

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique

Spécialité Électronique des Systèmes Embarqués

présenté par

Semmar Belkacem

&

Lehachi Aimed Eddine

Conception et réalisation d'un module de communication radio longue portée

Proposé par : Benrezki Rabei Riadh & Naceur Djamilia

Année Universitaire 2021-2022

Remerciements

On remercie Dieu le tout puissant de nous donner la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de M. Benrezki.R, on le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Nos remerciements s'adressent à Mme Naceur.D pour son aide pratique et son soutien moral et ses encouragements.

Nos remerciements s'adressent également à tous nos professeurs pour leur générosité et la grande patience dont ils ont su faire preuve malgré leurs charges académiques et professionnelles.

Je remercie également ma famille : mon père pour son soutien et sa présence pour me clarifier toutes les notions qui me sont inconnu, ma mère pour sa tendresse et mes deux sœurs qui ont supporté ma folie et mes sauts d'humour tout au long de la réalisation de ce modeste travail.

Merci d'être là, à mes cotes, au moment où j'en avais besoin.

Merci d'être si compréhensifs, si intelligents et si présents dans ma vie.

Je suis reconnaissant de l'accueil de M. Lafri.A, M. Rebika.S, M. Bouanzoul.M et M. Taleb.A au sein de la société DOOFAS Innovative Solutions. Je salue leur amabilité, leur patience, leur souplesse d'esprit et leur savoir.

Je dédie un merci à M. Mezzah.I un chercheur au niveau du Centre de Développement des Technologies Avancées, pour son aide à travers ses remarques pertinentes.

Semmar Belkacem.

Mes vifs remerciements vont dans un premier temps à notre encadreur mr R.Benrezki, et notre Copromotrice mme D.Naceur leurs soutient quotidien, et leurs confiance.

Je remercie également mes oncles H.Lehachi et K.Lehachi qui ont contribué à l'acheminement des cartes utilisées dans ce projet.

Je remercie aussi mr Mezzah.I de nous avoir donné de précieux conseils concernant le fonctionnement du module radio, et également mr Lafri Abdelmoumen pour son accueil a DOOFAS avec un grand respect.

Mes dédicaces vont aussi à mes chers parents qui m'ont supporté durant toute ma carrière d'ingénieur.

Lehachi Aïmed Eddine

ملخص:

في غالبية المنشآت الصناعية، يجب أن تكون الترابطات بين العناصر المختلفة لنظام المراقبة والتحكم عن بعد كافية من أجل تلبية موضوع نقل البيانات بشكل كامل.

هذا التحدي الذي رفعه مركز تطوير التكنولوجيات المتطورة CDTA بالتعاون مع شركة DOOFAS العاملة في مجال التركيبات الكهربائية لمحطات ضخ المياه.

الهدف من عملنا هو تصميم وحدة تواصل بالتردد اللاسلكية، ذات مدى طويل تعمل حول نطاق 433 ميغا هرتز، المجاني وواسع الانتشار في البيئة الصناعية.

الكلمات المفتاحية: وحدة الاتصالات، التسيير عن بعد، نظام التحكم الإشرافي والحصول على البيانات، وحدة التحكم عن بعد، الدائرة المتكاملة CC1120.

Résumé :

Dans la majorité des installations industrielles, les interconnexions entre les différents éléments d'un système de télésurveillance et de commande doivent être en adéquation afin de satisfaire pleinement la thématique de transport de données.

Ce défi est relevé par le CDTA en collaboration avec la société DOOFAS activant dans le domaine des installations électriques appliqué aux stations de pompage d'eau.

Le but de notre travail est la conception d'un module RF longue portée fonctionnant autour de la bande ISM (433MHZ), gratuite et très répandue dans le milieu industriel.

Mots clés : Module de communication ; télégestion ; SCADA ; RTU ; le circuit intégré CC1120.

Abstract: In the majority of industrial installations, the interconnections between the various components of a remote monitoring and control system must be adequate in order to fully satisfy the data transport theme.

This challenge is raised by the CDTA in collaboration with the DOOFAS, a company active in the field of electrical installations for water pumping stations.

The aim of our work is for the design of a long-range RF module operating around the ISM band (433MHZ), which is free and widely used in industry.

Keywords : Communication module ; Remote monitoring ; SCADA ; RTU ; CC1120 IC.

Listes des acronymes et abréviations

6LoWPAN: IPv6 Low power Wireless
Personal Area Networks

AC: Alternating Current

ADC: Analogue to Digital Converter

ADR: Adaptive Data Rate

AES: Advanced Encryption Standard

AGC: Automatic Gain Control

AHB: Advanced High Performance
Performance Bus

AN: analog

APB: Advanced Peripheral Bus

API: Application Programming Interface

Arm: Advanced RISC Machines/ Acorn RISC
Machine

ASCII: American Standard Code for
Information Interchange

ASK: Amplitude Shift Keying

AWG: American Wire Gauge

BALUN: Balanced/ Unbalanced

BOM: Bill of Materials

BPSK: Binary Phase Shift Keying Modulation

BW: Band Width

CAN: Controller Area Network

CAO: Computer Aided Office

CDT: C/C++ Development Tooling

CDTA : Centre de Développement des
Technologies Avancées

CEI: Commission Electrotechnique
Internationale

CMOS: Complementary Metal Oxide
Semiconductor

COM: Communication

CPHA: Clock Phase

CPOL: Clock Polarity

CR LF: Carriage Return, Line Feed

CRC: Cyclic redundancy check	I/O: Input/ Output
CTS: Clear To Send	I2C: Inter-Integrated Circuit
DAC: Digital to Analog Converter	IDE: Integrated Development Environment
dBm: decibel-milliwatts	IdO: Internet des Objets
DC: Direct Current	IEC: International Electrotechnical Commission
DE: Driver Enable	IEEE: Institute of Electrical and Electronics Engineers
DFSDM: Digital Filter for Sigma-Delta Modulators	IF: Intermediate-frequency
DR: Data Rate	IHM: Interface Homme-Machine
DSP: Digital Signal Processor	Inc.: Incorporated
DSSS: Direct-sequence spread spectrum	INT: Hardware Interrupt
E/S: Entrée / Sortie	IoT: Internet of Things
FIFO: First In First Out	IP: Internet Protocol
FPGA: Field Programmable Gate Array	IrDA: Infrared Data Association
FPU: Floating Point Unit	ISA: International Society of Automation
FSK: Frequency Shift Keying	ISM: Industrial, Scientific and Medical band
FSMC: Flexible Static Memory Controller	ISO: International Organization for Standardization
GCC: GNU Compiler Collection	JLC: Shenzhen JIALICHUANG Electronic Technology Development Co., Ltd.
GDB: GNU Debugger	Kbps: Kilobits Per Second
GFSK: Gaussian Frequency Shift Keying	Ko: Kilo octet
GND: Ground	LD: Ladder Diagram
GPIO: General Purpose Input/Output	LED: Light emitting diode
HART: Highway Addressable Remote Transducer	
HTTP: Hypertext Transfer Protocol	

LIN: Local Interconnect Network	OPC-UA: Open Platform Communications Unified Architecture
LNA: Low-noise amplifier	OSI: Open Systems Interconnection
LoRaWAN: Long Range Wide Area Network	OTG: On The Go adapter
LPRF: Low Power Radio Frequency	PAC: Programmable Automation Controller
LPWA: Low Power Wide Area	PC: Personal Computer
LSB: Least Significant Bit	PCB: Printed Circuit Board
mA: Milliampère	PCLK: Peripheral Clock
MARC: Main Radio Control Unit	PHY: Physical Layer
MBAP: ModBus Application Protocol Header	PI : PROFIBUS & PROFINET Intrnational
MCU: Microcontroller	PLC: Programmable logic controller
MHz: mégahertz	PWM: Pulse Width Modulation
MISO: Master Input Slave Output	Rev: Revision
Mo: Méga octet	RF: Radio frequency
ModBus: Modicon Bus	RISC: Reduced Instruction Set Computer
MOSI: Master Output Slave Input	RS-232: Recommended Standard 232
MPU: Microprocessor Unit	RS-422: Recommended Standard 422
MQTT: Message Queuing Telemetry Transport	RS-485: Recommended Standard 285
MS/s: Mega Simple per second	RSSI: Received Signal Strength Indicator
MSB: Most Significant Bit	RST: Reset
Msp/s: Mega-Symbols per Second	RTC: Real-Time Clock
National Institute of Standards and Technology	RTS: Request To Send
NSS: Slave Select	RTU: Remote Terminal Unit
OOK: On-off Keying	RX: Reception

SAI: Serial Audio Interface

SCADA: Supervisory Control and Data Acquisition

SCK/SCLK: Serial Clock

SDI: Serial Data In

SDMMC: Secure Digital and MultiMedia Card

SDO: Serial Data Out

SI: Slave In

SNMP: Simple Network Management Protocol

SO: Slave Out

SPI: Serial Peripheral Interface

SRAM: Static Random Access Memory

SS/CS: Chip Select

ST: Structured Text

SWCK: Serial Wire Clock

SWDIO: Serial Wire Data I/O

SWPMI: Single Wire Protocol Master Interface

TCP/IP: Transmission Control Protocol/Internet Protocol

TCP: Transmission Control Protocol

TCXO: Temperature Compensated Xtal (Crystal) Oscillator.

THT: Through Hole Technology

TI: Texas Instruments

TX: Transmission

UART: Universal Asynchronous Receiver Transmitter

UDP: User Datagram Protocol

UNB: Ultra Narrow Band

USA: Unites States of America

USART: Universal Synchronous & Asynchronous Receiver Transmitter

USB: Universal Serial Bus

V: Volt

VFD: Variable-frequency drive

WirelessHART : Wireless Highway Addressable Remote Transducer

XOSC: Crystal Oscillator

XTAL: Cristal Oscillator

Table des matières

Introduction générale	19
Chapitre 1 Système de télégestion SCADA	21
1.1 Introduction	21
1.2 Le Système SCADA	22
1.2.2 Les composants d'un système SCADA :	23
1.3 Les standards de communication filaires industriels.....	27
1.3.1 Protocole ModBus	27
1.3.2 Protocole PROFINET.....	32
1.4 Les standards de communication sans fil industriels.....	35
1.4.1 Protocole WirelessHART	35
1.4.2 Protocole ISA100.11a.....	36
1.5 Systèmes de communication pour L'IOT	40
1.5.1 LoRaWAN	40
1.5.2 Sigfox.....	44
1.6 Conclusion	47
Chapitre 2 Outils logiciels et matériel.....	48
2.1 Introduction	48
2.2 Carte de développement RF	49
2.2.1 Schéma électronique de la carte ccRF 3 click.....	51
2.2.2 Schéma fonctionnel du CC1120.....	54
2.2.3 Diagramme d'état du circuit	57
2.2.4 La configuration	59
2.3 Carte à microcontrôleur.....	62
2.3.1 Carte Nucleo-64	62
2.3.2 Microcontrôleur STM32L476.....	63
2.3.3 L'interface SPI	65
2.3.4 L'interface USART	74

2.3.5	Débugueur et programmeur ST-Link.....	79
2.3.6	Environnement STM32CubeIDE	80
2.4	Carte d'adaptation	81
2.5	Logiciel Arduino IDE	82
2.6	Analyseur logique.....	82
2.7	Carte STEVAL-FKI433V1 et le logiciel S2-LP-DK.....	87
2.8	Logiciel KiCad.....	88
2.9	Conclusion	89
Chapitre 3	Conception et réalisation	90
3.1	Introduction	90
3.2	Programmation	90
3.2.1	Configuration du circuit CC1120.....	91
3.2.2	Développement de la bibliothèque sous l'environnement CubeIDE	98
3.2.3	Choix de l'Arduino.....	111
3.2.4	Implémentation du code de transmission.....	113
3.2.5	Implémentation du code de réception.....	118
3.2.6	Synthèse.....	121
3.3	Conception du module.....	122
3.3.1	Module RF en version CC1120 – STM32F103C8T6.....	122
3.3.2	Implémentation	122
3.3.3	Conception du schéma électronique.....	123
3.3.4	Schéma électronique	124
3.3.5	Les différentes parties du schéma électronique	124
3.4	Conception du PCB.....	128
3.4.1	Les couches du PCB.....	128
3.4.2	Visualisation 3D	131
3.5	Prototypage et tests :.....	133
3.5.1	Assemblage du module.....	133
3.5.2	Chargement du code dans le module.....	135
	Conclusion générale.....	137
	Annexes A :	138
	Annexes B :	146

Liste des figures

Figure 1-1 Système SCADA.....	Erreur ! Signet non défini.
Figure 1-2 Salle de contrôle SCADA	23
Figure 1-3 Composants du SCADA	24
Figure 1-4 Différence entre PLC et RTU	26
Figure 1-5 Trame ModBus RTU	28
Figure 1-6 Trame ModBus ASCII	29
Figure 1-7 Communication ModBus TCP/IP.....	30
Figure 1-8 Trame ModBus TCP/IP	30
Figure 1-9 PROFINET	32
Figure 1-10 Couches de protocole PROFINET	33
Figure 1-11 Réseau WirelessHART	35
Figure 1-12 Couches du modèle OSI	37
Figure 1-13 Couches du protocole ISA100.11a	39
Figure 1-14 Consommation d'un terminal Sigfox	45
Figure 2-1 Carte ccRF 3 click.....	49
Figure 2-2 Brochage de la carte ccRF 3 click	50
Figure 2-3 Schéma de la carte ccRF 3 click.....	51
Figure 2-4 Configuration des broches du circuit cc1120	52
Figure 2-5 Circuit BALF-112X-02D3	53
Figure 2-6 Schéma fonctionnel du système	54
Figure 2-7 Ensemble des composant de la partie RF du cc1120	55
Figure 2-8 Diagramme d'état simplifié du cc1120.....	58
Figure 2-9 Opérations d'écriture et de lecture des registres de configuration	60
Figure 2-10 Carte de la mémoire du circuit cc1120.....	61
Figure 2-11 Carte Nucleo-64	62
Figure 2-12 Connexion SPI entre un périphérique maitre et un périphérique esclave	67
.....	
Figure 2-13 Connexion SPI mode full-duplex.....	68

Figure 2-14 Connexion SPI mode simplex (maitre-récepteur)	68
Figure 2-15 Connexion SPI mode simplex (maitre-émetteur)	69
Figure 2-16 Connexion SPI mode Half-duplex	69
Figure 2-17 Topologie SPI Multi-slave Etoile	70
Figure 2-18 Topologie SPI Multi-slave Circulaire	71
Figure 2-19 Topologie SPI Multi-master	72
Figure 2-20 Modes du SPI	73
Figure 2-21 Trame du protocole SPI	73
Figure 2-22 Communications supportées par l'interface USART	74
Figure 2-23 Mode Synchrone et Asynchrone de l'USART	75
Figure 2-24 Trame du protocole USART.....	76
Figure 2-25 Configuration Full-duplex et Half-duplex.....	76
Figure 2-26 Connexion RS-232 sous USART	78
Figure 2-27 Débogueur/programmeur ST-Link	79
Figure 2-28 Bouclier Arduino Uno - click	81
Figure 2-29 Assemblage des trois cartes de test	81
Figure 2-30 Digilent Analog Discovery™	83
Figure 2-31 Schéma fonctionnel du Digilent Analog Discovery™	84
Figure 2-32 Capture de l'interface WaveForms.....	85
Figure 2-33 Carte STEVAL-FKI433V1 avec la NUCLEO-L152RE.....	87
Figure 3-1 Interface graphique du logiciel SmartRF Studio7	91
Figure 3-2 Première étape dans la configuration du CC1120	92
Figure 3-3 Panneau de contrôle principal.....	93
Figure 3-4 réglages typiques	94
Figure 3-5 boîte de dialogue paramètres.....	94
Figure 3-6 Paramètres choisis	95
Figure 3-7 Profile Packet RX	95
Figure 3-8 Profile Packet TX	96
Figure 3-9 Volet des registres exportés	96
Figure 3-10 Bouton export des registres	97
Figure 3-11 Environnement de développement Intégré CubeIDE	98
Figure 3-12 Création d'un nouveau projet.....	99

Figure 3-13 Choix du microcontrôleur	99
Figure 3-14 CubeMX.....	100
Figure 3-15 Configuration des lignes du debugger	101
Figure 3-16 Configuration du Crystal	101
Figure 3-17 Cadencement du bus APB.....	102
Figure 3-18 Configuration du SPI1	103
Figure 3-19 Adaptation du Pré Scalaire	103
Figure 3-20 Implémentation UART1	104
Figure 3-21 Configuration des interruptions externes EXTI	105
Figure 3-22 Configuration de la broche CS	106
Figure 3-23 Fonction cc112xRead	108
Figure 3-24: Fonction cc112xWrite	108
Figure 3-25: Fonction cc112xStrobe	109
Figure 3-26 Fonction cc112xExtRead	110
Figure 3-27 Fonction cc112xExtWrite	110
Figure 3-28 Organigramme du code TX	114
Figure 3-29 Paramètres RF.....	115
Figure 3-30 Test de puissance du signal	115
Figure 3-32 Paramètres du Packet CC1120.....	116
Figure 3-33 Octets du SYNC WORD.....	117
Figure 3-34 Test de transmission	117
Figure 3-35 Organigramme du code RX	119
Figure 3-36 Fonctionnement du mode réception.....	121
Figure 3-37 Disposition des Broches SPI1	123
Figure 3-38 Disposition des broches UART1	123
Figure 3-39 schéma électronique	Erreur ! Signet non défini.
Figure 3-40 Modes de BOOT du STM32.....	125
Figure 3-41 PCB comportant les deux couches.....	128
Figure 3-42 PCB couche supérieure	129
Figure 3-43 Couche inférieure	129
Figure 3-44 Couche composants.....	130
Figure 3-45 Visualisation 3D.....	131

Figure 3-46 Ordre des PCB et des composants.....	133
Figure 3-47 PCB du module final.....	133
Figure 3-48 PCB avec carte ccRF monté.....	134
Figure 3-49 PCB avec carte ccRF monté.....	134
Figure 3-50 Programmation du module.....	135
Figure A.1 Registre Octet d'état.....	138
Figure A.2 Accès SPI	139
Figure A.3 Suite de l'espace registre SPI	140
Figure A.4 Espace registre SPI	141
Figure A.5 Espace registre SPI étendu.....	143
Figure A.6 Suite Espace registre SPI étendu	144
Figure A.7 Commandes du circuit cc120.....	144
Figure A.8 Adressage du FIFO	145
Figure B.1 BOM	146

Liste des tableaux

Tableau 3.1 Etude comparative entre les deux MCU

Introduction générale

L'essor technologique en industrie a fortement contribué au passage de la technologie analogique vers la technologie numérique. Il a été donc nécessaire d'adapter les équipements et machines afin d'assurer le traitement de données numériques et de communication à l'aide de divers protocoles pour le transfert et l'acquisition de l'information.

Toutes ces nouvelles technologies de l'information et de la communication permettent d'améliorer l'efficacité, les performances, la sécurité des déplacements de données, et la réduction des coûts et de maintenance des réseaux de communication.

L'objectif du travail que nous avons mené, est la communication sans fil et la gestion de la circulation des paquets de données entre les différents éléments et compartiments d'installations dédiées aux systèmes de télégestion d'eau.

En effet, notre travail s'articule autour d'un projet PNR, intitulé "Système de communication RF numérique radio longue portée", un projet qui est en collaboration entre le CDTA et la société DOOFAS. Cette dernière a déjà développé un RTU nommé SAFIR pour les systèmes de télégestion de station de pompage d'eau.

La télégestion désigne un ensemble de solutions technologiques permettant de piloter à distance des installations autonomes géographiquement dispersées. Ces systèmes permettent entre autres, d'alerter l'exploitant en cas de problème technique ou alors enregistrer le fonctionnement des équipements surveillés.

Les systèmes de communication par radio sont des systèmes qui se prêtent le mieux aux communications à moyenne ou longue distance. Les gammes d'ondes les plus employées jusqu'à aujourd'hui à travers le monde sont les ondes VHS et les ondes UHF dans la bande 420-470 MHz.

Ce mémoire est organisé comme suit :

Dans le premier chapitre, on présente brièvement le système de télégestion SCADA qui est l'un des types de systèmes de contrôle industriel, spécialement utilisé par les contrôleurs PLC et RTU. Suite à cela, les différents systèmes de communication filaires et sans fil utilisés dans ce système sont également présentés.

Le deuxième chapitre décrit les outils matériels et logiciels utilisés durant le développement du module, tels que la carte Nucleo-I476rg, la carte ccRF 3 click, l'environnement STM32Cube IDE et le logiciel KiCad.

Quant au troisième chapitre, on définit la démarche suivie pour la conception du module RF, puis on présente les réalisations du circuit, le PCB et les tests nécessaires pour l'évaluation du fonctionnement.

1.1 Introduction

La télégestion est l'ensemble des dispositifs, qui associe l'électronique, l'informatique et la télécommunication afin de permettre le suivi et le contrôle à distance des installations.[1]

Parmi ces dispositifs, on cite les capteurs, les enregistreurs, les concentrateurs, les automates, et les transpondeurs.

La digitalisation de la gestion [2] permet de centraliser les données et de créer une intelligence collective pour prévoir les conséquences d'un événement sur les autres sites.[3]

Elle est indispensable dans les secteurs d'énergie, d'eau, industriels, et également la gestion des bâtiments, comme le domaine de traitement des eaux usées, la fourniture en eau potable, la gestion de l'énergie et l'éclairage publique ...etc.

Elle permet l'accès à distance à des sites géographiquement réparties, en les reliant au même réseau à travers des éléments de télécommunications.

La mise en place d'un système de télégestion a plusieurs intérêts [3] :

- La mesure et la comparaison régulière des indicateurs, ont pour but d'avoir une traçabilité qui facilite et améliore le suivi, la maintenance et la durée de vie du matériel.
- La modification des réglages des équipements
- La réduction du nombre des visites et des déplacements, optimise les interventions et réduit ses frais.
- La visualisation et le pilotage

Le système le plus répandu dans les installations industrielles nécessitant une gestion technique et statique, est le système SCADA. Ce dernier sera bien détaillé dans la suite du ce chapitre.

1.2 Le Système SCADA

1.2.1 Définition

SCADA est l'acronyme de « Supervisory Control And Data Acquisition » (système de contrôle et d'acquisition de données), un terme qui décrit les fonctions de base d'un système SCADA. Les entreprises utilisent des systèmes SCADA pour contrôler les équipements sur tous leurs sites, mais aussi pour collecter et enregistrer des données au sujet de leurs opérations.[1]

Un système SCADA est un ensemble de logiciels et matériels (figure 1-1), des contrôleurs logiques programmables (PLC) et des unités terminales distantes (RTU), peuvent être connectés à des capteurs et à des actionneurs embarqués dans une machine de production, les données issues de ces périphériques, sont envoyées vers un poste opérateur.

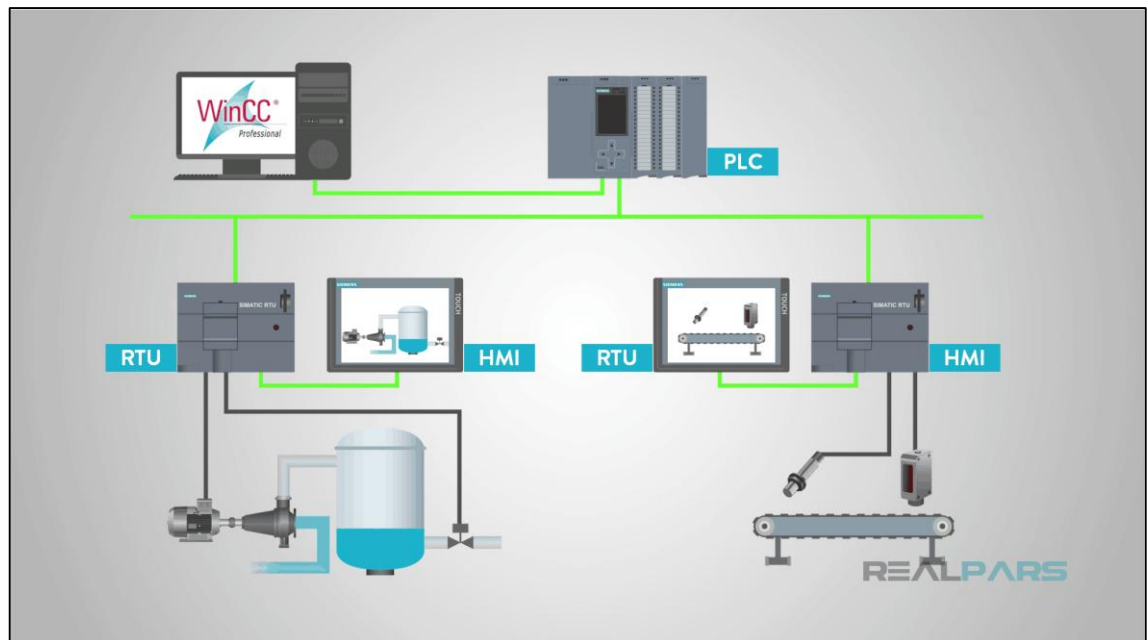


Figure 1.1 Système SCADA

Le post opérateur (figure 2-1) fait partie de la salle de contrôle, qui est connectée au réseau de l'usine par une liaison filaire, sans fil ou bien les deux, selon l'architecture de l'installation.

On trouve un matériel essentiel du système qui est une interface utilisateur connue sous le nom d'interface homme-machine (IHM), composée par des écrans et des tableaux de bords.

L'interface peut contenir des boutons, des voyants et des alertes en modélisant le système de télésurveillance sous une forme synoptique, permettant aux opérateurs de suivre en temps réel toutes les parties d'une infrastructure.

Cela permet le contrôle sur place et à distance des processus complexes dans des environnements hasardeux, sans risquer les vies du personnel et en protégeant le matériel, qui garantit le bon fonctionnement sous les normes.



Figure 1.2 Salle de contrôle SCADA

Grâce à ces systèmes, l'opérateur peut interagir directement avec les différents équipements comme des vérins, des moteurs et des pompes, qu'il soit alerté lors d'une anomalie ou un défaut, permettant ainsi de faire des ajustements et des réglages au moment adéquat de façon que l'occurrence des incidents soit réduite, afin d'aider à minimiser les frais de production et de maintenance.

1.2.2 Les composants d'un système SCADA :

Un système SCADA se compose des éléments suivants [5] :

- L'interface homme-machine est un dispositif d'entrées / sorties, qui représente les données du système, et à travers cette interface, l'opérateur supervise et contrôle les processus.
- Le système de surveillance est un lien entre le logiciel de l'interface et l'appareillage de contrôle, tels que les automates, les RTU, les capteurs, il fait l'acquisition des données des procédés, et envoie des commandes.
- Les Unités terminales distantes (RTU) : notamment appelé unités de télémétrie à distance, sont des outils de supervision, le rôle principal des RTU est de relier les appareils au poste central.

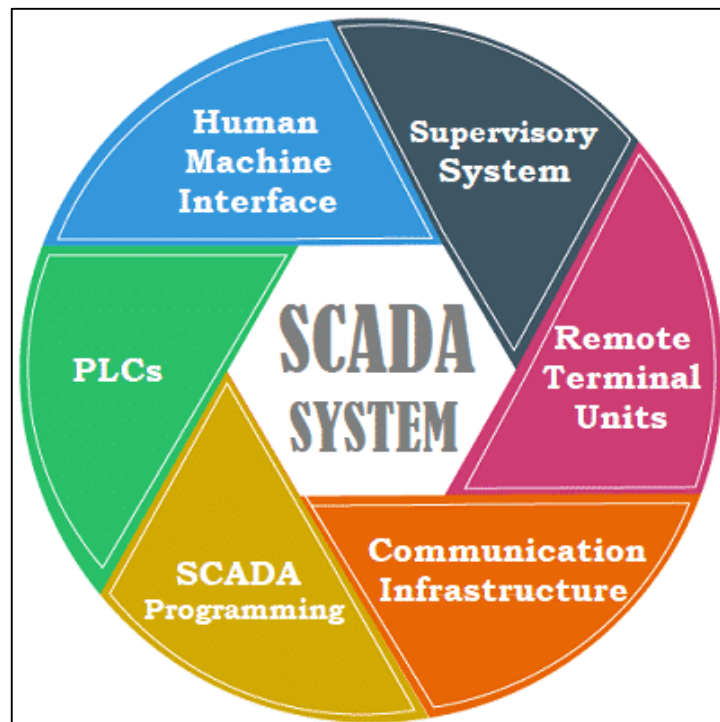


Figure 1.3 Composants du SCADA

- L'automate programmable (PLC) est un outil configurable, qui commande les pré-actionneurs et reçoit les signaux des capteurs afin de les convertir en données numériques.

- L'Infrastructure de communication est un système réseau doté de technologie de télécommunication et de radiodiffusion, basé sur des installations câblées en cuivre, des installations fibre-optiques, ou transmission sans fil fournissant le support de communication pour les différents organes du SCADA.
- La programmation SCADA comprend l'interface d'utilisateur et de la centrale, elle est utilisée pour composer les diagrammes d'affichage des différentes informations sur les processus et les événements.

1.2.3 Différences entre un PLC et un RTU

Ces deux dispositifs coexistent dans les mêmes installations, et ont des fonctionnalités similaires, c'est pour cela qu'ils sont souvent confondus l'un avec l'autre, néanmoins nous notons quelques différences majeures entre eux [6].

L'unité de télémétrie est plus tolérante à l'environnement difficile dans des sites isolés avec des températures extrêmes, et ces RTU sont fréquemment utilisés avec les ondes radios, les micro-ondes et la communication satellitaire.

Le RTU a un système d'alimentation de secours, parfois une batterie est mise en place avec un circuit de charge utilisant l'énergie solaire, lui permettant la continuité de travail lors d'une coupure du réseau électrique.

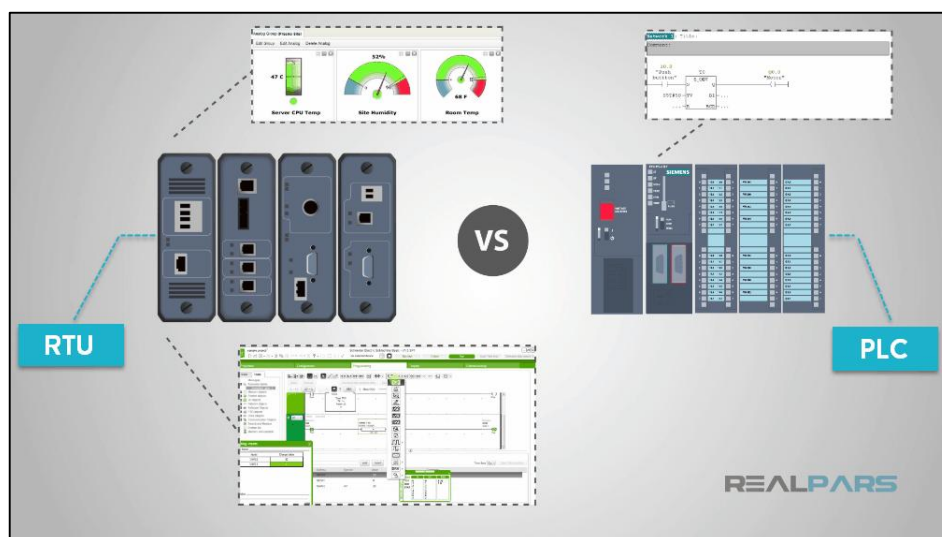


Figure 1.4 Différence entre PLC et RTU

L'automate nécessite un logiciel particulier pour la programmation ainsi que des connaissances dans les langages de programmation, comme le Ladder (LD), le langage texte structuré (ST) etc. tandis que le RTU peut être programmé par une interface web simple à utiliser ou bien un logiciel de configuration des entrées, des sorties et de communication.

Et Si on compare leur prix, le PLC est moins cher par rapport au RTU.

Avec l'évolution de la technologie des nouveaux Automates et instruments combinant les diverses propriétés des RTUs comme le PAC (Programmable Automation Controller), rend les avantages de chacun disponibles dans les autres, et la comparaison devient difficile entre eux.

Le seul avantage qui reste valable est la tolérance aux environnements sévères et stricts, qui met le RTU le dispositif préféré pour les installations éloignées.

1.3 Les standards de communication filaires industriels

1.3.1 Protocole ModBus

Modbus est un protocole de communication publié pour la première fois par Modicon en 1979, incorporé pour les systèmes d'automatisation industrielle et à utiliser avec ses automates programmables (PLC) Modicon.[7]

Plusieurs versions du protocole Modbus existent pour le port série et Ethernet et les plus courantes sont [8] :

- Modbus RTU
- Modbus ASCII
- Modbus TCP

Modbus est un standard ouvert qui décrit la boîte de dialogue de communication de messagerie.[8]

Modbus communique sur plusieurs types de supports physiques [8] :

- RTU et ASCII utilise un lien série :
 - RS-232, RS-485 et RS-422 comme couche physique
 - Configuration Maître et esclave
 - La couche application Modbus est l'équivalent de la couche Application du Modèle OSI
- TCP/IP utilise un réseau Ethernet, le maître est remplacé par un serveur, et l'esclave par un client, les deux sont identifiés par une adresse IP unique.

a Protocole Modbus® RTU

Le mode de transmission RTU, envoie les données codés en binaire, la structure du message est représentée par la figure

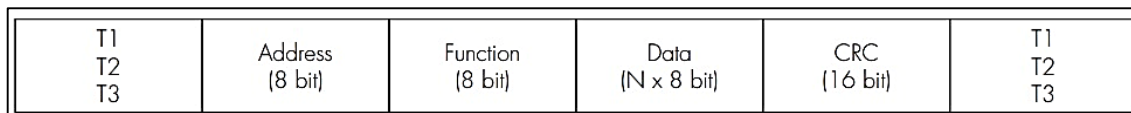


Figure 1.5 Trame ModBus RTU

- La séquence T1 T2 T3 correspond à la durée pendant laquelle aucune donnée ne doit être échangée sur le bus de communication, pour permettre aux appareils connectés de reconnaître la fin d'un message et le début du suivant. Cette durée doit être égale à 3,5 caractères.[9]
- Le champ Adresse contient l'adresse de l'instrument esclave auquel le message est envoyé. Il est théoriquement possible de connecter 245 appareils (esclaves).[9]
- Le champ Fonction contient le code de la fonction que l'esclave doit exécuter.[9]
- Le champ Données contient les données transmises à l'esclave ou celles envoyées par l'esclave comme réponse à une interrogation. [9]
- Le champ CRC permet au Maître et à l'esclave de s'assurer de l'absence d'erreurs de transmission. Ceci permet, en présence d'interférences sur la ligne de transmission, d'ignorer le message transmis pour éviter les problèmes côté Maître et côté esclave.[9]

b Protocole Modbus® ASCII

Ce protocole est utilisé sur les applications qui nécessitent une communication par modem. [9]

Les fonctions et les adresses disponibles sont identiques à celles de la version RTU sauf les caractères transmis sont en ASCII et que la fin du message n'est pas établie sur la base d'une durée mais en fonction des caractères de retour à la ligne.[9]

La structure du message sur le port de communication est la suivante :

:	Adresse 2 caractères	Fonction 2 caractères	Données (N caractères)	LRC 2 caractères	CR LF
---	-------------------------	--------------------------	---------------------------	---------------------	-------

Figure 1.6 Trame ModBus ASCII

- Le début du message est indiqué par le caractère ':' (ASCII 3Ah).
- Le champ Adresse contient l'adresse de l'appareil esclave auquel le message est envoyé.
- Le champ Fonction contient le code de la fonction que l'esclave doit exécuter.
- Le champ Données contient les données transmises à l'esclave ou celles envoyées par l'esclave comme réponse à une interrogation. La longueur maximum admise est de 32 registres consécutifs.
- Le champ CRC permet au Maître et à l'esclave de s'assurer de l'absence d'erreurs de transmission.
- Ceci permet, en présence d'interférences sur la ligne de transmission, d'ignorer le message transmis pour éviter les problèmes côté Maître et côté esclave.
- Le message se termine toujours par les caractères de contrôle CRLF (0D 0A).

c Protocole Modbus® TCP/IP

Modbus TCP/IP (également Modbus-TCP) est simplement le protocole Modbus RTU avec une interface TCP qui fonctionne sur Ethernet.[10]

TCP/IP permet d'échanger des blocs de données binaires entre ordinateurs. C'est aussi une norme mondiale qui sert de base au World Wide Web. La fonction principale de TCP est de s'assurer que tous les paquets de données sont reçus correctement, tandis que IP s'assure que les messages sont correctement adressés et routés. [10]

La combinaison TCP/IP est simplement un protocole de transport, et ne définit pas ce que les données signifient ou comment les données doivent être interprétées. [10]

Modbus TCP/IP utilise TCP/IP et Ethernet pour transporter les données de la structure de message Modbus entre les périphériques compatibles. Autrement dit, Modbus TCP/IP combine un réseau physique (Ethernet), avec un standard réseau (TCP/IP), et une méthode standard de représentation des données (Modbus comme protocole d'application). Essentiellement, le message Modbus TCP/IP est simplement une communication Modbus encapsulée dans une enveloppe Ethernet TCP/IP.

