Table des matières

I. Présentation

1.1. Présentation de l'entreprise

L'Ircam (Institut de Recherche et Coordination Acoustique et Musique) est un institut de recherche français public basé à Paris, au statut un peu particulier parmi les instituts de ce type. Il regroupe en son sein des équipes de recherches spécialisées par domaines, mais aussi un pôle production musicale. L'Ircam propose donc, parallèlement aux activités de recherche, divers services pour la création musicale contemporaine, en mettant à disposition des compositeurs ses studios et laboratoires. L'institut organise annuellement un festival dédié à la musique contemporaine (Festival Agora) et dispose également d'une aile pédagogique qui propose des stages et des formations à l'informatique musicale. Les départements de recherche travaillent pour partie en collaboration avec les artistes, en leur proposant les outils développés au sein de l'Ircam, participant ainsi à l'élaboration des Créations musicales et autres Performances.
De par sa structure et son mode de fonctionnement, l'Ircam est une institution unique en France. Créé en 1970 à l'initiative du compositeur Pierre Boulez, l'Ircam s'est imposé comme un acteur majeur de la recherche dans le domaine du son et de la musique en développant des concepts aujourd'hui devenus standards dans le domaine du traitement du signal, et des musiques électroniques et contemporaines. L'Ircam s'est illustré en proposant des outils de composition, d'analyse et de performance novateurs, à l'image de Max MSP, OpenMusic, ou plus récemment des systèmes de modélisation physique tel le logiciel Modalys, ou bien le système de diffusion sonore WFS (spatialisation et diffusion sonore basée sur le synthèse de fronts d'ondes). L'Ircam distribue ses logiciels via une communauté d'utilisateurs (le « forum »), et a conservé un lien étroit avec le monde du logiciel libre.
L'Ircam s'est donc imposé au fil des années comme un institut leader dans le domaine et bénéficie aujourd'hui d'un rayonnement international.

Supprimé : performances

Le département recherche soutient 8 équipes de recherche (Analyse et synthèse, acoustique des salles, acoustique instrumentale, design sonore, représentations musicales, Analyse des pratiques musicales, services en ligne et Interactions musicales temps réel), dans lesquelles sont répartis 90 chercheurs.
Nous nous intéresserons plus particulièrement dans ce rapport à l'équipe Interactions Musicales Temps réel (IMTR) puisque c'est en son sein que le stage s'est déroulé.

L'équipe IMTR de l'Ircam est naturellement associée à ce projet, pour ses compétences et développements en matière autour de l'interaction musicale. L'équipe travaille notamment sur le geste musical, le traitement sonore en temps réel et l'augmentation d'instruments, et est notamment auteur du *Gesture Follower*, un moteur de suivi du mouvement. L'équipe travaille donc également dans un contexte de création musicale, en collaborant avec des compositeurs, mais aussi dans des contextes pédagogiques et industriels.

Supprimé : compsiteurs

Sur le plan de la recherche, l'Ircam participe à nombres de projets d'impulsions européennes ou nationales, à l'image des projets SAME, i-Maestro ou Interlude, projet dans le cadre duquel ce stage s'est déroulé.

Le projet Interlude (projet ANR Contenu et Interaction ANR-08-CORD-010) est un projet national, lauréat d'un appel à projet lancé par l'Agence Nationale de la Recherche (ANR). Ce type de projet ANR se présente généralement sous la forme d'un consortium de partenaires. Interlude rassemble donc 6 partenaires aux compétences complémentaires :
L'Ircam joue le rôle de coordinateur du projet, et participe aux développements matériels et logiciels.
NoDesign est une agence spécialisée et reconnue dans le domaine du design numérique, et assure le design des différents éléments matériels prévus par le projet.
Da Fact est une PME travaillant sur les contrôleurs musicaux en proposant des contrôleurs de « nouvelle génération », et travaille conjointement à NoDesign et l'Ircam sur les aspects design, développement électronique et intégration mécanique.
Voxler est une entreprise française spécialisée dans le traitement de la voix pour des applications du Jeu Vidéo. Elle est donc en charge de la valorisation industrielle du projet ainsi que de l'intégration éventuelle de moteurs de traitement du signal dans des applications grand public et ludoéducatives.
Le Grame est centre de création musicale basé à Lyon, et contribue au projet avec son travail autour de la partition augmentée permettant une représentation du temps musical associée à des annotations et une représentation du geste ou de l'interaction intégrés au déroulement de la partition.
L'Atelier des Feuillantines est un organisme proposant des formations musicales à des niveaux variés, et représente donc un lien direct avec la pédagogie musicale. Son rôle est l'élaboration des scénarios pédagogiques, leur dissémination et leur validation.

Mis en forme : Police :Tahoma, 10 pt

Supprimé : un

Supprimé : implantée dans le domaine du jeu vidéo,

1.2. Présentation de la mission

Mon stage précédent (ING4) s'étant également déroulé à l'Ircam et au sein du même projet, ce stage s'inscrit dans la continuité du précédent.

Le projet Interlude prévoit la réalisation d'un ensemble d'objets tangibles inter opérants destinés au contrôle musical.
Le contrôle musical et les interfaces qui lui sont dédiées ont été des domaines largement investis en recherche cette dernière décennie. Rétrospectivement, ces interfaces sont la plupart du temps dédiées à un contexte d'utilisation, voire à une performance ou expérimentation spécifique. Un des points clés du projet est de créer un ensemble d'objets constituant une interface au caractère évolutif et multimorphe.

Mis en forme : Français (France)

Dans cette optique, les composants principaux du système ont été imaginés de façon à pouvoir être augmentés et combinés: des ports supplémentaires permettent le branchement d'accessoires autour d'un appareil central, les accessoires pouvant être actifs comme passifs. Les objets peuvent ainsi être détournés de leur utilisation première.

L'architecture prévoit quatre principaux éléments. L'ordinateur en tant qu'interface utilisateur permet la réception, la visualisation des données, la configuration du système ainsi que la synthèse sonore. C'est lui qui gèrera l'exercice (également nommé « scénario » dans ce document).
La base OSC (que l'on pourra aussi appeler « récepteur ») est le pont entre le domaine radio 802.15.4 et le réseau filaire câblé en Ethernet. Elle prend donc en charge la conversion des protocoles OSC vers radio et inversement, ainsi que la création et la maintenance du réseau.
L'émetteur est en quelque sorte l'interface physique : c'est grâce à lui que l'utilisateur pourra interagir avec le système, contrôler différents paramètres musicaux.
Enfin l'accessoire est l'élément qui couplé à l'émetteur, fait du système une interface capteurs. L'accessoire peut prendre différentes formes, et donc abriter divers capteurs, ce qui confère au système son caractère modulaire et évolutif.
Parallèlement à ces éléments, le système propose une base dédiée à la recharge des émetteurs.

La mission menée au cours de ce stage s'articule autour de ces principaux éléments. De nombreux choix technologiques ont été fait à l'issue d'une réflexion en amont, et nécessitent donc d'être mis en œuvre pour validation et poursuite des recherches. L'architecture matérielle du système étant définie dans les grandes lignes, il s'agit donc de participer aux différentes itérations de développement d'un prototype opérationnel du système.

1.3. Exemples d'applications

Des scénarios d'utilisation ont été imaginés puis mis en place afin de caractériser les besoins du système et d'illustrer son application. Nous reviendrons ici sur deux exercices qui ont été réalisés pour Interlude, qui se concentrent ici principalement sur le tempo et la musicalité du geste. Les deux exercices ont pour objet d'étude les variations Goldberg de Jean-Sébastien Bach, morceau dont la structure en « canon » se prête parfaitement aux jeux de type « question - réponse » par exemple.

Mis en forme : Justifié

Le premier exercice s'articule autour d'un jeu d'échec : deux élèves dirigent le déroulement du morceau en contrôlant la lecture des phrase musicales par le déplacement des pièces sur l'échiquier. Du fait de la structure en canon, les phrases se répondent successivement, et chaque élève contrôle tour à tour différents paramètres par le geste qu'il choisira (temps, phrasé, dynamique). Des paramètres comme l'accélération à la levée ou au posé, l'amplitude et la vitesse du mouvement peuvent ainsi se traduire par une interprétation différente lors de la lecture d'une phrase musicale.

Le deuxième exercice comprend certaines similitudes avec le premier, notamment au niveau des modalités d'interaction sur le son, puisque le même type de paramètres pourra être extrait du geste des participants. Cet exercice, toujours basé sur les variations Goldberg, prend la forme d'un jeu de balle : de ce fait, ce scénario peut faire intervenir un plus grand nombre d'élèves, manipulant une ou plusieurs balles sous formes de « passes », mettant en avant l'aspect collaboratif que peut adopter l'interface, qui reste simple et ludique.

II. Cahier des charges

Développement réseau:

- Mise en place du protocole réseau (IEEE 802.15.4)
- Gestion autonome et coordination du réseau
- Développement des plateformes Base (Coordinateur) et Émetteurs (End Device)
- Messagerie
  - Création d'un système de messagerie
  - Trames formatées et différenciées
  - Encodage et décodage des informations transitant sur le réseau
  - Optimisation et respect de la contrainte temps réel (latence)

Développement module coordinateur/base:

- Interface OSC pour la communication coordinateur/ordinateur, pile bidirectionnelle:
  - Encodage de données en OSC
  - Décodage des messages OSC envoyés par l'ordinateur
  - Pile de traitement (FIFO) 802.15.4/OSC
  - Table de routage
- Liaison Ethernet coordinateur/ordinateur
- Modification du mode configuration de la base (ajout de possibilités de paramétrisation)
- Gestion des transmissions et ajustement automatique du débit

Supprimé : ¶

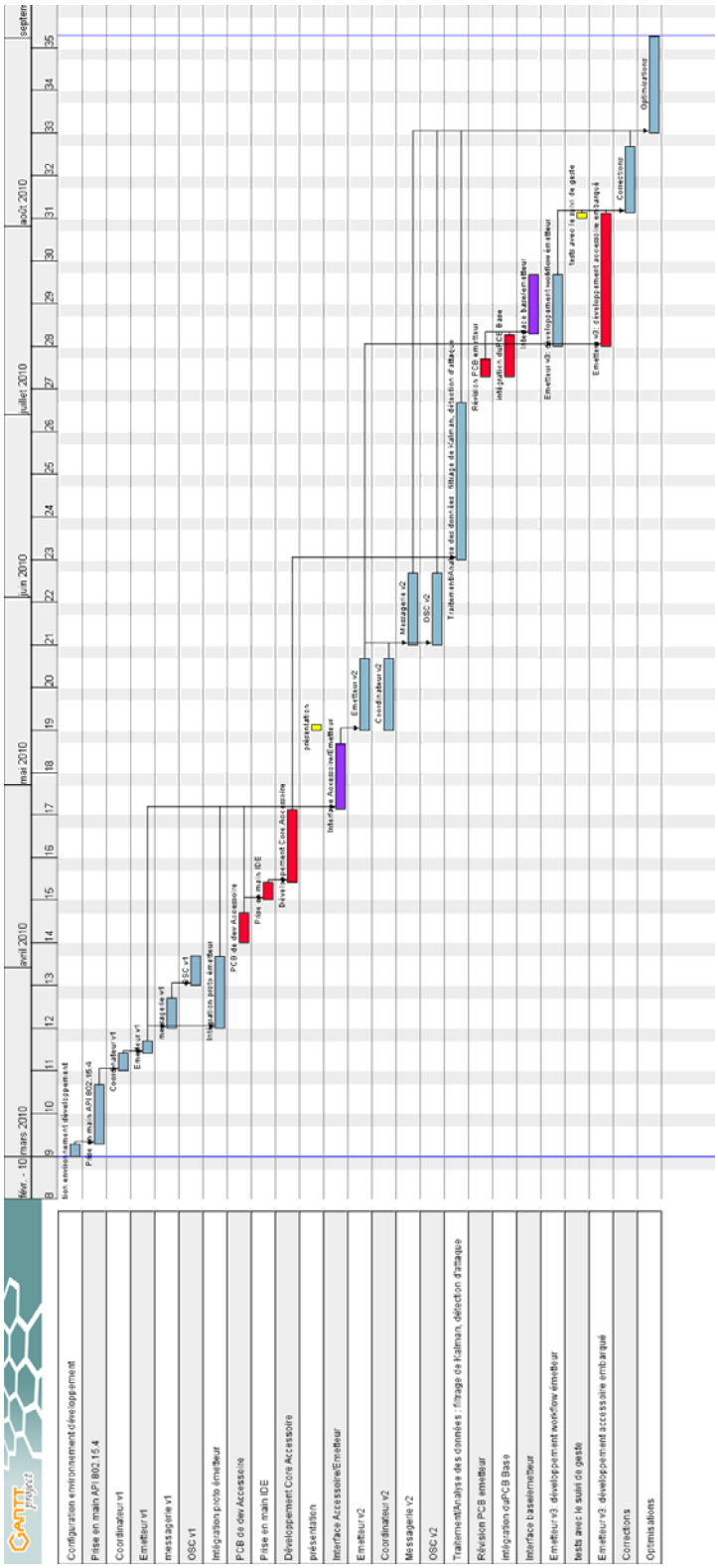
Développement modules émetteurs:

- Gestion de l'interface utilisateur : boutons, afficheurs etc.
- Configuration du périphérique I2C
- Développement de l'architecture orientée services
- Implémentation de services « internes »
- Gestion de la recharge de la batterie
- Développement d'une interface de configuration

Développement des accessoires:

- Conception d'une carte de développement spécifique aux accessoires
  - Choix du micro contrôleur
  - Choix des périphériques utilisés
- Réalisation d'un prototype regroupant les fonctionnalités communes avec tous les accessoires
- Développement spécifique aux accessoires
  - Système d'interprétation des commandes (service/commande)
  - Mise en place de traitements et analyses adaptés à chaque accessoires, embarqués sur l'accessoire même
  - Mise à jour du système de messagerie en conséquence

III. Planning



IV. Choix technologiques

4.1. Design

Le design des différents éléments du système a été réalisé par No Design, agence de design spécialiste du design numérique.
Le design de certains éléments n'est cependant pas fixé à ce jour, comme c'est le cas pour certains accessoires ou pour la base OSC. Le design des émetteurs en revanche est abouti, puisque primordial étant donné son importance dans l'interaction avec l'utilisateur. Ce design apparaît sur les illustrations ci-dessous.

Les versions de la coque utilisées actuellement ont été réalisées à l'aide d'une imprimante 3D, les coques sont donc fabriquées en dépôt de fil. L'intégration mécanique, qui consiste à la mise en adéquation des géométries de la carte électronique avec celle du boitier, a été réalisée par Da Fact.
(voir photo ci-contre).

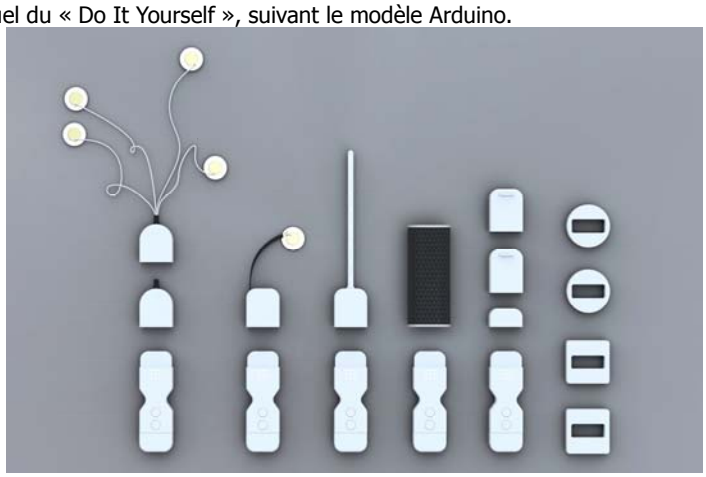
Supprimé : les

Les versions futures devraient être moulées en stéréo lithographie, procédé permettant une finition largement plus appréciable mais également plus onéreuse et donc moins adapté au contexte de prototypage (voir ci-contre).
De nombreux accessoires ont également été imaginés conjointement à des scénarios d'utilisations (voire illustration ci dessous).

4.2. L'aspect modulaire

Nous avons déjà succinctement évoqué, dans les objectifs du projet Interlude, l'aspect modulaire. La conception du système est basée sur le constat suivant : Les interfaces destinées au contrôle musical sont en général conçues pour satisfaire des exigences propres à une performance unique, on un type d'expérience, et restent de ce fait réservées à des utilisateurs experts. L'idée des objets d'Interlude est donc de donner un caractère évolutif à des interfaces initialement conçues pour une utilisation non experte, en fondant son concept sur le succès actuel du « Do It Yourself », suivant le modèle Arduino.

Le système prévoit donc un ensemble d'éléments fixe constituant la base du système, mais les émetteurs sont conçus pour être « augmentés » avec des accessoires. Si plusieurs accessoires ont été imaginés par l'équipe de design et de développement, il est également envisageable de créer ses propres accessoires, et certains accessoires ont d'ailleurs été conçus pour faciliter cette démarche : à titre d'exemple, un accessoire offrant plusieurs entrées analogiques permettant à l'utilisateur d'y brancher librement des capteurs rend ainsi l'interface adaptable à d'autres scénarios d'utilisations.



Afin de permettre cette ouverture, certains choix technologiques sont requis pour uniformiser les échanges entre accessoires et émetteurs, sur lesquels nous reviendrons plus en détail dans la suite de ce document.

