

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 sweep 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

## Table des matières

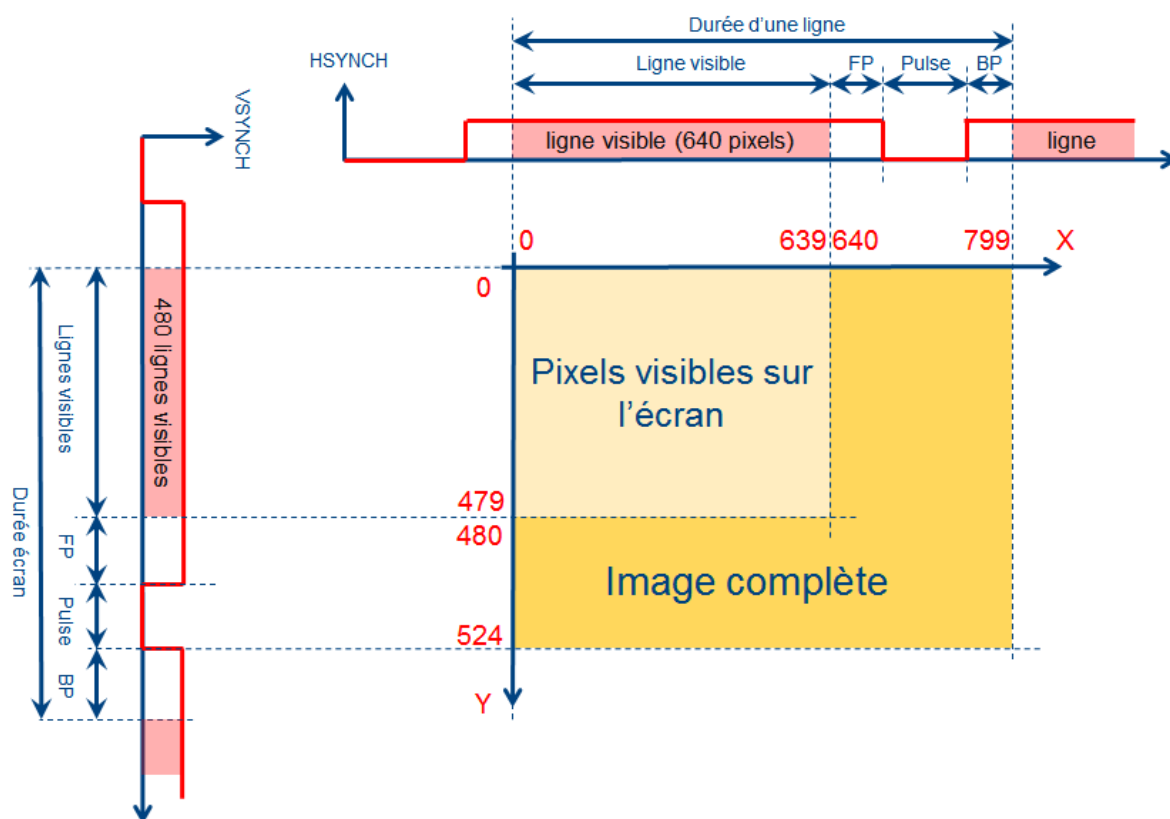
1.	Présentation de la norme VGA et de la solution intermédiaire .....	3
1.1.	Norme VGA.....	3
1.2.	Objectif de la solution intermédiaire .....	4
2.	Modules.....	5
2.1.	Module d'horloge : Phase-Locked Loop (PLL) .....	5
2.2.	Module de synchronisation (Synchro) .....	5
3.	Réalisation de la solution Intermédiaire .....	6
4.	Plan de validation de la solution intermédiaire .....	7
4.1.	Validation des signaux horizontaux.....	8
4.2.	Validation des signaux verticaux .....	8

