



Guillaume BASSERIE

**Master Science De L'ingénieur
D'Informatique Industrielle et Systèmes Automatisés**

Projet ALFA

(Auroral Light Fine Analysis)



**Rapport de stage de fin d'étude
Sous la tutelle d'Elena SERAN**



Promo 2004 – 2006

SOMMAIRE

REMERCIEMENTS	3
I - INTRODUCTION	4
II - PRESENTATION DU CETP.....	5
2.1 - Le CETP.....	5
2.1.1 - <i>Thèmes de recherche</i>	5
2.1.2 - <i>Les différents départements scientifiques</i>	6
2.2 - Résultats Scientifiques Majeurs.....	7
2.2.1 - <i>Plasmas et Planètes</i>	7
2.2.2 - <i>Atmosphère météorologique et interfaces</i>	7
2.3 - Le phénomène physique et les moyens des mesures	8
2.3.1 - <i>Histoire de la découverte des zones aurorales</i>	8
III - PRESENTATION DU PROJET	9
3.1 - L'objectif du projet	9
3.2 - Le phénomène physique observé	9
3.3 - L'instrumentation d'ALFA	12
3.4 - Plein ciel.....	14
3.5 - Utilité des différents appareils	14
IV - LES ETAPES DU PROJET.....	15
4.1 - La compression JPEG	16
4.1.1 - <i>Définition</i>	16
4.1.2 - <i>Le Fonctionnement</i>	16
4.1.3 - <i>La DCT</i>	17
4.1.4 - <i>Le codage de Huffman</i>	20
4.2 - Les fichiers JPEG.....	22
4.2.1 - <i>Rappel pour la lecture des fichiers JPEG</i>	22
4.2.2 - <i>Structure des fichiers JPEG</i>	23
4.2.3 - <i>Détail d'un en-tête de type EXIF</i>	23
4.2.4 - <i>Les zones des données EXIF</i>	24
4.2.5 - <i>La structure des données EXIF</i>	24
4.2.6 - <i>Structure d'un répertoire</i>	25
4.3 - Le choix de la bibliothèque graphique.....	31
4.3.1 - <i>Pourquoi la bibliothèque du groupe indépendant JPEG</i>	31
4.3.2 - <i>Les étapes de la construction d'une librairie statique avec « Labwindows »</i>	32
4.3.3 - <i>Améliorations</i>	37
4.3.4 - <i>Détail des modifications dans la librairie</i>	37
4.3.5 - <i>Les programmes réalisés</i>	38
4.4 - L'appareil photo.....	39
4.4.1 - <i>Caractéristiques</i>	39
4.4.2 - <i>Le super CCD (Charged Couple Device)</i>	40

4.4.3 - <i>L'implantation des CCD</i>	40
4.4.4 - <i>L'interpolation</i>	41
4.4.5 - <i>Fonctionnement du super CCD</i>	41
4.4.6 - <i>Les objectifs utilisés</i>	42
4.5 - <i>Le GPS</i>	43
4.5.1 - <i>Détail de fonctionnement du DGPS</i>	44
4.5.2 - <i>Schéma DGPS</i>	44
4.5.3 - <i>L'utilisation du GPS dans le projet</i>	45
4.5.4 - <i>La communication avec le GPS</i>	45
4.6 - <i>Le port parallèle</i>	48
4.6.1 - <i>Définition</i>	48
4.6.2 - <i>Communication sur le port parallèle</i>	49
4.6.3 - <i>Les adresses des ports parallèles</i>	51
4.6.4 - <i>Mode de fonctionnement</i>	52
4.6.5 - <i>Détail des 3 premiers ports</i>	55
4.6.6 - <i>L'écriture</i>	55
4.6.7 - <i>La lecture</i>	56
4.6.8 - <i>Le choix</i>	58
4.6.9 - <i>Les tests</i>	58
4.6.10 - <i>Aperçu de l'interface réalisée</i>	61
4.7 - <i>Les acquisitions sonores</i>	62
4.7.1 - <i>L'objectif</i>	62
4.7.2 - <i>Le wave</i>	63
4.7.3 - <i>Le PCM</i>	64
4.7.4 - <i>Les choix</i>	65
4.7.5 - <i>Les essais</i>	65
4.8 - <i>L'architecture temps réel</i>	70
4.8.1 - <i>Définition</i>	70
4.8.2 - <i>Les threads</i>	70
4.8.3 - <i>Architecture</i>	71
4.8.4 - <i>Détail des opérations</i>	72
4.9 - <i>Détection sur l'image du centre et des bords de l'objectif</i>	73
V - <i>CONCLUSION</i>	75
VI - <i>BIBLIOGRAPHIE</i>	76
VII - <i>ANNEXE</i>	77

REMERCIEMENTS

Je remercie tout spécialement mon tuteur, Elena SERAN, pour m'avoir transmis ses connaissances sur les phénomènes physiques étudiés et ces explications mathématiques, ainsi que, Fabrice BERTRAND, pour son aide au niveau informatique et aussi Michel GODEFROY pour ses explications au niveau électronique.

Je tiens à remercier toute l'équipe du CETP St-Maur, pour son accueil toujours chaleureux et pour sa disponibilité.

Je fais aussi un petit clin d'œil aux membres du CETP qui étaient fidèles au post au mois d'août et avec qui, abandonnés de tous, même du cuisinier, nous avons fait d'inoubliable barbecue.

Je remercie encore tous les sportifs du laboratoire avec qui j'ai passé de bon moment de détente.

I - INTRODUCTION

J'ai effectué mon stage au sein d'un laboratoire de recherche, le « Centre d'Etude des Environnements Terrestres et Planétaires » de St-Maur-les-Fossés.

J'étais sous la tutelle de l'Astronome-adjointe Elena SERAN avec qui j'ai travaillé en collaboration avec Michel GODEFROY.

Le projet auquel j'ai participé est le projet ALFA (Auroral Light Fine Analysis).

Et plus particulièrement sur la première partie de son système de mesures « Plein ciel », car la deuxième partie « Haute résolution spatiale » a son développement prévu pour 2007.

Ce projet a pour mission d'étudier et de caractériser la vitesse des ions qui sont précipités au niveau des zones polaires et le positionnement de ces couloirs d'accélération que représentent les aurores boréales.

Mon travail consistait à :

- Trouver une librairie graphique gratuite pour manipuler des images au format JPEG.
- Réaliser des programmes pour effectuer des tests et caractériser l'appareil photo à notre disposition.
- Etablir un programme de communication avec notre GPS pour ne récupérer que les informations voulues.
- Définir les différents modes de communication avec le port parallèle d'un ordinateur.
- Créer un programme d'acquisition sonore (WAVE) avec modification des paramètres et traitement des résultats.
- Mettre en place une architecture temps réel pour le pilotage des différents appareils.
- Intégration de différents programmes de commande du prototype.
- trouver des améliorations en continu pour la rapidité de réaction des programmes.

II - PRESENTATION DU CETP

2.1 - Le CETP

Centre d'Etude des Environnements Terrestres et Planétaires (CETP) est une Unité Mixte de Recherche du Département Sciences de l'Univers du CNRS et de l'Université de Versailles Saint-Quentin au sein de l'Institut Pierre Simon Laplace.

Il regroupe 120 permanents, chercheurs, ingénieurs et administratifs et accueille régulièrement une trentaine d'étudiants en thèse, visiteurs ou stagiaires.

Il est implanté actuellement sur 2 sites, une centaine de permanents travaillent sur le site du Centre Universitaire de Vélizy, le reste du Laboratoire est implanté sur le site de l'Observatoire du Parc de Saint-Maur où j'ai effectué mon stage. Le domaine scientifique du CETP concerne tous les aspects de la Physique de l'environnement du système solaire, depuis la couronne solaire et le vent solaire, jusqu'aux surfaces et même sous-sols planétaires, en incluant le sondage des premiers kilomètres de la planète Mars et les processus liés à l'absorption de l'eau par la couche racinaire de la Terre, en passant par le dégazage des comètes. Dans la pratique, deux grands domaines d'études concernent l'activité scientifique du Laboratoire :

- les études de l'atmosphère et des processus d'interface avec les diverses surfaces de notre planète,
- et l'étude des plasmas du système solaire et des environnements planétaires.

2.1.1 - Thèmes de recherche

Le CETP s'intéresse principalement à trois grandes thématiques.

La première est centrée sur le cycle de l'eau et sur l'interaction et les échanges entre les surfaces continentales, océaniques et l'atmosphère ; la formation et la structure des systèmes précipitants.

La seconde thématique porte sur l'étude de l'interaction du vent solaire (expansion de l'atmosphère ionisée du soleil) et les objets du système solaire : les planètes, dont la Terre, et les comètes. Elle inclut des travaux en physique des plasmas et en planétologie.

La troisième, qui nous concerne plus, porte sur des travaux méthodologiques : le

CETP fait largement appel aux sondages et mesures radioélectriques. Il s'agit de comprendre les propriétés émettrices de l'atmosphère et des surfaces dans les domaines des micro-ondes,

infra-rouge et visible, d'interpréter les signaux rétro diffusés par l'atmosphère et les surfaces lorsqu'elles sont « éclairées » par des ondes radar.

Dans cette thématique sont également conduits des travaux sur le traitement des signaux et des images produites par télédétection. L'ensemble de ces travaux fournit des résultats importants pour l'étude du climat et des processus météorologiques, ainsi que pour la compréhension des interactions entre le soleil et la Terre. Ils le sont également pour les télécommunications car les performances des liaisons en hyperfréquences dépendent très largement des caractéristiques de l'atmosphère et des précipitations et, ils servent à l'amélioration de la fiabilité et de l'efficacité des satellites opérationnels puisque ces travaux permettent de mieux connaître les effets du milieu environnant.

L'activité du Laboratoire est à dominante instrumentale, avec une forte composante spatiale, cependant la modélisation est très présente dans la thématique plasmas et tient une place non négligeable pour l'étude des interactions de l'atmosphère avec les surfaces continentales et pour la compréhension des processus atmosphériques.

L'instrumentation que développe le Laboratoire repose principalement, pour les deux grands domaines thématiques, sur les méthodes actives et passives en radio-électricité, dans deux domaines de fréquences bien distincts : inférieures au MHz pour les études des plasmas, supérieures ou de l'ordre du GHz pour les études atmosphériques. Aux côtés de ces méthodes radio, le CETP est un spécialiste de la spectrométrie de masse pour les milieux très dilués, plasmas ou neutres.

2.1.2 - Les différents départements scientifiques

Le CETP compte deux départements scientifiques pour les études des plasmas :

- Ondes dans les Plasmas Naturels (OPN).
- Electrodynamique des Milieux Ionisés (EMI).

Ainsi que deux pour les études atmosphériques :

- Atmosphère Basse et Moyenne (ABM).
- Interfaces Océan, Terre, Atmosphère (IOTA).

Un département rassemble les études méthodologiques :

- Electromagnétisme et Méthodes d'Analyse (EMA).

Aux côtés de ces départements scientifiques, les moyens techniques sont regroupés dans deux départements :

- Instrumentation Terrestre et Spatiale (ITS) pour l'instrumentation et l'électronique.
- Informatique Distribuée et Applications (IDA) pour les moyens informatiques qu'ils soient destinés aux études scientifiques et à l'instrumentation (électronique temps réel) ou à l'équipement général du Laboratoire (bureautique).

Enfin les moyens administratifs et généraux sont regroupés dans le Département :

- Moyens Généraux Communs (MGC).

2.2 - Résultats Scientifiques Majeurs

2.2.1 - Plasmas et Planètes

Pour l'activité scientifique du domaine des plasmas et les planètes, ces dernières années ont été dominées par la mission CLUSTER, désormais en phase opérationnelle.

Au-delà des acquis scientifiques, la mission CLUSTER a ouvert la voie aux missions multi-satellites, qui vont se multiplier dans un proche avenir. De nouvelles méthodes d'interprétation et de nouvelles stratégies d'observation ont été développées par le CETP, en collaboration avec d'autres laboratoires, au premier rang duquel il faut placer le LPCE (Orléans) et le CESR (Toulouse), cet investissement dans des travaux méthodologiques place le CETP en excellente position pour aborder les missions multi-satellites à venir, à commencer par les missions NASA : THEMIS puis MMS dédiées notamment à l'étude de processus éruptifs comme les sous-orages magnétosphériques.

Ont été également développées des stratégies d'observations conjointes d'instruments au sol et de mesures in situ, en satellite.

2.2.2 - Atmosphère météorologique et interfaces

Dans le domaine de l'atmosphère météorologique et des interfaces, l'activité scientifique du CETP est principalement orientée vers l'étude des processus aux échelles inférieures ou de l'ordre de celles des mailles des modèles, en vue de leur meilleure représentation dans ces processus.

Les études sur la quantification et l'analyse des flux à l'interface océan-atmosphère se sont renforcées et ont conduit à des liens scientifiques plus étroits avec les océanologues, au premier rang desquels il faut compter le LODYC.

2.3 - Le phénomène physique et les moyens des mesures

2.3.1 - Histoire de la découverte des zones aurorales

Les aurores polaires, phénomènes naturels observables depuis les hautes latitudes, constituent probablement l'un des plus impressionnants spectacle de la nature. Observées depuis longtemps par les peuples nordiques comme le relatent leurs légendes, leurs premières études furent entreprises par PYTHEAS, le navigateur phocéén du 5ème siècle avant notre ère, qui a observé des aurores boréales lors de son voyage en mer du Nord.

Il faut attendre le 18ème siècle pour que des tentatives d'explications scientifiques du phénomène soient proposées. La présence d'émissions aurorales autour du pôle nord magnétique suggère une relation avec le champ magnétique terrestre et un lien fût établi entre magnétisme terrestre et activité solaire en 1879 lorsque H. BECQUEREL propose une explication par la précipitation de particules chargées originaires du soleil guidées par le champ magnétique terrestre jusqu'aux régions aurorales.

Des observations réalisées par K. BIRKELAND en 1902 et 1903 au nord de la Norvège ont montré que des courants électriques circulant le long des lignes de champ magnétique pouvaient être à l'origine des aurores.

Au début du 20ème siècle, A. KENNELLY et O. HEAVISIDE ont émis l'existence de l'ionosphère, couche supérieure ionisée et conductrice de l'atmosphère terrestre, pour expliquer les transmissions radio-transatlantiques (si on projette une onde vers le ciel de façon non verticale ; par phénomène de réflexion contre la couche de l'ionosphère l'onde est redirigée vers le sol).

L'existence de l'ionosphère est alors établie et son altitude mesurée dans les années

1920 par l'envoi d'impulsions radio à incidence verticale, tandis que les émissions aurorales visibles ont été identifiées par spectroscopie par H. BADCOCK.

Depuis, l'étude de l'ionosphère n'a plus cessé, elle s'est même étendue avec la possibilité de faire des mesures directement dans l'espace grâce aux satellites.

III - PRESENTATION DU PROJET

Le projet a pour nom : ALFA (Auroral Light Fine Analysis).

Son investigateur principal est : Elena SERAN

Co. Investigateur : Jean Claude CERISIER

Responsable Expérimental : Michel GODEFROY

Tous rattachés au laboratoire du CETP / IPSL

3.1 - L'objectif du projet

Son but et le développement d'instruments optiques pour effectuer des mesures fines des arcs auroraux. A terme le projet inclura deux systèmes différents :

- Plein ciel
- Haute résolution spatiale

Ils permettront d'observer des structures lumineuses à une altitude de ~200 kilomètres avec une résolution horizontale entre 10 et 100 mètres. Ce projet fera le lien entre les projets présents et futurs, et permettra d'étudier les structures aurorales et les phénomènes associés (précipitations de particules, courants alignés, structures de champ électrique et de convection) sur une grande gamme d'échelles transversalement par rapport au champ magnétique.

Aux grandes échelles (10-100 km) observées par les projets SuperDARN, Image, Themis et aux moyennes échelles (1-10 km) couvertes par les missions Oersted, CHAMP, MMS, il existe une volonté d'adjoindre les petites échelles (10m-1km) par l'utilisation d'un ensemble comprenant des caméras "plein ciel" et "haute résolution spatiale".

3.2 - Le phénomène physique observé

Le transfert d'énergie et de masse entre le vent solaire et la magnétosphère de la Terre est effectué essentiellement à travers les frontières magnétosphériques (magnétopause et queue). L'énergie est principalement portée par les particules chargées qui s'accélèrent le long des lignes du champ magnétique en effectuant leur parcours entre ces frontières et l'ionosphère. La

radiation observée dans les arcs auroraux dans le spectre visible est due à l'interaction entre les électrons entrant dans l'ionosphère (avec une énergie initiale comprise entre ~ 0.5 et 10 keV) et les atomes et molécules neutres rencontrés sur leur chemin. L'énergie cinétique des particules énergétiques provoque l'ionisation de ces « neutres » et le changement de niveau d'excitation de ceux-ci. L'excitation est accompagnée d'une émission de photons à une longueur d'onde qui est unique pour chaque « neutre » et liée au niveau de l'excitation (figure 1).

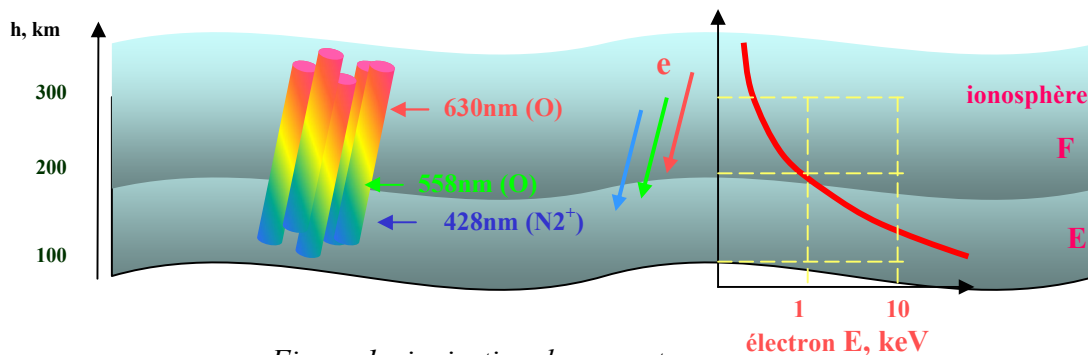


Figure 1 : ionisation des « neutre »

La pénétration des électrons dans les couches basses de l'ionosphère dépend de l'énergie initiale des électrons et de la composition de l'atmosphère neutre. Le spectre en longueur d'onde observé dans le visible est directement associé à l'énergie des électrons précipités et l'intensité de l'émission est proportionnelle au flux d'électrons.

