

**Université Paul Sabatier
Toulouse**

Compte Rendu

TP caméra CCD Programmation

M1 EEA

- ✧ Module : Capteurs optiques et formation des images
- ✧ Réalisé par :
 - KABOU Abdeldjalil
 - GRENI Wail
 - BENAMAROUCHE Nail

I. OBJECTIF

L'objectif de ce TP est de comprendre le rôle des chronogrammes dans le contrôle d'un capteur CCD, d'analyser leur influence sur les modes de lecture des images et d'expérimenter la nécessité de la synchronisation entre l'ordinateur et la caméra pour un fonctionnement optimal.

II. Analyse du listing tp.c

1. *Résumé simplifié de la fonction cmdCamAcqNormal()* :

Cette fonction réalise une acquisition d'image standard avec un capteur CCD en suivant plusieurs étapes clés :

1. Vidage du capteur CCD

Avant l'exposition, le capteur est nettoyé en supprimant les charges résiduelles à l'aide de la fonction fast_vidage(), répétée plusieurs fois.

2. Exposition à la lumière

La caméra attend pendant un temps d'intégration (exptime), permettant aux pixels de collecter la lumière et de générer des électrons proportionnels à l'intensité lumineuse.

3. Lecture de l'image

- * Les électrons accumulés sont transférés ligne par ligne vers un registre horizontal.
- * Ils sont ensuite convertis en tension par un Convertisseur Analogique-Numérique (CAN) pour produire une image numérique en niveaux de gris (16 bits).

4. Traitement du signal

- * Une correction est appliquée avec un masque (0x8888) pour ajuster les artefacts dus au port parallèle.
- * Les données sont ensuite stockées en mémoire sous forme d'un tableau de pixels.

5. Transfert vers l'ordinateur

Les données de l'image sont transférées du port parallèle vers la mémoire RAM, puis converties dans un format compatible avec l'interface graphique Tcl/Tk.

6. Finalisation et stockage

Les métadonnées de l'image (taille, format FITS) sont enregistrées pour une analyse ultérieure, sans impact direct sur la caméra.

En résumé, cette fonction assure une prise d'image complète, du nettoyage du capteur jusqu'à l'affichage et l'enregistrement des données.

2. Correspondance des variables avec les différentes parties du capteur CCD Kodak AF400 :

nb_deadbeginphotox (14) et nb_deadendphotox (14)

- ✖ Correspondance : Pixels masqués horizontalement, utilisés pour la calibration et l'élimination du bruit.
- ✖ Sur le schéma : Zones de bordure au début et à la fin des lignes actives.

nb_deadbeginphotoy (4)

- ✖ Correspondance : Lignes masquées en haut de la matrice pour éliminer les artefacts de transfert de charges.
- ✖ Sur le schéma : Les 4 "Dark lines" en haut de la matrice.

binx et biny

- ✖ Correspondance : Paramètres de binning regroupant plusieurs pixels pour augmenter la sensibilité au détriment de la résolution.
- ✖ Sur le schéma : Non directement représentés, mais influencent la lecture des pixels.

exptime

- ✖ Correspondance : Temps d'exposition déterminant la durée de capture de la lumière.
- ✖ Sur le schéma : Non représenté directement, mais affecte la quantité de lumière captée par la zone active.

bufno

- ✖ Correspondance : Numéro du buffer mémoire stockant l'image avant traitement.
- ✖ Sur le schéma : Non visible, géré au niveau logiciel.

3.

Le capteur présenté est un CCD à transfert de trame (Frame Transfer CCD), identifiable par sa structure où une partie active de l'image est accompagnée de zones de stockage et de lecture. Concernant le nombre d'électrodes par photosite, on peut observer la présence de deux électrodes horizontales ($\Phi H1, \Phi H2$) et deux électrodes verticales ($\Phi V1, \Phi V2$), ce qui suggère que chaque photosite est contrôlé par deux électrodes verticales et deux horizontales pour le transfert des charges.

4.

Le capteur KAF-0401E transfère les charges en deux étapes : d'abord verticalement (ligne par ligne) vers la zone de stockage, puis horizontalement (colonne par colonne) vers la sortie. Ce transfert est contrôlé par quatre horloges : deux verticales ($\Phi V1, \Phi V2$) et deux horizontales ($\Phi H1, \Phi H2$).