4.3. Le principe de « services »

Autre principe clé du système, l'organisation en « services ». L'idée maîtresse est la suivante : l'accessoire est un élément polymorphe, qui décline l'émetteur en différents objets. Chaque accessoire peut fournir des données différentes, et des traitements de ces données différents également : ils proposent chacun des services particuliers. Cependant, ils gardent en commun la même base logicielle, qui alliée aux services proposés constituent l'accessoire. Cette base logicielle permet à n'importe quel accessoire de venir augmenter l'émetteur de la même manière, rendant en quelques sortes l'émetteur reconfigurable et multifonctions. Le terme de service est propre aux accessoires : chaque accessoire possède un ou plusieurs service, qu'il met à la disposition de l'émetteur auquel il est relié et donc de l'utilisateur. Un service propose donc à l'utilisateur de récupérer des données sous une forme bien particulière. A titre d'exemple, un couple accéléromètre/gyroscope peut bien entendu proposer un service de lecture des données brutes, mais peut également proposer des données traitées, comme par exemple une reconstitution des angles dans l'espace de rotation. Dans ce cas, une analyse est faite directement dans l'accessoire, qui lit les données, les traite selon le service spécifié, puis les restitue à l'émetteur.
Tout service à bien évidemment son lot de paramètres : on peut ainsi décider de ne récupérer qu'un axe d'un accéléromètre, de modifier la fréquence d'échantillonnage, de modifier les paramètres d'un filtre...
L'émetteur engage un processus de découverte de services lors de la connexion d'un accessoire : l'accessoire lui fournit un identifiant spécifique (qui fera également office d'adresse si l'utilisateur souhaite le paramétrer par la suite), ainsi qu'une liste détaillée des services qu'il est en mesure de fournir et des paramètres de ces services. Ces informations sont transférées jusqu'à l'ordinateur, de telle sorte que l'utilisateur puisse communiquer avec le nouvel accessoire.

Les avantages de ce fonctionnement sont évidents : Si un prétraitement des données peut être fait par un microcontrôleur, cela permet de ne pas avoir à la faire ailleurs, et ce traitement est d'autant plus performant qu'il a lieu juste après l'acquisition. La charge de calcul de la machine hôte peut donc être utilisée à un autre profit, d'autant plus que le champ d'application pédagogique envisagé pour le système laisse à penser que les machines hôtes dont sont équipés les structure pédagogiques peuvent avoir des performances modestes comparées à celles de machines plus récentes ou simplement plus performantes. Enfin, la dimension collaborative impliquée par le même contexte pédagogique mais aussi pour d'autres types de performances interdit la lecture et l'envoi des données en « flux tendu » (streaming). En effet la bande passante des réseaux sans fil type 802.15.4 ne permet pas à plusieurs nœuds d'envoyer des données en flux tendu tout en conservant un temps de latence acceptable pour des applications dîtes « temps réel ».Pour ce type d'utilisation, pourtant fréquent, une architecture « point à point » est préférable, autrement dit une liaison unique entre la base et l'émetteur, pour obtenir une latence la plus faible possible. Le fonctionnement avec plusieurs émetteurs simultanément est possible, mais implique inévitablement une augmentation de la latence proportionnelle au nombre d'émetteurs en fonctionnement (en « streaming » bien entendu).
L'architecture des services permet donc une alternative : si un émetteur peut transférer des données le plus rapidement possible, il peut aussi attendre la détection d'un événement particulier décrit dans par un service pour notifier le réseau, et ainsi limiter son utilisation de la bande passante à l'envoi de ce message.

V. Spécifications techniques

5.1. Réseau

Le réseau utilisé est un réseau de type « Private Area Network » (PAN) conformément à la norme IEEE 802.15.4. Ce type de réseau, situé sur la bande 2,4 GHz, est particulièrement adapté aux réseaux de capteurs et aux applications nécessitant une consommation d'énergie réduite.
Le débit nominatif de ce réseau est de 250 Kb/s, et il offre une portée locale intermédiaire, dépendant des applications (entre 20 et 30 m, confortablement). Les modules électroniques utilisés pour la mise en place, la gestion et la participation à ce réseau sont des microcontrôleurs JENNIC JN-5139, qui sont des microcontrôleurs permettant la prise en charge réseau via une pile fournie par le constructeur, tout en mettant à disposition de l'utilisateur un certain de nombre de périphériques standards chez les microcontrôleurs. A la différence des modules Xbee par exemple, les modules Jennic disposent d'une interface de programmation dotée d'APIs pour le matériel comme pour le réseau, et offrent donc une flexibilité accrue comparés aux populaires Xbee et consorts, plus « plug and play » mais moins personnalisables. La topologie en « étoile » a été adoptée pour le projet: les émetteurs sont connectés à une même base, elle même reliée à la machine hôte. La base est le coordinateur du réseau.

5.2. Open Sound Control (OSC)

Une interface est nécessaire entre le domaine défini par le réseau sans fil et la machine hôte. La base OSC/récepteur prévoit une liaison Ethernet avec la machine hôte, permettant ainsi l'utilisation du protocole UDP.
L'OSC est un standard de plus en plus populaire, utilisant les technologies réseaux modernes, et que l'on peut donc envisager être le successeur du MIDI (la prise en charge OSC se développe chez les applications audio prenant en charge le MIDI). Le protocole OSC définit un format de données et d'adressage et se place en couche supérieure aux protocoles réseau UDP et TCP. Il permet un adressage arborescent, la transmission de différents types de données (entiers, flottants, chaines de caractères, symboles…), sans autre contrainte que la longueur maximale d'une trame UDP classique (de l'ordre de 1500 octets).
Tous les échanges entre la base OSC et la machine hôte sont donc standardisés conformément à ce protocole.

5.3. Le bus I2C

Le standard de communication adopté pour les échanges entre les émetteurs et les accessoires est le bus I2C. Il s'agit d'un protocole de communication introduit par Philips dans les années 90 et devenu standard depuis. De nombreux composants proposent directement une interface I2C.
Le bus I2C propose un fonctionnement de type maître/esclave, et la technologie d'adressage permet de connecter plusieurs appareils sur le même bus. Le bus peut donc comporter plusieurs maîtres, bien que ce ne soit pas le cas dans notre situation.
Le module Jennic JN-5139 ne peut fonctionner qu'en tant que maître du bus, le microcontrôleur connecté au bus si un accessoire est présent fonctionne donc en tant qu'esclave, tout comme les capteurs inertiels embarqués dans l'émetteur. Le bus est configuré pour fonctionner à la vitesse maximale autorisée (400 kHz).

5.4. Composants

5.4.1. JENNIC JN-5139

Le module JN-5139, déjà évoqué, est au centre du système en sa qualité de microcontrôleur avec contrôleur réseau 802.15.4 intégré. Les modules « Base OSC » et « émetteur » sont donc organisés autour de ce composant.
Ses caractéristiques principales sont les suivantes:

- processeur RISC 32 bits (fréquence d'horloge de 16 MHz)
- 96ko de RAM
- Nombreux périphériques :
  - Flash
  - bus I2C
  - bus SPI
  - UARTs
  - Timers
  - Convertisseur analogique numérique 12 bits 4 canaux
  - Convertisseur numérique analogique 12 bits 2 canaux
  - Comparateurs
  - Capteur de température intégré

Les spécifications techniques du Jennic JN-5139 le placent plutôt dans une catégorie intermédiaire de microcontrôleurs : puissance de calcul modérée (16 Mhz), architecture 32 bits, doté des périphériques standards et essentiels. Cette catégorie de microcontrôleurs est en général programmée en C, via un compilateur spécifique, souvent fourni par le fabricant. C'est également le cas de Jennic, qui met à disposition une version personnalisée de Code::Blocks pour programmer les modules, associée à un programmateur flash via un simple câble USB/UART.
Jennic met également à disposition diverses API matérielle et réseau, déclinées en plusieurs versions pour s'adapter au différentes utilisations : configuration peu poussée mais mise en route rapide ou configuration plus approfondie mais nécessitant bien entendu des outils (APIs) et un temps de développement plus poussés.

D'un point de vue réseau, le JN-5139 peut être programmé en tant que coordinateur, routeur ou terminal (« end device ») L'API la plus approfondie fournie par Jennic propose tous les mécanismes nécessaires au fonctionnement du réseau que l'on soit du côté « end device », coordinateur ou routeur.

L'Ircam avait adopté les modules Xbee pour créer des systèmes sans fil point à point depuis l'apparition des modules Xbee mis sur le marché par Digi, anciennement Maxstream. Les modules Jennic offrent une alternative de choix aux Xbee de par leurs spécifications techniques, et les modules issus du projet Interlude sont envisagés comme successeurs aux modules Xbee.
En effet, l'environnement de programmation offre bien plus de contrôle à tous les niveaux (fréquence d'émission, conversion, dialogue, traitement des données...), et les périphériques proposés par le Jennic offrent des performances supérieures à ceux du Xbee, en remplaçant par exemple une liaison UART par un bus SPI, ou en offrant la possibilité de relier des capteurs dotés de d'une interface numérique sur un bus I2C.
Le facteur « taille » est également décisif : le Jennic est de taille plus réduite, au format CMS à l'inverse du Xbee, ce qui rend plus simple son intégration mécanique.

5.4.2. Microcontrôleur Microchip PIC

Nous avons déjà parlé brièvement de l'aspect modulaire. En effet l'évolutivité et la possibilité de brancher de nouveaux accessoires pour venir compléter les émetteurs nécessitent la définition de standard de communication, tant au niveau logiciel que matériel (brochage des ports...).

Deux catégories d'accessoires sont à distinguer : actifs et passifs. Les accessoires passifs n'ont en leur sein aucun capteur ou processeur, et n'ont donc d'incidence que sur les aspects haptique et visuel. Les accessoires passifs sont au contraire équipés de capteurs, et c'est naturellement dans ce cas qu'il est nécessaire de standardiser la liaison avec l'émetteur, de façon à ce que ce dernier n'ai pas (ou le moins possible) de cas particuliers à prendre en charge (faible dépendance matérielle vis-à-vis des accessoires).
C'est à ce niveau qu'intervient le choix du microcontrôleur PIC pour plusieurs raisons:

- différents types de capteurs selon les accessoires

Considérant le choix du bus I2C comme medium de communication entre l'émetteur et l'accessoire, il est évident que certains accessoires imaginés sont basés sur des capteurs analogiques, qui ne pourront donc pas être branchés directement sur un bus I2C. Il est donc indispensable de placer une interface de conversion analogique numérique (I2C) en amont.

- Adressage I2C

Considérons un instant que les capteurs soient numérisés par un convertisseur capable de restituer les données sur un port I2C. Un problème d'adressage se pose alors, car il serait contraire aux principes de modularité et d'évolutivité et très limitant de devoir maintenir chez l'émetteur une liste des adresses I2C de tous les accessoires que l'utilisateur pourrait potentiellement brancher.
Le placement d'un microcontrôleur entre les capteurs pallie à ce problème car son périphérique I2C est entièrement configurable, et l'on peut donc lui assigner une adresse unique. De cette manière, l'émetteur connaît peut communiquer avec tout type d'accessoire à la même adresse, mais peut également la modifier en cours d'exercice.
Avec un tel système l'émetteur n'est agit comme simple passerelle des données vers le réseau : son interprétation de celles-ci est limitée au transfert d'un protocole (I2C) vers un autre (radio).

- Prétraitement des données

Enfin le système, de par son orientation « services », suggère un traitement des données différent selon les services proposés. Pour les mêmes raisons que l'adressage, il paraît impensable de stocker les algorithmes de traitement de données de tous les accessoires sur le microcontrôleur JENNIC puisque les limites d'un tel système sont évidentes, tant en terme de puissance de calcul (le module gère également le fonctionnement radio), de mémoire disponible pour stocker le code, que de maintenance et de mise à jour du logiciel enfoui (firmware).
Dans le cas d'une liaison numérique, en l'occurrence I2C, reliant les microcontrôleurs sur l'émetteur et sur l'accessoire, ces algorithmes de traitement spécifiques à chaque service de chaque accessoire peuvent être exécutés directement dans l'accessoire, de telle sorte que le module JENNIC ne soit alors qu'une passerelle entre l'accessoire et le réseau.
Cette solution permet donc, en définissant un protocole de découverte et d'échange de services, de déporter le traitement sur un autre microcontrôleur afin de préserver la fluidité du fonctionnement de la couche réseau, et de conserver une politique d'adressage simple et cohérente.
Le choix du microcontrôleur s'est porté vers la famille de processeurs 16 bits PIC24F, et plus particulièrement le PIC 24FJ64GA004, dont les principales spécifications figurent en annexe de ce document.

Ce modèle fait partie de la nouvelle génération « nanowatt » de microcontrôleurs Microchip, et s'illustre comme cette appellation le laisse entendre par sa faible consommation d'énergie.
Les ingénieurs de l'Ircam étant déjà familiers avec les microcontrôleurs Microchip et possédant les outils de développements adaptés (compilateur, programmateur), c'est tout naturellement que le PIC a été retenu. Plusieurs modèles ont été testés, dont notamment les PIC 32 bits (PIC32MX), qui offrent de très bonnes performances en matière de puissance de calcul, de nombre de périphériques disponibles, de mémoire. Cependant les PIC24F leur ont été préférés, s'imposant comme un excellent compromis entre puissance et consommation, disposant d'un brochage partiellement ré-arrangeable, et d'un panel de périphériques suffisamment complets pour satisfaire les objectifs du projet (I2C, SPI, entrées analogiques et architecture 16 bits/16 MIPS).

5.4.3. Le module réseau WizNet

Du côté de la base, le Jennic JN-5139 ne prend évidemment pas nativement en charge le protocole les protocoles réseaux courant que sont TCP ou UDP, ni les couches physiques et MAC comme spécifié par le modèle OSI. Certains microcontrôleurs prennent en charge TCP/UDP, la couche MAC, mais rarement la couche PHY. Suivant les fonctionnalités proposées par le microcontrôleur, il conviendra donc de choisir le module complémentaire adapté. Dans notre cas, le JN-5139 ne dispose d'aucune interface réseau compatible avec les protocoles Ethernet, UDP ou autre. Il était donc nécessaire d'associer un module assurant la prise en charge depuis la couche réseau jusqu'à la couche physique.
WizNet, comme d'autres fabricants, proposent des modules intégrant un microcontrôleur réseau, interface entre un protocole commun et présent sur la plupart des microcontrôleurs (présentement un bus SPI). Ce module comporte donc une prise RJ45 en « sortie », et un bus SPI en « entrée », et prend en charge divers protocoles réseau (TCP IP, UDP, ARP, MAC...). WizNet fournit également un pilote succin aisé à porter sur une plateforme « microcontrôleur », ce qui a facilité l'établissement de la communication entre les deux modules via le bus SPI.

VI. Méthodologie

Le projet Interlude est un projet reçu lors d'un appel à projets lancés par l'Agence Nationale de la Recherche (ANR). S'agissant d'un projet recherche, le processus de développement est certainement plus itératif que dans un contexte industriel : le temps et la nature du projet laisse la possibilité d'essayer divers technologies, plus ou moins expérimentales, et d'en valider certaines. L'objectif de la démarche étant d'étudier les possibilités apportées au domaine par des technologies améliorées et plus performantes. Ainsi et à titre d'exemple, lors de mon stage précédent, effectué dans la même structure et au sein du même projet, l'alternative Jennic n'était pas encore préférée aux modules Xbee. Ils sont pourtant un élément décisif dans les progrès que peut représenter le résultat de ce projet. Le milieu de la recherche est aussi contribuable de ce fonctionnement : les publications permettent régulièrement de rendre publiques des recherches desquelles d'autres initiatives de recherche pourront s'inspirer.

Le projet est justement motivé par un état de l'art de l'interaction gestuelle avec des contenus musicaux. En effet, ce type d'interfaces, en quelques sortes « non-expertes », trouvent facilement des applications : dans la pédagogie, l'éveil musical, mais également la performance, ou des applications plus professionnelles et industrielles avec apprentissage rapide. Certains aspects clés, comme la dimension collaborative, sont issus de ce constat sur les interfaces gestuelles pour le contrôle musical et représente un challenge d'un point de vue recherche.
L'Ircam a en effet développé des outils de pointe en matière de suivi et reconnaissance du geste, et la mise en phase de ces outils de pointes avec des interfaces dédiées est une étape logique de la recherche sur le geste musical à laquelle se livre l'équipe interactions musicales temps réel de l'institut.
Autre aspect issu du même constat, la volonté de préserver un côté modulaire sur tout l'ensemble du système. L'Ircam est en effet à l'origine de logiciels comme Max MSP, environnement de programmation modulaire visuel, et cette volonté à travers le projet de permettre à l'utilisateur final d'adapter, d'augmenter l'interface pour la lier au mieux avec l'utilisation qu'il souhaite en faire et laisser une place à l'expérimentation n'est pas sans rappeler cette aspect modulaire et ouvert qu'a apporté le concept de Max MSP, qui a très certainement multiplié les possibilités en terme d'interaction et de temps réel. Autre raison à cette approche, la popularité grandissante des interfaces comme en proposent Arduino, qui élargissent et quelque part vulgarisent l'électronique pour la rendre accessible au plus grand nombre. Le « Do It Yourself », tout comme l'Open Source, sont des tendances actuelles au cœur des préoccupations de la recherche. Par conséquent, la porte laissée ouverte à travers l'accessoire se place dans une optique qui n'est pas étrangère à ces principes : les accessoires « breakout » (connectique « éclatée » avec des entrées analogiques standards pour recevoir des capteurs), ou la possibilité de pousser un peu plus loin et de créer, programmer son propre accessoire... En bref, le système doit être « augmentable » à différents niveaux.

D'un point de vue technologique, les progrès dans le domaine des microcontrôleurs permettent désormais de bénéficier d'une vraie puissance de calcul sans grosse consommation d'énergie, et donc la possibilité de porter/déporter des moteurs et algorithmes de traitement de données sur les plateformes embarquées. Le projet se doit également d'intégrer les technologies émergentes que sont l'Open Sound Control, les réseaux de capteurs 802.15.4, ou encore des capteurs performants comme les gyroscope 3 axes, un des objectifs étant l'optimisation de la contrainte temps réel, de façon à créer une valeur ajoutée aux alternatives existantes par l'intégration de ces technologies.