Le protocole ModBus TCP fonctionne en mode client/serveur, figure 1.7.

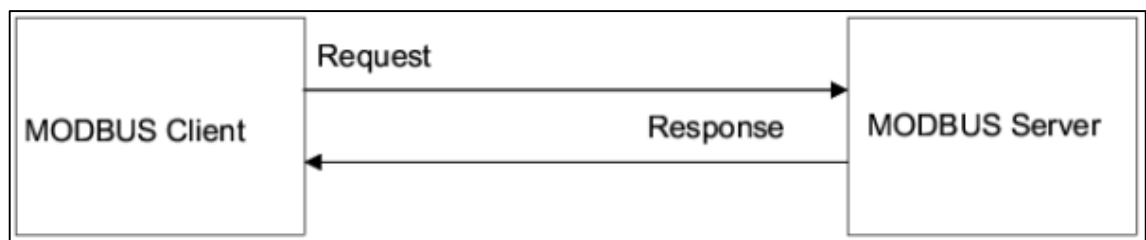


Figure 1.7 Communication ModBus TCP/IP

Le format de la trame du ModBus est représenté par Figure 1-8 :

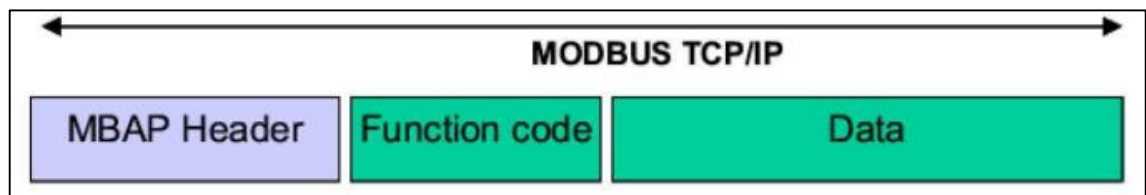


Figure 1.8 Trame ModBus TCP/IP

Le premier champ "numéro de l'esclave" en ModBus RTU est remplacé par un champ "MBAP Header" et le champ "Error check (CRC)" disparaît car les couches TCP/IP effectuent déjà un contrôle d'erreur (CRC). [11]

L'en-tête MBAP (ModBus Application Protocol header) d'une longueur de 7 octets se décompose comme suit [11] :

- Transaction Identifier (2 octets) : il s'agit en fait du numéro d'un message MODBUS. Chaque message (requête + réponse) est identifié par un numéro différent.
- Protocol Identifier (2 octets) : il s'agit du numéro du protocole pour les systèmes multiplexés qui en utilisent plusieurs (MODBUS = 0).
- Length (2 octets) : il s'agit de la longueur du message à suivre (donc Unit Identifier + Data).
- Unit identifier (1 octet) : numéro de l'esclave.

1.3.2 Protocole PROFINET

Le protocole PROFINET est un standard de communication industriel Ethernet ouvert basé sur les normes CEI 61158 et CEI 61784, développé pour l'échange des données entre les contrôleurs et les dispositifs dans un réseau d'automatisation.[1]



Figure 1.9 PROFINET

L'organisation PROFIBUS & PROFINET International (PI) a introduit PROFINET au début des années 2000, et il est devenu maintenant le premier choix dans toutes les applications qui concerne l'automatisation des processus, l'automatisation de la production et dans plusieurs autres domaines.[12]

Caractéristiques du PROFINET [13] :

- PROFINET comme le ModBus utilise le concept maître-esclave, mais il est capable de gérer un réseau à plusieurs maîtres.
- PROFINET définit la communication cyclique et acyclique entre les composants, y compris les diagnostics, la sécurité fonctionnelle, les alarmes et les informations supplémentaires.
- Pour relier tous ces composants, PROFINET utilise Ethernet standard non modifié comme support de communication. Des câbles Ethernet industriels blindés, connectent les composants PROFINET au sein d'un réseau, permettant à d'autres protocoles basés sur Ethernet de coexister au sein de la même infrastructure.
- En plus de PROFINET, on peut utiliser d'autres protocoles basés sur Ethernet pour compléter le réseau, tel que OPC, UA, SNMP, MQTT ou HTTP.

Les environnements d'automatisation industrielle nécessitent souvent des communications à haut débit et déterministes, qui consistent à diffuser des messages exactement au moment où ils sont attendus.[13]

Typiquement il établit une communication bidirectionnelle avec une vitesse de 100Mbit/s, les câbles peuvent atteindre une longueur de 1000 mètres, mais pour des débits plus élevés des câbles plus courts sont nécessaires.[13]

PROFINET est un protocole de communication qui se situe à la septième couche du modèle OSI. Le modèle de cette couche décrit de manière générique les couches d'abstraction d'un système de communication.

La Figure 1-10 présente l'analogie entre le modèle OSI et le modèle de communication du PROFINET.

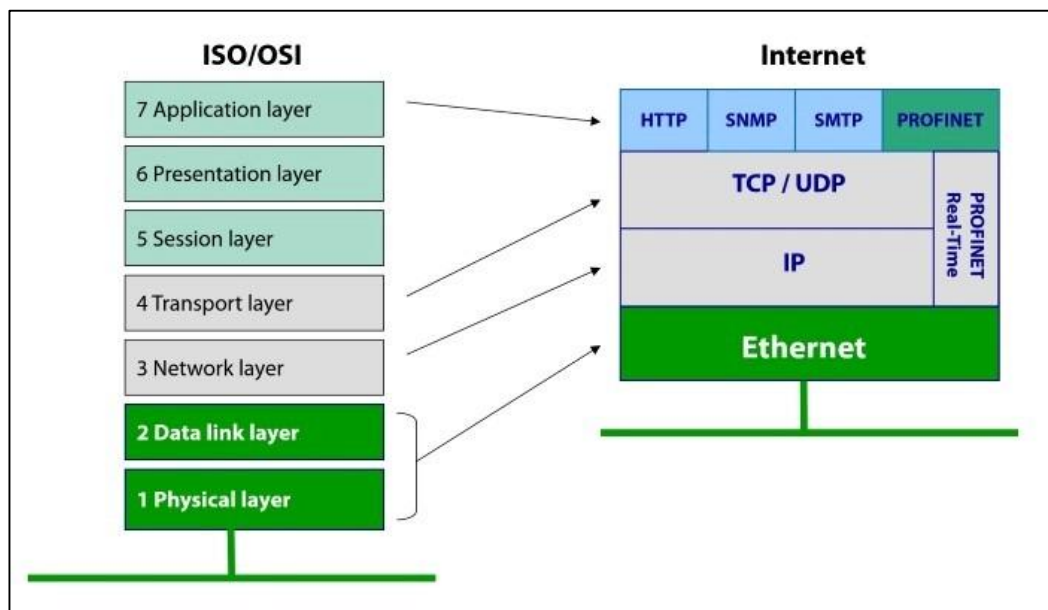


Figure 1.10 Couches de protocole PROFINET

PROFINET doit s'assurer que les messages sont livrés avec la vitesse et le déterminisme approprié en fonction de la tâche.

Les applications ne nécessitent pas les mêmes performances. Par exemple, le chargement des données de configuration d'un appareil de processus peut prendre plusieurs minutes sans affecter la production.

D'autre part, un retard de communication de quelques millisecondes entre un automate et un variateur électronique de vitesse (VFD) à grande vitesse peut avoir un impact significatif sur le processus.

Un profil PROFINET peut utiliser la communication basée sur TCP/IP ou UDP/IP pour les tâches non urgentes, telles que la configuration et le paramétrage.

Toutefois, en raison de la latence et des retards potentiels associés à la communication basée sur IP, cette méthode n'est pas adaptée aux tâches urgentes.

PROFINET utilise un canal en temps réel pour fournir des données de manière rapide et déterministe pour les applications urgentes.

Pour l'utilisation du mécanisme temps réel de PROFINET, aucun matériel ou configuration spéciale n'est requise, car tous les produits sont équipés de cette fonctionnalité.

PROFINET est idéal pour les applications industrielles grâce à sa flexibilité, sa grande vitesse de transmission et la fiabilité du transport des données.

1.4 Les standards de communication sans fil industriels

1.4.1 Protocole WirelessHART

WirelessHART est un sous-ensemble de la norme de communication d'instruments industriels HART de la version 7, qui communique les données de processus sur les ondes radio de 2,4 GHz. Les instruments individuels communiquent avec un dispositif commun de « passerelle » servant d'interface entre le réseau sans fil et un réseau câblé ou un système de contrôle hôte.[14]

La topologie de ce réseau est la suivante, tous les nœuds communiquent avec un concentrateur (gateway) qui est une interface entre le réseau filaire et le réseau sans fil (Figure 1-11) [14].

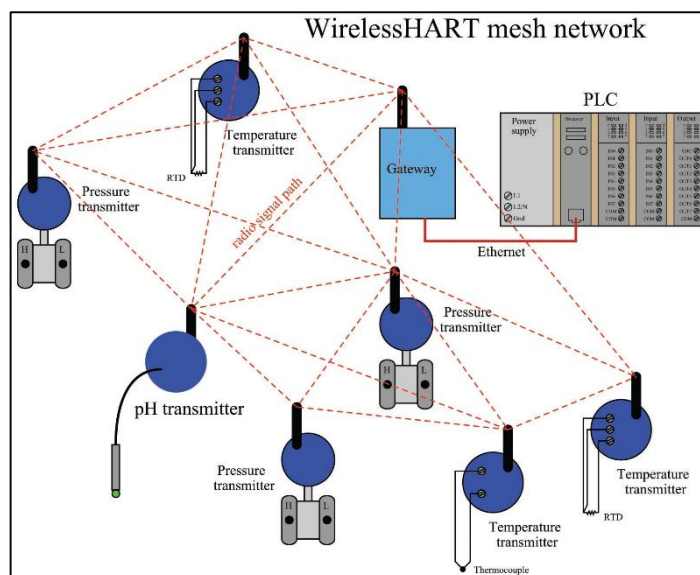


Figure 1.11 Réseau WirelessHART

Tous les appareils du WirelessHART sont une source de signal et un répéteur en même temps, ce qui forme un réseau maillé non étoilé. Il permet d'avoir des itinéraires alternatifs, et fournit une communication redondante et une longue portée.[14]

Lorsqu'un nœud essaye d'envoyer un message vers le concentrateur mais le chemin est bloqué ou l'appareil se situe loin, le message passe par un ou plusieurs nœud(s) intermédiaire(s) afin d'arriver au concentrateur. [14]

Ceci est utile quand un échec au niveau d'un appareil ou une interruption des ondes radio se produit, cela signifie aussi que la taille du réseau peut être plus grande que la portée d'un seul nœud. [14]

Pour garantir une liaison fiable et moins susceptible aux interférences, le spectre étalé à séquence directe (DSSS) est appliqué avec un saut de fréquences entre les 15 chaînes définies dans IEEE802.15.4, et aussi une puissance de transmission de 10 dBm est configurée par défaut.

Chaque réseau a son identifiant, alors on peut avoir plusieurs réseaux dans la même zone géographique sans le problème du bruit.

1.4.2 Protocole ISA100.11a

Le standard sans fil ISA100 a été développé par la Société Internationale d'Automatisation (ISA), pour simplifier toutes les spécifications, procédures et processus nécessaires à utiliser dans la mise à l'essai et la certification des produits sans fil, ainsi que pour réduire le temps, les coûts et les risques liés au développement et au déploiement d'appareils et de systèmes sans fil industriels pour les utilisateurs et les fabricants.[15]

ISA100 prend en charge le concept de tunnel (Tunneling) qui permet le mouvement sécurisé des données d'un réseau à l'autre, ce principe consiste à permettre l'envoi de communications sur un réseau privé à travers un réseau public, comme Internet par le biais d'un processus appelé encapsulation.[16]

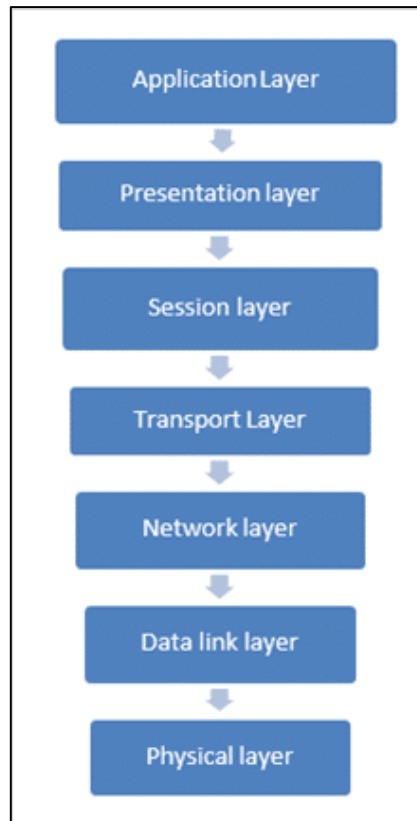


Figure 1.12 Couches du modèle OSI

Les communications sont protégées à l'aide d'une application accélérée du chiffrement par blocs AES (Advanced Encryption Standard) 128 bits standard du NIST. Seuls les appareils qui ont des clés de jointure secrètes légitimes et uniques sont admis au réseau. Les clés AES-128 à plusieurs niveaux sont dérivées des clés de jointure et mises à jour périodiquement.[16]

Le protocole prend la même architecture que les 7 couches du modèle OSI, figure1-12 , mais il est constitué par les cinq couches suivantes (présentées par la figure 1-13, [15] [17] [18] :

- **Couche physique**

Cette couche est basée sur la couche 2.4GHz DSSS IEEE 802.15.4-2006.

- **Couche de données**

La couche donne un support pour la création, la maintenance, les fonctions et les transmissions des paquets requis pour les capteurs sans fil. Elle inclut en plus des fonctions de la deuxième couche du modèle OSI, parmi les autres fonctionnalités, on cite :

- Adressage lien-local
- Gestion de PHY
- Saut de canaux adaptative
- Vérifications d'adressage minutage et intégrité des messages
- Détection et récupération des messages perdus
- Synchronisation d'horloge

Les messages sont communiqués dans des intervalles de temps qui sont généralement des intervalles synchronisés de 10 millisecondes, mais ils sont configurables.

La synchronisation de temps offre un horodatage précis et le saut de canal adaptatif augmente la fiabilité en évitant les canaux occupés.

De plus, les créneaux horaires synchronisés et le saut de canal réduisent l'utilisation d'un seul canal, améliorant la coexistence avec d'autre réseau RF dans le même spectre.

- **Couche réseau**

La couche réseau dans ISA100.11a utilise 6LoWPAN [15]. L'adressage IPv6 est utilisé pour le routage de bout en bout. Dans un réseau ISA100.11a, il est possible pour les paires serveur/client de générer des paquets IPv6 qui sont ensuite acheminés via des routeurs périphériques 6LoWPAN vers des périphériques ISA100.11a compatibles 6LoWPAN.

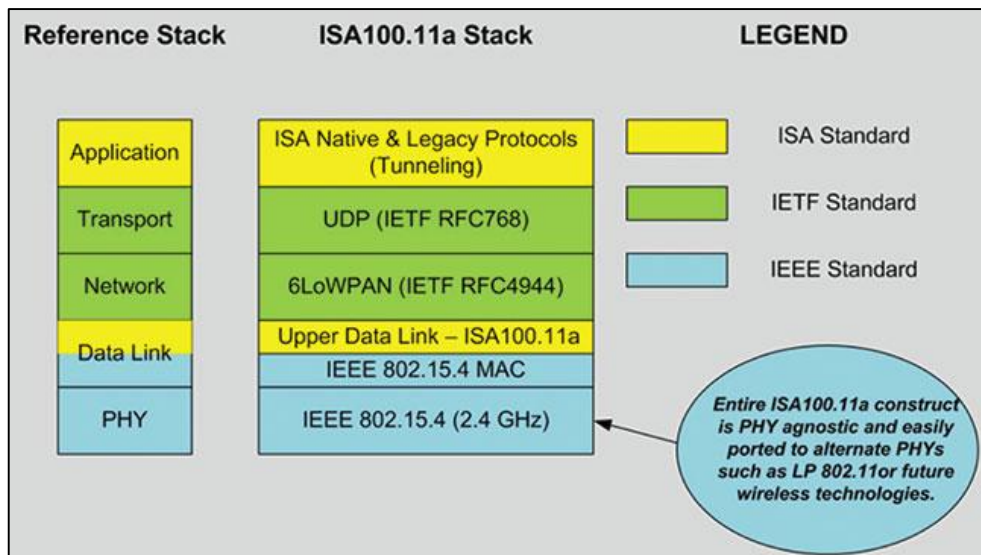


Figure 1.13 Couches du protocole ISA100.11a

- **Couche de transport**

La couche de transport prend en charge un service sans connexion basé sur le protocole UDP avec une vérification améliorée de l'intégrité des messages et une sécurité de bout en bout.

- **Couche d'application**

L'ISA100.11a spécifie uniquement un ensemble de services pour les applications utilisateurs et non une application d'automatisation de processus. Seule l'application gestion du système est spécifiée.

1.5 Systèmes de communication pour L'IOT

L'arrivée de l'Internet des objets fait bouger les lignes. En effet, l'IoT industriel dispose d'avantages notables comme une très bonne capacité de communication sûr de grandes distances, des coûts d'équipements peu élevés et de faibles tarifs d'abonnements mensuels. Elle peut s'avérer être une technique complémentaire, utile aux systèmes traditionnels de télégestion.[3]

On mentionne deux systèmes de communication radio fréquence les plus utilisés dans le monde de l'internet des objets, LoRaWAN et Sigfox.

1.5.1 LoRaWAN

La spécification LoRaWAN® est un protocole de réseau à faible puissance et à grande portée (LPWA) conçu pour connecter sans fil des « objets » alimentés par batterie à Internet dans des réseaux régionaux, nationaux ou mondiaux, et cible les principales exigences en matière d'Internet des objets (IdO), comme les services de communication bidirectionnelle, de sécurité de bout en bout, de mobilité et de localisation.[19]

a Topologie

L'architecture de réseau LoRaWAN® est déployée dans une topologie en étoile dans laquelle les passerelles transmettent les messages entre les dispositifs terminaux et un serveur de réseau central. Les passerelles sont connectées au serveur réseau via des connexions IP standard et agissent comme un pont transparent, convertissant simplement les paquets RF en paquets IP et vice versa. [19]

b Classes

LoRaWAN a trois classes différentes de terminaux pour répondre aux différents besoins reflétés dans le large éventail d'applications :

- **Classe A – Dispositifs bidirectionnels de faible puissance**

La classe par défaut qui doit être supportée par tous les terminaux LoRaWAN, la communication de classe A est toujours initiée par le terminal et est totalement asynchrone. Chaque transmission de liaison montante peut être envoyée à tout moment et est suivie de deux courtes fenêtres de liaison descendante, ce qui permet une communication bidirectionnelle ou des commandes de contrôle réseau si nécessaire. C'est un protocole de type ALOHA. [19]

L'appareil final est capable de passer en mode veille de faible puissance aussi longtemps que défini par sa propre application : il n'y a pas d'exigence réseau pour les réveils périodiques. Cela fait de la classe A le mode de fonctionnement à la puissance la plus faible, tout en permettant la communication ascendante à tout moment. [19]

Étant donné que la communication en liaison descendante doit toujours suivre une transmission en liaison montante avec un calendrier défini par l'application de l'appareil final, la communication en liaison descendante doit être mise en tampon au serveur réseau jusqu'au prochain événement de liaison montante. [19]

- **Classe B – Dispositifs bidirectionnels avec latence déterministe en liaison descendante**

En plus des fenêtres de réception initiées de classe A, les appareils de classe B sont synchronisés avec le réseau à l'aide de balises périodiques et ouvrent des « emplacements de ping » en liaison descendante aux heures prévues. [19]

Cela permet au réseau d'envoyer des communications en liaison descendante avec une latence déterministe, mais au détriment d'une certaine consommation d'énergie supplémentaire dans l'appareil final. [19]

La latence est programmable jusqu'à 128 secondes pour différentes applications, et la consommation d'énergie supplémentaire est suffisamment faible pour être encore valable pour les applications alimentées par batterie. [19]

- **Classe C – Dispositifs terminaux bidirectionnels à faible latence**

En plus de la structure de liaison montante de classe A suivie de deux fenêtres de liaison descendante, la classe C réduit davantage la latence sur la liaison descendante en maintenant le récepteur de l'appareil final ouvert à tout moment que l'appareil ne transmet pas (semi-duplex). [19]

Sur cette base, le serveur réseau peut lancer une transmission en liaison descendante à tout moment en supposant que le récepteur du périphérique final est ouvert, donc pas de latence. Le compromis est la perte de puissance du récepteur (jusqu'à ~50 mW) et ainsi la classe C est appropriée pour des applications où la puissance continue est disponible. [19]

Pour les appareils alimentés par batterie, il est possible de basculer temporairement entre les classes A et C, et est utile pour les tâches intermittentes telles que les mises à jour du firmware en direct. [19]

c Débit de données

Outre le saut de fréquence, tous les paquets de communication entre les terminaux et les passerelles incluent également un paramètre variable de « débit de données » (DR). La sélection du DR permet un compromis dynamique entre la plage de communication et la durée du message. [19]

De plus, en raison de la technologie de diffusion du spectre, les communications avec les différents RD n'interfèrent pas les unes avec les autres et créent un ensemble de canaux virtuels de « code » augmentant la capacité de la passerelle. [19]

Afin de maximiser la durée de vie de la batterie des dispositifs d'extrémité et la capacité globale du réseau, le serveur réseau LoRaWAN® gère le réglage DR et la puissance de sortie RF pour chaque dispositif d'extrémité individuellement au moyen d'un schéma de débit de données adaptatif (ADR). [19]

Les débits en bauds LoRaWAN® varient de 0,3 kbps à 50 kbps. [19]

d Sécurité

Les algorithmes AES sont utilisés pour fournir l'authentification et l'intégrité des paquets au serveur réseau et le chiffrement de bout en bout au serveur d'applications.

En fournissant ces deux niveaux, il devient possible de mettre en œuvre des réseaux partagés « multi-locataires » sans que l'opérateur de réseau ait la visibilité des données utiles des utilisateurs. [19]

La spécification LoRaWAN® est élaborée et maintenue par la LoRa Alliance®, une association ouverte de membres collaborateurs.[19]

1.5.2 Sigfox

Sigfox est une communication cellulaire sans fil à longue portée qui offre des solutions personnalisées principalement pour les applications à faible débit de l'Internet des objets (IdO) et M2M en utilisant ses services de connectivité IdO de bout en bout en utilisant ses technologies brevetées. [20]

Le protocole réseau Sigfox dispose de stations de base brevetées qui sont intégrées avec des radios définies par logiciel. Les dispositifs finaux utilisent la modulation binaire de modulation par décalage de phase (BPSK) pour se connecter aux stations de base.[20]

Le réseau Sigfox a été conçu pour faciliter une communication efficace sur une faible consommation d'énergie. Une faible consommation d'énergie garantit que les appareils à distance fonctionnent longtemps avec un minimum de charge ou d'entretien de la batterie. Sigfox permet la communication IoT sur de longues distances permettant d'émettre avec des stations de base minimales. [20]

a **Caractéristiques de Sigfox :**

- **Capacité de réseau élevée**

Sigfox a une capacité élevée qui lui permet d'augmenter jusqu'à des milliards d'objets, c'est à cause de la modulation de bande ultra-étroite qui tient compte de la résistance et de la résilience aux interférences. [20]

- **Rendement énergétique élevé**

Le réseau Sigfox se vante d'une efficacité énergétique élevée, grâce aux semi-conducteurs à faible consommation d'énergie qui sont utilisés lors de la fabrication du matériel Sigfox. Ces semi-conducteurs permettent aux modules Sigfox de transmettre des données avec seulement environ 10 mA à 50 mA de courant.[20]

L'architecture du réseau Sigfox n'exécute pas la synchronisation des messages entre les appareils et la station de base avant de procéder à la transmission des données, ce qui signifie qu'il n'y a pas de tri requis. [20]

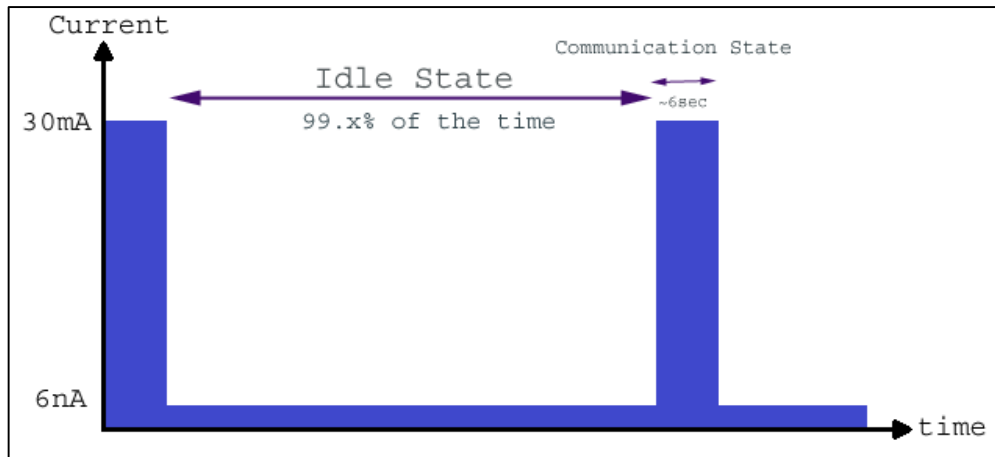


Figure 1.14 Consommation d'un terminal Sigfox

Sigfox ne consomme que quelques nano ampères lorsqu'il est inactif, ce taux de consommation d'énergie est négligeable améliorant encore l'efficacité. [20]

- **Longue portée**

Le déploiement de Sigfox couvre une vaste zone géographique avec un nombre minimal de stations de base. La plage de radiofréquences (RF) est estimée par le débit de données signifiant pour chaque sortie, la longue portée est atteinte avec un débit de données faible.[20]

Dans les régions rurales, un réseau Sigfox couvre une portée moyenne de 30 km à 50 km en raison d'une interférence minimale. Cependant, dans les centres-villes où il y a beaucoup d'obstacles, l'aire de répartition de Sigfox se réduit à 10km. [20]

- **Résilience aux interférences**

Sigfox utilise 192KHz Ultra Narrow Band (UNB) intégré à la diversité spatiale de la station de base de +20 dB qui facilitent ses caractéristiques antibrouillages uniques. Le réseau Sigfox utilise la bande ultra étroite pour fonctionner dans la bande industrielle, scientifique et médicale (ISM), car les signaux à spectre étalé sont brouillés par les signaux UNB. [20]

b L'architecture du réseau Sigfox

L'architecture de réseau de Sigfox est horizontale et mince, et a 2 couches. [20]

La première couche comprend l'équipement de réseau, principalement les stations de base et d'autres éléments. Le but de cette couche est de recevoir des messages du dispositif IoT et de les transmettre aux systèmes de support Sigfox. Ce réseau a une topologie en étoile à un saut permettant aux dispositifs IoT de se connecter à n'importe quelle station de base à proximité. [20]

La deuxième couche est le Sigfox Support System où les messages des stations de base sont traités et envoyés par appels au système client. Cette couche fournit également le point d'entrée des différents acteurs tels que Sigfox, les clients finaux, les opérateurs Sigfox et les canaux à l'écosystème en utilisant des interfaces web ou des API. [20]

La couche deux dispose d'un référentiel et d'autres outils pour analyser les données collectées par le réseau. La couche contient également des modules et des fonctionnalités qui prennent en charge le déploiement, le fonctionnement et la surveillance du réseau. [20]

c Sécurité et confidentialité de Sigfox

Comme l'objectif principal de Sigfox est d'être utilisé avec l'IoT, le réseau a été conçu pour améliorer la sécurité en mettant en œuvre des pare-feux pour sécuriser les données en mouvement et au repos. [20]

- **Sécurité du pare-feu :**

Sigfox dispose d'un pare-feu intégré qui limite les objets IoT à se connecter ou communiquer en utilisant le protocole Internet. Pour communiquer sur Internet, un objet envoie un message radio qui est capté par les stations d'accès disponibles. [20]

La station d'accès transmet ensuite le message au système de support Sigfox qui, à son tour, le transmet à la destination spécifiée. Le système de support Sigfox transmet également la réponse à l'objet expéditeur via les stations de base. Le pare-feu de sécurité sécurise ainsi les objets IoT contre les attaques liées à Internet. [20]

- **Sécurité des données :**

L'architecture Sigfox fournit un mécanisme de sécurité qui garantit l'authentification sécurisée des données et l'évitement des rediffusions, elle fournit également un mécanisme anti-espionnage supplémentaire. La chaîne IoT de Sigfox implique des clés d'authentification stockées qui sont stockées par les appareils et le client, cette clé sera nécessaire pour accéder aux données stockées par le système Sigfox. [20]

En conséquence, les mécanismes de données de Sigfox garantissent la sécurité au sein de l'écosystème diversifié et des différentes réglementations locales. La clé d'authentification de chaque périphérique est unique, ce qui signifie qu'une compromission de la clé de sécurité d'authentification d'un périphérique n'affecte pas la sécurité des autres périphériques. Toutefois, la sécurité de tout appareil est laissée à la discrétion du fabricant. [20]

1.6 Conclusion

En conclusion, l'étude du système de télégestion SCADA nous a permis de connaître l'environnement de déploiement du RTU.

Le développement de notre module de communication nécessite aussi qu'on ait des informations concernant tous les composants et les outils entrant dans sa conception.

2.1 Introduction

L'objectif premier de notre projet a été celui de réaliser un module complet d'émission et de réception avec tout son environnement. Pour cela, différents axes traduits en objectifs d'action ont été mis en exécution. Nous avons tout d'abord étudié le circuit émetteur-récepteur à travers son datasheet, ensuite des tests de programmation du microcontrôleur ont été réalisés afin de tester les fonctionnalités du circuit transceiver.

Pendant ces tests, l'analyseur logique « Digilent Analog Discovery », nous a permis de visualiser les messages échangés entre le microcontrôleur et le circuit radio.

Après avoir établi le lien entre les deux circuits, nous avons commencé le développement des fonctions constituant une bibliothèque, afin de créer un programme bien ordonné et structuré. Le temps imparti au développement des fonctions était trop long. Après des recherches, on a opté pour une autre bibliothèque afin d'accélérer la conception du système.

Le circuit de communication radio a besoin d'une configuration précise dans l'intention d'obtenir la meilleure performance, le constructeur offre alors un logiciel qui génère une configuration selon l'application.

Pendant cette phase de recherche, on a utilisé également une carte STEVAL433, qui dispose d'un autre système RF, adressée aux chercheurs et développeurs, qui nous a permis de faire les tests de transmission.

À la suite de la description de toutes les étapes entreprises, on va décrire dans ce chapitre tous les outils matériels et logiciels utilisés durant le développement du module.

2.2 Carte de développement RF

La carte utilisée est la carte ccRF 3 click (figure 2-1), un module conçu par Mikroelektronika. Il porte l'émetteur-récepteur RF hautes performances CC1120 pour les systèmes à bande étroite de Texas Instruments. [21]

Ces cartes permettent d'ajouter un émetteur-récepteur radio à faible consommation d'énergie à la fréquence 433 MHz. Ce dernier fonctionne avec une alimentation de 3,3 V. [21]

Il communique avec le microcontrôleur via l'interface SPI, avec des fonctionnalités supplémentaires fournies par les broches suivantes sur la ligne mikroBUS™ : AN, RST, CS, PWM, INT. [21]

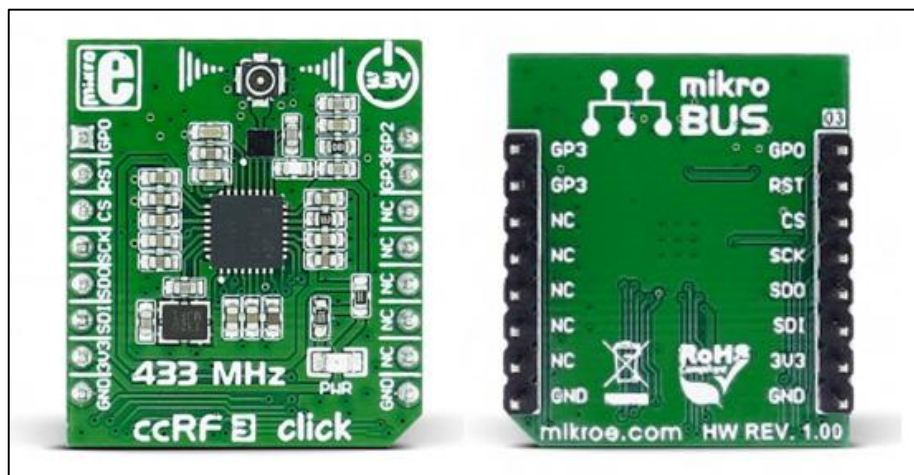


Figure 2.1 Carte ccRF 3 click

MikroBUS est une norme de socket de cartes utilisées pour interfacier des microcontrôleurs ou des microprocesseurs avec des circuits intégrés et des modules (add-on) de MikroElektronika.[22]

La norme spécifie la disposition physique du brochage de la carte, les broches de communication et d'alimentation utilisées, la taille et la forme des cartes d'extension, le positionnement des prises mikroBUS™ sur la carte mère.

