



OBJETS CONNECTÉS

Sujets des projets 2024

Christophe Bernard

Station Météo – Ethernet/Processing

Christophe BERNARD - Sujet 1 - 2 élèves

Objectifs :



Conditionnement de signaux électriques issus de différents capteurs, conversion Analogique/numérique, mémorisation d'un historique sur une carte de type micro SD. Affichage de ces valeurs en temps réel sur l'écran de la carte **Manuella** (2 lignes, 16 colonnes). En parallèle si la carte **Manuella** est couplée à un ordinateur compatible Pc on

reçoit toutes les informations sur une interface graphique et numérique. Cette interface sera développée en utilisant l'environnement de programmation graphique **Processing**. **Processing** permettant un lien direct avec les cartes *Arduino*.

3

Cette station météo doit être capable de mesurer les paramètres suivants :

- température de **-20° C** à **+50° C** avec une précision de **±0.5° C**,
- le vent de **0** à **120 km/h**,
- l'hygrométrie : humidité relative de **5** à **100 %**,
- la luminosité de **0** (*noir complet*) à **100 %** (*plein soleil*).

Partie Software

Sur la 1^{ère} ligne de l'afficheur de la carte **Manuella**, on ne visualisera qu'un seul paramètre à la fois : **"intitulé de la mesure :"** suivi de la **valeur instantanée**.

À l'aide d'un capteur gestuel :

- un mouvement de votre main de la droite vers la gauche permet de visualiser :
 - la valeur instantanée,
 - le maximum et minimum depuis le dernier **RESET** (*sauvegardés dans la SD Card*),
 - la moyenne des grandeurs (*sauvegardées dans la SD Card*).

De plus **3** actionneurs (*servomoteur analogique*) permettront de simuler l'ouverture et la fermeture de **2** volets et **d'1** store en **fonction des conditions météorologiques** autrement dit suivant la logique de votre algorithme...,

Sur la 2^{ère} ligne de l'afficheur de la carte **Manuella**, celle-ci sera consacrée à mentionner l'état des 2 volets ainsi que celui du store. N'oubliez pas que l'utilisateur peut avoir accès à l'ouverture/fermeture manuelle des volets et store !



Exemple de représentation ...

4

Lors d'un affichage général de paramètre (*comme celui de la figure ci-dessus par exemple, et ce, quelque soit le paramètre visible !*), il est possible d'ouvrir et/ou fermer les volets et le store :

- ✓ un appui sur la touche **#** suivi de la touche **1**, permet d'actionner le **volet 1**,
 - ✓ un appui sur la touche **#** suivi de la touche **2**, permet d'actionner le **volet 2**,
 - ✓ un appui sur la touche ***** suivi de la touche **1**, permet d'actionner le **store**,
- un mouvement de votre main de bas en haut permet de visualiser :
- les différents paramètres,

Le système permettra d'envoyer les informations mesurées (*valeurs instantanées*) sur un *serveur web* à l'aide d'une connexion Ethernet (*rafraîchissement des informations toute les 5 minutes*), la visualisation sera la plus élémentaire. Cependant, une fois **Manuella** en relation avec un PC, nous disposerons d'une interface plus élaborée (*graphique et numérique*), interface réalisée à l'aide de Processing.

Christophe BERNARD
IMT Nord-Europe



MOTS CLES - A DECOUVRIR

Afficheur graphique, bases de la conversion A/N sur Arduino, mémorisation/réservation espace RAM, écriture/lecture sur carte SD, opération bits à bits, Processing, réseau Ethernet, etc...

Analyse qualité de l'air intérieur - Bluetooth

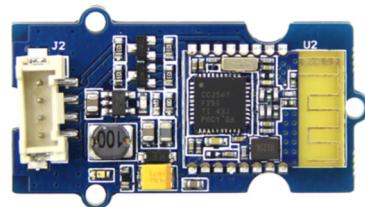
Christophe BERNARD - Sujet 2, 4 élèves

5

Objectifs

L'objectif de ce sujet n°2, est de traiter les informations en provenance de différents capteurs spécialisés à la **détection de gaz et de poussières**, d'**analyser** les données reçues en fonction de la **température et de l'humidité** de la pièce **ou** l'ensemble des capteurs se situent. Les résultats de cette analyse vous permettront de signaler **la qualité de l'air** sur votre smartphone **Android**, et de **programmer ou déclencher**

manuellement une éventuelle **ventilation de cette pièce**. Pour établir le lien entre l'Arduino et votre smartphone, vous utiliserez une liaison hertzienne **Bluetooth**.



Module BLUETOOTH Grove

Le **Bluetooth** est une norme de communication utilisant les ondes radio sur la bande de fréquences comprise entre 2,4 GHz et 2,4835 GHz, c'est une liaison faible débit (*2 Mbps*) avec une portée de transmission pouvant atteindre 350 mètres – **Version 5.0 (2016)**. Dans ce projet, vous exploitez une couche logicielle baptisée **Logical Link Control & Adaptation Protocol (L2CAP)**, et plus particulièrement un service de communication nommé "*service RFCOMM*". Ce service vous permettra d'établir une liaison de données similaire à une liaison série de type **RS232**.