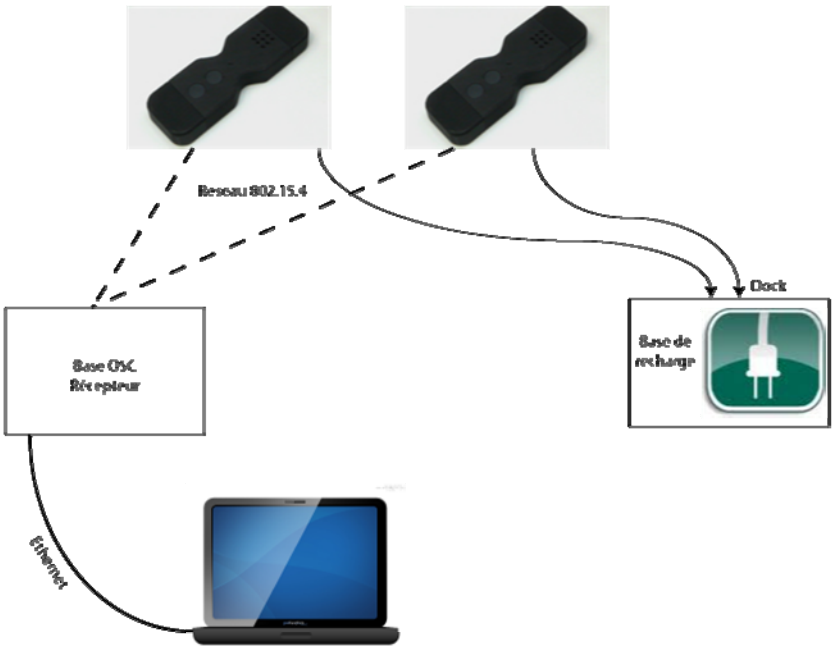
Le design n'est pas en reste puisqu'il s'agit d'un travail sur le geste. Si l'on parle d'interaction, il est donc aussi question d'ergonomie : le design pensé pour les objets d'Interlude fait l'objet d'une réflexion sur ce point mais cherche aussi à s'accorder avec le design numérique que nous connaissons, en intégrant des éléments comme le « dock » (station de recharge et de communication), la prise en main des émetteurs proche de celle d'une « wii-mote »…

Enfin, les débouchés de ce type de projet participent à une politique de valorisation des travaux de recherche, ici notamment pour la pédagogie musicale.

VII. Architecture générale et Fonctionnement du Système

7.1. Aperçu global du système

Le système comporte plusieurs éléments, dont les rôles peuvent être schématisés comme suit:



Le cahier des charges initiales prévoit une utilisation dans un contexte pédagogique en premier lieu, bien que cela n'exclue en rien d'autres types d'utilisation (performances, expériences, installations...). La composition prévue d'un système « basique » est décrite comme « mallette pédagogique ». Cette mallette comporte 8 émetteurs, une base de recharge, une base « OSC ». D'autres éléments pourront être ajoutés afin de diversifier les possibilités offertes par le système : les accessoires. Enfin un câble dit de programmation (prenant la forme d'un accessoire particulier) permet de reprogrammer les modules JENNIC et donc d'effectuer des mises à jour du logiciel enfoui (firmware).

7.2. Base OSC

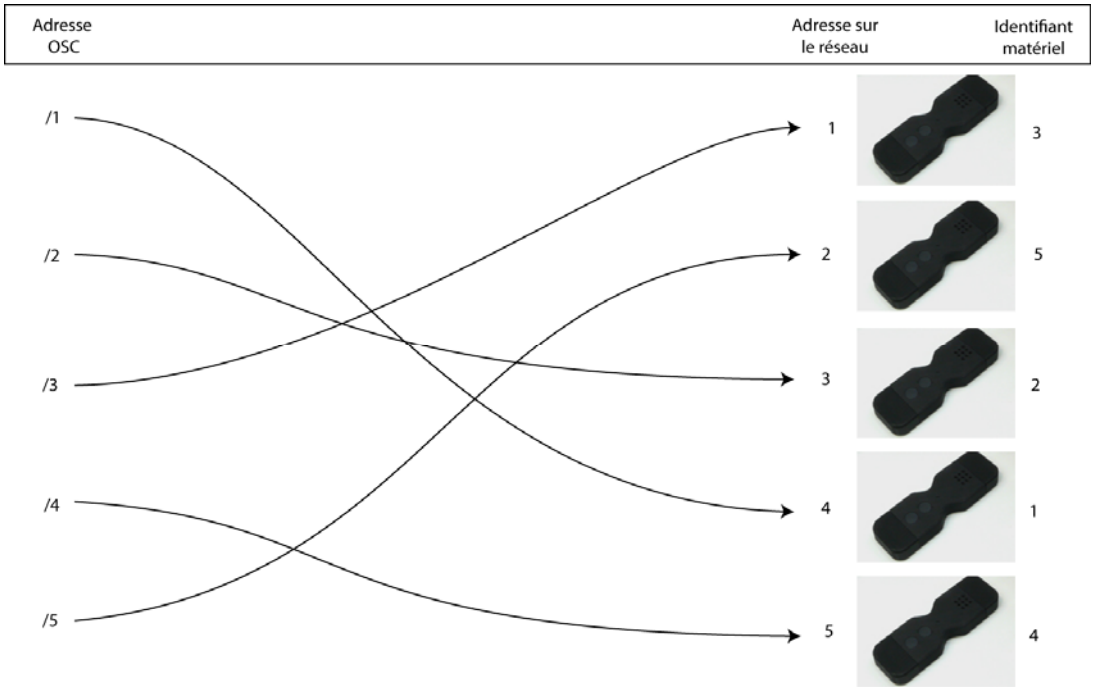
La base, qu'on pourra aussi appeler base OSC ou récepteur, est l'interface entre l'ordinateur, avec lequel elle communique via une liaison Ethernet, et le réseau sans fil. Comme décrit un peu plus loin dans ce document, le formatage choisi pour les données circulant sur le réseau sans-fil 802.15.4 est relativement proche du standard décrit par l'Open Sound Control (OSC). Cependant, les capacités d'un tel réseau n'étant évidemment les mêmes en termes de débit et de bande passante que celle d'un réseau filaire type LAN, le format est optimisé de façon à minimiser le nombre d'octets utilisés pour chaque transmission.

7.2.1. Gestion du réseau

La base OSC/récepteur joue le rôle de coordinateur du réseau sans fil. C'est donc elle qui gère les associations des « end devices », qui sont en l'occurrence les émetteurs. A sa mise sous tension, la base crée donc un réseau de façon plus ou moins dynamique : les différents canaux de la bande 2,4 GHz sont scannés de façon à sélectionner le canaux le moins parasité pour y former le réseau. L'identification du réseau se fait par le biais d'un PAN Id, fixé par le coordinateur. Ce PAN Id est un paramètre configurable par l'utilisateur (voir paragraphe « configuration »).

A sa mise sous tension, l'émetteur se met dans une phase de recherche de réseau à travers un procédé appelé le « scan », sur lequel nous reviendrons plus en détail dans le paragraphe dédié aux émetteurs.
La base, en tant que coordinateur du réseau, peut autoriser ou non l'association d'un émetteur au réseau

qu'elle gère, selon différent critère. Ainsi le réseau crée par la base prévoit certains paramètres « restrictifs », comme par exemple un nombre limite de modules connectés simultanément. Les émetteurs sont identifiés par des identifiants matériel (numéro figurant sur l'émetteur et mémorisé dans la carte électronique).
La base maintient à jour une table d'association. Cette table permet à tout moment de faire le lien entre un adresse sur le réseau, une adresse MAC et l'identifiant hardware d'un émetteur. Les émetteurs ont donc un identifiant fixe, mais leur adresse sur le réseau diffère en fonction par exemple du nombre d'émetteurs connectés au moment de l'association. La table d'association permet à l'utilisateur et à la base de dialoguer avec les émetteurs sans avoir à nécessairement connaître leur adresse sur le réseau. Le principe est similaire à une table ARP dont le rôle est de ne manipuler que des adresses IP « courtes et logiques » au lieux de manipuler des adresses MAC « longues, statiques et dédiées ».



La base dispose également d'un mécanisme de file (premier entré premier sorti ou « FIFO ») à la réception de paquets radio, afin d'éviter un phénomène d'étranglement (trop de paquets à traiter en trop peu de temps).

7.2.2. Formatage Open Sound Control (OSC)

Mis en forme : Police :(Par défaut) Tahoma, 11 pt, Italique, Anglais

Mis en forme : Anglais (Royaume-Uni)

En sa qualité d'intermédiaire entre l'ordinateur et le réseau sans fil, la base doit interpréter les messages afin d'assurer la conversion entre le domaine radio et le format de données qui lui est propre et un protocole interprétable par l'ordinateur. Comme mentionné précédemment dans ce document, le protocole Open Sound Control (OSC) a été adopté pour la réception des données du côté de l'ordinateur.

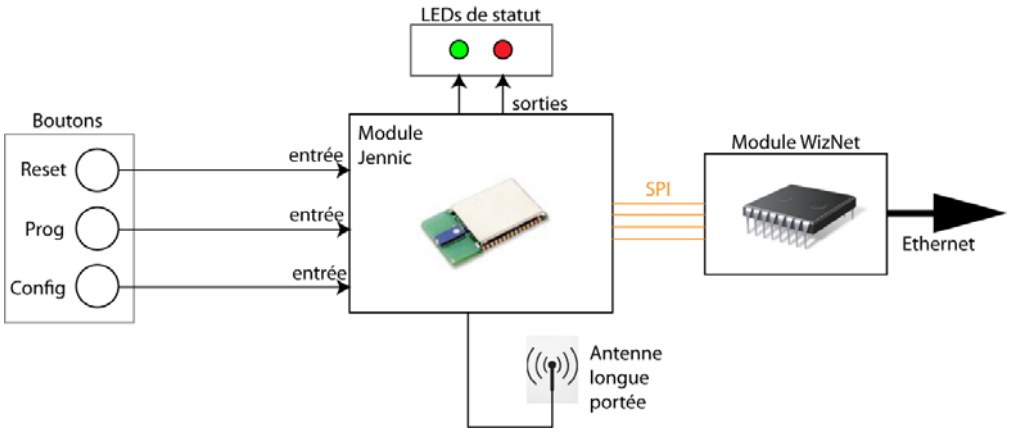
L'OSC est un standard de plus en plus populaire, que l'on peut envisager être le successeur du MIDI. Le protocole OSC défini un format de données et d'adressage et se place en couche supérieure aux protocoles réseau UDP et TCP. Il permet un adressage arborescent, la transmission de différents types de données, sans autre contrainte que la longueur maximale d'une trame UDP classique.
La base OSC vérifies l'intégrité des paquets radio et les interprète de façon à en extraire l'adresse destination au format OSC, les données ou commandes à transmettre et sous quel type/format les présenter.
D'une manière générale, l'adressage OSC est basé sur le modèle suivant:

/Destination/accessoire/service/paramètre **ou** */Destination/paramètre*

Le champ destination désigne un émetteur en particulier ou la base même. Ce format d'adresse n'est pas forcément aussi complet, l'utilisateur choisit avec quel niveau de l'arborescence il souhaite dialoguer : il peut ainsi s'adresser à un émetteur, mais aussi à un accessoire connecté à un émetteur, ou encore à un service de cet accessoire, pour « lire » ce service (c'est à dire récupérer les données traitées) ou pour modifier un de ses paramètres (fréquence d'échantillonnage par exemple).

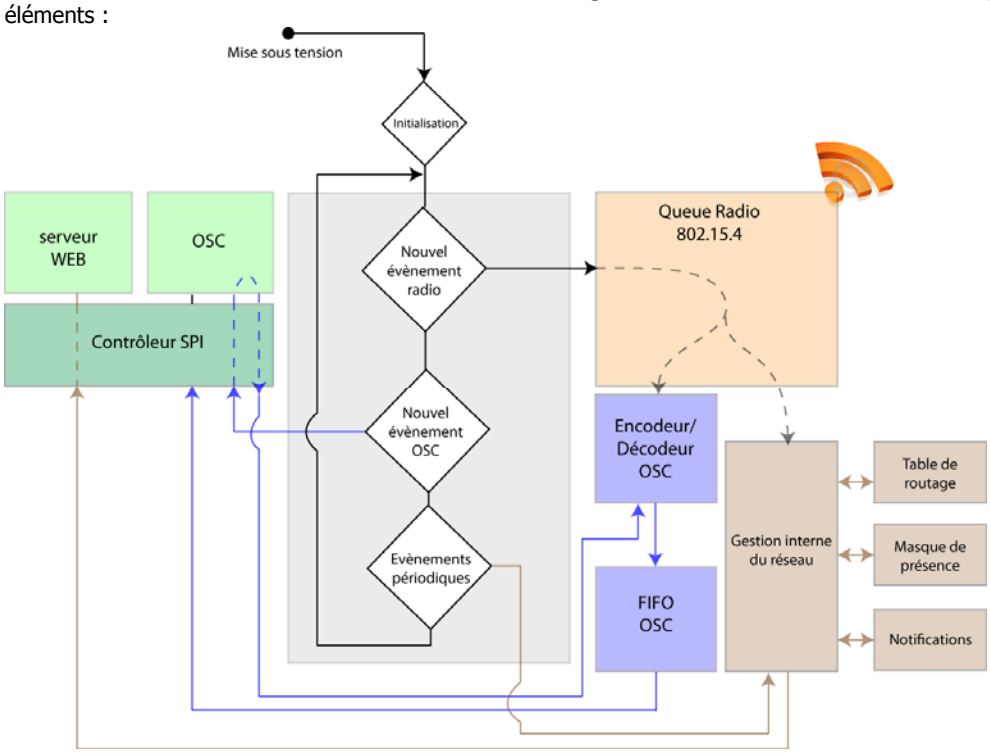
7.2.3. Architecture matérielle

Sur le schéma ci dessous figurent les principaux composants matériels ainsi que leurs relations.



7.2.4. Architecture logicielle

Le schéma ci-dessous illustre le fonctionnement du logiciel interne et les liens entre ses principaux éléments :



7.2.5. Principe de fonctionnement

La base est l'élément central du système, puisque toute communication passe par elle. Elle offre deux modes de fonctionnement très distincts : fonctionnement normal et configuration. Le mode activé par défaut lors de la mise sous tension est bien entendu le mode normal, et le mode configuration est accessible en démarrant en maintenant le bouton « config » enfoncé.

- Le mode configuration

Le mode configuration limite les fonctionnalités « courantes » (celle du mode normal) afin d'offrir à l'utilisateur une interface de configuration. Toute configuration se fait par l'intermédiaire du serveur web qui tourne en permanence sur le module Jennic. Il est ainsi possible de modifier les paramètres suivants :

- Paramètres IP : adresse IP de la base, de la machine destination, masque de sous-réseau, passerelle
- Paramètres OSC : Port utilisé pour le protocole OSC
- Paramètres réseau 802.15.4 : PAN Id du réseau créé par la base.

Toute modification de ces paramètres ne prend effet qu'après le redémarrage de la base.
Le mode configuration prévoit aussi un page sur laquelle il est possible de modifier des paramètres directement sur les émetteurs connectés au réseau. Pour cette raison, le mode configuration initialise le fonctionnement en tant que coordinateur du réseau, afin de permettre l'association d'émetteurs. Les fonctionnalités du réseau sont alors limitées à la modification de paramètres.

Les paramètres modifiables sur les émetteurs sont les suivants :

- PAN Id du réseau 802.15.4
- Identifiant Hardware de l'émetteur

Comme pour la base, toute modification de paramètre prend effet après le redémarrage de la base.
Une modification de PAN Id peut s'avérer utile dans le cas où, par exemple, deux bases seraient utilisées simultanément dans le même périmètre, afin de dissocier leurs réseaux. Il faut donc modifier le PAN Id sur la base et sur les émetteurs. Cette opération doit être contrôlée car il n'y a pas de moyen « facile » de connaître le PAN Id d'un émetteur lorsque celui-ci a été changé : il faut donc connaître et sauvegarder la configuration des émetteurs lorsqu'on la change, et veiller à ne pas faire d'erreurs de manipulation durant cette opération. Le branchement d'un port série au démarrage permet néanmoins de retrouver le PAN Id d'un émetteur « égaré », et la reprogrammation ou la mise à jour du logiciel enfoui réinitialisent le champs du PAN Id à une valeur par défaut (0xCAFE) si l'option de réinitialisation de la mémoire flash est activée. Dans le cas où l'utilisateur souhaite préserver cette mémoire, la valeur du PAN Id sera conservée.

Les identifiants matériels présents sur chaque émetteur permettent à l'utilisateur de ne pas avoir à connaître l'adresse de chaque émetteur sur le réseau. En effet, l'adressage sur le réseau fonctionne de telle sorte que l'adresse donnée à un émetteur dépend de l'ordre de connexion des émetteurs et donc aussi du nombre d'émetteurs déjà connectés. Ainsi sur le réseau 802.15.4, le même émetteur peut avoir une adresse différente d'un exercice à l'autre.

Pour permettre à l'utilisateur d'avoir des émetteurs identifiés d'une manière fixe sur le réseau, le système prévoit un identifiant matériel fixe (mais modifiable en mode configuration). Cette identifiant est donc indépendant de l'adresse réseau, sauf lorsqu'il est égal à zéro : dans ce cas, l'identifiant est mis à jour avec l'adresse sur le réseau.