La figure 2.2, illustre la disposition des broches. [22]

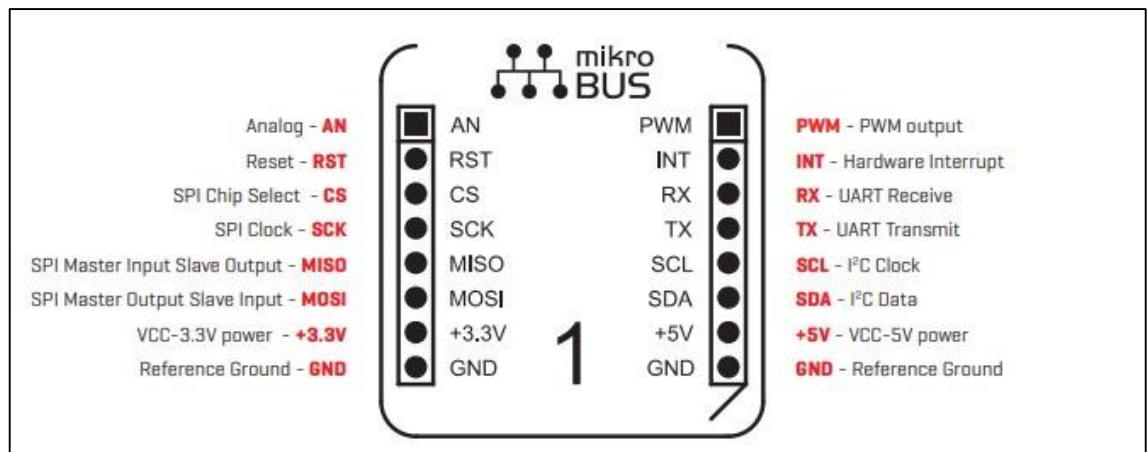


Figure 2.2 Brochage de la carte ccRF 3 click

La prise mikroBUS™ comprend une paire d'en-têtes (headers) femelles 1 × 8 avec une configuration de broche propriétaire et des marquages sérigraphiés.[22]

MikroBUS se compose de trois groupes de broches de communication (UART, I2C, et SPI), six broches supplémentaires (PWM, interruption, entrée analogique, réinitialisation et sélection de puce) et deux groupes d'alimentation (+3,3 V et 5V).[22]

2.2.1 Schéma électronique de la carte ccRF 3 click

La figure 2-3 représente le schéma électronique et les deux parties du module ccRF3 click. [2]

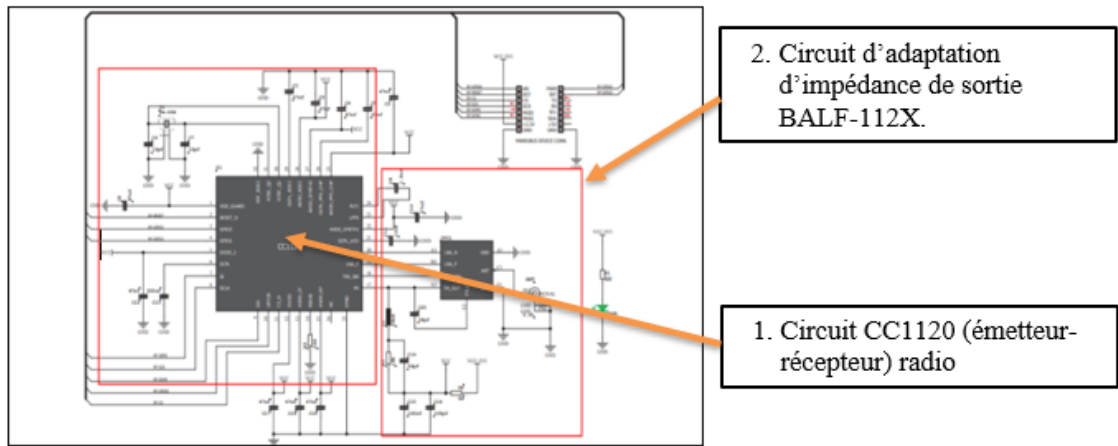


Figure 2.3 Schéma de la carte ccRF 3 click

La première partie est le circuit de la communication CC1120 :

Le CC1120, figure 2-4 est un circuit intégré d'émission et de réception radio à très haute performance, fabriqué par Texas Instruments. Il a été conçu pour fonctionner avec de très faibles puissances ce qui rend son choix idéal pour les systèmes sans fil, économique en matière d'énergies.[23]

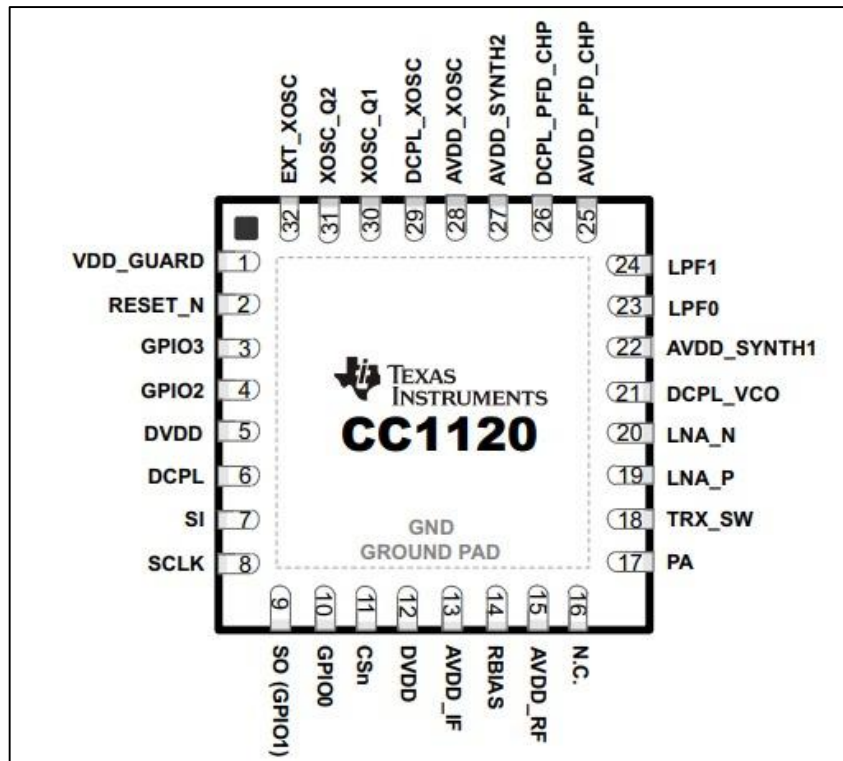


Figure 2.4 Configuration des broches du circuit cc1120

Le dispositif est principalement destiné aux applications industrielles, scientifiques et médicales (ISM) et aux bandes de fréquences des dispositifs à courte portée (SRD) de 164 à 192 MHz, 274 à 320 MHz, 410 à 480 MHz et 820 à 960 MHz. [23]

Le dispositif CC1120 offre une prise en charge matérielle étendue pour la gestion des paquets, la mise en mémoire tampon des données, les transmissions en rafale, l'évaluation du canal clair, l'indication de la qualité de la liaison et le réveil par radio (Wake-On-Radio). [23]

Les principaux paramètres de fonctionnement du dispositif CC1120 peuvent être contrôlés via une interface SPI. Dans un système typique, le dispositif CC1120 est utilisé avec un microcontrôleur et seulement quelques composants passifs externes. [23]

La deuxième partie est un circuit d'adaptation d'impédance :

Le BALF-112X-02D3 [24], est un microcircuit intégré de la famille des BALUN (Balanced/Unbalanced) spécialement conçu pour travailler avec les circuits de la famille CC112X de chez Texas Instruments.

Sa fonction principale est d'adapter l'impédance de sortie du CC1120, pour éviter les perturbations d'amplitude et le déphasage pendant la transmission et la réception.

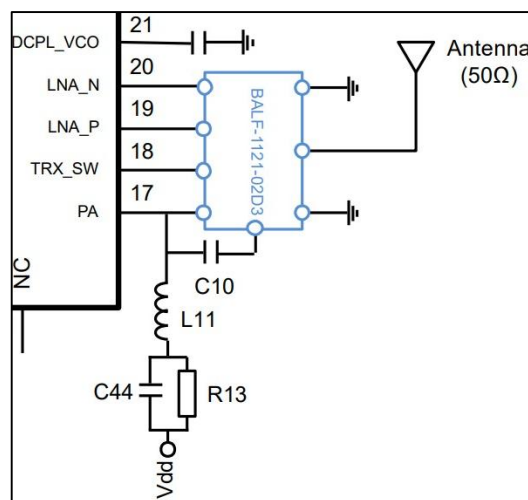


Figure 2.5 Circuit BALF-112X-02D3

2.2.2 Schéma fonctionnel du CC1120

Les blocs internes les plus importants dans le fonctionnement du CC1120, figure2-6 [25], sont présentés brièvement comme suit :

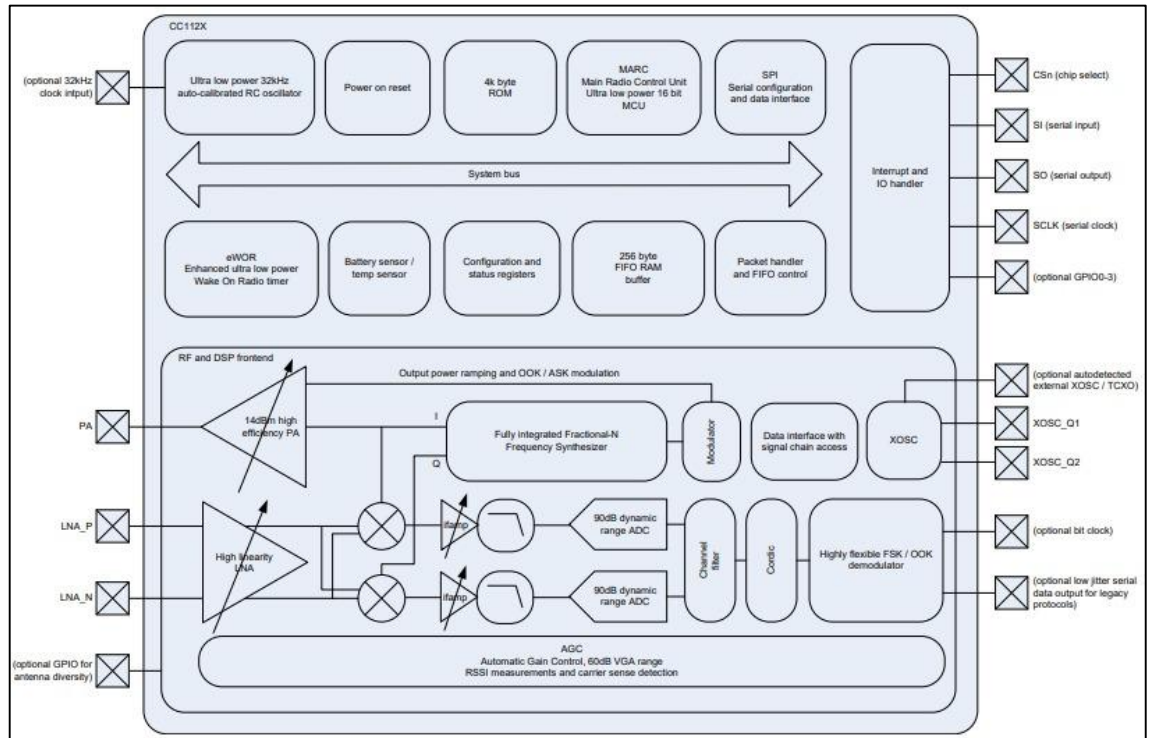


Figure 2.6 Schéma fonctionnel du système

a Le synthétiseur de fréquence

Au centre du CC1120 se trouve un système entièrement intégré, fractionnaire-N, ultra-haute performance figure 2-7. Le synthétiseur de fréquence est conçu pour d'excellentes performances de bruit de phase, offrant une sélectivité et des performances de blocage très élevées. [23]

Le système est conçu pour se conformer aux masques spectraux réglementaires les plus stricts à la puissance d'émission maximale. [23]

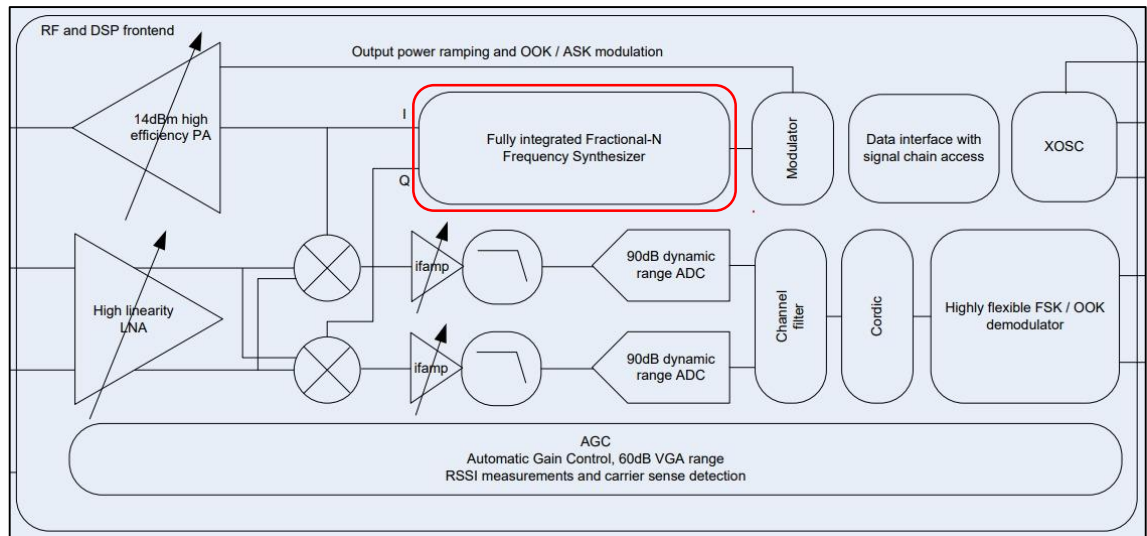


Figure 2.7 Ensemble des composant de la partie RF du cc1120

Soit un cristal connecté à XOSC_Q1 et XOSC_Q2, soit un TCXO connecté à l'entrée EXT_XOSC. L'oscillateur génère la fréquence de référence pour le synthétiseur, ainsi que des horloges pour la partie analogique-numérique (ADC) et la partie numérique. [23]

b Le récepteur

Le CC1120 dispose d'un récepteur très flexible. Le signal RF reçu est amplifié par l'amplificateur à faible bruit (LNA) et est abaissé en quadrature (I/Q) à la fréquence intermédiaire (IF). À IF, les signaux I/Q sont numérisés par les ADC à plage dynamique élevée. [23]

Une unité avancée de contrôle de gain automatique (AGC) ajuste le gain frontal et active le CC1120 pour recevoir des signaux forts et faibles, même en présence d'interférences fortes. [23]

c L'émetteur

L'émetteur du CC1120 est basé sur la synthèse directe de la fréquence RF ou (modulation en boucle). Pour utiliser le spectre de manière efficace, le CC1120 dispose d'un filtrage et d'une mise en forme étendus des données en mode TX pour prendre en charge la communication de données à haut débit dans des canaux à bande étroite.[23]

Le modulateur contrôle également la montée en puissance pour éliminer les problèmes tels que les perturbations spectrales lors du fonctionnement avec des amplificateurs RF externes à haute puissance. [23]

d Radiocommande et interface utilisateur

Le système de contrôle numérique du CC1120 est construit autour de la radiocommande principale (MARC), qui est implémentée à l'aide d'un processeur 16 bits interne ultra-basse consommation haute performance. MARC gère les modes d'alimentation, la radio séquençage et la synchronisation du protocole. [23]

Une interface série SPI à 4 fils est utilisée pour la configuration et l'accès au tampon de données. La bande de base numérique inclut la prise en charge de la configuration des canaux, de la gestion des paquets et de la mise en mémoire tampon des données.[23]

Le microcontrôleur peut rester dans le mode hors tension jusqu'à ce qu'un paquet RF valide soit reçu. Cela réduit considérablement la consommation d'énergie.[23]

Quand le MCU hôte reçoit un paquet RF valide, il lit les données en rafale. Cela réduit la puissance de calcul requise. [23]

2.2.3 Diagramme d'état du circuit

Le CC112X peut être configuré pour obtenir des performances optimales pour de nombreuses applications différentes à l'aide de l'interface SPI. Les paramètres clés suivants peuvent être programmés [23] :

- Mode mise hors tension/sous tension (VEILLE/IDLE)
- Mise sous/hors tension de l'oscillateur à quartz (IDLE/XOFF)
- Mode réception/transmission (RX/TX)
- Fréquence porteuse
- Débit de symbole
- Format de modulation
- Bande passante du filtre du canal RX
- RF output power
- Mise en mémoire tampon des données avec FIFO de réception et de transmission séparées de 128 octets
- Prise en charge du matériel radio par paquets
- Le blanchiment des données (Data Whitening)
- Réveil par radio amélioré (eWOR)

La figure 2-8 montre les différents états du cc1120 [25]

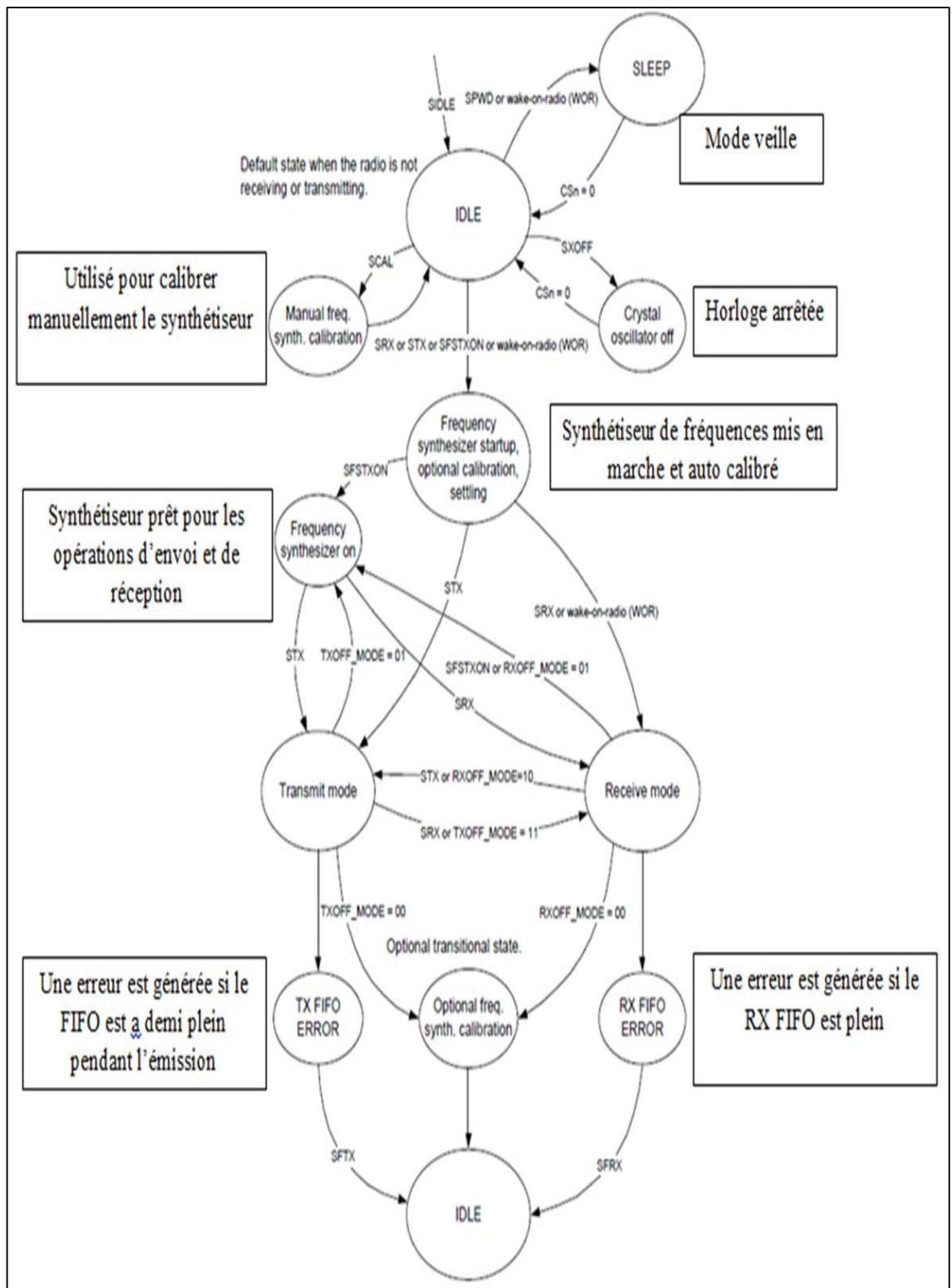


Figure 2.8 Diagramme d'état simplifié du cc1120.

2.2.4 La configuration

a Logiciel de configuration

Le CC112X peut être configuré à l'aide du logiciel SmartRF™ Studio [7]. SmartRF Studio est hautement recommandé pour obtenir des réglages de registre optimaux et pour évaluer les performances et les fonctionnalités.[25]

Après la réinitialisation de la puce, tous les registres ont des valeurs par défaut et celles-ci peuvent différer du paramètre de registre optimal. Il est donc nécessaire de configurer/reconfigurer le circuit via l'interface SPI après la réinitialisation de la puce.[25]

SmartRF Studio fournit une fonction d'exportation de code facilitant la mise en œuvre dans le micrologiciel.[25]

b Configuration du circuit

Dans un système typique, le CC112X s'interfacera avec un MCU. Ce MCU doit être capable de communiquer avec le CC112X via une interface SPI à 4 fils pour pouvoir [25]:

- Configurez le CC112X
- Programmez le CC112X dans différents modes (RX, TX, SLEEP, IDLE, etc.)
- Lecture et écriture de données mises en mémoire tampon (RX FIFO et TX FIFO)
- Lire les informations d'état

Les transferts de données via l'interface SPI passent avec le MSB en premier, dont le circuit cc1120 est configuré en tant qu'esclave.[25]

Toutes les transactions sur l'interface SPI commencent par un octet d'en-tête contenant un bit R/W, un bit d'accès en rafale (B) et une adresse de 6 bits (A5 - A0). Un octet d'état est envoyé sur la broche SO chaque fois qu'un octet d'en-tête est transmis sur la broche SI.[25]

La broche CSn doit être maintenue basse pendant les transferts sur le bus SPI. La synchronisation des transferts d'adresses et de données sur l'interface SPI est illustrée dans la figure 2-9.[25]

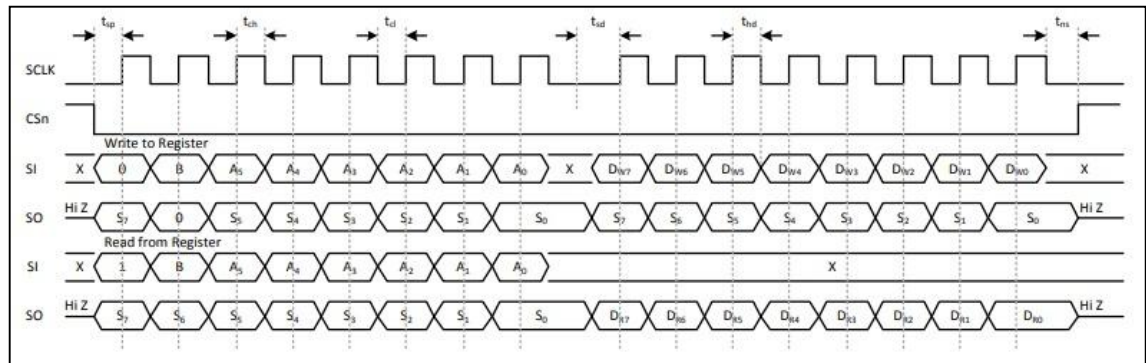


Figure 2.9 Opérations d'écriture et de lecture des registres de configuration

Lorsque CSn est à l'état bas, le MCU doit attendre que la broche CC112X SO passe à l'état bas avant de commencer à transférer l'octet d'en-tête. Cela indique que le cristal est stable. Sauf si la puce vient d'être réinitialisée ou était en état de veille ou XOFF, ou si la configuration XOSC a été modifiée, la broche SO passera toujours au niveau bas immédiatement après avoir tiré CSn au niveau bas.[25]

Lorsque l'octet d'en-tête, l'octet de données ou l'impulsion de commande est envoyé sur l'interface SPI, l'octet d'état de la puce est envoyé par le CC112X sur la broche SO. L'octet d'état contient des signaux d'état clés, utiles pour le MCU.[25]

À l'état IDLE, le XOSC et l'alimentation du cœur numérique sont allumés et tous les autres modules sont hors tension. [25]

La mémoire interne est composée des sections, avec des différents types d'accès SPI, illustrés dans la figure 2-10. [25]

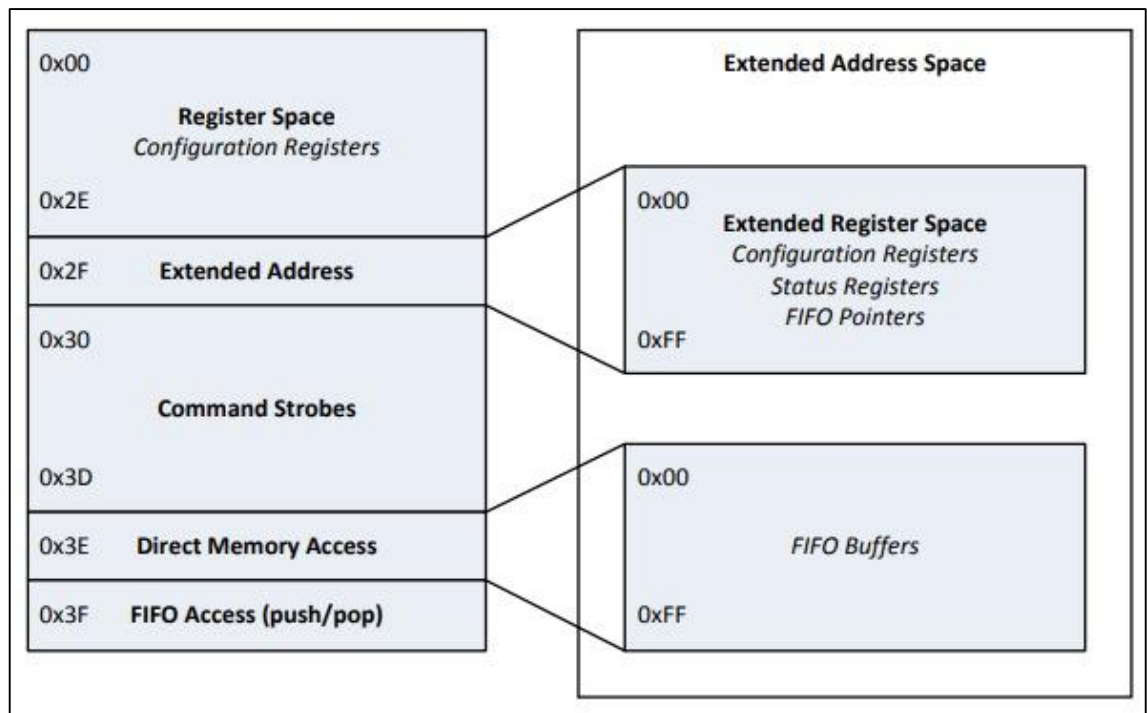


Figure 2.10 Carte de la mémoire du circuit cc1120

Chaque espace mémoire à un type d'accès spécifique, la description détaillée de chacun est mentionnée en Annexe A, avec tous les registres et ses adresses.

c Le logiciel SmartRF Studio

SmartRF™ Studio est une application Windows qui aide les concepteurs de systèmes RF à évaluer facilement la communication radio à un stade précoce du processus de conception pour tous les dispositifs RF basse consommation TI CC1xxx et CC2xxx. Il simplifie la génération des valeurs et des commandes du registre de configuration, ainsi que les tests pratiques et le débogage du système RF.[26]

2.3 Carte à microcontrôleur

Le circuit RF a besoin d'un microcontrôleur, notre choix s'est porté sur une carte contenant un microcontrôleur performant, puissant, et résistant aux environnements scabreux, la carte Nucleo-64 avec un microcontrôleur STM32.

2.3.1 Carte Nucleo-64

La carte STM32 Nucleo-64, figure 2-11, offre aux utilisateurs un moyen abordable et flexible d'essayer de nouveaux concepts et de construire des prototypes en choisissant parmi les différentes combinaisons de fonctionnalités de performance et de consommation d'énergie, fournies par le microcontrôleur STM32. [27]

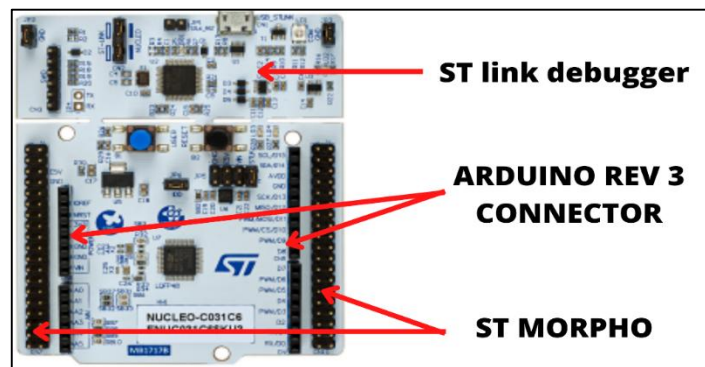


Figure 2.11 Carte Nucleo-64

Le support de connectivité ARDUINO® Uno V3 et les en-têtes (headers) ST morpho permettent une extension facile des fonctionnalités de la plate-forme de développement ouverte STM32 Nucleo avec un large choix de boucliers spécialisés. [27]

La carte STM32 Nucleo-64 ne nécessite aucune sonde séparée puisqu'elle intègre le débogueur/programmeur ST-LINK.[27]

2.3.2 Microcontrôleur STM32L476

La famille STM32 de microcontrôleurs 32 bits basés sur le processeur Arm® Cortex®-M est conçue pour offrir de nouveaux degrés de liberté aux utilisateurs de MCU. [28]

Elle propose des produits combinant de très hautes performances, des capacités en temps réel, un traitement numérique du signal, un fonctionnement bas consommation/basse tension et une connectivité, tout en conservant une intégration complète et une facilité de développement.[28]

Les dispositifs STM32L476xx sont des microcontrôleurs ultrabasse consommation basés sur le cœur RISC 32 bits Arm® Cortex®-M4 hautes performances fonctionnant à une fréquence pouvant atteindre 80 MHz. [29]

Le cœur Cortex-M4 comprend une unité à virgule flottante (FPU) simple précision qui prend en charge toutes les instructions de traitement de données et tous les types de données Arm® à simple précision. [29]

Il implémente également un ensemble complet d'instructions DSP et une unité de protection de la mémoire (MPU) qui améliore la sécurité des applications. [29]

Les microcontrôleurs STM32L476xx embarquent des mémoires à grande vitesse (mémoire Flash jusqu'à 1 Mo et jusqu'à 128 Ko de SRAM), un contrôleur de mémoire externe flexible (FSMC) pour les mémoires statiques (pour les appareils avec des boîtiers de 100 broches et plus), une interface de mémoires flash pour le Quad SPI (disponible sur tous les packages) et une gamme étendue d'E / S améliorées et de périphériques connectés à deux bus APB, deux bus AHB et une matrice de bus multi-AHB 32 bits.[29]

Ils offrent jusqu'à trois ADC 12 bits rapides (5 Msps), deux comparateurs, deux amplificateurs opérationnels, deux canaux DAC, un tampon de référence de tension interne, un RTC basse consommation, deux temporisateurs 32 bits à usage général, deux temporisateurs PWM 16-bits dédiés au contrôle du moteur, sept temporisateurs 16 bits à usage général et deux temporisateurs 16 bits basse consommation, ils prennent en charge aussi quatre filtres numériques pour les modulateurs sigma delta externes (DFSDM).[29]

Ils disposent également d'interfaces de communication standard et avancées :[29]

- Trois I2C
- Trois SPI
- Trois USART, deux UART et un UART basse consommation.
- Deux SAI (interfaces audios série)
- Un SDMMC (Secure Digital and MultiMedia Card)
- Un CAN
- Un port USB OTG pleine vitesse
- Une interface SWPMI (Single Wire Protocol Master Interface)

Les appareils STM32L476xx comportent jusqu'à 1 Mo de mémoire Flash intégrée disponible pour stocker les programmes et les données. La mémoire Flash est divisée en deux banques permettant des opérations de lecture pendant l'écriture.[29]

Cette fonctionnalité permet d'effectuer une opération de lecture à partir d'une banque pendant qu'une opération d'effacement ou de programmation est effectuée sur l'autre banque. Le démarrage à double banque est également pris en charge. Chaque banque contient 256 pages de 2 Ko. [29]

Les dispositifs STM32L476xx comportent jusqu'à 128 Ko de SRAM intégrée. Cette SRAM est divisée en deux blocs :

- 96 Ko mappés à l'adresse 0x2000 0000 (SRAM1)
- 32 Ko situés à l'adresse 0x1000 0000 avec contrôle de parité matériel (SRAM2).

Le Module RF, s'interface avec le microcontrôleur via SPI et communique avec le RTU SAFIR via l'interface UART, les deux interfaces seront détaillées dans ce qui suit.

2.3.3 L'interface SPI

Le SPI est un bus de communication série synchrone Full-duplex, il a été développé par Motorola pour remplacer les interfaces parallèles. Les circuits communiquent selon un schéma maître-esclave, où le maître s'occupe totalement de la communication [30].

Il peut assurer ainsi une communication entre un microcontrôleur et des périphériques divers.

Ce protocole a été conçu pour permettre des transferts de données entre des circuits intégrés différents, situés sur le même PCB à des vitesses très élevées.

Il est préférable de maintenir les lignes de bus le plus court possible entre les différents éléments pour assurer une bonne liaison.

Il est caractérisé par : [30]

- Mode maître et mode esclave
- Mode bidirectionnel
- Sortie sélection d'esclave
- Registre de données à double tampon
- Indicateur d'erreur de mode avec capacité d'interruption du processeur
- Horloge série avec polarité et phase programmable.

a Les lignes du bus SPI

- MOSI : Cette broche est utilisée pour transmettre des données hors du module SPI lorsqu'il est configuré en tant que maître et recevoir des données lorsqu'il est configuré en tant qu'esclave. [30]
- MISO : Cette broche est utilisée pour transmettre des données hors du module SPI lorsqu'il est configuré en tant qu'esclave et recevoir des données lorsqu'il est configuré en tant que maître. [30]
- SCK/ SCLK : Cette broche est l'horloge générée par le maître. [30]
- SS/CS : Cette broche est utilisée pour émettre le signal de sélection en cas de configuration en tant que maître et est utilisé comme entrée en cas de configuration en tant qu'esclave. [30]

On trouve souvent SDI (Serial Data In) et SDO (Serial Data Out) désignant MOSI et MISO respectivement, dans le nommage du circuit esclave.