Nous utiliserons des modules **Grove** ; ces modules sont un ensemble de composants "**Plug-and-Play**" open source permettant ainsi de simplifier le prototypage et par conséquent de vous concentrer rapidement sur l'interaction Hardware-Software.

Partie Software

A l'aide d'un **Arduino Uno**, vous devez établir un algorithme qui permet de déterminer la qualité de l'air d'une pièce. Le résultat de vos analyses sera représenté sous forme d'un histogramme empilé sur une interface réalisée sous **Android** pour smartphone (*cf. Figure 1*). Cet histogramme représentera la qualité de l'air suivant trois seuils "**Très Bon/Bon – Moyen – Mauvais/Médiocre**".

Analyse des résultats - 20 relevés par heure

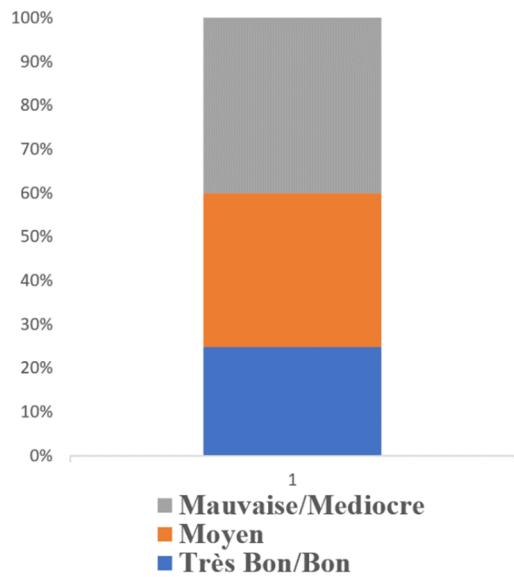
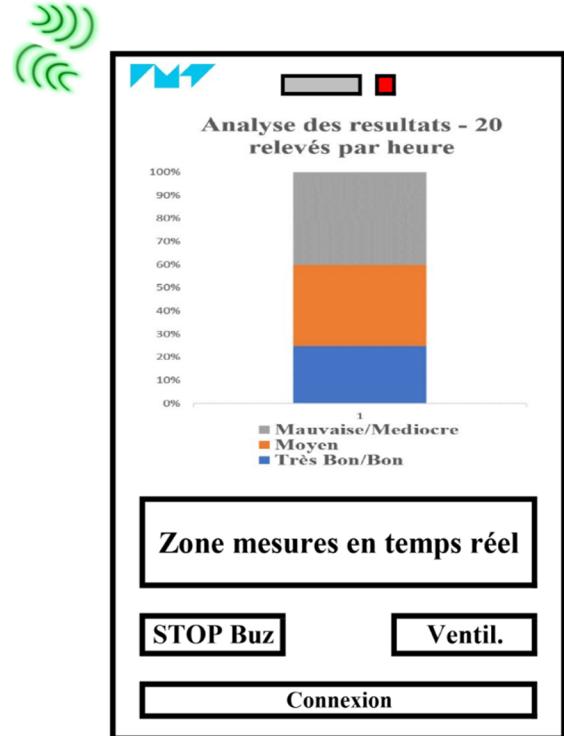


Figure 1 : Exemple de représentation



Exemple de présentation sur Smartphone

Pour cela vous ferez une série de **20** mesures par heure du module Grove eqCO2 en ppm (**D****i****O**xyde de **C**arbone), du TCOV en ppb (Taux de Composés Organiques Volatils) et de la concentration de poussières μ/m^3 . Le résultat des mesures sera corrélé en tenant compte de la température et de l'humidité, effectivement ces deux derniers paramètres favorisent plus ou moins une bonne ou mauvaise qualité de l'air. Dès que nous atteignons un seuil critique de **30 % d'analyse mauvaise et\ou médiocre** nous déclenchons un **son "strident"** sur le **buzzer** (**3 bips d'une durée de 10s**), suivi de la mise en **ventilation** de la pièce. La ventilation reste active **tant que** l'on ne redescend pas en dessous du seuil critique **ou** que l'utilisateur ne l'arrête volontairement par le biais de l'interface *Android*.

Le seuil critique atteint, provoque en parallèle une **information lumineuse** sur le smartphone (**carré de 64 pixels**) clignotant alternativement **rouge\blanc** à la fréquence de **0,5 Hz**.

Sur cette interface *Android* il sera possible :

- de se **connecter** à l'application matérielle,
- d'afficher dans une **zone de texte**, le **relevé des mesures avec leurs unités** respectives en **temps réel**,
- de **déclencher** ou **d'arrêter** la **ventilation** de la pièce,
- de **stopper le buzzer**.

Christophe BERNARD

IMT Nord-Europe



MOTS CLÉS - À DECOUVRIR

Arduino, Bluetooth, Liaison série RS232, Android, CO₂, Taux de composés organiques volatils, concentration de poussière, PWM, capteur de Gaz SGP30, température, humidité, relais, etc...