## 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 :



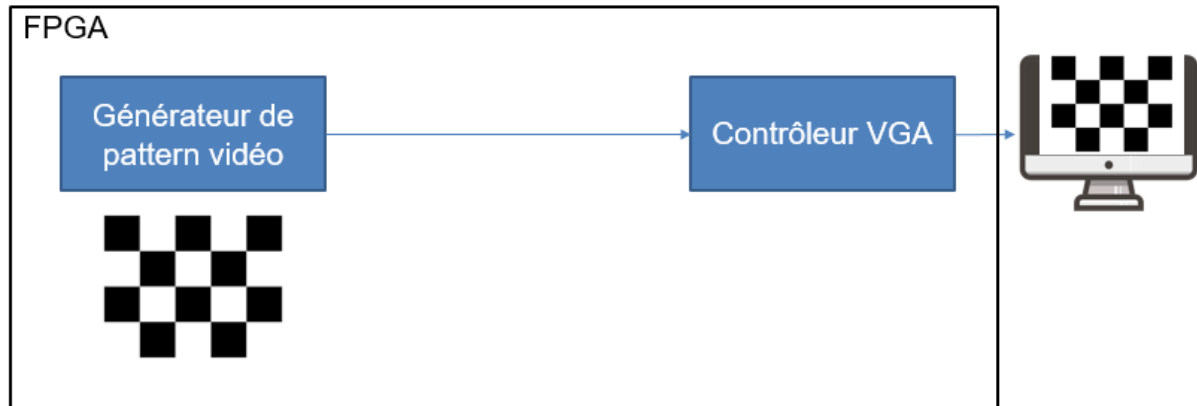
On y retrouve notre partie visible de l'image (640 x 480 pixels) plus des pixels supplémentaires qui nous permettront d'avoir une marge lors de la synchronisation de l'image.

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
<b>Nb total de pixel</b>	<b>800</b>	<b>525</b>

## 1.2. 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).



## 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 100 MHz. Pour ce faire nous allons créer un module PLL.



Pour cela nous aurons les signaux suivants :

En entrée

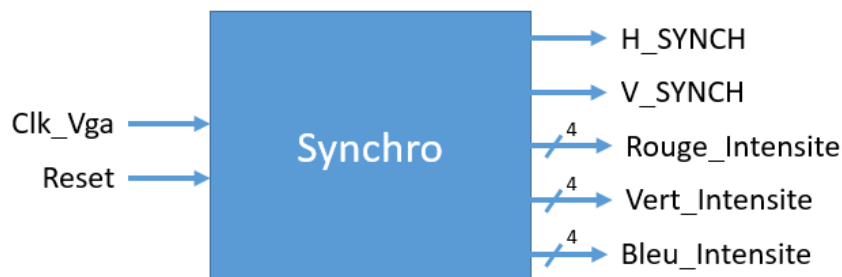
- Clk : signal d'horloge de 100 MHz.
- Reset : signal permettant de réinitialiser notre module.

En Sortie

- Clk\_Vga : signal d'horloge cadencé à 25.175 MHz.

### 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

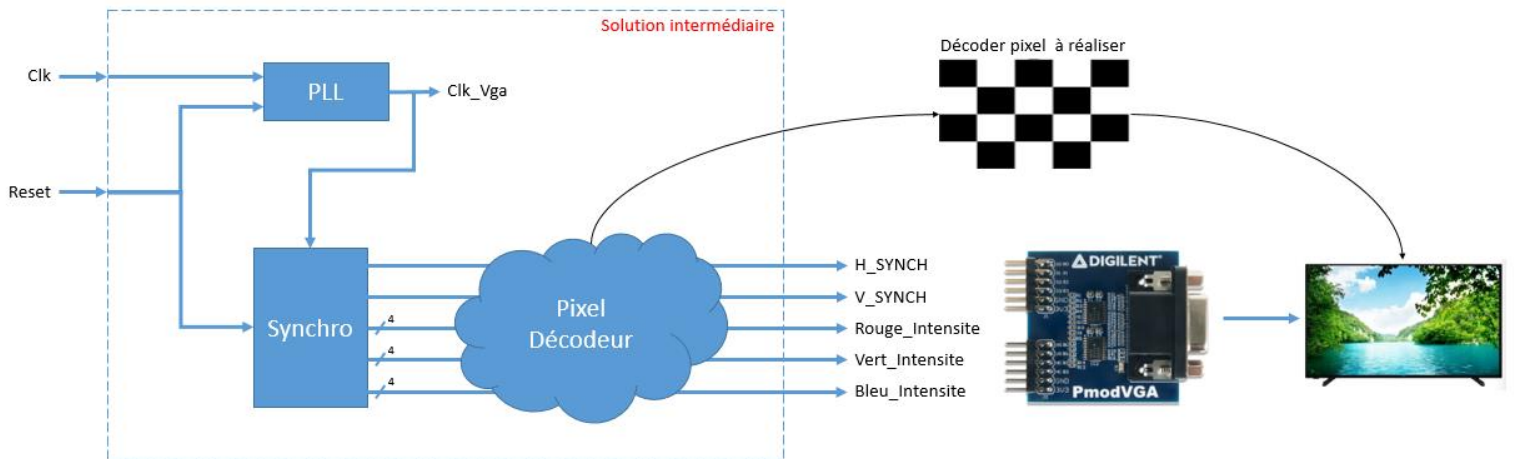
- Clk\_Vga : signal d'horloge de 100 MHz.
- Reset : signal permettant de réinitialiser notre module.

