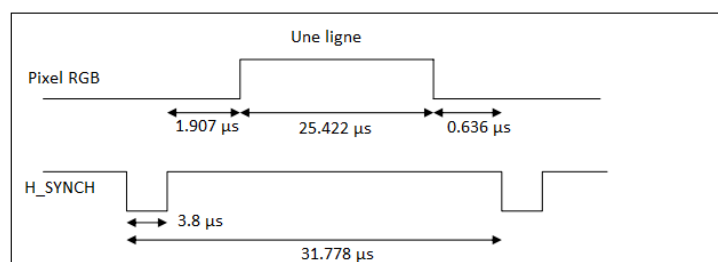
- La réalisation de la synthèse du projet
 - o Vérifier que le schéma obtenu est bien en accord avec la solution proposée
- La réalisation de l'implémentation du projet
 - o Vérifier qu'il n'y a pas de déviation de timing
- La génération du Bitstream et l'envoi du projet sur la carte Cora.
 - o Validation à l'oscilloscope de l'impulsion et de la période de H_SYNC
 - o Validation à l'oscilloscope de l'impulsion et de la période de V_SYNC
 - o Validation de la représentation d'une ligne avec 2 parties blanches
 - o Validation de la représentation d'une ligne avec 3 parties blanches

Pour valider les signaux à l'aide de l'oscilloscope, le bitstream doit être envoyé à la carte cora. Et les sondes placées aux différents endroits du connecteur Pmod de la carte Cora, comme expliqué dans la partie : Plan de validation de la solution intermédiaire.

4.2.1. Signal H_SYNC

Pour valider le signal H_SYNC, on va connecter une sonde d'oscilloscope à la broche 7 du connecteur J2 et connecter la masse de cette sonde à la masse de notre système. On connecte également une seconde sonde à l'une de nos broche de pixel afin de faire valider la taille de l'image (exemple broche 1 du connecteur J2 pour visualiser la couleur vert0).

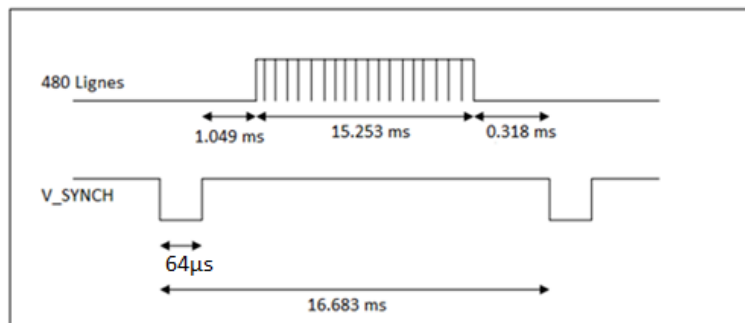
Pour réaliser la capture, on synchronise notre oscilloscope pour détecter un front descendant de notre signal H_SYNC. Et on valide les timings suivant :



4.2.2. Signal V_SYNC

Pour valider le signal V_SYNC, on va connecter une sonde d'oscilloscope à la broche 8 du connecteur J2 et connecter la masse de cette sonde à la masse de notre système. On connecte également une seconde sonde à l'une de nos broche de pixel afin de faire valider la taille de l'image (exemple broche 1 du connecteur J2 pour visualiser la couleur vert0).

Pour réaliser la capture, on synchronise notre oscilloscope pour détecter un front descendant de notre signal H_SYNC. Et on valide les timings suivant :



4.3. Démonstration de la solution intermédiaire

La démonstration de la solution intermédiaire va nous permettre de voir que l'on arrive bien à projeter sur un écran notre pattern. Pour ce faire nous allons réaliser le montage présenté dans la partie Plan de validation de la solution intermédiaire.

Il y a deux cas possibles :

- Lorsque le bouton reset est activé, le damier ne s'affiche pas
- Lorsque le bouton reset n'est pas activé, le damier est affiché.