



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

UNIVERSITE KASDI MERBAH OUARGLA



Faculté des Nouvelles Technologies de l'Information et de la
Communication

Département de l'électronique et de télécommunication

MEMOIRE MASTER ACADEMIQUE

Domaine : Électronique
Spécialité : Automatique

Présenté par :

MAHDI Youcef

TELLI Ala Edinne

Thème

**Conception et réalisation d'un système de
contrôle et de communication à base de la carte à
microcontrôleur STM32F7**

Soutenu publiquement

Le : 31/05/2016

Devant le jury :

Mr	BENHELLAL	MAA	Président	UKM Ouargla
Mr	HAMMOUCHI	MAB	Examineur	UKM Ouargla
Mr	HAMZA	MAB	Examineur	UKM Ouargla
Mr	TIDJANI Zakaria	MAA	Encadreur	UKM Ouargla
Mr	KAFI Abdellah Islam	Doctorant	Co-Encadreur	UFA Sétif

Année Universitaire : 2015 /2016

Remerciements

Premièrement, nous remercions dieu, le tout puissant, de nous avoir donné la santé et la volonté pour faire cet humble travail.

Nous tenons à exprimer nos vifs remerciements à nos deux encadreur M. KAFI Abdellah Islam et M. TIDJANI Zakaria pour l'aide et pour ses conseils précieux qu'ils nous ont données durant notre étude et la réalisation de ce travail.

Nos remerciements s'adressent ainsi aux: M. Le président et les membres de jury d'avoir accepté à juger et à évaluer notre travail.

Nous remercions aussi tous les professeurs, les enseignants, nos parents, nos frères, nos sœurs, nos chers amis, pour leurs aides et leurs sacrifices.

Dédicaces

Nous dédions cet humble travail.

A ceux qui ont nous aide à faire ce modeste travail par leurs

conseils et leurs encouragements :

Nos parents

Nos encadreur M. KAFI Abdellah Islam et

M. TIDJANI Zakaria

Nos professeurs

Nos chers amis

A toute la promotion 2016

Nous les remercions et les dédions ce travail

Sommaire

Sommaire

Introduction générale	1
Problématique, Cahier des charges et Proposition	3
Chapitre I : La carte STM32 et le module Sim900a	
I.1 Introduction	6
I.2 La carte STM32F4-DISCOVERY	6
I.2.1 Fonctionnalités	7
I.2 Logiciels de configurations et de programmation	9
I.2.1 STM32CubeMx	9
I.2.2 Environnement de développement Keil (MDK) V5	11
I.3 La carte STM32F746G-DISCO (32F746G DISCOVERY)	12
I.3.1 Microcontrôleur 32 bits avec FPU STM32F7	12
I.3.2 Kit de découverte STMicroelectronics 32F746GDISCOVERY	13
I.4 Module GSM Sim900a	14
I.4.1 Caractéristiques du module sim900a	15
I.4.2 La commande AT	15
I.4.3 Référence des commandes AT du modem GSM	16
I.5 Conclusion	17
Chapitre II : Conception et réalisation de la section contrôle	
II.1 Introduction.	18
II.2 Conception de la configuration matérielle	18
II.2.1 Alimentation	18
II.2.2 Adaptation des entrées	19
II.2.2.1 Entrées digitales	19
II.2.2.2 Entrées analogiques	20
II.2.3 Adaptation des sorties	22
II.3 Configuration et Programmation des fonctions nécessaires	24
II.3.1 Les entrées et les sorties digitales	24
II.3.2 Les sorties analogiques	26
II.3.3 Mise à l'échelle des entrées analogiques	28

II.4 Conclusion	29
Chapitre III : conception de la section IHM et communication	
III.1 Introduction	30
III.2 Schéma de visualisation et de communication	31
III.3 Le Module IHM	31
III.3.1 Mise en œuvre Matériel	32
III.3.2 Mise en œuvre Logiciel	33
III.3.2.1 Le cadre de Logiciel (TouchGFX)	33
III.4 Module de Communication Modbus /TCP	34
III.4.1 Protocole Modbus	35
III.4.2. Protocole Modbus/TCP	37
III.4.3 Mise en œuvre Matériel	38
III.4.4 Mise en œuvre logiciel	39
III.5 Module de Mémoire	41
III.5.1 Mise en œuvre Matériel	41
III.5.2 Mise en œuvre Logiciel	42
III.6 La Communication par le module Sim900a	44
III.7 Conclusion	46
Chapitre IV : Réalisation et Tests	
IV.1 Introduction	47
IV.2 La réalisation de la carte	47
IV.2.1 Réalisation du circuit	47
IV.2.2 Réalisation du typon (PCB)	48
IV.2.3 Réalisation de la circuit et intégration des composants	49
IV.3 Les tests expérimentaux	50
IV.3.1 Test 01 : Entrée digitale 0~5 VDC	50
IV.3.2 Test 02 : Entrée analogique (0-5VDC)	51
IV.3.3 Test 03 : Communication à distance	52
IV.3.4 Test 04 : la communication via Modbus/TCP	53
IV.4 Conclusion	54
Conclusion générale	55
Bibliographies	56

La liste de figure

La liste des figures

Figure(1) : Le schéma de principe du système	05
Chapitre I : la carte STM32 et le module Sim900a	
Figure (I.1) : La carte STM32F-DESCOVERY.	07
Figure (I.2) : Page d'accueil du STM32CubeMx	10
Figure (I.3) : La fenêtre de nouveau projet.	11
Figure (I.4) : la fenêtre principale de STM32CubeMX	11
Figure (I.5) : Page principale de l'éditeur/ compilateur Keil MDK v5	12
Figure (I.6) : La carte STM32F746G-DISCO.	13
Figure (I.7): Module Sim900a.	14
Figure (I.8) : schéma de fonctionnement de la transmission via module de communication	16
Chapitre II : conception et réalisation de section contrôle	
Figure (II.1) : schéma du circuit d'adaptation de l'entrée digitale 0~24VDC	19
Figure (II.2) : schéma d'adaptation de l'entrée digitale 0~5VDC	19
Figure (II.3) : schéma d'entrée 4-20 mA analogique	20
Figure (II.4) : Schéma d'interface de l'entrée analogique 0-10V	21
Figure (II.5) : schéma d'entrée 0-5V analogique	21
Figure (II.6) : Schéma de la sortie digitale 24VDC	22
Figure (II.7) : Schéma de la sortie digitale 5VDC	23
Figure (II.8) : La configuration d'entrée et de sortie et la mode d'interruption en STM32CubeMx	24
Figure (II.9) : La configuration d'entrée et de sortie et la mode d'interruption en Keil	25
Figure (II.10) : Configuration du Timer2 en keil v5.	26
Figure (II.11) : Choix des pins et les canaux de l'ADC1 en STM32 CubeMx.	27
Figure (II.12) : Configuration de l'ADC en STM32 CubeMx.	27
Figure (II.13) : Les fonctions d'initialisation et d'appel de l'ADC1 en Keil v5.	28
Figure (II.14) : Fonction de mise à l'échelle	29
Chapitre III : conception de la section IHM et communication	
Figure(III.1) : Schéma de visualisation et de communication.	31
Figure (III.2) : programme d'affichage du entre digitale.	34
Figure (III.3) : Principe de communication Client/ serveur.	35
Figure (III.4) : Trame de requête Modbus.	35

Figure (III.5) : Trame de réponse Modbus.	36
Figure (III.6) : Trame MODBUS/TCP	37
Figure (III.7) : Connecteur RJ45 Ethernet CN9 (vue de face).	38
Figure (III.8) : Diagramme de communication Modbus TCP.	41
Figure (III.9) : diagramme d'interaction avec la mémoire flash.	43
Figure (III.10) : Programme de transmission des données vers le serveur web.	44
Figure (III.11) : Organigramme générale de la supervision	45
Figure (III.12) : Schéma de transfert et d'intégration des données dans le serveur.	46
Chapitre IV : Réalisation et Tests	
Figure(IV.1) : La carte réalisée sur ISIS Proteus	47
Figure(IV.2) : Typon du circuit imprimé	48
Figure(IV.3) : La carte d'interfaçages finale avec les composants	49
Figure(IV.4) : Assemblage final de la carte de commande et de communication	50
Figure (IV.5) : Test de l'entrée digitale 0~5VDC.	51
Figure (IV.6) : Test de l'entrée analogique 4-20mA.	52
Figure (IV.7) : Test de la transmission à distance	52
Figure (IV.8) : Test de la communication en Modbus TCP/ écriture.	53
Figure (IV.9) : Test de la communication en Modbus TCP.	54

