Université de Mons
Faculté des Sciences
Département d'Informatique
Réseaux et Télécommunications

Conception et évaluation d'une architecture hybride de réseaux de capteurs reposant sur les technologies radio LoRa et IEEE 802.15.4

Directeur : Mr Bruno QUOITIN

Rapporteurs : Mr Prénom NOM
Mr Prénom NOM

Mémoire réalisé par
Arnaud PALGEN

en vue de l'obtention du grade de
Master en Sciences Informatiques

UMONS
Université de Mons

Faculté
des Sciences

Année académique 2020-2021

Remerciements

Nous remercions ...

Table des matières

Table des figures

TODO: hello

0.1 Plateformes de développement

Zolertia RE-Mote

Pour ce mémoire, la plateforme Zolertia RE-Mote revision B(Fig. 1) est utilisée.

Cette plateforme, basée sur un system on chip (SoC) CC2538 ARM Cortex-M3, a été conçue par des universités et des industriels dans le but de permettre aux chercheurs et makers de développer des applications IoT et des objets connectés.

Le Zolertia RE-Mote a été choisi car elle est équipée de deux radios compatibles IEEE 802.15.4, permet une consommation électrique faible et possède de nombreux pins de connexion qui peuvent être utilisés pour y connecter des capteurs, actuateurs, radios, etc.

Le prix du consturcteur pour cette plateforme est de 93,95€ [?].

FIGURE 1 – Zolertia RE-Mote révision B [?].

La table 1 reprend les principales spécifications du Zolertia RE-Mote rev.b et sa table ... la consommation électique.

Element	Spécification
Radio	Deux radios IEEE 802.15.4 à 2.4 GHz et 863-950 MHz
CPU	ARM® Cortex® -M3 jusqu'à 32 MHz
RAM	32 KB (16 KB pour tous les Power Modes)
Flash programmable	512KB
I/O	RGB led, boutton user et reset, USB 2.0 à 12Mbps, Real-Time Clock

TABLE 1 – Spécifications du Zolertia RE-Mote rev.b [**?**].
reprend la consommation de courant du RN2483

RN2483

Le RN2483 (Fig. 2) est un modem LoRa compatible LoRaWAN^{TM} basse énergie. La communication avec ce modem ce fait par des commandes ASCII envoyée via une interface UART. Il prend en charge les modulations FSK, GFSK et LoRa. Il possède également 14 GPIOs ? ?pour le contrôle et le status, partagés avec 14 inputs analogiques. ? ?Ses fréquences opérationnelles sont situées dans les bandes de fréquences 433 MHz et 868 MHz. D'après la datasheet, sa portée maximale est de 15km en agglomération et 5km en zone urbaine. Comme l'illustre la figure. 2, pour ce mémoire, le RN2483 a été monté sur un carte d'interface réalisée par B.Quoitin qui comporte deux leds, une petite antenne ainsi que les connecteurs permettant d'utiliser des câbles de prototypages.

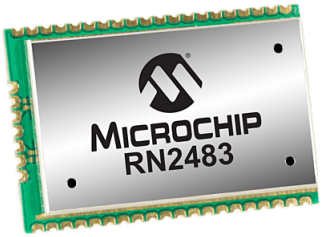

FIGURE 2 – RN2483 [**?**].

La figure 3 reprend le schéma-bloc du RN2483. Il contient notemment l'interface UART, les antennes 433 MHz et 868Mhz ainsi que les GPIOs et la stack LoRaWan.