La dégradation de l'énergie depuis les grandes échelles vers les échelles plus petites fait intervenir des processus variés parmi lesquels les phénomènes de turbulence et de filamentation (phénomène qui conduit à une concentration de l'énergie de l'onde en de fins filaments magnétiques parallèles à la direction de propagation). L'association de deux appareils photos, "plein ciel" et "haute résolution spatiale", permettra une mesure en 2D du phénomène sur la gamme complète d'échelles concernées.

Dans la mesure du possible et des opportunités, ces mesures seront couplées avec celles d'autres observables (convection, précipitations de particules et structures de courants alignés).

Les observations d'émissions dans le spectre visible qui seront effectuées avec l'expérience ALFA permettront de :

- mesurer :
 - Les structures lumineuses de tailles différentes, i.e. 10, 100 et 1000 m ;
 - Le spectre entier d'émission en visible ;
 - L'émission associée à chaque longueur d'onde spécifique ;
- analyser :
 - La relation entre des irrégularités d'émissions d'échelles différentes ;
 - La variation d'énergie des électrons précipités et de la densité de courant portés par les électrons énergétiques;
 - La fermeture du courant aligné aux lignes du champ magnétique ;
- explorer :
 - La couche de l'ionosphère entre ~100 et 350 km (régions E et F) ;
 - Une région de ~1000 km dans le plan horizontal ;
- corréler les résultats avec des mesures conjuguées de :
 - Radars HF de superDARN et de magnétomètres au sol ;
 - Magnétomètre, antennes électriques, analyseurs plasma et caméras UVI à bord de satellites ionosphériques (Oersted, Demeter) et magnétosphériques (Image, Thermis, MMS).

Par la suite ALFA sera intégré dans le réseau SuperDARN comme une composante essentielle et complémentaire, car il apportera l'information sur :

- les sources de la convection du plasma dans l'ionosphère observée par les radars ;
- les cibles de la rétro-diffusion d'ondes HF (8-20 MHz) ;
- les raisons des pertes du signal rétro-diffusé vers les radars dans les zones de la précipitation d'électrons énergétiques.

3.3 - L'instrumentation d'ALFA

Elle se décompose en deux parties :

- plein ciel (PLC)
- haute résolution spatiale (HRS)

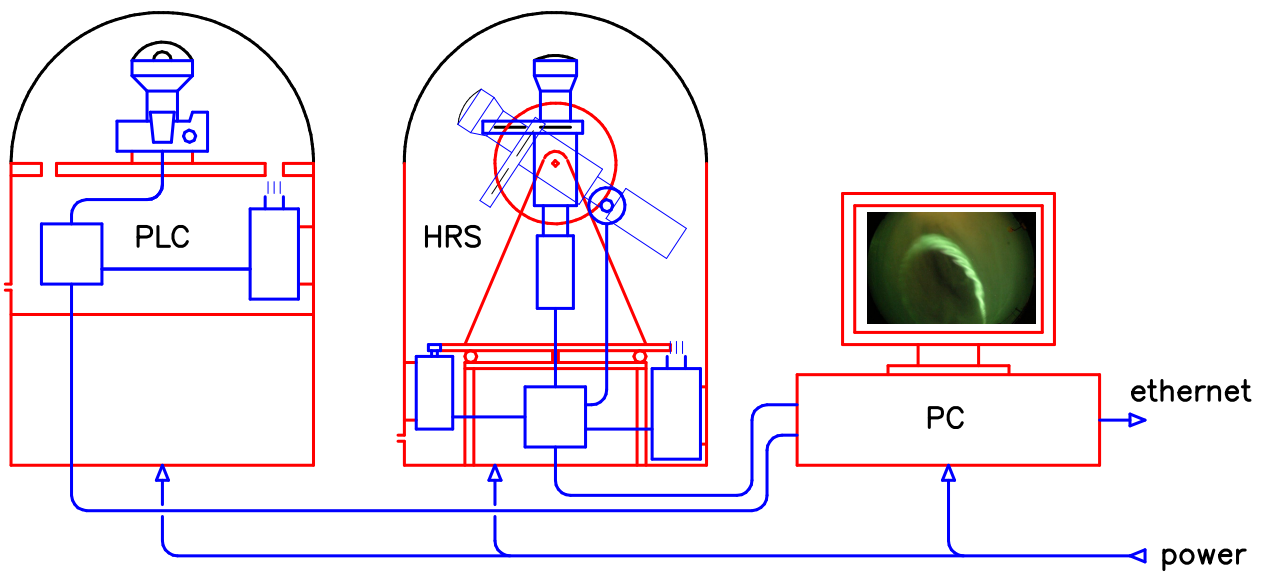


Figure 2 : L'instrumentation d'ALFA

Le contrôle et le traitement des données, des deux systèmes sont ensuite réalisés par l'intermédiaire d'un ordinateur.

3.4 - Plein ciel

La première partie du projet ALFA consiste au développement de l'instrument « plein ciel ». C'est dans la création de ce système qu'intervient mon projet de stage.

Le principal travail qui m'a été confié fut de gérer la partie informatique pour l'élaboration de programmes pour le traitement de données mais aussi le dialogue entre les différentes fonctions de contrôles et des appareils du dispositif. Tout au long du stage d'autres tâches me furent confiées, notamment des études de recherche pour la mise en place du système.



Figure 3 : Plein ciel

Cette première étape a pour objectif de réaliser une manipulation qui permet de faire des acquisitions de photos grand angle pour couvrir un large secteur et de connaître la position du dispositif.

Les images collectées sont ensuite traitées pour récolter les premières informations concernant la présence de données intéressantes.

La composition de l'instrument « plein ciel » est :

- Un appareil photo FinePix S3 Pro de chez Fuji.
- Un GPS Lassen LP de chez Trimber Navigation.
- Une batterie d'alimentation de tension 14,4V.
- Deux convertisseurs DC/DC
 - o 12 V pour le GPS.
 - o 5 V pour l'appareil photo.
- Un boîtier électronique de régulation de température gérant :
 - o La ventilation et le chauffage.
 - o La mesure de température.



Figure 4 : Plein ciel sans bâti

- Un niveau à bulle.
- Un boîtier de commande externe.
- Un boîtier de protection avec dôme en plexiglas.

Le système plein ciel est commandé par un ordinateur et plus précisément un programme temps réel écrit en langage C.

3.5 - Utilité des différents appareils

Comme on dans la figure 5, la batterie est le cœur du système, il alimente l'appareil photo et le GPS via les deux convertisseurs de tension continue.

L'appareil photo est notre appareil de mesure, il réalise les acquisitions d'images qui sont ensuite traitées pour récolter des informations.

Le GPS donne l'information sur la date et l'heure précise toutes les secondes, la position en latitude, en longitude et en altitude.

Le boîtier assure la protection du dispositif et son étanchéité du fait des zones climatiques dans lesquelles ils sera implanté (zone polaires). Le niveau à bulle permet de donner au système une bonne assise et un bon aplomb grâce aux pieds réglables.

Le régulateur de chauffage permet de maintenir une température faible dans l'habitacle pour éviter les bruits sur les photos dus à l'échauffement des capteurs CCD de l'appareil.

Le boîtier de commande externe permet de mettre le système sous tension via un interrupteur, de connecter le système à un PC, de choisir le seuil de la régulation de la température, mais aussi de raccorder une alimentation externe.

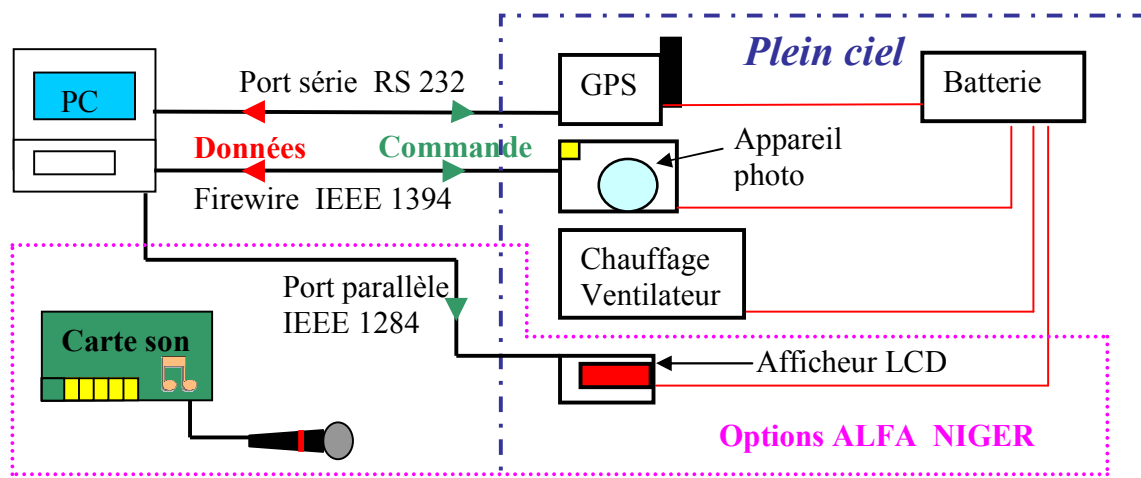


Figure 5 : Composition de Plein ciel

IV - LES ETAPES DU PROJET

Pour une réalisation objective du projet, ce dernier fut décomposé en une série de tâches dont certaines imposaient un ordre de priorité et d'autres pouvaient se réaliser en parallèle.

La toute première étape pour moi fut bien évidemment théorique : prendre connaissance du projet dans les grandes lignes mais aussi des caractéristiques imposées ou souhaitées pour sa réalisation. Une grande différence avec le monde industriel et aussi un point important qui caractérise le domaine de la recherche et du développement, tient dans le fait qu'il n'y a pas de cahier des charges, ou s'y on parvient à en établir un, il n'est que temporaire. Toutes parties restent ouvertes à des améliorations ou des modifications en fonction des réalisations et résultats obtenus.

La deuxième étape était plus pratique. Elle consistait à la prise en main du logiciel de programmation Labwindows développé par National Instrument. C'est un logiciel de programmation en langage C qui dispose de bibliothèques prédéfinies pour faciliter la conception d'interface graphique pour l'utilisateur.

Par la suite les étapes se composent toute d'une étude théorique et une réalisation pratique. Elles sont ci-après citées par thème.

LE JPEG

L'APPAREIL PHOTO

LE GPS

LE PORT PARALLELE

LES ACQUISITIONS SONORES

L'ARCHITECTURE TEMPS REEL

LES PROGRAMMES REALISES

LES ESSAIS

4.1 - La compression JPEG

Dans le projet, les mesures sont réalisées grâce à un appareil photo. Je récupère donc des images compressées et la toute première action avant de pouvoir accéder aux données est de décompresser.

L'appareil photo utilisé (S3Pro des chez « Fuji ») me fournit les photos sous forme de fichier « JPEG » avec un entête « EXIF » qui est le format le plus utilisé par les concepteurs d'appareils photos. La décompression est utilisée pour lire des fichiers JPEG lors de la caractérisation de l'appareil et aussi pour la lecture du buffer JPEG de ce dernier, lors du traitement en temps réel.

4.1.1 - Définition

Le mot JPEG (prononcé "jay-peg") est l'acronyme de Joint Photographic Experts Group, qui est le nom d'un comité créé en 1986 résultant de la fusion du CCITT (Commission Consultative Internationale de la Télégraphie et de la Téléphonie) et de l'ISO (International Standards Organisation).

Ce comité a donné son nom à la norme dite « ouverte » (libre d'utilisation) de compression d'images numériques JPEG, puis par extension au format de données et finalement au format de fichier le plus utilisé pour contenir ces données.

Cependant, en général le terme JPEG fait référence à la norme JPEG, formellement ISO/CEI 10918-1 ou UIT-T Recommandation T.81.

4.1.2 - Le Fonctionnement

Le JPEG est juste une méthode de compression mais n'est pas à proprement parlé un format : le format .jpg le plus utilisé est en fait le format JFIF (Jpeg Interchange Format), mais la méthode JPEG est aussi utilisée dans d'autres formats comme le TIFF ou le TGA, et même pour stocker les images dans les appareils photographiques numériques.

C'est une méthode de compression dite destructive (elle perd une partie des données). Elle est néanmoins très adaptée aux images naturelles pour lesquelles elle donne de très forts taux de compression pour une perte de qualité indécélable. En effet, elle exploite les défaillances de notre système de vision, mais en revanche elle convient mal aux images géométriques.

Cette norme utilise une méthode de compression basée sur le DCT (Discrete Cosine Transform). La transformée en cosinus discret est une transformation basée sur la transformée de Fourier discrète.

4.1.3 - La DCT

Comme la transformée de Fourier discrète, elle transforme un signal discret.

La différence entre la DCT et la transformée de Fourier se situe au niveau de la périodisation du signal. En effet pour pouvoir transformer un signal en sinusoïdes, il faut que celui-ci soit périodique. Mais quand il ne l'est pas, il est transformé afin de le devenir. Avec la transformée de Fourier, il est périodisé en reproduisant l'intervalle défini du signal « bout à bout » alors que la transformée en cosinus le fait en « dépliant » l'intervalle de sorte qu'il n'y ait plus de discontinuité aux bords (voir figure 6).



Figure 6 : Comparaison TF / DCT

Ainsi la DCT est plus adaptée aux images, car en principe le signal n'est pas du tout périodique et les discontinuités causées par la FFT provoquent l'apparition de fréquences élevées parasites, qui n'apparaissent pas avec la DCT.

La fonction représentant le signal est alors paire, et sa série de Fourier ne comporte plus que des termes en cosinus, d'où le nom de transformée en cosinus. De plus une transformée de Fourier donne des coefficients complexes (partie réelle et imaginaire), qui donnent ainsi une information sur la phase des sinusoïdes (le décalage dans le temps). Mais comme la DCT ne comporte plus de termes en sinus, elle donne des coefficients réels. Ceci est avantageux car les coefficients réels sont plus faciles à utiliser, et la phase n'est pas utile dans le cas des images.

Pour la DCT bidimensionnelle, le principe est exactement le même, mais les fréquences sont représentées sur deux axes. L'élément (0,0) est appelé composante DC (pour *Direct Component*) ou coefficient continu. Il représente la valeur moyenne des éléments avant transformation à un

coefficient près et est le plus grand coefficient de la matrice. Les autres éléments sont les composantes AC (*Alternative Component*) qui représentent l'amplitude des fréquences spatiales (horizontales et verticales). Plus on s'éloigne de la composante DC plus cela concerne des fréquences élevées.

On peut se représenter la DCT d'une matrice de 8 x 8 éléments ainsi :

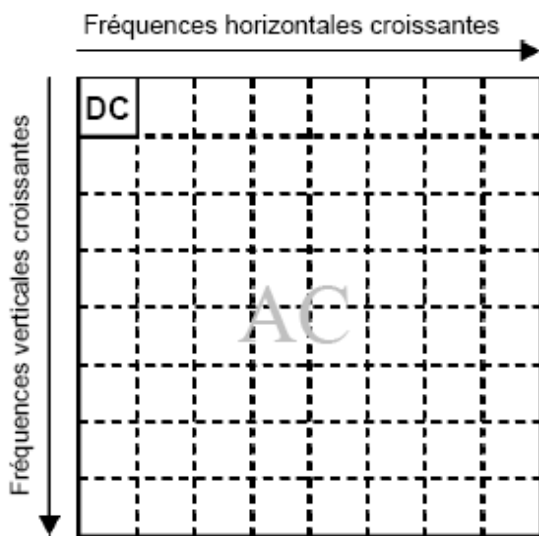


Figure 7 : DCT d'une matrice de 8 x 8

La transformation inverse existe, on l'appelle souvent IDCT et elle permet de restituer exactement les données de départ (en l'absence d'erreurs d'arrondis).

Les formules de la DCT bidimensionnelle et de son inverse sont les suivantes :

$$DCT(i, j) = \frac{1}{\sqrt{2N}} c(i) c(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{Img}(x, y) \cdot \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cdot \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

$$IDCT(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c(i) c(j) DCT(i, j) \cdot \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cdot \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

avec $c(a) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } a = 0 \\ 1 & \text{si } a > 0 \end{cases}$

Pour écrire le bloc dans un fichier il nous faut une suite d'octets et une matrice. Une étape consiste donc à linéariser la donnée. Pour cela le JPEG emploie une méthode particulière, dite de zigzag (figure 8), de façon à regrouper ensemble les éléments les moins importants.

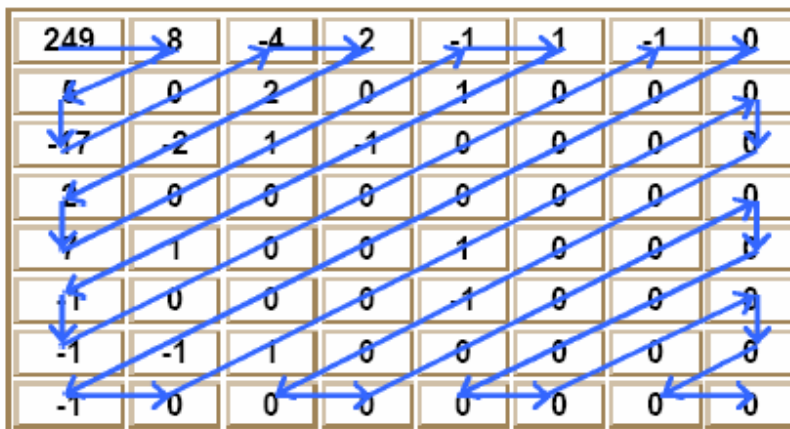


Figure 8 : Lecture zigzag

Les composantes DC sont généralement de grands nombres mais en revanche elles varient lentement d'un bloc à l'autre, puisque ce sont les moyennes de leur bloc. On va donc coder non pas la composante DC elle-même mais sa différence avec celle du bloc précédent, ce qui donnera de plus petits nombres prenant moins de place. Cette méthode est appelée DPCM (*Differential Pulse Code Modulation*). On applique ensuite un codage de Huffman.