Remarque

*Dans ce sujet, il y a une grande part d'études de documents sur la législation concernant les seuils pour la qualité de l'air dans **une pièce** (habitat). Ce sujet ne concerne **absolument pas** la qualité de l'air extérieur définie par d'autres relevés de mesures, de modèles mathématiques ; dans ce cas on parle de pollution atmosphérique.*

Luminothérapie - XBEE

Christophe BERNARD - Sujet 3, 2 élèves

8

Objectifs

La luminothérapie utilise des rayons lumineux artificiels afin de pallier les manques de lumière naturelle ressentis chez certaines personnes, ce qui se traduit par un dérèglement de l'horloge biologique interne : *de la fatigue, une dépression saisonnière, etc...* Les rayons lumineux créés artificiellement se rapprochent des rayons lumineux à large spectre émis par notre étoile, ... le soleil.

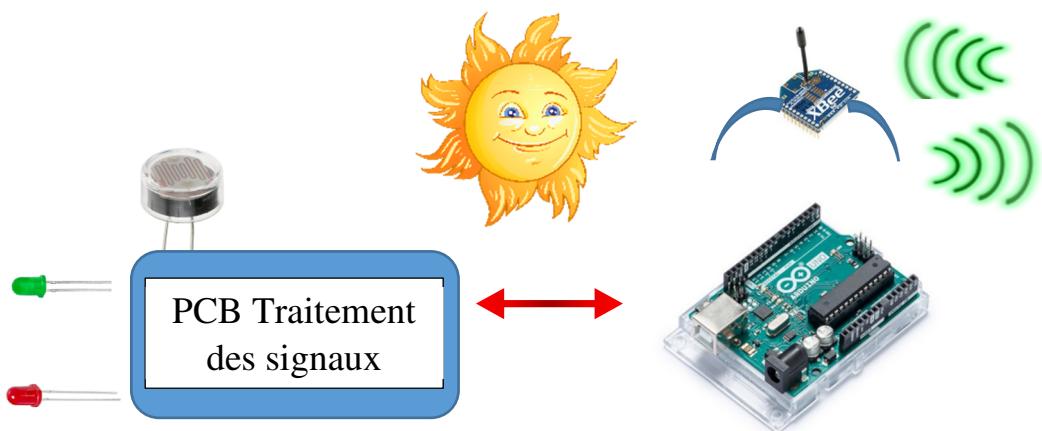


Figure 1 : Principe de la luminothérapie, émission de la consigne par ondes radio 2,4 GHz

L'objectif général est de concevoir **2** ensembles électroniques communiquant entre eux par ondes radio fréquence **Xbee (2,4GHz)**.

Un premier **ensemble** nous permettra de recevoir "*l'information lumineuse*" émise par les rayons solaires et de la transmettre par ondes radio, tandis que le second ensemble permettra de recevoir ces mêmes ondes radio, d'en décoder "*l'information lumineuse*" afin d'**augmenter ou de diminuer (moduler)** une source lumineuse artificielle pour disposer d'un éclairage permanent proche des rayons lumineux du soleil. Pour ce projet, **nous considérerons que** nous avons un ensemble proche d'une fenêtre (*lumière naturelle*), l'autre ensemble étant éloigné de plus d'une vingtaine de mètres dans une pièce ne recevant aucune lumière naturelle. Nous parlerons d'ensemble "**émission de la consigne**" pour l'un et pour le second ensemble "**station de base**".



Figure 2 : Module de transmission Xbee

Partie Hardware

Émission de la Consigne : équipé entre autres d'un montage **diviseur de tension**, dont l'élément central est une **photorésistance**, cf. Figure 3. Cette photorésistance, plus souvent appelée par son acronyme anglais **LDR** pour *Light Dependant Resistor* est un composant permettant de détecter la lumière (*attention sa réponse n'est pas quasi-linéaire*) ; de façon plus générale, c'est une résistance dont sa valeur ohmique et fonction de la quantité de lumière qu'elle reçoit sur une surface du composant prévue à cet effet.

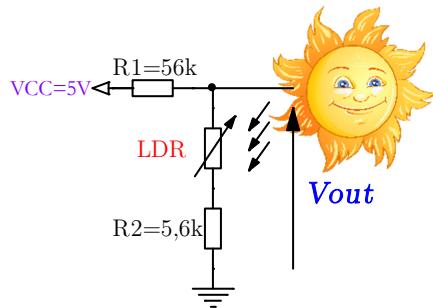


Figure 3 : Exemple typique de montage d'une **LDR**

A partir d'un montage diviseur de tension, il convient de **conditionner** et de **traiter** le signal. Cette grandeur **image** de la quantité de lumière présente sur le capteur **LDR** sera convertie en une valeur numérique. Par mesure de sécurité pour le **CAN** intégré au μC , vous réaliserez une fonction permettant de protéger son/ses entrée(s), la tension en entrée du **CAN** ne devant excéder une tension continue comprise [0, 5 Volts]. De plus, dans notre système, nous disposerons de deux diodes électroluminescentes en guise d'information visuelle indiquant le mode de fonctionnement du système.