Le maître gère la communication en utilisant les deux lignes SCLK et CS, appelées lignes de contrôle, l'échange des données est initié par le maître en générant le signal d'horloge, ensuite il sélectionne l'esclave désiré par la broche CS en la tirant à l'état bas.[30]

Afin d'établir le lien entre les périphériques, le brochage illustré dans la figure 2-12 doit-être respecté [31] :

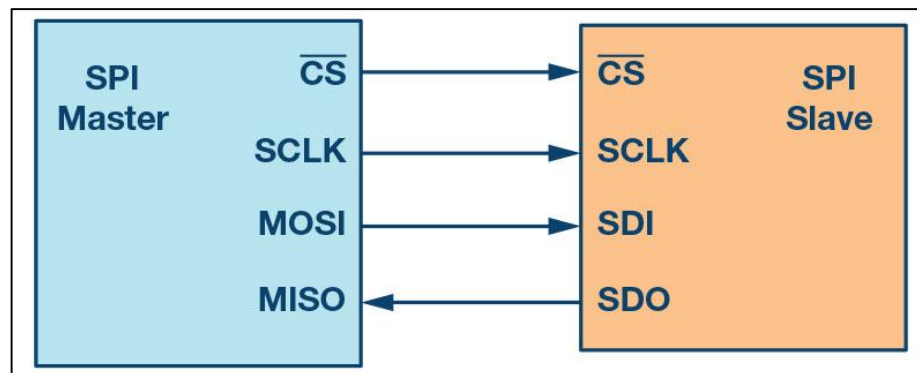


Figure 2.12 Connexion SPI entre un périphérique maitre et un périphérique esclave

L'interface SPI du STM32, est configurable pour supporter plusieurs protocoles de communication série standard. Les applications bénéficient d'une simple et directe connexion qui requiert quelques pins seulement. [31]

Dans l'architecture des microcontrôleurs STM32 de la famille L4, les registres consacrés à la transmission et à la réception peuvent avoir une taille de 16 bits.[32]

b Les modes de communication

Plusieurs interconnexions maitre-esclaves sont supportées :

b.1 Mode Full-duplex : communication bidirectionnelle

Le maitre contrôle toujours le flux dans le bus, et fournit le signal de l'horloge à l'esclave par la ligne SCK. Il sélectionne l'esclave avec qui, il va communiquer, à travers un signal de sélection NSS (Slave Select).[32]

Les données enregistrées dans les registres à décalage appropriés disposent de la capacité d'être échangés de manière synchrone par les lignes MOSI et MISO.[32]

Lorsque les deux nœuds maitre et esclave communique en mode full-duplex, les deux lignes sont utilisées et un flux de données synchrones passe dans les deux directions en même temps.[32]

La figure 2-13 illustre ce mode.[32]

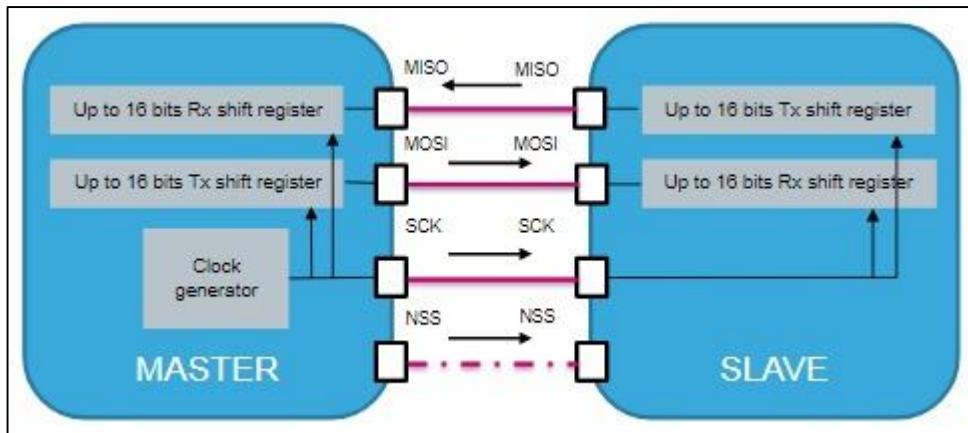


Figure 2.13 Connexion SPI mode full-duplex

b.2 Mode Simplex : communication unidirectionnelle

Un nœud écoute quand l'autre transmet. Les données circulent dans une seule direction, et la broche dédiée est exclusivement employée pour le transfert de données. Les broches non utilisées peuvent être exploitées pour d'autres objectifs.[32]

La connexion entre le maitre et l'esclave dans les deux configurations est montré par les deux figures 2-14 et 2-15.[32]

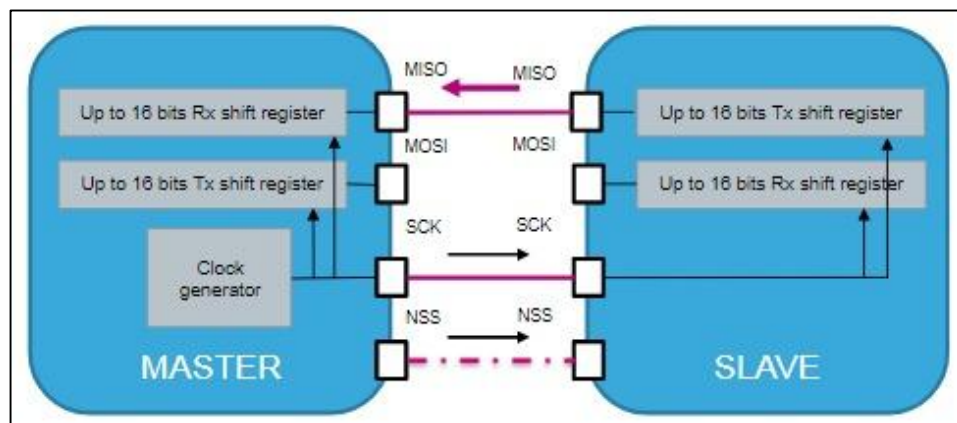


Figure 2.14 Connexion SPI mode simplex (maitre-récepteur)

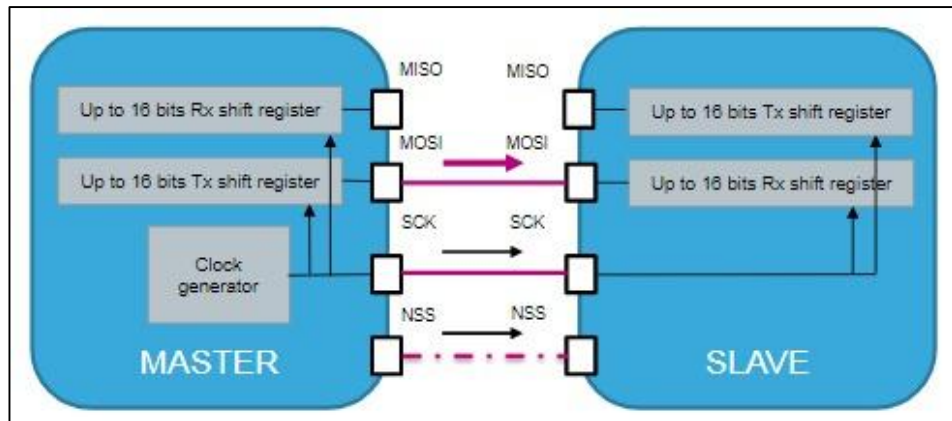


Figure 2.15 Connexion SPI mode simplex (maitre-émetteur)

b.3 Mode Half-duplex : communication quasi-bidirectionnelle

Les deux nœuds, maitre et esclave alternent entre la transmission et la réception des données de manière synchrone. Ils partagent une ligne commune pour le transfert.[32]

Ce mode intègre les deux modes précédents par l'occupation d'une seule ligne pour dialoguer, en un seul sens à la fois. Ce mode fonctionne avec une connexion croisée entre le MOSI du maitre et le MISO de l'esclave. [32]

La reconfiguration des directions au niveau des nœuds n'est souvent pas en accord, d'où le rajout d'une résistance sur la ligne des données pour éviter les courts-circuits, comme illustré dans la figure 2-16.[32]

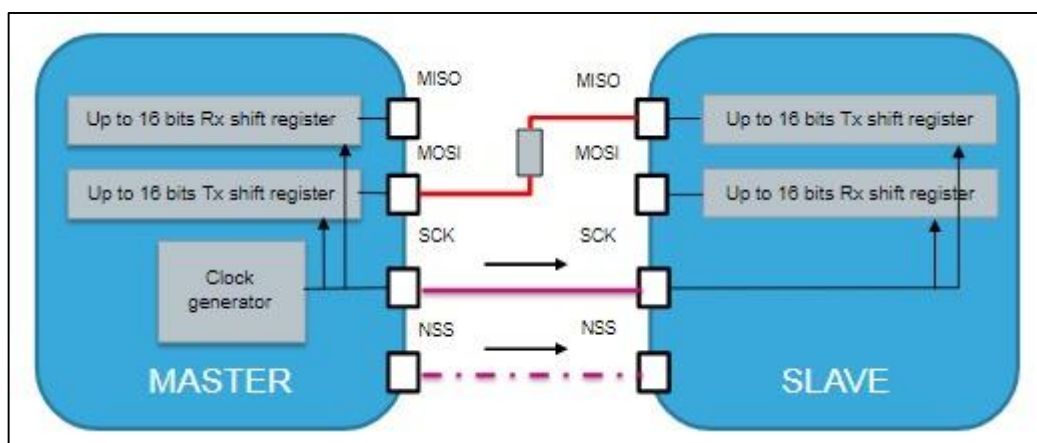


Figure 2.16 Connexion SPI mode Half-duplex

c Topologies des connexions SPI

Parfois on a besoin de connecter plusieurs nœuds au même réseau SPI, d'où l'exemple d'un système contenant un afficheur et un lecteur de carte mémoire, la figure ci-dessous montre l'interfaçage du microcontrôleur avec ces différents composants ?

Le microcontrôleur STM32L4xx supporte différentes topologies :

c.1 Topologies Multi-Slave

- Topologie Etoile

Généralement le maître sélectionne un seul esclave à la fois, donc des signaux de sélection séparés sont indispensables, et ils sont simulés par des pins GPIO. [32]

Quand un réseau SPI inclut plus qu'un seul esclave, une topologie en étoile est typiquement appliquée lorsque tous les signaux SCK, MOSI et MISO sont interconnectés. [32]

Lors de l'utilisation des signaux de sélection séparés, des différents formats de configuration de données et d'horloge sont facultativement appliqués pour les esclaves. [32]

La figure 2-17 montre la connexion entre les différents dispositifs. [32]

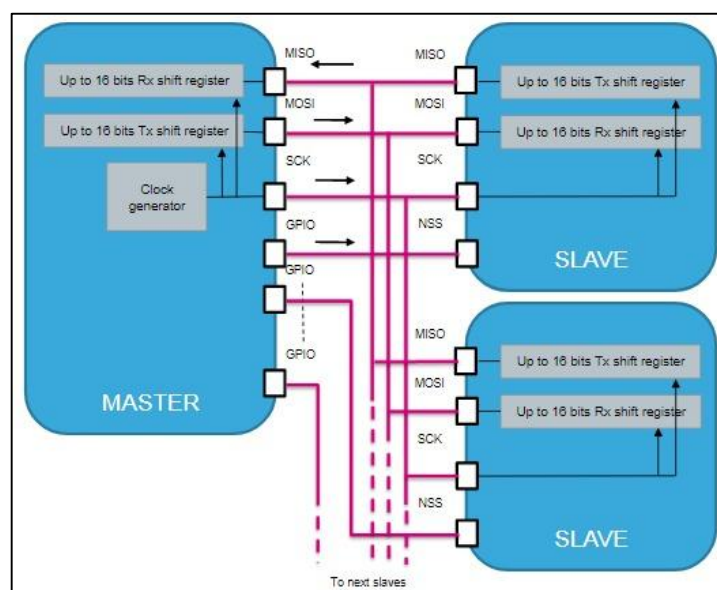


Figure 2.17 Topologie SPI Multi-slave Etoile

- Topologie Circulaire

Dans cette topologie toutes les entrées et sorties des nœuds sont connectées entre eux, formant une chaîne série fermée. Elle illustré dans la figure [32]

Les données circulent par tous les nœuds, ces derniers doivent être configurés avec le même format des données et d'horloge. [32]

Le traitement de données se passe à l'intérieure du programme, un temps entre les transmissions est lissé par le maitre. [32]

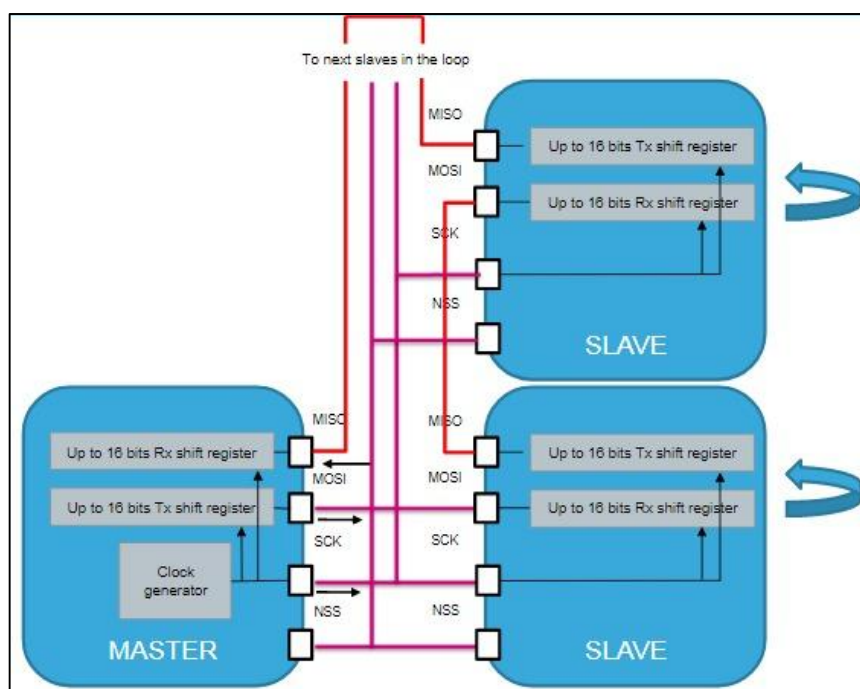


Figure 2.18 Topologie SPI Multi-slave Circulaire

c.2 Topologie Multi-Master

Ce mode permet à deux maitres d'être connectés ensemble, ils sont configurés par default en tant que des esclaves.[32]

Au début de la communication, un nœud change son statut en maitre pour avoir le contrôle sur le bus.[32]

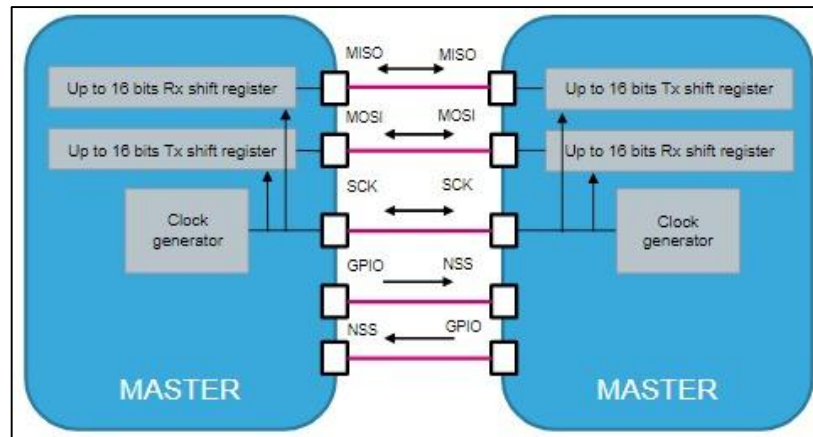


Figure 2.19 Topologie SPI Multi-master

Les broches de sélection fonctionnent comme entrées, figure 2-19 pour détecter un conflit de collision potentiel entre les nœuds parce qu'un seul peut utiliser le bus. [32]

A la fin d'une session de communication, le maître désactive le signal de sélection et revient en mode esclave.[32]

d Paramètres du SPI

Quelques paramètres du bus SPI caractérisant l'interface SPI du STM32L4xx sont mentionnés ci-dessous, d'autres informations supplémentaires se trouvent en référence [32].

- La taille de la donnée : de 4 à 16 bits
- **L'ordre de décalage** : peut-être configurer comme MSB (Most significant bit) en premier ou bien LSB (least significant bit).
- **L'horloge** : il existe quatre modes définis dans la spécification du SPI Motorola figure 2-20.[33]

La combinaison de deux bits contrôle la polarité et la phase de l'horloge, Lorsque le bit de phase est mis à 0, les bits sont échantillonnés sur le front impair, et le front pair synchronise le décalage du bit prochain vers la ligne de données.[32]

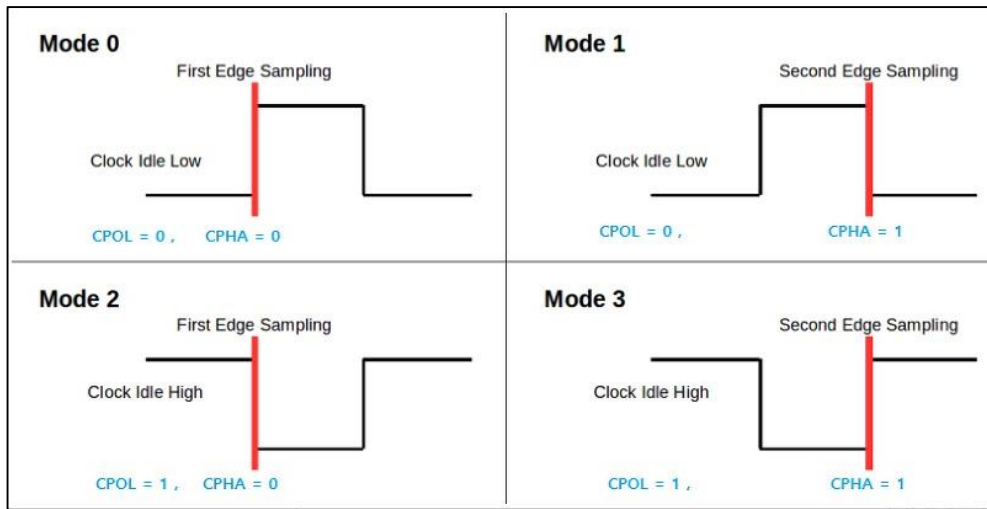


Figure 2.20 Modes du SPI

Le contraire passe lorsque le bit de phase est mis à 1.

La polarité de l'horloge définit l'état de repos initial du signal d'horloge, donc elle détermine quel est le front utilisé pour l'échantillonnage et pour le décalage.[32]

Un exemple de mode 2 du SPI est illustré dans la figure 2-21.[32]

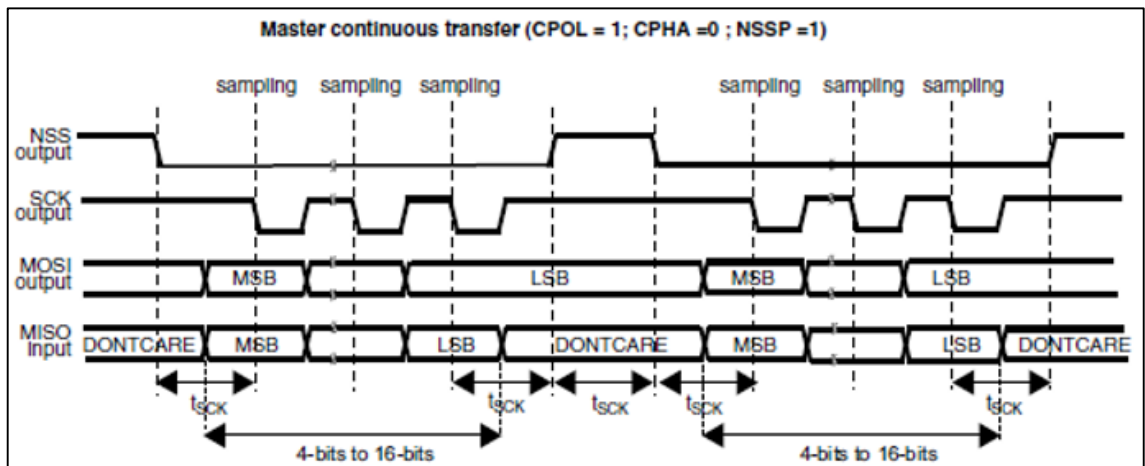


Figure 2.21 Trame du protocole SPI

Le SPI est utilisé dans les applications où le transfert simple de données est requis, sans besoin d'un protocole de communication compliqué.

2.3.4 L'interface USART

L'interface de communication USART permet aux microcontrôleurs d'échanger des données avec d'autres systèmes embarqués, ces données sont envoyées et reçues en série un bit à la fois.

Cette interface existante au niveau du microcontrôleur STM32L476, qu'on a utilisé pour relier le Module RF avec le RTU SAFIR, est un module série très flexible qui prend en charge :[34]

- Communication UART asynchrone,
- Le mode maître SPI (interface périphérique série),
- Mode LIN (réseau d'interconnexion local).

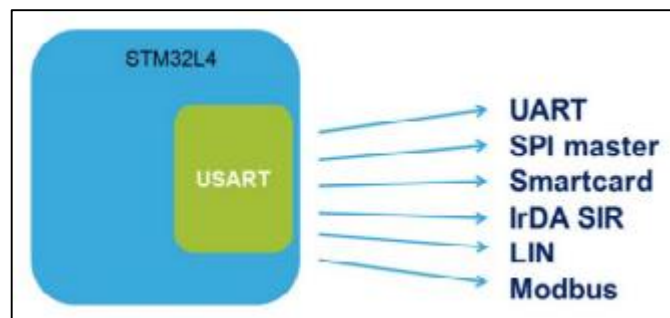


Figure 2.22 Communications supportées par l'interface USART

Il peut également s'interfacer avec les cartes à puce ISO/IEC 7816 et les dispositifs IrDA. Il fournit également certaines fonctionnalités qui sont utiles lors de la mise en œuvre des communications Modbus.[34]

Les applications utilisant l'USART bénéficient d'une connexion simple et peu coûteuse entre les appareils, qui ne nécessite que quelques broches. De plus, le périphérique USART est fonctionnel dans les modes basse consommation.[34]

L'USART est une interface série entièrement programmable comprenant :[34]

- la longueur de données : les données peuvent être de 7, 8 ou 9 bits.
- la parité : pair, impair et sans parité

- le nombre de bits de stop : 0.5, 1, 1.5 et 2 bits d'arrêt
- l'ordre des données : est programmable avec MSB ou LSB en premier.
- générateur de débit en bauds et mode de suréchantillonnage configurable par 8 ou par 16.

Elle offre également la possibilité d'utiliser le contrôle de flux RS-232 de base avec les signaux CTS (Clear To Send) et RTS (Request To Send). Le signal RS-485 DE (Driver Enable) est également pris en charge.[34]

L'USART prend en charge un domaine d'horloge double permettant le réveil à partir du mode d'arrêt et la programmation de la vitesse de transmission indépendamment de l'horloge périphérique (PCLK). [34]

L'USART dispose d'un mode multiprocesseur qui permet à l'USART de rester inactif lorsqu'il n'est pas adressé. [34]

Il existe deux de type de liaison : synchrone et asynchrone, une différence entre eux est qu'une ligne d'horloge n'est pas présente dans la liaison asynchrone, mais des bits de synchronisation sont ajoutés au mot à envoyer.

La figure 2-23 représente comment une liaison est établit entre deux circuits.[35]

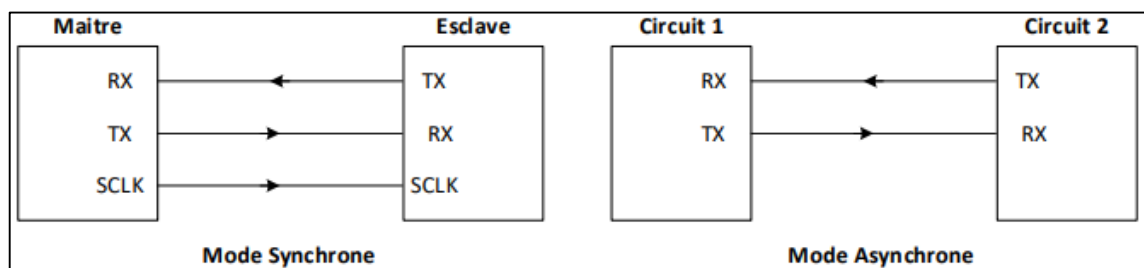


Figure 2.23 Mode Synchrone et Asynchrone de l'USART

La trame commence avec un bit de début de transmission (start-bit) qui est équivalent à amener la ligne au niveau bas pour une période d'un bit, son objectif est la synchronisation du transfert des données.[34]

La figure suivante illustre comment la trame est formée [34] :

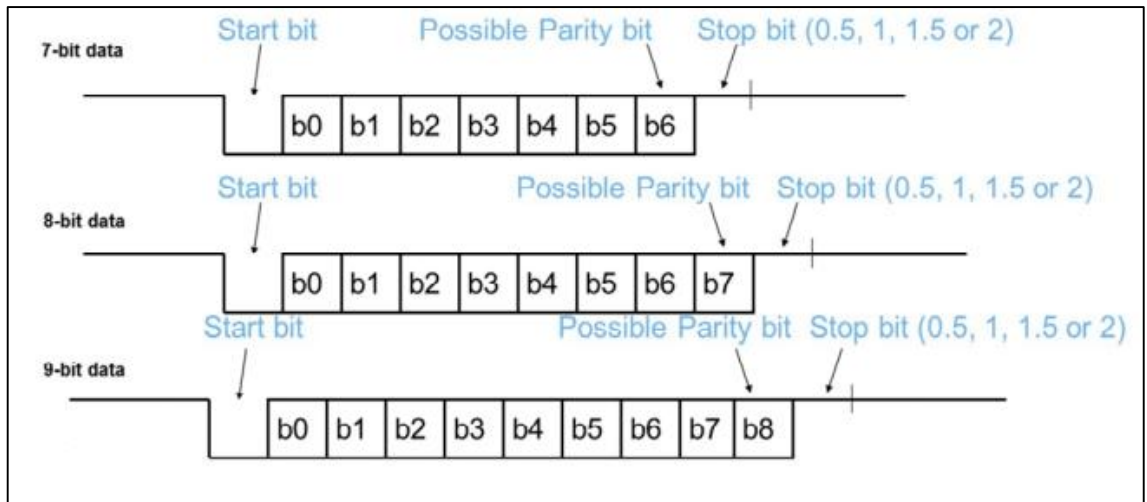


Figure 2.24 Trame du protocole USART

Ce bit est suivi par les bits du message de longueur 7,8, ou 9 bits, si le contrôle de parité est activé un autre bit est ajouté, le bit de parité fait partie du message. A la fin, un/des bit(s) de fin de transmission, signalent que la transmission est terminée.

L'USART prend en charge la communication en full-duplex où les lignes Tx et Rx sont respectivement connectées aux lignes Rx et Tx de l'autre interface. [34]

Elle peut être configurée pour suivre un protocole monofilaire semi-duplex où les lignes Tx et Rx sont connectées en interne. Dans ce mode de communication, seule la broche Tx est utilisée pour l'émission et la réception.[34]

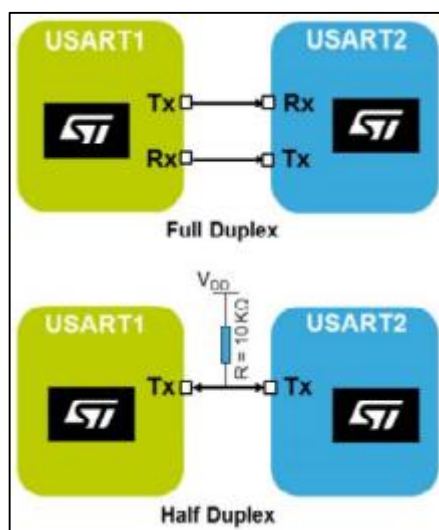


Figure 2.25 Configuration Full-duplex et Half-duplex

La broche Tx est toujours relâchée lorsqu'aucune donnée n'est transmise, elle agit donc comme une E/S standard en mode inactif ou réception. Cela signifie que les E/S doivent être configurées de manière que Tx soit configuré en tant que drain ouvert à fonction alternative avec un pull-up externe.[34]

L'interface supporte aussi les standards de communication RS-232 et RS-485.

- Le RS-232

En communication RS-232, il est possible de contrôler le flux de données série entre 2 appareils en utilisant l'entrée nCTS et la sortie nRTS. Ces deux lignes permettent au récepteur et à l'émetteur de s'alerter mutuellement de leur état. [34]

La figure 2-26[34], montre comment connecter 2 appareils dans ce mode.

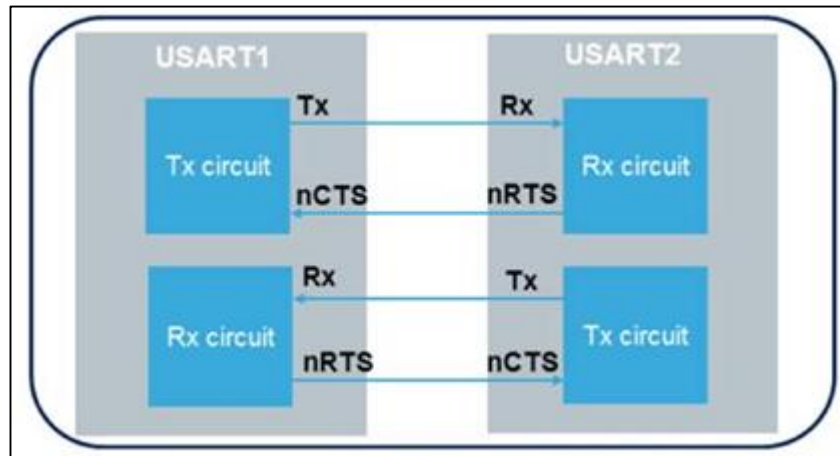


Figure 2.26 Connexion RS-232 sous USART

L'idée est d'éviter les pertes d'octets ou les conflits en cas de communication Half-duplex. [34]

- Le RS-485

Pour les protocoles de communication semi-duplex série comme RS-485, le maître doit générer un signal de direction pour contrôler l'émetteur-récepteur (couche physique). Ce signal informe la couche physique si elle doit agir en mode émission ou réception. [34]

En mode RS-485, une ligne de contrôle est utilisée : la broche Driver Enable est utilisée pour activer le contrôle externe de l'émetteur-récepteur. DE partage la broche avec nRTS. [34]

2.3.5 Débogueur et programmeur ST-Link

Le ST-LINK est un débogueur et programmeur en circuit pour les familles de microcontrôleurs STM8 et STM32. [36]

Le ST-LINK est un périphérique USB figure 2-27 [37], et doit être connecté à un PC hôte, car il est la partie responsable dans la communication entre le PC et le microcontrôleur.[36]

Il peut être soit intégré sur des cartes ST, soit fourni en tant que dongle autonome.[36]

Le débogueur est tout aussi important pour le fonctionnement du code car non seulement il alimente le microcontrôleur, mais il a aussi un accès direct aux registres internes, l'utilisateur pourra alors changer ces valeurs en cas d'erreurs ou de bugs.



Figure 2.27 Débogueur/programmeur ST-Link

2.3.6 Environnement STM32CubeIDE

STM32CubeIDE est un outil de développement multi-OS tout-en-un faisant partie de l'écosystème logiciel STM32Cube. STM32CubeIDE est une plate-forme de développement C/C++ avancée avec des fonctionnalités de configuration de périphérique, de génération de code, de compilation de code et de débogage pour les microcontrôleurs et microprocesseurs STM32.[38]

Il est basé sur le Framework Eclipse®/CDT™ et la chaîne d'outils GCC pour le développement, et GDB pour le débogage. Il permet l'intégration des centaines de plugins existants qui complètent les fonctionnalités de l'IDE Eclipse®.[38]

STM32CubeIDE intègre les fonctionnalités de configuration et de création de projets STM32 de STM32CubeMX pour offrir une expérience d'outil tout-en-un et gagner du temps d'installation et de développement.[38]

Après la sélection d'un MCU ou MPU STM32 vide, ou d'un microcontrôleur ou d'un microprocesseur préconfiguré parmi la sélection d'une carte ou la sélection d'un exemple, le projet est créé et le code d'initialisation est généré. [38]

À tout moment du développement, l'utilisateur peut revenir à l'initialisation et à la configuration des périphériques ou middleware et régénérer le code d'initialisation sans impact sur le code utilisateur. [38]

STM32CubeIDE comprend également des fonctionnalités de débogage standard et avancées, notamment des vues des registres centraux du processeur, des mémoires et des registres périphériques, ainsi qu'une surveillance des variables en direct, une interface Serial Wire Viewer ou un analyseur de défauts. [38]

2.4 Carte d'adaptation

Etant donné que la carte Nucleo-64 comporte un connecteur Arduino Uno Rev3, on a ainsi pu utiliser un adaptateur Arduino-Click, figure 2-28 de chez MikroElektronika.[39]

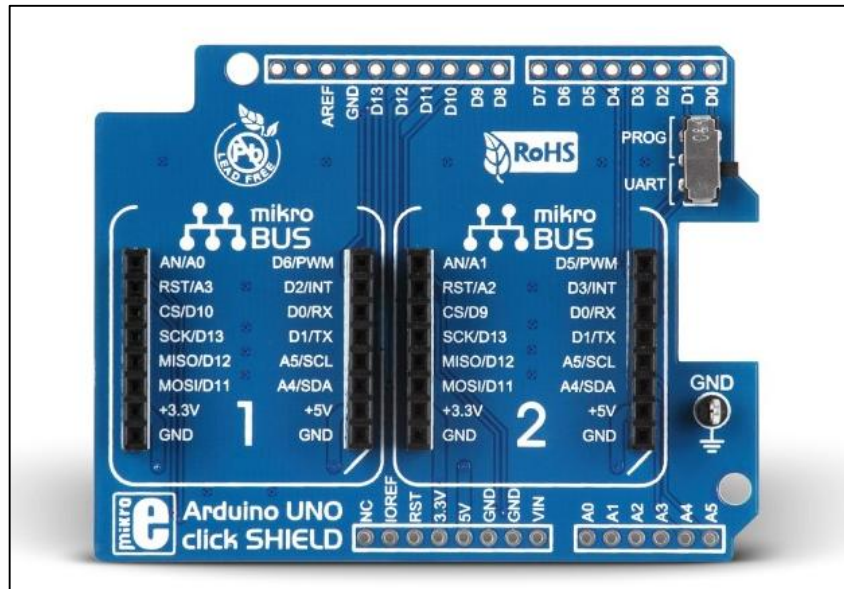


Figure 2.28 Bouclier Arduino Uno - click

Le bouclier de click est une extension pour Arduino Uno et toute autre carte compatible Arduino. Il s'agit d'un simple bouclier avec deux prises hôtes mikroBUS qui vous permettent de connecter plus de 750 types différents de cartes de clic à l'Arduino.[39]

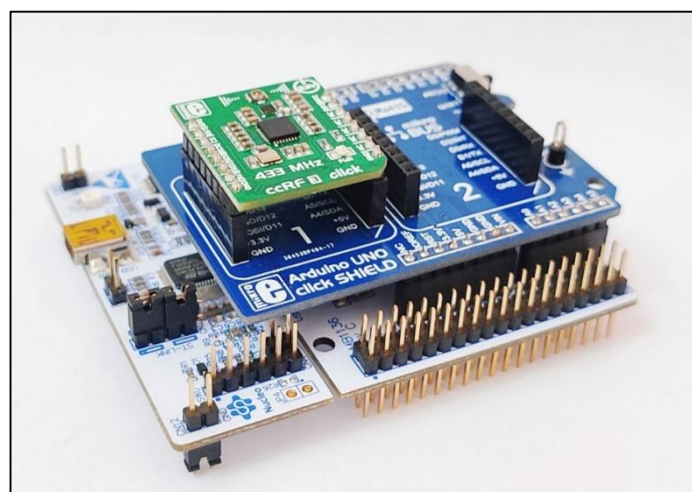


Figure 2.29 Assemblage des trois cartes de test

Après avoir franchie toutes les étapes nécessaires, on a procédé au montage de nos trois cartes, comme le montre la Figure 2-29.

2.5 Logiciel Arduino IDE

Une plateforme de prototypage open-source permettant aux utilisateurs et aux développeurs de créer librement leurs codes (sketches).

L'Environnement de développement intégré ARDUINO utilise le C++ comme langage d'interprétation de base, et comporte des fonctions et des méthodes spéciales.