- Le mode normal

Le mode normal est adapté à l'exercice dans des conditions normales. Dans ce mode de fonctionnement, la base assume toutes les fonctionnalités énoncées ci-dessus, y compris le serveur web, bien qu'il ai une fonctionnalité limité.
La base crée donc un réseau 802.15.4 après avoir initialisé ses périphériques et initialise la connexion avec le module réseau WizNet. Lorsque ces deux éléments sont en place, la base est prête à permettre l'association des émetteurs et à assurer l'interface entre le réseau 802.15.4 et le réseau ethernet/UDP/OSC.

7.2.6. Périphériques utilisés

- l'UART est activé pour permettre par exemple la récupération d'informations au démarrage ou le monitoring dans un contexte « debug »
- Le bus SPI : il est utilisé pour la liaison avec le module réseau WizNet.
- Entrées/Sorties numériques: utilisées pour les boutons et LEDs.

7.3. Émetteur

L'émetteur, répondant au doux nom de « Mo », est le terminal du côté utilisateur, et est donc l'autre élément clé du système. Il est donc lié au réseau sans fil 802.15.4 via lequel il communique avec la base et les autres modules.
L'émetteur regroupe en son sein plusieurs éléments. Le système s'articule autour d'un microcontrôleur Jennic JN-5139. Il s'agit d'un microcontrôleur dédié orienté réseau de capteurs, il comporte donc le matériel nécessaire à la mise en place et l'utilisation d'un réseau sans fil 802.15.4, mais également un panel complet de périphériques classiques que l'on retrouve sur la plupart des microcontrôleurs du marché : convertisseur analogique/numérique et inversement, port série, bus I2C et SPI, mémoire Flash, entrées sorties standards...

Il constitue donc un processeur particulièrement adapté aux applications organisées autour d'un réseau sans fil mais pour lesquelles une configuration particulière et des possibilités de personnalisation sont souhaitables, et offrent une alternative plus qu'intéressante aux systèmes associant un microcontrôleur classique avec un contrôleur réseau.

D'un point de vue interface utilisateur, les contrôles présents sur le modules sont sommaires : 2 boutons standards, une matrice 3x3 de LEDs destinée à afficher des pictogrammes concernant le fonctionnement logiciel ou à générer des stimuli visuels, ainsi qu'une LED bicolore centrale renseignant l'utilisateur sur l'état matériel du module (charge de la batterie).
Le module est alimenté par une batterie rechargeable lithium-ion 3,7V, gérée par un contrôleur de charge.
Chaque émetteur dispose de 2 ports sur lesquels peuvent venir se brancher des accessoires. Ces ports permettent la communication entre un accessoire et le microcontrôleur Jennic via un bus I2C (voir section choix technologiques).
Il n'est cependant pas nécessaire de brancher un accessoire pour obtenir des données capteurs des émetteurs : ils sont équipés d'une centrale inertielle à 6 degrés de liberté (accéléromètre 3 axes, gyroscope 3 axes). Ces capteurs sont numériques et par conséquent reliés au même bus I2C que les éventuels accessoires. Ils sont vus comme un accessoire par l'utilisateur et l'ensemble du système, répondant au même principe de services : la seule différence est donc que cet accessoire est disponible sans aucun branchement supplémentaire.

7.3.1. Système de messagerie du réseau

Un protocole de messagerie évolué est évidemment nécessaire au bon fonctionnement du réseau. Par messagerie, on entend un mécanisme d'encapsulation des données circulant sur le réseau sous forme de trames permettant d'avoir rapidement accès aux informations recherchées.

Toutes les trames circulant sur le réseau respectent donc des règles d'encapsulation. Cela est utile pour plusieurs raisons:
Le récepteur d'un message peut vérifier l'intégrité de la trame qu'il reçoit. Le premier comme le dernier octet est aisément identifiable, ce qui permet donc de délimiter la trame, et de connaître la position des informations utiles. Un octet renseignant sur la taille du champ « données » par exemple, permet de connaître immédiatement la taille totale du message, étant donné que ce champ est la seule partie du paquet dont la taille est variable.

Un message standard est donc constitué comme le datagramme suivant :

Délimiteur	Taille	Identifiant de trame	Identifiant de commande	Données	Checksum

Ce format permet aussi de standardiser les échanges afin de les rendre « convertibles » par la base en OSC, ou inversement convertir des messages en OSC en messages réseau.

Chaque module contient une liste de valeur que peuvent respectivement prendre les champs identifiants.
Suivant les valeurs des identifiants des trames et de commandes, les modules récepteurs pourront interpréter le message différemment.

7.3.2. Fonctionnement de la couche réseau 802.15.4 sur les modules Jennic

Du côté du réseau 802.15.4, le module Jennic JN-5139 prend en charge les couches MAC et PHY, et propose une API au niveau de la couche réseau à partir de laquelle il est possible créer la partie de réseau de l'application.
A la différence des périphériques matériels et de l'API qui les accompagne, la pile réseau ne propose pas de fonctionnement sous interruptions : une interruptions ne peut être générée, par exemple, par la réception d'une trame. Les évènements ayant lieu sur le réseau, tels que les envois ou associations, devront donc être placés dans une queue, qui sera traitée périodiquement à chaque « tour » dans la boucle principale.

L'API Jennic propose deux principaux mécanismes de communication entre l'application (MAC user) et la couche MAC (MAC layer) , qui décrivent deux types d'interactions entre ces deux couches : initié par l'application à destination de la couche MAC et inversement, de la couche MAC vers l'application.
Dans les deux cas, le mécanisme est basé sur un principe d' « appel/rappel » (call/callback). Ainsi lorsqu'une requête est formulée à l'attention de la couche MAC (par exemple), il lui est aussi passé un pointeur vers une structure informant sur l'état de la requête. Ce pointeur pourra être analysé dans la fonction de traitement des évènements de la queue, exécutée à chaque itération de la boucle principale (« polling »).

- Par exemple : Requête d'envoi de message passée à la couche MAC depuis l'application, l'appel de la fonction appropriée pousse la requête dans la queue réseau et enregistre un pointeur vers une structure détaillant l'état de la requête. Lorsque cette requête sera traitée, la structure sera mise à jour avec le résultat de ce traitement et la couche MAC génèrera une confirmation à ce propos, qui pourra être traitée par une fonction de prise en charge des confirmations en fonction du contenu de la structure.

A l'inverse, la couche MAC peut également être à l'origine d'un échange en formulant une indication, nécessitant (ou pas) une réponse de l'application.
Les échanges peuvent de ce fait être synchrones ou asynchrones : l'application peut être bloquée jusqu'à récupérer une confirmation de la part de la couche MAC.

7.3.3. Principe de fonctionnement

- interface utilisateur embarquée sur l'émetteur

Les contrôles qu'offre l'émetteur sont volontairement simplistes : il ne comporte que 2 boutons. Ces deux boutons auront donc naturellement des modalités d'interaction multiples. En effet ils contrôlent l'extinction et l'allumage, puisqu'ils sont reliés au circuit d'alimentation de l'émetteur ainsi qu'au module Jennic JN-5139. Les procédures d'allumage et l'extinction sont déclenchées par une certaine configuration des boutons.
Pendant toute la durée du fonctionnement, ces mêmes boutons pourront être utilisés comme contrôleur : une pression sur l'un d'eux déclenchera l'envoi d'un message notifiant cette action à la base, qui sera bien entendu transféré jusqu'à l'utilisateur final sur la machine hôte.

Deux niveaux d'interfaces « graphiques » sont présents directement sur l'émetteur : la matrice de LEDs (3x3) informera l'utilisateur de l'état de l'émetteur. La LED centrale, baptisée le « nombril », délivre une information sur l'état matériel de l'émetteur, à savoir principalement le niveau de charge de l'émetteur. Cette LED est bicolore afin de pouvoir afficher différents niveaux d'état de la charge de la batterie.

Les processus de mise sous tension et d'extinction sont déclenchés par la pression simultanée des deux boutons. Cette pression doit être maintenue environ 2 secondes, afin de confirmer l'intention de l'utilisateur d'engager un de ces processus.

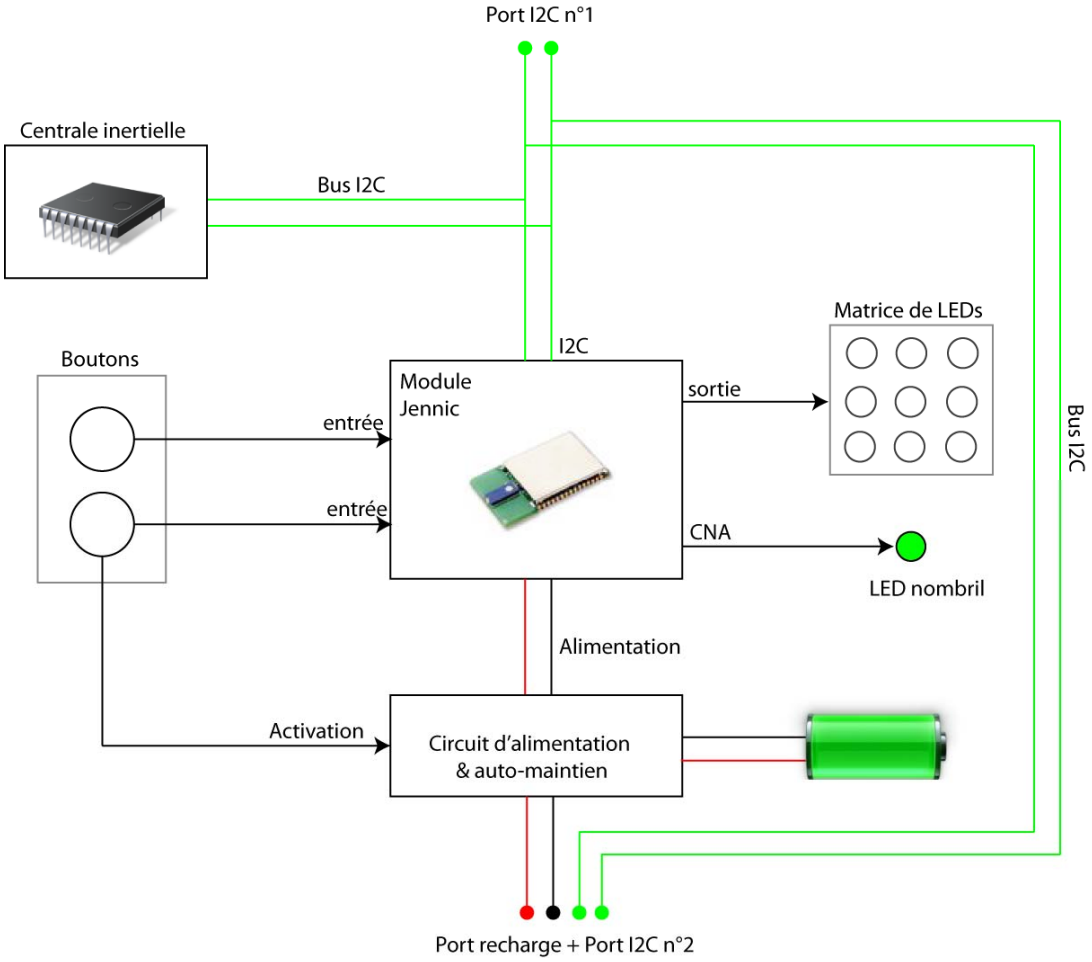
- Contrôle de l'émetteur durant l'exercice

Une fois l'émetteur allumé et hors les pressions sur les 2 boutons, toute forme de d'interaction ne pourra se faire que depuis la machine hôte. C'est un choix dans la continuité de la logique des services : Embarquer un menu directement sur l'émetteur serait, d'un point de vue design, tout à fait anti ergonomique. Pour cette raison, la découverte de services, la lecture d'un service ou la modification d'un paramètre sont des actions

forcément effectués depuis l'ordinateur.

7.3.4. Composition matérielle de l'émetteur

Les principaux composants embarqués ainsi que les liens entre eux sont illustrés par la figure ci dessous:



7.3.5. Architecture logicielle de l'émetteur

Une majorité de périphériques sont configurés pour fonctionner sous interruptions.

Supprimé : maximum

On peut distinguer plusieurs axes nécessaires au bon fonctionnement d'un module émetteur.

- L'axe réseau
- L'axe interface matérielle
- L'axe affichage
- L'axe énergie
- L'axe accessoire

Comme décrit plus haut, l'interface matérielle permettant à l'utilisateur d'interagir avec l'émetteur est simpliste. C'est également le point d'entrée du programme interne des modules Jennic, puisque la pression sur le bouton supérieur active la broche « chip enable » (activation du composant) du régulateur. Autrement dit, cette pression permet de mettre le Jennic JN-5139 sous tension. Cependant le mécanisme d'auto maintien, comme décrit ci-dessus n'est pas activé tant que la sortie du JN-5139 reliée au régulateur n'est pas mise à l'état haut (3,3V logique). Un compte à rebours est donc déclenché lorsque les 2 boutons sont enfoncés. Si la durée d'enfoncement de ces deux boutons excède 2 secondes, l'auto maintien est activé par un passage à l'état haut sur la broche du Jennic JN-5139 reliée au régulateur. Lorsque cette sortie est à l'état haut, la pression sur le bouton supérieur n'est plus nécessaire à l'alimentation du module JN-5139.

Conjointement à l'activation de l'auto maintien, le microcontrôleur initialise la plupart de ses périphériques, ainsi que la couche réseau. Il peut donc, une fois sorti de la boucle d'auto-maintien, initialiser le processus de connexion propre au réseau PAN 802.15.4.

L'utilisation d'un ou plusieurs émetteur(s) requiert bien évidemment une installation du système complète, mais en premier lieu une connexion au réseau 802.15.4 (créé par une base). Si ces conditions sont rassemblées, l'émetteur pourra s'associer au réseau avec succès. En effet, la phase de mise sous tension avec auto-maintien est forcément suivie d'une phase durant laquelle l'émetteur cherche à s'associer à un coordinateur.

- Association au réseau

Lors de cette opération, l'émetteur est à la recherche d'un PAN Id correspondant par défaut à celui de la base de la même mallette pédagogique. Le PAN Id peut éventuellement reconfiguré, nous reviendrons plus en détail sur les possibilités de reconfiguration et leurs mécanismes de fonctionnement plus loin dans ce document. La recherche du réseau, communément appelé « scan » s'organise comme suit :
L'émetteur scanne les différents canaux de la bande 2,4 GHz jusqu'à y trouver un coordinateur dont le PAN Id correspond à celui qu'il recherche.
Cette phase de scan s'ensuit de l'envoi d'une requête au coordinateur. Ce dernier peut refuser l'association si, par exemple, un nombre trop important d'émetteurs sont déjà sur son réseau. Si l'association est refusée, l'utilisateur en est notifié par une animation affichée sur la matrice de LEDs, si au contraire elle est acceptée, l'émetteur en est notifié et reçoit également une trame contenant la table d'adressage maintenue à jour par le coordinateur. La présence de cette table chez l'émetteur peut s'avérer utile voire indispensable si certains services prévoit une interaction entre deux modules particuliers. Bien que ce scénarios d'utilisation n'ai pas été implémenté à ce jour, il reste tout à fait envisageable de créer un service ou deux modules seraient synchronisés par une connexion direct entre eux, sans passer par la base : les émetteurs auraient alors besoin de connaître le adresses et identifiants des émetteurs concernés sur le réseau.
Pour cette raison, la table est mise à jour à chaque nouvelle association de module.

- Découverte et utilisation des services

Une fois l'association terminée et effective, une découverte de services est engagé automatiquement. Elle permet donc d'une part de notifier la base et la machine hôte de la présence de l'émetteur, et d'autre part de faire connaître les services qu'il propose. Dans le cas où aucun accessoire n'est connecté, seul l'accessoire « interne » (la centrale inertielle à 6 degrés de liberté) apparaîtra.
La découverte de service peut être engagée manuellement depuis la machine hôte si le besoin se présente.

La lecture d'un service ne peut être engagée que si l'accessoire concerné a préalablement été découvert. De cette manière, l'émetteur connaît l'accessoire qui lui est connecté, c'est à dire qu'il dispose de toutes les informations qui sont nécessaire à l'établissement d'une communication, ainsi que le type de donnée (littéralement : flottants, entiers...) qu'il peut récupérer de sa part.