Au cas où il y a utilisation d'amplificateurs opérationnels, leur alimentation sera de 0V, +8V. L'ensemble du montage sera réalisé sur une carte d'essai, qui couplé à un **Arduino Uno** associé au module **Xbee**, vous permettra grâce au software associé de transmettre l'image de *l'information* reçue sur la **LDR**.

Station de base : une carte d'essai rapide couplée quant à elle à la carte **Manuella** et d'un **Xbee** permettra de recevoir la consigne. À la suite des informations recueillies, vous contrôlerez une ampoule à incandescence. Pour des questions de sécurité, nous utiliserons certes une ampoule à incandescence mais fonctionnant sous 5V-100mW, et non une ampoule connectée sur le secteur **EDF** ($\sim 220V$ -75W). Vous devez donc concevoir une petite interface permettant de contrôler la puissance lumineuse de votre ampoule.

Partie Software

Émission de la consigne :

- à la première mise sous tension du système, **on transmet un message** pour signifier à la **station de base** que nous sommes en attente de mesures cohérentes ; en effet, il faut attendre un certain *laps de temps* pour que la valeur ohmique de la **LDR** se stabilise suivant la variation brutale de lumière. On considéra ce temps d'attente à **10s**, nous transmettrons donc la chaîne de caractères suivante "**Initialisation de système**",
- durant les **10s** d'attente et avant la 1^{ère} transmission de consigne, les diodes électroluminescentes **verte et rouge** clignotent alternativement à la fréquence de **2 Hz**,
- une fois cette première transmission réalisée, on réitère cette opération toutes les **15s**, et ce tant que nous sommes en **mode automatique**,
- quand le mode **automatique est activé**, la diode électroluminescente **verte** est **allumée en permanence**, si l'utilisateur passe en **mode manuel** via la **station de base** c'est la diode électroluminescente **rouge** qui **s'allume**,
- vous devez transmettre toutes ces informations avec le **format de trame** défini à la fin de ce sujet.

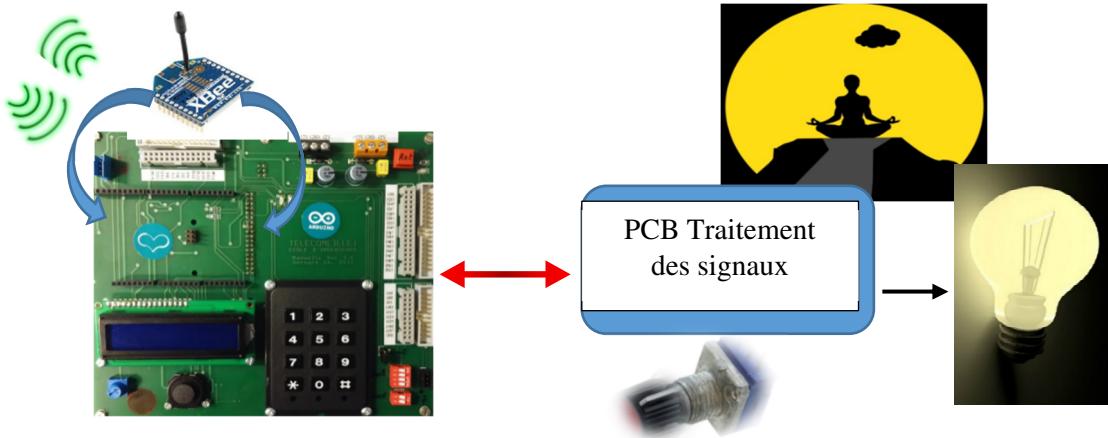


Figure 4 : Principe de la luminothérapie, station de base

Station de base mode automatique :

- à la mise sous tension du système, nous sommes **impérativement en mode automatique**, la station de base est donc en attente du message "**Initialisation du système**", qu'elle affiche sur les deux lignes de l'afficheur **LCD** de **Manuella**, message **clignotant à la fréquence de 1 Hz**,
- lors de cette phase, les deux diodes électroluminescentes **verte (D5)** et **orange (D4)** sur **Manuella** clignotent alternativement à la fréquence de **2 Hz**,

- une fois que la station de base reçoit le premier message qui renseigne sur l'éclairement de la **LDR**, on vient moduler la puissance lumineuse de l'ampoule en conséquence ; la diode électroluminescente **D5** reste **allumée**, tandis que **D4** est **éteinte**,
- en parallèle : sur la **1^{ère}** ligne du **LCD** de **Manuella**, le message suivant apparaît "**AUTO Tx écl. (%) :**", tandis que la **2^{ème}** ligne est réservée pour réaliser un **effet β** de **80 colonnes de pixels** qui s'illuminent progressivement en fonction **du taux d'éclairement de la LDR**, un appui (**$\leq 1,5s$**) sur la touche **"#"** du clavier numérique de **Manuella** bascule l'affichage "**effet β** " par un affichage décimal de type **wxy, z %** (*le chiffre des centaines n'est pas indiqué en dessous de 100*), ce mode d'affichage bascule au grès de l'appui sur la touche **#** mais revient de lui-même au bout de **5s** sur un affichage de type "**effet β** ".