2.6 Analyseur logique

Le Digilent Analog Discovery™, développé en collaboration avec Analog Devices Inc., est un instrument multifonction qui peut mesurer, enregistrer et générer des signaux analogiques et numériques. [10]

Le dispositif Discovery analogique portable et à faible coût (figure 2-30) a été créé pour que les étudiants en ingénierie puissent travailler avec des circuits analogiques et numériques à tout moment, n'importe où, directement depuis leur PC. [10]

Les entrées et sorties analogiques et numériques de l'Analog Discovery se connectent à un circuit à l'aide de simples sondes à fil. Les entrées et les sorties sont contrôlées à l'aide du logiciel gratuit WaveForms basé sur PC qui peut configurer le Discovery pour qu'il fonctionne comme l'un des nombreux instruments traditionnels. [10]

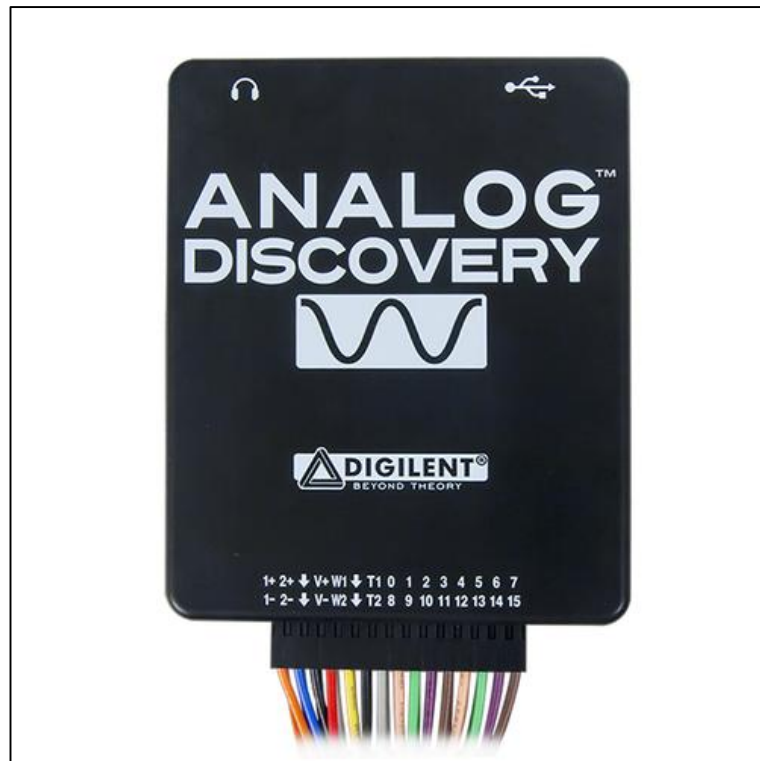


Figure 2.30 Digilent Analog Discovery™

Les instruments comprennent [10] :

- Oscilloscope à deux canaux (1 M Ω , ± 25 V, différentiel, 14 bits, 100 MS/s, bande passante de 5 MHz) ;
- Générateur de fonctions arbitraires à deux canaux (22 Ω , ± 5 V, 14 bits, 100 MS/s, bande passante de 5 MHz) ;
- Amplificateur audio stéréo pour piloter des écouteurs ou des haut-parleurs externes avec des signaux AWG répliqués ;
- Analyseur logique numérique 16 canaux (CMOS 3,3 V, 100 MS/s) ;
- Générateur de motifs 16 canaux (CMOS 3,3 V, 100 MS/s) ;
- E/S numériques virtuelles à 16 canaux comprenant des boutons, des commutateurs et des voyants - idéal pour les applications d'entraînement logique ;
- Deux signaux de déclenchement numériques d'entrée/sortie pour relier plusieurs instruments (CMOS 3,3 V) ;

- Deux alimentations (+5V à 50mA, -5V à 50mA).
- Voltmètre monocanal (AC, DC, $\pm 25V$) ;
- Analyseur de réseau - Bode, Nyquist, diagrammes de transfert de Nichols d'un circuit. Gamme : 1 Hz à 10 MHz ;
- Analyseur de spectre - spectre de puissance et mesures spectrales (plancher de bruit, SFDR, SNR, THD, etc.) ;
- Analyseurs de bus numériques (SPI, I2C, UART, Parallèle) ;

Le schéma fonctionnel de haut niveau d'Analog Discovery est présenté dans la figure 2-31. Le cœur de l'Analog Discovery est le FPGA Xilinx Spartan 6 (en particulier, le dispositif XC6SLX16-1L).[10]

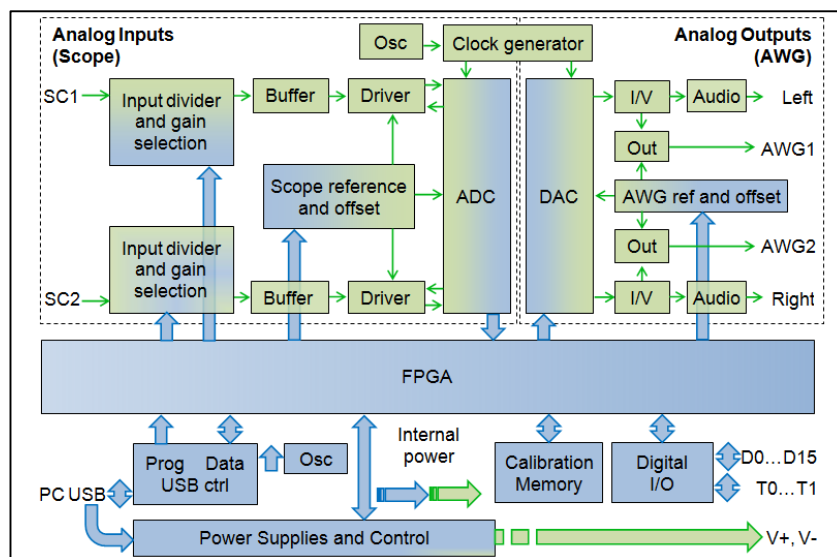


Figure 2.31 Schéma fonctionnel du Digilent Analog Discovery™

Le logiciel WaveForms programme automatiquement le FPGA de Discovery au démarrage avec un fichier de configuration conçu pour mettre en œuvre un instrument de test et de mesure multifonction. [10]

Une fois programmé, le FPGA communique avec le logiciel WaveForms sur PC via une connexion USB 2.0. Le logiciel WaveForms fonctionne avec le FPGA pour contrôler tous les blocs fonctionnels de l'Analog Discovery, y compris la définition des paramètres, l'acquisition des données, ainsi que le transfert et le stockage des données.[10]

Caractéristiques et performances de l'analyseur logique [10] :

- 16 signaux partagés entre l'analyseur, le générateur de modèles et les E/S discrètes.
- 100 MS/s, avec des tampons supportant jusqu'à 16 000 transitions par broche.
- Entrées de niveau logique LVCMOS (3,3 V).
- Plusieurs options de déclenchement, y compris le changement de broche, le modèle de bus, etc.
- L'entrée/sortie de déclenchement permet de relier plusieurs instruments.
- Déclenchement croisé entre les canaux d'entrée analogiques, l'analyseur logique, le générateur de modèles ou le déclenchement externe.
- Interprète pour SPI, I2C, UART, bus parallèle.
- Les signaux capturés peuvent être enregistrés et exportés dans des formats de fichier standard.

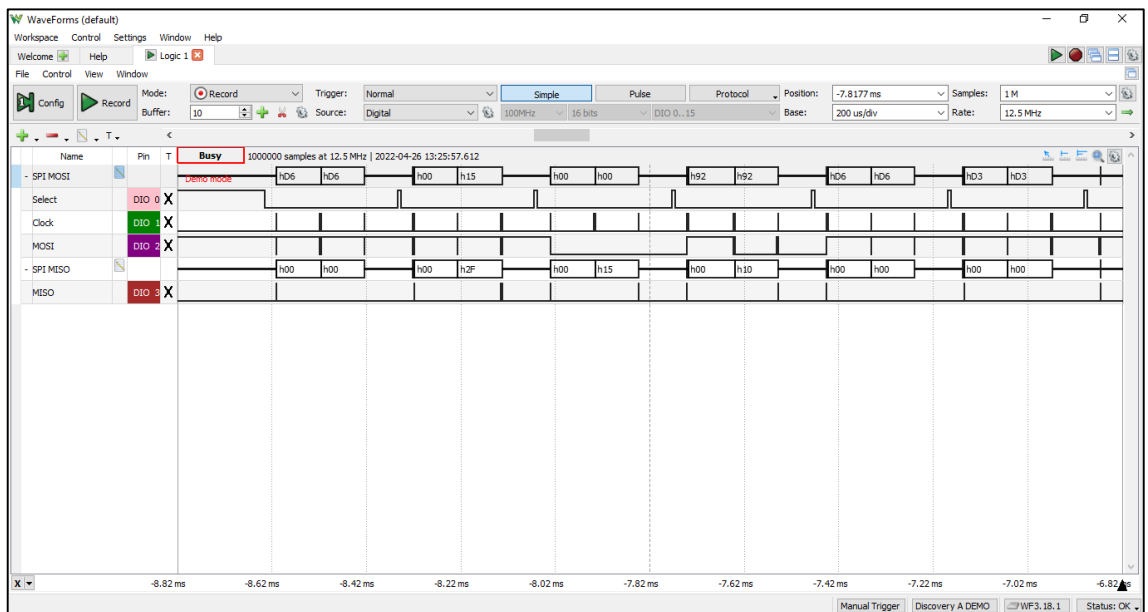


Figure 2.32 Capture de l'interface WaveForms

La figure 2-32, présente la visualisation des signaux du bus SPI sur l'interface du logiciel WaveForms, pendant des tests de configuration du circuit cc1120.

2.7 Carte STEVAL-FKI433V1 et le logiciel S2-LP-DK

La carte d'évaluation STEVAL-FKI433V1 est basée sur l'émetteur-récepteur S2-LP sous-1 GHz à faible débit de données ultra-basse consommation adaptée aux bandes ISM et au M-Bus sans fil.[11]

La carte NUCLEO-L152RE est équipée d'un microcontrôleur basse consommation STM32L1 pour contrôler le S2-LP, cette carte intègre un débogueur/programmeur ST-LINK/V2-1 pour la mise à jour du firmware.[11]

La figure 2-33 [11], montre le système utilisé durant les tests de transmission lors du développement des programmes.



Figure 2.33 Carte STEVAL-FKI433V1 avec la NUCLEO-L152RE

Caractéristique du circuit S2-LP [11] :

- Puissance du sorite RF programmable jusqu'à +16dBm
- Modulations supportées : 2-FSK, 2GFSK, 4-FSK, 4-GFSK, OOK et ASK.
- Vitesse de transmission de 0.3 à 500 kbps
- Faible consommation : 6.7 mA RX, 10 mA TX at + 10 dBm
- une sensibilité de réception jusqu'à -130 dBm
- Interface SPI pour le microcontrôleur

- Interface USB

Pour une utilisation simple et efficace le constructeur a mis en place une application graphique, S2-LP_DK sous forme d'une interface PC interactive pour les registres sur le S2-LP, pour la configuration de la partie radio et le gestionnaire de paquets.[12]

Il fournit une bibliothèque S2-LP avec un ensemble complet d'API pour s'interfacer avec les fonctionnalités S2-LP, ainsi qu'un ensemble d'applications démontrant l'utilisation des fonctionnalités du même appareil. Chaque application de démonstration est accompagnée d'un ensemble complet de fichiers source.[12]

2.8 Logiciel KiCad

Nous avons opté pour le logiciel KiCad pour réaliser notre schéma car non seulement il offre toutes les bibliothèques dont on a besoin dans notre conception De plus il est gratuit. Il se compose de trois parties CAO, schematic capture, PCB design et 3D Viewer.

- L'éditeur de schémas de KiCad prend en charge tout, du schéma le plus basique à une conception hiérarchique complexe avec des centaines de feuilles.
- L'éditeur de PCB de KiCad est suffisamment accessible pour faciliter la conception de votre premier PCB et suffisamment puissant pour les conceptions modernes complexes.
- La visionneuse 3D de KiCad permet une inspection facile du circuit imprimé pour vérifier l'ajustement mécanique et prévisualiser me produit fini.

2.9 Conclusion

Pour conclure, cette de tous les outils, les éléments du système de transmission, la carte à microcontrôleur et ses environnements nous à réaliser et concevoir le module de communication dont la démarche est détaillé dans le chapitre suivant.

3.1 Introduction

Dans ce chapitre de conception et de réalisation, nous allons définir toutes les étapes de conception du module RF, décrites et évaluées par des tests et par des réalisations.

Pour implémenter notre système de communication, nous avons conçu le module RF qui répond aux besoins suivants :

- Prise en charge de la technologie RF Narrow Band.
- Conception d'une bibliothèque qui gère les différentes fonctions du module RF
- Permet de faciliter l'accès à l'utilisateur pour envoyer ou recevoir des données avec un terminal UART.
- S'alimente avec le RTU SAFIR ou en Standalone avec une alimentation 5VDC
- Des ports analogiques et I/O pour des capteurs externes.
- Un microcontrôleur embarqué avec une mémoire Flash et un Processeur ARM.
- Un PCB qui répond au mieux aux fluctuations électromagnétiques.

3.2 Programmation

3.2.1 Configuration du circuit CC1120

a Le Logiciel SmartRF Studio V7

Smart RF studio [6] est un logiciel qui a été conçu par Texas Instruments permettant de simplifier le développement d'une application RF.

Le logiciel permet d'évaluer les circuits LPRF (Low Power Radio Frequency) et génère leurs registres de configuration.

Le programme peut aussi tester les performances RF et donne la main à l'utilisateur de les ajuster pour avoir les optimisations désirées. La figure qui suit montre l'interface graphique du logiciel SmartRF Studio V7.

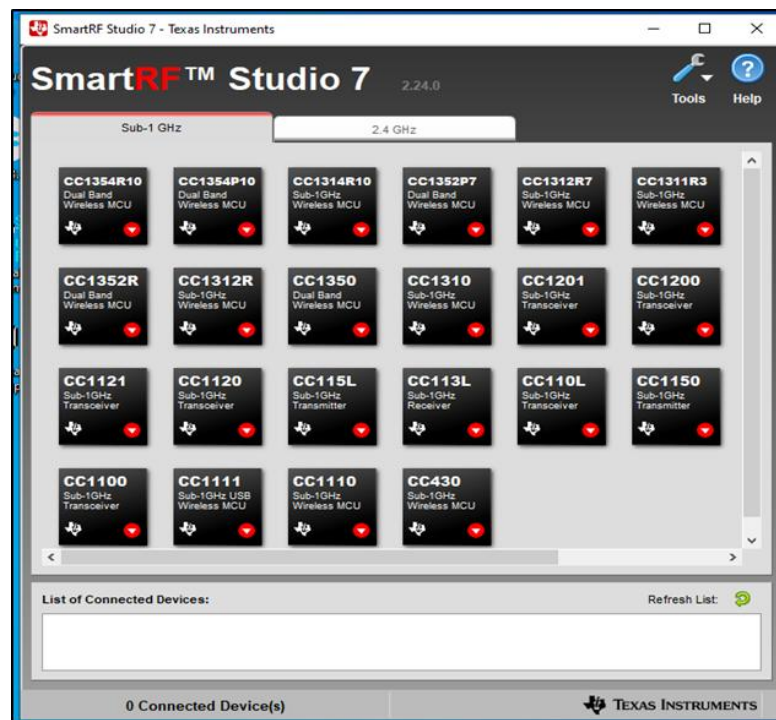


Figure 3.1 Interface graphique du logiciel SmartRF Studio7

Le logiciel peut aussi accomplir d'autres fonctions comme :

- La liaison des émetteurs et récepteurs entre eux grâce à un simple clic.

- Donne à l'utilisateur l'accès aux registres internes du circuit RF CC1xxx et CC2xxx.
- Exécute directement les commandes STROBE depuis l'interface graphique.
- Permet l'exportation et l'importation des fichiers config sous une multitude de formats.
- Permet à l'utilisateur de définir sa propre cartographie de GPIO's.
- Sert comme interface de communication et de gestion de cartes d'essai fournies par Texas Instruments.

a.1 La configuration avec Smart Studio

On a utilisé le logiciel de développement Smart RF Studio proposé par le constructeur, afin de simplifier la configuration des registres principaux, et diminuer le temps de développement

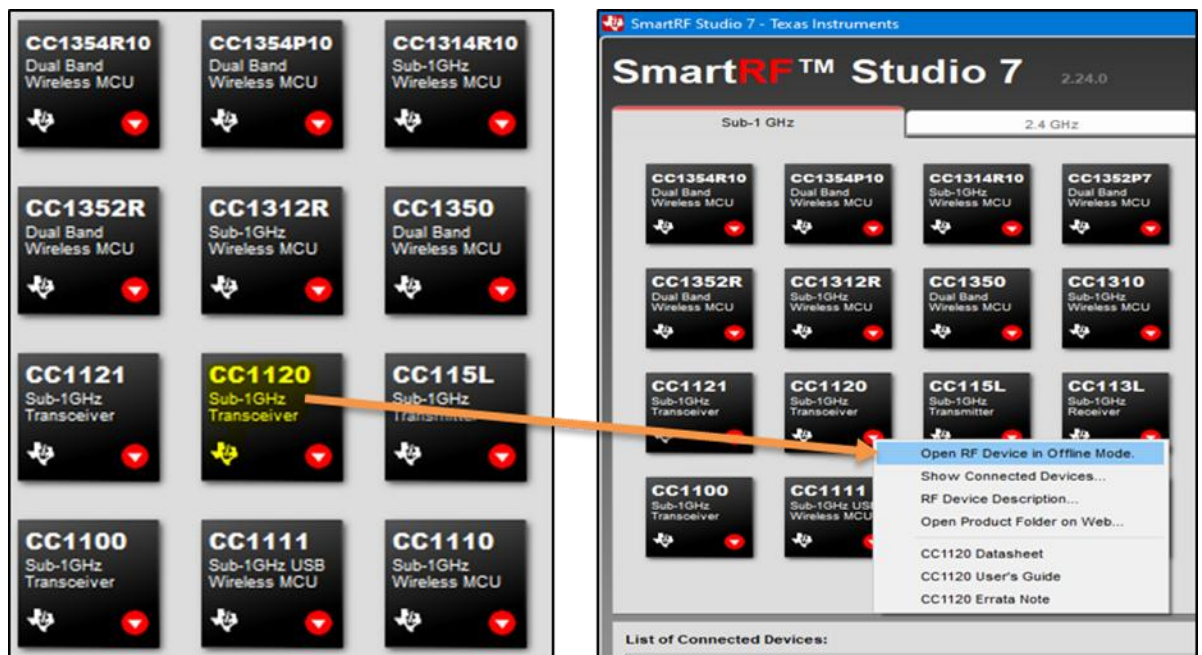


Figure 3.2 Première étape dans la configuration du CC1120

a.2 Panneau de contrôle principal

Le panneau de contrôle principal est la partie qu'on a utilisé pour définir tous les réglages importants qui ont une relation avec la fréquence, les filtres de largeur de bande RX, et la fréquence de l'oscillateur XTAL de 32MHz,

Le panneau de contrôle permet aussi à l'utilisateur de modifier la puissance de transmission, et la modulation, la **figure 3.3** montre la fenêtre du panneau de contrôle principal.

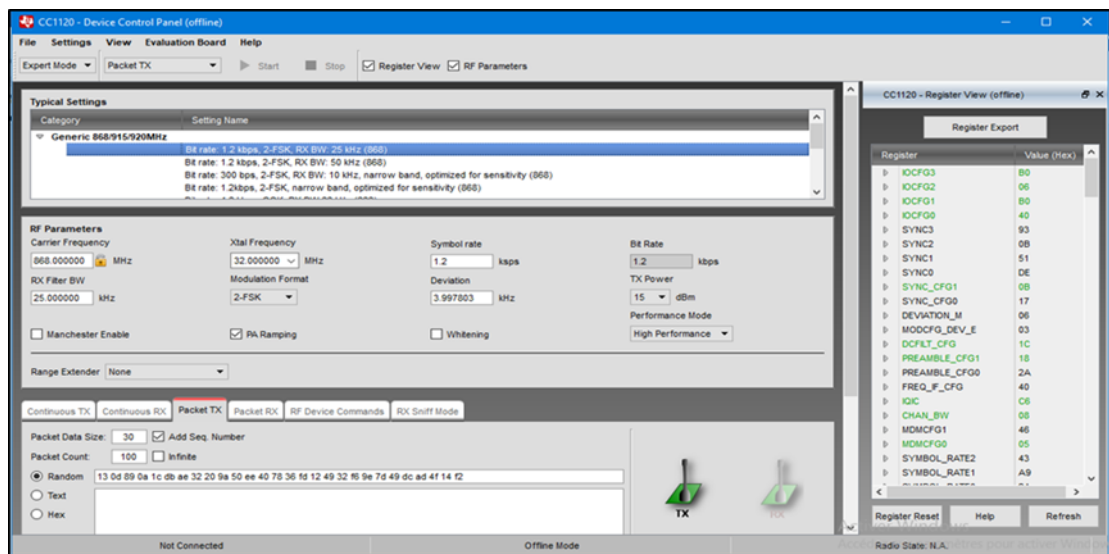


Figure 3.3 Panneau de contrôle principal

Le panneau de contrôle principal contient les paramètres standards ainsi que des profils préconfigurés de chez le constructeur.

Il nous a permis de faire la configuration totale et visualiser les registres internes du CC1120 avant de les exporter.

a.3 Réglages typiques

Les réglages typiques permettent de charger un profil RF préconfiguré par le constructeur.

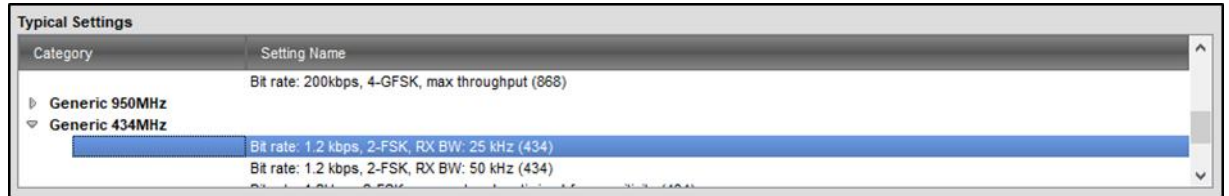


Figure 3.4 réglages typiques

C'est dans cet onglet, qu'on a fixé la fréquence RF du CC1120 à 434 MHz, le bit rate à 1.2 kbps, la modulation à 2-FSK et le RX BW à 25KHz.

a.4 Les paramètres RF

Cet onglet offre une assistance avancée à un utilisateur expérimenté afin qu'il puisse apporter des modifications supplémentaires à l'application RF

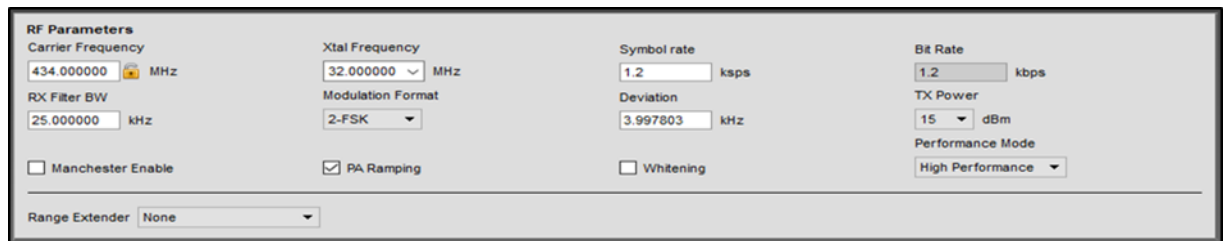


Figure 3.5 boîte de dialogue paramètres

Dans notre projet nous avons voulu garder la configuration suivante pour les deux modes de communication RF (RX et TX).

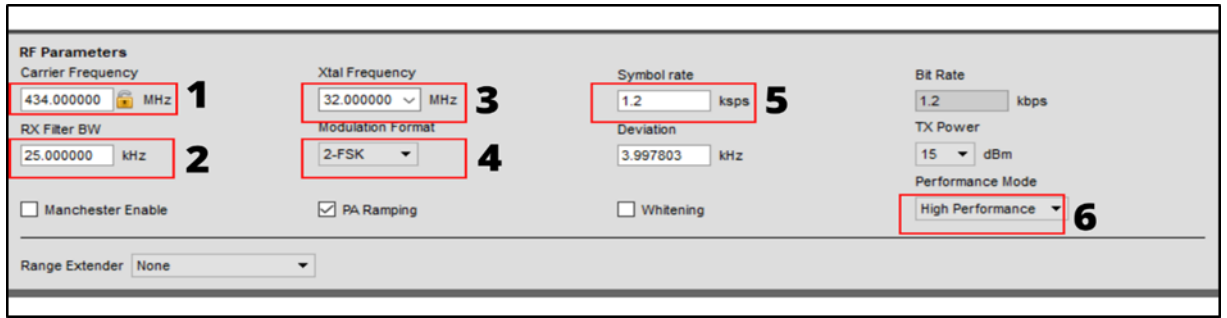


Figure 3.6 Paramètres choisis

a.5 Les modes du CC1120

Le CC1120 est configuré pour qu'il prenne en charge un des deux modes récepteur RX ou émetteur TX.

a.6 L'utilisation du profil Packet RX

Afin de recevoir des trames d'un CC1120 émetteur, on active le mode Packet RX, le Smart Studio génère la configuration requise par le mode récepteur. La **figure 3.7** montre le mode Packet RX qui met le CC1120 en mode récepteur de données.

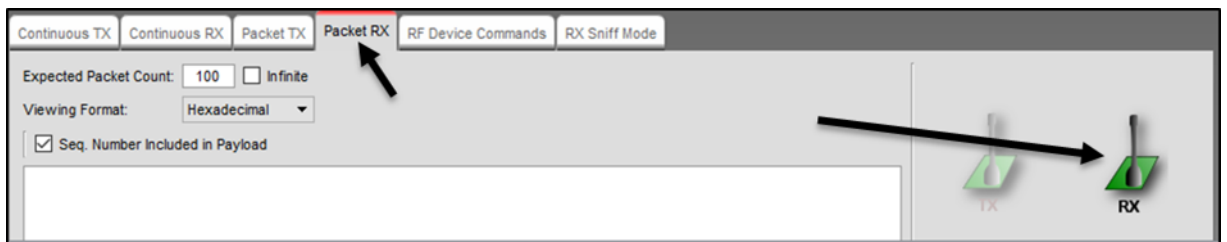


Figure 3.7 Profile Packet RX

a.7 L'utilisation du profil Packet TX

Pour l'émetteur, on a configuré le CC1120 en mode Packet TX, ce qui est le bon choix lors de l'envoi d'une trame.

Pour rentrer dans ce mode, on clique sur l'onglet Packet TX, Smart RF studio configure lui-même les registres responsables à mettre le CC1120 en mode émetteur

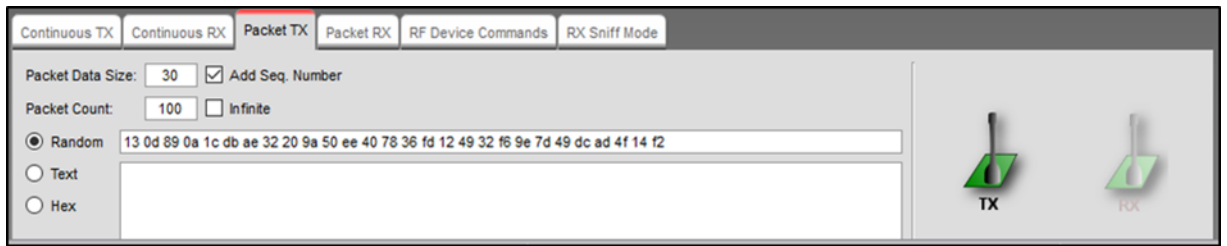


Figure 3.8 Profile Packet TX

a.8 L'export des registres

Après avoir fini de configurer chaque mode de transmission et son type, soit (packet RX) ou (Packet TX), le logiciel va nous permettre d'exporter la configuration qui se trouve dans l'onglet des registres montré dans la **figure 3.9**

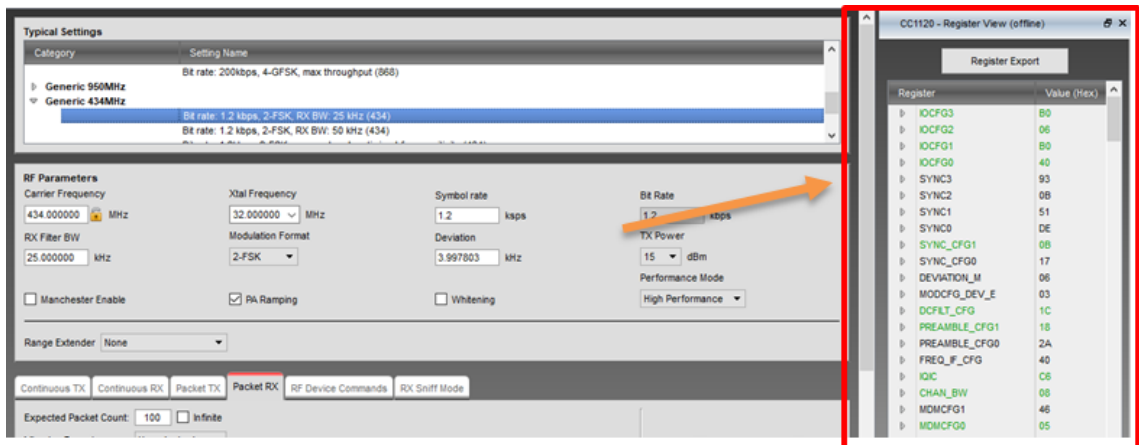


Figure 3.9 Volet des registres exportés

La valeur assignée aux registres est sous un format d'octets qui représentent des bits de configuration nécessaires au fonctionnement du CC1120.

Pour exporter les registres générés par le logiciel, on clique sur le bouton Register Export comme le montre la **figure 3.10** :

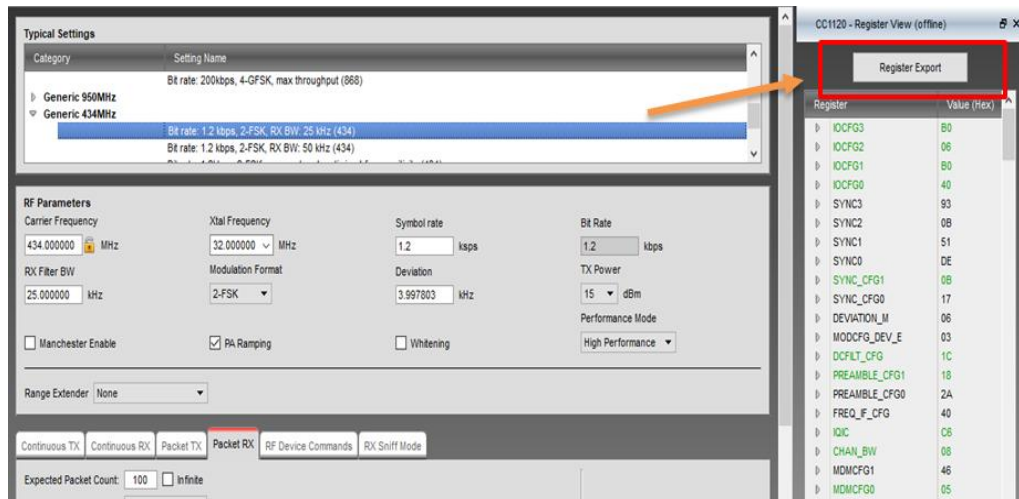


Figure 3.10 Bouton export des registres

Après la génération des registres de configuration et leurs valeurs associés, on a copié le résultat et on l'a mis dans un fichier d'en tête et d'extension header pour développer notre bibliothèque CC1120.

3.2.2 Développement de la bibliothèque sous l'environnement CubeIDE

a Création et configuration d'un projet

La **figure 3.11** montre l'interface de programme :

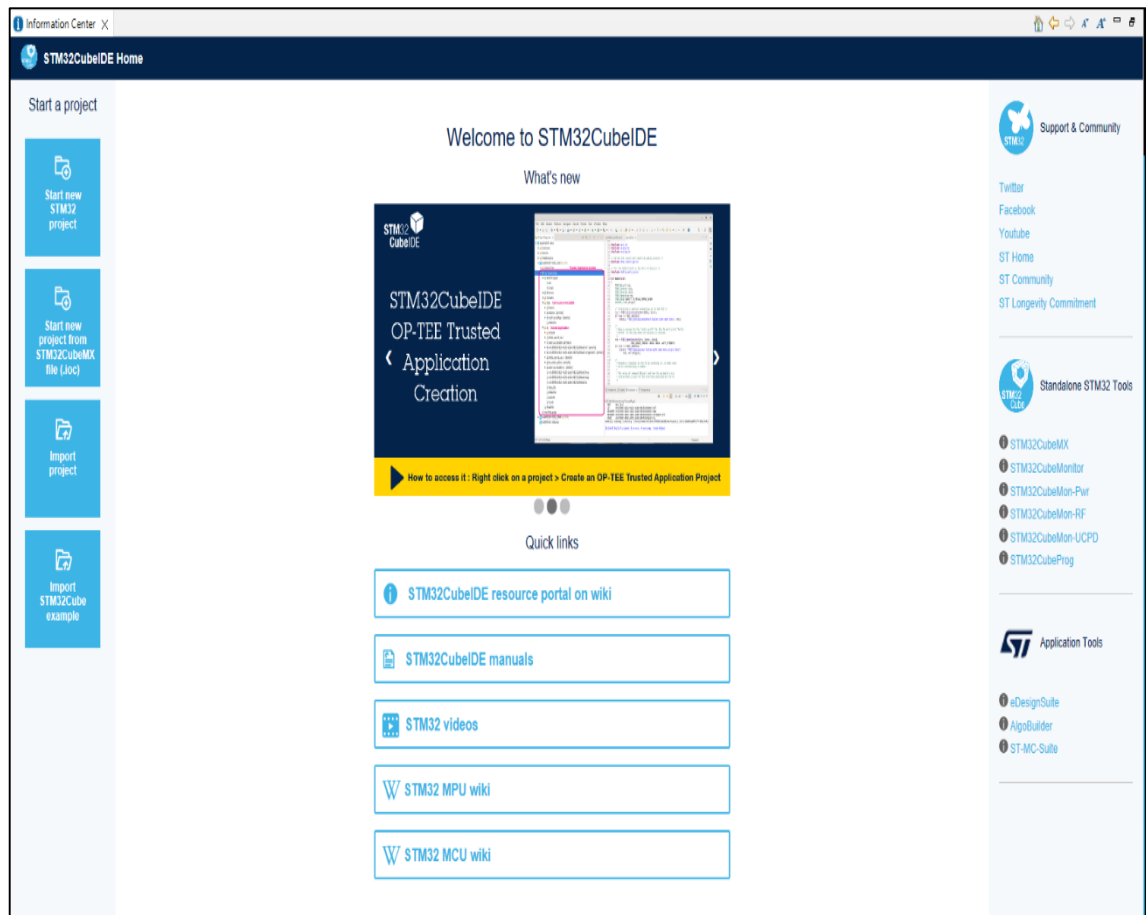


Figure 3.11 Environnement de développement Intégré CubeIDE

Pour nos tests, on a travaillé sur le microcontrôleur STM32L476RG, le projet sur CubeIDE sera autour de ce dernier

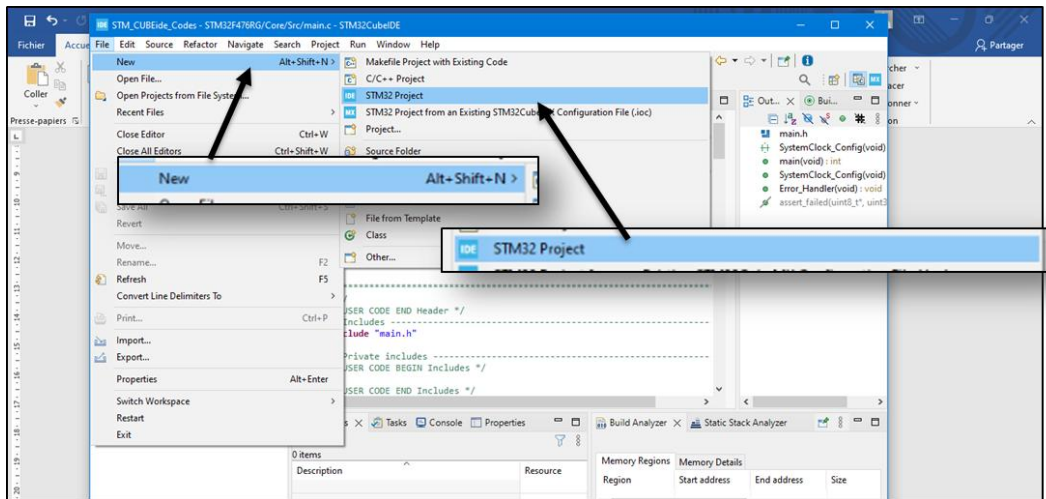


Figure 3.12 Création d'un nouveau projet

Après la création du projet, on sélectionne le microcontrôleur désiré, dans nos tests on a utilisé le STM32L476RG.