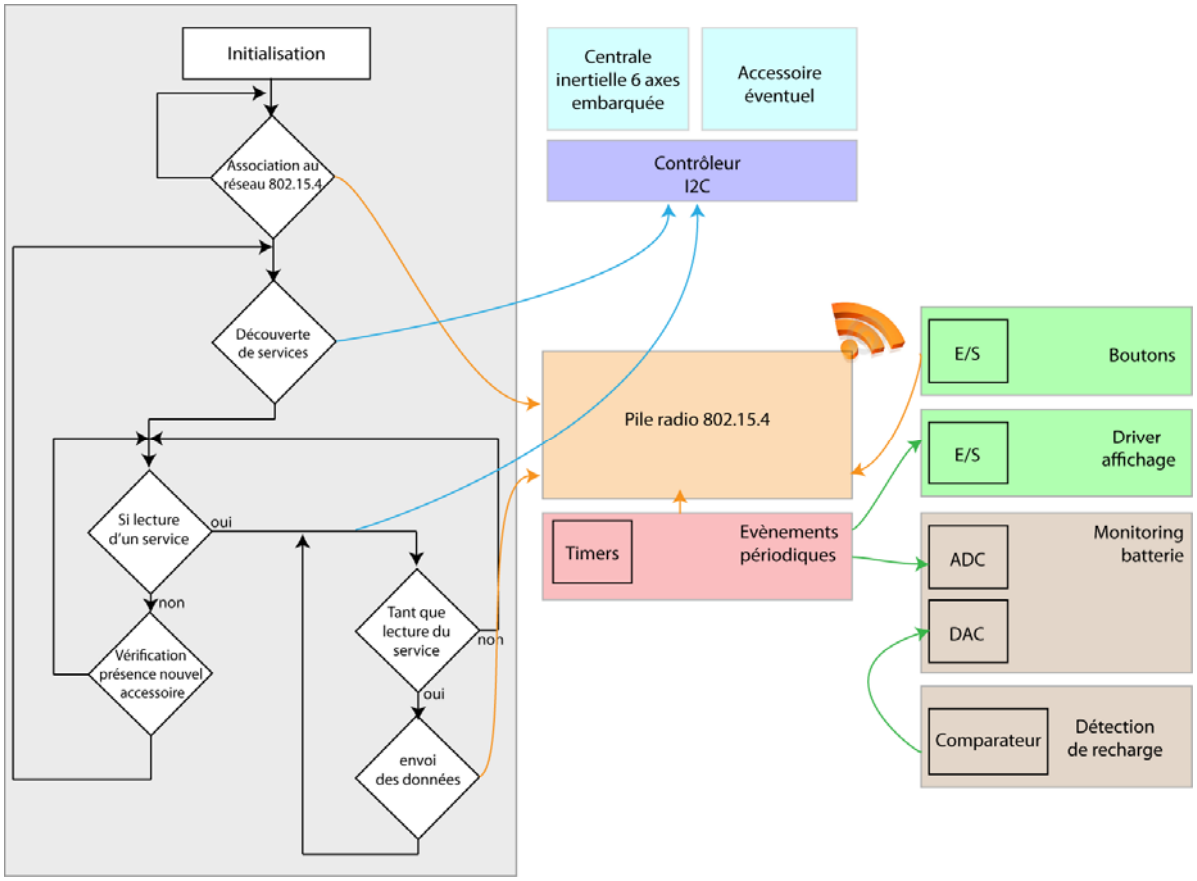
Supprimé : d'une

La lecture ne peut ensuite se faire que depuis l'ordinateur : il n'est pas possible d'activer un service manuellement sur l'émetteur. Comme nous l'avons déjà évoqué, il en est de même pour la plupart des contrôles. De ce fait, le respect du format d'adressage OSC est indispensable pour établir la communication avec un émetteur, un accessoire (interne ou externe), un service (voir section formatage Open Sound Control). Chaque trame OSC valide contient un identifiant en tant qu'argument, qui identifie l'action à mener. La modification de paramètre, la découverte ou lecture de services font donc l'objet d'identifiants spécifiques.

Une fois l'émetteur associé à un réseau et ses services découverts, les conditions sont réunies pour débuter l'exercice.

- Diagramme fonctionnel

Le diagramme ci dessous, volontairement simpliste, illustre le fonctionnement normal d'un émetteur, et fait apparaître les modules principaux et les relations entre eux.



7.3.6. Périphériques utilisés

Parmi les périphériques utilisés par l'émetteur, on retrouve :

- Le port I2C : essentiel à l'utilisation de l'émetteur, son fonctionnement est nécessaire et central.
- Les entrées/sorties classiques : certaines d'entre elles sont utilisées pour différentes chose : piloter la matrice de LEDs, maintenir le circuit d'alimentation...
- Le convertisseur analogique numérique : il est utilisé (ponctuellement mais périodiquement) pour déterminer le niveau de charges ainsi que les différentes situations possibles à ce niveau (recharge, batterie, charge complète...).
- Le convertisseur numérique analogique : il est utilisé pour piloter la LED bicolore « nombril »
- UART : utilisé pour les éventuelles situations de debug ou de mise à jour du logiciel enfoui (firmware).
- Comparateur : il est utilisé pour détecter la connexion à la base (dock) de recharge.
- Les timers du JN-5139 sont utilisés pour de nombreuses applications : timing de l'affichage, déclenchement d'évènement réseau, déclenchement de conversion analogique numérique...

VIII. Description Détaillée des développements

8.1. Prise en main et développement de la partie « réseau »

8.1.1. Gestion du réseau

Le choix des modules Jennic est la première et certainement une des plus importantes nouveautés du projet au début de ce stage. Un prototype d'émetteur a été réalisé (version 0.1), comportant les éléments principaux de l'émetteur (module Jennic, boutons et Leds, régulateur et circuit d'auto-maintien). L'Ircam dispose également d'un kit de développement Jennic composé de plusieurs cartes de développement. Ces cartes permettent une première prise en main des modules Jennic, ainsi que de jouer le rôle de coordinateur/base. Les premières versions du système ont donc été développées sur ce modèle.

Le besoin d'une dissociation entre l'adressage « réseau » et l'adressage utilisateur est apparu assez vite : l'utilisateur doit pouvoir communiquer avec les émetteurs sans connaître leur adresse sur le réseau. La table de routage est imaginée pour faire l'interface entre le monde de l'utilisateur et le domaine réseau.
L'adressage réseau est en effet simpliste : le premier module connecté se verra attribué la première adresse disponible dans l'ordre croissant. Il est donc nécessaire dans cette situation de permettre à l'utilisateur de pouvoir communiquer avec un module via une adresse unique d'un exercice à l'autre. La table de routage « convertit » donc les identifiants matériels en adresse réseau et inversement.

Divers mécanismes de gestion du réseau ont été implémentés, en particulier du côté de la base vu qu'elle est l'élément central du réseau. Avant le système de messagerie, il a été nécessaire de mettre en place d'autres fonctionnalités réseau, dont le maintien de la table de routage au cours des associations et de l'exercice. Parmi les fonctionnalités proposées, un émetteur pourra conserver sa « place » dans la table de routage pendant une durée déterminée si pour une raison quelconque (dis-association, extinction par exemple) il est amené à quitter le réseau. Ainsi, si l'émetteur est formule à nouveau une demande d'association durant le temps imparti, il récupèrera sa position dans la table de routage, et donc son « ancienne » adresse sur le réseau.
Ce mécanisme permet de conserver une table de routage à jour au niveau de la présence des modules, et ainsi de connaître le nombre de modules connectés et donc le nombre de places disponibles sur le réseau (étant donné que le nombre d'émetteurs connectés simultanément connectés est limité). En effet l'extinction d'un émetteur d'une manière matérielle (hard reset par exemple) dans un réseau « basique » ne laisse pas le temps nécessaire à une procédure de dis-association, et dans ce cas l'émetteur conserve, bien qu'il ne soit plus connecté, son adresse sur le réseau.
Le mécanisme est simple dans son implémentation : tous les émetteurs associés envoient à intervalles périodiques des trames très courtes notifiant de leur présence sur le réseau. A la réception de ces trames, la base met à jour un champs de bits représentant les émetteurs présents sur le réseau (masque de présence). Ce champ de bit est comparé périodiquement avec sa valeur précédente de manière à identifier les éventuels émetteurs manquants et à lancer un décompte au delà duquel leurs adresses seront libérées.

- Association au réseau

L'association des modules Jennic au réseau est plutôt standard : la topologie du réseau étant orientée «en étoile » autour du coordinateur, celui-ci va autoriser les associations dans la limite du nombre de « places disponibles » sur le réseau. Bien entendu, il faut que les émetteurs partagent les mêmes réglages de configuration au niveau du réseau (principalement : le PAN id).

- Système de messagerie

Le système de messagerie a également été implémenté assez tôt, puisqu'il s'agit d'une fonctionnalité vitale du réseau.
Il n'existe pas de protocole « standard » permettant de standardiser le format des trames circulant sur un réseau 802.15.4. Des librairies sont disponibles, comme micro-OSC, et permettent de faire transiter des trames au format conforme au protocole OSC sur un réseau 802.15.4 composé de microcontrôleurs. Cependant, ces ports de protocoles sur microcontrôleur ont d'évidentes limites. En effet le débit sur les réseaux sans-fils type 802.15.4 est très largement inférieur au réseau filaires type LAN, sur lesquels est

généralement utilisé le protocole OSC. De ce fait, l'utilisation du protocole OSC sur l'ensemble du système rallonge considérablement la longueur des trames sans pour autant contenir plus d'informations « utiles » qu'un autre protocole.

Supprimé : ans

C'est pour cette raison qu'un système de messagerie indépendant a été développé afin d'assurer la transmission des informations sur le réseau de la manière la plus efficace possible. Le réseau 802.15.4 mis en place a ici une architecture dont les principaux éléments et niveaux sont clairement définis. Il n'est donc pas utile de conserver une ouverture comme le propose l'OSC au niveau de l'adressage.
Les besoins du système de messagerie permettent donc de limiter le nombre d'octets destinés à définir la qualité de l'information contenue dans une trame.

Toute trame doit au demeurant être définie en « longueur », et comporte donc un identifiant de début de trame (délimiteur), un champ renseignant la longueur du champs donnée, et un octet de checksum, délimitant la fin de la trame.
Les adresses du destinataire et de la source sont récupérables par le protocole 802.15.4, il n'est donc pas nécessaire de les faire apparaître dans les trames. Par contre, les trames venant d'accessoires doivent pouvoir être identifiées.
Dans cet optique, chaque trame comporte deux octets identifiants le type de trame dont il s'agit : l'identifiant de trame et l'identifiant de commande. Les combinaisons permises par 2 octets sont largement suffisantes pour définir toutes les trames, et les 2 octets ne sont pas forcément « utiles » pour toute commande. Par exemple, la découverte de service est déclenchée par l'émetteur sur réception d'un certain identifiant dans le champ « identifiant de trame », par contre dans le cas de la lecture d'un service l'identifiant de trame contiendra une commande de lecture, l'identifiant de commande informera sur l'accessoire concerné, et le service concerné apparaîtra dans les données transmises par la trame.

8.2. Implémentation du protocole Open Sound Control (OSC)

Si des prototypes d'émetteurs ont été réalisés assez tôt par l'équipe, les cartes de développement du kit Jennic ont continué à jouer le rôle de base pendant quelques mois en l'attente du premier design de la base. Ces cartes ne prévoient pas les composants choisis pour la base, comme notamment le module WizNet, qui assure la liaison ethernet avec l'ordinateur hôte. De ce fait, les premières implémentations de la pile OSC ont utilisé le port série des cartes de développement. Ceci n'étant finalement pas un problème majeur puisque la liaison avec le module WizNet dans les versions futures de la base est une liaison SPI, bus série également. Le mode de fonctionnement est donc comparable, hormis la configuration du WizNet, puisque le module Jennic n'est de toute façon pas destiné à gérer les couches UDP, TCP/IP, Ethernet, MAC etc.

La pile OSC est bidirectionnelle : elle prévoit un décodeur, qui décode les messages OSC et les interprète pour formuler un message sur le réseau 802.15.4, ainsi qu'un encodeur, qui à l'inverse interprète les messages réseau pour en extraire le message OSC approprié.
Afin d'optimiser le temps de traitement et l'utilisation de la mémoire, le champ d'adresse du protocole est aussi redéfini dans une optique un peu plus « statique »: Le format d'adressage, tel que décrit plus haut dans les spécifications OSC, prévoit un nombre limite de caractère par champs, les champs étant ici les parties de l'adresse désignant les éléments comme l'émetteur destinataire, l'accessoire, le service... Cela simplifie l'analyse de la chaîne de caractère réservée à l'adresse et permet d'identifier clairement et rapidement la destination du message.
La couche OSC implémentée sur la base ne prévoit pas la prise en charge de tous les types prévu initialement par le protocole (entiers signés, non signés, flottants, couleurs RVB, caractères...) puisqu'il n'en a pas l'utilité. Dans un premier temps, le seul format pris en charge est l'entier non signé. En effet les besoins initiaux du cahier des charges étaient relativement sommaires : permettre la transmission en continu (streaming) de données capteurs. Ces données capteurs étant généralement numérisées sur 10 bits, au format d'un entier non signé, les autres types étaient superflus.
Ce choix sera revu à l'arrivée des traitements effectués sur les accessoires, pour lesquels d'autres types de données sont plus appropriés. Le type de données « en sortie » peut varier d'un service à un autre sur un même accessoire, et l'on peut donc avoir des entiers signés, non signés, ou des flottants.
Enfin l'implémentation de la découverte de services, bien qu'étant un cas spécifique, imposera aussi la prise en charge des chaînes de caractères dans le but de rendre plus lisible le message OSC correspondant à une découverte de services.

8.3. Développement de la base commune aux accessoires

L'émetteur dans sa première version n'embarquant aucun capteur, le besoin d'intégrer un premier accessoire est apparu comme évident pour valider différents aspects, comme l'architecture en service ou l'utilisation du bus I2C. Les choix technologiques au sujet des accessoires ont déjà été évoqués : l'implantation d'un microcontrôleur en tant qu'interface entre le bus I2C et les capteurs s'est imposé.
Un design d'accessoire a donc été réalisé dans le but de valider ce choix et de proposer une plateforme de développement pour les accessoires. Ces derniers doivent en effet avoir une base commune, notamment au niveau de l'interface I2C.

Le design réalisé est donc destiné à être branché sur les émetteurs, et comporte donc 2 ports I2C (un premier pour la connexion avec l'émetteur et un deuxième pour accueillir différentes éventualités : capteurs numériques, branchement d'un deuxième accessoire...), un port série pour le développement, 7 entrées analogiques (reliées au convertisseur analogique numérique), ainsi qu'un port ICSP, nécessaire à la programmation *in situ* des microcontrôleurs.

Mis en forme :

8.3.1. Configuration du bus I2C

La première étape du développement consiste à mettre en place la base supposée être commune aux accessoires, à savoir outre la configuration du microcontrôleur, la configuration du bus I2C. Malheureusement, le module Jennic ne permet pas de fonctionner en mode esclave sur le bus I2C, ce qui aurait permis au microcontrôleur de l'accessoire de prendre la main de façon autonome sur le bus pour transmettre des données. Ce mode de fonctionnement aurait simplifié le mode opératoire sur des services dont les données ne sont pas sensées être transmises en continu : les détections de mouvement par exemple.
Le module Jennic est donc configuré en tant que maître du bus, et le microcontrôleur de l'accessoire est esclave. Le maître, qui fournit l'horloge sur le bus, est configuré pour fonctionner à vitesse d'horloge maximale (400 kbits/s). L'adresse d'un accessoire sur le bus est en fait celle du microcontrôleur. Elle est donc identique d'un accessoire à l'autre mais peut être modifiée en cours d'exercice si besoin.

8.3.2. Fonctionnement du bus I2C

Le fonctionnement du bus est basé sur un principe de commandes, de manière analogue aux commandes que l'on peut retrouver sur le réseau. Certaines commandes ont d'ailleurs les mêmes identifiants à tous les niveaux : OSC, réseau, I2C.
A chaque événement, le module Jennic transmet sur le bus un octet de commande, par exemple : requête de découverte de service, activation et lecture d'un service, modification d'un paramètre. A la réception de cette commande, le microcontrôleur peut se préparer à l'opération qu'elle décrit : si par exemple la commande correspond à une découverte de service, le microcontrôleur sait que l'émetteur est sur le point d'engager une lecture, et remplit donc un « tampon » (buffer) d'octets contenant toutes les informations relatives aux services, et ce dans un ordre défini de façon à ce que l'émetteur puisse parcourir ces informations sans erreur. Ce tampon est ensuite vidé d'un octet à chaque interruption I2C, jusqu'à la condition « stop », insérée sur la ligne par le maître.

8.3.3. Découverte de services

Comme présenté précédemment, les accessoires proposent des données sous la forme de services. A la connexion d'un accessoire, il est indispensable de connaître les services qu'il propose. La découverte de services renseigne donc sur :

- L'identité de l'accessoire
- Le nombre de services proposés
- L'identité de chaque service
- Le type de donnée proposé par chaque service
- Le nombre de données transmises lors de la lecture d'un service
- Les paramètres de chaque service (nom et valeur par défaut)

Toutes ces informations sont relayées jusqu'à l'utilisateur final sous la forme d'une trame OSC descriptive. Un message OSC indépendant est généré pour chaque service d'un accessoire. Les informations figurant dans ces trames de découverte de services définissent également la manière dont on y accédera par la suite, car les identifiants des services ou des accessoires découverts doivent apparaître dans les adresses OSC des trames de lecture, de configuration...

Comme évoqué un peu plus haut (fonctionnement du bus I2C), tous les échanges sur le bus I2C se font octet par octet, conformément aux spécifications du bus I2C, créé par Philips dans les années 1990. Il est donc nécessaire de définir un format de transmission pour les différents niveaux d'informations que peut contenir une trame I2C : d'une part pour chaque type de donnée transmise, il faut assurer la cohérence de l'ordre de réception (par exemple, l'octet de poids fort doit être vu comme tel par l'émetteur comme le récepteur), d'autre part l'ordre de transmission des informations doit être défini (par exemple, l'identifiant de l'accessoire ne doit pas être confondu avec celui d'un service).

8.3.4. Lecture d'un service

Deux types de services doivent être distingués, puisqu'ils seront traités différemment : les services dont la lecture entraîne la transmission en flux tendu (streaming) et les services dont la lecture a pour but de signifier un événement spécifique.
Le premier cas est le plus simple à gérer : la réception d'une commande de lecture de service entraîne une lecture I2C du service, qui est répétée à intervalles de temps réguliers tant qu'une commande d'arrêt n'est pas reçue. De ce fait, le maître (module Jennic) interroge par « polling » dés que possible l'accessoire pour extraire les nouvelles données.

Supprimé : (polling)

Le second cas doit contourner le problème du « maître unique » du bus. En effet, le fait que le microcontrôleur de l'accessoire ne puisse pas être maître sur le bus complexifie les choses à ce niveau : lorsqu'un événement spécifié par le service a lieu, le microcontrôleur ne peut pas simplement le signaler en prenant la main sur le bus pour transmettre les données appropriées à l'émetteur. La solution choisie est alors de reprendre le même principe d'interrogation (polling) sur le bus I2C de l'accessoire par l'émetteur, mais en ajoutant à chaque lecture I2C un champ indiquant à l'émetteur si cette donnée doit être relayée à l'utilisateur final, et donc transférée sur le réseau. Ce mode d'opération ne remplace pas les avantages d'un prise de main autonome sur le bus par le microcontrôleur, et le fonctionnement en interrogation (polling) reste gourmand en énergie, cependant ce procédé permet de préserver la bande passante du réseau et de la laisser disponible aux autres émetteurs tant qu'aucun événement n'est détecté.