Station de base mode manuel :

- un appui (**$\leq 1,5s$**) sur la touche **"*"** du clavier numérique fait passer la station de base en mode manuel,
- la diode électroluminescente **orange D4 est allumée, D5 est éteinte**,
- **envoi d'une trame** pour inverser l'allumage des diodes électroluminescentes sur l'ensemble **émission de la consigne**,
- lors de cette phase, les deux diodes électroluminescentes **verte (D5)** et **orange (D4)** sur **Manuella** clignotent en même temps à la fréquence de **1 Hz**,
- sur la **1^{ère}** ligne du **LCD** de **Manuella**, le message suivant apparaît "**MANU Tx écl. (%) :**" tandis que la **2^{ème}** ligne est réservée pour un **effet β** mais de 60 colonnes de pixels s'illuminant progressivement **en fonction d'un nombre entier à 3 chiffres** que l'utilisateur saisit au préalable à l'aide du clavier numérique (*exemple : 082 → pour 82%*).

Christophe BERNARD

IMT Nord-Europe

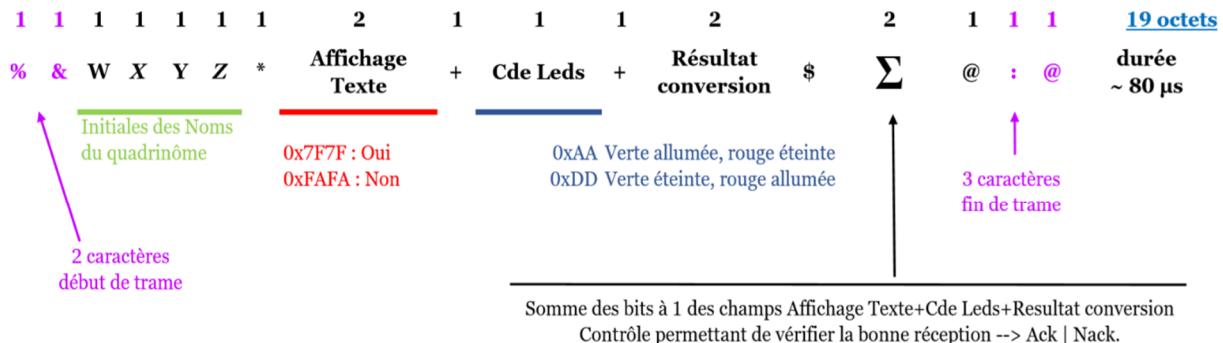


MOTS CLÉS - À DECOUVRIR

LDR, Arduino, conversion Analogique- Numérique, principe de la PWM (Pulse Width Modulation), XBee, etc...

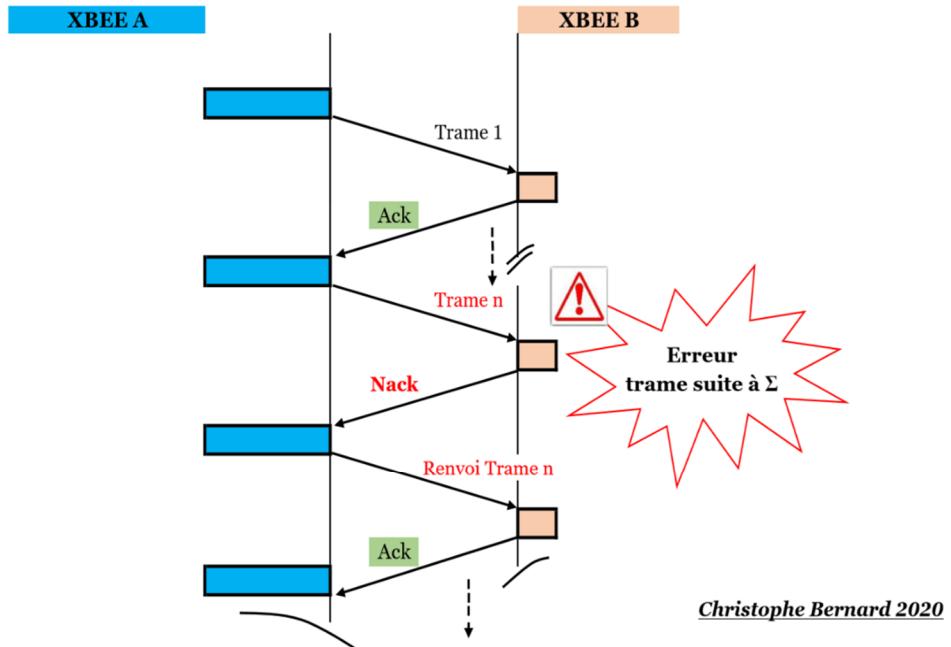
Format de la trame pour envoyer/recevoir sur les modules XBEE

LUMINOTHERAPIE



Exemple : [%&WXYZ*64250+170+511\\$25@:@](#), pas d'affichage texte, verte allumée, rouge éteinte, résultat conversion 511 somme des bits à 1 : 25

Contrôle d'envoi de trame "Send & Acknowledge"