La liste de tableau

La liste des tableaux

Chapitre I : La carte STM32 et le module sim900a

Tableau I.1 : Caractéristiques principales du module Sim900a 15

Tableau (I.2) : Principales commandes du langage AT 17

Chapitre II : conception et réalisation de section contrôle

Tableau (II.1) : valeurs des résistances calculés et résistances utilisées 23

Tableau (II.2) : Les fonctions de lecture est d'écritures d'E/S digitales 25

Chapitre III : conception de la section IHM et communication

Tableau (III.1) : La configuration des paramètres des entrées et de sorties. 32

Tableau (III.2) : les signaux d'interface de LTDC. 33

Tableau (III.4) : Les fonctions de Modbus. 36

Tableau (III.5) : Encapsulation de la trame MODBUS TCP/IP. 38

Tableau (III.6) : Attribution des pins d'Ethernet TCP sur la carte STM32F7 39

Tableau (III.7) : Quelques fonctions utilisées dans le programme de Modbus TCP. 40

Tableau (III.8) : les paramètres de configuration. 40

Tableau (III.9) : Espace de la mémoire flash de la carte STM32F7 42

Tableau (III.10) : Quelques fonctions du pilote HAL associé à la mémoire flash. 43

Introduction générale

Introduction général

La maîtrise des procédés industriels de production exige l'acquisition de grandeurs physiques des différentes natures telle que les températures, les débits, les pressions, les concentrations, les vitesses, des concentrations chimiques ...etc. Des instruments de mesures et de contrôle réalisent la collecte de ces données et leur conversion en signaux standards exploitable par la majorité des plateformes de contrôle existante.

La majorité de ces instruments délivre des signaux de nature électrique (courant et tension) image des données d'origine. Certains nombre de ces instruments, en particulier, ceux basées sur des microcontrôleurs, sont doté de fonctionnalité de communication en réseau industriel local.

Dans plusieurs applications, l'exploitant nécessite la collecte de donnée de contrôle à partir de plusieurs sites éloignés les uns des autres pour mieux gérer une installation distribuée.

Dans ce travail, nous envisageons le développement d'une carte de contrôle et de communication permettant à la fois l'acquisition des données de mesure, la commande et l'échange des données de contrôle à travers un réseau industriel local. En plus, cette carte doit être en mesure de transmettre des données supposées critiques vers des serveurs distants à travers un réseau de communication existant.

Au départ, nous allons développer la problématique, établir un cahier de charge et par la suite proposer une solution.

Ce qui vient dans ce, le mémoire sera organisé comme suit :

Le premier chapitre présentera les deux principaux modules nécessaires à la réalisation de la carte à savoir : La carte STM32F ainsi que ses principales fonctionnalités liée au projet et le module de communication à travers le réseau GSM/GPRS (Sim900a).

Dans le deuxième chapitre, nous allons développer l'interface d'acquisition des entrées digitales et analogiques et d'adaptation des sorties digitales. Le programme associé sera aussi élaboré.

La section d'interface homme machine (IHM) et les modules de communication soit locale (Modbus/TCP) ou distant (Sim900a) seront examinés dans le troisième chapitre. Ainsi, l'aspect théorique, matériel et logiciel seront présentés.

Le quatrième chapitre sera dédié à la réalisation et au test de validation de la carte de contrôle et de communication développée.

A la fin, nous terminons par une conclusion.

Problématique, Cahier des charges et Proposition

a- Problématique

Dans les procédés industriels, les instruments de contrôle et de commande nécessitent l'acquisition et le traitement de grandeurs digitales et analogiques standardisés mais selon différentes normes. Ainsi, nous trouvons des signaux tout ou rien ou analogique, souvent de nature tension électrique tels que 0-5VDC, 0-10VDC 0-24VDC, 0-48VDC, 0-24VAC, 0-48VAc et même 0-120VAC.

Dans le but de réduire l'influence des longueurs des câbles et des champs électromagnétiques envoisant, d'autres instruments dédiés aux mesure des grandeurs analogiques délivrent des signaux électriques mais en courant standards. Les normes de mesure en courant recommandent les standards suivants : 0-20mA, 4-20mA ou 0-10mA.

D'autre part, la tendance actuelle vise à intégrer les instruments de caractère intelligents (Smart instrument). Basés sur des microprocesseurs et des microcontrôleurs, ces instruments ont la capacité de communiquer en réseau pour l'échange de données. Ainsi, ce type de communication numérique permettra en plus de véhiculer d'autre information que les mesures telles que des paramètres de configuration et/ou d'activation. Dans l'industrie, nous trouvons plusieurs protocoles de communication industrielle tels que Modbus, DeviceNet, Profibus et autres.

Certains procédés industriels sont composés de plusieurs unités de production situées dans des zones relativement éloignées. Si une supervision ou même un échange de données de commande entre les unités est nécessaire, nous devons réaliser un réseau de communication distante.

Le problème que nous allons traiter dans ce travail c'est la conception et la réalisation d'une carte de contrôle et de communication pour un procédé industriel. Cette carte doit remédier aux contraintes des normes d'interfaçage et de communication tels que indiqués ci-dessus.

b- Cahier des charges

La considération de toutes les contraintes statués en haut est une tâche difficile vu leur nombres et leur nature. Pour prendre en charge les fonctions principales avec certaine norme très utilisées, nous avons établis un cahier de charges réduit.

La carte de contrôle et de communication à concevoir et à réaliser doit répondre aux contraintes suivantes :

- 1- L'acquisition des entrées digitales de type 0~5VDC, 0~10VDC et 0~24VDC
- 2- L'acquisition périodique des entrées analogiques de type 0~5VDC, 04-20mA avec une résolution de 12bits.
- 3- La mise à l'échelle des entrées analogiques selon l'échelle de l'instrument de mesure.
- 4- Sauvegarde des paramètres de contrôle jugés importants en mémoire flash interne.
- 5- Affichage des mesures digitales et analogiques sur interface homme machine.
- 6- Affectation de sorties digitales 0~24VDC.
- 7- Communication en réseau avec le protocole Modbus/TCP. La carte développée sera configurée en tant que maitre.
- 8- Transmission des données de contrôle vers un serveur web distant à travers le réseau GSM/GPRS.

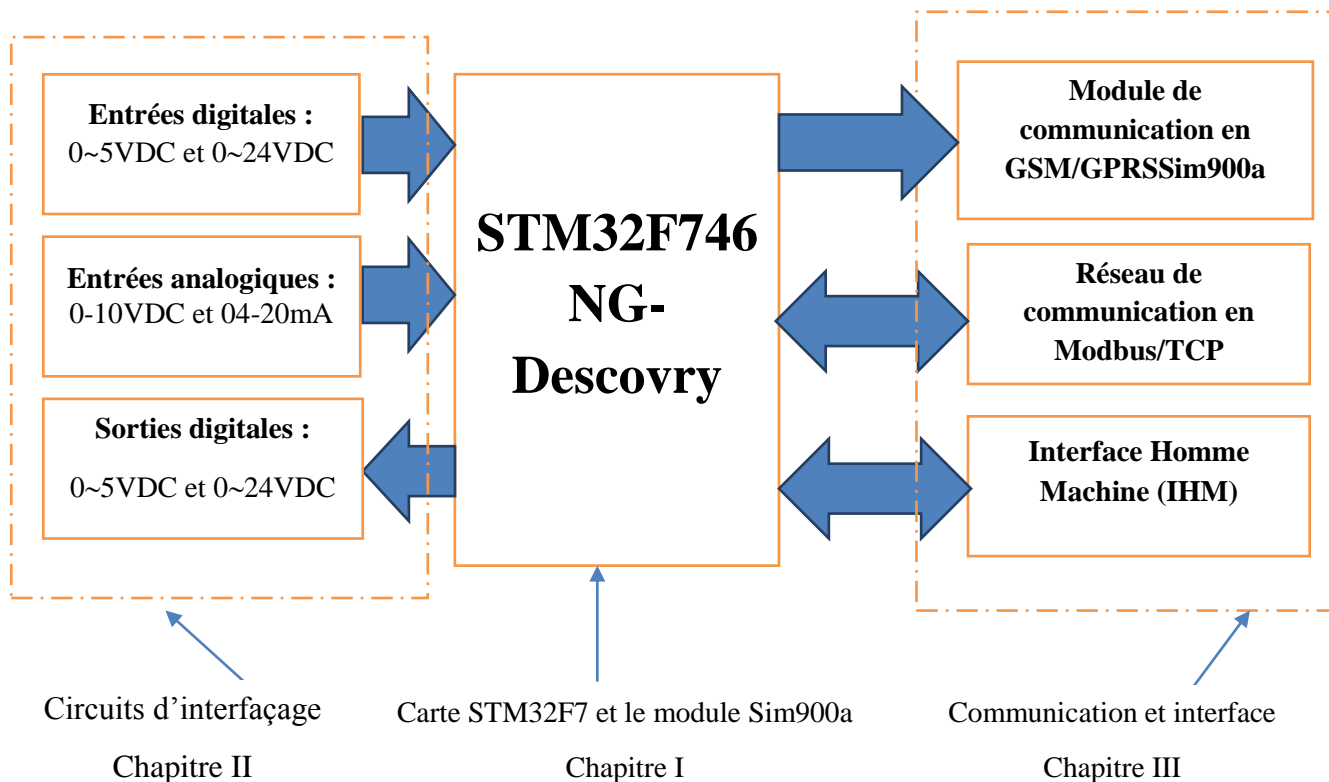
c- Proposition de la solution

Dans le but de réaliser cette carte, nous proposons d'utiliser la carte STM32F746NG-Discovery. Cette carte est dotée de plusieurs fonctions d'acquisition et de contrôle des variables digitales et analogiques. Elle assure la communication en mode série et dispose de port Ethernet.