Le fonctionnement sous interruptions du périphérique I2C du microcontrôleur permet de ne pas fonctionner en interrogation (polling) du côté de l'accessoire : dès qu'un événement est déclenché sur le bus I2C par le maître, le microcontrôleur entre en interruption et traite cet événement immédiatement. Ceci évite de retarder le traitement des échanges I2C, qui pourrait aboutir à des phénomènes d'étirement d'horloge par l'esclave (clock stretching) qui ne sont pas pris en charge par le port I2C du module Jennic et pourrait être vus à tort comme une perte de l'arbitrage du bus.
Les mêmes précautions que pour la découverte de service s'impose quand au formatage des données : il faut respecter un ordre défini.

8.4. Accessoire 5 axes

Comme nous l'avons mentionné au sujet de la carte réalisée en tant que plateforme de développement pour les accessoires, elle comporte 7 entrées câblées vers le convertisseur analogique numérique du PIC. Ce périphérique a des caractéristiques plutôt standard pour la gamme de microcontrôleurs : dynamique de 10 bits, 12 canaux en entrées, vitesse de conversion dérivée de l'horloge interne permettant des fréquences d'échantillonnage au voisinage de 500 kHz.
L'Ircam a en sa possession un certain nombre de modules sans fil basé sur les modules *Xbee* de Digi, qui offrent jusqu'à 5 canaux de conversion analogique numérique. L'Ircam a donc fait l'acquisition de plusieurs modules fabriqués par Sparkfun regroupant une centrale inertielle à 5 degrés de liberté (5 DOF IMU pour « Degrees Of Freedom Inertial Measurement Unit»), comprenant un accéléromètre 3 axes et un gyroscope 2 axes analogiques. Ces modules se prêtent aussi à être associés au prototype d'accessoire, par leur caractère analogique.
Le premier accessoire réalisé associe donc un microcontrôleur à une centrale 5 axes. Bien que les versions suivantes de l'émetteur prévoient une centrale 6 axes intégrée, l'accessoire « centrale inertielle » fait également partie du cahier des charges pour plusieurs raison :

- l'accessoire peut être déporté à un autre endroit via une liaison filaire
- la puissance de calcul du PIC est entièrement dédiée aux éventuels traitements du signal, ce qui n'est pas le cas du Jennic qui doit gérer une pile radio 802.15.4 et doit répondre à d'importantes contraintes de temps dans ce contexte.

D'une manière générale, le programme interne du microcontrôleur opère trois tâches principales : numérisation des capteurs, traitement des données capteurs, et transfert sur le bus I2C.

8.4.1. Numérisation des capteurs

Les 5 sorties des capteurs sont branchées dans les entrées du convertisseur du PIC, qui les échantillonne à intervalles réguliers : la conversion est déclenchée par un timer.
L'utilisation du timer a un double intérêt. Elle permet de régler la fréquence d'échantillonnage avec précision et de rendre accessible ce paramètre à l'extérieur de l'accessoire, mais aussi d'avoir une base temporelle précise pour tous les traitements éventuellement effectués par un service pour les données échantillonnées.
Chaque service comporte un paramètre décrivant le nombre de canaux à convertir, ce paramètre est donc modifiable.

8.4.2. Analyse et traitement des données

La première étape consistait à valider le principe de la lecture de services avec tous les maillons de la chaîne de transmission : OSC, réseau sans fil 802.15.4, I2C. Aucune analyse des données n'étant nécessaire pour cette validation, seul le transfert en flux tendu (streaming) des données devait être fonctionnel. Le « streaming » des données est aussi un outil important pour les travaux de l'Ircam indépendants du projet Interlude : nombre d'applications, actuellement basées sur les modules Xbee, fonctionnent avec des données en flux tendu en entrée.
Le « streaming » des données capteurs fait l'objet d'un service dédié pour chaque accessoire. Un fois la conversion configurée, l'opération est assez simple : il faut transférer la mémoire tampon (buffer) contenant les résultats de conversions sur le bus I2C lorsqu'une lecture est engagée.

Lorsqu'un autre service est activé, un traitement doit être effectué sur les données entre la conversion et le transfert vers l'émetteur. Nous reviendrons sur les traitements du signal développés au cours de ce stage un peu plus loin.

8.4.3. Transfert sur le bus I2C

Chaque service doit disposer d'un espace mémoire dans lequel il entrepose les données un fois qu'elles sont « prêtes ». Cette mémoire tampon doit respecter, dans le format des données qu'elle contient, les normes établies pour la transmission sur le bus I2C, notamment l'emplacement des octets de poids fort et faible (« endianness »). A chaque interruption I2C, un octet du tampon est dépilé et placé dans le registre de transmission du bloc I2C du microcontrôleur.

8.5. Algorithmes de traitements des données

8.5.1. Filtrage de Kalman

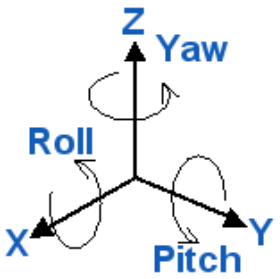
Un des premiers algorithmes envisagés pour les accessoires de mesure inertielle a été le filtre de Kalman. Le but est de reconstituer les 6 axes (3 axes d'accélérations ou de vitesse, trois axes angulaires) filtrés.

En effet, les capteurs type accéléromètres ou gyroscopes posent des problèmes typiques. Les gyroscopes analogiques (la plupart des gyroscopes avec sortie numérique corrigent ce phénomène en interne) mesurent une vitesse angulaire et sont sujet à un phénomène de dérive : un gyroscope au repos à plat donnera une réponse dérivant au cours du temps. Cette dérive doit pouvoir être estimée pour être corrigée et permettre l'obtention de donnée reflétant avec précision la vitesse angulaire, et il n'est donc pas rigoureux ni fiable de recomposer les angles de l'orientation du capteur en effectuant une simple intégration de la vitesse angulaire: la dérive doit être prise en compte et corrigée.

Les accéléromètres échappent à cela mais posent un autre problème: la mesure de l'accélération sur un axe n'est pas indépendante des autres axes. C'est à dire que pour une mesure d'accélération selon un axe effectuée dans une position donnée (par exemple à plat), la mesure selon le même axe d'accélération ne sera pas la même dans une autre position (par exemple avec l'accéléromètre incliné). On peut donc en déduire que l'accélération mesurée selon un axe doit être normalisée en fonction de l'orientation du capteur dans l'espace pour permettre une mesure précise quelque soit l'orientation du capteur.

Enfin, comme tout capteur, les accéléromètres et gyroscopes contiennent une certaine proportion de bruit dans leur signal, qu'il convient donc de filtrer.

Les applications du filtre de Kalman sont nombreuses, et il est particulièrement populaire en aéronautique. Il est souvent utilisé pour filtrer et corriger les mesures d'inclinaison de drones, par exemple. Il permet par exemple, de fournir des données fiables aux moteurs d'autopilotage.
Une méthode particulière est employée couramment pour corriger la dérive des gyroscopes et permettre d'obtenir les angles d'Euler filtrés, il s'agit de la « fusion de capteurs ». C'est cet algorithme qui est utilisé dans l'accessoire « 5 axes », et qui permet donc d'obtenir 2 angles dits d' « Euler » : le pitch, angle « vers l'avant », et le roll, angle latéral.



Avant de revenir plus en détails sur le principe de la fusion de capteurs, il convient d'exposer le principe du filtre de Kalman.
Le filtre de Kalman est un estimateur récursif : l'état présent ne tient compte que de l'état précédent. Le système dont les données sont filtrées doit être associables à un système linéaire, pouvant être modélisée par l'équation d'état d'un système linéaire. Le filtrage se base sur 4 éléments recalculés à chaque itération : l'état du système, la matrice de covariance d'erreur, le bruit dit « de processus » et le bruit dit « d'observation ».

L'état du système représente les données qui vont traverser les différentes composantes du filtre. Il décrira donc l'état du système en entrée, entre les phases de filtrage et en sortie. Dans notre cas, cet état du système comporte un angle et une estimation de la dérive. Les instances du filtre de Kalman seront aussi nombreuses que d'angles que l'on souhaite filtrer.

La matrice de covariance d'erreur représente une estimation de précision de l'état estimé. C'est en quelques sortes le un poids avec lequel on va équilibrer l'importance relative l'état mesuré et l'état estimé.

Les bruits de processus et d'observation sont deux paramètres décisifs sur l'exécution du filtre, car ils sont responsables de la vitesse à laquelle l'estimation va converger vers l'état « réel ».

Deux phases distinctes constituent le filtre : la phase de prédiction et la phase de mise à jour.
La phase de prédiction se base sur l'état précédent du système pour estimer l'état présent. Concrètement, cette étape consiste en premier lieu à intégrer la vitesse angulaire. Dans le domaine discret, une intégration est un procédé relativement sommaire, et ainsi on peux déduire estimer l'angle en intégrant la vitesse angulaire comme ceci :

$$\begin{bmatrix} \text{angle} \\ \text{bias} \end{bmatrix}_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} \text{angle} \\ \text{bias} \end{bmatrix}_{n} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} * \begin{bmatrix} \dot{\text{angle}} \\ 0 \end{bmatrix}$$

où « angle » est l'angle en radian, Δ_t l'intervalle de temps entre l'itération n et l'itération n+1, angle° (point) la vitesse angulaire mesurée par le gyroscope à l'état présent et bias la valeur estimée de la dérive. La valeur de la dérive n'est pas modifiée à cette étape.

$$P_{n+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} * P_n * \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} + Q$$

En second lieu, c'est la matrice de covariance d'erreur qui est mise à jour comme décrit dans l'algorithme:

La phase de mise à jour reprend l'estimation réalisée en phase de prédiction pour mettre à jour et affiner l'état du système.
On calcule donc dans cette phase l'innovation, qui représente la différence entre l'estimation et l'observation, au niveau de l'état comme au niveau de l'erreur. Les valeurs de l'innovation de l'erreur seront déterminantes pour le calcul du gain optimal de Kalman.

L'état du système est mis à jour « a posteriori » en fonction du gain de Kalman et de la valeur de l'innovation de l'état. La covariance de l'erreur est mise à jour en fonction du gain de Kalman.

Cet algorithme est appliqué sur l'accessoire 5 axes suivant la méthode de la fusion de capteurs. Le principe, une fois l'algorithme étudié, est simple : la phase de prédiction sera basée sur la mesure du gyroscope qui représente donc l'estimation: l'état estimé de l'angle concerné est déduit de l'état précédent en effectuant une intégration de la vitesse angulaire par rapport au temps. La base temporelle utilisée n'a pas besoin d'être variable : les données issues des capteurs sont échantillonnées à intervalle très réguliers réglés par un timer, on connaît donc « l'âge » des données, et la base temporelle n'est autre que la période d'échantillonnage si chaque itération du filtre de Kalman est bien effectué entre 2 conversions successives.

La phase de mise à jour effectuera en quelque sorte une correction de l'état estimé en phase de prédiction basée sur les mesures de l'accéléromètre, qui représente donc ici l'observation. D'où l'appellation de « fusion de capteurs ».
En effet certains angles de l'espace angulaire peuvent être calculés à l'aide l'accéléromètre via un simple calcul trigonométrique. L'axe « Z » de l'accéléromètre fournit une base unitaire puisque sa valeur n'est pas nulle lorsque l'accéléromètre est au repos, à la différence des autres axes. La fonction arc tangente de deux axes d'accélération permet d'obtenir une valeur de l'angle, et permet même l'obtention d'une valeur cohérente et fiable lorsque l'un des deux axes est l'axe de « référence », à savoir l'axe Z.
Le troisième angle de l'espace angulaire, le « yaw », ne peut pas être obtenu aussi simplement avec un accéléromètre du fait de la dépendance à la gravité terrestre d'un capteur MeMS.

8.5.2. Détection d'attaque

La détection d'attaque est une analyse qui consiste à détecter les transitions « état repos – état actif » sur un signal. On peut définir l'attaque comme une variation suffisamment importante dans le signal pour signifier un événement bien particulier. On traite ici avec le geste et le mouvement en utilisant des capteurs inertiels, la détection d'attaque doit donc permettre d'identifier des mouvements francs et prononcés. La détection d'attaque correspond tout à fait aux traitement requis par différents scénarios d'utilisation pédagogiques : pour un exercice de synchronisation de la battue musicale du tempo avec la lecture par exemple.
L'algorithme de détection est basé sur une comparaison de la valeur médiane et de la moyenne calculées sur une fenêtre d'échantillons assez petite. Une valeur élevée de cette différence signifie donc une attaque.
L'inconvénient de cet algorithme mais de la détection d'attaque est qu'il est difficile d'extraire une information sur l'intensité de l'attaque : pour des raisons évidentes, il faut que la détection soit immédiate, alors qu'une analyse de l'intensité de l'attaque requiert la prise en compte d'une fenêtre d'échantillons plus large sur laquelle faire l'analyse d'intensité. La détection d'attaque est finalement peu quantifiable, et proche de l'information type « tout ou rien ».

La détection d'attaque fait l'objet d'un service indépendant, comportant donc ses propres paramètres. La taille de la fenêtre peut être modifiée selon le type d'attaque recherchée. Elle doit cependant rester suffisamment petite pour rester au plus près de la latence minimale. La détection peut avoir lieue sur plusieurs axes, et chaque axe comporte un paramètre « seuil ». Ce paramètre ajuste la valeur du seuil auquel sera comparé la différence entre la médiane et la moyenne pour évaluer la présence ou non d'une attaque. De ce fait, la détection d'attaque peut être plus ou moins sensible.
Le service de détection d'attaque a l'avantage d'être un exemple de service dont les données ne doivent pas systématiquement être transférées ce qui est un gain de temps indéniable.

8.6. Deuxième version des émetteurs

Une nouvelle version des émetteurs a pu être réalisée, basée cette fois sur le design de NoDesign. En effet le design électronique précédent ne prenait pas en compte les contraintes mécaniques, le design des modules émetteurs (les « coques ») étant alors au centre du débat.
La deuxième version de l'émetteur reprend certains éléments électronique de la version précédente : mécanisme d'auto-maintien, circuit du module Jennic, LEDs... Et en ajoute de nouveaux pour avancer vers un produit quasiment fini. Elle intègre donc une batterie Lithium-ion 3,7V 250 mAh et un contrôleur de charge Texas Instruments, un LED bicolore centrale (LED chargé de l'affichage de message « bas niveau » ou matériels), et la centrale inertielle 6 axes embarquées (Accéléromètre 3 axes et gyroscope 3 axes).

Le PCB réalisé par Da Fact prend donc bien en compte les contraintes mécaniques instaurées par le design, et les nouveaux modules ont ainsi été livrés à l'Ircam complets : Le PCB fixé dans une coque prototype en dépôt de fil.Cette nouvelle version a néanmoins dû faire l'objet d'une mise à jour de « dernière minute » : la broche d'activation du composant (chip enable) n'était pas forcée à l'état haut lors de la présence d'une tension de charge, et le contrôleur de charge ne pouvait donc pas fonctionner en situation de charge.
En plus des nouveautés sur les émetteurs, un premier design de base de recharge voit le jour, permettant la recharge simultanée de deux émetteurs grâce à 2 ports de recharge.
Les nouveautés qui accompagnent cette version sont nombreuses, et vont permettre d'incorporer de nouvelles fonctionnalités dans le logiciel enfoui (firmware) de l'émetteur, que nous allons décrire ci dessous.

8.6.1. Contrôleur de charge

L'ajout d'un contrôleur de charge, allié à la batterie et à la base de recharge, permet de développer la partie « workflow » du logiciel émetteur, c'est-à-dire les procédures définitives de l'utilisation du module et de son interface utilisateur : mise en route, extinction, affichage de pictogrammes spécifiques etc. Jusqu'ici, les modules étaient reliés en filaire à une alimentation de laboratoire, et ne disposait donc pas de contrôleur de charge. Le contrôleur de charge permet ici de s'adapter aux différentes situations que l'on peut rencontrer quant à l'alimentation. En effet, l'émetteur, peut être en charge ou non, et il convient dans chaque cas de se renseigner sur le niveau de charge de la batterie.
Le contrôleur permet de gérer la recharge et l'alimentation sous batterie, en permettant un suivi du niveau de charge. Le suivi et la différenciation des situations se fait en utilisant 4 broches du contrôleur de charge, reliées à des entrées du Jennic JN 5139.

Parmi ces broches, la broche ACPG (pour AC plugged) détermine si une alimentation externe est branchée en entrée du contrôleur de charge. Si tel est le cas, le contrôleur doit passer en mode recharge, et cette broche change d'état. Pour permettre à l'information sous-jacente de remonter jusqu'au « cerveau » du système, ce signal est relié à l'entrée d'un des 2 comparateurs que propose le Jennic 5139. Ce périphérique est configuré pour fonctionner sous interruptions, et de ce fait une interruption pourra être généré sur front ascendant ou descendant. Étant donné qu'un front ascendant est ici forcément suivi d'un front descendant, l'interruption est modifiée à chaque interruption pour générer une interruption dans le sens opposé (par exemple, si une interruption par front ascendant est générée, l'interruption est immédiatement reconfigurée pour avoir lieue lors d'un front descendant). De cette manière, tout changement de situation au niveau de la charge est détecté immédiatement.

