Transmission d'images sur canal audio à faible bande passante : Application aux technologies spatiales

Capitaine T., Bourny V., Barrandon L., Senlis J.
Etudiants: Champagne D., Dromas C., Dautreppe J
thierry.capitaine@u-picardie.fr, valery.bourny@u-picardie.fr, ludovic.barrandon@u-picardie.fr, jacky.senlis@u-picardie.fr,
Université de Picardie Jules Verne
INSSET 48, rue Raspail, 02100 Saint-Quentin

RESUME: L'institut INSSET de Saint-Quentin est impliqué depuis cette année dans un projet visant à concevoir l'intégralité des modules d'un pico satellite (CubeSat) avec l'objectif de le lancer en orbite d'ici quelques années. Un de ces modules aura en charge la transmission d'image vers la terre en s'appuyant sur un protocole normalisé (SSTV). Cet article décrit la mise en place organisationnelle de ce projet étudiant ainsi que les stratégies de développement matériel et logiciel retenues afin de tenir compte des différentes contraintes technologiques spécifiques au contexte de cette application.

Mots clés: protocole de communication, programmation optimale, systèmes embarqués, technologies spatiales, radioamateur.

1 INTRODUCTION

L'INSSET de Saint-Quentin [1] propose depuis 2003 un Master dans le domaine des systèmes embarqués qui repose sur une formation par projets (30% des crédits ECTS) dans lesquels des groupes d'étudiants sont tenus d'élaborer des solutions qui répondent aux besoins de laboratoires de recherche ou d'industriels. Ces dernières années, le développement de plates-formes plus complexes sur plusieurs années, a été favorisé. À ce jour, nous travaillons sur trois plateformes différentes:

- PRO.MO.CO (plate-forme Robotique Mobile Communicante), composé d'un ensemble de robots mobiles dont l'architecture repose sur un échange de données de modules autonomes (capteurs et actionneurs intelligents) communiquant sur un bus CAN [2].
- La station au sol de réception et de décodage des données de télémétrie de satellites GENSO (Global Education for Network Satellites Operations), contrôlable à distance depuis Internet [3].
- Les projets CubeSAT (pour lequel cet article fait l'objet) basé sur le développement de l'ensemble des modules constituant un picosatellite (1W, 1dm^3, 1kg). Une des charges utiles aura pour objectif de transmettre des d'images à différentes stations au sol [4].

2 PROBLEMATIQUE

2.1 Envoyer des images depuis l'espace

Les images, prises par une caméra embarquée dans le satellite, sont encodées par un microcontrôleur qui pilote un émetteur radio afin de les envoyer vers les stations au sol. Les bandes radioamateurs ont étés retenues pour les transmissions depuis le satellite, permettant à un maximum de personnes d'acquérir ces images. Le mode de transmission retenu pour notre application est le protocole SSTV car il est le plus utilisé par la communauté radio amateurs pour la transmission d'images sur un canal audio. En effet le décodage et l'affichage des images reçues ne nécessitent qu'une chaine de réception simple comme le montre la figure n : °1. Une antenne, un récepteur radio sur la fréquence de descente du satellite, une liaison entre la sortie BF et l'entrée de la carte son d'un ordinateur ainsi qu'un logiciel de décodage (freeware).

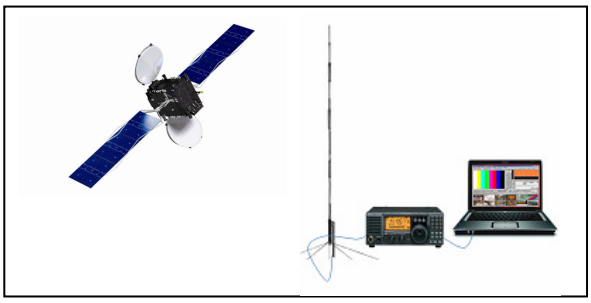


fig 1 : Chaîne de transmission d'images SSTV

3 SSTV (TÉLÉVISION À BALAYAGE LENT)

3.1 Histoire

Le concept de la SSTV fut introduit par Copthorne Macdonald en 1957-1958. Il a développé le premier système SSTV en utilisant un moniteur électrostatique et un tube vidicon. Une des premières utilisations de la SSTV fut la transmission de photos de la face cachée de la lune pour la mission Luna 3 de 1959 (Voir figure 2). Ce protocole est maintenant extrêmement répandu au sein de la communauté internationale radio amateur pour l'échange

d'images et de photos à travers le monde sur différentes bandes.de fréquences via le biais d'un simple canal audio.