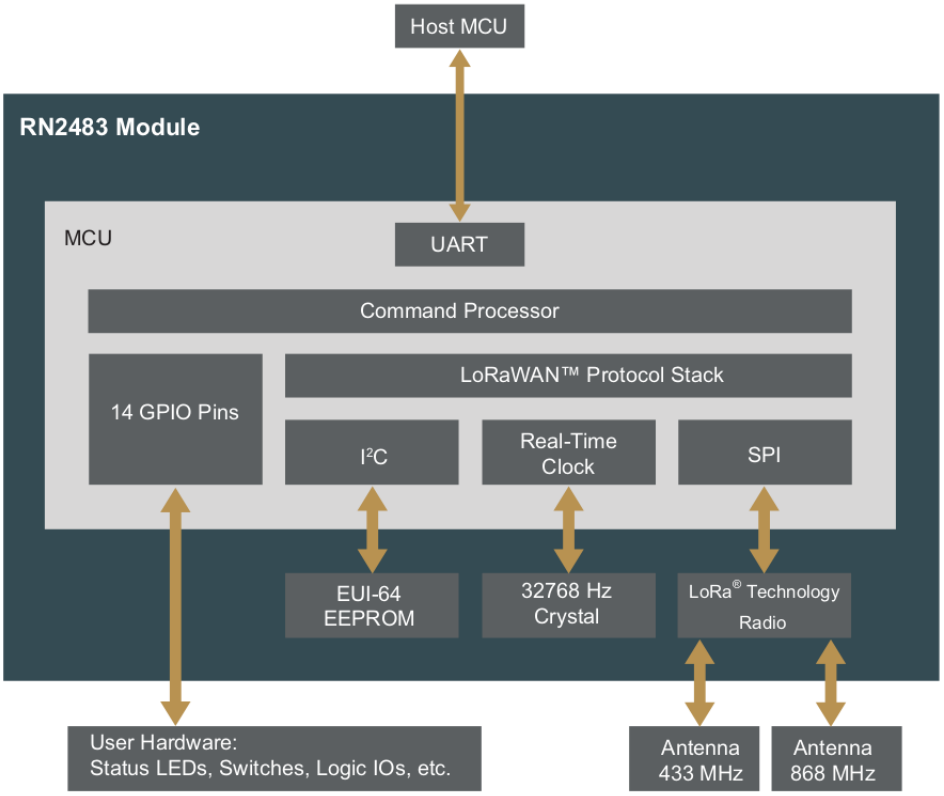

FIGURE 3 – Schéma-bloc du RN2483 [?].

La table 2 reprend la consommation électrique du RN2483 en fonction de son mode de fonctionnement.

Mode	Courant (mA)		
	VDD = 2.1V	VDD = 3.3V	VDD = 3.6V
Idle	1.7	2.8	3.1
Transmit	28.6	38.9	44.5
Sleep	0.0015	0.0016	0.0016
Receive	12.96	14.22	14.69

TABLE 2 – Consommation de courant (à 25 °C) [?].

Raspberry Pi

Le Raspberri Pi est un ordinateur monocarte. Le modèle utilisé pour ce projet est un Raspberry Pi 3 modèle B+ (Fig. 4). La table 3 reprend les principales caractéristiques de ce modèle.

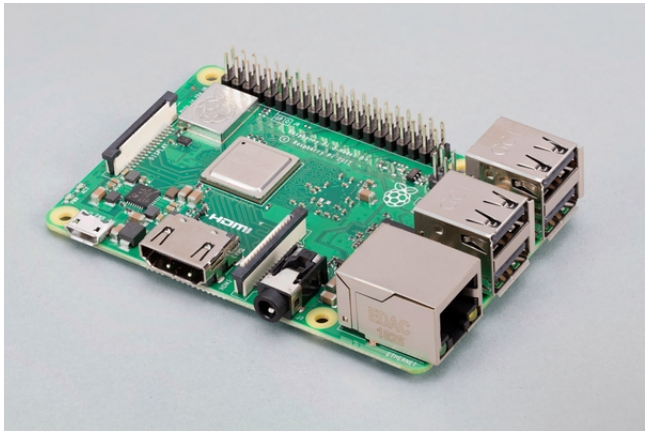

FIGURE 4 – Raspberry Pi 3B+ [?].