Cette carte exige des entrées à faible niveau de tension (de l'ordre de 3.3VDC) ainsi, en premier lieu nous devons concevoir un circuit d'interfaçage adéquat.

En deuxième lieu, la mise en réseau et la communication nécessite un modem qui assurera la transmission des données à travers un réseau GSM/GPRS. Ainsi, nous proposons d'utiliser le module Sim900a. Le transfert de données entre ce module et la carte sera établi via le langage AT. Cependant, la trame à transmettre vers le serveur doit être conforme au protocole http.

La figure suivante illustre le schéma de principe du système.



Figure(1) : Le schéma de principe du système

Chapitre I :

La carte STM32 et le module Sim900a

I.1 Introduction

Le développement technologique a connu la création de plusieurs cartes électroniques à base de microprocesseur ou microcontrôleurs. Pour incorporer ces cartes dans des projets industriels, l'utilisateur amateur ou professionnel aura comme mission principale l'adaptation de l'interface entrées sorties et la programmation des tâches à réaliser.

STMicroelectronics est une société de semi-conducteurs internationaux indépendants et un leader pour le développement et la réalisation de solutions destinées à un grand nombre d'applications microélectroniques. Son expertise du silicium et des systèmes, sa puissance industrielle, son portefeuille de propriétés intellectuelles et ses alliances stratégiques placent STMicroelectronics à l'avant-garde des technologies de systèmes sur puce [1].

STMicroelectronics fournit une vaste gamme des cartes électronique (La famille des microprocesseurs STM32FXXX), parmi ces cartes, nous citons la STM32F7NG-*DISCOVERY* développée autour du microcontrôleur ARM Cortex-M7 qui constitue l'élément central de notre projet.

Dans ce chapitre, nous allons examiner les principales caractéristiques et fonctionnalités de cette carte ainsi que celle du module GSM 901.

I.2 La carte STM32F4-*DISCOVERY* :

Afin de permettre la création d'un large éventail d'applications, le kit de découverte STM32F7 est fourni avec la vaste bibliothèque de logiciels embarqués (firmware) STM32Cube. La plateforme matérielle du kit de découverte permet d'accéder aux fonctionnalités du microcontrôleur STM32F7 grâce à une conception compacte. Ce design inclut un écran en couleurs tactile WQVGA, ainsi que la prise en charge de plusieurs capteurs et des fonctions audio stéréo, de sécurité et de connectivité haut débit.

La société STMicroelectronics a lancé deux gammes de produits STM32F7 : les séries STM32F745 et STM32F746/756 qui disposent du cœur Cortex-M7 avec unité de calcul en virgule flottante et extensions DSP et dont la fréquence de fonctionnement atteint 216 MHz. Ces microcontrôleurs embarque jusqu'à 1 Mo de mémoire flash, ainsi que 320 ko de mémoire RAM. [1]

Ces cartes sont dotées du simple GPIO (port d'entrée-sortie généraliste) et interface de communication série synchrone (SPI) ou asynchrone (RS485) aux interfaces aussi complexes que l'USB, Ethernet ou HDMI...etc.

La carte de développement STM32F4-DISCOVERY est illustrée par la Figure (I.1).

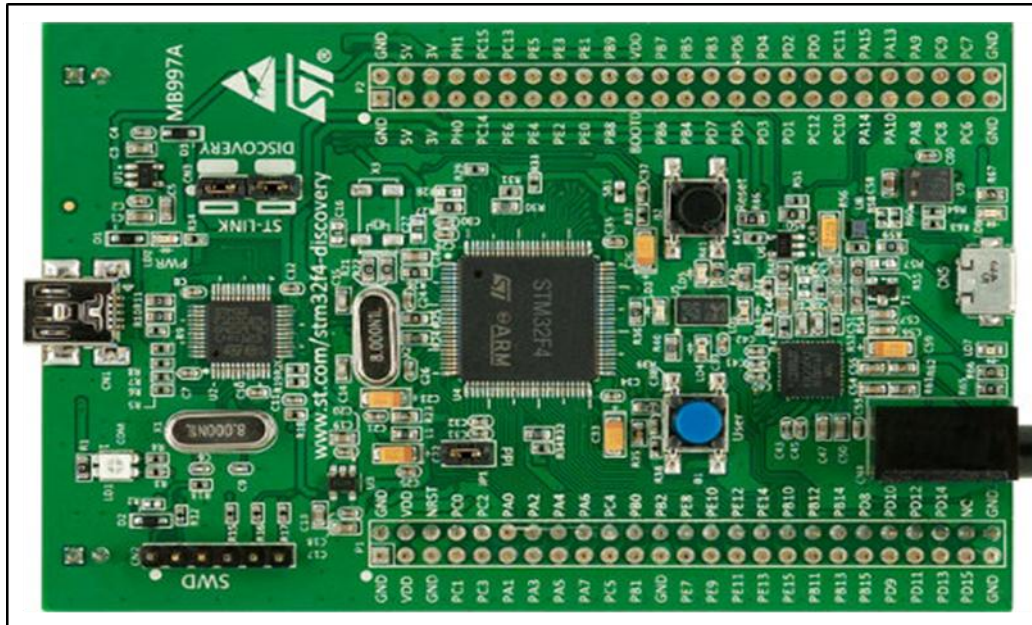


Figure (I.1) : La carte STM32F-DESCOVERY.

I.2.1 Fonctionnalités

STM32F4-DESCOVERY présente plusieurs fonctions d'acquisitions, de traitement et de communication. Ici, nous présenterons seulement les trois fonctionnalités en liaison avec notre projet qui sont :

a-Entrées et sorties à usage général (GPIO : General Purpose Input Output)

Le STM32 est bien servi en broches E/S à usage général, ayant typiquement 80 broches E/S bidirectionnels à un niveau de tension 3.3V. Ces broches sont disposées en cinq ports ayant chacun 16 lignes E/S. Ces ports sont nommés A, B, C, D et E et sont tolérant à 5V. Un grand nombre des broches externes peut être commuté d'usage général E/S pour servir d'Entrée / Sortie d'un périphérique d'utilisateur, par exemple un USART ou périphérique Ethernet.[2]

Il existe une unité d'interruption externe qui permet aux 16 lignes d'interruption externe à être mis en correspondance avec une combinaison quelconque de lignes GPIO. Les caractéristiques principales de la GPIO sont :

- ✓ Jusqu'à 16 E / S sous contrôle.
- ✓ Un multiplexage très flexible des broches, par une simple configuration, permet l'utilisation de broches d'E/S comme GPIO ou comme un de plusieurs fonctions périphériques
- ✓ Attribution séparée des vitesses pour les E/S.

b- Convertisseur analogique numérique (ADC : Analogue Digital Convertir)

La carte STM32F4-DISCOVERY possède 3 modules de conversion ADC. Elle utilise la méthode d'approximations successives avec une résolution jusqu'à 12 bits. Elle a jusqu'à 19 canaux multiplexés permettant de mesurer des signaux à partir de 16 sources externes, deux sources internes. La conversion A / N des canaux peut être effectuée en mode Singulier, continu ou avec balayage discontinu. Le résultat de l'ADC est stocké dans un registre de donnée dédiée à 16 bits [2]. Nous résumons les principales caractéristiques comme suit :

- ✓ Résolution configurable à 12 bits, 10 bits, 8 bits ou même 6 bits.
- ✓ Mode de conversion simple et continu.
- ✓ Option de déclenchement externe avec une polarité configurable pour les deux ordinaire, et injecté.
- ✓ Demande de conversion ou d'accès à la donnée en mode DMA (Direct Memory Access).
- ✓ Cadence sur un résonateur interne ou sur un quartz externe haute fréquence 16 MHz.

c- USART (Universal Synchronous Asynchronous Receiver Transmitter)

L'émetteur récepteur asynchrone série universelle (USART) offre un moyen flexible d'échange de données bidirectionnel en full-duplex avec un instrument externe équipé de fonction de asynchrone. Le codage le plus utilisé est celui du non-retour à zéro (NRZ). Cette interface offre une très large gamme de vitesses de transmission. La communication de données à grande vitesse est possible en utilisant le DMA pour la configuration multi-espace.