4.1.4 - Le codage de Huffman

C'est un codage statistique, c'est-à-dire qu'il se base sur la fréquence d'apparition d'un caractère pour le coder : plus le caractère apparaît souvent plus son code sera court et vice-versa. C'est pourquoi on appelle aussi ce codage un **VLC** (*Variable Length Code*, code à taille variable) **préfixé**. En effet chaque code n'est le préfixe d'aucun autre, d'où *préfixé*, ce qui permet un décodage unique.

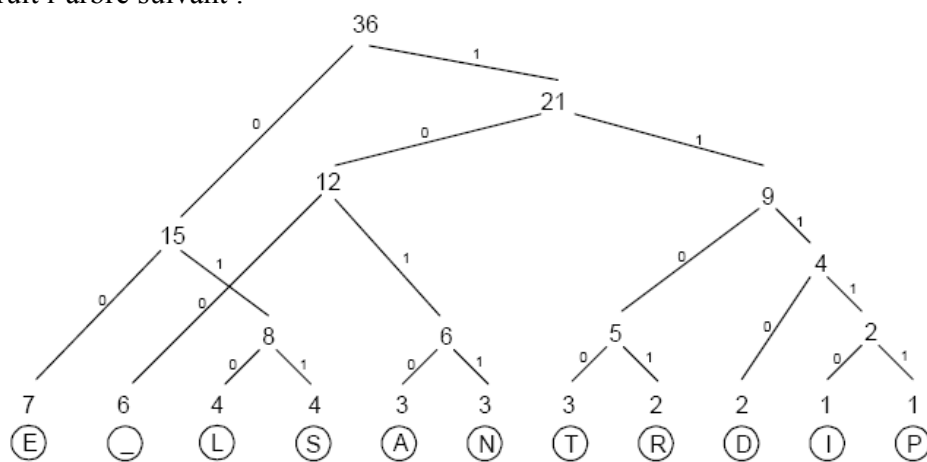
Dans la pratique, pour déterminer le code de chaque caractère on construit l'arbre de Huffman

- On calcule la fréquence d'apparition de chaque caractère (ou poids).
- On rassemble les deux caractères de plus faible poids pour former un noeud, dont le poids est égal à la somme des poids des deux caractères qui le composent.
- On affecte la valeur 0 au caractère le plus petit et 1 au caractère le plus grand des deux.
- On recommence les deux étapes précédentes en considérant chaque noeud formé comme un caractère, jusqu'à n'avoir plus qu'un noeud, la racine.

Par exemple pour la phrase « LE PRESIDENT EST ENTRE DANS LA SALLE » on a le tableau de fréquences d'apparition suivant :

E	Espace	L	S	A	N	T	R	D	I	P
7	6	4	4	3	3	3	2	2	1	1

Puis on construit l'arbre suivant :



Pour coder une lettre

Figure 9 : fréquences d'apparition

on part de celle-ci et

on remonte jusqu'à la racine en relevant le « chemin », et pour décoder on part de la branche et on choisit son « chemin » en suivant les 0 ou 1 du code pour finalement arriver à la lettre décodée.

Ainsi la lettre E la plus fréquente est codée sur 2 bits : 00, le caractère P le moins fréquent est codé sur 5 bits : 11111. On constate qu'ainsi on n'a pas besoin de dire quand le code est terminé puisqu'il est impossible de descendre plus bas.

La phrase compressée occupe alors $(7*2 + 6*3 + 4*3 + 4*3 + 3*4 + 3*4 + 3*4 + 2*4 + 2*4 + 1*5 + 1*5) = 118$ bits, au lieu de $(36 * 8) = 288$ bits, pour un gain de compression de $((288 - 118) / 288) = 59\%$.

4.2 - Les fichiers JPEG

Comme tous les fichiers, ils sont constitués d'un en-tête et des données « codées » de l'image. Un point important dans notre cas est que l'image « JPEG » est une photo et qu'elle est réalisée par un appareil photo. Or, il se trouve que les fabricant d'appareils photo ne se contentent pas de réaliser un fichier avec un en-tête de type JPEG, mais ils font un fichier de type EXIF qui contient l'en-tête JPEG.

4.2.1 - Rappel pour la lecture des fichiers JPEG

Les fichiers JPEG utilisent l'alignement « Motorola » dit aussi « Big Endian ». Le stockage des octets se fait du poids le plus fort au plus faible, c'est-à-dire dans le même sens que leur valeur. Les fichiers sont constitués de plusieurs sections repérées par des marqueurs (entiers non signés sur 2 octets et commençant par FF).

Le fichier commence toujours par un marqueur SOI (Start Of Image) dont la valeur vaut FFD8 et se termine toujours par le marqueur EOI (End Of Image) valant FFD9.

Détail d'un entête de type JPEG

A l'exception des marqueurs SOI et EOI, chaque section d'un fichier JPEG obéit à une structure générale : le marqueur est immédiatement suivi d'un mot (16 bits) qui indique le nombre d'octets de la section. Les données d'une section sont donc limitées à 65533 octets, leur taille étant incluse dedans.

Grâce à des marqueurs d'applications (Application Markers) définis de APP0 (\$FFE0) à APP15 (\$FFEF), certains clichés ajoutent à l'image elle-même des méta-données qui lui sont relatives, utilisables par les applications les lisant (vignette, et/ou données techniques). Ces marqueurs APPX suivent immédiatement le marqueur SOI, précédant donc le reste de l'image elle-même.

En ce qui concerne ces marqueurs d'applications un des premiers formats de fichiers pour le JPEG, le JFIF (Jpeg File Interchangeable Format) utilise le marqueur APP0. Ce format a été

supplanté par le format EXIF (EXchangeable Image File Format) pour structurer sa zone de données.

4.2.2 - Structure des fichiers JPEG

JPEG image	Contents	Name	Description
	0xFF 0xD8	SOI	Start of image
	Segments (see below)		
	0xFF 0xD9	EOI	End of image

JPEG segments	Description
	Segment marker (2 bytes)
	Segment size (2 bytes) excl. marker
	Segment data

Some JPEG segment markers		
Contents	Name	Description
0xFF 0xE0	APP0	Application marker (in every JPEG file)
0xFF 0xDB	DQT	Quantization Table
0xFF 0xC0	SOF0	Start of frame
0xFF 0xC4	DHT	Define Huffman Table
0xFF 0xDA	SOS	Start of scan

Figure 10 : Structure des fichiers JPEG

La liste proposée dans le tableau précédent n'est pas exhaustive mais regroupe les principaux marqueurs d'un fichier JPEG.

4.2.3 - Détail d'un en-tête de type EXIF

L'EXIF a sa zone de données matérialisée par un en-tête spécifique, elle organise les données en répertoires ou IFD (Image File Directory). Chaque répertoire contient une ou plusieurs entrées. Chaque entrée est identifiée par un **tag** correspondant à la signification du champ. De taille fixe, les entrées codent la nature du champ (octet, réel, chaîne...), sa taille, et la donnée elle-même, ou un offset vers elle.

Les IFD imposent certaines conditions : il doit exister au moins un IFD et chaque IFD doit comporter au moins une entrée. Bien entendu, le premier IFD doit toujours être relatif à l'image principale (car il peut y avoir plusieurs échantillons d'images) : IFD0 ou Main Image IFD. Les entrées d'un IFD doivent être dans l'ordre croissant des tags, numériques (*mots*).

4.2.4 - Les zones des données EXIF

Les zones des données EXIF démarrent au marqueur APP1, immédiatement après le marqueur SOI du fichier JPEG. Elles sont en quelque sorte insérée entre SOI et la première section du cliché sans méta-données (ce qui montre comment extraire la photo seule d'un cliché EXIF) :

```
FFD8 FFE1 LLLL data.....data (FFXX LLLL data..... data)xN FFD9
SOI APP1 Long Données EXIF N sections du cliché JPEG EOI
[-----> section EXIF <-----]
```

Comme toutes les sections JPEG, la longueur de la zone EXIF est codée par le *mot* qui suit son marqueur (APP1). Cette longueur vaut 2 octets de plus que les données elles-mêmes, et 2 de moins que la section entière !

4.2.5 - La structure des données EXIF

Elle débute par un en-tête spécifique, de 6 octets :

45 78 69 66 00 00

E x i f #0 #0

Répondant à la norme TIFF, on trouve ensuite l'en-tête TIFF, capitale, car elle définit le code pour l'alignement des données. A partir de cette zone, les données peuvent conserver l'ordre Motorola du reste du fichier JPEG, ou adopter le format Intel = Little Endian, qui stocke les octets de faible poids en premier :

4D 4D 00 2A 00 00 00 08

M M Value offset IFD0 : alignement Motorola

Ou

49 49 2A 00 08 00 00 00

I I Value offset IFD0 : alignement Intel, le plus fréquent

Suivant les 2 octets d'alignement, un mot de contrôle valant \$2A (42) permet de vérifier que l'on est bien en présence d'une section TIFF, et que l'on est dans le bon alignement.

Le cardinal (entier non signé sur 4 octets) qui suit est un offset vers l'IFD0, premier répertoire, relatif à l'image principale.

Cet offset, et tous ceux qui suivent, sont donnés à partir du premier octet de l'en-tête TIFF, en **rouge** dans le schéma ci-dessus. Cet offset vaut en général 8 (taille de l'en-tête), mais pas nécessairement.

Tous les offsets sont normalement alignés sur des mots, donc pairs. Les données sont ensuite organisées à l'intérieur de répertoires et sous-répertoires.

4.2.6 - Structure d'un répertoire

Chaque répertoire débute par un mot définissant le nombre d'entrées (données) de l'IFD.

Suivent les entrées elles-mêmes, chacune codée sur 12 octets : 2 mots, 1 cardinal, et 4 octets dont la signification dépend de ce qui précède. On a le schéma :

```
tg tg    ft ft      ct ct ct ct    vo vo vo vo
Tag Id  Format  [-- Count ---] Value / Offset
```

Le premier *mot* identifie la donnée. Les tags sont des identifiants uniques, attribués en principe par Adobe®, qui a créé le format TIFF. On trouve une revue de la signification des tags sur le site : <http://www.awaresystems.be>

Le second mot définit le format de la donnée (toute donnée entière non signée peut être lue comme un cardinal).

Le cardinal suivant représente le nombre de données du type défini. On peut ainsi avoir 5 données de type Double, soit un tableau de 5 réels.

Les 4 derniers octets de l'entrée stockent la valeur elle-même, ou un offset vers cette valeur, si sa taille dépasse 4 octets. Usuellement, la zone de stockage des données ainsi pointées suit l'IFD, mais cela n'est pas nécessaire. On se souviendra en écriture d'aligner les offsets sur les mots (donc écrire un octet nul après toute donnée de taille impaire, pour que l'offset suivant soit pair). Enfin, le répertoire se termine par un cardinal qui est l'offset vers le répertoire suivant (0 s'il n'y en a pas). Ci-après, sont listés les tableaux récapitulatifs de la construction des fichiers JPEG et EXIF pour la partie APP1.

Table de description des fichiers de compression JPEG

Offset (Hex)	Maker Segment	Name	Code (Hex)	Comment
0000	SOI	SOI Marker	FFD8	Start of Image
0002 0004 0006 000B 000C	APP1	APP1 Marker APP1 Length Identifier Pad APP1 Body	FFE1 1FBC "Exif"00 00	See Next Table
1FC0 1FC2 1FC4 1FC5 2005 2006	DQT	DQT Marker DQT Length Pq, Tq table elements Pq,Tq table elements	FFDB 0084 00 08060607.... 01 0909090C....	Define Quantization Table 64 Bytes of Data 64 Bytes of Data
2046 2048 204A 204B 205B 2067 2068 2078 211A 211B 212B 2137 2138 2148	DHT	DHT Marker DHT Length Th Li Vij Th Li Vij Th Li Vij	FFC4 01A2 00 00010501.... 00010203.... 10 00020103.... 01020300.... 01 00030101.... 00010203.... 11 00020102.... 00010203....	Table 0 for DC Table 0 for AC Table 1 for DC Table 1 for AC
21EA 21EC 21EE	DRI	DR Marker DRI Length Ri	FFDD 0004 0004	
21F0 21F2 21F4 21F5 21F7 21F9 21FA 21FB 21FC 21FD 21FE 21FF 2200 2201 2202	SOF0	SOF0 Marker SOF0 Length P Y X Nf C1 H1,V1 Tq1 C2 H2,V2 Tq2 C3 H3,V3 Tq3	FFC0 0011 08 01DB 0280 03 01 22 00 02 11 01 03 11 01	
2203 2205 2207 2208 2209 220A	SOS	SOS Marker SOS Length Ns Cs1 Td1,Ta1 Cs2	FFDA 000C 03 01 00 02	

220B		Td2,Ta2	11	
220C		Cs3	03	
220D		Td3,Ta3	11	
220E		Ss	00	
220F		Se	3F	
2210		Ah,Al	00	
2211	Image Data		F9FE....	Compressed Data
13BA4	EOI		FFD9	

Table de description des fichiers de compression EXIF APP1

Header	Offset (Hex)	Name	Data (Hex or "ASCII")			
	0000	Byte Order	4D4D ("II")			
	0002	42	2A00			
	0004	0th IFD Offset	08000000			
0th IFD	Offset (Hex)	Name	Interoperability Number (Hex)			
	0008	Number of Interoperability	0B00			
	Offset (Hex)	Name	Tag ID (Hex)	Type (Hex)	Count (Hex)	ValueOffset (Hex or "ASCII")
	000A	ImageDescription	0E01	0200	0E000000	92000000
	0016	Make	0F01	0200	09000000	A0000000
	0022	Model	1001	0200	08000000	AA000000
	002E	Orientation	1201	0300	01000000	01000000
	003A	XResolution	1A01	0500	01000000	B2000000
	0046	YResolution	1B01	0500	01000000	BA000000
	0052	ResolutionUnit	2801	0300	01000000	02000000
	005E	DateTime	3201	0200	14000000	C2000000
	006A	YCbCrPositioning	1302	0300	01000000	02000000
	0076	Copyright	9882	0200	15000000	D6000000
	0082	Exif IFD Pointer	6987	0400	01000000	EC000000
Offset (Hex)	Name	Offset (Hex)				
008E	Next IFD Offset	C402				
Value longer than 4byte of 0th IFD	Offset (Hex)	Name	Data (Hex or "ASCII")			
	0092	ImageDescription Value	"Exif_JPEG_420"00			
	00A0	Make Value	"DSCompany"00			
	00AA	Model Value	"Example"00			
	00B2	XResolution Value	4800000001000000			
	00BA	YResolution Value	4800000001000000			
	00C2	DateTime Value	"1997:09:01 12:00:00"00			
	00D6	Copyright Value (Last Byte is Padded)	"Copyright,ABCDE,1997"0000			
Exif IFD	Offset (Hex)	Name	Interoperability Number (Hex)			
	00EC	Exif IFD Number	1900			
	Offset (Hex)	Name	Tag ID (Hex)	Type (Hex)	Count (Hex)	ValueOffset (Hex or "ASCII")
	00EE	ExposureTime	9A82	0500	01000000	1E020000
	00FA	FNumber	9D82	0500	01000000	26020000
	0106	ExifVersion	0090	0700	04000000	"0200"
	0112	DateTimeOriginal	0390	0200	14000000	2E020000
	011E	DateTimeDigitized	0490	0200	14000000	42020000
	012A	ComponentsConfiguration	0191	0700	04000000	01020300
	0136	CompressedBitsPerPixel	0292	0500	01000000	56020000
	0142	ShutterSpeedValue	0192	0A00	01000000	5E020000
	014E	ApertureValue	0292	0500	01000000	66020000
	015A	BrightnessValue	0392	0A00	01000000	6E020000
	0166	ExposureBiasValue	0492	0A00	01000000	76020000
	0172	MaxApertureRatioValue	0592	0500	01000000	7E020000
	017E	SubjectDistance	0692	0500	01000000	86020000
	018A	MeteringMode	0792	0300	01000000	01000000

	0196	LightSource	0892	0300	01000000	01000000	
	01A2	Flash	0992	0300	01000000	00000000	
	01AE	FocalLength	0A92	0500	01000000	8E020000	
	01BA	UserComments	8692	0700	2E000000	96020000	
	01C6	SubSecTime	9092	0200	04000000	30303000	
	01D2	SubSecTimeOriginal	9192	0200	04000000	"000"00	
	01DE	SubSecTimeDigitized	9292	0200	04000000	"000"00	
	01EA	FlashPixVersion	A000	0700	04000000	"0100"	
	01F6	ColorSpace	A001	0300	01000000	01000000	
	0202	Pixel X Dimension	A002	0400	01000000	6C020000	
	020E	Pixel Y Dimension	A003	0400	01000000	DB010000	
	Offset (Hex)	Name	Offset (Hex)				
	021A	Next IFD Offset	00000000				
Value longer than 4 Bytes of Exif IFD	Offset (Hex)	Name	Data (Hex or "ASCII")				
	021E	ExposureTime Value	010000003C000000				
	0226	FNumber Value	0400000001000000				
	022E	DateTimeOriginal Value	"1997:09:01 12:00:00"00				
	0242	DateTimeDigitized Value	"1997:09:01 12:00:00"00				
	0256	CompressedBitsPerPixel Value	0200000001000000				
	025E	ShutterSpeedValue Value	0600000001000000				
	0266	ApertureValue Value	0400000001000000				
	026E	BrightnessValue Value	0000000001000000				
	0276	ExposureBiasValue Value	0000000001000000				
	027E	MaxApertureRatioValue Value	0100000001000000				
	0286	SubjectDistance Value	0F0000000A000000				
	028E	FocalLength Value	3200000001000000				
0296	UserComment Value (46 Bytes)	4A49530000000000....					
1st IFD	Offset (Hex)	Name	Interoperability Number (Hex)				
	02C4	Number Of Interoperability	0F00				
	Offset (Hex)	Name	Tag ID (Hex)	Type (Hex)	Count (Hex)	ValueOffset (Hex or "ASCII")	
	02C6	ImageWidth	0001	0400	01000000	50000000	
	02D2	ImageLength	0101	0400	01000000	3C000000	
	02DE	BitsPerSample	0201	0300	03000000	7E030000	
	02EA	Compression	0301	0300	01000000	01000000	
	02F6	PhotometricInterpretation	0601	0300	01000000	06000000	
	0302	StripOffsets	1101	0400	01000000	94030000	
	030E	SamplesPerPixel	1501	0300	01000000	03000000	
	031A	RowsPerStrip	1601	0400	01000000	3C000000	
	0326	StripByteCounts	1701	0400	01000000	201C0000	
	0332	XResolution	1A01	0500	01000000	84030000	
	033E	YResolution	1B01	0500	01000000	8C030000	
	034A	PlanarConfiguration	1C01	0300	01000000	01000000	
	0356	ResolutionUnit	2801	0300	01000000	02000000	
	0362	YCbCrSubSampling	1202	0300	02000000	02000200	
	036E	YCbCrPositioning	1302	0300	01000000	02000000	
		Offset (Hex)	Name	Offset (Hex)			
		037A	Next IFD Offset	00000000			
Value longer than	Offset (Hex)	Name	Data (Hex or "ASCII")				

4byte of 1st IFD	037E	BitsPerSample Value	080008000800
	0384	XResolution Value	4800000001000000
	038C	YResolution Value	4800000001000000
Thumbnail Image Data	Offset (Hex)	Name	Data (Hex or "ASCII")
	0394	Thumbnail Image Data Strip	:

4.3 - Le choix de la bibliothèque graphique

Après avoir étudié et testé plusieurs bibliothèques permettant de traiter des images enregistrées sous le format JPEG, mon choix s'est finalement arrêté sur celle proposée par (IJG) « Independent JPEG Group ». (www.iijg.org).