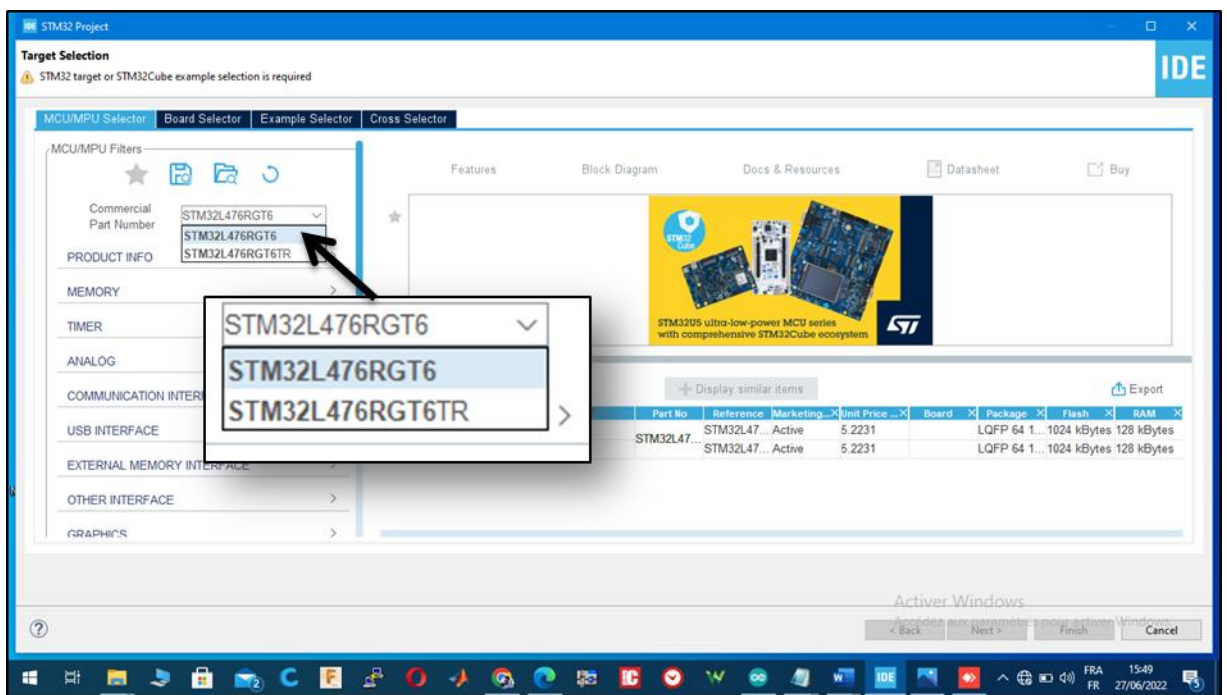


Figure 3.13 Choix du microcontrôleur

Le logiciel CubeMX intégré dans l'environnement de CubeIDE nous donne accès à la configuration graphique des broches de notre circuit comme ainsi.

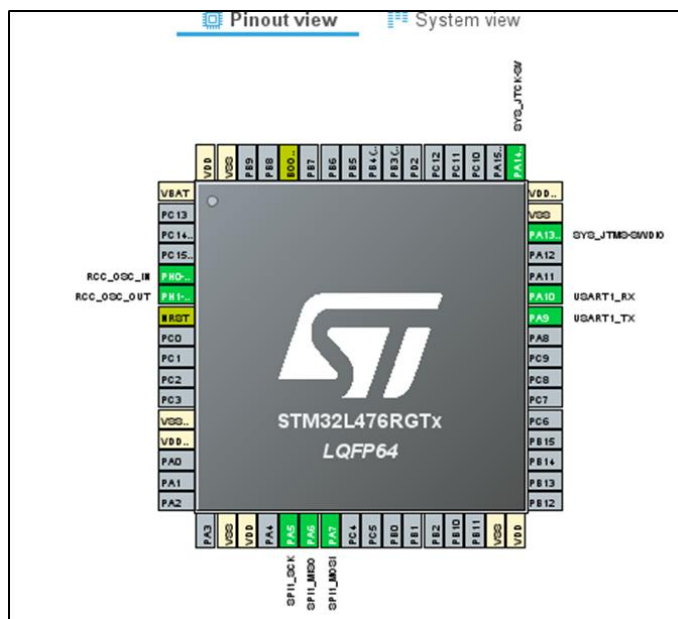


Figure 3.14 CubeMX

Les broches en couleur verte correspondent aux lignes des interfaces du STM32 utilisées dans notre application.

a.1 Configuration des lignes du debugger

Avant de commencer chaque application en Standalone, on doit impérativement sélectionner les lignes du debugger qui sont SWDIO (Serial Wire Data I/O) et SWCK (Serial Wire Clock) afin que le ST-Link puisse traiter les données fournies par les registres du STM32. La procédure de configuration est montrée dans la **figure 3.15**.

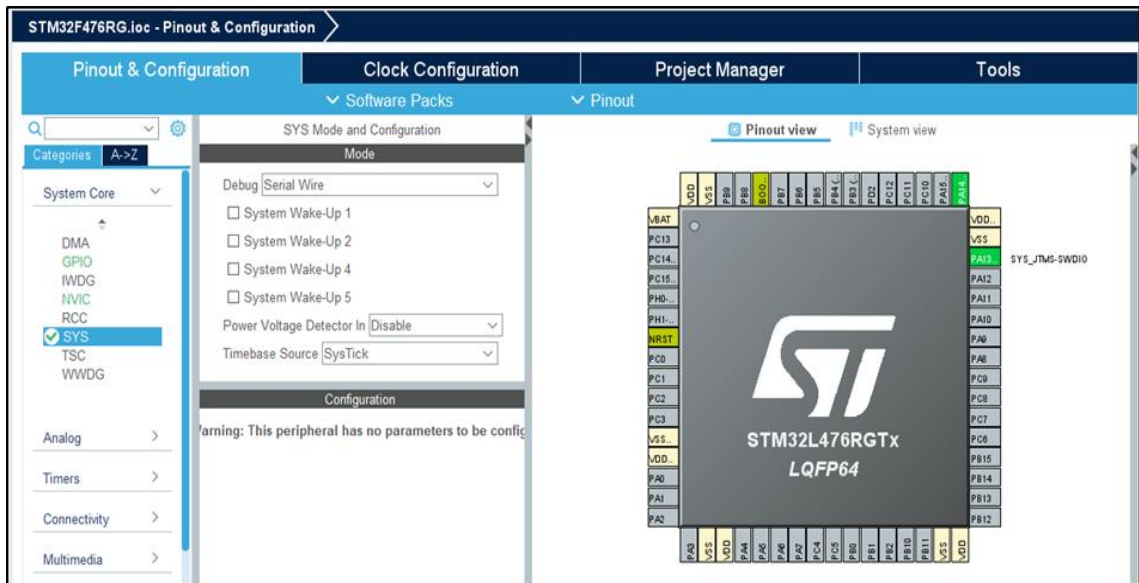


Figure 3.15 Configuration des lignes du debugger

a.2 Configuration du système d'horloge

Le STM32 fonctionne avec un Crystal résonnant à une fréquence de 8MHz monté sur les broches PH0 et PH1,

Après la configuration, ces pins permettent au microcontrôleur de se brancher sur l'horloge externe. La figure 3.16 qui suit montre la configuration du Crystal d'horloge :

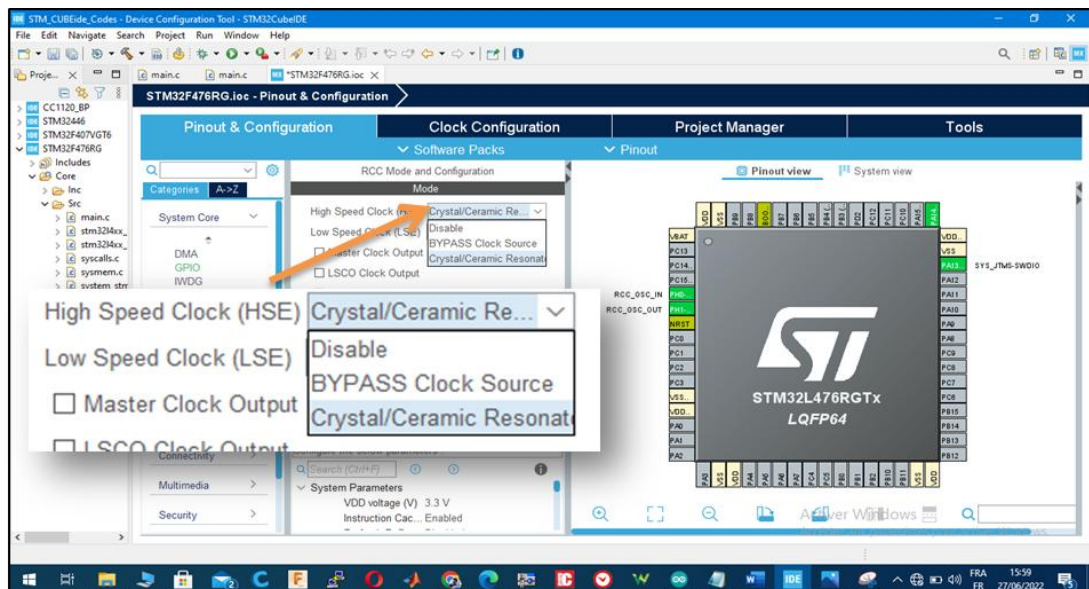


Figure 3.16 Configuration du Crystal

Le STM32 contient à l'intérieur une circuiterie entière de diviseurs et de synthétiseurs de fréquences, afin que chaque interface puisse fonctionner avec une temporisation bien précise.

a.3 Réglage du cadencement du bus APB1

Etant donné que nous allons utiliser l'interface SPI1 et que celle-ci reçoit sa temporisation du BUS APB1, on a donc fixé la fréquence de ce dernier sur 4MHz qui est largement suffisante pour le transfert des données vers le CC1120 :

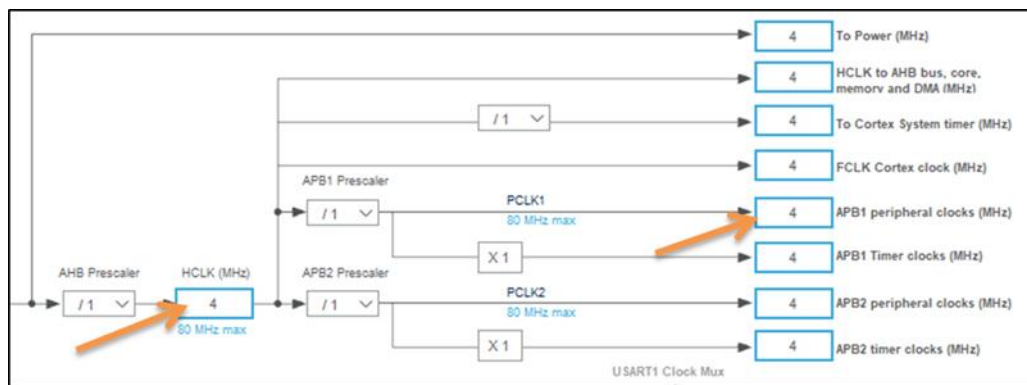


Figure 3.17 Cadencement du bus APB

a.4 Configuration SPI

Toujours dans le volet connectivité, on sélectionne SPI1 comme le montre la figure 3.18.

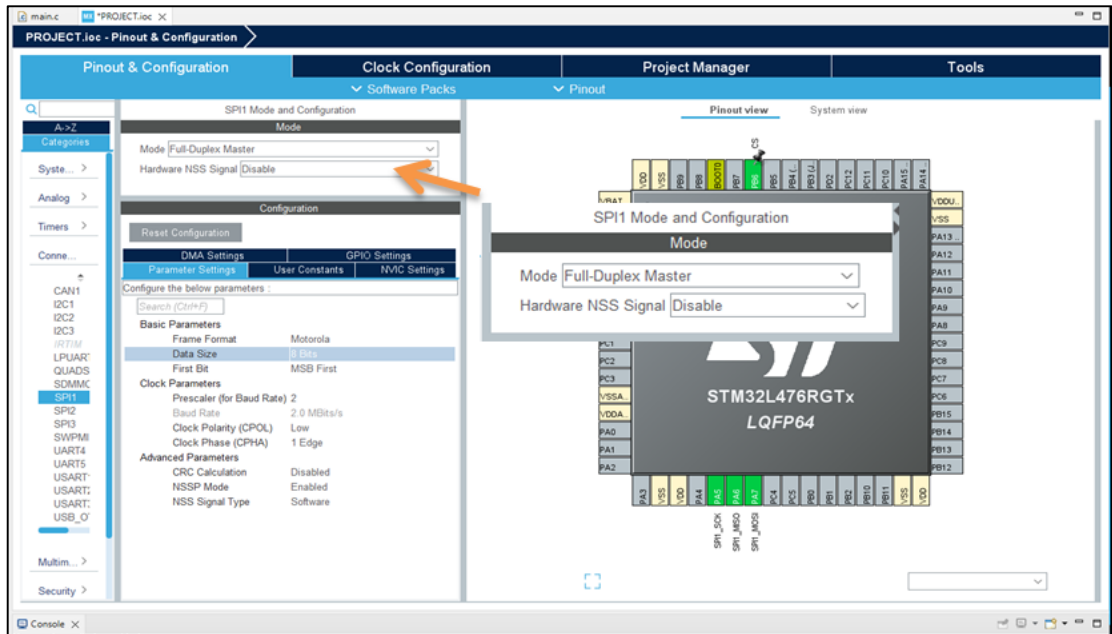


Figure 3.18 Configuration du SPI1

Le CubeMX définit les connexions du BUS SPI1 sur les broches (PA5, PA6, PA7, et pour CS la broche PB6).

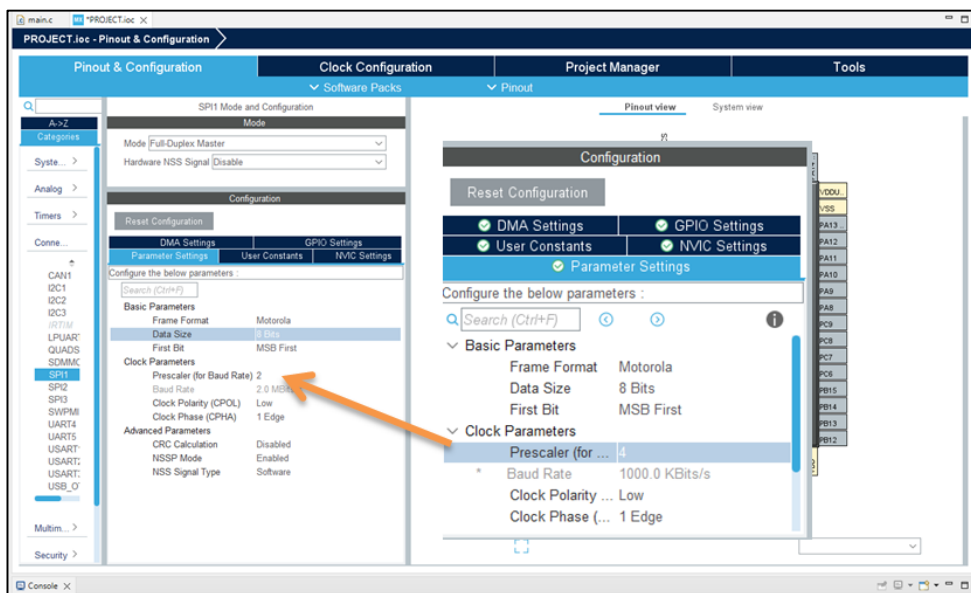


Figure 3.19 Adaptation du Pré Scalaire

Les paramètres que nous avons modifiés sont la taille des données à 8-bits, et le pré scalaire du SPI à 4, comme on l'a déjà dit nous utilisons un BUS APB1 d'une horloge cadencée à 4MHz. Donc pour avoir une vitesse de 1Mbps on divise par 4 l'horloge du BUS APB1.

a.5 Configuration UART

Nous allons voir maintenant l'étape de la configuration UART

Comme on l'a déjà mentionné notre module reçoit des données de la pile du CC1120 via SPI et les envoient au SAFIR ou à un autre microcontrôleur via UART

D'où la configuration de l'UART1 est impérative pour permettre à notre module de s'interfacer aisément avec n'importe quel type de systèmes embarqués.

La **figure 3.20** montre notre les paramètres UART1 pris en charge :

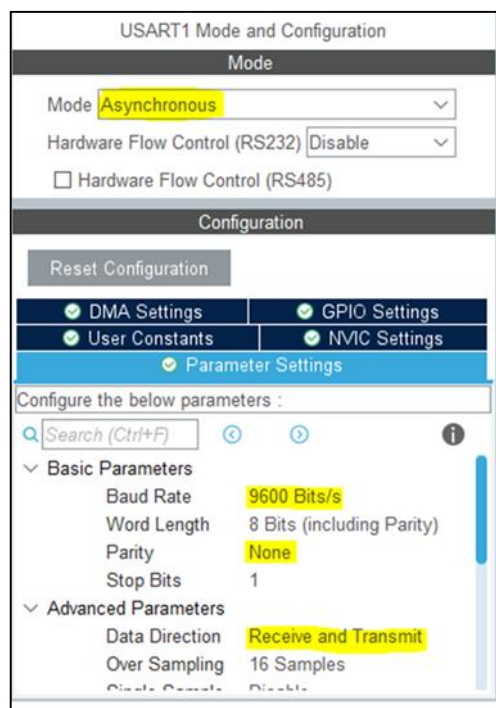


Figure 3.20 Implémentation UART1

On a choisi le mode UART asynchrone, un Baud Rate de 9600 Bit/s, pas de bit de parité et une direction de données RX/TX.

a.6 Configuration des interruptions

Et pour finir la configuration, on a finalisé la procédure avec le paramétrage des broches GPIO, on les a configurés comme interruptions externes EXTI (External Interrupt) dans le cas où le CC1120 génère une interruption. Les paramètres sont les suivants.

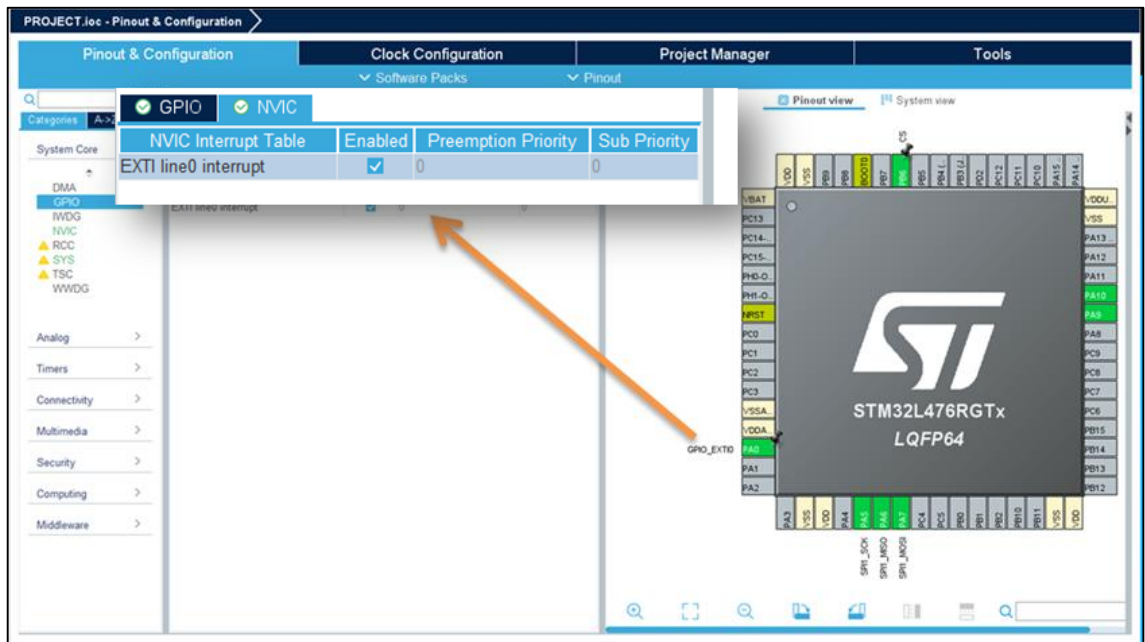


Figure 3.21 Configuration des interruptions externes EXTI

a.7 Configuration GPIO CS (Chip Select Pin)

Comme décrit précédemment, nous avons dit que le BUS SPI comporte 4 fils, (MISO, MOSI, SCK, CS)

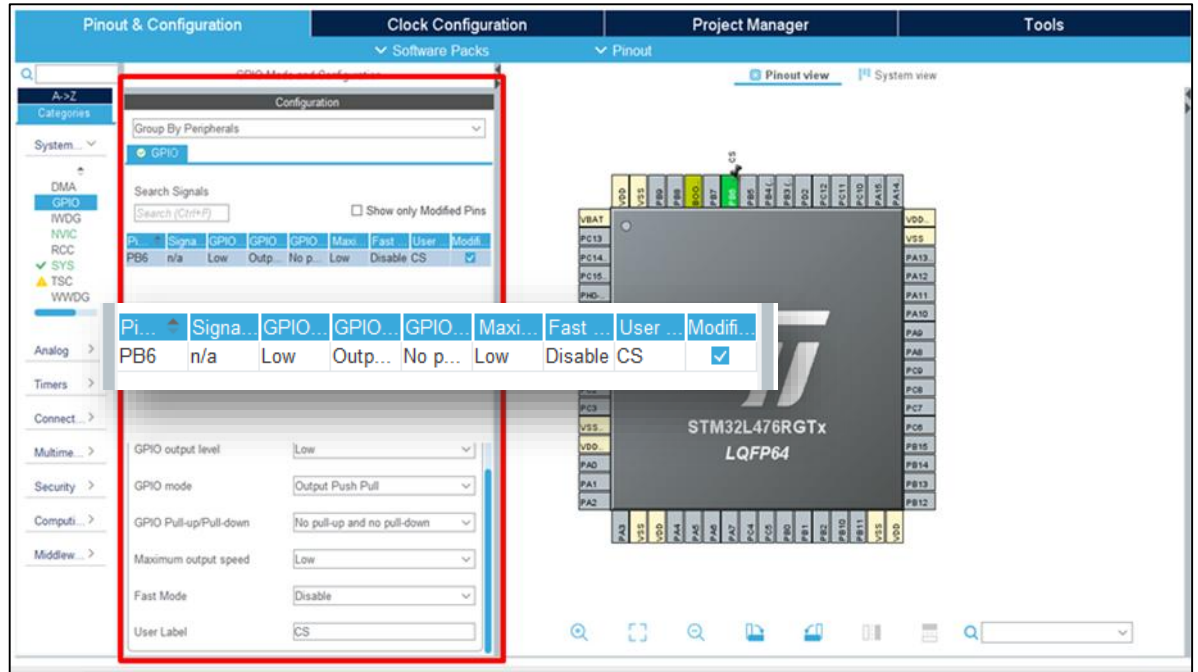


Figure 3.22 Configuration de la broche CS

Nous pouvons désormais grâce au CubeMX assigner la broche CS à n'importe quel GPIO en utilisant le protocole Motorola. C'est pour cela qu'on a choisi de la mettre sur la broche PB6, en la déclarant en tant que sortie GPIO de type Push-Pull (General Purpose I/O).

b Les fonctions créées

L'idée était de développer une bibliothèque pour le CC1120 qui fonctionne sur le logiciel CubeIDE.

Dans les parties suivantes on montre notre parcours franchi pour aboutir à des fonctions de base qui peuvent configurer le CC1120.

b.1 La bibliothèque HAL_SPI

La bibliothèque HAL (Hardware Abstraction Layer) est un outil essentiel dans le développement software de nos fonctions, elle contient une pré-configuration nécessaire à l'implémentation de l'interface SPI.

- **HAL_SPI_Transmit**

Cette fonction comporte le pointeur de données, l'octet de la donnée, la taille de la donnée et le timeout comme paramètres et elle retourne un octet Statut qui confirme que la transmission a été effectuée avec succès.

```
HAL_SPI_Transmit(SPI_HandleTypeDef *hspi, uint8_t *pData, uint16_t Size, uint32_t Timeout)
```

- **HAL_SPI_Receive**

Cette fonction comporte le pointeur de données, la quantité de données à recevoir, et le Timeout comme paramètres, et elle retourne un octet statut comme accusé de réception.

```
HAL_SPI_Receive(SPI_HandleTypeDef *hspi, uint8_t *pData, uint16_t Size, uint32_t Timeout)
```

- **HAL_SPI_TransmitReceive**

Cette fonction combine les deux fonctions précédemment décrites, elle comporte les paramètres des deux et permet au bus SPI d'envoyer et recevoir en même temps.

```
HAL_SPI_TransmitReceive(SPI_HandleTypeDef *hspi, uint8_t *pTxData, uint8_t *pRxData, uint16_t Size, uint32_t Timeout)
```

Nous allons donc discuter les fonctions que nous avons créés pour que le STM32 puisse communiquer avec le CC1120 et le configurer pour qu'il puisse opérer dans les deux modes (RX / TX).

- **cc112xRead**

```
uint8_t cc112xRead(uint8_t address){  
  
    uint8_t data ;  
    uint8_t byte = address | 0x80 ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, RESET); //Pulling down CS pin  
  
    HAL_SPI_Transmit(&hspi1, &byte, 1, 10) ;  
    HAL_SPI_Receive(&hspi1, &data, 1, 10);  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, SET); //Pulling up CS pin  
  
    return data;  
}
```

Figure 3.23 Fonction cc112xRead

La fonction, retourne un octet de registre principal tel le registre STATUS, ou bien d'autres registres importants qui rentrent dans la configuration Radio

- **cc112xWrite**

Comme son nom l'indique, la fonction cc112xWrite permet au STM32 d'écrire dans un registre indexé par une adresse, c'est pour cela qu'elle comporte deux champs (adress et data).

```
uint8_t cc112xWrite(uint8_t address, uint8_t data){  
  
    uint8_t status ;  
    uint8_t bytes[2] ;  
  
    bytes[0] = address ;  
    bytes[1] = data ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, RESET); //Pulling down CS pin  
  
    HAL_SPI_Transmit(&hspi1, &bytes[0], 1, 10) ;  
    HAL_SPI_TransmitReceive(&hspi1, &bytes[1], &status, 1, 10) ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, SET); //Pulling up CS pin  
  
    return status;  
}
```

Figure 3.24: Fonction cc112xWrite

- **cc112xStrobe :**

Cette fonction permet d'envoyer l'adresse qui équivaut à une commande STROBE, les commandes STROBE sont utilisés pour mettre le CC1120 dans différents états.

Pour créer la fonction, nous nous sommes basés sur le tableau de l'annexe B des commandes STROBE et leurs actions.

La fonction `cc112xStrobe` est montrée en détail dans la **figure 3.25** :

```
uint8_t cc112xStrobe(uint8_t strb){  
  
    uint8_t status ;  
    uint8_t strob = strb ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, RESET); //Pulling down CS pin  
  
    HAL_SPI_TransmitReceive(&hspi1, &strob, &status, 1, 100) ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, SET); //Pulling up CS pin  
  
    return status ;  
}
```

Figure 3.25: Fonction cc112xStrobe

- **cc112xExtRead**

La fonction `cc112xExtRead` lit une adresse du registre Extended (`address | ext`) et retourne l'information qui se trouve dans l'emplacement indexé par l'adresse. La figure 3.26 Fonction `cc112xExtRead` :

```
uint8_t cc112xExtRead(uint8_t address){  
  
    uint8_t status ;  
    uint8_t ext = 0x2F | 0x80 ;  
    uint8_t addr = address | ext ;  
    uint8_t data ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, RESET); //Pulling down CS pin  
  
    HAL_SPI_Transmit(&hspi1, &ext, 1, 10) ;  
    HAL_SPI_Transmit(&hspi1, &addr, 1, 10) ;  
    HAL_SPI_Receive(&hspi1, &data, 1, 10);  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, SET); //Pulling up CS pin  
  
    return data ;  
}
```

Figure 3.26 Fonction cc112xExtRead

- **cc112xExtWrite**

La fonction `cc112xExtWrite` permet d'écrire dans une adresse de l'espace registres Extended du CC1120. Les registres Extended, contiennent des informations supplémentaires tels que le numéro de Série 0x48 et la version 0x23 du CC1120.

```
uint8_t cc112xExtWrite(uint8_t address, uint8_t data){  
  
    uint8_t status ;  
    uint8_t bytes[2] ;  
    uint8_t ext = 0x2F ;  
    bytes[0] = address | ext ;  
    bytes[1] = data ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, RESET); //Pulling down CS pin  
  
    HAL_SPI_TransmitReceive(&hspi1, &ext, &status, 1, 10) ;  
    HAL_SPI_Transmit(&hspi1, &bytes[0], 1, 10) ;  
    HAL_SPI_Transmit(&hspi1, &bytes[1], 1, 10) ;  
  
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, SET); //Pulling up CS pin  
  
    return status ;  
}
```

Figure 3.27 Fonction cc112xExtWrite

3.2.3 Choix de l'Arduino

Le logiciel open source Arduino IDE facilite l'écriture de code et son téléchargement sur la carte. Ce logiciel peut être utilisé avec n'importe quelle carte

Pour commencer nos tests sur l'ARDUINO IDE, on devait ajouter la famille des microcontrôleurs STM32, nous avons dû appliquer les étapes suivantes :

- Ouvrir le volet des performances
- Coller le lien des cartes STM32 dans l'espace URL de gestionnaire de cartes supplémentaires.
- Télécharger les cartes depuis le gestionnaire des cartes.
- Redémarrer le logiciel pour que tout s'installe correctement.

La bibliothèque SPI est incluse dans chaque cœur/plate-forme Arduino, nous n'avons donc pas besoin de l'installer en externe, nous avons utilisé ses fonctions dans la conception de notre code CC1120.

a Arduino IDE

Le logiciel open source Arduino IDE facilite l'écriture de code et son téléchargement sur la carte. Ce logiciel peut être utilisé avec n'importe quelle carte

Pour commencer nos tests sur l'ARDUINO IDE, on devait ajouter la famille des microcontrôleurs STM32, nous avons dû appliquer les étapes suivantes :

- Ouvrir le volet des performances
- Coller le lien des cartes STM32 dans l'espace URL de gestionnaire de cartes supplémentaires.
- Télécharger les cartes depuis le gestionnaire des cartes.
- Redémarrer le logiciel pour que tout s'installe correctement.

La bibliothèque SPI est incluse dans chaque cœur/plate-forme Arduino, nous n'avons donc pas besoin de l'installer en externe, nous avons utilisé ses fonctions dans la conception de notre code CC1120.

b La bibliothèque HydroSense

Nous avons dû modifier le fichier d'origine pour avoir un mode **PACKET TX** à une fréquence de 433MHz, une modulation 2-FSK, une puissance de transmission max 15 dbm, un mode hautes performances, et un filtre de bande étroite réglé à 25KHz.

Pour cela nous avons copié la configuration déjà générée du logiciel SmartRF Studio et on l'a mise dans le fichier d'en-tête rfSettings.h.

3.2.4 Implémentation du code de transmission

Nous avons dû modifier le fichier d'origine pour avoir un mode **PACKET TX** à une fréquence de 433MHz, une modulation 2-FSK, une puissance de transmission max 15 dbm, un mode hautes performances, et un filtre de bande étroite réglé à 25KHz.

Pour cela nous avons copié la configuration déjà générée du logiciel SmartRF Studio et on l'a mise dans le fichier d'en-tête rfSettings.h.

a Implémentation et Organigramme du code de transmission

Au début du fonctionnement Le CC1120 sort du mode Idle, à ce moment-là le synthétiseur de fréquences se déclenche et se calibre automatiquement, si la prochaine commande envoyée est STX. Sinon il doit être calibré manuellement en envoyant la commande SCAL.

Dès que le synthétiseur de fréquences est calibré et opérationnel, une commande strobe STX est envoyée afin que le CC1120 se met en mode émetteur.

L'utilisateur rentre une chaîne de données (chaînes de caractères, valeurs entières) dans la pile TX FIFO, l'émission est donc déclenchée avec une commande STX.

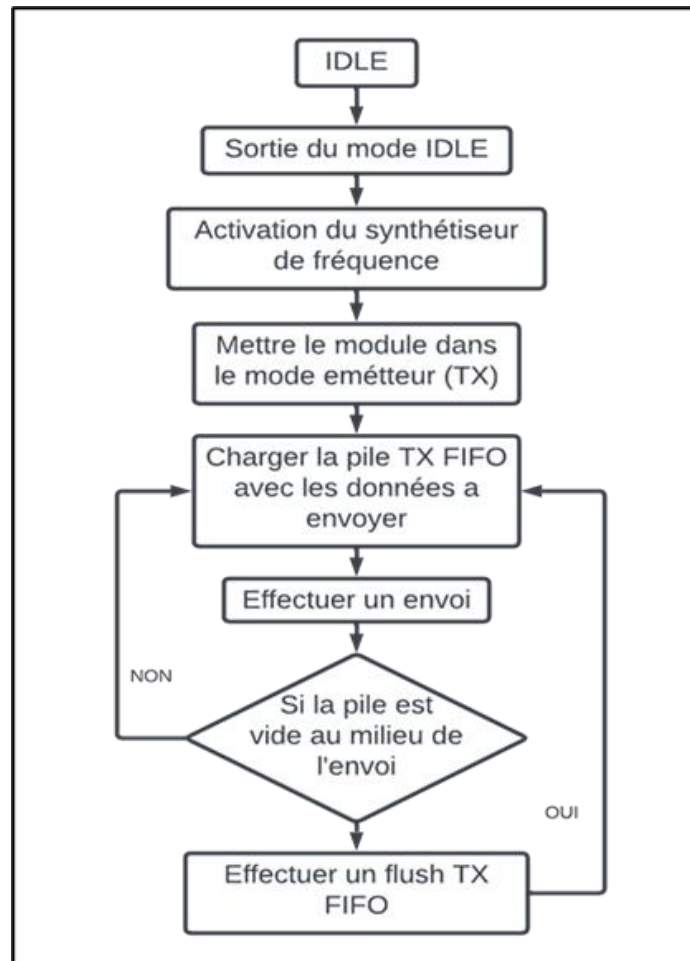


Figure 3.28 Organigramme du code TX

La chaîne de données peut être envoyée par un utilisateur ou bien par un RTU SAFIR hôte.

Si la pile TX FIFO est vide au milieu de l'émission, une erreur est donc générée par le timer de réception, alors un Flush FIFO doit être établi en envoyant la commande SFTX ou (Strobe Flush TX) pour que le CC1120 puisse quitter l'erreur et effectuer un nouvel envoi, après le nouveau calibrage du synthétiseur de fréquences.

Si on ne rencontre pas d'erreurs pendant l'envoi des données, la transmission continue et le CC1120 se prépare pour l'envoi d'un nouveau paquet.

b Tests RSSI avec le logiciel S2LP

Dans cette partie, nous allons réaliser un test de puissance du signal émis, ou (RSSI), ce test va permettre de confirmer que le CC1120 génère réellement un signal RF.

Après avoir connecté la carte, on sélectionne le port COM correspondant, nous avons copié les mêmes paramètres de la partie **paramètres RF** depuis SmartRF Studio vers S2-LP puis appuyé sur **Configure Radio** comme le montre la **figure 3.29** :

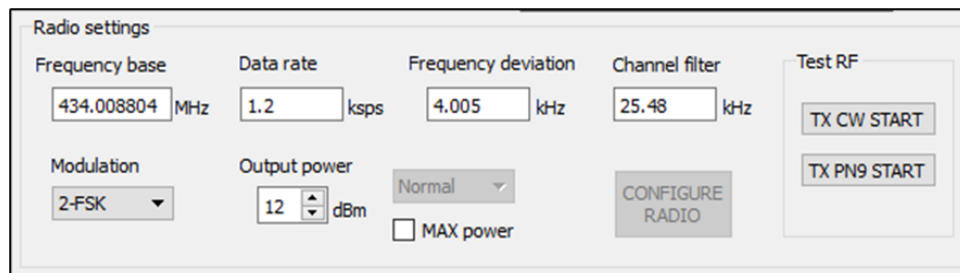


Figure 3.29 Paramètres RF

La **figure 3.30** montre notre test RSSI réalisé avec S2-LP, on peut clairement voir le graphe en jaune qui montre la présence d'un signal RF d'une puissance allant jusqu'à -20 dBm (décibels milliWatt).