Sur la carte STM32F4-DISCOVERY, l'affectation de la fonction de communication en USART est réalisée via l'activation de ces fonctionnalités disponible sur deux ports prédéfinies. [2]

I.2 Logiciels de configurations et de programmation :

La mise en œuvre d'applications basées sur la carte STM32F4-DISCOVERY ne peut s'effectuer sans passer par les phases de configuration et de programmation. Le fabricant mis à notre disposition le software STM32CubeMx pour la configuration. Cependant, pour la programmation des tâches à exécuter plusieurs softwares de compilations sont développées par différents concepteurs qui sont disponible dans le marché. Dans notre travail, nous avons opté pour le compilateur Keil (MDK) V5.

I.2.1 STM32CubeMx :

STM32Cube™ est un outil logiciel de conception de microcontrôleurs STM32. Disponible gratuitement, cette puissante plate-forme de développement simplifie et accélère les projets des clients.

La plate-forme de développement STM32Cube™ comprend le configurateur graphique STM32CubeMX, un générateur de code C pour initialisation qui guide les utilisateurs étape par étape, et un riche ensemble de composants logiciels embarqués qui évitent d'intégrer des logiciels provenant de sources multiples. Le logiciel comprend une nouvelle couche d'abstraction matérielle (HAL) qui simplifie la migration ou le transfert entre deux microcontrôleurs STM32.

En réunissant sur un seul boîtier tous les composants logiciels génériques nécessaires pour développer des applications sur microcontrôleurs STM32, cette plate-forme élimine la tâche complexe que représente l'évaluation des liens de dépendance entre les différents composants logiciels.

Les principales caractéristiques du STM32CubeMX sont :

- ✓ Sélection de microcontrôleur facile couvrant la famille STM32 entière.
- ✓ Sélection de la carte à partir d'une liste des cartes de STMicroelectronics.
- ✓ La facilité de la configuration du microcontrôleur (broches, arbre d'horloge, périphériques, middleware) et la génération du code d'initialisation correspondant en langage C.

- ✓ Génération de rapports de configuration. [1]

Les principales pages de configuration dans le software STM32CubeMX sont :

a- Page d'accueil de STM32CubeMX :

La page d'accueil est la première fenêtre qui s'ouvre au lancement du programme STM32CubeMX. Il reste ouvert aussi longtemps que la demande est en cours d'exécution. Fermeture qu'il referme l'application. La page d'accueil représenté dans la Figure(I.2) suivante.

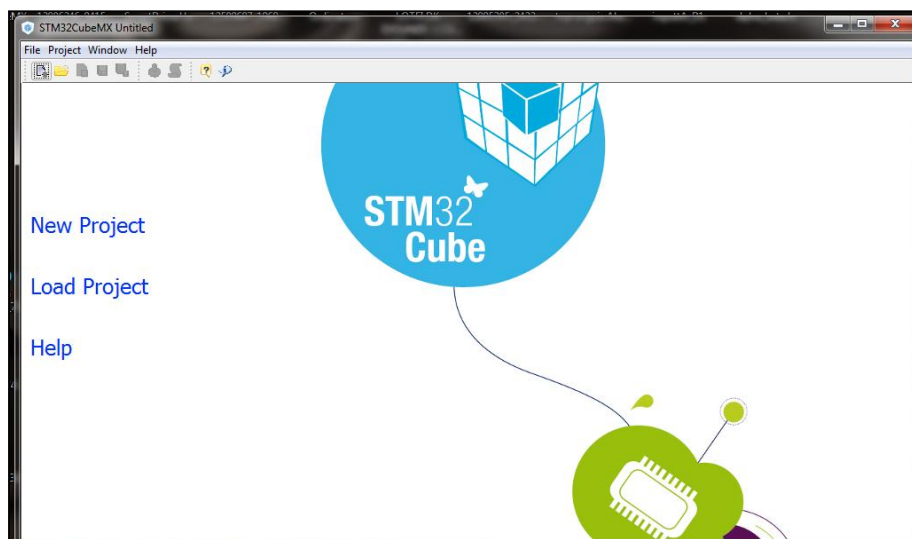


Figure (I.2) : Page d'accueil du STM32CubeMX

b- La fenêtre de nouveau projet : Cette fenêtre affiche deux onglets au choix:

- L'onglet de sélection MCU proposant une liste de processeurs cible, voir (1) dans la figure (I.3).
- Un onglet Sélecteur conseil montrant une liste de conseils STMicroelectronics (2) dans la figure (I.3)

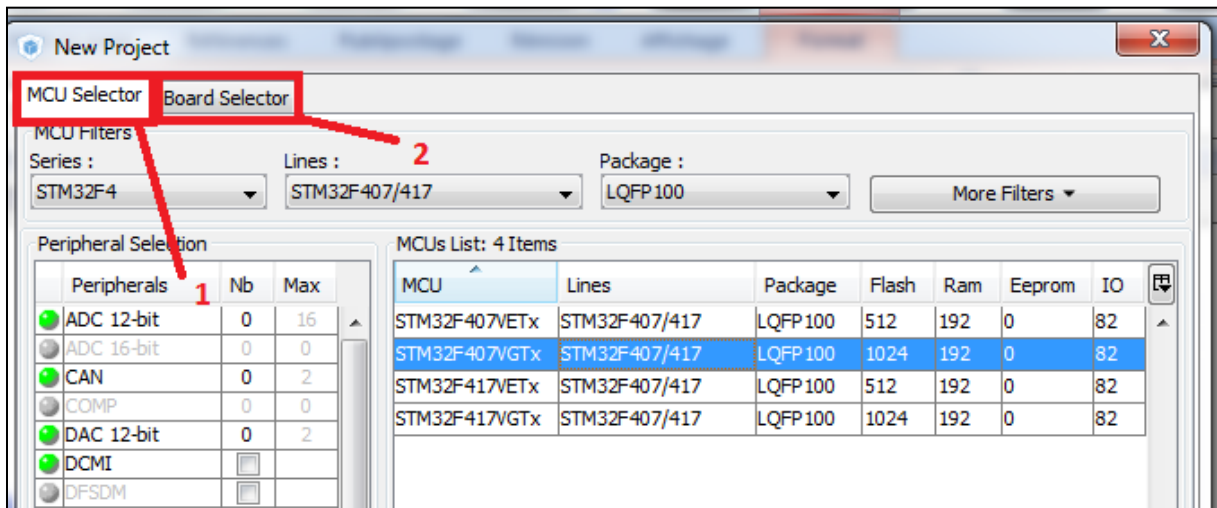


Figure (I.3) : La fenêtre de nouveau projet.

c- La fenêtre principale : la fenêtre principale affiche tous les composants et les menus du STM32CubeMX, comme montré dans la figure (I.4).

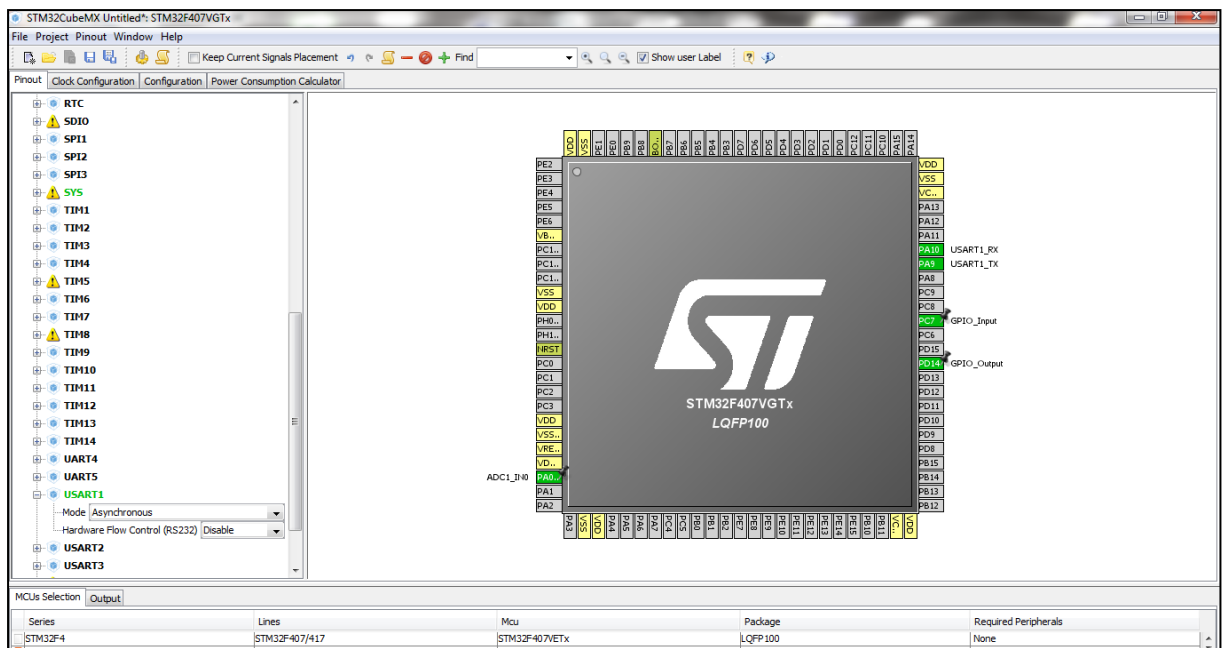


Figure (I.4) : la fenêtre principale de STM32CubeMX.