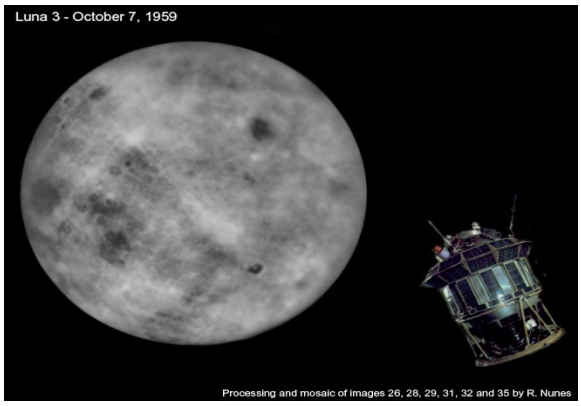


fig 2 : Transmission SSTV de l'image de la Lune par la sonde Luna 3 en 1959

3.2 Protocole SSTV (Slow Scan Television)

Ce protocole permet la transmission de photos sur un canal radio à faible bande passante (environ 3 kHz). Le signal SSTV impose un protocole très rigoureux pour les temps de génération des fréquences codifiant les pixels. En effet, l'image est transmise ligne par ligne, chaque pixel est constitué de 3 composantes couleurs dont le niveau (intensité) est codé sous forme d'une fréquence spécifique (modulation AFSK).

3.2.1 Constitution d'une trame

Une trame est composée :

- d'un signal de synchronisation (modulation FSK)
- d'un entête numérique qui permet au logiciel de décodage d'identifier le protocole de codage utilisé lors de la transmission (modulation FSK).
- de la transmission successive des lignes composant l'image, chaque ligne étant décomposée en ces composantes RVB.
- chaque fin de transmission est identifiée par des signaux de synchronisation distincts.

De nombreux modes existent [5] et leurs différences concernent la résolution globale de l'image, la codification des pixels (nombre de niveau de codage des couleurs). Le choix d'un mode conditionne le temps de trans fert complet d'une image (environ 2 minutes pour le protocole retenu Scottie 1.

3.2.2 Entête SSTV

Les modes SSTV standards utilisent un code numérique unique, appelé « VIS » (Vertical Interval Signal), qui permet d'identifier le mode lors de la réception. Bien que l'en-tête de calibration complète soit souvent désigné comme le «VIS code », le code en lui-même n'est qu'une partie de celui-ci. Un code de sept bits est transmis, le bit le moins significatif (LSB) est transmis en premier le VIS intègre un bit de parité comme le détaille la figure 3.

TIME(ms)	FREQUENCY(hz)	IDENTITY
300	1900	Leader tone
10	1200	break
300	1900	Leader tone
30	**1200**	**VIS start bit**
30	**bit 0**	**1100hz = "1", 1300hz = "0"**
30	**bit 1**	**""**
30	**bit 2**	**""**
30	**bit 3**	**""**
30	**bit 4**	**""**
30	**bit 5**	**""**
30	**bit 6**	**""**
30	**PARITY**	**Even=1300hz, Odd=1100hz**
30	**1200**	**VIS stop bit**

fig 3 : Entête de calibration avec le code VIS

3.3 Mode Scottie 1

Nous avons choisi ce mode (protocole Américain) parce qu'il est le plus utilisé dans le monde (80 % des images sont envoyées dans ce mode). La première ligne transmise commence par une pulsation de synchronisation « Start » de 9.0mS à 1200Hz. Des impulsions régulières de synchronisation sont ensuite positionnées entre les scans bleu et rouge (voir Figure 4 et 5). Chacun des pixels est représenté par une fréquence comprise entre 1500Hz (noir) et 2300Hz (blanc) pour la codification de son intensité pour chacune des composantes RVB (Rouge, Vert, Bleu).

TIMING SEQUENCE		
(1) "Starting" sync pulse (first line only!)	9.0ms	1200hz
(2) Separator pulse	1.5ms	1500hz
(3) Green scan		
(4) Separator pulse	1.5ms	1500hz
(5) Blue scan		
(6) Sync pulse	9.0ms	1200hz
(7) Sync porch	1.5ms	1500hz
(8) Red scan		

fig 4 : Timing et signaux pour le mode Scottie 1

Après la transmission de la première ligne, les étapes 2 à 8 sont répétées pour les lignes suivantes. La durée totale de transmission d'une image est d'environ 110,5 secondes pour une horloge de pixel donnée à 432µS. La figure 5 détaille le format d'une trame selon le protocole retenu. Elle fait apparaitre les différents signaux de synchronisation, le VIS code et la succession de codage des pixels d'une ligne sous forme de 3 informations de variation de couleurs (3 lignes monochromatiques). Cette technique permet au dispositif de réception de reconstituer l'intégralité des différentes couleurs pour chacune des lignes et permet de réduire les brusques variations de fréquences.