NACK : Permet de définir que l'ensemble de la trame reçue n'est pas correcte caractère utilisé : ?, trame &%WXYZ?:@:

ACK : Permet de définir que l'ensemble de la trame reçue est correcte caractère utilisé : !, trame &%WXYZ!:@:

* **ACK**nowledge or **Not ACK**nowlege pour acquittement ou non (un accusé)

Robot contrôlé par radio fréquence 2,4 GHz

Christophe BERNARD - Sujet 4, 5 élèves

13

Objectifs

L'objectif de ce sujet, est de concevoir 2 ensembles électroniques communiquant entre eux. Cette communication permettra le contrôle d'un petit robot mû par deux moteurs à courant continu. La communication s'effectue via une liaison hertzienne **2,4 GHz** protocole **XBEE** 802.15.4, débits de 250 Kbits/s.



Exemple de module Xbee

Les deux ensembles permettront de :

- contrôler les interfaces, d'encoder/décoder les ordres émis/reçus, et ce via le programme adéquat. Nous parlerons de **Radio commande** pour l'ensemble **émetteur/récepteur** et simplement de **récepteur** pour l'ensemble embarqué sur le châssis du robot.

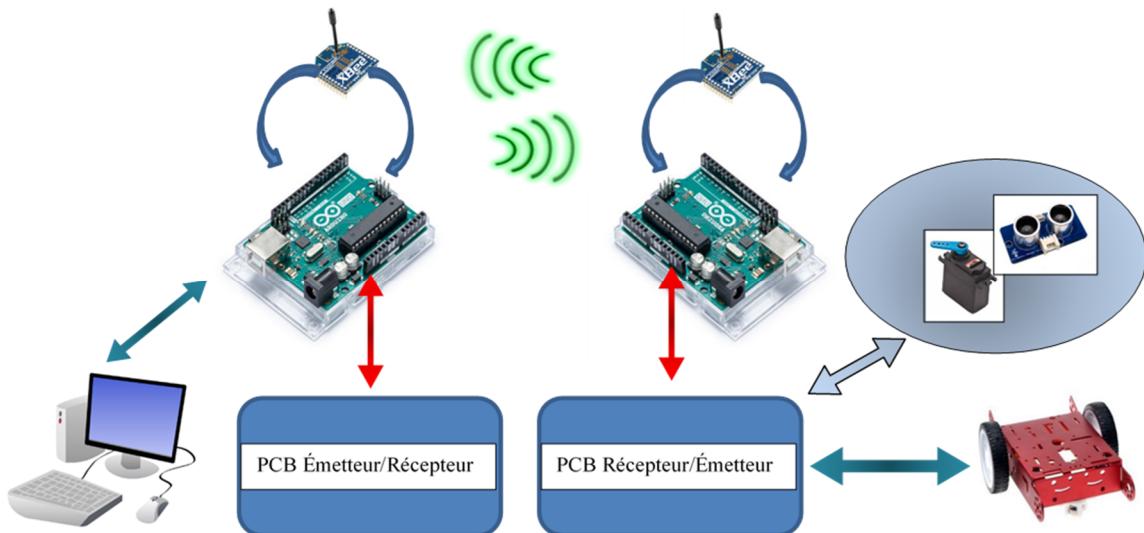


Figure 1 : Principe du robot contrôlé par liaison hertzienne 2,4 GHz

Hardware

Radio commande : couplé à un **Arduino Uno** et module **Xbee**, il sera équipé d'un montage potentiométrique fournissant les ordres progressifs de déplacement du robot en marche avant et arrière. Quant aux ordres droite et gauche, ils seront réalisés à l'aide de deux boutons poussoirs. Des Leds de visualisation affectée à chacune des actions sont à prévoir permettant ainsi de

visualiser l'ordre émis. L'information déplacement progressif avant/arrière (*vitesse*) sera visualisable sur un **bagraph** composé de **dix leds**. Toutes les actions de contrôle seront donc **transmises** à votre Arduino et envoyées au robot via le module **Xbee**.

Récepteur : l'ensemble est aussi composé d'un **Arduino Uno** (ou éventuellement un **Arduino Nano Every** ... *à voir ?*) et d'un module **Xbee**. Cet ensemble doit réaliser le contrôle de deux moteurs et ce de façon indépendante via une interface de puissance, de plus une visualisation par **2 Leds bi-couleurs**, nous permettra de savoir l'état des moteurs (*fonctionnement ou non, moteur gauche/moteur droit*).