I.2.2 Environnement de développement Keil (MDK) V5 :

Le kit de développement de microcontrôleur Keil (MDK) nous aide à créer des applications embarquées pour les microcontrôleurs ARM basés sur le processeur Cortex-M. Le MDK est outil puissant en programmation ou en compilation. La version 5 se

compose du Core MDK, plus Packs de logiciels spécifiques à l'appareil, qui peut être téléchargé et installé sur la base des exigences de votre application. La figure (I.5) montre la page principale de cet outil.[3]

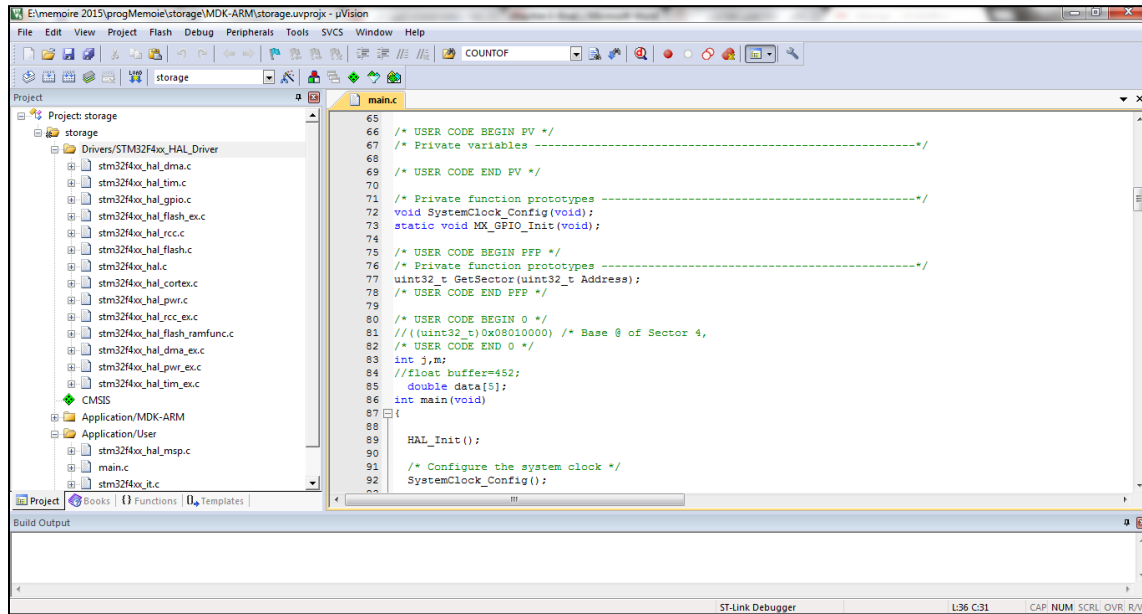


Figure (I.5) : Page principale de l'éditeur/ compilateur Keil MDK v5.

I.3 La carte STM32F746G-DISCO (32F746G DISCOVERY):

La carte STM32F746G-DISCO (32F746G DISCOVERY) est une plate-forme de démonstration et de développement complet pour microcontrôleur STM32F746NGH6 base-core STMicroelectronics ARM® Cortex®-M7.

I.3.1 Microcontrôleur 32 bits avec FPU STM32F7 :

Les microcontrôleurs 32 bits avec unité de calcul en virgule flottante (FPU) STM32F7 de STMicroelectronics sont basés sur le cœur haute performance ARM® Cortex®-M7 32 bits RISC, dont la fréquence de fonctionnement atteint 216 MHz. Le cœur Cortex-M7 inclut une unité de calcul en virgule flottante en simple précision (SFPU). Il intègre également une série complète d'instructions DSP ainsi qu'une unité de protection de mémoire (MPU) qui améliore la sécurité de l'application.

Parmi les caractéristiques contient dans microcontrôleur :

- ✓ Une mémoire cache L1 : mémoire cache de données de 4 Ko et mémoire cache d'instruction de 4 Ko.
- ✓ Jusqu'à 1 Mo de mémoire Flash.
- ✓ Quatre USART.
- ✓ Trois CAN (convertisseur analogique numérique) 12 bits, 2.4 MSPS : jusqu'à 24 canaux et 7.2 MSPS en mode triple canal entrelacé.
- ✓ Jusqu'à 168 ports avec capacité d'interruption.
- ✓ Alimentation d'application 1,7 V à 3,6 V et E/O.[1]

I.3.2 Kit de découverte STMicroelectronics 32F746GDISCOVERY :

Le kit de découverte STMicroelectronics 32F746GDISCOVERY est conçu pour découvrir et exploiter la famille de microcontrôleurs STM32F7 basée sur le cœur ARM Cortex-M7. Il permet de développer et de partager des applications et prend en charge les lecteurs audio/vidéo et enregistreurs audio, ainsi que les alarmes. Il est également possible de concevoir des HMI pour bénéficier de toutes les fonctions audio, vidéo et d'écran tactile couleur.[1]

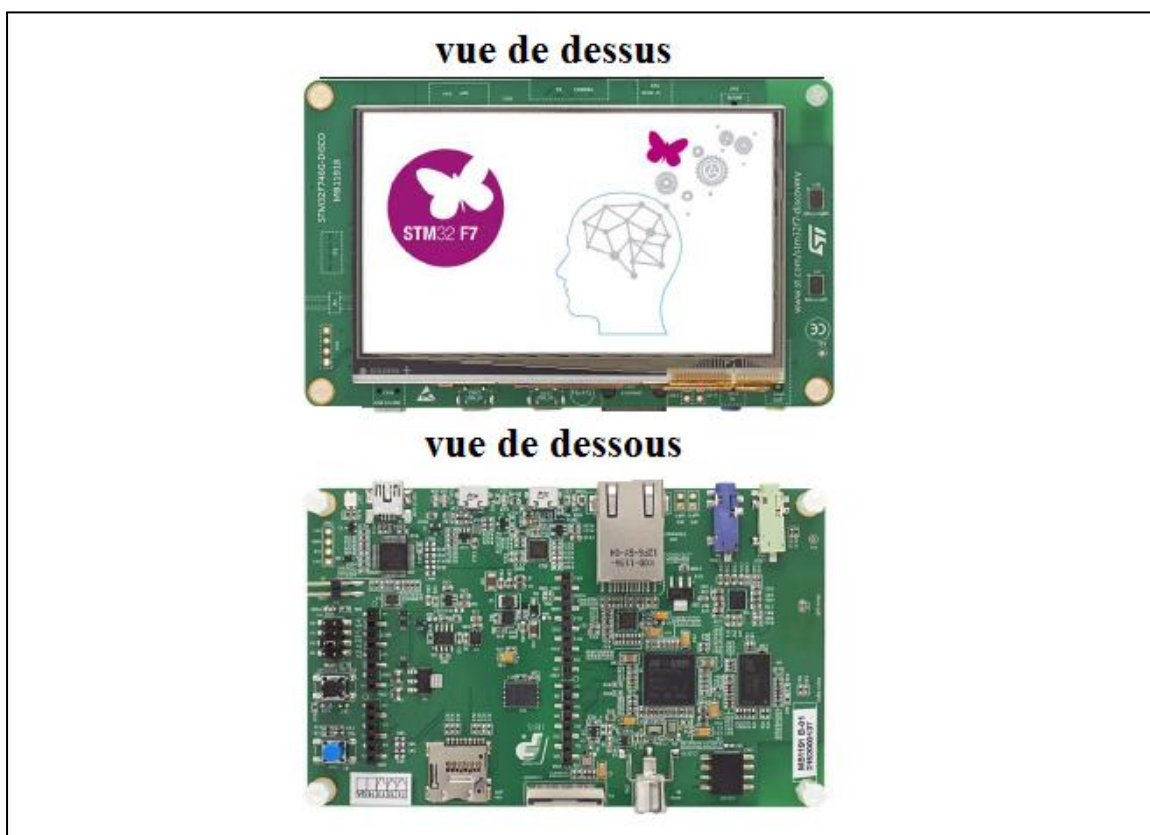


Figure (I.6) : La carte STM32F746G-DISCO.