4.3.1 - Pourquoi la bibliothèque du groupe indépendant JPEG.

Après une semaine passée dans les documentations, j'ai testé trois bibliothèques.

Mes critères de choix étaient les suivants :

- La bibliothèque doit être complète et avoir fait ses preuves, pas de « version bêta ».
- Elle doit pouvoir être utilisable facilement, ce qui implique que la documentation existe, qu'elle soit disponible gratuitement et qu'elle soit claire.
- Une intégration de cette bibliothèque simple dans le logiciel « Labwindows » que j'utilise. Donc, l'existence de fichiers source de la bibliothèque compilable en langage C.

Les sources des bibliothèques que fournissait IJG ont tout particulièrement attiré mon attention, premièrement car elles étaient en Langage C et qu'elles offraient une abondante documentation très abordable. Mais aussi car elles sont utilisées de façons simplifiées par d'autres logiciels de programmation et notamment IDL qui est un logiciel que je connais bien car je l'ai déjà utilisé dans le passé.

De plus c'est avec IDL que mon tuteur a effectué les premiers tests et mesures sur les images compressées en JPEG dans le cadre de notre projet, car contrairement à Labwindows, ce logiciel possède dans son installation de base des fonctions toutes faites pour le traitement d'images JPEG. Ces fonctions de base étant certaines fonctions de la librairie de IJG.

Comment se procurer les sources de IJG.

Aller sur leur site :

<http://www.iijg.org/>

Pour tout ce qui concerne la documentation de cette bibliothèque, et pour accéder au fichier « .c » et « .h » ainsi qu'aux exemples, il convient d'enregistrer le fichier compressé disponible par le lien :

<ftp://ftp.simtel.net/pub/simtelnet/msdos/graphics/jpegsr6.zip>.

Cette bibliothèque est fournie sous forme de fichiers « .c » et d'autres « .h »

Pour joindre cette bibliothèque à mes programmes sous « Labwindows » j'ai opté pour le regroupement des fichiers en une seule librairie, et donc de créer un fichier de type « .lib ».

De cette façon je peux joindre uniquement les parties de la bibliothèque qui m'intéressent et non la totalité. De ce fait l'utilisateur n'a plus qu'à joindre un nombre limité de fichiers à son projet au lieu d'une multitude de fichiers sources.

La librairie que fournit IJG est très bien faite et dispose de plusieurs fichiers « makefile » pour créer automatiquement les fichiers « .lib » compatibles avec les compilateurs les plus courants, mais malheureusement aucune n'est utilisable directement avec le logiciel « Labwindows ».

Je me suis donc inspiré d'un fichier « makefile » pour connaître les fichiers à regrouper.

La création de la librairie statique fut ensuite réalisée grâce aux fonctionnalités de Labwindows (cette étape est décrite ci-après).

4.3.2 - Les étapes de la construction d'une librairie statique avec « Labwindows »

Créez un nouveau projet.

Le nom du projet doit être significatif car la librairie statique qui va être créée portera le même nom avec une extension « .lib ».

Spécifiez que vous voulez réaliser une librairie statique.

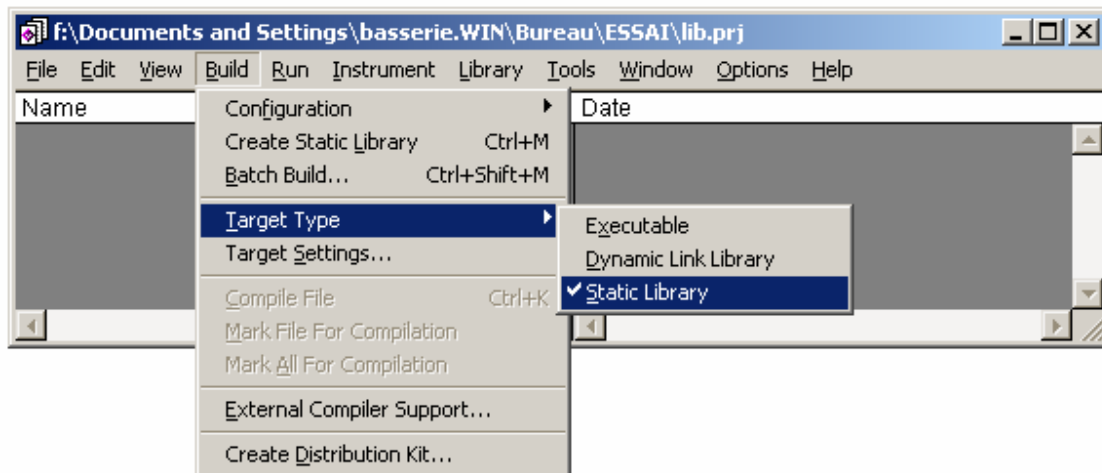


Figure 11 : sélection Static Library

Commande (figure 11) : Build ->Target Type->Static Library.

Importez-y les fichiers « .h » et « .c » que vous souhaitez regrouper en librairie.

Pour connaître la liste des fichiers j'ai utilisé le fichier « makefile.ansi » car la norme « ansi » est la plus standard.

Partie du fichier : « makefile.ansi »

source files: JPEG library proper

```
LIBSOURCES= jcapimin.c jcapistd.c jcoefct.c jccolor.c jcdctmgr.c jchuff.c \
  jcinit.c jmainct.c jmarker.c jmaster.c jcomapi.c jcparam.c \
  jcp Huff.c jprepct.c jcsample.c jctrans.c jdapimin.c jdapistd.c \
  jdatadst.c jdatasrc.c jdcoefct.c jdcolor.c jddctmgr.c jd Huff.c \
  jdinput.c jdmainct.c jdmarker.c jdmaster.c jdmerge.c jdphuff.c \
  jdpostct.c jdsample.c jdtrans.c jerror.c jfdctflt.c jfdctfst.c \
  jfdctint.c jidctflt.c jidctfst.c jidctint.c jidctred.c jquant1.c \
  jquant2.c jutils.c jmemmgr.c
```

memmgr back ends: compile only one of these into a working library

```
SYSDEPSOURCES= jmemansi.c jmemname.c jmemnobs.c jmemdos.c jmemmac.c
```

source files: cjpeg/djpeg/jpegtran applications, also rdjpgcom/wrjpgcom

```
APPSOURCES= cjpeg.c djpeg.c jpegtran.c rdjpgcom.c wrjpgcom.c cdjpeg.c \
  rdcolmap.c rdswitch.c transupp.c rdppm.c wrppm.c rdgif.c wrgif.c \
```

```

rdtarga.c wrtarga.c rdbmp.c wrbmp.c rdrle.c wrtle.c
SOURCES= $(LIBSOURCES) $(SYSDEPSOURCES) $(APPSOURCES)
# files included by source files
INCLUDES= jchuff.h jdchuff.h jdct.h jerror.h jinclude.h jmehsys.h jmorecfg.h \
    jpegint.h jpeglib.h jversion.h cdjpeg.h cderror.h transupp.h

```

J'ai donc importé tous les fichiers « .c » listés après « LIBSOURCES ».

Dans la liste de « SYSDEPSOURCES » j'ai choisi de ne garder que « jmemansi.c » car je sais que Labwindows utilise la norme « ansi ».

Pour finir les fichiers « .h » ajoutés au projet pour la réalisation de la librairie sont ceux listés dans la partie « INCLUDES » soit tous les fichiers de ce type fournis par IJG, car les fichiers « .h » ne servent qu'à l'édition des liens entre les différents fichiers sources « .c » et ne jouent donc pas sur la taille finale de la librairie statique.

Les fichiers listés par : « APPSOURCES » ne sont surtout pas à joindre car ce sont des fichiers d'exemples d'architecture de programme et contiennent tous un « main ».

Par exemple le fichier « djpeg.c » détaille la marche à suivre pour réaliser une décompression d'image JPEG.

L'exemple ci-dessous (figure 12) montre la méthode à suivre pour ajouter au projet les fichiers « .h » mais la démarche est la même pour les fichiers « .c ».

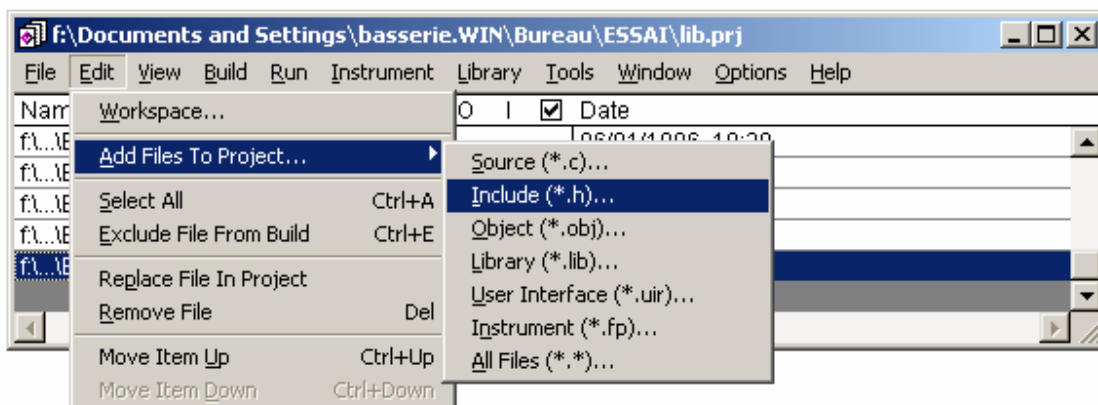


Figure 12 : inclusion fichier .h

La fenêtre suivante (figure 13) apparaît alors, il ne reste plus qu'à sélectionner les fichiers.

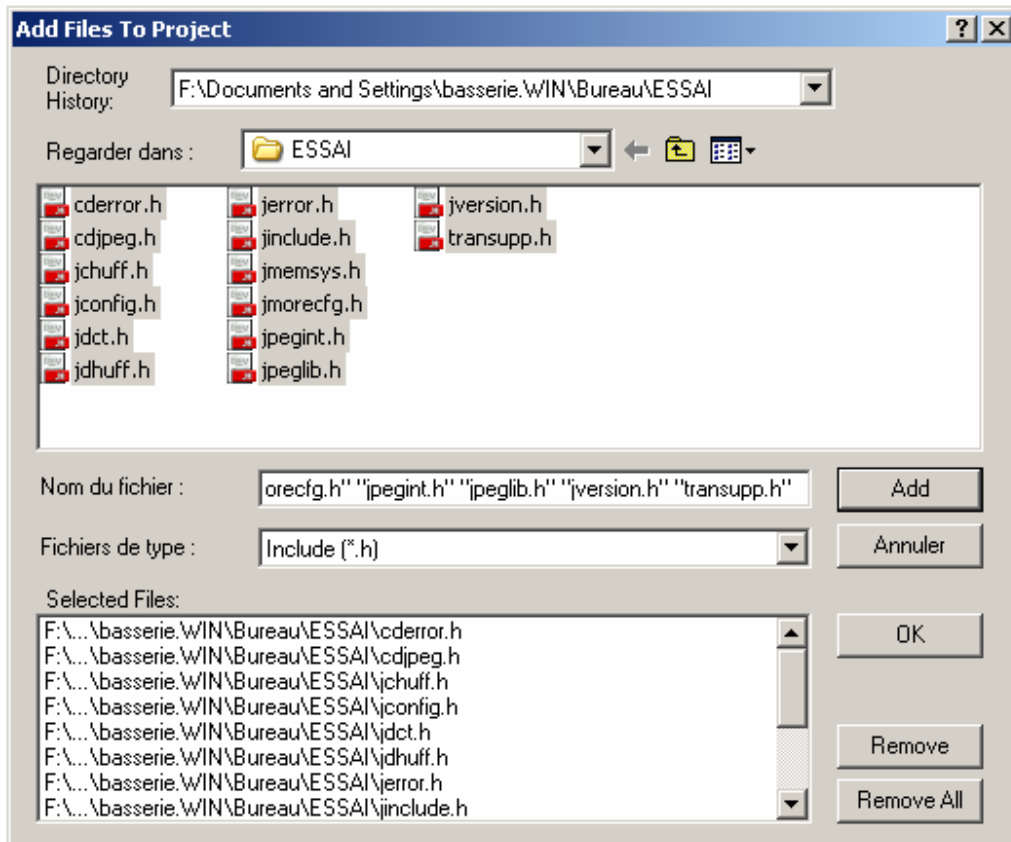


Figure 13 : ajout fichier .h

On peut maintenant réaliser la bibliothèque statique. Pour ce fait on vérifie que l'on est bien en mode « Release » (figure 14).

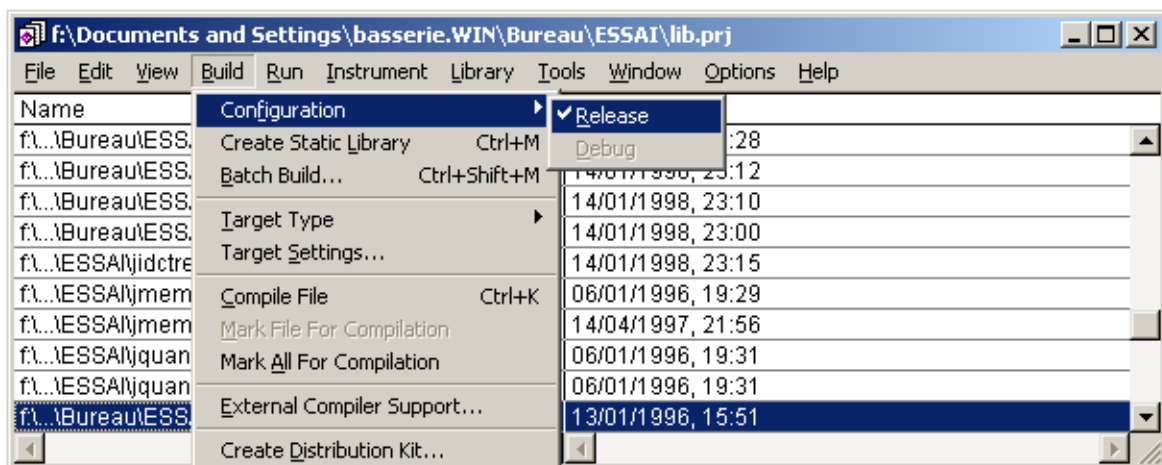


Figure 14 : Configuration Release

Ensuite il ne reste plus qu'à donner l'ordre de création (figure 15).

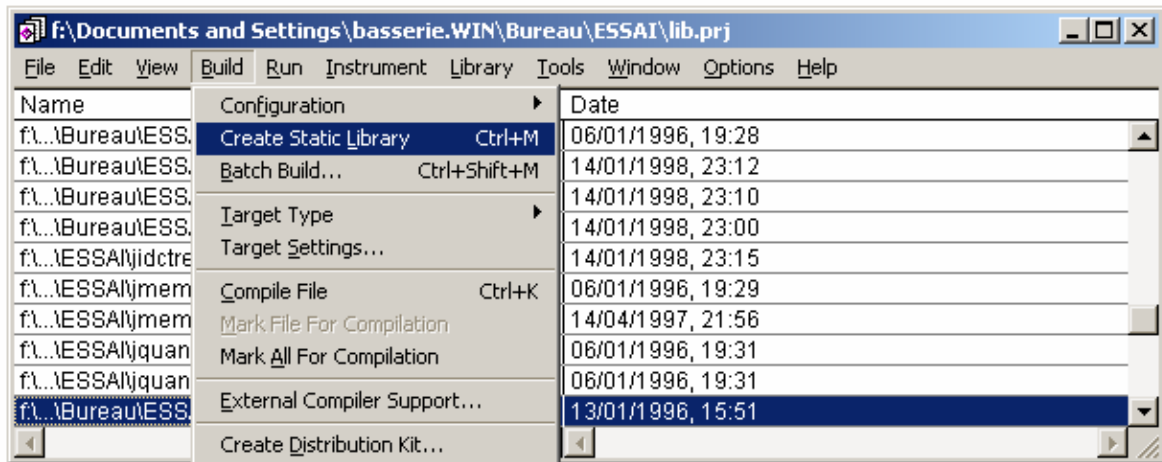


Figure 15 : création librairie

Vous venez de créer votre bibliothèque (figure16), elle se trouve dans votre répertoire de travail et porte le nom de votre projet suivi de l'extension « .lib » ainsi que son fichier « .h ».