Prévoir un **bouton poussoir** arrêt d'urgence ainsi qu'un système permettant à l'utilisateur de mettre le robot en mode fonctionnement **autonome ou radio-contrôlé**. L'interface **détection** d'obstacles / télémétrie est composée d'un module de **détection à ultrasons**, le tout fixé sur l'avant du robot. Prévoir à cette effet, un dispositif permettant de produire un son en cas d'obstacle à une certaine distance définie. Toutes les actions de contrôle **reçues** via l'interface **Xbee** seront donc traitées par *l'Arduino* et envoyées aux interfaces permettant ainsi le déplacement du robot. L'alimentation de l'ensemble se fera via une batterie style « Power Bank » **5V – 2,4A max.**

Software

Émetteur - radio-contrôlé :

- à la mise sous tension on attend une trame "**Ready**", signalée par **effet φ** d'une durée de 4s sur le bagraph (*10 leds*) indiquant à l'utilisateur que le système est prêt,
- le système *envoie une trame toute les x millisecondes* (*à voir*) au récepteur (*cf. Format de trame*), trame composée des instructions pour contrôler le robot,
- lorsque le potentiomètre (*mono-tour*) permettant de mouvoir votre robot est dans sa position **médiane** : votre robot est à l'**arrêt** ; cet état est visualisable par une alternance de fonctionnement des **leds centrales** (*leds 5, 6*) de votre bagraph à une fréquence de **1 Hz**,
- en cas d'appui **maintenu** sur **un** des boutons poussoirs **droite ou gauche, et si** le potentiomètre est **toujours** en position médiane alors votre robot tourne sur lui-même à **vitesse constante** : cette action est visualisable en **permanence** par la *led* dédiée la plus à droite **ou** la plus à gauche (*leds 1, 10*),
- une rotation du potentiomètre de la **position médiane vers la droite** permet au robot de se déplacer vers **l'avant, et ce**, avec une **vitesse évoluant en fonction de la position du potentiomètre** (*position maximale du potentiomètre = vitesse maximale*) : cette action de "*vitesse progressive avant*" sera quant à elle visualisable sur le bagraph par **effet β** de la *led 7* (*inclus*) vers la *led 9* (*leds 7, 8, 9 allumées = vitesse maximale*). À partir du moment où la

marche avant est enclenchée, la *led* affectée à cette action reste allumée jusqu'à retour en position médiane du potentiomètre,

- **de même**, une rotation du potentiomètre de la *position médiane vers la gauche* permet au robot de se déplacer vers *l'arrière, et ce*, avec une *vitesse évoluant en fonction de la position du potentiomètre* (*position minimale du potentiomètre = vitesse maximale*) : cette action de "vitesse progressive arrière" sera quant à visualisable sur le bargraphe par *effet β inverse* de la *led 4 (inclus)* vers la *led 2 (leds 4, 3, 2 allumées = vitesse maximale)* ; tout comme la marche avant, dès que la marche arrière est enclenchée, la led affectée à cette action reste allumée,

- Dans les deux cas précédents, il va de soi qu'une **action maintenue** sur les pousoirs droite **ou** gauche, permettra au robot d'avancer ou de reculer vers la droite **ou** vers la gauche. Attention toutefois à la vitesse de déplacement lorsque l'on tourne ☺ !

- l'utilisateur est prévenu par un **son émis par le robot** si un obstacle se présente à une distance **≤ 25cm**, l'utilisateur prendra donc les mesures nécessaires afin d'éviter toute collision... *attention à votre bonus*,

- lors de la réception d'une trame "**Urgency**" toutes les *leds* du dispositif *clignotent* à la fréquence de **4 Hz** pendant **5s**, permettant ainsi à l'utilisateur de remettre à **zéro** les commandes (*position médiane du potentiomètre*) ; **5s** après cette remise à zéro, on reçoit de nouveau la trame "**Ready**".

Récepteur en mode radio-contrôlé :

- à la mise sous tension le système envoie une trame "**Ready**", les moteurs étant en mode "**arrêt**" les **2 led vertes** sont allumées (*led bi-couleur*). À partir de ce moment, il est en attente de trames pour effectuer les ordres reçus (*les déplacements*),

- l'information "**mise en fonctionnement d'un/des moteur(s)**" est signalée(s) par l'allumage de la/les *led(s)* "**rouge(s)**" et la/les *led(s)* "**verte(s)**" pour l'arrêt,

- dès la mise sous tension le dispositif de télémétrie est en fonctionnement par une fonction **non bloquante**,

- le module à ultrasons informe le programme si un obstacle est détecté à une distance **≤ 25 cm**, ce qui a pour but de *déclencher le dispositif sonore*, émission *permanente* d'un **son strident** (sauf si l'obstacle est de nouveau à une distance **≥ 20 cm**),

- un déclenchement de l'arrêt d'urgence *stoppe toutes actions en cours*, envoie une "**Urgency**" à l'émetteur : le système se réinitialise et envoie une trame "**Ready**" **9 s** plus tard.

Récepteur en mode autonome :

Pour des raisons de simplicité au niveau algorithme, on se propose d'évoluer dans un environnement où tous les objets sont de formes rectangulaires de dimensions 32*21,5*10 cm (couvercle d'emballage des ramettes de feuilles A4) dont la face la plus importante est posée au sol. Les objets seront espacés entre eux d'une distance minimale équivalente au plus grand côté du parallélépipède.