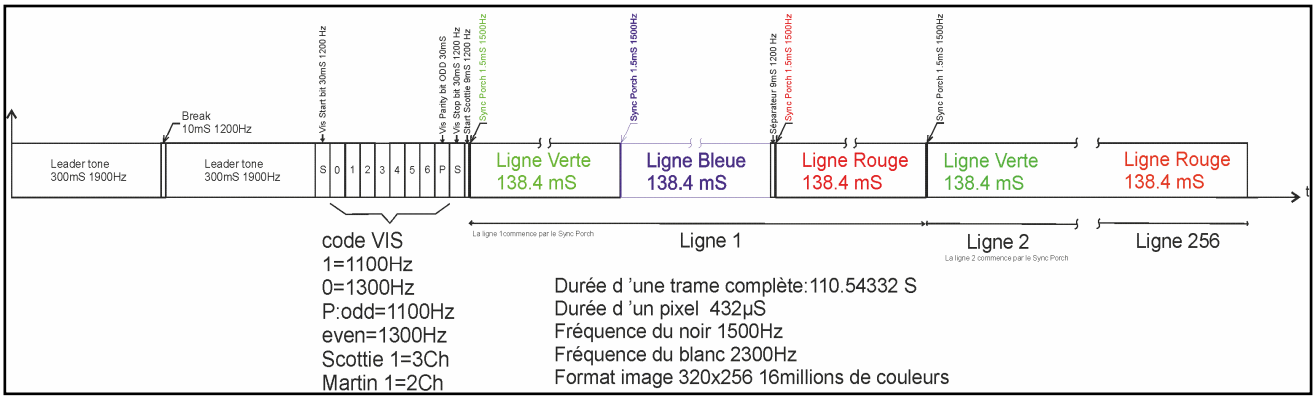


fig : 5 : Constitution d'une trame du protocole SSTV Scottie 1

4 CHOIX DE L'ARCHITECTURE MATERIELLE ET PROGRAMMATION OPTIMALE

4.1 Dispositif d'acquisition : la CMUCAM3

La génération d'un signal SSTV nécessite l'usage d'un ordinateur équipé d'une carte son et de logiciels spécifiques (généralement freeware). Cette approche est évidemment incompatible avec nos exigences d'encombrement et de consommation dans un contexte CubeSat. La CMUcam3 est une carte électronique combinant une caméra I2C et un microcontrôleur ARM7 de Philips (cf figure 6). Cette caméra a été conçue à l'origine par l'Université de Carnegie Mellon pour des applications de robotique mobile. Son principal avantage est la disponibilité gratuite de ses outils de développement et d'une importante source de documentation et d'applications (essentiellement de traitements d'images) via son site Web collaboratif [6].

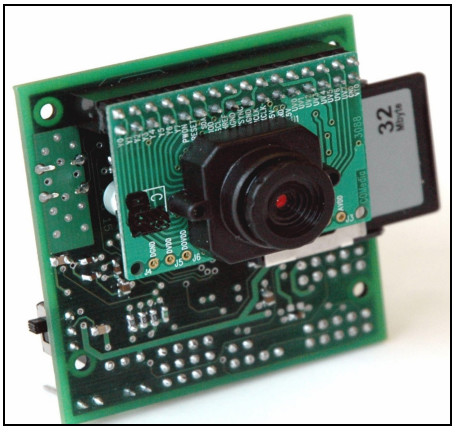


Figure 6 :CMUCAM3 utilisée

Bien que ne répondant pas aux exigences spécifiques de l'environnement spatial (plage de fonctionnement de température non compatible), ce module a été retenu car nos étudiants maitrisent l'architecture et la programmation (assembleur et C) des microcontrôleurs ARM7 et ARM9 (cours M1 et M2 du master SET). L'objectif de ce projet restant l'étude de faisabilité d'intégration de la fonctionnalité SSTV sur une cible de faible encombrement et de portage rapide sur d'autres cibles.

4.2 Programmation des fonctionnalités pour la transmission SSTV

Afin de répondre à notre application les étudiants ont développé les fonctionnalités suivantes

- Programmation d'un timer pour une mise en place d'une interruption périodique à la fréquence pixel
- Configuration et gestion de la caméra via son bus I2C (code disponible en [6])
- Prise et récupération de l'image dans la ram de l'ARM code disponible en [6])
- Génération d'un signal sinusoïdal de fréquence variable au moyen de fonctionnalité PWM (Pulse Wave modulation) associée à la gestion d'une table d'échantillons des différentes valeurs de sinus stockées en mémoire Flash.
- Générateur de code VIS et des différents signaux de synchronisation
- Programme principal de capture et transmission d'image

L'intégralité de la gestion de ces différents modules est pilotée par l'interruption timer pixel durant les phases de transmission SSTV. Cette interruption est inactive durant les opérations de capture et de stockage de l'image en RAM.