5.

La fonction *fast_vidage(cam)* :

Vidage rapide de la matrice

But : Cette fonction permet de vider rapidement le capteur CCD en déplaçant les charges verticalement et en lisant le registre horizontal.

Principe de fonctionnement :

1. Paramètres définis :

- ✗ *decaligne* = 32 : Ce paramètre définit la quantité de lignes qui seront vidées par cycle, ici 32 lignes à la fois.
- ✗ *imax* et *jmax* : Ces variables sont calculées pour déterminer le nombre de pixels et de lignes à traiter.

2. Boucle principale :

- ✗ Vidage vertical (*zi_zh(cam)*) : Cette fonction effectue 32 décalages verticaux pour évacuer les lignes de pixels vers la zone morte du CCD.
- ✗ Lecture rapide du registre horizontal (*read_pel_fast2(cam)*) : Cette fonction lit toutes les colonnes du registre horizontal sans effectuer de réinitialisation de la charge, permettant une lecture rapide des pixels.

Horloges utilisées :

- ✗ *V1, V2* : Ces horloges sont déclenchées dans *zi_zh(cam)* pour permettre le décalage des lignes verticalement.
- ✗ *H1, H2* : Ces horloges sont activées dans *read_pel_fast2(cam)* pour effectuer la lecture horizontale du registre.

La fonction *read_win(cam,p)* :

Lecture d'une fenêtre du CCD

But : Lire une portion spécifique du capteur CCD, avec binning et suppression des pixels morts, puis numériser l'image.

Principe de fonctionnement :

1. Calcul des dimensions :

- ✗ *imax* = *cam->nb_photox* / *cam->binx* : Détermine le nombre de colonnes utiles après binning.

- ✖ **jmax = cam->nb_photoy / cam->biny** : Détermine le nombre de lignes utiles après binning.
- ✖ **cx1, cx2, cy1** : Ces paramètres définissent le nombre de pixels morts à ignorer au début et à la fin de chaque ligne ou colonne.

2. Étapes principales :

Suppression des premières lignes mortes (cy1) :

- ✖ Un décalage vertical est effectué avec `zi_zh(cam)` pour déplacer les lignes mortes hors du champ de vision utile.
- ✖ Les lignes sont vidées avec `fast_line(cam)` pour nettoyer les pixels inutiles.

Lecture ligne par ligne :

- ✖ Après chaque ligne, un décalage vertical est effectué pour préparer la ligne suivante avec `zi_zh(cam)`.
- ✖ Les premiers pixels morts (définis par `cx1`) sont supprimés avec `read_pel_fast(cam)`.
- ✖ Lecture pixel par pixel : Chaque pixel est lu en effectuant plusieurs étapes :
 - Réinitialisation du signal avec l'horloge Reset (bit 4).
 - Stabilisation du signal avec l'horloge Clamp (bit 5).
 - Conversion analogique-numérique avec l'activation de l'horloge Start Convert (bit 6).
 - Lecture des 4 nibbles successifs pour reconstituer un pixel de 16 bits.
- ✖ Suppression des pixels morts en fin de ligne (`cx2`) avec un dernier appel à `read_pel_fast(cam)`.

Horloges utilisées :