Dans ce mode :

- dès la mise sous tension, le robot émet 5 bips de 1s, et commence à évoluer dans son environnement,
- on supposera un déplacement *droite/gauche de 90° ou 45°* à vitesse moyenne constante,
- dès qu'un objet est détecté à distance *égale à 20 cm*, on engage un déplacement *à droite ou à gauche de 90°*, suivi d'un déplacement en *marche avant de 35cm*, pour de nouveau enclencher un déplacement *à gauche ou à droite de 90°*, suivi enfin d'un d'une *marche avant de 55 cm*.

Christophe BERNARD

IMT Nord-Europe

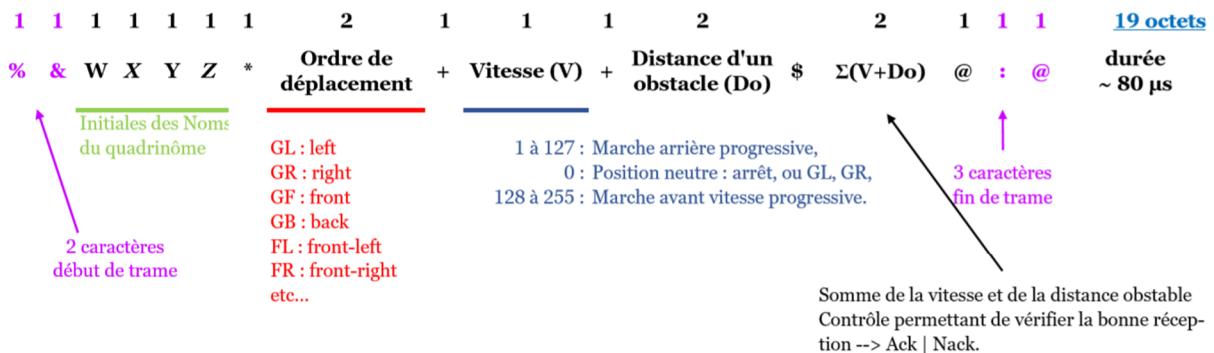


MOTS CLÉS - À DECOUVRIR

Arduino, moteur à courant continu, pont en H, L293E, principe de la PWM (Pulse Width Modulation), Ne555, servo-moteur, XBee, principe de la détection par ultrasons, processing. Prévoir Tests Moteur, circonférence roue, etc...

Format de la trame pour envoyer/recevoir sur les modules XBEE

Mode radio-controlé et autonome



Exemple : %&WXYZ*RB+064+328\$528@:@, déplacement en marche arrière-droite, vitesse médiane, objet à 328 cm

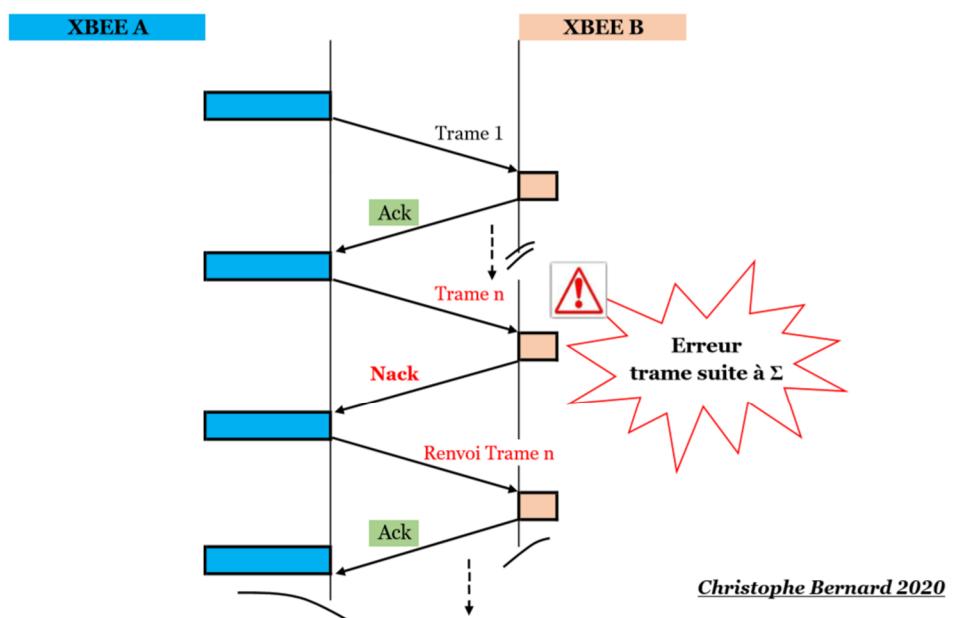
Trame 'Ready'

1 1 1 1 1 1 1 2 1 1 1 12 octets
% & W X Y Z * GO @ : @ ~ 48 µs

Trame 'Urgency'

1 1 1 1 1 1 1 2 1 1 1 12 octets
% & W X Y Z * ST @ : @ ~ 48 µs

Contrôle d'envoi de trame "Send & Acknowledge"



NACK : Permet de définir que l'ensemble de la trame reçue n'est pas correcte caractère utilisé : ?, trame %&WXYZ?@:@

ACK : Permet de définir que l'ensemble de la trame reçue est correcte caractère utilisé : !, trame %&WXYZ!@:@

* ACKnowlede or NOTACKnowlege pour acquittement ou non (un accusé)