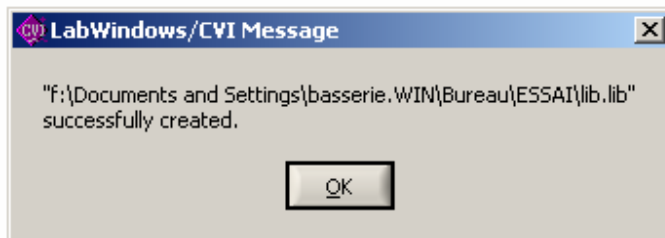


Figure 16 : message création

Pour utiliser cette librairie il vous suffit d'ajouter dans votre futur projet utilisant des fonctions pour le traitement d'image JPEG, le fichier portant le nom de votre projet « Labwindows » de création de bibliothèque ayant l'extension « .lib ».

Plus les fichiers « .h » intitulés :

- jpeglib.h
- jconfig.h
- jmorecfg.h

Ces fichiers contiennent les prototypes des différentes fonctions de la librairie JPEG.

4.3.3 - Améliorations

Pour effectuer les tests sur les photos prises par notre appareil et prendre en main cette bibliothèque de décompression JPEG, j'ai commencé par travailler à partir de fichiers (.JPEG). Puis, au moment du traitement direct des photos avec l'appareil relié au PC, une simplification de la procédure s'est imposée : il n'est plus utile d'ouvrir un fichier qui est une instruction gourmande en temps. En effet l'information sur la photo est envoyée sur la mémoire sous forme de « buffer » dont l'adresse est connue.

Le problème est que la bibliothèque par défaut ne prend pas en compte autre chose que des fichiers en entrée. J'ai dû réaliser des modifications et créer de nouvelles fonctions dans cette bibliothèque pour pouvoir lire des buffers en entrée.

Je me suis donc inspiré de la fonction existante pour lire les fichiers et charger les informations en mémoire, pour en créer une autre identique mais en travaillant directement à partir d'une adresse mémoire et non plus d'une adresse de fichier.

Evidemment une fois ces modifications réalisées, un fichier « .lib » prenant en compte les améliorations fut construit en suivant la même procédure que celle décrite précédemment.

4.3.4 - Détail des modifications dans la librairie

Les fichiers sources de la bibliothèque qui ont été modifiés sont :

- jpeglib.h ; dans lequel j'ai rajouté la déclaration de la nouvelle fonction créée.

La fonction existante est :

```
EXTERN(void) jpeg_stdio_src JPP((j_decompress_ptr cinfo, FILE * infile));
```

Elle a besoin du pointeur de la structure recevant les données et d'une adresse de type fichier.

La nouvelle est :

```
EXTERN(void) jpeg_stdio_src2 JPP((j_decompress_ptr cinfo, unsigned char *pbytes_tab, size_t imageSize));
```

Elle nécessite toujours le pointeur de la structure recevant les données. Il lui faut aussi une adresse mais cette fois c'est directement l'adresse du buffer en mémoire contenant l'image compressée ; de plus il lui faut l'information sur la taille du buffer pointé.

- `jdatasrc.c` ; dans lequel le corps de la nouvelle fonction est implanté juste après celui de la fonction de départ. La fonction permet à partir d'un buffer contenant toutes les informations sur une image JPEG, de diriger les données nécessaires vers la structure « src » déterminant la source et sa compression.

De plus le corps de la fonction : `fill_input_buffer (j_decompress_ptr cinfo)` a lui aussi été modifié de façon à pouvoir décompresser l'image par paquet si elle vient d'un buffer.

4.3.5 - Les programmes réalisés

Grâce à cette bibliothèque, j'ai réalisé des programmes permettant notamment de :

- Récupérer la matrice de pixel couleur par couleur (RGB).
- Calculer la valeur moyenne des pixels ainsi que l'écart type, couleur par couleur.
- Faire des représentations des valeurs RGB (rouge, vert, bleu) avec possibilité de les comparer et de les moyenner en ligne ou en colonne, permettant de localiser des zones de pixels défaillants.
- Réaliser des améliorations assurant un contrôle et la création de fichiers de rapports des listes de fichier JPEG à traiter.
- Mise en œuvre de nouvelles fonctions dans la librairie.

4.4 - L'appareil photo



L'appareil photo que j'utilise dans le projet est le « Digital camera FinePix S3Pro » de chez FUJIFILM.

Il est composé d'un capteur de 12.3 millions de pixels.

Délivrant une très haute résolution d'image de 4256 x 2848 pixels, autorisant la capture de détails extrêmement fins.

Figure 17 : appareil photo

Il est composé d'un capteur de 12.3 millions de pixels. Délivrant une très haute résolution d'image de 4256 x 2848 pixels, autorisant la capture de détails extrêmement fins.

4.4.1 - Caractéristiques

Capteur :

Super CCD IV SR 23 x 15,5 mm

12,3 millions

Résolutions :

4256 x 2848 pixels (12,1 millions de pixels).

3024 x 2016 pixels (Résolution utilisée pour le projet).

2034 x 1536 pixels.

1440 x 960 pixels.

Temps d'exposition :

De 30 à 1/4000 sec,

Sensibilité :

Equivalent à 100, 200, 400, 800, 1600 ISO.

Le choix de cet appareil photo s'est fait de par ses caractéristiques mais aussi par la possibilité de pouvoir le piloter directement par ordinateur via un câble Firewire IEEE 1394 et la bibliothèque SDK.

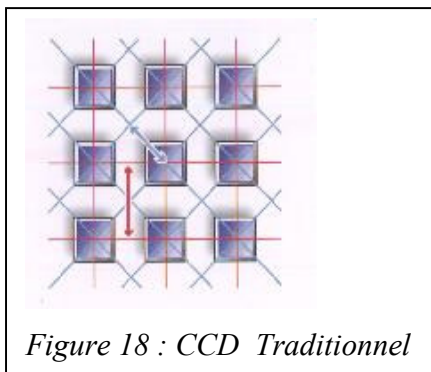
Cette dernière spécification n'est normalement pas disponible pour le public car les constructeurs ne souhaitent pas révéler leurs codes.

4.4.2 - Le super CCD (Charged Couple Device)

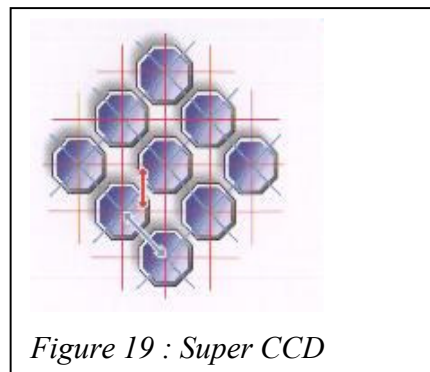
Les capteurs CCD traditionnels (figure 18) ont des photodiodes (ou photosites) carrées, le super CCD (figure 19) de FUJI a des photodiodes de forme octogonale et les pixels sont orientés à 45°. L'organisation de l'espace permet :

- De s'affranchir des limites technologiques actuelles puisque les pixels ont une surface plus élevée que sur un capteur traditionnel, donc d'obtenir une sensibilité plus élevée.
- D'obtenir des résolutions horizontales et verticales qui correspondent pratiquement aux caractéristiques de perception de l'oeil humain.

4.4.3 - L'implantation des CCD



Photodiodes rectangulaires



Photodiodes octogonales

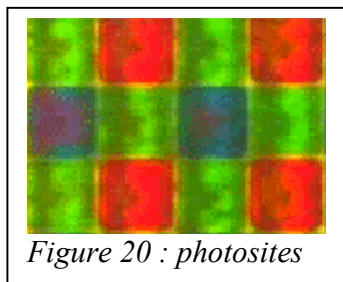


Figure 20 : photosites

Dans tous les cas, un pixel est composé de quatre photosites (figure 20) : 2 verts, 1 bleu et 1 rouge. Il y a plus de vert pour tenir compte du fait que l'oeil humain est plus sensible au vert qu'aux autres couleurs.

4.4.4 - L'interpolation

La grande différence entre les CCD traditionnels et le super CCD, se fait sur l'interpolation ; elle consiste à insérer un nouveau point (pixel) dans une image au moyen d'un calcul entre plusieurs points (pixels) existant déjà. En bref on augmente la définition (nombre de pixels) de l'image de façon artificielle puisque les points sont fabriqués par un procédé logiciel. Ce procédé dénature l'image d'origine en terme de qualité. Il peut augmenter le défaut d'une image en le rendant plus visible.

4.4.5 - Fonctionnement du super CCD

Du photosite octogonal, on détermine un carré au centre du photosite qui va constituer un pixel (figure 21). Les surfaces restantes (au nombre de quatre) sont rassemblées pour constituer un autre pixel. Il y a effectivement un calcul, d'où interpolation, mais cette dernière se fait d'après des éléments existants déjà physiquement



Figure 21 : constitution d'un pixel

Pour information, l'interpolation traditionnelle consiste à fabriquer des pixels au moyen d'un logiciel et d'après les données de couleurs, luminosité ... Partant de deux pixels le logiciel va en intercaler un nouveau qui sera la résultante de deux ou plusieurs. Dans ce cas là, c'est véritablement artificiel, car le pixel n'était pas présent physiquement.

4.4.6 - Les objectifs utilisés

Les deux objectifs utilisés sont de type 8mm « fish eye » avec un angle de champs de 180° pour voir le plus loin possible. De plus ils sont utilisables avec des reflex numériques.



4.5 - Le GPS

Dans mon projet j'utilise un GPS (Global positioning system) développé pour les systèmes mobiles, le « Lassen LP GPS » de la société « Trimble ».



Figure 23 : GPS

Il est composé d'une carte de communication et d'une antenne.

Ses caractéristiques principales sont :

- Consommation : 180mW 3,6V.
- Protocoles : TSIP, TAIP et NMEA 0183.
- Fréquence de communication : 1Hz.
- DGPS (précision du positionnement ramené à 2 mètres).
- Température de fonctionnement : - 40°C à + 85°C.

Le choix du GPS devait prendre en compte :

- les fortes contraintes climatiques pour une exploitation aussi bien au niveau des pôles que dans un désert (température et taux d'humidité acceptés en fonctionnement).
- La consommation car il sera alimenté par batterie.
- Sa précision de positionnement.

Ce dernier point est amélioré dans ce modèle grâce au mode DGPS (Differential Global Positioning System) dont la technique consiste en la correction transmise par une station fixe dont la position est connue.

4.5.1 - Détail de fonctionnement du DGPS

La station fixe compare sa position donnée par GPS avec sa position précise et calcule l'erreur de mesure de position que lui transmettent les satellites. Les données constituant l'erreur de position sont ensuite transmises selon le protocole industriel standard de communication RTCM SC-104 aux récepteurs GPS avoisinants. Le support de la transmission est de type FM, UHF ou encore VHF.

L'utilisation du système DGPS permet de réduire l'erreur de position en dessous de 5 mètres dans 95% du temps.

Sa précision reste cependant très dépendante de la qualité, de l'âge de la correction différentielle et de la proximité du récepteur avec le site de référence.

4.5.2 - Schéma DGPS

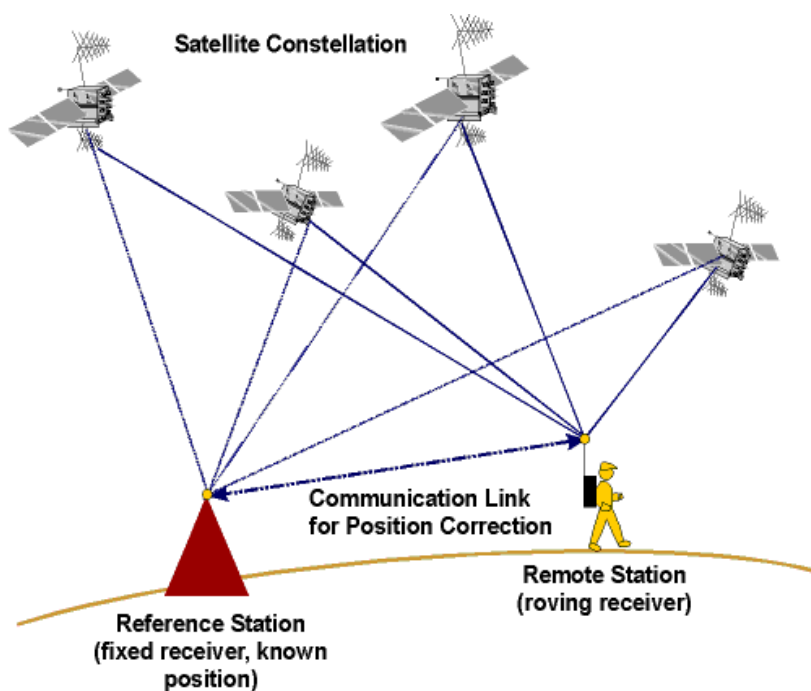


Figure 24 : DGPS

4.5.3 - L'utilisation du GPS dans le projet

Le GPS nous permet de connaître la latitude, la longitude et l'altitude de notre système, ainsi que l'heure exacte de l'exécution de nos applications.

Pour récupérer l'heure exacte par rapport à celle de la carte-mère et du processeur, j'ai réalisé un petit programme.

Les données de départ sont :

- La possibilité de connaître l'heure du PC à la milliseconde près ainsi que celle du GPS.
- Lors d'un front d'horloge (tous les 1hz), le GPS donne l'heure exacte.

Nous avons décidé de prendre comme référence l'heure donnée par le PC et de quantifier son décalage en fonction de l'heure donnée par le GPS. Pour ce faire, je compare l'heure de la carte mère du PC que je décompte en millisecondes grâce à un « timer » et le moment d'un front d'horloge du GPS ; je calcule alors l'erreur de temps entre la carte et le GPS.

Après des tests, l'erreur sur l'heure exacte par comparaison avec les fronts d'horloge du GPS est donc ramenée de 1s à 20ms, durée qui s'explique par le temps nécessaire au traitement des données et à la synchronisation du « timer ».

4.5.4 - La communication avec le GPS

La liaison entre le PC et le GPS se fait via le « port série » du PC. Le mode utilisé dans le GPS est TSIP (Trimble Standard Interface Protocol), ce qui est intéressant car l'appareil photo lui se connecte par le « firewire ». De ce fait nous sommes sûrs qu'il n'y a pas de conflit possible entre les différents appareils connectés car ils utilisent des protocoles de communication différents.

Pour la communication avec le GPS il a fallu tout d'abord isoler les commandes utiles. Les données reçues par le GPS sont retransmises selon un cycle défini dont la plupart données n'ont pas un grand intérêt dans notre application.

Au final, un paquet seul d'informations m'est utile :

Le 0x4A, qui a un format de 20 octets.

Il donne la position du GPS en latitude, longitude, altitude et la date donnée avec la précision de la seconde.

Table du paquet 0x4A

Octet	Détail	Type	Unité
0-3	Latitude	Simple	Radiant ; + pour le nord, - pour le sud
4-7	Longitude	Simple	Radiant ; + pour l'est, - pour l'ouest
8-11	Altitude	Simple	Mètre
12-15	Décentrer	Simple	Mètre
16-19	Temps	Simple	Secondes

Les données que je récupère via le GPS sont des « réels » codés sur 32 bits (en simple précision) et leur codage est quelque peu différent de celui des entiers. Elles sont représentées par la mantisse du nombre mais également son exposant.

Une partie du mot est utilisée pour représenter l'a puissance de 2 dont il faut affecter le nombre.

Le reste des bits du mot étant réservé aux chiffres significatifs.

Soit, sous forme dite normalisée, tout nombre réel peut s'écrire:

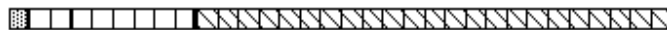
$$\pm 0.xxxxxx b^{+/-yyy}$$

Où xxxxx est la mantisse, b la base, et yyy l'exposant.

Le codage sur 32 bits d'un réel (en simple précision) est donc :

signe	exposant	mantisse
1 bit	8 bits	23 bits

représentation de la trame



L'exposant peut être compris entre 0 et 255, la mantisse entre 22 zéros suivis d'un 1 et 23 bits à 1, ce qui donne pour le plus grand nombre représentable dans cette notation, la valeur $3.4 \cdot 10^{38}$ et pour le plus petit positif $1.4 \cdot 10^{-45}$. Le nombre de chiffres significatifs est alors de 7 à 8.

Dans ces types de codage, le nombre vaut 0 si l'exposant et la mantisse sont tous les deux égaux à 0.

Pour information, un codage sur 64 bits (1 bit de signe, 11 bits pour l'exposant et 52 bits pour la mantisse) donne un domaine de variation de $1.7 \cdot 10^{308}$ à $5.0 \cdot 10^{-324}$, le nombre de chiffres significatifs étant alors de 15 à 16. Le codage en mode "extended" sur 80 bits, correspond à un domaine de variation qui va de $1.1 \cdot 10^{4932}$ à $1.9 \cdot 10^{-4951}$, soit de 19 à 20 chiffres significatifs.

4.6 - Le port parallèle

Dans le projet nous souhaitions mettre au point un moyen de communication pour recevoir une information venant d'un système autre que le GPS ou l'appareil photo et pourquoi pas aussi avoir la possibilité de commander autre chose mais de façon simple, sans protocoles de communication complexe.

C'est dans ce but et pour une mission d'observation des éclairs au Niger où nous avons testé « Plein ciel », que j'ai étudié les capacités offertes par le port parallèle.

Le port parallèle a finalement été utilisé pour cette mission pour envoyer un signal « start » vers un afficheur LCD indiquant un temps en secondes.

Lors des mesures en Afrique l'afficheur LCD était activé au moment des prises de vue d'éclairs. Le contenu de l'afficheur était alors visible lorsqu'un éclair était présent ce qui nous donnait l'indication sur le moment où il avait illuminé le ciel.

4.6.1 - Définition

Le port parallèle fut créé par IBM et se présente sous la forme d'un connecteur de type : 1284-A, qui porte aussi le nom : DB25 femelle (figure 25). Historiquement son but était de communiquer avec une imprimante dont la communication des données était unidirectionnelle et vers l'imprimante. De nos jours ce port peut recevoir des périphériques plus évolués et échanger des données dans les deux sens. Ces différents modes de communication sont définis dans la norme IEEE 1284 réalisée en 1994.