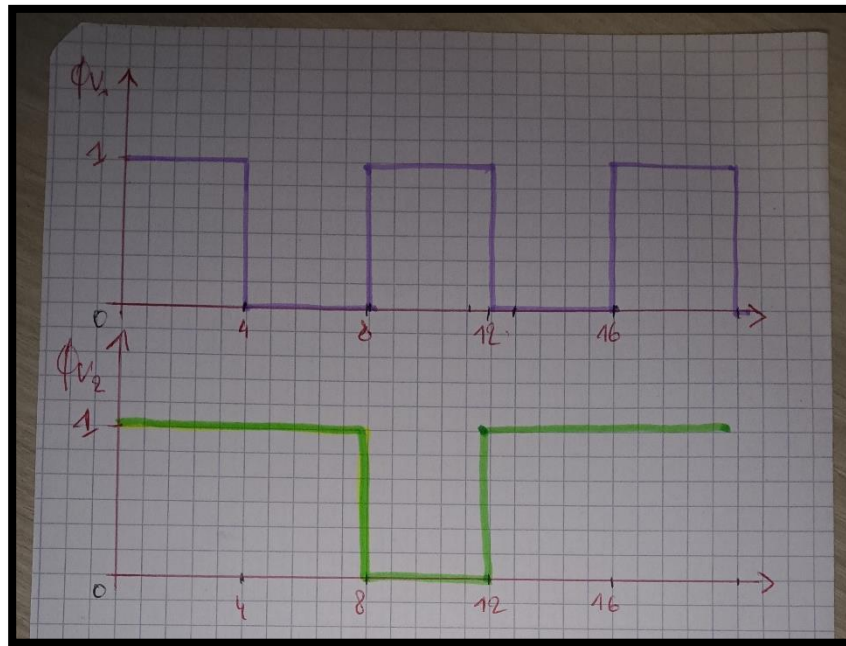
- ✖ V1, V2 : Pour le décalage vertical des lignes (fonction `zi_zh(cam)`).
- ✖ H1, H2 : Pour le transfert des pixels vers le registre de sortie.
- ✖ R (Reset) : Active la réinitialisation du signal avant la lecture de chaque pixel.
- ✖ CL (Clamp) : Stabilise le signal avant la numérisation.
- ✖ Start Convert (bit 6) : Déclenche la conversion analogique-numérique pour chaque pixel.

6.

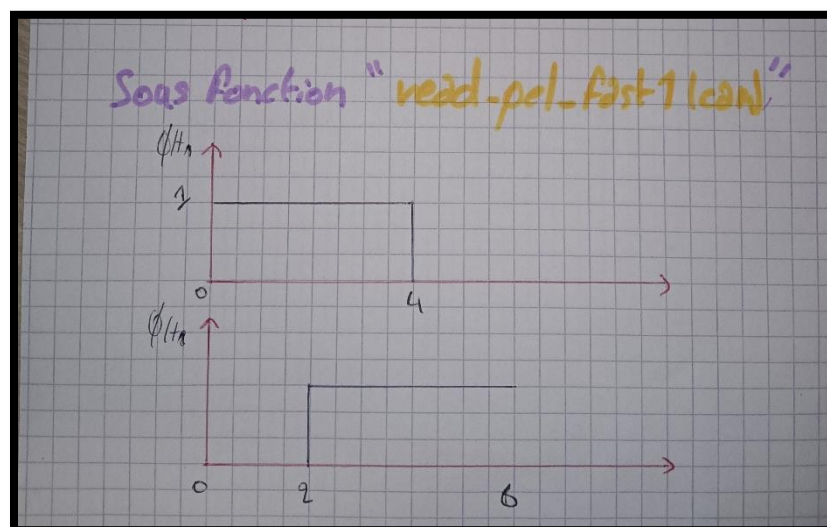
Chronogrammes des sous-fonctions de fast_vidage(cam)

Voici une description générale des chronogrammes des sous-fonctions dans la fonction fast_vidage. Les chronogrammes indiquent l'activation des horloges et les durées des différents paliers.

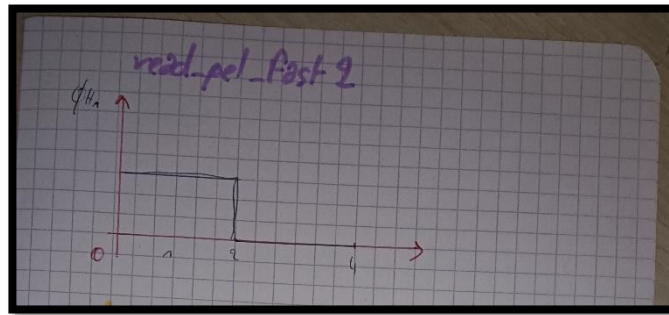
1. Sous-fonction zi_zh(cam) :



2. Sous-fonction read_pel_fast1(cam) :



3. Sous-fonction read_pel_fast2(cam) :



Pourquoi certaines commandes libcam_out() sont-elles répétées plusieurs fois ?