Le bouton STOP permet à l'utilisateur d'interrompre le test à n'importe quel moment voulu.

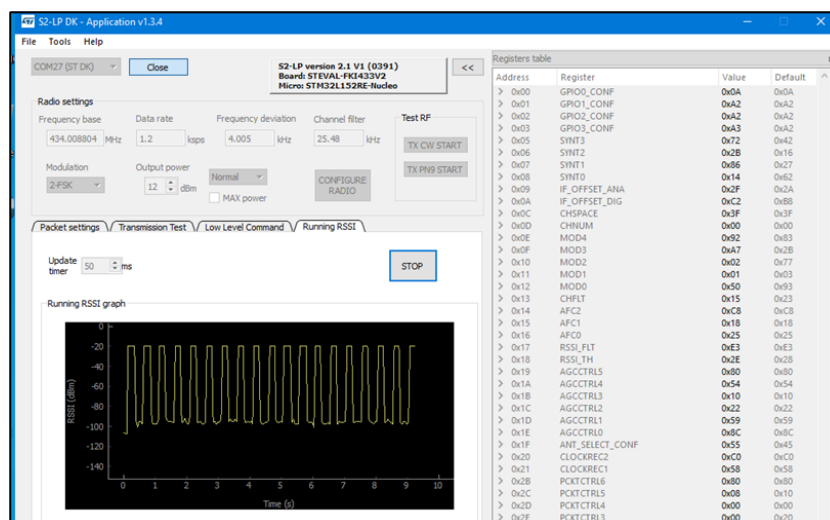


Figure 3.30 Test de puissance du signal

c Test de la transmission



Le test RSSI (Received Signal Strength Indicator) a prouvé que le CC1120 envoie bien des données RF. Pour les visualiser on effectue maintenant un test RX avec le logiciel S2-LP.

Dans l'onglet **Packet settings**, on modifie le format de la trame qui va être reçue par le S2-LP de manière à ce qu'elle soit compatible avec celle que le CC1120 émettra.

La trame du CC1120 comporte deux octets **PREAMBLE**, quatre octets **SYNC WORD**, un octet **LENGTH**, et un octet **PAYLOAD** comme le décrit la **figure 3.31**

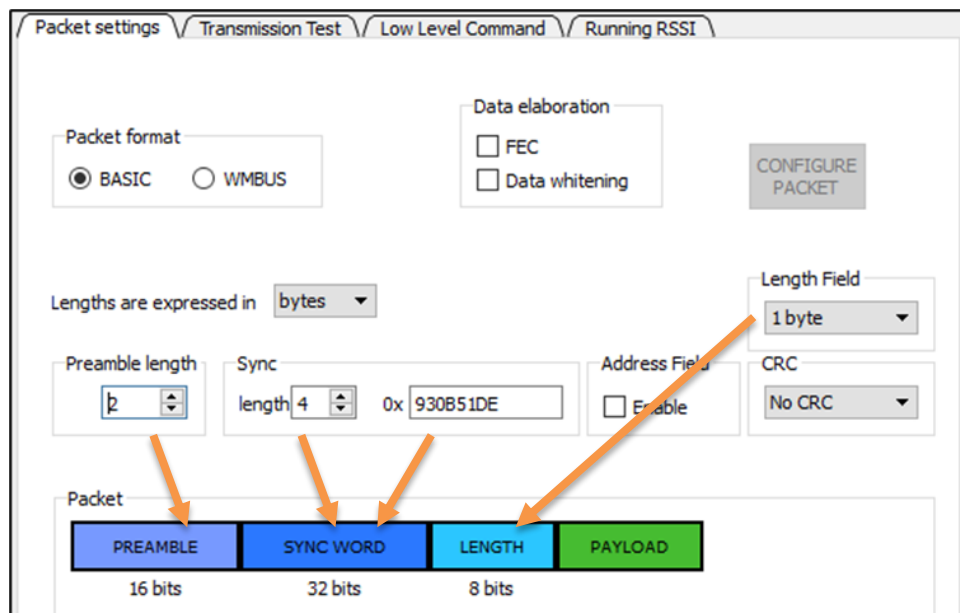


Figure 3.31 Paramètres du Packet CC1120

La trame a été configurée en respectant le format donné par le constructeur dans la datasheet du CC1120.

Les quatre octets SYNC WORD portent les valeurs fournies par le logiciel SmartRF Studio.

▷	SYNC3	93
▷	SYNC2	0B
▷	SYNC1	51
▷	SYNC0	DE

Figure 3.32 Octets du SYNC WORD

On appui sur le bouton **CONFIGURE PACKET** après avoir apporté toutes les modifications. Puis on va sur l'onglet **TRANSMISSION TEST**, on clique sur le bouton **BEGIN TRANSMISSION** afin de lancer les tests.

Le CC1120 envoie le message 'BONJOUR', et on le visualise sur le S2-LP comme le montre la **figure 3.33**.

The screenshot shows the 'Transmission Test' interface. At the top, there are tabs for 'Packet settings', 'Transmission Test', 'Low Level Command', and 'Running RSSI'. The 'Transmission Test' tab is active. On the right, there are controls for 'Total packets' (10), 'Ref timer' (500 ms), and radio buttons for 'HEX' and 'ASCII' (selected). A 'STOP' button is also present. Below these are settings for 'RX timeout' (100 ms), 'RSSI threshold' (-130 dBm), and 'Payload length' (18 bytes). The 'Sniff mode' section includes 'WakeUp timer' (3 ms) and 'FastRx timeout' (1228 us). The 'Test indicator' section shows 'Packet received' (8), 'Packet discarded' (0), 'PER [%]' (0.0), and 'RSSI [dBm]' (-14.0). A 'Sense RSSI' button is also visible. At the bottom, there is a table titled 'Information about the test' with columns for 'Time stamp', 'Info', 'RSSI', and 'Data Received'. The 'Data Received' column is highlighted with a red box and contains the text 'BONJOUR' for each of the eight rows. At the very bottom, there is a 'Dump log to file' input field and a checked 'Automatic scrolling' checkbox.

Time stamp	Info	RSSI	Data Received
2 15:17:30.66	Packet received (7 bytes)	-14	BONJOUR
3 15:17:31.16	Packet received (7 bytes)	-14	BONJOUR
4 15:17:31.66	Packet received (7 bytes)	-14	BONJOUR
5 15:17:32.15	Packet received (7 bytes)	-14	BONJOUR
6 15:17:32.65	Packet received (7 bytes)	-14	BONJOUR
7 15:17:33.14	Packet received (7 bytes)	-14	BONJOUR
8 15:17:33.64	Packet received (7 bytes)	-14	BONJOUR

Figure 3.33 Test de transmission

3.2.5 Implémentation du code de réception

Comme le mode TX, on a gardé la fréquence, la modulation, la puissance de transmission, et le mode en hautes performances, on a juste changé la partie **PACKET TX** en **PACKET RX**, et on a copié la configuration déjà établie avec le logiciel SmartRF dans le fichier d'en-tête rfSettings.h.

a Création du programme RX

Pour réaliser le code Hydrosense RX, nous avons cherché dans la datasheet du CC1120 une manière pour qu'on puisse lire une donnée dès que le circuit reçoit un paquet valide.

On a donc découvert que le circuit peut générer une interruption qu'on peut exploiter sur l'une des broches GPIO. Cette broche est le GP2.

Quand le CC1120 reçoit une trame valide, c'est-à-dire qu'il n'y ait aucune perturbation pendant le chargement des octets reçus dans la pile, il génère un signal logique d'état bas '0' sur la broche GP2. Nous avons donc saisi cette occasion pour dresser l'organigramme suivant qui montre le déroulement du code en mode RX.

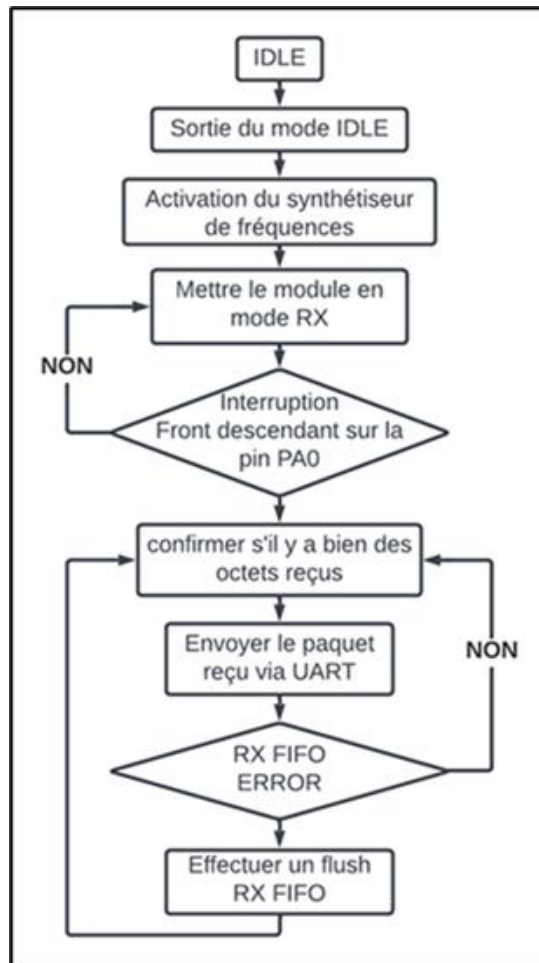


Figure 3.34 Organigramme du code RX

b Implémentation et organigramme du code de réception

La partie réceptrice diffère légèrement de la partie émettrice car elle marche avec des interruptions.

Pour faire fonctionner le mode RX du CC1120, on a suivi l'organigramme précédent

Au début du fonctionnement Le CC1120 sort du mode Idle, à ce moment-là le synthétiseur de fréquences se déclenche et se calibre automatiquement si la commande envoyée est SRX sinon il est calibré manuellement en envoyant SCAL.

Dès que le synthétiseur de fréquences est calibré et opérationnel, une commande strobe SRX peut être envoyée afin que le CC1120 se met en mode Récepteur.

Une interruption à front descendant est donc générée par le CC1120, ce qui signifie qu'un paquet valide a été reçu correctement.

Un paquet RF est valide dès que les octets qui le constituent à savoir le Préambule, La taille du paquet, l'adresse et le Payload sont tous présents dans la trame et sont aussi valides.

Le microcontrôleur se branche directement dans le programme de la subroutine et stocke les données dans une variable buffer, puis elles sont acheminées vers l'UART pour finir dans le RTU SAFIR comportant notre module.

Si la pile RX FIFO est pleine, une erreur est donc générée, à ce moment-là un Flush RX FIFO doit être établi en envoyant la commande SFRX ou (Strobe Flush RX) pour que le CC1120 puisse quitter l'erreur et attendre une nouvelle interruption, après le nouveau calibrage du synthétiseur de fréquences.

S'il n'y a aucune erreur pendant la réception des données l'opération continue et le CC1120 se prépare pour la réception d'un nouveau paquet.

c Tests de réception

Pour les tests de réception, nous avons gardé les mêmes paramètres RF et les mêmes paramètres de trame, on a donc seulement apporté des changements au mode de transmission du S2-LP en le mettant sur TX de façon à ce que l'émetteur de données soit la carte STEVAL433.

Pour la visualisation des messages reçus par le CC1120, on a utilisé le moniteur série intégré du ARDUINO IDE, la **figure 3.35** montre la réception du message sur le moniteur série 'STMicroelectronics' envoyé par le S2-LP :

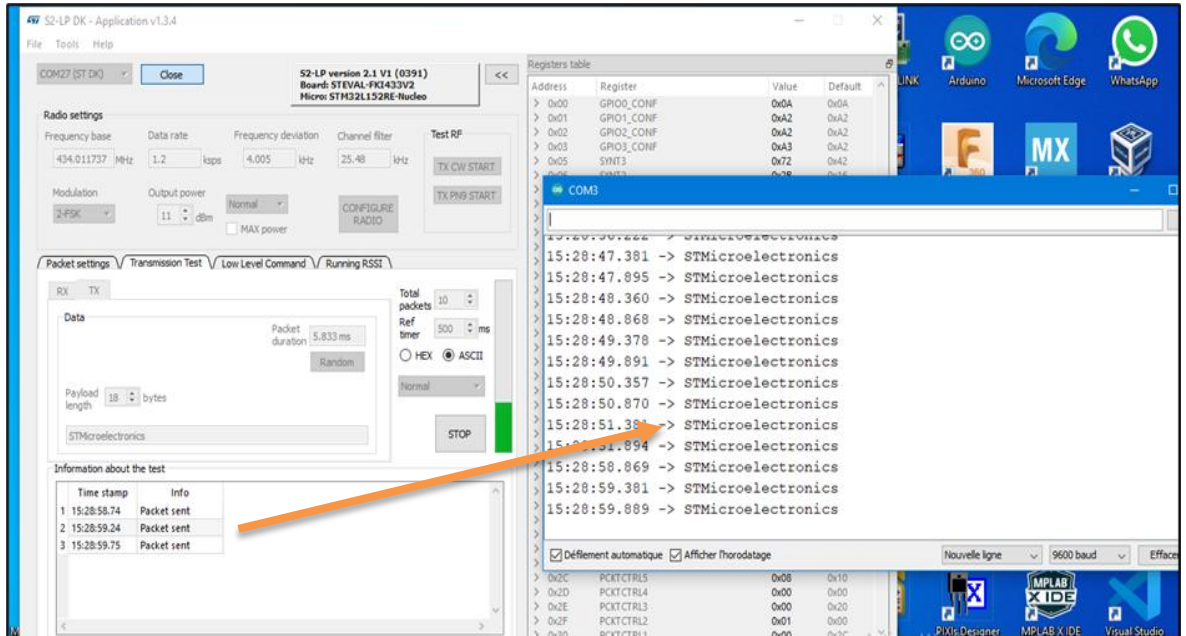


Figure 3.35 Fonctionnement du mode réception

3.2.6 Synthèse

La réalisation de la version NUCLEO 64 – CCRF a été utile pour valider notre projet en module déjà disponible au CDTA.

Cela nous a permis de faire beaucoup de développement et d'avancement dans la partie software avant de faire la conception finale du module.

Nous allons présenter dans ce qui suit le circuit final du module RF.

3.3 Conception du module

3.3.1 Module RF en version CC1120 – STM32F103C8T6

A cause de la rupture de stock du microcontrôleur STM32L476RG sur le marché des composants électroniques, on a dû le remplacer par la suite avec un autre microcontrôleur disponible qui est le STM32F103C8T6. Un changement qui n'a pas affecté le bon fonctionnement du module suite aux recherches faites sur le datasheet du STM32F103C8T6 [4], pour voir s'il répond au mieux aux besoins de notre application comme sont illustrés les résultats dans le tableau suivant :

STM32 F103 C8T6	STM32 L476 RGT6
CPU: ARM 32 bits cortex M3+	CPU: ARM 32 bits cortex M4
Mémoire : 128 Kb / SRAM : 20 Kb	Mémoire : 1 Mb Flash / SRAM : 128 Kb
7 TIMERS, 2 I ² C, 3 UART, 2 SPI, USB OTG, CAN BUS, RTC interne	16 TIMERS, LCD Drive, 3*SPI, CAN, 2*SAI, 5UART, 3 I ² C
ADC, 80 I/O, Temperature Sensor	Capacitive touch Inputs, RTC interne

Tableau 3.1 Etude comparative entre les deux MCU

3.3.2 Implémentation

Pour l'implémentation de notre module de communication Radio RF, le circuit CC1120 va être maintenant directement connecté au STM32 maître pour établir une communication matérielle via le bus SPI à 4 fils (SI, SO, SCLK et CS).

Dans notre module le CC1120 est l'esclave, et le microcontrôleur STM32 est monté en Standalone sur le PCB.

Le maître utilise le bus SPI pour lire et écrire des données mises en mémoire tampon comme déjà vu précédemment.

3.3.3 Conception du schéma électronique

Après avoir fait le choix du microcontrôleur, nous nous sommes intéressés à la conception du schéma électronique du module.

Le tableau dans la figure 3.36 montre la disposition des broches de l'interface SPI1 [4] utilisée dans le microcontrôleur STM32F103C8T6 [4] :

Alternate function	SPI1_REMAP = 0	SPI1_REMAP = 1
SPI1_NSS	PA4	PA15
SPI1_SCK	PA5	PB3
SPI1_MISO	PA6	PB4
SPI1_MOSI	PA7	PB5

Figure 3.36 Disposition des Broches SPI1

La broche SPI1_NSS avec la configuration Motorola peut être configurée sur n'importe quelle broche GPIO, cependant nous avons choisi le PB0 au lieu de PA4.

La figure 3.37 pris de la datasheet [4] montre la disposition des broches de l'UART1

C9	D10	30	C7	42	68	21	PA9	I/O	FT	PA9	USART1_TX ⁽⁹⁾ / TIM1_CH2 ⁽⁹⁾	-
D10	C12	31	C6	43	69	22	PA10	I/O	FT	PA10	USART1_RX ⁽⁹⁾ / TIM1_CH3 ⁽⁹⁾	-

Figure 3.37 Disposition des broches UART1

La broche PA9 est la broche USART TX, et la broche PA10 est celle de l'USART RX.

3.3.4 Schéma électronique

En ayant toutes les informations nécessaires au câblage de notre application, nous avons utilisé le logiciel gratuit KiCad pour dessiner le schéma électronique de la figure 3.38 :

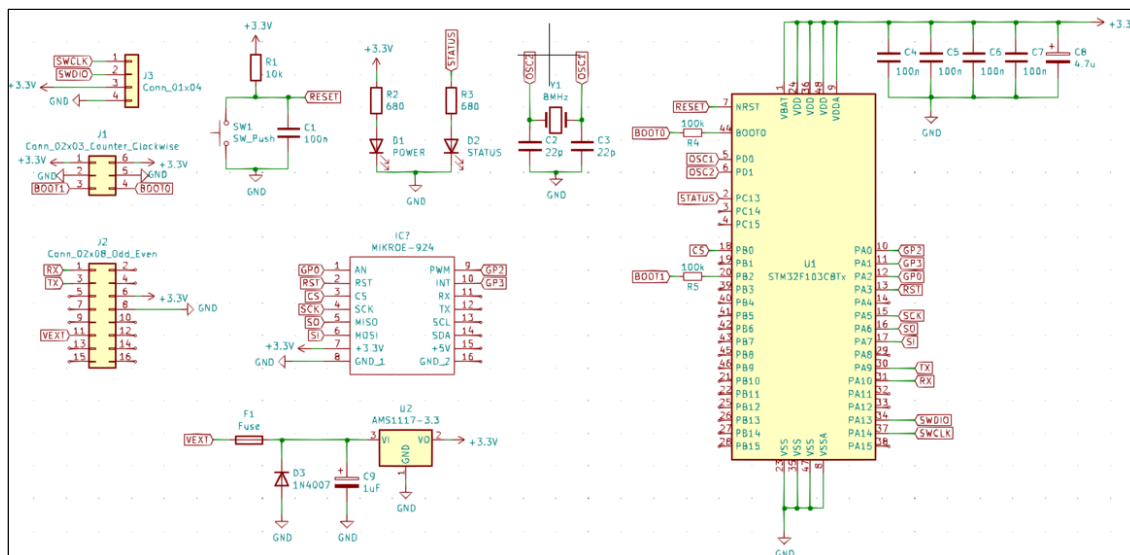


Figure 3.38 Schéma électronique

3.3.5 Les différentes parties du schéma électronique

a MCU STM32F103C8T6 (U1)

Le microcontrôleur est l'élément clé et le cœur du module, il permet de communiquer avec le CC1120 via le protocole SPI, de le configurer à chaque mise sous tension, de traiter les données reçus par le RTU et par les capteurs, et peut aussi commander des actionneurs ou des prés actionneurs.

b Les condensateurs de filtrage (C4, C5, C6, C7, C8)

Les condensateurs sont impératifs au bon fonctionnement du circuit, ils sont connectés le plus proche possible du microcontrôleur. Ce qui est un conseil de conception décrit par le fabricant dans la datasheet, et on l'a bien respecté durant notre réalisation.

c Le circuit d'horloge (Y1, C2, C3)

Nécessaire au fonctionnement du microcontrôleur, il cadence le processeur et les périphériques de communication. Le circuit d'horloge comporte un Crystal résonateur d'une fréquence de 8MHz relié à des condensateurs de filtrage 22pF.

d Le circuit RESET (R1, C1, SW1)

Le circuit Reset est responsable de la réinitialisation du circuit, en cas de disfonctionnement ou de blocage du programme.

e Bornier BOOT

Ce bornier permet de sélectionner le mode de démarrage du microcontrôleur, parmi les modes décrits dans le tableau suivant :

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
X	0	User Flash memory	User Flash memory is selected as the boot space
0	1	System memory	System memory is selected as the boot space
1	1	Embedded SRAM	Embedded SRAM is selected as the boot space

Figure 3.39 Modes de BOOT du STM32

Nous avons choisi le mode User flash memory où le vecteur d'initialisation du processeur ARM va se brancher sur une adresse mémoire flash qui est la mémoire système.

Pour cela on met un cavalier sur la borne BOOT0 qui est mise à la masse (GND) et la borne BOOT1 qui peut prendre n'importe quel potentiel +VCC 3.3V ou GND.

f Bornier de programmation (J3)

Notre module comporte également un bornier de programmation, nous l'avons rajouté pour programmer et déboguer le code directement sur notre module, et si par la suite on voudrait apporter des modifications et des ajustements.

g Les LED témoins (D1, D2)

Notre module comporte des LED qui décrivent l'état du bon fonctionnement de notre module, la LED Power s'allume lors de la mise sous tension de la carte, alors que la LED STATUS s'illumine lors d'une exécution d'une fonction déterminée par l'utilisateur.

h Alimentation à faible puissance (U2, C9, D3, F1)

Bien que notre module s'alimente en direct du SAFIR, il peut aussi s'alimenter avec une alimentation externe (5V à 12VDC max) afin de s'interfacer avec des modules capteurs ou actionneurs.

On a choisi d'utiliser un régulateur linéaire de type AMS1117-3.3 car notre module ne demande pas beaucoup de courant ce qui en résulte une faible consommation d'énergie.

On a eu l'idée d'utiliser un régulateur linéaire, car les alimentations de découpage génèrent des perturbations de ligne, ce qui affecte le bon fonctionnement.

h.1 Calcul du rendement du régulateur

- Loi du rendement : **rendement= Pout/Pin*100** et **P = U*i**
- Ce qui veut dire : **rendement= (U*i (sortie))/ (U*i (entrée))*100**
- Le module consomme en moyenne **25mA**, la tension de sortie est **3.3V**.
- Avec une alimentation externe de **5Vdc** on aura donc :

$$r = (3.3*0.025)/(5*0.025) *100 = 66\% \text{ (bon)}$$

h.2 Calcul des résistances des LED

- En supposant que le module s'alimente avec 5V et que les LED SMD consomment 10 mA,
- On applique la loi d'ohm : $U = R*I$ d'où $R = U/I$
- Une petite application numérique : $R = 5/0.010 = 500\Omega$ Alors
- Les LED vont recevoir des résistances de 470Ω valeur proche et existante.

i Socket MikroBUS (IC1)

Décrit précédemment dans la partie ccRF click, ce socket permet d'exposer les lignes du bus SPI et assurer une liaison matérielle solide entre notre module et la carte de développement ccRF click.

j Connecteur Header SAFIR (J2)

Ce bornier a été conçu spécialement pour que notre carte puisse se fixer sur le RTU SAFIR et assure une connexion matérielle le moment de l'échange des données

Ce connecteur apporte également du courant à notre carte car il comporte le 3.3V et le GND.

3.4 Conception du PCB

Comme décrit dans notre cahier des charges, nous avons opté pour un PCB double faces et qui répond bien aux fluctuations magnétiques.

3.4.1 Les couches du PCB

a PCB avec les deux couches

Le logiciel KiCad fait une différenciation entre les deux surfaces cuivrées en leur donnant deux couleurs différentes, (TOP Copper) celle du haut en rouge et (BOTTOM COPPER) celle du bas en vert, afin de faciliter la lisibilité du PCB par le constructeur.

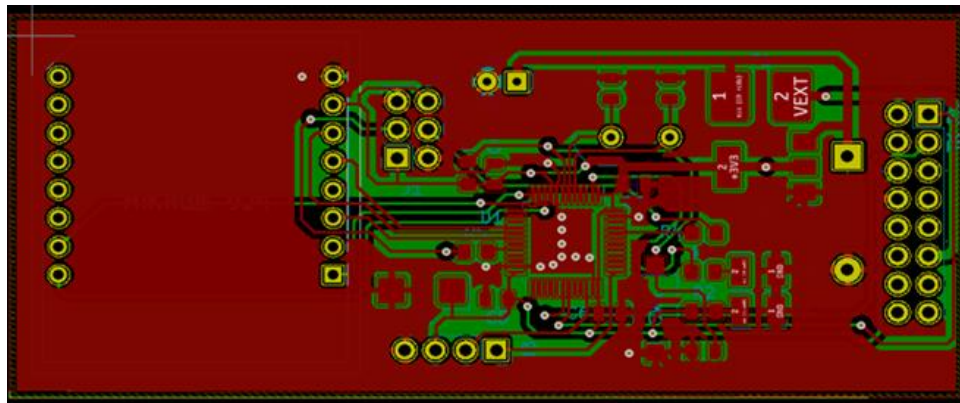


Figure 3.40 PCB comportant les deux couches

***b* Couche supérieure**

Cette couche comporte la majorité des composants utilisés tel que le STM32, les condensateurs de filtrage, les LED et leurs résistances.

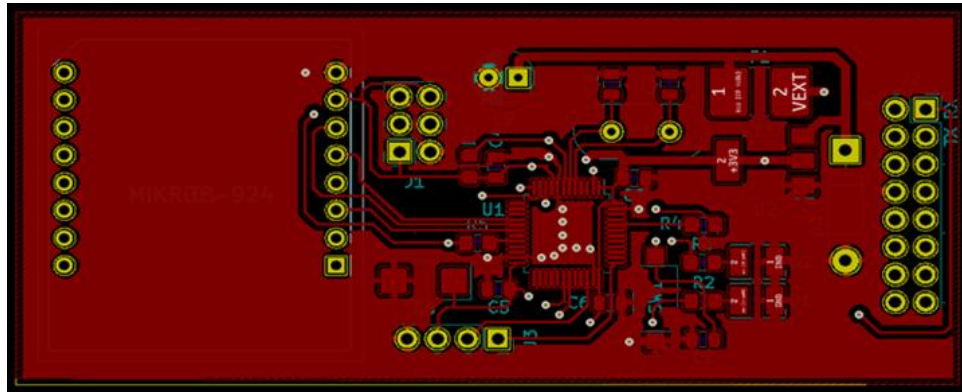


Figure 3.41 PCB couche supérieure

***c* Couche Inférieure**

Cette couche comporte les composants THT (Through Hole Technology), dont les connecteurs, le socket MikroBUS, le Crystal, et le condensateur C9 de l'alimentation font partie.

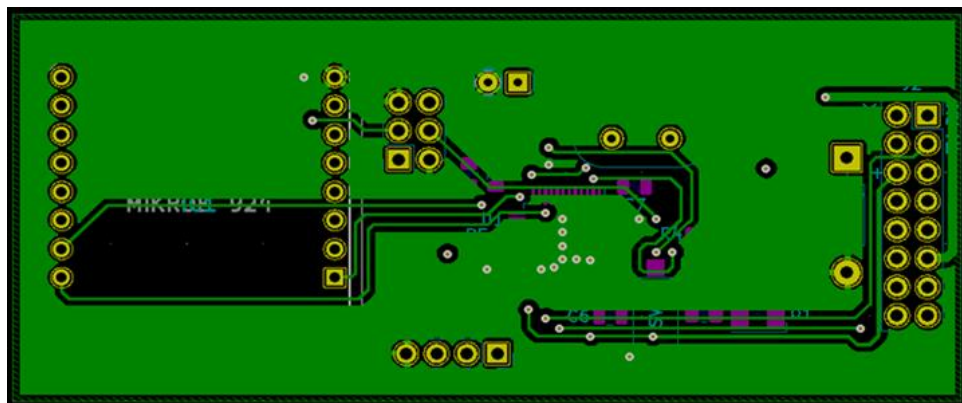


Figure 3.42 Couche inférieure

***d* Couche composants**

Cette couche comporte les empreintes (Footprint) de chaque composant utilisé, les empreintes représentent la taille exacte des connexions du composant.

Tous les footprints que nous avons utilisés font partie de la bibliothèque des composants KiCad.

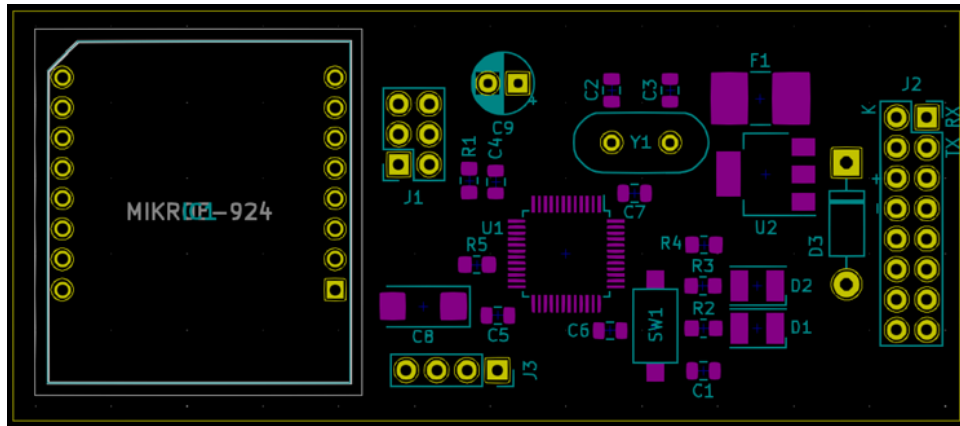


Figure 3.43 Couche composants

3.4.2 Visualisation 3D

La visualisation 3D est un outil très puissant qu'offre KiCad, il nous a permis de visualiser notre conception en 3D, et de voir aisément la disposition des composants ainsi que leurs marquages et leurs emplacements.

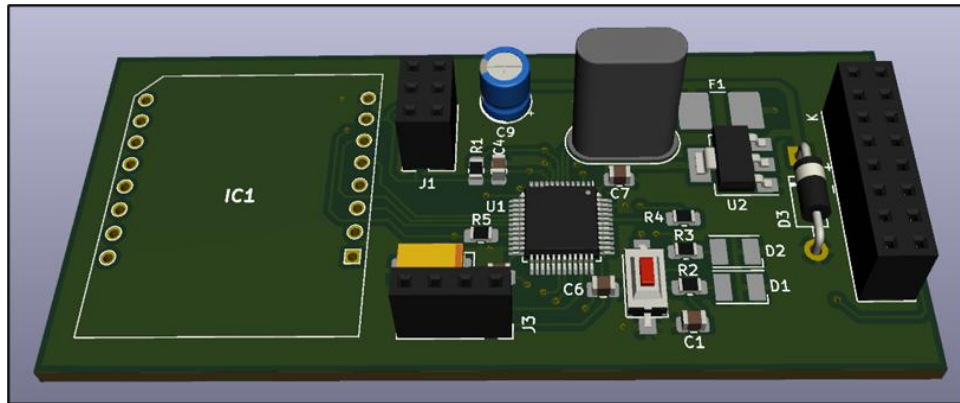


Figure 3.44 Visualisation 3D

3.4.3 Fichiers de Fabrication

Après la conception du circuit imprimé, KiCad donne la possibilité de générer deux types de fichiers,

a Fichiers Gerber

Le fichier gerber rassemble toutes les données de fabrication, tels que le format du découpage '**edge cuts**', le perçage '**drill**', la sérigraphie des deux couches '**bottom and top layer**' et le marquage de composants '**silk screen**'.

b Fichier BOM

Ce fichier comporte le nombre de composants utilisés ainsi que leurs tailles, leurs formes, la quantité, le nom de l'empreinte, et parfois même le constructeur du composant.

c Footprint position

Ce fichier comporte le nombre de composants utilisés ainsi que leurs tailles, leurs formes, la quantité, le nom de l’empreinte, et parfois même le constructeur du composant.

3.5 Prototypage et tests :

3.5.1 Assemblage du module

Les performances élevées en qualité de fabrication de circuits imprimés, le cout également très attractif, la livraison, le service et sa rapidité ont fait qu'on a opté pour le choix du fabricant chinois JLC PCB.



Figure 3.45 Ordre des PCB et des composants

La figure 3.46 présente le module RF (commande au nombre de 5) réalisé par JLC PCB.

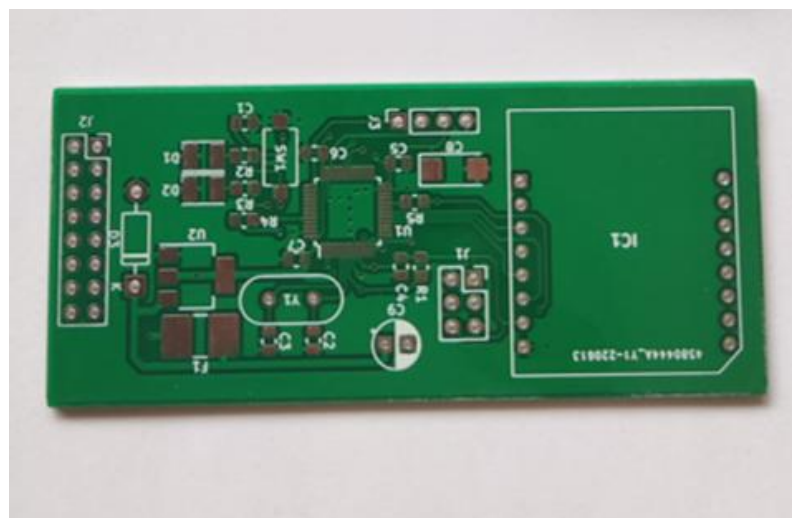


Figure 3.46 PCB du module final

La **figure 3.47** montre Le module ccRF click monté sur la carte, également commandé au nombre de trois et fabriqué par MikroElektronika, USA.

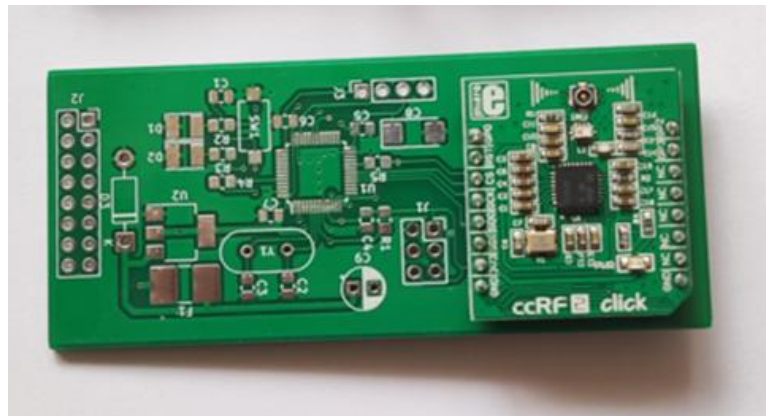


Figure 3.47 PCB avec carte ccRF monté

Nous avons procédé à la soudure des composants mentionnés dans l'annexe B, on a ainsi obtenu les résultats suivants.