Caractéristiques :

- ✓ Microcontrôleur STM32F746NGH6 avec 1 Mo de mémoire Flash et 340 Ko de RAM.
- ✓ Ecran tactile capacitif LCD-TFT couleur 4.3" 480x272.
- ✓ Connecteur pour carte micro SD.
- ✓ Connecteur Ethernet conforme à IEEE-802.3-2002.
- ✓ Sortie d'alimentation pour applications externes : 3,3 V ou 5 V.

I.4 Module GSM Sim900a:

Le célèbre modem de communication SIM900A, de SIMCOM, permet d'intégrer la fonction de communication sans fil, via les fréquences 900/1800 MHz à travers les bandes GSM ou GPRS. Le modem est alimenté avec une tension d'interface sélectionnable, ce qui vous permet de connecter 5V et 3.3V sans nécessité de conversion de niveau. La vitesse de transmission est configurable 9600-115200 bits/s par commande selon le protocole AT.

Ce Modem GPRS/ GSM est équipé d'une pile TCP / IP interne pour vous permettre de communiquer avec Internet via GPRS. Il est adapté pour les SMS, la voix ainsi que les demande de transfert de données dans l'interface M2M(Machine to Machine). La figure (I.7) montre le module de communication Sim900a.[4]



Figure (I.7) : Module Sim900a.

I.4.1 Caractéristiques du module sim900a :

Le module sim900a incorpore des solutions qu'on peut créer à rendement élevé grâce à son architecture qui est composée principalement d'un microcontrôleur de bande de base, le circuit complet de fréquence radio compris un amplificateur de fréquence et un émetteur récepteur des fréquences radio, d'une autre part il nous offre une flexibilité pour l'intégration facile avec l'interface homme-machine dont il dispose de deux interfaces série (Tx-Rx). Ce type d'interface sera utilisé dans notre projet pour développer le transfert de donnée de la carte de commande vers ce module pour qu'elle soit envoyée vers un serveur de données sur internet. Le tableau résume quelque caractéristique du module Sim900a.[4]

Tableau I.1 : Caractéristiques principales du module Sim900a

Tension d'entrée	5 – 12 DC
Bandes de fréquence	GSM 900 MHz, GSM 1800 MHz, GSM 1900 MHz
Puissance de transmission	Classe 4 (2w) à GSM900 MHz Classe 1 (1W) à GSM1800 MHz
Commande	Des commandes AT (GSM 07.07, 07.05 et SIMCOM améliorée Commandes AT)
Connecteur	connecteur SMA avec antenne GSM. Connecteur interface audio Connecteur carte SIM
Température normale	-20°C +55°C

I.4.2 La commande AT :

La plupart des modems modernes disposent d'un jeu de commandes textuelles qui peuvent être appelées en mode de commande. Les modems plus anciens ne disposent que des commandes par signaux de contrôle ou de commandes sous la forme de télégrammes.

Le jeu le plus courant de commandes textuelles est appelé le jeu **AT** ou encore **Hayes**, du nom de l'entreprise ayant commercialisé les premiers modems dits intelligents. Le jeu se compose de commandes simples préfixées par AT (toujours présent pour commencer une ligne de commande). En général, ces deux codes sont utilisés pour synchroniser le modem à la bonne vitesse. Il est donc recommandé de les taper avec la même casse. Les commandes agissent sur des registres internes qui sont manipulables directement. Il est souvent possible

de sauvegarder la configuration dans une mémoire interne non volatile. Il faut savoir que la plupart du temps beaucoup d'extensions incompatibles ont été implantées: il n'existe pas à proprement parler de norme. On parlera cependant des commandes les plus courantes.[5]

Dans le Figure (I.8), illustre le schéma de fonctionnement d'une requête AT. Un équipement terminal TE (ici la carte de contrôle STM32F7) envoie une requête de transmission formulée en AT vers l'adaptateur terminal TA (le module Sim900a). Différentes liaisons sont disponibles telles que liaison série, infra rouge ou autre. Cet adaptateur transmet par ondes radio la requête vers l'équipement destinataire ou Mobile ME (le serveur de données) pour qu'elle soit exécutée. Des messages d'état, de réponses et de confirmation sont échangés entre les différents terminaux.

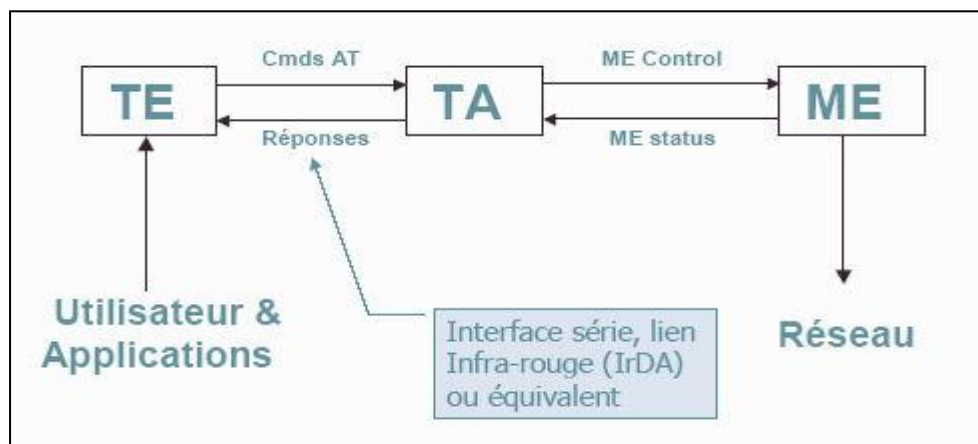


Figure (I.8) : schéma de fonctionnement de la transmission via module de communication.

I.4.3 Référence des commandes AT du modem GSM :

Les commandes AT sont des commandes que l'on peut directement envoyer au modem, lorsque celui-ci est en mode command. Chaque commande est envoyée sous la forme d'une ligne de texte encodée en ASCII, terminée par le caractère `\r` seul (code ASCII 13). Le modem retourne une réponse sous la forme d'une ou de plusieurs lignes selon la commande envoyée, chaque ligne se termine par les caractères `\r` suivi de `\n` (codes ASCII 13 et 10).

Toutes les commandes doivent être précédées d'AT, par exemple, ATA signifie "répondre". Certaines commandes, comme ATA, sont communes à tous les modems, d'autres sont variables selon le modèle spécifique, comme l'affichage de la configuration peut varier

selon l'équipement. Il faut donc se reporter à la documentation de son modem, dans le tableau suivant les commandes AT dédiées au GPRS (server web).

Tableau (I.2) : Principales commandes du langage AT

Fonction	Discrétion
<AT+CGATT?\r\n >	la vérification de la connexion GPRS
<AT+SAPBR=3,1,"CONTYPE\","GPRS"\r\n>	La configuration de mode GPRS
< AT+SAPBR=3,1,"APN\","APN"\r\n>	La configuration d'APN
<AT+SAPBR=1,1\r\n>	La fin de configuration de la mode GPRS
<AT+HTTTPARA="URL">	définit les paramètres du protocole HTTP

En suite la trame de HTTPARA :

```
<AT+HTTTPARA="URL","lien de site web? donnee1& donnee12& donnee3">
```

I.5 Conclusion

La carte STM32F7 regroupe les principales fonctions d'acquisition, de traitement de signaux et de stockage avec une grande flexibilité de sélection et de configuration. Les convertisseurs analogiques numériques à 12 bits permettent l'acquisition avec une bonne précision des signaux analogiques. Le port de communication Ethernet permet l'intégration matérielle directe de la carte dans un réseau d'instruments ou d'équipement. La mémoire flash fournie un large espace pour le stockage des paramètres de configuration de la carte.

Le module Sim900a peut être couplé avec la carte Stm32F7 à travers une liaison série. A ce moment, nous sauront en mesure d'envoyer des données à travers le réseau GSM ou GPRS vers un serveur distant.

Chapitre II :

Conception et réalisation de la section contrôle

II.1 Introduction

La carte STM32F7 utilise par défaut des signaux d'entrée et de sortie digitales dont le niveau haut est à 3.3VDC. Cependant, en standard, la majorité des instruments de mesure et de commande sont à des niveaux différents tels que 5VDC, 24 VDC, De même les signaux analogiques de mesures en courant 04-20mA nécessite d'être adaptés au niveau de tension de références des convertisseurs analogiques numériques de la carte qui sont aussi à 3.3VDC.