Il existe plusieurs raisons pour lesquelles certaines commandes libcam_out() sont répétées dans le cadre du fonctionnement du capteur CCD :

1. Respect des durées minimales de pulse (Datasheet §3.4) :

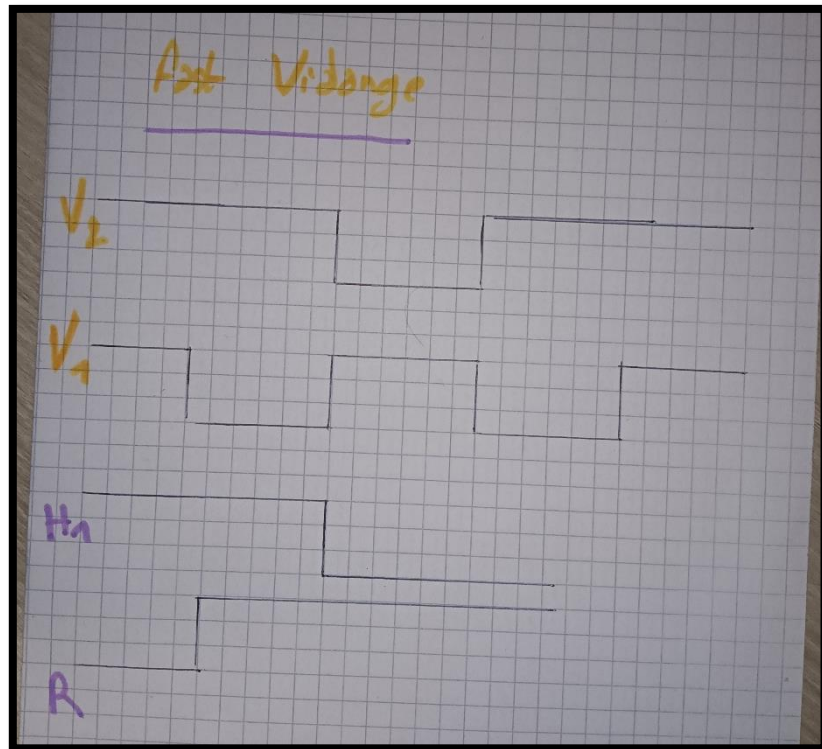
- ✗ Le capteur CCD nécessite des impulsions d'horloge avec des durées minimales pour assurer un transfert fiable des charges. Cela garantit que les signaux sont correctement synchronisés avec le matériel du capteur.

2. Stabilisation des signaux analogiques :

- ✗ Certaines commandes, telles que les contrôles **Clamp** et **Reset**, servent à stabiliser les signaux analogiques avant la lecture des pixels. Ces signaux doivent être maintenus pendant un certain temps pour garantir la fiabilité du signal avant la conversion numérique.

3. Synchronisation des phases multiples (CCD 3 phases) :

- ✗ Le capteur CCD à trois phases (V1, V2, V3) nécessite une séquence précise pour transférer les charges. Chaque phase doit être synchronisée correctement avec les autres pour assurer un transfert de charges sans erreurs. Par conséquent, certaines commandes sont répétées plusieurs fois pour coordonner le mouvement des charges à travers les différentes phases du capteur.



8)

Le binning est une technique utilisée avec les capteurs CCD/CMOS pour regrouper plusieurs pixels adjacents en un seul "super-pixel".

Utilité :

Augmente la sensibilité : Améliore le rapport signal/bruit en captant plus de lumière.

Accélère l'acquisition : Diminue le temps de lecture des pixels, utile en imagerie rapide

Conclusion

Ce TP a permis d'explorer en détail le fonctionnement d'un capteur CCD et les principes de son pilotage par un programme informatique. L'étude des chronogrammes a mis en évidence l'importance de la gestion des signaux d'horloge pour assurer un transfert efficace des charges et une lecture correcte des données

L'analyse des fonctions du fichier tp.c a également aidé à comprendre comment les commandes sont envoyées à la caméra pour acquérir des images dans différents modes. Ce travail offre ainsi une vision concrète du fonctionnement d'un capteur CCD, tout en soulignant l'importance de la synchronisation et du traitement des signaux dans les systèmes d'imagerie numérique.