Les trois autres signaux provenant du contrôleur de charge sont le signal de suivi du niveau de charge (BAT_MON pour *Battery Monitoring*), et deux signaux logiques permettant décrire les différentes situations possibles. Ces signaux sont reliés à 3 des 4 entrées du convertisseur analogique numérique du Jennic JN 5139. Il n'est pas utile d'échantillonner ces signaux à une fréquence très élevée, car d'une part les signaux

de statut ne sont susceptibles de changer que lorsque un événement particulier intervient, et que d'autre part, une actualisation trop fréquente du niveau de charge de la batterie n'a aucun intérêt en soi, un échantillonnage à fréquence « humaine » (soit une période entre 1 et 3 secondes par exemple) suffit amplement.
Il est par contre nécessaire de détecter le changement de statut dès lors que ce dernier a lieu, et pour cette raison, un échantillonnage supplémentaire des signaux de statut est effectué lorsqu'une interruption du comparateur intervient : l'analyse des signaux permet alors de vérifier la situation de l'émetteur, et de le passer par exemple en mode « charge ».
Le suivi du niveau de charge est une information qui, typiquement, relève du matériel mais doit être relayée à l'utilisateur. Ce type d'information doit être communiquée par la LED centrale (le « nombril »).

8.6.2. LED « nombril » centrale

La LED centrale regroupe deux LEDs de couleurs différentes (rouge et vert) dans un seul composant. Ce composant est positionné entre une sortie du convertisseur numérique analogique du module Jennic et deux sorties standards du même module. Le circuit permet donc de contrôler l'allumage des deux LEDs de façon distincte, mais ne permet pas de contrôler les intensités lumineuses des deux LEDs indépendamment.
Les sorties du Jennic contrôle donc l'extinction/allumage des LEDs, et le convertisseur numérique analogique permet de contrôler leur intensité lumineuse.
Le suivi du niveau de batterie, hors situation de charge, aboutit sur un affichage classique du niveau d'autonomie restant : du vert au rouge en passant par l'orange.
Lorsque le module Jennic détecte une charge, seule la LED verte est active, et un effet de respiration est appliqué à l'intensité lumineuse par le convertisseur numérique analogique. Le convertisseur va alors chercher des valeurs dans une table de recherche (lookup table) contenant un signal triangulaire, l'index de lecture étant incrémenté en synchronisation avec un timer.

8.6.3. Centrale inertielle embarquée

La centrale inertielle, composée d'un accéléromètre 3 axes et d'un gyroscope 3 axes, est l'une des nouveautés majeures introduites dans la nouvelle version des émetteurs. Les capteurs utilisés se situe dans le haut de gamme des capteurs du marché qui se prêtent à l'application qui en est prévue dans le projet : ce sont des capteurs numériques, proposant une résolution convenable (supérieure à celle souvent offerte par les convertisseurs de microcontrôleurs), ainsi qu'un certain nombre de fonctionnalités pouvant s'avérer utiles (ajout d'offset au valeur, pile de traitement FIFO, modification de la sensibilité du capteur...).

Ces capteurs sont reliés sur le même bus I2C que les éventuels accessoires, et sont prise en charges de la même façon. Ils sont donc vus comme un accessoire à part entière par l'utilisateur, la seule différence étant que les traitements sont fait directement dans le Jennic.
Bien entendu, les modules Jennic ont déjà la tâche de la gestion du réseau et leur puissance de calcul laisse une place modérée pour les éventuels algorithmes de traitements. Il s'agissait donc là d'un point à valider : le module Jennic serait-il capable d'assurer un débit suffisamment important de donnée tout en assurant un traitement des données sur les 6 axes?

Le module Jennic partage une structure rigoureusement identique à celle du microcontrôleur accessoire pour décrire l'accessoire embarqué. De cette manière, l'utilisateur communique avec cet accessoire selon les mêmes modalités. Les avantages liés à l'utilisation de l'accessoire embarqué sont nombreux. Il permet en effet, d'un point de vue pratique, d'avoir à disposition un accessoire dès lors que l'on a un émetteur. Cet accessoire est également de bonne facture : les convertisseurs du PIC ne permettent pas une résolution aussi précise (11 bits signés pour l'accéléromètre, 16 bits signés pour le gyroscope), alors que la taille mémoire des données est rigoureusement identique (2 octets par axes). Leur qualité numérique permet également, au niveau des contraintes de temps, d'obtenir de très bon résultats en flux tendu, en éliminant les étapes intermédiaires de conversion, interfaçage microcontrôleur sur le bus I2C, etc. C'est donc un accessoire particulièrement adapté aux applications à un ou 2 émetteurs en streaming, pour lesquelles la faible latence est un paramètre indispensable.
Cependant, la liaison directe avec le module Jennic n'offre pas, à priori, les mêmes possibilités en terme de traitement du signal que le PIC.
Un portage du filtre de Kalman programmé sur le PIC a été réalisé en tant que second service offert par l'accessoire embarqué de l'émetteur. Le résultat s'est avéré plutôt positif, bien que le débit soit assez largement inférieur au service « streaming » : le traitement est suffisamment rapide pour rendre le service utilisable en contexte.

IX. Conclusion

Le travail réalisé au cours de ce stage aura permis, conjointement aux travaux de design mécanique et électronique, de franchir différentes étapes de prototypage. Le cahier des charges du projet Interlude comprend divers choix technologiques et particularités novatrices, qui devaient être validées par implémentation de prototypes. La conception des différents logiciels embarqués est évidemment un élément décisif dans cette optique, conjointement à la conception mécanique, physique et électronique. Si des détails pourront et seront certainement amenés à être modifiés, la base du logiciel embarqué est fonctionnelle et stable: tous les éléments essentiels sont pris en charge pour permettre la maintenance du réseau, la gestion de l'alimentation, la communication avec les accessoires, un des objectif de ce stage ayant été de valider l'intégration des différentes technologies choisies entre elles et ainsi d'aboutir sur un prototype se rapprochant au plus possible d'un système opérationnel.

Ces objectifs sont atteints : des prototypes fonctionnels, prêts à l'utilisation ont pu être remis à l'aile « pédagogique » de l'équipe du projet (l'Atelier des Feuillantines) afin d'effectuer des essais en situations, constituant ainsi une sorte de bêta-test. Ces prototypes se rapprochent grandement d'un produit fini (la mallette pédagogique) bien que certains aspects (boîtier en dépôt de fil, design non définitif pour les bases…) soient en retrait, et devraient permettre la réalisation de certains scénarios d'utilisations. Cette phase de mise en situation aura très certainement une plus grosse influence sur l'évolution future du logiciel embarqué que sur le design : des problèmes peuvent encore apparaître puisque les situations peuvent s'avérer assez différentes de celles offertes par l'environnement de développement.
D'autres objectifs, moins évidents mais tout aussi importants, sont à considérer : en effet le développement mené sur ce projet doit représenter une innovation, d'autant plus qu'il s'agit d'un projet de recherche. L'abandon des modules Xbee au profit des modules Jennic, par exemple, devait apporter une plu value. C'est effectivement le cas, puisque ces modules ont prouvés à travers ce projet leurs performances au niveau radio et ont permis, grâce à leur interface de programmation ouverte, non seulement de respecter et d'améliorer la contrainte intrinsèque au temps réel, mais également d'interfacer différents périphériques. Ces aspects, une fois mis en place, mettent en avant certains concepts clés qu'apporte le système : l'interface standardisée entre les microcontrôleurs Jennic et PIC, la découverte de services, le traitement embarqué des données.
En effet, l'état du développement à l'issue de ce stage devrait permettre aux accessoires imaginés par l'équipe d'être développés sans que le logiciel de l'émetteur ou de la base OSC ne nécessite de modification, en utilisant la base de logiciel prévue pour les accessoires actifs (ce code étant donc prévu pour les PIC24F). De la même manière, ces processeurs ont été validés du fait de leurs performances en terme de consommation, de périphérique disponibles et de puissance de calcul : à l'image du filtrage de Kalman, d'autres moteurs d'analyse pourront venir s'ajouter, spécifiquement aux accessoires.
La mise en place effective du côté « hardware » du projet devrait également permettre aux développeurs et designers de travailler sur une interface utilisateur adaptée, puisque l'architecture matérielle du système définit certaines modalités, comme notamment au niveau de l'adressage OSC. Le format de réception des données (Ethernet/UDP/OSC) autorise bien entendu la réception des données sans passer par une interface prédéfinie, mais le cahier des charges prévoit cependant de proposer une interface graphique, en vue du contexte d'utilisation pédagogique envisagé.

Si ce projet reste dans le domaine de la recherche, les partenaires industriels (Voxler, Da Fact) pourront également y voir une base pour le développement d'un produit commercialisable.

XI. Annexes

Modular Musical Objects Towards Embodied Control Of Digital Music

Nicolas Rasamimanana, Frederic Bevilacqua, Norbert Schnell, Fabrice Guedy, Emmanuel Flety, Come Maestracci, Bruno Zamborlin
Ircam - CNRS STMS
Real Time Interaction Team
1 place Igor Stavinsky, 75004 Paris
Nicolas.Rasamimanana@ircam.fr

Jean-Louis Frechin, Uros Petrevski
NoDesign
12, passage Turquetil, 75011 Paris France
jlfrechin@nodesign.net

ABSTRACT
We present an ensemble of tangible objects and software modules designed for musical interaction and performance. The tangible interfaces form an ensemble of connected objects communicating wirelessly. A central concept is to let users determine the final musical function of the objects, favoring customization, assembling, repurposing. This might imply assembling the wireless interfaces with existing everyday objects or musical instruments. Moreover, gesture analysis and recognition modules allow the users to define their own action/motion for the control of sound parameters. Various sound engines and interaction scenarios were built and experimented. Some examples that were developed in a music pedagogy context are described.

Author Keywords
Music, Gesture, Interface, Gesture Recognition, Audio Processing, Design, Interaction, Design

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: Miscellaneous—*Optional sub-category*

INTRODUCTION

The expressive control of digital sound and audio processing has been the focus of important research over the last decade. Gestural and tangible interfaces actually are promising approaches in order to control abstract, disembodied digital sounds. Such interfaces indeed afford the use of physical movements to control digital sounds, and can therefore be used to create an embodied music experience. As described by several authors, the use of physical gestures is an essential link between a musician and his/her instrument [9, 21, 6]. Thanks to the increasing availability of gesture sensing systems, innovative digital musical instruments [14] emerged recently from the computer music community, opening several novel approaches for musical expression using digital sounds [7].

Using gesture input and tangible interfaces brings up important design and usage issues. In [22], authors posited that gestural controllers were in most cases designed to fit idiosyncratic needs, and as a consequence were inextricably bound to their creators. As a matter of fact, musical interfaces are often considered as part of an artistic endeavor, and not always meant to be embraced by a large community of users. Even if some counterexamples can be found [22, 10], no clear methodologies or approaches to design and evaluate musical interfaces has clearly emerged yet, which makes generally his research community not integrated in the mainstream HCI field.

Interestingly, low-level hardware components such as Arduino [23] have been largely embraced by the computer community, showing clearly that subcomponents, if generic enough, can be widely adopted as a means to built complex systems. This can be paralleled to software environments such as Max/MSP [11] or Pd [17], that are based on building complex applications from relatively simple modular components. In this paper, we describe a project on tangible music interfaces, based on a similar bottom-up approach. Our general goal is to provide users with both hardware and software components, considered as building block to create various types of embodied music experiences. We aim at giving users opportunities to incorporate objects, gesture and sound materials of their choice in the interaction. We are particularly interested in scenarios where objects and gestures are taken or inspired from other (non-musical) applications, in order to provide some grounds to facilitate gesture control. From a general perspective, this can thus be compared to the use of natural user interfaces discussed recently by [1].

The paper is structured as follows. We first present our general design approach. Second, we describe hardware and software components. Finally, we describe some examples that were experimented in a music school.

DESIGN STRATEGY

Submitted for review to TEI Conference 2010

One can oppose two different strategies in designing complex systems: *top-down* and *bottom-up* approaches. The first one consists in creating objects[1] that are fully completed and developed with specific functionalities and uses. As such, objects can hardly be modified and users need to "adapt" to the preconceived functionnalities through learning [12]. On the contrary, in a *bottom-up* design strategy, users can actively create the final working system using simple elements . As such, this approach favors customization, assembling, repurposing. Typically, this approach encompasses do-it-yourself (DIY) such as proposed by Arduino.

As illustrated in Figure 1, we propose a bottom-up approach (points 2, 3, and 4 in Figure 1) we call *user-completed systems* [16], opposed to fully finished interfaces and applications with fixed, predefined usages. Major possibilities include:

- assemble existing basic modules
- "parasite/hack" existing interfaces or musical instruments
- invent new interfaces/usages

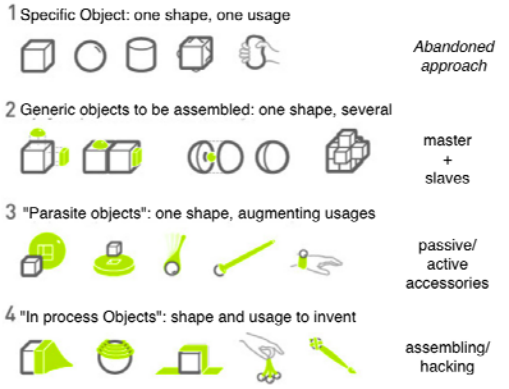


Figure 1. Design approaches

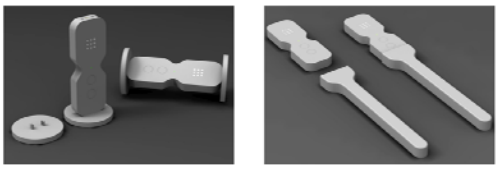
In this project our goal is to develop hardware and software modules that are intermediate between low-level DIY systems like Arduino [23] or Max/MSP [11] and high level musical system or instruments. Therefore we propose basic elements that require no electronic or programming expertise (without precluding expert users to add new elements). We also propose scenarios or "recipes" to use to create musical interfaces. Similar intermediate level and modular approach in music interfaces can be found systems such as the ReacTable [8], BlockJam [15], or the Siftables [13]. However, unlike these examples, we encourage the possibility of hacking and repurposing other objects. Applications hence to borrow performing concepts from other disciplines, in visual art (e.g. painting), game (e.g board games) and even sports (e.g ball games). Moreover, using dedicated gesture analysis and recognition software modules, users can propose and incorporate particular motion and actions in the interaction (see the section on software modules).

[1] here the term object must be understood in a broad sense as anything that is produced, a physical object or a software

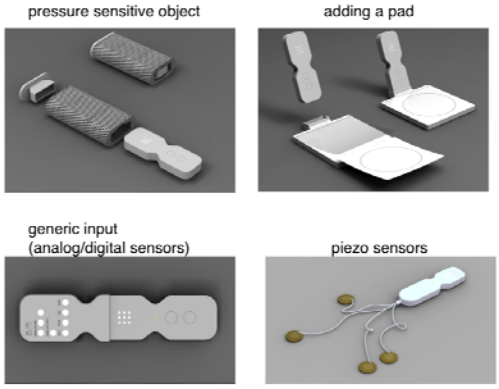
Hardware Modules

We developed a modular set of objects, called MO, that can be assembled to create tangible sound interfaces. The platform is based on a central wireless module shown in Figure 2. This module, that can be easily held in one hand, contains 6 DOF inertial sensors (accelerometers and gyroscopes), two buttons and LEDs displays and a rechargeable battery. The wireless is based on the IEEE 802.15.4, generally called "Zigbee". We use a Jennic OEM JN-5139.

Figure 2. MO principal module



Wireless motion module combined with passive accessories



Wireless motion module combined with active accessories

Figure 3. MO accessories (simulation)

Alone, this module allows for motion interactions. Hand gesture can directly be used. But, as already mentioned, we want to stress that its usages can be varied using either passive or active accessories. In particular, active accessories can be directly assembled at each extremity of the modules, using a I2C bus. A first set of accessories is in developments (see Figure 3):

- Piezos sensors, e.g. to create a touch sensitive surface
- A pressure sensitive cover, e.g. to make a motion sensible graspable object.
- Generic analog and digital inputs to add any other sensors
- An additional 6 DOF inertial sensors, connected through a cable, e.g. to get relative motion between the module and the accessories

We plan to progressively add new components. Thanks to the "generic input" accessory, any new sensors can already easily be incorporated.

Software Modules

Similarly to the hardware developments, we conceived modular software components for both gesture/motion analysis and audio processing/synthesis. The elements are programmed in Max/MSP, which facilitates rapid prototyping.

One of the key components of the gesture analysis consists in a gesture recognition and synchronization engine called "gf" (standing for gesture follower) [2]. This component allows the user to simply record gestures that are then recognized by the system. The main algorithm is based on Hidden Markov Models, but using a non-classical implementation to take into accounts the two following specific constraints. First, a single example can be used for the training, which allows users to easily choose or adapt gesture to their capabilities. Second, the system can perform early-recognition of gestures and continuously synchronize gestures to audio processing. Specifically, the recognition system is designed to accommodate interaction paradigms using both discrete triggering and continuous control.

For audio processing, we integrate a set of synthesis and sound transformation modules [18, 20, 19]. Several of them integrate the possibility to interactively work on recorded sounds. For example, we use various granular and phase vocoder techniques to specifically alter sound characteristics while preserving others (temporally stretching sounds without changing pitch or vice-versa).

EXAMPLE USES

In this section, we present different works integrating the presented hardware and software components. These works were made and used by music teachers and students with the help of our research team. The interaction systems were elaborated in a context of music creation and music pedagogy throughout the year, and were presented during the end-of-year student concert. These experiments were designed within a larger pedagogical context, which description is beyond the scope of this paper.