4.2.1 PWM (Pulse Width Modulation)

Pour générer un signal sinusoïdal depuis un signal PWM, on module son rapport cyclique en respectant la loi de variation du sinus dont les coefficients sont stockés dans une table en mémoire Flash. Le signal obtenu est ensuite filtré (filtre RC) pour en extraire la sinusoïde et éliminer la pulsation PWM comme le montre la figure 7.

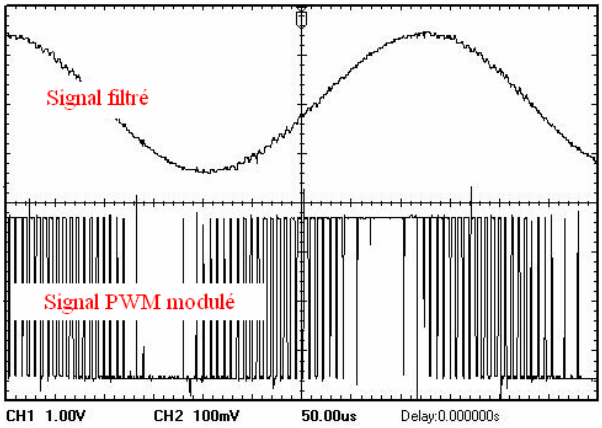
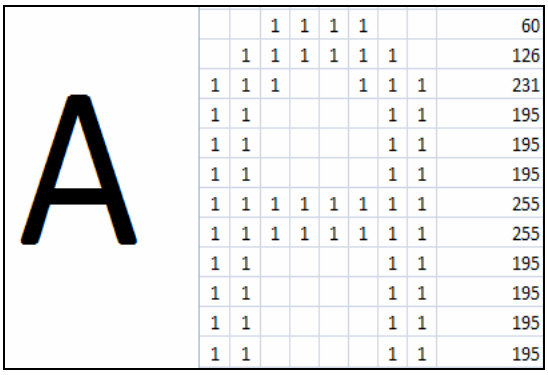


fig 7 : Sinusoïde obtenue depuis un signal PWM

4.2.2 Incrustation image

Bien que cette fonctionnalité ne soit pas une exigence dans le cadre de ce projet, elle a été proposée par les étudiants et intégrée dans la CMUCAM. En effet, il peut être pertinent de rajouter des informations supplémentaires sur l'image (indicatif, heure et date de prise de vue, etc..). La mise en place de cette incrustation repose sur la modification des pixels de l'image prise dans la ram en travaillant sur une table de représentation graphique (Figure 8) des principaux caractères en mémoire Flash. Les principaux paramètres d'incrustation (chaine de caractères, variable, positions dans l'image) sont saisis via une interface USB et stockés en EEPROM.



		1	1	1	1		60
	1	1	1	1	1	1	126
1	1	1			1	1	1	231
1	1					1	1	195
1	1					1	1	195
1	1					1	1	195
1	1	1	1	1	1	1	1	255
1	1	1	1	1	1	1	1	255
1	1					1	1	195
1	1					1	1	195
1	1					1	1	195
1	1					1	1	195

fig : 8 : Matrice du caractère "A"

4.2.3 Interface utilisateur

Nous avons doté la CMUcam3 d'une interface USB qui émule un port série et permet le contrôle ou la configuration de la caméra depuis un terminal (figure 9). Nous pouvons prendre et envoyer des photos, charger des mires de différentes résolutions, définir les paramètres de l'incrustation et l'utiliser comme outil de débogage.

```
Console de controle de la CMUcam3 SSTV
Identifiant utilisé : F1UFP***************************
Faîtes votre choix :
        1 - Photo
        2 - Mire de couleur
        3 - Mire de niveaux de gris
        4 - Modifier indicatif
                Choix : _
```

fig 9 : Interface utilisateur de la caméra

4.2.4 Informations logicielles et validation

Pour caractériser les pertes de la chaine de codage/décodage sans utiliser de canal radio nous nous sommes appuyés sur l'option de chargement d'une mire couleur connue via l'interface USB, de son encodage et de sa transmission. Nous avons mis en place une liaison directe entre la sortie BF de la CMUCAM et l'entrée BF d'un ordinateur, le décodage étant réalisé avec le logiciel MMSSTV (figure 10). Les seules différences entre la mire et l'image décodée sur l'ordinateur concernent de petites variations de couleurs (non visible à l'œil nu) dues à la résolution du signal PWM. Aucune perte des signaux de synchronisation n'a été constatée.