Element	Spécification
CPU	Broadcom BCM2837B0, Cortex-A53 64-bit SoC à 1.4GHz
Mémoire	1GB LPDDR2 SDRAM
Connectivité	• IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE • Gigabit Ethernet over USB 2.0 • 4 × USB 2.0 ports
Alimentation	5V/2.5A DC

TABLE 3 – Spécifications du Raspberry Pi 3B+ [?].

0.2 RTOS

Un RTOS (Real Time Operating System) est un sytème d'exploitation temps réels principalement destiné aux systèmes embarqués.

Etant donné que ce projet utilise le protocole 802.15.4 ainsi que TSCH, le RTOS choisis doit les prendre en charge ainsi qu'un ou plusieurs algorithme d'ordonnancement de TSCH

Il est également préférable, que le RTOS choisis, supporte déjà la plateforme utilisée. Une implémentation de la LoRa n'est pas nécessaire car les communications Lora sont réalisées via le RN2483 qui est controlé par UART.

Pour effectuer ce choix, les RTOS suivants ont été comparés : Contiki OS, FreeRTOS, RIOT OS et Zephyr.

Contiki OS

Le développement publique de Contiki a débuté en octobre 2012 [1] Dans un premier repository : contiki [**?**]. Un nouveau développement a démarré en mai 2017 sous le nom de Contiki-NG [**?**]. C'est donc ce dernier qui est utilisé pour la comparaison est qui est dénommé par la suite Contiki.

Cet OS open-source et multi-plateforme implémente toute une série de protocoles de communications basse énergie tels que IEEE 802.15.4, 6TiSCH, IPv6/6LoWPAN et RPL. En plus de 6TiSCH, un ordonnanceur TSCH, Orchestra, est implémenté. TODO udp/tcp

Contiki est également compatible avec le Zolertia RE-Mote. Il est accompagné de Cooja, un simulateur réseau qui permet de simuler les communications entre plusieurs noeuds utilisant Contiki.

FreeRTOS

D'après le site officiel de FreeRTOS [**?**], ce RTOS est développé depuis 15 ans. TODO compatible avec zolertia ?

RIOT OS

Le développement publique de RIOT a débuté en décembre 2012[1].

D'après le site officiel, cet OS supporte 229 carte de développement et 64 CPU dont le Zolertia RE-Mote.

La pile réseau de RIOT comporte les protocoles notemment 6LoWPAN, IPv6, RPL, LoRaWan, 802.15.4. TODO openwsn

1. Date de création du repository Github.

Zephyr

TODO pas supporté tiny os plus de commit depuis 1 an. stack réseau ? supporté ?

Le table 4 résume la comparaison de ces RTOS.

Le RTOS choisi est Contiki OS. Il a été choisi pour sa maturié, la prise en charge du Zolertia RE-Mote et sa pile réseau complète.

RTOS	802.15.4	ord. TSCH	LoRa	IPv6	routage IP	comp.
Contiki OS	√	6Tisch, Orchestra	Projet KRATOS	√	RPL	√
FreeRTOS	×	×	√	√	×	×
RIOT OS	√	×	√	√	RPL	√
Zephyr	√	×	√	√	Thread	×

TABLE 4 – Comparatif de différents RTOS.

0.3 RPL

Routing Protocol for Low-Power and Lossy Networks (RPL) est un protocole de routage IPv6 destiné aux réseaux dont les noeuds sont contraints en énergie et dont les liens entre ces noeuds sont soumis à des pertes importantes de paquets (Low-power and Lossy Networks (LLNs)). Ce protocole à vecteur de distance est un protocole proactif, c'est à dire que les routes sont établies avant qu'elles ne soient nécessaires.

RPL sépare le traitement et la transmission des paquets de l'optimisation de l'objectif de routage. Cela permet de l'adapter à un large éventail d'applications des LLNs.