Dans ce chapitre, nous allons concevoir le circuit d'interface des entrées et des sorties de la carte STM32F7 pour qu'elles soient compatibles et pour assurer la protection de la carte par isolation des tensions de niveau ou de puissance élevées. Nous allons aussi développer la configuration et la programmation des fonctions d'acquisition dans les logiciels dédiés STM32CubeMx et keil.

II.2 Conception de la configuration matérielle

Pour concevoir la carte de contrôle avec les fonctionnalités du cahier de charges, nous allons considérer les sections principales suivantes : Alimentation, Adaptations des entrées, adaptation des sorties.

II.2.1 Alimentation

L'alimentation de la carte de contrôle doit assurer des tensions de plusieurs niveaux pour alimenter les différents module et composants. La carte Stm32F7 ainsi que la module de communication GSM nécessitent une alimentation et des niveaux d'entrées 3.3VDC quant aux amplificateurs opérationnels, nous utiliserons des tensions de 12VDC.

D'autre part, si nous prévoyons une éventuelle l'alimentation des relais associés aux entrées et aux sorties nous aurons besoins des tensions 5VDC, 10VDC, 24VDC ayant une puissance suffisante.

Ainsi, nous avons trouvé qu'une très bonne solution sera d'utiliser une alimentation d'un ordinateur de bureau (PC). Elle délivre les différentes tensions nécessaires à notre carte de contrôle à savoir (3.3VDC, 5VDC, 12VDC et -12VDC). Ce qui est intéressant c'est la puissance élevée que procure ce type d'alimentation.

II.2.2 Adaptation des entrées

Pour acquérir les différents types d'entrées, nous devons les adapter au niveau des tensions de la carte 3.3VDC en digitale ou en analogique.

II.2.2.1 Entrées digitales

a- **L'entrée 0~24VDC** : Nous utilisons le relais RL1 pour l'isolation et l'adaptation de la tension à 3.3VDC. La diode de roue libre D1 protège la bobine du relais. Cette entrée est affectée au pin PB14 de la carte Stm32F7 comme montre le schéma de la figure (II.1). La LED témoins indique d'état de l'entrée.

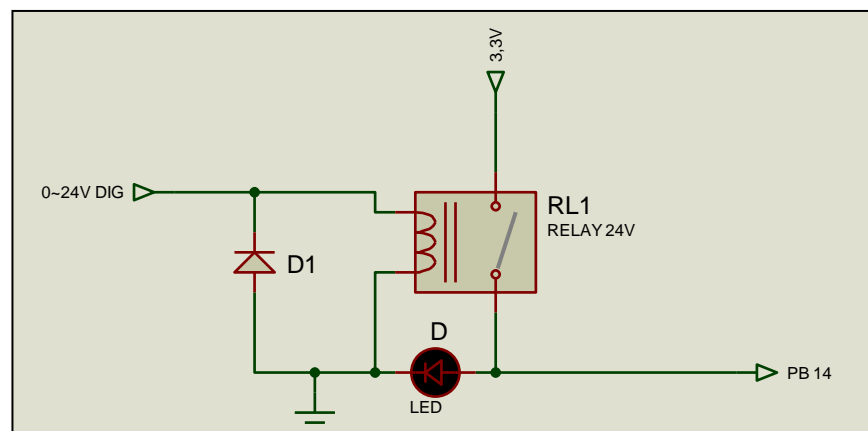


Figure (II.1) : schéma du circuit d'adaptation de l'entrée digitale 0~24VDC

b- **L'entrée 0~5V** : Affectée à l'entrée PI1, la tension est adaptée et isolée via le relais RL2 avec protection de diode libre D2. La LED D indique l'état de l'entrée comme illustre le schéma de la figure (II.2).

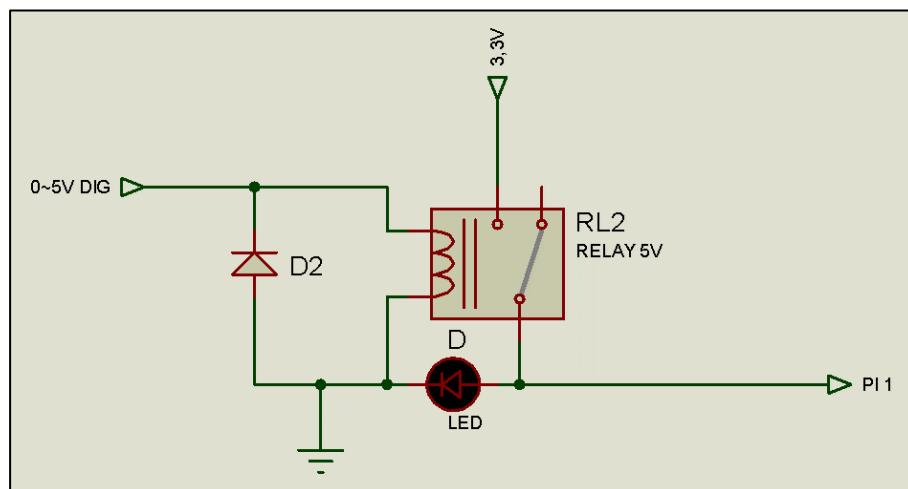


Figure (II.2) : schéma d'adaptation de l'entrée digitale 0~5VDC

II.2.2.2 Entrées analogiques

a- **Entrée courant 04-20mA** : Le convertisseur courant tension est réalisé à l'aide d'un amplificateur opérationnel. L'entrée 4–20mA est convertie en tension compatibles 0-3.3VDC accessible par la carte au pin PA0. La figure (II.3) montre le schéma du circuit adopté.

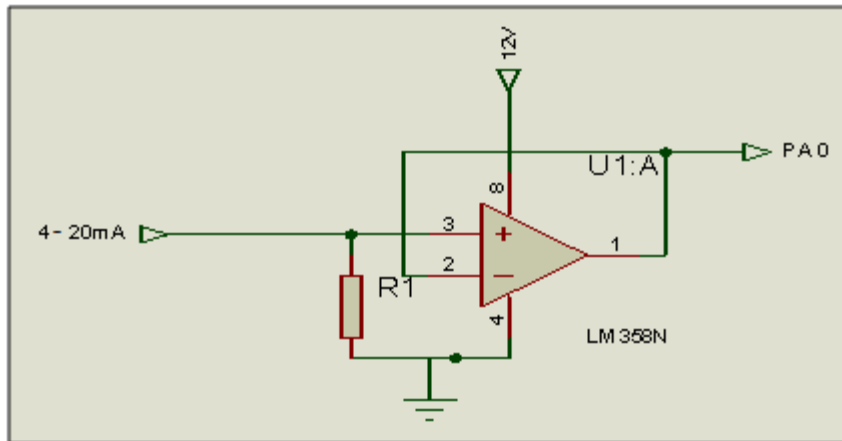


Figure (II.3) : schéma d'entrée 4–20 mA analogique

Pour choisir la résistance R1, nous considérons la contrainte de limite de la tension d'entrée au niveau de la carte Stm32F7 qui est de 3.3VDC. Ainsi, à la limite le courant de 20mA doit correspondre à cette tension.

D'où selon le schéma précédent, la relation $V=R1.I$ permet de déterminer cette résistance par calcul directe où nous trouvons $R1=165\text{Ohm}$.

Cependant, à la limite inférieure, le courant 04mA produit une tension à l'entrée de la carte égale à 0.66VDC. Ainsi, dans la partie programmation, nous devons faire correspondre les deux tensions limites 0.66VDC et 3.3VDC à la plage de variation de la grandeur d'entrée.

b- Entrée analogique tension : 0–10V : Ce type d'entrée est adapté en utilisant un montage en pont diviseur afin de réduire la tension supérieur à 3.3VDC. La figure (II.4) illustre le schéma complet l'interface qui délivre une tension normalisée au pin PF9. Ici, le filtre (R2,C1) réalise un filtre passe bas, tandisque l'isolation des étages est assurée par le montage en suiveur des deux amplificateurs U1 :A et U1 :B implanté dans le circuit LM358N.