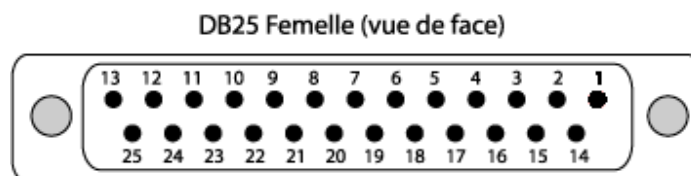


Figure 25 : prise port parallèle

Au fil des années, le port parallèle a subi des améliorations, et donc des changements de spécifications.

On en distingue actuellement trois déclinaisons, du plus ancien au plus récent :

Le port parallèle d'origine, appelé "SPP" (Simple Parallel Port) unidirectionnel.

Le port "EPP" (Extended Parallel Port) bidirectionnel.

Le port "ECP" (Extended Capability Port) possibilité de Plug and Play.

Les PC actuels offrent toutes ces possibilités sur un même port. On parle donc maintenant de "mode": mode SPP, mode EPP ou encore mode ECP.

4.6.2 - Communication sur le port parallèle

Comme son nom l'indique, le port parallèle permet d'envoyer des données simultanément sur plusieurs canaux. Dans le cas de ceux des PC on peut envoyer 8 bits (ou un octet) de façon simultanée (figure 26).

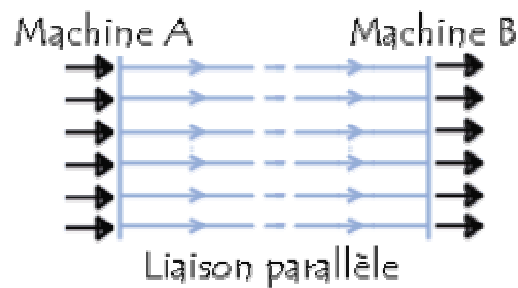


Figure 26 : schéma de liaison parallèle

Par définition un signal allant du PC vers un appareil périphérique est dit sortant et dans le cas contraire il est défini comme entrant.

Les niveaux de tension en sortie sont 0/5 Volts. La technologie est de type TTL (Transistor Transistor Logic) : technologie électronique à base de transistors employés comme interrupteurs de courant ou circuits commutateurs.

En tant que circuits commutateurs, les transistors délivrent de 2.5 à 5 volts pour indiquer une condition vraie (TRUE) ou de 0 à 0.7 volts pour indiquer une condition fausse (FALSE).

Tableau des différentes fonctions du port parallèle selon le mode considéré (Table 1).

Table 1. The IEEE 1284 Signal Line Descriptions

SPP Signal Name	EPP Signal Name	ECP Signal Name	Source	Connector Pinout
-----------------	-----------------	-----------------	--------	------------------

Data8-1 Unidirectional data lines. Data8 is the most significant.	AD8-1 Bi-directional address and data lines. AD8 is the most significant.	Data8-1 Bi-directional address and data lines. Data8 is the most significant.	Host/ Peripheral	1284-A: 9 - 2 1284-B: 9 - 2 1284-C: 13 - 6
STROBE* Data is valid during an active low pulse on this line.	WRITE* This signal is low during a write operation and high during a read operation.	HostClk This forward direction handshaking line is interlocked with PeriphAck and driven low when data is valid.	Host (*HI)	1284-A: 1 1284-B: 1 1284-C: 15
AUTOFD* Usage of this line varies. Most printers will perform a line feed after each carriage return when this line is low, and carriage returns only when this line is high.	DSTROBE* This signal denotes data cycles. During a write operation, data is valid when this signal is active. During a read operation, this signal is low when the host is ready to receive data.	HostAck In the forward direction, this line is driven low for a command transfer, and high for a data transfer. In the reverse direction, this signal is a handshaking line interlocked with PeriphClk.	Host (*HI)	1284-A: 14 1284-B: 14 1284-C: 17
INIT* This line is held low for a minimum of 50 μ s to reset the printer and clear the print buffer.	INIT* This line is driven low to terminate EPP mode and return to SPP mode.	ReverseRequest* This line is driven low to place the parallel port interface in the reverse direction.	Host	1284-A: 16 1284-B: 31 1284-C: 14
SelectIn* The host drives this line low to select the peripheral.	ASTROBE* This line denotes address cycles. When this signal is low, AD8-1 is an address.	1284 Active The host drives this line high while in ECP mode, and low to terminate ECP mode.	Host (*HI)	1284-A: 17 1284-B: 36 1284-C: 16
ACK* The peripheral pulses this line low when it has received the previous data and is ready to receive more data. The rising edge of ACK* can be enabled to interrupt the host.	INTR* The peripheral can enable this signal to interrupt the host on the low to high transition.	PeriphClk The peripheral drives this reverse direction handshaking line low to indicate that the data is valid. PeriphClk is interlocked with HostAck.	Peripheral	1284-A: 10 1284-B: 10 1284-C: 3
BUSY The peripheral drives this signal high to indicate that	WAIT* The peripheral drives this signal low to acknowledge	PeriphAck This forward direction handshaking line is interlocked with HostClk and driven by the	Peripheral	1284-A: 11 1284-B: 11 1284-C: 1

it is not ready to receive data.	that it has successfully completed the data or address transfer initiated by the host.	peripheral to acknowledge data received from the host. During reverse direction transfers, the peripheral drives this line high during data transfers and low during command transfers.	(*HI)	
PError Usage of this line varies. Printers typically drive this signal high during a paper empty condition.	User Defined	AckReverse* The peripheral drives this line to follow the level of the ReverseRequest* line.	Peripheral	1284-A: 12 1284-B: 12 1284-C: 5
Select The peripheral drives this signal high when it is selected and ready for data transfer.	User Defined	XFlag The peripheral drives this line high to indicate that it uses ECP mode.	Peripheral	1284-A: 13 1284-B: 13 1284-C: 2
FAULT* Usage of this line varies. Peripherals usually drive this line low when an error condition exists.	User Defined	PeriphRequest* The peripheral drives this signal low to request a reverse transfer. This line can be used to interrupt the host.	Peripheral	1284-A: 15 1284-B: 32 1284-C: 4

Attention dans la colonne « Source », les pattes 1, 11, 14, 17 (référence des types 1284-A) sont notées « (*HI) » pour Hardware Inverted ; en clair si vous avez un 1 logique dans la commande ou la lecture du port, la tension aux bornes de ces pattes sera « nulle » et inversement.

4.6.3 - Les adresses des ports parallèles

Aujourd'hui 99.9% des PC ne possèdent qu'un seul port parallèle physique (dont l'adresse est : 378 en hexa), mais avec des cartes on peut aller jusqu'à 3 et c'est pour cela qu'en général on trouve dans la littérature 3 adresses 378, 278 et 3BC. Dans la littérature les adresses sont parfois précédées de : LPT pour Line Printing Terminal (en référence aux imprimantes auxquelles était dédié ce port à l'origine).

Pour chacune de ces adresses on a accès à plusieurs registres, qui sont référencés en fonction d'un offset :

Offset	Adresse du registre correspondant
--------	-----------------------------------

0	378
1	379
2	37A Nous sommes en hexa !
3	37B
4	...

Le nombre de registres varie selon le mode dans lequel s'effectue la communication.

4.6.4 - Mode de fonctionnement

La communication devant être effectuée via le port parallèle, les besoins du projet m'imposent une possibilité de pouvoir communiquer dans les deux sens.

La nature de ces signaux est très importante pour le choix du mode de fonctionnement à utiliser pour le port parallèle. S'ils sont uniquement de type « indicatif » (mise à l'état haut ou bas d'un bit), ou alors s'ils sont du type « informatif » (envoi d'une valeur ou information codée sur plusieurs bits). De plus, l'information transite-t-elle dans les deux sens par les mêmes broches du port ?

Ce dernier point levé, il m'a été demandé d'étudier les différents cas.

Premier cas : « indicatif » mode SPP

Je m'intéresse ici à une transmission simple de type : mise dans un état voulu un bit du port par l'alimentation ou mise à la masse d'une broche. Une signalisation peut être effectuée dans les deux sens, mais pas sur les mêmes broches du port selon qu'elle s'effectue en entrée ou en sortie. Dans ce cas l'utilisation du mode SPP est tout à fait suffisante, son avantage est qu'il est très facile à mettre en œuvre car il ne demande pas de gestion particulière de la communication (état, acquittement...). Ce mode permet dans sa configuration de base d'avoir 8 broches pour la sortie et 5 broches pour l'entrée. On utilise ainsi les diverses entrées/sorties du port pour échanger des signaux.

Une possibilité nous permet même d'échanger des informations sur plusieurs broches en parallèle (8/5) mais cela implique un traitement (informatique en entrée et électronique en sortie) pour regrouper les broches considérées.

De plus il est possible de faire du « Nibble » (mot de 4 bits en anglais) en utilisant 4 des 5 broches d'entrée pour envoyer des mots de 8 bits en 2 temps. Dans ce dernier cas le périphérique doit pouvoir couper les octets avant de les envoyer et le temps de transfert est multiplié par deux.

Les 8 broches de sortie sont :

Numéro	Nom
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7

Elles sont accessibles par programmation (pour l'écriture) à l'adresse 0x378

Les 5 broches d'entrée sont :

Numéro	Nom
11	/BUSY (*HI, voir Table 1)
10	ACK
12	PError
13	Select
15	Fault

Elles sont accessibles par programmation (pour la lecture) à l'adresse 0x379

Le taux utile de transfert est de 10 kbits/s si on considère qu'on ne change qu'un bit à la fois pour le transmettre.

Les registres sont donnés par la table suivante (Table2) :

Table 2. SPP Registers

Table 2. SPP Registers	Register	Offset	7	6	5	4	3	2	1	0
Data Register	0	D7	D6	D5	D4	D3	D2	D1	D0	
Status Register	1	BUSY*	ACK*	PError	Select	FAULT*	IRQ*	Reserved	Reserved	
Control Register	2	Reserved	Reserved	Reserved	IRQEN	SelectIn	INIT*	AUTOFD	STROBE	

Deuxième cas : « informatif » mode EPP

Je m'intéresse ici à une transmission parallèle de type mise en transfert de données sur 8 bits par l'alimentation ou mise à la masse des broches. Une signalisation peut être effectuée dans les deux sens et sur les mêmes broches (half duplex) du port (D0 à D7). Dans ce cas l'utilisation du mode EPP est conseillée, son avantage est qu'il permet un taux utile de transfert allant jusqu'à 2 Moctests/s. Par contre il impose de mettre en œuvre un procédé de gestion de la communication (état, acquittement...).

La fonction des registres et leurs droits d'accès sont donnés ci-dessous (Table 3).

Table 3. EPP Mode Parallel Port Register Map

Register	Read or Write	Register Offset
Parallel Port Data Register	Write	0
Parallel Port Status Register	Read	1
Parallel Port Control Register	Read/Write	2
Auto Address Strobe Register	Read/Write	3
Auto Data Strobe Register	Read/Write	4
Auto Data Strobe Register	Read/Write	5
Auto Data Strobe Register	Read/Write	6
Auto Data Strobe Register	Read/Write	7

Cela semble faire beaucoup de ports à gérer mais en fait les trois premiers (378, 379, 37A) suffisent et sont souvent les seuls accessibles. Les concepteurs avaient prévu large au départ.

4.6.5 - Détail des 3 premiers ports

Table 4. EPP Registers of importance

Register	Offset	7	6	5	4	3	2	1	0
Data Register	0	D7	D6	D5	D4	D3	D2	D1	D0
Status Register	1	WAIT	INTR	User Defined	User Defined	User Defined	IRQ	Reserved	Reserved
Control Register	2	Reserved	Reserved	Reserved	IRQEN	ASTROBE	INIT	DSTROBE	WRITE

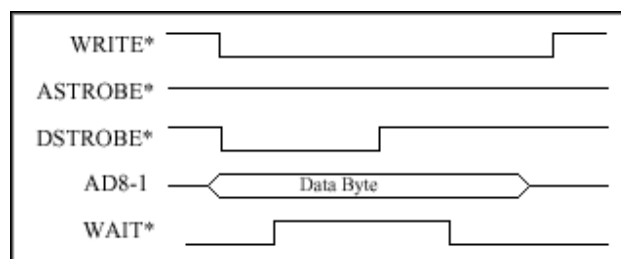
4.6.6 - L'écriture

La figure 1 indique le chronogramme d'un cycle d'écriture. Les états sont donnés directement pour l'implémentation dans les registres (les ports 378, 379 et 37A). Je rappelle que le niveau de tension en sortie est inversé pour les pattes 1, 11, 14 et 17 (WRITE, WAIT, DSTROBE et ASTROBE).

Au départ le port est dans la configuration décrite dans la partie « Le branchement » et la patte 11 (WAIT) commandée par le périphérique est à la masse (patte 25).

Pour écrire une donnée sur le port il faut mettre ASTROBE à 1 car on envoie des données et non une adresse. On impose ensuite 0 à WRITE, on place les données à transmettre sur AD8-1 et on met 0 à DSTROBE. Le périphérique déconnecte la patte 11 (WAIT) de la masse pour signaler qu'il est prêt à lire la donnée. On fait alors passer DSTROBE à 1 ; cet acte signale au périphérique qu'il récupère les données. Enfin lorsque le périphérique a lu les informations il reconnecte la patte 11 (WAIT) à la masse ; il signale ainsi qu'il est prêt à une nouvelle acquisition.

Figure 27: EPP Data Write Cycle



Avec les registres :

On va écrire 31 en décimale soit 11111 en binaire.

Les pattes 10, 12, 13 et 14 sont connectées en permanence à la masse (patte 25).

Le périphérique relie la patte 11(WAIT) à la masse (patte 25).

Le registre 379 qui ne peut qu'être lu affiche donc normalement : 1000 0000.

Mettre 0000 1110 dans le registre 37A (écrire une donnée en mode EPP).

Mettre 0001 1111 dans le registre 378 (soit 31 en décimale).

Mettre 0000 1100 dans le registre 37A (donnée prête transfert possible).

Le périphérique déconnecte alors la patte 11 (WAIT) de la masse (périphérique prêt à lire).

Mettre 0000 1110 dans le registre 37A (ordre de lecture de la donnée).

Le périphérique remet alors la patte 11 (WAIT) de la masse (donnée lue).

4.6.7 - La lecture

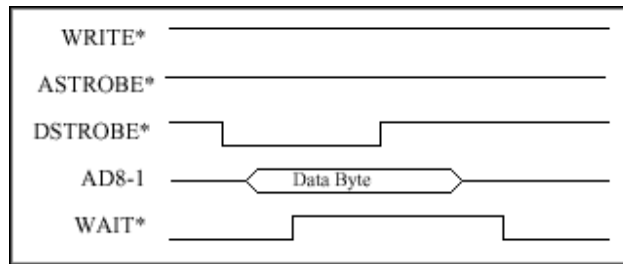
De la même façon, la figure 2 indique le chronogramme d'un cycle de lecture. Les états sont donnés directement pour l'implémentation dans les registres (les ports 378, 379 et 37A). Je rappelle que le niveau de tension en sortie est inversé pour les pattes 1, 11, 14 et 17 (WRITE, WAIT, DSTROBE et ASTROBE).

Là aussi, au départ le port est dans la configuration décrite dans la partie « Le branchement » et la patte 11 (WAIT) commandée par le périphérique est à la masse (patte 25).

Pour lire une donnée sur le port il faut mettre ASTROBE à 1 car on envoie des données et non une adresse on impose ensuite 1 à WRITE. A cet instant les lignes AD8-1 sont à l'état de haute impédance. On met alors DSTROBE à 0 pour signaler au périphérique qu'on est prêt à lire.

Le périphérique place alors ses données en mettant à 5 Volts les pattes correspondantes puis il déconnecte la patte 11 (WAIT) de la masse pour signaler qu'il est prêt à transmettre. L'utilisateur peut alors lire les données, quand il l'a fait il replace DSTROBE à 1. Le périphérique reconnecte finalement la patte 11 (WAIT) à la masse, on est prêt pour un nouveau cycle.

Figure 28 : EPP Data Read Cycle



Avec les registres :

On va écrire 31 en décimale soit 11111 en binaire.

Les pattes 10, 12, 13 et 14 sont connectées en permanence à la masse (patte 25).

Le périphérique relie la patte 11(WAIT) à la masse (patte 25).

Le registre 379 qui ne peut qu'être lu affiche donc normalement : 1000 0000.

Mettre 0000 1111 dans le registre 37A (lire une donnée en mode EPP).

L'utilisateur est prêt à lire les lignes D8-1 sont à l'état haute impédance.

Mettre 0000 1101 dans le registre 37A (le PC est prêt, transfert possible).

Le périphérique impose 5 Volts aux pattes 2, 3, 4, 5, 6 (soit 31 en décimale).

Le périphérique déconnecte alors la patte 11 (WAIT) de la masse (données prêtes).

Lire le registre 378.

Mettre 0000 1111 dans le registre 37A (lecture effectuée).

Le périphérique remet alors la patte 11 (WAIT) de la masse (fin du cycle).

Le principal inconvénient de ce mode est qu'un signal doit être reçu à la suite de chaque transfert de données et ce dans un laps de temps d'environ 10µs. Le bit qui indique l'acquittement est le bits 0 (Time-out bit) dans le registre « Status Port », il passe à 1 lorsque la broche « Wait » n'est pas revenue à la masse dans les 10µs qui suivent sa mise à l'état haute impédance.

Ce détail impose un traitement de l'information et de son acquisition de la part du périphérique, ce qui minimise l'utilisation de ce mode.

4.6.8 - Le choix

Dans le cadre du projet les signaux étant des « alertes » ou des « autorisations de mesures » et non des données, le mode SPP a donc été préféré du fait de la simplicité de sa mise en œuvre.

4.6.9 - Les tests

Le contrôle du port étant réalisé par le logiciel «Labwindows», ce dernier nous préconise l'utilisation d'un « driver » qui nous permet d'utiliser le port sans pouvoir l'endommager.

Son utilisation est simple, il suffit de le lancer au début de l'application pour que ce dernier vérifie nos actions sur le port.