Topologie

La topologie utilisée par RPL est le DODAG (Destination Oriented Dag). Un DODAG est un graphe dirigé acyclique (DAG) ayant une seule racine (Fig. 5). De part cette architecture RPL est adapté aux applcations de collecte de données ce qui est le principale objectif de ce projet. En effet, la route d'un noeud à la racine est facilement établie car tous les noeuds du chemins envoient les données à leur parent.

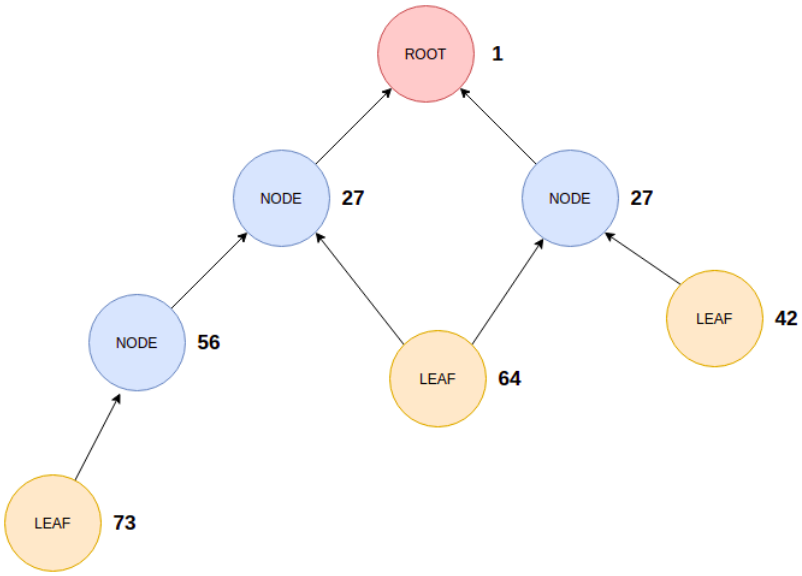

FIGURE 5 – DODAG.

Fonctions objectif

Une fonction objectif (OF) défini comment plusieurs métriques sont utilisées pour calculer le rang d'un noeud. Le rang d'un noeud détermine sa position dans le DODAG par rapport aux autres noeuds. Le rang augmente stictement dans le sens descendant et diminue strictement dans le sens montant. Ainsi, pour un noeud n, $rang(n) > rang(parent(n))$. La figure 5 illustre un DODAG avec des valeurs de rang fictives attribuées aux noeuds. Sur cet exemple, la racine du DODAG a comme rang, la valeur par défaut ROOT_RANK définie dans le RFC.

Cette section décris brièvement deux fonctions objectif implémentées dans Contiki : OF0 et MRHOF.

- **OF0**

  Objective Function Zero est une fonction dont l'objectif est de choisir un parent qui permettra à un noeud d'avoir la racine du DODAG le plus proche possible. Le rang d'un noeud $R(N)$ est calculé comme suit :

  $$ranck\_increase = (Rf * Sp + Sr) * MinHopRankIncrease$$

  $$R(N) = R(P) + ranck\_increase$$

  où
  - Rf est le $rank\_factor$ et $Sr \leq stretch\_of\_rank$ avec $rank\_factor$ et $stretch\_of\_rank$ deux paramètres de la fonction
  - Sp est le $step\_of\_rank$ qui est une valeur basée sur les propriétés du lien
  - $MinHopRankIncrease$ est une constante
  - $R(p)$ est le rang d'un parent p

  $R(N)$ est calculé pour chaque parent potentiel. Le parent potentiel qui implique le plus petit $R(N)$ sera choisi.