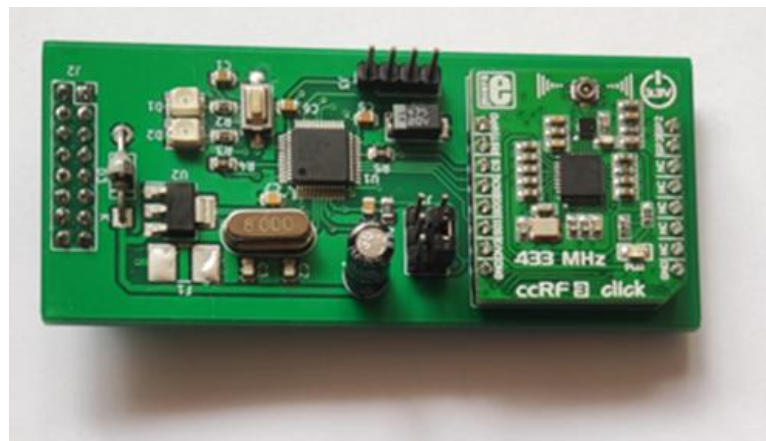


Figure 3.48 PCB avec carte ccRF monté

3.5.2 Chargement du code dans le module

Pour envoyer le code sur notre module, nous utilisons un ST Link V2 externe auquel on connecte les lignes SWDIO et SWCLK prévus à cet effet.

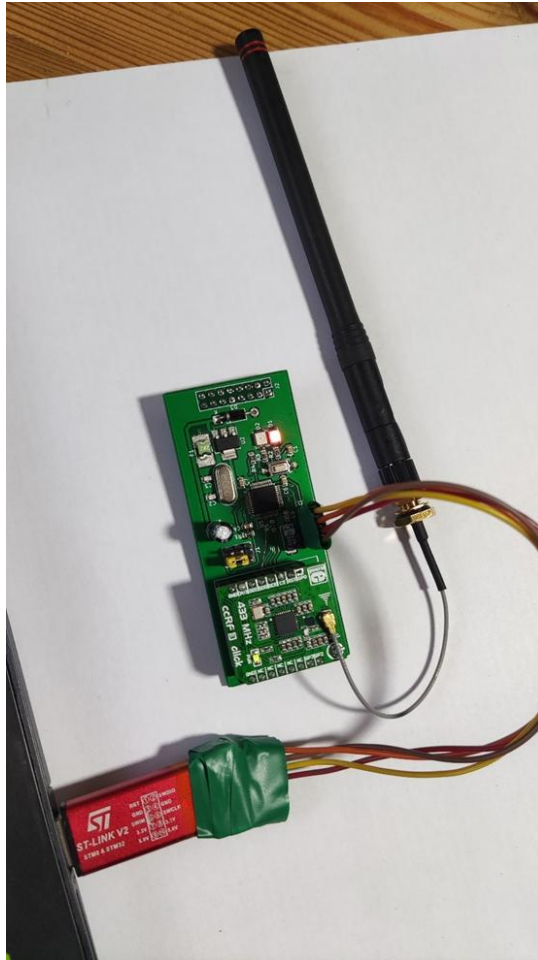


Figure 3.49 Programmation du module

On appui sur le bouton Téléverser, et le programme Arduino est télécharger sur le microcontrôleur, et le module est prêt à l'emploi.

Pour faire une différenciation entre les deux modules, nous avons conçu deux codes pour chacun des microcontrôleurs

Le code du STM32F103C8T6 diffère dans la partie d'affectation des fonctions des broches

3.6 Conclusion

Après avoir procédé à toutes les investigations nécessaires à l'étude, à la conception et au développement de notre module RF, nous avons réussi à le réaliser et à le mettre en œuvre.

Conclusion générale

L'objectif de notre travail est de réaliser un module de communication numérique longue portée.

Pour cela on a présenté dans notre mémoire, toutes les notions sur les outils, les modules, les composants, et les logiciels nécessaires dans la conception et le développement du module de communication.

Afin d'améliorer la qualité et la performance de l'unité télémétrique SAFIR pour la société DOOFAS spécialisée dans les installations des stations de pompage, nous avons pu atteindre notre objectif, celui de trouver une solution aux contraintes :

- D'économie d'énergie ;
- De longue portée de transmission ;
- De frais d'installation et de déploiement bas ;
- de maintenance.

Notre démarche nous a conduits à la réalisation du module RF, à savoir la programmation, la conception du module, la conception du PCB, le prototypage et le test du module.

Au final, on peut dire que notre objectif est atteint, le module de communication réalisé est fonctionnel, prêt à être utilisé, répondant aux besoins du demandeur de service, assurant l'interconnexion entre les différents dispositifs de l'installation de télégestion.

Annexes A :

Les registres du circuit cc1120 sont présentés dans la partie suivante.

- Octet d'état de la puce (Status) :

Le premier bit, *S7*, est le signal *CHIP_RDYn* et ce signal doit passer à l'état bas avant le premier front montant de *SCLK*. Le signal *CHIP_RDYn* signifie que le cristal est stable. *S6*, *S5* et *S4* comprennent la valeur du *STATE* qui reflète l'état de la puce. Les autres bits sont réservés.[5]

Bits	Name	Description																																				
7	CHIP_RDYn	Stays high until power and crystal have stabilized. Should always be low when using the SPI interface.																																				
6:4	STATE[2:0]	Indicates the current main state machine mode																																				
		<table border="1"> <thead> <tr> <th>Value</th> <th>State</th> <th>MARC State</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>IDLE</td> <td>IDLE</td> <td>IDLE state</td> </tr> <tr> <td>001</td> <td>RX</td> <td>RX RX_END</td> <td>Receive mode</td> </tr> <tr> <td>010</td> <td>TX</td> <td>TX TX_END</td> <td>Transmit mode</td> </tr> <tr> <td>011</td> <td>FSTXON</td> <td>FSTXON</td> <td>Fast TX ready</td> </tr> <tr> <td>100</td> <td>CALIBRATE</td> <td>BIAS_SETTLE_MC REG_SETTLE_MC MANCAL STARTCAL ENDCAL</td> <td>Frequency synthesizer calibration is running</td> </tr> <tr> <td>101</td> <td>SETTLING</td> <td>BIAS_SETTLE REG_SETTLE BWBOOST FS_LOCK IFADCON RXTX_SWITCH TXRX_SWITCH IFADCON_TXRX</td> <td>PLL is settling</td> </tr> <tr> <td>110</td> <td>RX FIFO ERROR</td> <td>RX_FIFO_ERR</td> <td>RX FIFO has over/underflowed. Read out any useful data, then flush the FIFO with an <i>SFRX</i> strobe</td> </tr> <tr> <td>111</td> <td>TX FIFO ERROR</td> <td>TX_FIFO_ERR</td> <td>TX FIFO has over/underflowed. Flush the FIFO with an <i>SFTX</i> strobe</td> </tr> </tbody> </table>	Value	State	MARC State	Description	000	IDLE	IDLE	IDLE state	001	RX	RX RX_END	Receive mode	010	TX	TX TX_END	Transmit mode	011	FSTXON	FSTXON	Fast TX ready	100	CALIBRATE	BIAS_SETTLE_MC REG_SETTLE_MC MANCAL STARTCAL ENDCAL	Frequency synthesizer calibration is running	101	SETTLING	BIAS_SETTLE REG_SETTLE BWBOOST FS_LOCK IFADCON RXTX_SWITCH TXRX_SWITCH IFADCON_TXRX	PLL is settling	110	RX FIFO ERROR	RX_FIFO_ERR	RX FIFO has over/underflowed. Read out any useful data, then flush the FIFO with an <i>SFRX</i> strobe	111	TX FIFO ERROR	TX_FIFO_ERR	TX FIFO has over/underflowed. Flush the FIFO with an <i>SFTX</i> strobe
Value	State	MARC State	Description																																			
000	IDLE	IDLE	IDLE state																																			
001	RX	RX RX_END	Receive mode																																			
010	TX	TX TX_END	Transmit mode																																			
011	FSTXON	FSTXON	Fast TX ready																																			
100	CALIBRATE	BIAS_SETTLE_MC REG_SETTLE_MC MANCAL STARTCAL ENDCAL	Frequency synthesizer calibration is running																																			
101	SETTLING	BIAS_SETTLE REG_SETTLE BWBOOST FS_LOCK IFADCON RXTX_SWITCH TXRX_SWITCH IFADCON_TXRX	PLL is settling																																			
110	RX FIFO ERROR	RX_FIFO_ERR	RX FIFO has over/underflowed. Read out any useful data, then flush the FIFO with an <i>SFRX</i> strobe																																			
111	TX FIFO ERROR	TX_FIFO_ERR	TX FIFO has over/underflowed. Flush the FIFO with an <i>SFTX</i> strobe																																			
3:0	Reserved																																					

Figure A.1 Registre Octet d'état

- Les types d'accès :

Access type	Command/Address byte	Description
Single Register Access (register space)	Address: R/W 0 A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ (A _{5..0} < 0x2F)	The R/W bit determines whether the operation is a read (1) or a write (0) operation The register accessed is determined by the address in A _{5..0} Exactly one data byte is expected after the address byte The chip status byte is returned on the SO line both when the address is sent on the SI line as well as when data are written
Burst Register Access (register space)	Address: R/W 1 A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ (A _{5..0} < 0x2F)	The R/W bit determines whether the operation is a read (1) or a write (0) operation The address in A _{5..0} determines the first register accessed, after which an internal address counter is incremented for each new data byte following the address byte Consecutive bytes are expected after the address byte and the burst access is terminated by setting CSn high The chip status byte is returned on the SO line both when the address is sent on the SI line as well as when data are written If the internal address counter reaches address 0x2E (last byte in register space) the counter will not increment anymore and the same address will be read/written until the burst access is being terminated
Single Register Access (extended register space)	Command: R/W 0 1 0 1 1 1 1 Address: A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ (A _{7..0} : See Table 5)	This access mode starts with a specific command (0x2F) The first byte following this command is interpreted as the extended address Exactly one data byte is expected after the extended address byte When the extended address is sent on the SI line, SO will return all zeros. The chip status byte is returned on the SO line when the command is transmitted as well as when data are written to the extended address

Figure A.02 Accès SPI

Access type	Command/Address byte	Description
Burst Register Access (extended register space)	Command: R/W 1 1 0 1 1 1 1 Address: A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ (A _{7..0} : See Table 5)	This access mode starts with a specific command (0x2F) The first byte following this command is interpreted as the extended address Consecutive bytes are expected after the extended address byte and the burst access is terminated by setting CS _n high When the extended address is sent on the SI line, SO will return all zeros. The chip status byte is returned on the SO line when the command is transmitted as well as when data are written to the extended address. If the internal address counter reaches address 0xFF (last byte in extended register space) the counter will wrap around to 0x00 Registers not listed in Table 5 can be part of a burst access
Command Strobe Access	Address: R/W 0 A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ (0x30 ≤ A _{5..0} ≤ 0x3D)	Accessing one of the command strobe registers triggers an event determined by the address in A _{5..0} , e.g. resetting the device, enabling the crystal oscillator, entering TX, etc. No data byte is expected. The chip status byte is returned on the SO line when a command strobe is sent on the SI line
Direct FIFO Access	Command: R/W B 1 1 1 1 1 0 Address: A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀ A _{7..0} < 0x80: TX FIFO 0x80 ≤ A _{7..0} ≤ 0xFF: RX FIFO	This access mode starts with a specific command (0x3E) which makes it possible to access the FIFOs directly through memory operations without affecting the FIFO pointers. The first byte following this command is interpreted as the address. The next byte is read/written to this address. If burst is enabled, consecutive bytes will be read/written by incrementing the address. ¹ FIFO pointers are available in extended register space for debug purposes.
Standard FIFO Access	Address: R/W B 1 1 1 1 1 1	The R/W bit determines whether the operation is a read (1) operation from the RX FIFO or a write (0) operation to the TX FIFO. If the burst bit B is 1, all bytes following the address byte are treated as data bytes until CS _n goes high. If the burst bit B is 0, the FIFOs are accessed byte-wise as a normal register.

Figure A.3 Suite de l'espace registre SPI

- Espace d'adressage SPI :

	Write		Read		
	Single Byte	Burst	Single Byte	Burst	
	+0x00	+0x40	+0x80	+0xC0	
0x00			IOCFG3		R/W configuration registers, burst access possible
0x01			IOCFG2		
0x02			IOCFG1		
0x03			IOCFG0		
0x04			SYNC3		
0x05			SYNC2		
0x06			SYNC1		
0x07			SYNC0		
0x08			SYNC_CFG1		
0x09			SYNC_CFG0		
0x0A			DEVIATION_M		
0x0B			MODCFG_DEV_E		
0x0C			DCFILT_CFG		
0x0D			PREAMBLE_CFG1		
0x0E			PREAMBLE_CFG0		
0x0F			FREQ_IF_CFG		
0x10			IQIC		
0x11			CHAN_BW		
0x12			MDMCFG1		
0x13			MDMCFG0		
0x14			SYMBOL_RATE2		
0x15			SYMBOL_RATE1		
0x16			SYMBOL_RATE0		
0x17			AGC_REF		
0x18			AGC_CS_THR		
0x19			AGC_GAIN_ADJUST		
0x1A			AGC_CFG3		
0x1B			AGC_CFG2		
0x1C			AGC_CFG1		
0x1D			AGC_CFG0		
0x1E			FIFO_CFG		
0x1F			DEV_ADDR		
0x20			SETTLING_CFG		
0x21			FS_CFG		
0x22			WOR_CFG1		
0x23			WOR_CFG0		
0x24			WOR_EVENT0_MSB		
0x25			WOR_EVENT0_LSB		
0x26			PKT_CFG2		
0x27			PKT_CFG1		
0x28			PKT_CFG0		
0x29			RFEND_CFG1		
0x2A			RFEND_CFG0		
0x2B			PA_CFG2		
0x2C			PA_CFG1		
0x2D			PA_CFG0		
0x2E			PKT_LEN		
0x2F			EXTENDED ADDRESS		
0x30	SRES		SRES		Command Stobes
0x31	SFSTXON		SFSTXON		
0x32	SXOFF		SXOFF		
0x33	SCAL		SCAL		
0x34	SRX		SRX		
0x35	STX		STX		
0x36	SIDLE		SIDLE		
0x37	SAFC		SAFC		
0x38	SWOR		SWOR		
0x39	SPWD		SPWD		
0x3A	SFRX		SFRX		
0x3B	SFTX		SFTX		
0x3C	SWORRST		SWORRST		
0x3D	SNOP		SNOP		
0x3E			DIRECT MEMORY ACCESS		
0x3F	TX FIFO	TX FIFO	RX FIFO	RX FIFO	

Figure A.4 Espace registre SPI

- Accès espace registre et accès espace de registre étendu :

Les registres de configuration du CC112X sont situés sur des adresses SPI de 0x00 à 0x2E (espace de registre) avec une commande d'extension d'adresse à l'adresse 0x2F pour accéder à l'espace de registre étendu (voir Figure A-1). Tous les registres de configuration peuvent être à la fois écrits et lus et ceci est contrôlé par le bit R/W dans l'octet d'en-tête. Tous les registres de configuration sont également accessibles avec le bit de rafale (B) défini sur 1 ou 0. Notez que tous les registres de l'espace de registre (adresse 0x00 - 0x2E) ont une rétention. Dans l'espace de registre étendu, les registres d'état et les pointeurs FIFO n'ont pas de rétention.

Extended Register Space (0x00 - 0x2F)		Retention
0x00	IF_MIX_CFG	Yes
0x01	FREQOFF_CFG	Yes
0x02	TOC_CFG	Yes
0x03	MARC_SPARE	Yes
0x04	ECG_CFG	Yes
0x05	CFM_DATA_CFG	Yes
0x06	EXT_CTRL	Yes
0x07	RCCAL_FINE	Yes
0x08	RCCAL_COARSE	Yes
0x09	RCCAL_OFFSET	Yes
0x0A	FREQOFF1	Yes
0x0B	FREQOFF0	Yes
0x0C	FREQ2	Yes
0x0D	FREQ1	Yes
0x0E	FREQ0	Yes
0x0F	IF_ADC2	Yes
0x10	IF_ADC1	Yes
0x11	IF_ADC0	Yes
0x12	FS_DIG1	Yes
0x13	FS_DIG0	Yes
0x14	FS_CAL3	Yes
0x15	FS_CAL2	Yes
0x16	FS_CAL1	Yes
0x17	FS_CAL0	Yes
0x18	FS_CHP	Yes
0x19	FS_DIVTWO	Yes
0x1A	FS_DSM1	Yes
0x1B	FS_DSM0	Yes
0x1C	FS_DVC1	Yes
0x1D	FS_DVC0	Yes
0x1E	FS_LBI	Yes
0x1F	FS_PFD	Yes
0x20	FS_PRE	Yes
0x21	FS_REG_DIV_CML	Yes
0x22	FS_SPARE	Yes
0x23	FS_VCO4	Yes
0x24	FS_VCO3	Yes
0x25	FS_VCO2	Yes
0x26	FS_VCO1	Yes
0x27	FS_VCO0	Yes
0x28	GBIAS6	Yes
0x29	GBIAS5	Yes
0x2A	GBIAS4	Yes
0x2B	GBIAS3	Yes
0x2C	GBIAS2	Yes
0x2D	GBIAS1	Yes
0x2E	GBIAS0	Yes
0x2F	IFAMP	Yes

Extended Register Space (0x30 - 0x86)		Retention
0x30	LNA	Yes
0x31	RXMIX	Yes
0x32	XOSC5	Yes
0x33	XOSC4	Yes
0x34	XOSC3	Yes
0x35	XOSC2	Yes
0x36	XOSC1	Yes
0x37	XOSC0	Yes
0x38	ANALOG_SPARE	Yes
0x39	PA_CFG3	Yes
0x3A - 0x3E	Not Used	
0x3F - 0x40	Reserved	
0x41 - 0x63	Not Used	
0x64	WOR_TIME1	No
0x65	WOR_TIME0	No
0x66	WOR_CAPTURE1	No
0x67	WOR_CAPTURE0	No
0x68	BIST	No
0x69	DCFILTOFFSET_I1	No
0x6A	DCFILTOFFSET_I0	No
0x6B	DCFILTOFFSET_Q1	No
0x6C	DCFILTOFFSET_Q0	No
0x6D	IQIE_I1	No
0x6E	IQIE_I0	No
0x6F	IQIE_Q1	No
0x70	IQIE_Q0	No
0x71	RSSI1	No
0x72	RSSI0	No
0x73	MARCSTATE	No
0x74	LQI_VAL	No
0x75	PQT_SYNC_ERR	No
0x76	DEM_STATUS	No
0x77	FREQOFF_EST1	No
0x78	FREQOFF_EST0	No
0x79	AGC_GAIN3	No
0x7A	AGC_GAIN2	No
0x7B	AGC_GAIN1	No
0x7C	AGC_GAIN0	No
0x7D	CFM_RX_DATA_OUT	No
0x7E	CFM_TX_DATA_IN	No
0x7F	ASK_SOFT_RX_DATA	No
0x80	RNDGEN	No
0x81	MAGN2	No
0x82	MAGN1	No
0x83	MAGN0	No
0x84	ANG1	No
0x85	ANG0	No
0x86	CHFILT_I2	No

Figure A.5 Espace registre SPI étendu

Extended Register Space (0x87 - 0x98)			Retention		
0x87	CHFILT_I1	No			
0x88	CHFILT_I0	No			
0x89	CHFILT_Q2	No			
0x8A	CHFILT_Q1	No			
0x8B	CHFILT_Q0	No			
0x8C	GPIO_STATUS	No			
0x8D	FSCAL_CTRL	No			
0x8E	PHASE_ADJUST	No			
0x8F	PARTNUMBER	No			
0x90	PARTVERSION	No			
0x91	SERIAL_STATUS	No			
0x92	MODEM_STATUS1	No			
0x93	MODEM_STATUS0	No			
0x94	MARC_STATUS1	No			
0x95	MARC_STATUS0	No			
0x96	PA_IFAMP_TEST	No			
0x97	FSRF_TEST	No			
0x98	PRE_TEST	No			

Extended Register Space (0x99 - 0xD9)			Retention		
0x99	PRE_OVR	No			
0x9A	ADC_TEST	No			
0x9B	DVC_TEST	No			
0x9C	ATEST	No			
0x9D	ATEST_LVDS	No			
0x9E	ATEST_MODE	No			
0x9F	XOSC_TEST1	No			
0xA0	XOSC_TEST0	No			
0xA1 - 0xD1	Not Used				
0xD2	RXFIRST	No			
0xD3	TXFIRST	No			
0xD4	RXLAST	No			
0xD5	TXLAST	No			
0xD6	NUM_TXBYTES	No			
0xD7	NUM_RXBYTES	No			
0xD8	FIFO_NUM_TXBYTES	No			
0xD9	FIFO_NUM_RXBYTES	No			

Figure A.6 Suite Espace registre SPI étendu

- Commandes du circuit cc120

Les commandes peuvent être considérées comme des instructions à un seul octet pour le CC112X. En adressant un registre de commande, des séquences internes seront lancées. Ces commandes sont utilisées pour activer le mode de réception et de transmission, entrer en mode veille, désactiver l'oscillateur à quartz, etc.

Address	Strobe Name	Description
0x30	SRES	Reset chip
0x31	SFSTXON	Enable and calibrate frequency synthesizer (if <code>SETTLING_CFG.FS_AUTOCAL = 1</code>). If in RX and <code>PKT_CFG2.CCA_MODE ≠ 0</code> : Go to a wait state where only the synthesizer is running (for quick RX/TX turnaround).
0x32	SXOFF	Enter XOFF state when CSn is de-asserted
0x33	SCAL	Calibrate frequency synthesizer and turn it off. SCAL can be strobed from IDLE mode without setting manual calibration mode (<code>SETTLING_CFG.FS_AUTOCAL = 0</code>)
0x34	SRX	Enable RX. Perform calibration first if coming from IDLE and <code>SETTLING_CFG.FS_AUTOCAL = 1</code>
0x35	STX	In IDLE state: Enable TX. Perform calibration first if <code>SETTLING_CFG.FS_AUTOCAL = 1</code> . If in RX state and <code>PKT_CFG2.CCA_MODE ≠ 0</code> : Only go to TX if channel is clear
0x36	SIDLE	Exit RX/TX, turn off frequency synthesizer and exit eWOR mode if applicable
0x37	SAFC	Automatic Frequency Compensation
0x38	SWOR	Start automatic RX polling sequence (eWOR) as described in Section 9.6 if <code>WOR_CFG0.RC_PD = 0</code>
0x39	SPWD	Enter SLEEP mode when CSn is de-asserted
0x3A	SFRX	Flush the RX FIFO. Only issue SFRX in IDLE or RX_FIFO_ERR states
0x3B	SFTX	Flush the TX FIFO. Only issue SFTX in IDLE or TX_FIFO_ERR states
0x3C	SWORRST	Reset the eWOR timer to the Event1 value
0x3D	SNOP	No operation. May be used to get access to the chip status byte

Figure A.7 Commandes du circuit cc120

Autrement dit, seuls le bit R/W, le bit d'accès en rafale (mis à 0) et les six bits d'adresse (dans la plage de 0x30 à 0x3D) sont écrits.

Lors de l'envoi d'un stroboscope, le bit R/W⁻ peut être soit un soit zéro. L'octet d'état est disponible sur la broche SO lors de l'envoi d'un stroboscope de commande.

- Accès directs du FIFO :

Direct FIFO Access Mapping		Retention
0x00 - 0x7F	TXFIFO	No
0x80 - 0xFF	RXFIFO	No

Figure A.08 Adressage du FIFO

Annexes B :

- **BOM**

Listes des composants :

Id	Designator	Package	Quantity	Designation
1	U2	SOT-223-3_TabPin2	1	AMS1117-3.3
2	J1	PinSocket_2x03_P2.54mm_Vertical	1	Conn_02x03_Counter_Clockwise
3	R1	R_0805_2012Metric_Pad1.20x1.40mm_HandSolder	1	10k
4	Y1	Crystal_HC18-U_Vertical	1	8MHz
5	U1	LQFP-48_7x7mm_P0.5mm	1	STM32F103C8Tx
6	SW1	SW_SPST_CK_RS282G05A3	1	SW_Push
7	R5,R4	R_0805_2012Metric_Pad1.20x1.40mm_HandSolder	2	100k
8	R3,R2	R_0805_2012Metric_Pad1.20x1.40mm_HandSolder	2	680
9	J3	PinSocket_1x04_P2.54mm_Vertical	1	Conn_01x04
10	J2	PinSocket_2x08_P2.54mm_Vertical	1	Conn_02x08_Odd_Even
11	IC1	MIKROE924	1	MIKROE-924
12	F1	Fuse_2816_7142Metric_Pad3.20x4.45mm_HandSolder	1	Fuse
13	D3	D_DO-41_SOD81_P10.16mm_Horizontal	1	1N4007
14	D2	LED_PLCC-2	1	STATUS
15	D1	LED_PLCC-2	1	POWER
16	C9	CP_Radial_D5.0mm_P2.50mm	1	1uF
17	C8	CP_EIA-6032-15_Kemet-U_Pad2.25x2.35mm_HandSolder	1	4.7u
18	C7,C6,C5,C4,C1	C_0805_2012Metric	5	100n
19	C3,C2	C_0805_2012Metric	2	22p

Figure B.01 BOM

Bibliographie

[1] – AutomationSense, (2015). Qu'est ce que la télégestion ? [en ligne]. AutomationSense. [Consulté le 13 juin 2022]. Disponible sur : <https://www.automation-sense.com/blog/automatisme/qu-est-ce-que-la-telegestion.html>

[2] – Jennifer, (2021). La télégestion : que retenir à propos ? [en ligne]. ma-pomme. [Consulté le 13 juin 2022]. Disponible sur : <https://ma-pomme.fr/la-telegestion-que-retenir-a-propos-10929.html>

[3] – CODRA. Télégestion [en ligne]. Codra. [Consulté le 13 juin 2022]. Disponible sur : <https://codra.net/fr/solution/plateforme-logiciel/communication/telegestion/>

[4] – COPADATA, (17 octobre). Qu'est-ce qu'un SCADA ? [en ligne]. COPADATA. [Consulté le 14 juin 2022]. Disponible sur : <https://www.copadata.com/fr/produits/zenon-software-platform/visualisation-controle/qu-est-ce-qu-un-scada-supervisory-control-and-data-acquisition-copa-data/>

[5] – Nandini Raghvendra, (12 février 2019). SCADA System - Components, Hardware & Software Architecture, Types [en ligne]. electricalfundablog. [Consulté le 21 juin 2022]. Disponible sur : <https://electricalfundablog.com/scada-system-components-architecture/>

[6] – Mondi Anderson, (3 septembre 2018). WHAT IS RTU ? [en ligne]. REALPARS. [Consulté le 21 juin 2022]. Disponible sur : <https://realpars.com/rtu/>

[7] – The Automization, (08 mai 2020). What is Modbus TCP/IP ? | Modbus Tcp/ip vs Ethernet ip [en ligne]. The Automization. [Consulté le 22 juin 2022]. Disponible sur : <https://theautomization.com/what-is-modbus-tcp-ip/>

[8] – Wally Gastrelch, (3 décembre 2018). WHAT IS MODBUS ? [en ligne]. REALPARS. [Consulté le 7 juin 2022]. Disponible sur : <https://realpars.com/modbus/>

[9] – Legrand, (). Protocole de communication Série Modbus® [Fichier PDF]. Legrand. [Consulté le 18 juin 2022]. Disponible sur : <https://www.electronique-mixte.fr/wp-content/uploads/2018/07/Formation-Interface-communication-30.pdf>

[10] – Acromag, Inc, (2005). INTRODUCTION TO MODBUS TCP/IP [Fichier PDF]. Acromag, Inc. [Consulté le 22 juin 2022]. Disponible sur : https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf

[11] – Philpetitpa, (18 novembre 2008). LE PROTOCOLE MODBUS [Fichier PDF]. Philpetitpa. [Consulté le 22 juin 2022]. Disponible sur : <https://philpetitpa.pagesperso-orange.fr/modbus.pdf>

[12] – Michael Bowne, (10 juin 2020). The Difference between PROFIBUS and PROFINET [en ligne]. PI North America. [Consulté le 7 juillet 2022]. Disponible sur : <https://us.profinet.com/the-difference-between-profibus-and-profinet/>

[13] – PROFIBUS &PROFINET International, (novembre 2018). Description du système PROFINET Technologie et application [Fichier PDF]. PROFIBUS &PROFINET International. [Consulté le 22 juin 2022]. Disponible sur : https://www.profibus.fr/wp-content/uploads/2021/04/profinet_descriptionsysteme_dec_2018_fr.pdf

[14] – Control Automation, (31 octobre 2019). WirelessHART Radio Communication Standard | Wireless Field Instruments and Long-range Wireless Data Links [en ligne]. Control Automation. [Consulté le 25 juin 2022]. Disponible sur : <https://control.com/textbook/wireless-instrumentation/wirelesshart/>

[15] – Penny Chen, (06 mars 2014). ISA100 Is Ready for the Internet of Things [en ligne]. Pumpsandsystems. [Consulté le 26 juin 2022]. Disponible sur : <https://www.pumpsandsystems.com/isa100-ready-internet-things>

[16] – Electronology, (20 décembre 2021). [en ligne]. Understanding ISA100 Wireless Standard. [en ligne]. Electronology. [Consulté le 26 juin 2022]. Disponible sur : <https://electronology.com/understanding-isa100-wireless-standard/>

[17] – Software Testing Help, (13 juin 2022). [en ligne]. 7 Layers Of The OSI Model (A Complete Guide). [en ligne]. Software Testing Help.[Consulté le]. Disponible sur : <https://www.softwaretestinghelp.com/osi-model-layers/>

[18] – Mark Nixon, (23 septembre 2012). A Comparison of WirelessHART™ and ISA100.11a [Fichier PDF]. Emerson Process Management. [Consulté le 25 juin 2022]. Disponible sur : <https://www.emerson.com/documents/automation/white-paper-a-comparison-of-wirelesshart-isa100-11a-en-42598.pdf>

[19] – LoRa Alliance®, (21 octobre 2020). What is LoRaWAN® Specification [en ligne]. LoRa Alliance®. [Consulté le 18 juin 2022]. Disponible sur : <https://loralliance.org/about-lorawan/>

[20] – Gabriel, (30 août 2019). An Introduction to Sigfox Technology – Basics, Architecture and Security Features [en ligne]. Circuitdigest. [Consulté le 8 juillet 2022]. Disponible sur : <https://circuitdigest.com/article/what-is-sigfox-basics-architecture-and-security-features>

[21] – MikroElektronika, (2017). CCRF 3 CLICK [en ligne]. MikroElektronika. [Consulté le 28 juin 2022]. Disponible sur : <https://www.mikroe.com/ccrf-3-click>

[22] – MikroElektronika, (juin 2015). mikroBUS™standard specifications. [Fichier PDF]. MikroElektronika. [Consulté le 28 juin 2022]. Disponible sur : <https://download.mikroe.com/documents/standards/mikrobus/mikrobus-standard-specification-v200.pdf>

[23] – Texas Instruments, (juillet 2015).CC1120 High-Performance RF Transceiver for Narrowband Systems datasheet (Rev. H) [en ligne]. Texas Instruments. [Consulté le 26 mai 2022]. Disponible sur : <https://www.ti.com/lit/ds/symlink/cc1120.pdf>

[24] – STMicroelectronics. BALF-112X-02D3 Description [en ligne]. STMicroelectronics. [Consulté le 29 juin 2022]. Disponible sur : <https://pdf1.alldatasheet.com/datasheet-pdf/view/883058/STMICROELECTRONICS/BALF-112X-02D3.html>

[25] – Texas Instruments, (27 septembre 2013).CC112x/CC1175 Low-Power High Performance Sub-1 GHz RF Transceivers User’s Guide (Rev. E). [Fichier PDF]. Texas Instruments. [Consulté le 2 juillet 2022]. Disponible sur : <https://www.ti.com/lit/ug/swru295e/swru295e.pdf>

[26] – Texas Instruments, (19 août 2011). SMARTRFSTM-STUDIO Calculation tool [en ligne]. Texas Instruments. [Consulté le 29 juin 2022]. Disponible sur : <https://www.ti.com/tool/SMARTRFSTM-STUDIO>

[27] – STMicroelectronics, (5 novembre 2022). NUCLEO-L476RG [en ligne]. STMicroelectronics. [Consulté le 30 juin 2022]. Disponible sur : <https://www.st.com/en/evaluation-tools/nucleo-l476rg.html>

[28] – STMicroelectronics. STM32 Arm Cortex MCUs - 32-bit Microcontrollers [en ligne]. STMicroelectronics. [Consulté le 30 juin 2022]. Disponible sur : <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

[29] – STMicroelectronics, (juin 2019).STM32L476xx Datasheet Rev 8 [Fichier PDF]. STMicroelectronics. [Consulté le 1 juillet]. Disponible sur : <https://www.st.com/resource/en/datasheet/stm32l476rg.pdf>

[30] – Motorola, Inc, (14 juillet 2004). SPI Block Guide V4 [Fichier PDF]. – Motorola, Inc. [Consulté le 18 juin 2022]. Disponible sur : https://www.nxp.com/files-static/microcontrollers/doc/ref_manual/S12SPIV4.pdf

[31] – STMicroelectronics.STM32L4-Peripheral-SPI interface [en ligne]. STMicroelectronics. [Consulté le 22 juin 2022]. Disponible sur : [https://st-onlinetraining.s3.amazonaws.com/STM32L4_Peripheral_SPI_interface_\(SPI\)/index.htm](https://st-onlinetraining.s3.amazonaws.com/STM32L4_Peripheral_SPI_interface_(SPI)/index.htm)

[32] – Piyu Dhaker, (18 septembre 2018). Introduction to SPI Interface [en ligne]. Signal Integrity Journal. [Consulté le 4 juillet 2022]. Disponible sur : <https://www.signalintegrityjournal.com/articles/967-introduction-to-spi-interface>

[33] – Rich Goodman, (25 octobre 2019).4 SPI modes. CPOL, CPHA[en ligne].Tistory . [Consulté le 6 juillet 2022]. Disponible sur : <https://igotit.tistory.com/entry/SPI-mode-4%EC%A2%85-CPOL-CPHA>

[34] – STMicroelectronics.STM32L4 – USART interface [en ligne]. STMicroelectronics. [Consulté le 24 juin 2022]. Disponible sur : [https://st-onlinetraining.s3.amazonaws.com/STM32L4 Peripheral USART interface \(USART\)/index.html](https://st-onlinetraining.s3.amazonaws.com/STM32L4 Peripheral USART interface (USART)/index.html)

[35] – Chap4-USART [Fichier PDF]. espace technologique. [Consulté le 24 juin 2022]. Disponible sur : <https://www.espacetechnologue.com/wp-content/uploads/2020/04/Chap4-USART.pdf>

[36] – STMicroelectronics, (4 avril 2022).ST-LINK [en ligne]. STMicroelectronics. [Consulté le 30 juin 2022]. Disponible sur : <https://wiki.st.com/stm32mpu/wiki/ST-LINK>

[37] – STMicroelectronics. ST-LINK/V2 [en ligne]. STMicroelectronics. [Consulté le 4 juillet 2022]. Disponible sur : <https://www.soselectronic.ro/products/stmicroelectronics/st-link-v2-80638>

[38] – STMicroelectronics.STM32CubeIDE - Integrated Development Environment for STM32 [en ligne]. STMicroelectronics. [Consulté le 4 juillet 2022]. Disponible sur : <https://www.st.com/en/development-tools/stm32cubeide.html>

[39] – MikroElektronika. Arduino Uno Click Shield [en ligne]. MikroElektronika. [Consulté le 30 juin 2022]. Disponible sur : <https://www.mikroe.com/arduino-uno-click-shield>