La commande à mettre en œuvre est la suivante :

```
int = CVILowLevelSupportDriverLoaded();
```

Les fonctions pour lire et écrire sur le port sont :

```
byteRead = inp(portNumber+offset); // To read
```

```
byteWritten = outp(portNumber+offset, byteToWrite[0]); // To write
```

Les niveaux de tension :

Le niveau de tension fourni par le PC est de 43.7mV à l'état bas et 3.36V à l'état haut.

Durée réelle :

Pour les tests la sensibilité de l'oscilloscope est de 100Mhz.

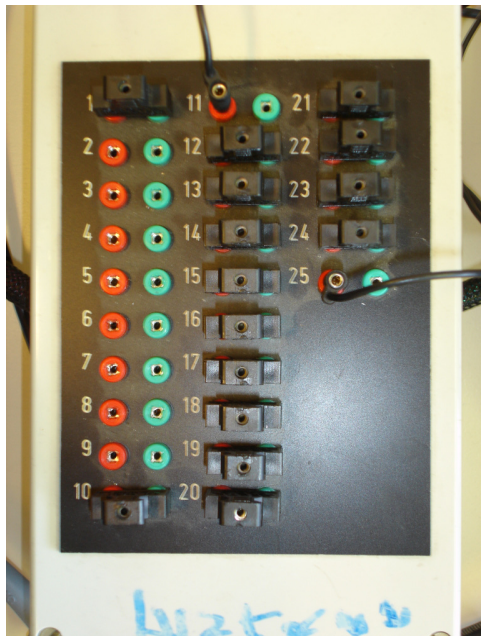
Pour les créneaux la durée minimale est de 2 ms contrairement au 1ms annoncé par CVI pour la fonction de contrôle du temps utilisée (la méthode « Sleep » de Labwindows).

Les signaux ont une précision maximale de +/- 2 ms selon les instructions réalisées à ce moment.

Montage réalisé pour les tests :



Figure 29 : Montage pour les tests



Je dispose d'une boîte d'éclatés du port parallèle (figure 30).

Comme spécifié ci-dessus les bornes sur lesquelles on vient lire les données sont celles numérotées de 2 à 9.

Figure 30 : éclaté du port parallèle

Les mesures effectuées nous donnent les résultats suivants :

Visualisation d'un train de quatre impulsions de 10ms espacées de 10ms.

La calibration de l'oscilloscope est :

- 500 mV / division
- 10 ms /division

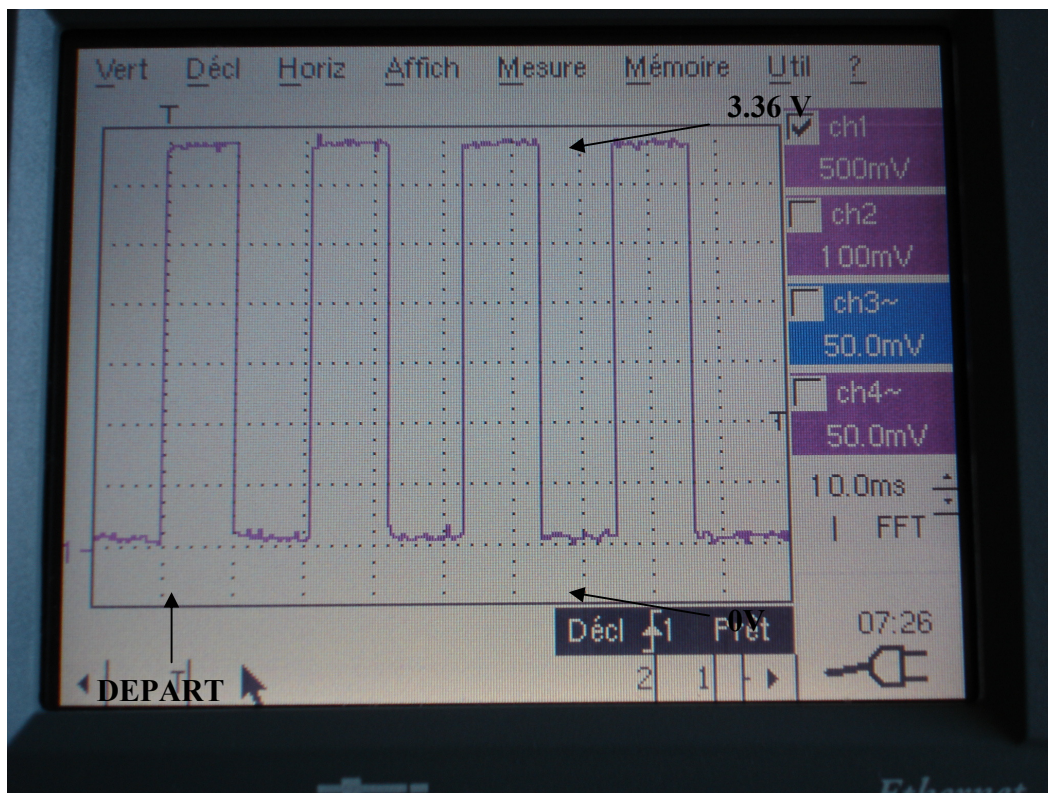


Figure 31 : visualisation créneau sur port parallèle

On constate sur la figure 31 que les durées à l'état haut et bas ne valent pas rigoureusement 10ms, de plus elles ne sont pas toujours constantes : le temps varie selon les applications lancées.

Néanmoins, ces résultats sont suffisamment précis pour réaliser leurs tâches, à savoir l'envoi d'un « start » pour commander des afficheurs CCD, ainsi que l'acquisition d'une alarme sur le contrôle de la température.

4.6.10 - Aperçu de l'interface réalisée

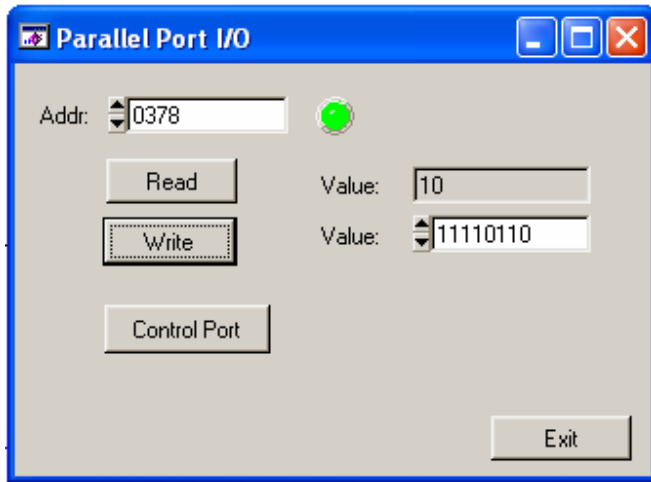


Figure 32 : Panel test port parallèle

Cet écran (figure 32) permet de choisir l'adresse sur laquelle l'utilisateur souhaite agir, puis de lire ou d'écrire des données sur le port parallèle.

J'ai aussi réalisé une protection pour interdire l'écriture sur les ports qui sont seulement prévus pour être lus.

4.7 - Les acquisitions sonores

4.7.1 - L'objectif

Dans le projet ALPHA il m'a été demandé de pouvoir réaliser une acquisition sonore en utilisant la carte « son » du PC.

Cette action n'est pas directement utilisée pour l'étude des aurores boréales, mais là aussi pour une application d'étude des éclairs menée en Afrique au cours de l'été pendant laquelle le prototype a subi des essais sur le terrain.

L'acquisition du son nous permet par la suite de calculer la distance d'un éclair par rapport à notre position et d'étudier les fréquences mises en jeu. Je me suis donc documenté sur la façon dont je peux construire un fichier contenant une information sur le son.

L'utilisation de l'appareil photo est simple : la chasse des éclairs avec notre dispositif s'effectue en laissant l'objectif ouvert sur une durée peu importante pour ne pas détériorer les capteurs CCD en les saturant en lumière. De plus un dispositif nous renvoie une information au moment où il y a un éclair. Sachant que le déclenchement de la prise de son commence en même temps que celle de l'image : la méthode de calcul de la distance est simple ; je compare la différence entre l'instant où l'information de la lumière de l'éclair est acquise et le moment où le grondement produit est reçu. Connaissant la vitesse de la lumière et celle du son, je déduis de mes mesures une durée et donc finalement, un rapport me donnant la distance.

Valeur de la vitesse des deux phénomènes :

- La vitesse de la lumière est de 299 792 458 m/s.
- la vitesse du son est de 340,29 m/s

Je cherche à récupérer par programmation la durée (Δt).

La distance (d) est donnée par :

$$d = \left(\frac{299079458}{340.29} \right) * \Delta t \quad d \text{ (m)}$$

Mes prérogatives sont :

- de pouvoir imposer une fréquence d'acquisition.
- Pour le moment, aucune compression des fichiers sonore n'est envisagée dans notre projet pour des questions de traitements supplémentaires qui ne sont pas une priorité, mais la question sera peut être envisagée par la suite.

Je me suis donc concentré sur les créations d'un fichier son « classique », reconnu et facile d'utilisation.

Du fait que je travaille sous « Windows » la solution du fichier « .wav » s'est imposée tout naturellement, car elle permet de choisir la fréquence d'acquisition de données.

4.7.2 - Le wave

Le nom vient de « wave from audio format » créé par Microsoft et IBM.

Ce n'est pas un format d'encodage spécifique, mais un conteneur capable de recevoir des données dans les formats MP3, WMA, ATRAC3, ADPCM et PCM.

Le plus répandu est le PCM qui est un format sans perte.

Les fichiers .wav sont souvent dits sans perte mais c'est un abus de langage car la fréquence d'acquisition est libre.

Pour explication les fréquences les plus couramment utilisées et donc conseillées sont :

- 11,025 kHz pour une qualité téléphone
- 22,05 kHz pour une qualité radio
- 44,1 kHz pour une qualité CD

Les fichiers wave sont constitués d'un en-tête qui a une taille constante de 44 octets

En-tête de fichier WAV

4 octets : Constante "RIFF"

4 octets : Taille du fichier moins 8 octets

14 octets : Constante "WAVEfmt"

2 octets : Mode (1 pour mono ou 2 pour stéréo)
4 octets : Fréquence d'échantillonnage (en Hertz)
4 octets : Nombre de bits par seconde de musique
2 octets : Nombre d'octets par échantillon
2 octets : Nombre de bits par donnée
4 octets : Constante "data"
4 octets : Taille du fichier moins 116 octets

Ensuite viennent les données sur le son qui sont acquises en mode PCM.

4.7.3 - Le PCM

Le PCM, ou Pulse Code Modulation, (Modulation d'impulsion codée), est une technique d'échantillonnage.

Elle permet la représentation numérique d'un signal analogique. La méthode consiste en la mesure de l'amplitude du signal de façon périodique (T).

Le théorème de Nyquist-Shannon pour la reconstruction d'un signal ($F_{ech} \geq 2 * F_{signal}$) nous montre que les fréquences supérieures à la moitié de la fréquence d'échantillonnage choisie ne seront pas prises en compte.

La fréquence d'échantillonnage influe donc énormément sur la qualité du son retranscrit dans le fichier résultat.

Il faut donc prendre en compte dès le départ que les fréquences élevées dues au phénomène atmosphérique que sont les éclairs ne seront pas observables.

4.7.4 - Les choix

Du fait de l'utilisation que j'aurai par la suite des données, un pas constant de données et surtout rond est souhaitable pour avoir une vision rapide des résultats.

De plus la qualité du son n'est pas ma priorité contrairement à la place sur le disque dur que prendront ces acquisitions de son ; plus la fréquence est élevée plus le fichier final est volumineux.

Je construis donc des fichiers avec une fréquence d'acquisition de 10 kHz pour avoir des données très facilement exploitables et en mono car je ne dispose que d'un capteur sonore.

Pour la réalisation de ces fichiers, j'ai incorporé à mon programme de traitement du son une bibliothèque du nom de « FMOD » dans sa version 3.75.

Cette dernière permet la création des fichiers donnés sur le son d'une manière très simple à aborder.

De plus elle était fortement conseillée par la communauté des programmeurs sous Labwindows.

4.7.5 - Les essais

J'ai réalisé des tests pour avoir une idée de la taille des fichiers que j'allais construire ainsi que les résultats que je pouvais avoir.

J'ai donc fait des mesures sur une durée de base de 10 secondes échantillonnées à 10hz en mono voie.

Les fichiers ont alors une taille réelle (pas sur le disque) de 195 Ko soit 200 044 octets.

Les caractéristiques des fichiers « wave » (figure 33) sont alors :

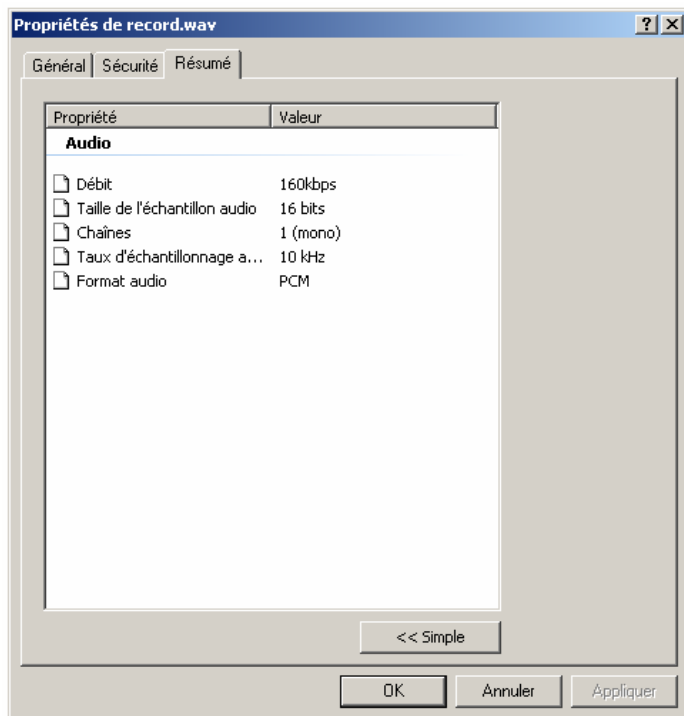


Figure 33 : propriétés fichier .wav

Actuellement le stockage des données est effectué sur 16 bits (PCM 16), l'information souhaitée étant juste l'acquisition d'une présence sonore (celle du tonnerre).

La qualité des données qui dépend aussi du nombre de bits sur lesquels ont va la stocker pourrait être revue à la baisse (8 bits).

J'ai donc effectué des tests d'acquisition sonores et des traitements de ces dernières lors d'un violent orage dont l'un des éclairs est tombé sur le paratonnerre de l'observatoire de St Maur.

C'est la capture de cet éclair que nous allons voir ci-après, il est référencé sous le nom de « recording3.wav ». Il a été enregistré grâce à un petit programme que j'ai réalisé sous Labwindows à l'aide de la librairie « FMOD ».

Un programme me permet par la suite d'afficher le contenu des fichiers « .wav » (figure 34) :

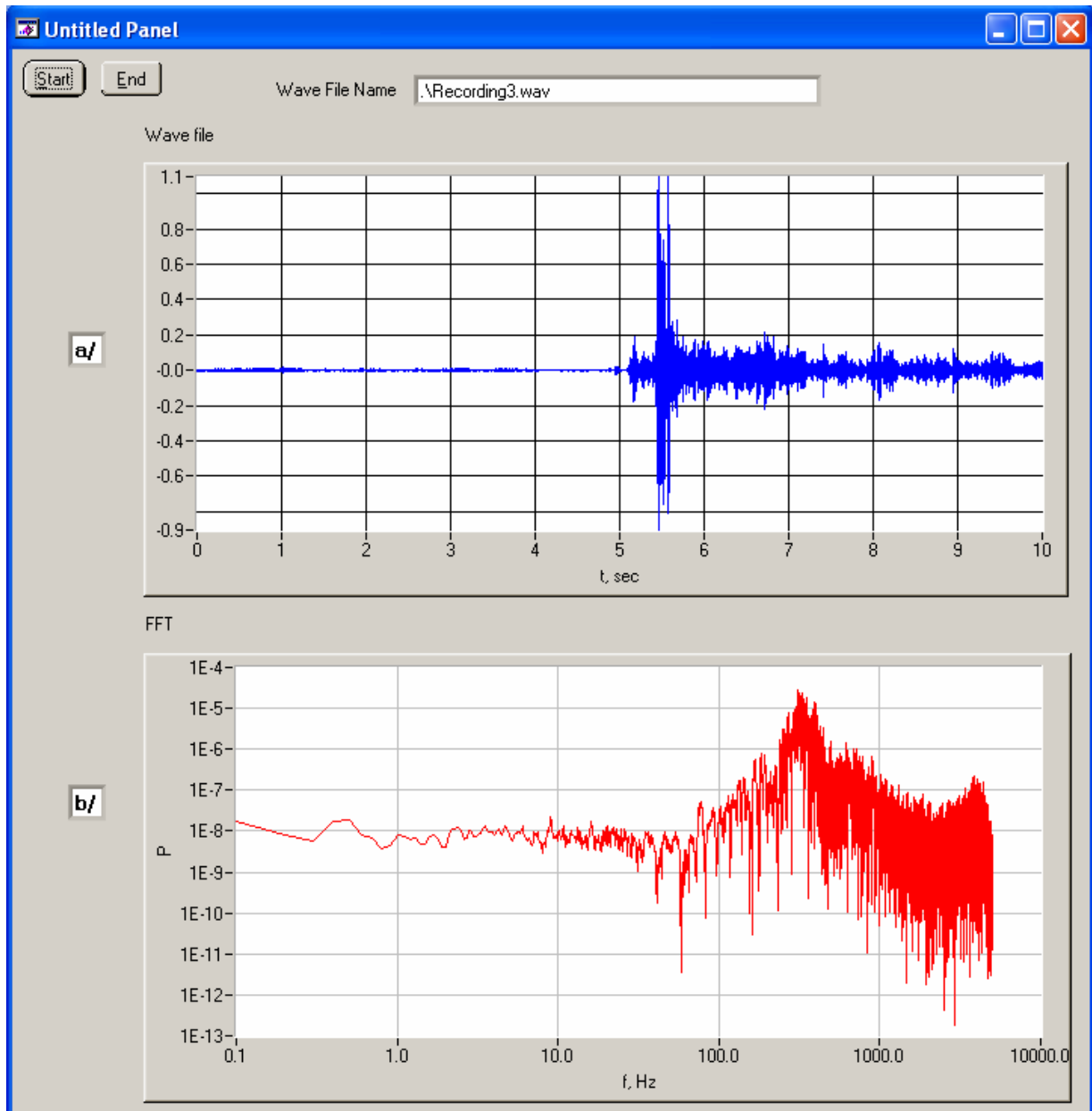


Figure 34 : traitement fichier .wav

a/ Représente l'amplitude du son en fonction du temps.

b/ Donne l'amplitudes des ondes acoustiques en fonction des fréquences. Elles sont calculées en faisant la FFT (Transformée de Fourier Rapide) du premier signal.