- **MRHOF**

  Minimum Rank with Hysteresis Objective Function est une fonction dont l'objectif est de

Messages RPL

DIO

Les DIOs (DODAG Information Object) annoncent des informations sur le DODAG qui permettent aux noeuds de découvrir une instant RPL, de sélectionner un parent ou encore de maintenir le DODAG. Un DIO, illustré à la figure 6, inclu notemment les champs suivants :

- InstanceID : Identifie l'instance RPL du DODAG
- Rank : Le rang du noeud qui émet le DIO
- MOP (Mode of Operation) : Le mode d'opération de l'instance RPL **TODO: c.f. SECTION**
  0. Pas de routes descendantes
  1. Non-storing mode
  2. Storing mode sans multicast
  3. Storing mode avec multicast
- DODAGID : Adresse IPv6 définie par la racine qui identifie le DODAG

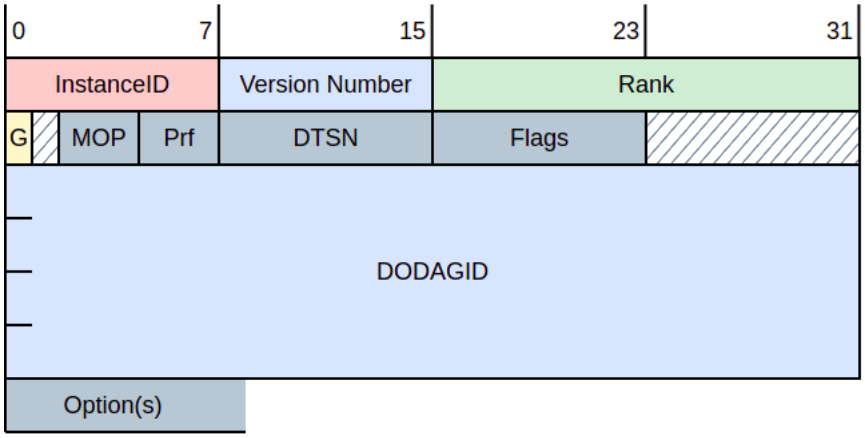

FIGURE 6 – DODAG Information Object.

DIS

Les DODAG Information Solocitation (DIS) sont utilisés pour solicité un DIO d'un noeud RPL.

DAO

Les DAOs (Destination Advertisement Object) sont utilisés pour établir les routes descendantes. Ils sont donc envoyés vers le haut du DODAG. Le

format du DAO est illustré à la figure 7. Un DAO est composé des champs suivants :

- InstanceID : identifie l'instance du DODAG
- K : indique que le destinataire doit répondre avec un DAO-ACK
- D : indique que le DODAGID est présent
- DAOSequence : Incrémenté à chaque DAO d'un noeud et répété dans le DAO-ACK
- DODAGID (optionnel)

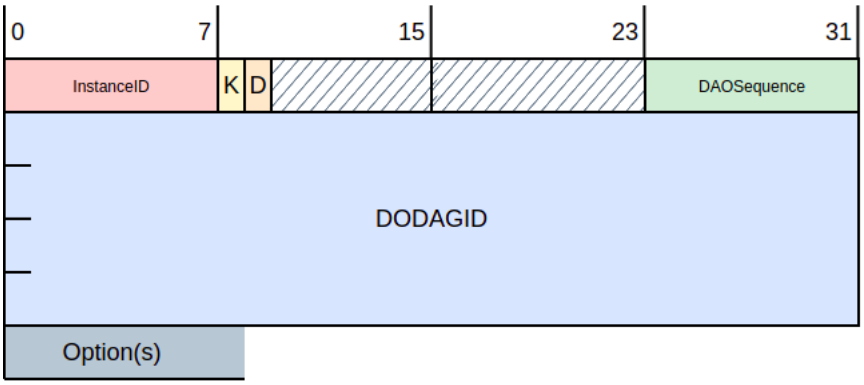

FIGURE 7 – DODAG Information Object.

DAO-ACK

Le format d'un DAO-ACK est similaire à celui d'un DAO. Il n'est donc pas utile de le décrire.

Construction du réseau

Modes de fonctionnements

Discussion