En Sortie

- H\_SYNC : signal de synchronisation des lignes
- V\_SYNC : signal de synchronisation des colonnes
- Rouge\_Intensite : intensité de la couleur rouge (variant de 0 à 15)
- Vert\_Intensite : intensité de la couleur vert (variant de 0 à 15)
- Bleu\_Intensite : intensité de la couleur bleu (variant de 0 à 15)

### 3. Réalisation de la solution Intermédiaire

La solution intermédiaire devra donner le résultat suivant :



La partie pixel décodeur 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

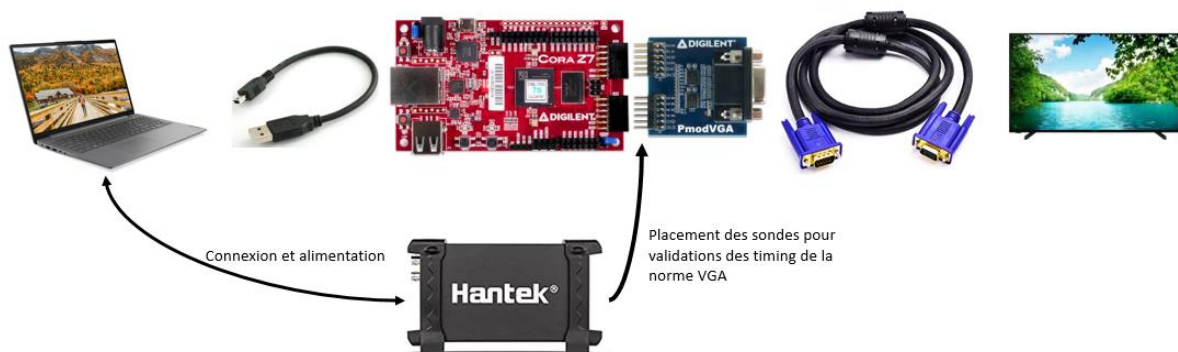
L'objectif de ce plan de validation est de proposer une solution afin de valider le bon fonctionnement de notre proposition de solution.

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  $\mu$ s, on obtient les timings suivant :

Nom	Horizontal (Pixel)	Horizontal (Timing $\mu$ 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
<b>Nb total de pixel</b>	<b>800</b>	<b>31.778</b>	<b>525</b>	<b>16.683</b>

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 :

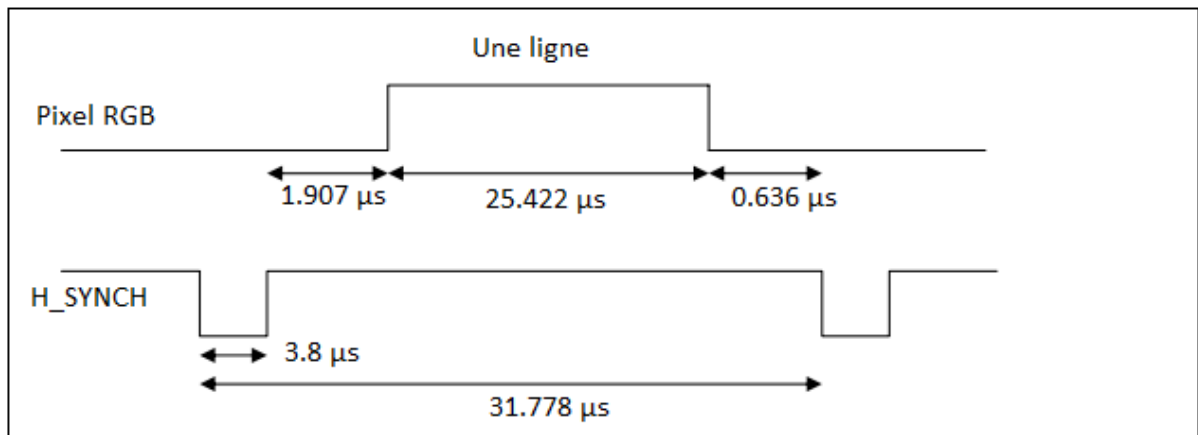
Header J1						Header J2					
Pin	Signal	Description	Pin	Signal	Description	Pin	Signal	Description	Pin	Signal	Description
1	R0	Red 0	7	B0	Blue 0	1	G0	Green 0	7	HS	Horizontal Sync
2	R1	Red 1	8	B1	Blue 1	2	G1	Green 1	8	VS	Vertical Sync
3	R2	Red 2	9	B2	Blue 2	3	G2	Green 2	9	NC	Not Connected
4	R3	Red 3	10	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. Validation des signaux horizontaux

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. Validation des signaux verticaux

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 :