MO ball in collaborative mode

In this case, the music interaction was based on a ball game, augmented with our wireless module (see Figure 4). Typically, a percussive sound is played each times the ball is thrown and caught. In one example we experimented with, string pizzicati were triggered one by one, respecting the score of an actual musical piece. The music performance was thus bound to the continuous and regular ball passing. This collaborative musical game could be played in pairs or with large groups. While everybody can easily plays such a game, yet, it requires interesting group interaction, eye-contact, and collective concentration to perform the music. As a matter of fact, ballistic movements (a found when throwing ball) have always been a salient metaphor in music [4]. This indeed encompass important musical notions such as up-beats, down-beats, phrasing and pulsation. Note that the use of balls as musical interfaces were previously experimented by other authors [5, 3].

Figure 4. MO ball setup

MO ball with continous control

Contrary to the previous mode where discrete events were triggered, this mode of MOBall focuses on the control of continuous digital sounds. Users hold the ball and can define their own gestures, which are then used to directly act on the pacing on the sound: sounds are stretched when slowing down and sounds are shortened when speeding up. This mode enables the use of any continuous gestures to lead music, using metaphors ranging from conducting to dance. Users can re-perform music piece using their own gestures and interfaces.

Figure 5. MO ball in with continous control. First, choose one or several sound files and record a set of different gestures. Then,the recognition system along with the sound engine allows the user to gesturally select and control continuously parameters of the sound. For example, sound can stretch according to the pace of the gestures

MO chess

In a particular music context, the teacher and students proposed to use a chess game as an interaction metaphor. The game was found to resonate with musical concepts about opposition and dialogue (e.g. canons or duets). A chess table was then augmented with a "piezo" to register when a piece was put on the table. Players has also attached a MO module to their wrists to track the hand motion above the table. All the sensors were used to control the tempo of a musical piece. Using this setup, performance of a music duet was possible, where each player controls a different music line.

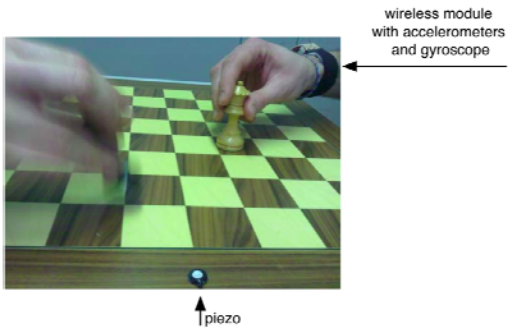


Figure 6. Sound control using a chess board game

CONCLUSION

We presented an modular set of tangible interfaces for sound control, emphasizing gesture and action. Our design strategy was to propose a bottom-up approach: assembling modules, augmenting existing objects and hacking in order to let emerge novel usages with basic interfaces. Moreover, the tangible interfaces are used in conjunction of gesture recognition modules, which invite users to define their own control gestures. Several sound modules were also implemented, based on recorded sound that remained opened to the users. Finally, we presented preliminary works achieved in a music school, which demonstrated promising potential of this approach for music creation and pedagogy. We are currently pursuing such applications, developing several other scenarios and applications.

ACKNOWLEDGMENTS

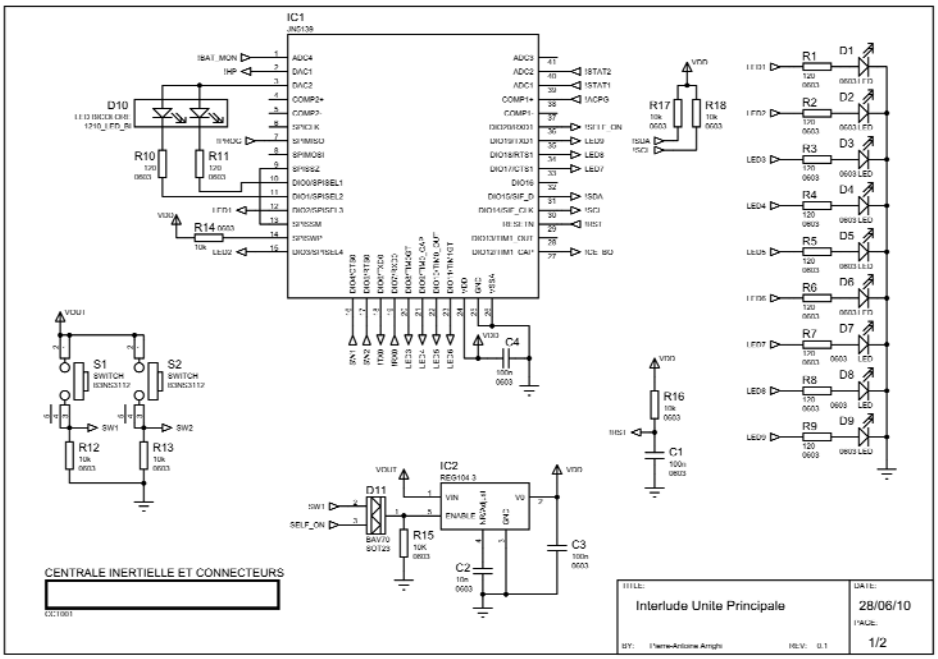
We acknowledge support of the ANR (French National Research Agency) project Interlude (ContInt). We also thank the whole Interlude consortium and students of l'Atelier des Feuillantines. The second generation of MO prototypes were finalized by Da Fact (www.dafact.com)

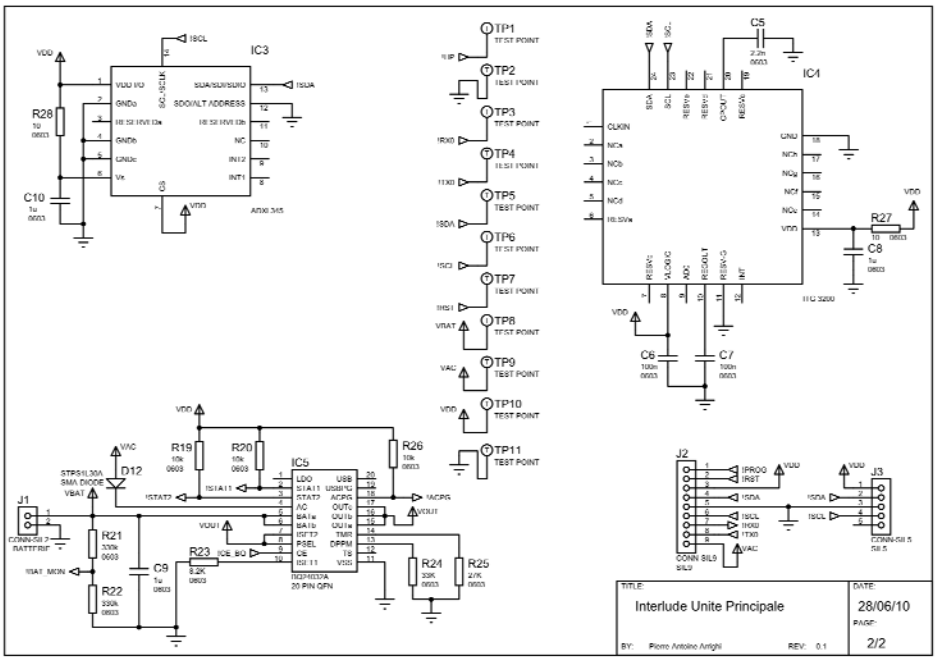
REFERENCES

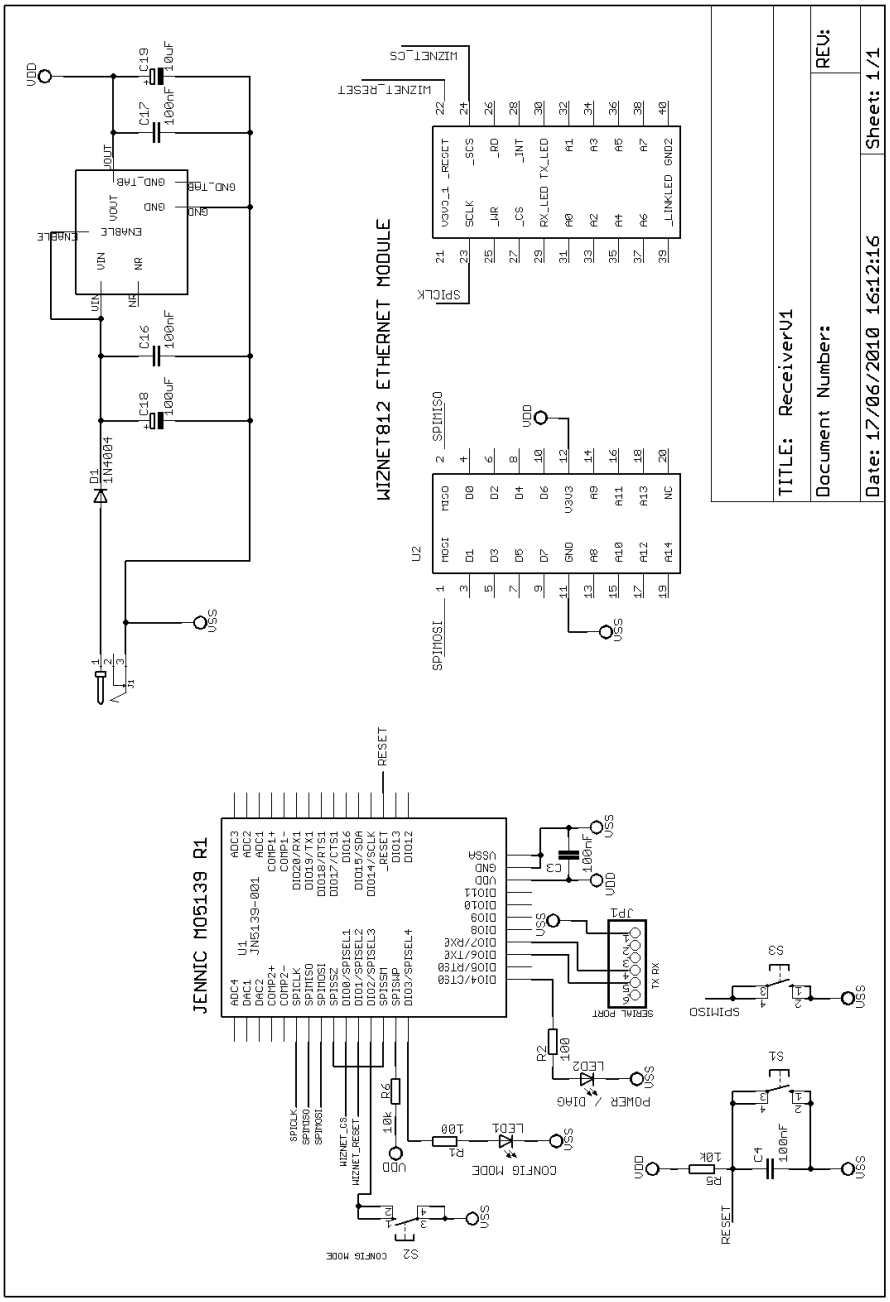
1. B. Buxton. Interview on Channel 9. CES 2010. http://channel9.msdn.com/posts/larrylarsen/ces-2010-nui-with-bill-buxton/.
2. F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guedy, and N. Rasamimanana. Continuous realtime gesture following and recognition. In *LNCS*. Springer Verlag, 2009.
3. T. Blaine and T. Perkis. The jam-o-drum interactive music system: A study in interaction design. In *DIS*, 2000.
4. E. Dalcroze. *Rhythm, music and education*. Library Reprints (January 2001), 1921.
5. EZ3kiel. Musical ball. http://www.dailymotion.com/video/x4rub7_ez3kiel-ballon-musical-live-la-ciga_music.
6. R. Godøy and M. Leman, editors. *Musical Gestures: Sound, Movement and Meaning*. Routledge, 2009.
7. International Conference on New Interfaces for Musical Expressions. http://www.nime.org.
8. S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: Exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI*, 2007.
9. M. Leman. *Embodied music cognition and mediation technology*. The MIT Press, Cambridge, Massassuchetts, 2007.
10. J. Malloch, D. Birnbaum, E. Sinyor, and M. Wanderley. Towards a new conceptual framework for digital musical instruments. In *DAFx*, 2006.
11. Max/MSP. http://www.cycling74.com.
12. J. McGrenere, R. Baecker, and K. Booth. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of CHI*, 2002.
13. D. Merrill, J. Kalanithi, and P. Maes. Siftables: Towards sensor network user interfaces. In *TEI*, 2007.
14. E. Miranda and M. Wanderley. *New Digital Musical Instruments: Control and Interaction beyond the Keyboard*. A-R, 2006.
15. H. Newton-Dunn, H. Nakano, and J. Gibson. Block jam: A tangible interface for interactive music. *Journal of New Music Research*, 32(4):383–393, 2003.
16. NoDesign. http://nodesign.net/.
17. Pure Data. http://puredata.info/.
18. N. Schnell, R. Borgbesi, D. Schwarz, F. Bevilacqua, and R. Muller. Ftm - complex data structures for max. In *ICMC*, 2005.
19. N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi. Mubu and friends: Assembling tools for content based real-time interactive audio processing in max/msp. In *ICMC*, Montreal, Août 2009.
20. N. Schnell and D. Schwarz. Gabor, multi-representation real-time analysis/synthesis. In *DAFx*, 2005.
21. M. Wanderley and M. Battier, editors. *Trends in Gestural Control of Music*. Ircam, 2000.

22. M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, 26(3):62–76, 2002.

23. N. Zambetti. http://www.arduino.cc/.


IC1
JN5139
D10
R10
R11
R14
S1
SWITCH
S2
SWITCH
R12
R13
D11
BAV70
SOT23
IC2
REG104-3
R15
C2
C3
C4
R1
R2
R3
R4
R5
R6
R7
R8
R9
D1
D2
D3
D4
D5
D6
D7
D8
D9
R16
C1
R17
R18
CENTRALE INERTIELLE ET CONNECTEURS
Interlude Unite Principale
28/06/10
1/2


IC3
AT30.345
R28
C10
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
TP9
TP10
TP11
C5
IC4
ITC 3200
R27
C8
C6
C7
D12
STPS130A
SMA DIODE
R19
R20
R26
IC5
20 PIN QFN
R21
R22
R23
R24
R25
C9
J1
J2
J3
CONN SIL9
Interlude Unite Principale
28/06/10
2/2


WIZNET812 ETHERNET MODULE
JENNIC M05139 R1
TITLE: ReceiverV1
Document Number:
REV:
Date: 17/06/2010 16:12:16
Sheet: 1/1

MICROCHIP

PIC24FJ64GA004 FAMILY

28/44-Pin General Purpose, 16-Bit Flash Microcontrollers

High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with 4x PLL Option and Multiple Divide Options
- 17-Bit by 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16-Bit x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture:
  - 76 base instructions
  - Flexible addressing modes
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

Special Microcontroller Features:

- Operating Voltage Range of 2.0V to 3.6V
- 5.5V Tolerant Input (digital pins only)
- High-Current Sink/Source (18 mA/18 mA) on All I/O Pins
- Flash Program Memory:
  - 10,000 erase/write
  - 20-year data retention minimum
- Power Management modes:
  - Sleep, Idle, Doze and Alternate Clock modes
  - Operating current 650 µA/MIPS typical at 2.0V
  - Sleep current 150 nA typical at 2.0V
- Fail-Safe Clock Monitor Operation:
  - Detects clock failure and switches to on-chip, low-power RC oscillator
- On-Chip, 2.5V Regulator with Tracking mode
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with On-Chip, Low-Power RC Oscillator for Reliable Operation
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins
- JTAG Boundary Scan Support

Analog Features:

- 10-Bit, up to 13-Channel Analog-to-Digital Converter:
  - 500 ksps conversion rate
  - Conversion available during Sleep and Idle
- Dual Analog Comparators with Programmable Input/Output Configuration

Peripheral Features:

- Peripheral Pin Select:
  - Allows independent I/O mapping of many peripherals
  - Up to 26 available pins (44-pin devices)
  - Continuous hardware integrity checking and safety interlocks prevent unintentional configuration changes
- 8-Bit Parallel Master/Slave Port (PMP/PSP):
  - Up to 16-bit multiplexed addressing, with up to 11 dedicated address pins on 44-pin devices
  - Programmable polarity on control lines
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions
- Programmable Cyclic Redundancy Check (CRC)
- Two 3-Wire/4-Wire SPI modules (support 4 Frame modes) with 8-Level FIFO Buffer
- Two I^2C™ modules support Multi-Master/Slave mode and 7-Bit/10-Bit Addressing
- Two UART modules:
  - Supports RS-485, RS-232, and LIN 1.2
  - On-chip hardware encoder/decoder for IrDA®
  - Auto-wake-up on Start bit
  - Auto-Baud Detect
  - 4-level deep FIFO buffer
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Five 16-Bit Capture Inputs
- Five 16-Bit Compare/PWM Outputs
- Configurable Open-Drain Outputs on Digital I/O Pins
- Up to 4 External Interrupt Sources

PIC24FJ Device	Pins	Program Memory (bytes)	SRAM (bytes)	Remappable Peripherals						I^2C™	10-Bit A/D (ch)	Comparators	PMP/PSP	JTAG
				Remappable Pins	Timers 16-Bit	Capture Input	Compare/ PWM Output	UART w/ IrDA®	SPI					
16GA002	28	16K	4K	16	5	5	5	2	2	2	10	2	Y	Y
32GA002	28	32K	8K	16	5	5	5	2	2	2	10	2	Y	Y
48GA002	28	48K	8K	16	5	5	5	2	2	2	10	2	Y	Y
64GA002	28	64K	8K	16	5	5	5	2	2	2	10	2	Y	Y
16GA004	44	16K	4K	26	5	5	5	2	2	2	13	2	Y	Y
32GA004	44	32K	8K	26	5	5	5	2	2	2	13	2	Y	Y
48GA004	44	48K	8K	26	5	5	5	2	2	2	13	2	Y	Y
64GA004	44	64K	8K	26	5	5	5	2	2	2	13	2	Y	Y