La durée d'enregistrement de l'échantillon est de dix secondes avec comme nous l'avons vu plus haut, un taux d'échantillonnage de 10 kHz soit 1000 mesures par seconde.

Sur l'aperçu du programme de traitement des fichiers « wave » que j'ai réalisé, on observe aisément que le gros de l'éclair s'est produit entre la 5 et la 6ème seconde. Je peux donc grâce aux données du fichier déterminer le moment où cet éclair s'est produit.

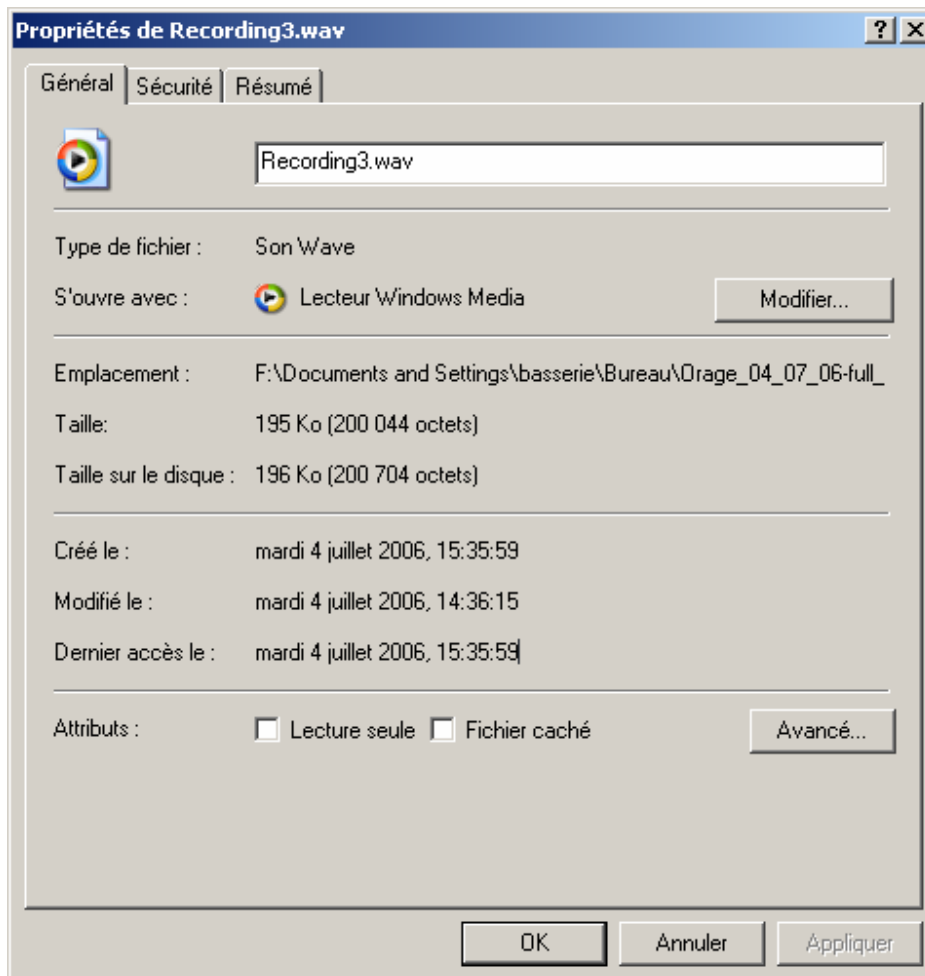


Figure 35 : date du fichier .wav

Sachant que le fichier est créé en fin d'acquisition (figure 35), l'éclair s'est donc produit le : mardi 4 juillet 2006 à 14 : 36 : 10 environ.

Pour des raisons de visualisation on ramène l'amplitude du signal entre -0.9 et 1.1 au lieu de -255 et +255.

Et pour la caractérisation on trace aussi la FFT (Transformée de Fourier Rapide) de l'échantillon ce qui me permet de connaître les fréquences mises en jeu par le tonnerre.

Après plusieurs observations des différentes mesures réalisées, j'ai constaté que la fréquence dominante des ondes acoustiques se trouve autour de 300 Hz, ce qui correspond bien à la théorie qui la situe entre quelques dizaines et quelques centaines de hertz.

4.8 - L'architecture temps réel

Dans le cadre du projet, une architecture temps réelle fut mise en place.

Le principal atout qu'elle nous apporte par rapport à une architecture séquentielle est de pouvoir contrôler et dialoguer avec plusieurs périphériques (appareil photo, GPS...) en même temps.

Le principal inconvénient de la communication avec les périphériques est le fait qu'ils gardent la « main » le temps qu'ils réalisent la tâche qu'on leur demande. Ce problème impose donc un temps de réaction très long de notre système.

La solution la plus simple est donc de mettre en place une architecture de programme en temps réelle avec la gestion des différents périphériques ou fonctions en parallèle par le biais de « thread ».

4.8.1 - Définition

Les threads sont des processus légers (en français) qui sont similaires aux processus en cela qu'ils représentent tous deux l'exécution d'un ensemble d'instructions du langage machine d'un processeur. Du point de vue de l'utilisateur ces exécutions semblent se dérouler en parallèle « multithreading ». Toutefois là où chaque processus possède sa propre mémoire virtuelle, les processus légers appartiennent au même processus père et partagent une partie de sa mémoire virtuelle.

4.8.2 - Les threads

Dans mon application ce qui coûte le plus de temps est la lecture des photos (fichier JPEG, entre 2s et 6s) et le dialogue avec l'appareil photo.

Toutes les actions de ce dernier sont longues à réaliser est ne demande que très peu d'utilisation du processeur, mis à part au lancement et lorsqu'il rend la main. Le fait de partir sur un système temps réel me permet d'utiliser ce temps où le processeur du PC n'est pas demandé pour effectuer d'autres tâches.

4.8.3 - Architecture

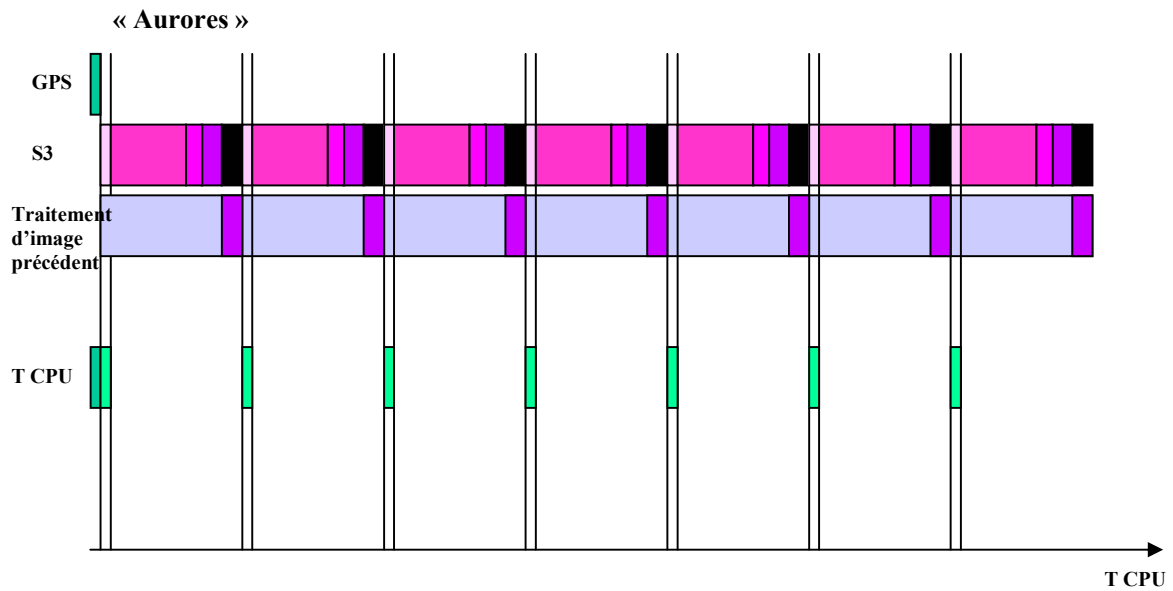


Figure 36 : Architecture temps réel

Dans cet exemple (figure 36) deux « threads » se partagent les accès au CPU. L'un est S3 : il comprend toutes les directives données à l'appareil photo. L'autre est le traitement de la photo réalisée précédemment. Cette opération utilise uniquement la mémoire du PC et son CPU.

4.8.4 - Détail des opérations

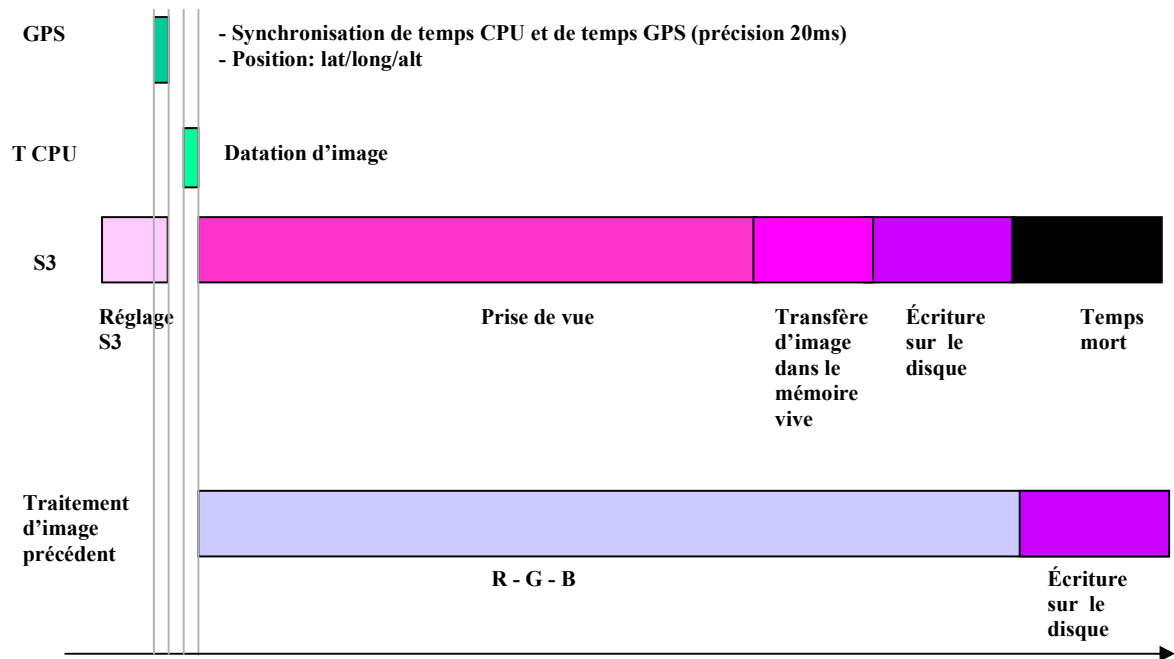


Figure 37 : détaille architecture temps réel

Comme nous le voyons figure 37, je commence par réaliser les réglages de l'appareil photo (initialisation), puis je lance la synchronisation du GPS pour obtenir la date précise pour ensuite pouvoir dater l'image qui va être réalisée. Finalement je lance en parallèle l'acquisition de l'image ainsi que le traitement de l'image précédente.

Un thread gère alors toutes les communications avec l'appareil photo et récupère les données de celui-ci.

Un autre thread, lui, utilise les données de l'image faite à l'instant d'avant pour les traiter et en sortir les informations voulues.

4.9 - Détection sur l'image du centre et des bords de l'objectif

Un point important dans l'étude des images que j'obtiens grâce à l'appareil photo est de définir le centre de l'objectif ainsi que ses bords (figure 38).



Figure 38 : centre de l'image

Dans cette optique j'ai dû réaliser un programme permettant de retrouver ces facteurs en partant d'une image et ce, quel que soit l'objectif placé sur l'appareil photo.

Un moyen simple est de prendre une photo sur laquelle il n'y a qu'une couleur claire à peu près constante visible sur l'image.

Le passage au noir sur l'image me détermine alors le bord de l'objectif.

Le problème est que le bord de l'objectif n'est pas un noir constant et net du point de vue des valeurs des pixels. Il y a une frontière sur le bord de l'objectif où l'image se déforme et où la lumière se diffuse sur les pixels (figure 39).

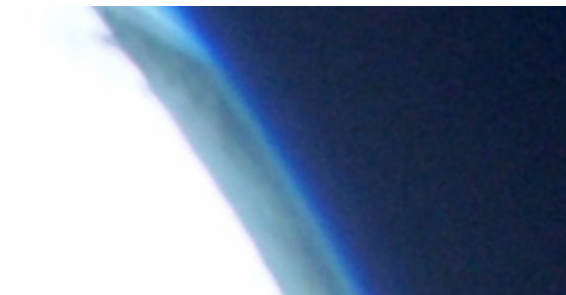


Figure 39 : Zone de lumière diffuse

En supposant que la couleur est constante, je commence donc par calculer la valeur moyenne de la couleur de mon image sur chaque ligne. Puis toujours pour chaque ligne, j'en prends la moitié

et je récupère ensuite les coordonnées du premier et du dernier pixel qui atteint cette valeur. Je le considère comme étant le bord de mon objectif.

Comme mon objectif est cylindrique j'obtiens donc un nuage de point qui représente un cercle.

Je calcule ensuite par la méthode des moindres carrés, le vrai cercle s'approchant le plus de celui formé par le nuage de points et j'en détermine le centre.

Nous savons que dans un plan muni d'un repère orthonormé, l'équation d'un cercle de centre $C(a,b)$ et de rayon r est :

$$(x - a)^2 + (y - b)^2 = r^2$$

L'approche par la méthode de moindre carré nous est donnée par :

$$\mathcal{E} = \sum_i^N \left\{ (x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right\}$$

x_0, y_0 définissent le centre du cercle.

x_i, y_i définissent les coordonnées d'un point du cercle.

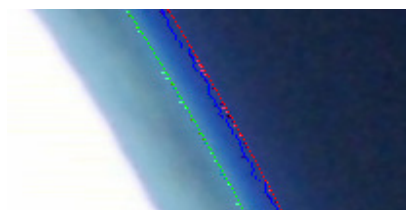
r est le rayon du cercle

i est l'indice des coordonnées

N est le nombre de points du cercle considéré

Je commence par fixer des valeurs x_0, y_0 et r de mon cercle et je les fais ensuite varier d'un delta de ± 1 en parcourant les N points du cercle. Je garde alors le cas où le \mathcal{E} est le plus petit.

Résultat obtenu :



En Bleu le nuage de points.

En Vert le cercle ayant pour origine le centre de l'image.

En Rouge le cercle obtenu par calcul avec le centre recalculé.

Figure 40 : Résultat obtenu

V - CONCLUSION

Ce stage s'effectuant sur une longue période m'a beaucoup appris tant sur le plan humain, car la réalisation d'un projet implique une communication importante au sein de l'équipe le réalisant, que sur le plan professionnel, car un stage de 6 mois permet de réaliser des choses dans lesquelles on a le temps de s'investir et de se documenter.

Mon stage m'a permis de me mettre dans différentes situations : le fait de travailler seul sur les tâches que l'on doit réaliser, mais aussi le fait d'avoir à faire des propositions et à débattre des solutions à envisager pour la suite des opérations.

De plus, mon encadrement et l'équipe de St Maur furent exceptionnels, toutes mes questions ont toujours trouvé une réponse claire ou une direction de recherche.

Sur le plan professionnel, j'ai pu compléter ma formation en apprenant énormément sur le pilotage des appareils, les acquisitions sonores et la compression JPEG avec ses variantes, le traitement des images et les calculs astronomiques.

J'ai été ravi des responsabilités qui m'ont été offertes, comme l'encadrement momentané d'un technicien informatique ou encore le fait de rechercher du matériel, d'en gérer la commande et la réception avec les fournisseurs.

Mon intégration n'étant plus à faire du fait que je connaissais déjà l'équipe du CETP, une confiance s'est installée et j'ai par moments été sollicité par des collègues pour essayer d'apporter des réponses à leurs questions dans le domaine de l'informatique.

Pour finir, à côté de mon projet, j'ai dû réaliser des recherches et de la documentation pour d'autres personnes sur certaines parties liées à mon projet.

Mon stage s'est fini mais pas mon travail au sein du CETP, car j'ai reçu deux propositions de CDD. J'en ai bien sûr accepté une (je ne pouvais pas prendre les deux). Je me lance donc dans la vie active directement après mon stage et dans le domaine que j'affectionne : l'informatique embarquée et temps réel.

VI - BIBLIOGRAPHIE

Décompression TIFF

<http://www.awaresystems.be/>

Bibliothèque JPEG et ses explications

JPEG Independent Group <http://www.ijg.org/>

Labwindows

CVI – The Rebel Site <http://www.perso.orange.fr/philippe.baucour>

Appareil photo

site fuji <http://www.fujifilm.fr/>

Software Development Kit Manual, Fuji, 2005.

GPS

Lassen LP GPS System Designer Reference Manual, Trimble, 1999.

Port parallèle

IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface For Personal Computers, Document Number: IEEE 1284-2000, 1994.

Bibliothèque WAVE

FMOD <http://www.fmod.org/>

Calculs astronomiques

Meeus, J., *Astronomical Algorithms*, Willmann-Bell, 1991.

Approche de l'astronomie <http://www.emilie.bodin.free.fr/>

VII - ANNEXE