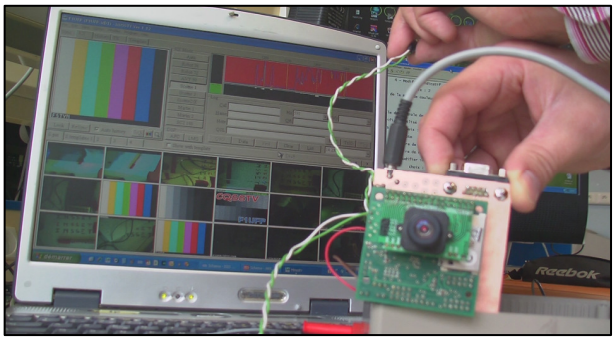


fig : 10 : CMUcam3 et carte d'interface BF en test

5 CONCLUSION

Le code source est développé en langage C, ce qui le rend portable sur d'autres plateformes, éventuellement avec d'autres sources vidéo pilotées via bus I2C. Le code

exécutable représente environ 60 kilo-octets (soit 50% de la capacité du ARM) pour environ 1500 lignes de codes source C. La possibilité de transmission d'images SSTV depuis la CMUCAM équipée d'un ARM7 est validée. Il nous reste à intégrer la gestion d'une carte SD afin d'étendre la capacité de stockage (actuellement limitée par la RAM du ARM7) et d'autoriser une transmission d'une image particulière (à la demande) depuis une station au sol.

La réussite de cette étude nous a mené à créer un produit dérivé pour les radioamateurs de la FNRASEC (Sécurité Civile) dans le cadre de leur mission de recherche de balises aviation (plan SATER). Afin de répondre aux besoins de ces opérateurs, nous avons intégré à notre dispositif un interrupteur, un bouton poussoir et une gestion automatique d'un signal PTT (Push To Talk). Cette adaptation autorise une prise de photo par un opérateur mobile et son envoi différé via la prise micro d'une radio VHF à destination du poste de commandement fixe comme l'illustre la figure 11.

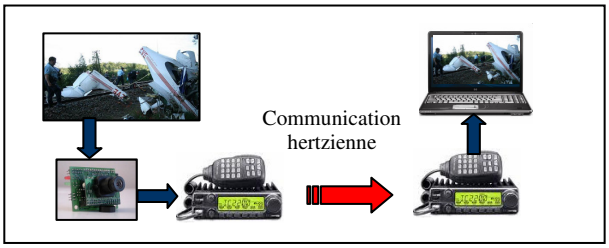


fig 11 : CMUcam3 pour plan SATER

6 CONTEXTE DE TRAVAIL

Ce projet a été réalisé par une équipe de trois étudiants composée d'un élève en 2ème année et deux de 1ère année de Master parcours « Systèmes Embarqués dans les Transports ». Cela représente une charge de travail d'environ 10h par semaine sur une durée de huit mois soit un quota de 320 heures de travail.

Les étudiants, encadrés par l'équipe d'enseignants-chercheurs, ont pu utiliser les instruments de mesures et tout le matériel nécessaire mis à leur disposition au sein du laboratoire de l'INSSET pour la réalisation de ce projet. De nombreuses réunions ont permis de définir les exigences (poids, consommation, encombrement, portabilité) afin de disposer d'une solution opérationnelle et évolutive. L'usage d'une cible reprogrammable (FPGA intégrant un ARM) associée à une caméra compatible avec les exigences de l'environnement spatial a été d'ores et déjà planifié.

Des démonstrations seront prévues lors du colloque et le code source sera mis à disposition sur notre site Web.

Bibliographie

[1] INSSET de Saint-Quentin : http://www.insset.u-picardie.fr/site/master_systemes_embarques

[2] V. Bourny, T. Capitaine, J. Senlis, M. Hamzaoui, A. Lorthois, L. Duval « Plate-forme PROMOCO (plate-forme Robotique Mobile Communicante). Présentation des impacts de formations, de prestations industrielles et de recherche » *Revue J3eA, n : °3, 09 Février 2010.*

[3] GENSO : http://www.genso.org/index.php/credits

[4] CUBSESAT : http://www.insset.u-picardie.fr/file/Recherche/flyercubesat.pdf

[5] SSTV Protocol: http://en.wikipedia.org/wiki/Slow-scan_television

[6] CMUCAM WebSite: http://www.cmucam.org/