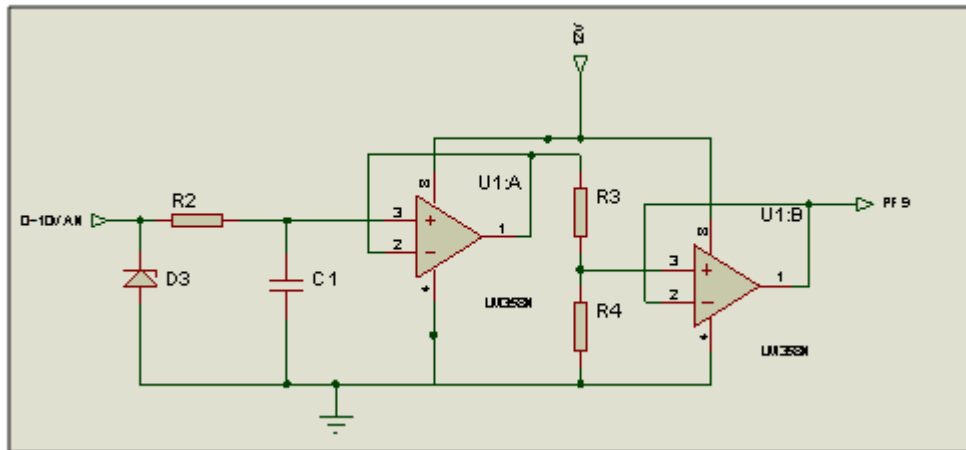


Figure (II.4) : Schéma d'interface de l'entrée analogique 0–10V

La transmission des signaux analogiques en milieu industriel est particulièrement susceptible d'être perturbée, pour cela on nous choisissons le filtre passe bas à fréquence de coupure $f_c=50\text{Hz}$ donnée aussi par la relation :

$$F = \frac{1}{2\pi \cdot R2 \cdot C1}$$

Ainsi, nous obtenons $R2=678\Omega$ si la capacité est choisie $C1=4.7\mu\text{F}$. Pour déterminer les résistances, nous utilisons la relation à la tension max 10VDC : $10 \cdot \frac{R4}{R3+R4} = 3.3\text{V}$. Si nous choisissons $R3=200\Omega$, nous obtenons $R4=98.50\Omega$.

c- Entrée analogique tension : 0–5V :

La figure (II.5) illustre le schéma du circuit d'adaptation. Avec un montage similaire au cas précédent (tension d'entrée 0-10V), cette entrée est appliquée au pin PF10.

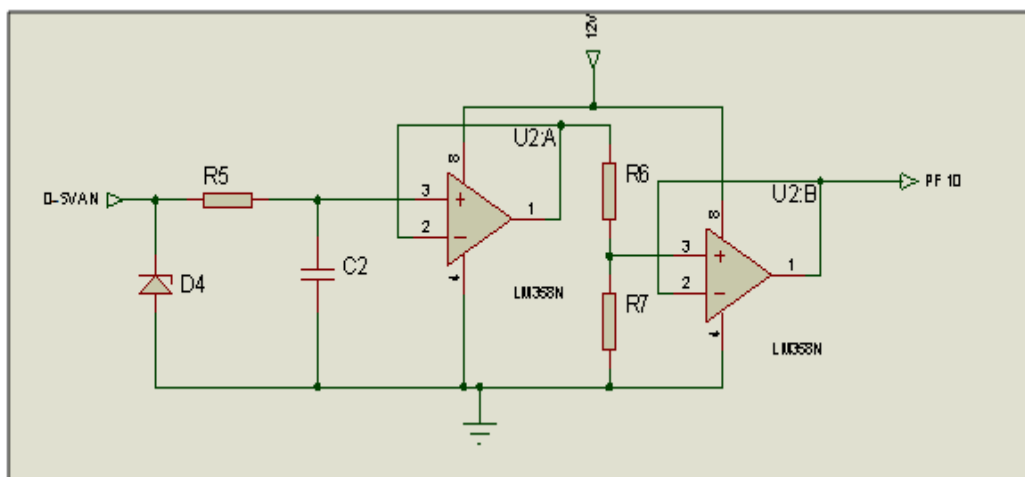


Figure (II.5) : schéma d'entré 0–5V analogique

Etage de filtrage : $C2=4.7\mu\text{f}$ et $R5=678\Omega$.

Etage de division de la tension : $R6=100\Omega$ et $R7=194.117\Omega$

II.2.3 Adaptation des sorties

Les sorties à considérer sont de type digitales au nombre de trois comme suit :

a- **Sortie digitale 0~24VDC** : Nous utilisons un amplificateur de courant, le transistor Q4 alimentée en 12VDC pour exciter la bobine du relais RL3.

Ce dernier assure à la sortie une tension 24VDC. La base du transistor est reliée au pin PB15 de la carte à travers la résistance de limitation de courant R8.

Le transistor type NPN BD177 supporte jusqu'à 60VDC sous un courant 3A entre le collecteur et l'émetteur, largement suffisant pour piloter la bobine du relais. Il joue le rôle d'interrupteur télécommandé.

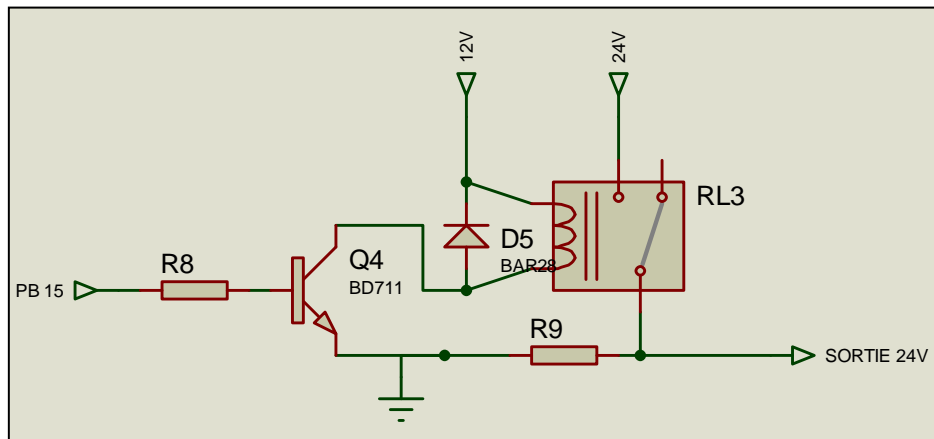


Figure (II.6) : Schéma de la sortie digitale 24VDC

La diode de roue libre D5 permet de créer une « boucle infinie » qui permet de limiter la surtension de la bobine et d'en évacuer le courant produit.

b- **Sortie digitale 0~5V**

Une sortie de niveau 05VDC est affectée au pin PA15 de la carte. De la même manière que le circuit précédent, un transistor Q5 adapte la tension et amplifie-le courant d'excitation de la bobine du relais de sortie associé RL5 telle que montre la figure (II.7).

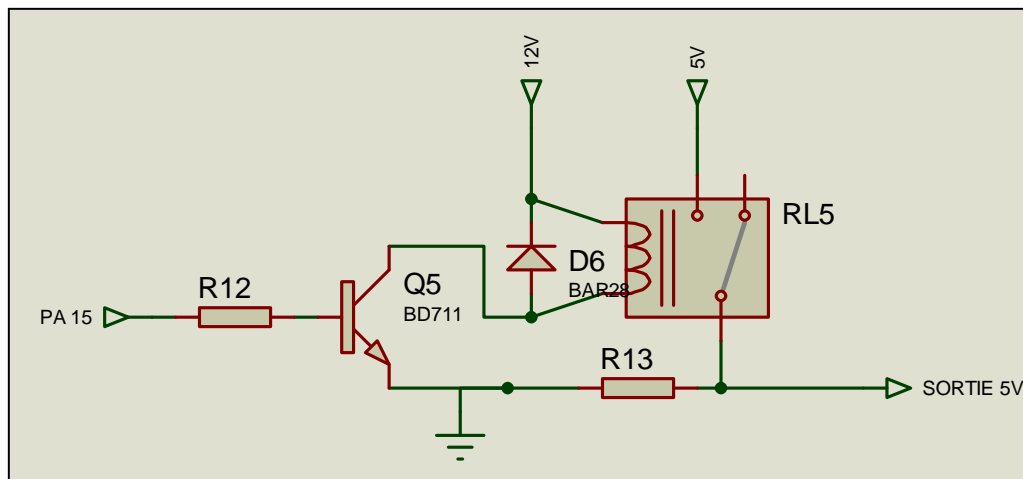


Figure (II.7) : Schéma de la sortie digitale 5VDC

Enfin, sous contrainte de disponibilité, nous avons adopté les valeurs des résistances indiquées dans le tableau suivant :

Tableau (II.1) : valeurs des résistances calculés et résistances utilisées

Les resistances	Les valeurs calculers	Les valeurs proches
R1	165 Ω	160 Ω
R2=R5	678 Ω	670 Ω
R3	200 Ω	200 Ω
R4=R6	98.50 Ω	100 Ω
R7	194.17 Ω	200 Ω
R8=R12	6.5k Ω	6.5k Ω
R9=R13	1k Ω	1K Ω

II.3 Configuration et Programmation des fonctions nécessaires :

Dans cette partie, nous allons développer le programme de configuration et d'activation des entrées et des sorties ainsi que celui de la conversion des entrées analogiques et le mode d'accès en interruption

II.3.1 Les entrées et les sorties digitales:

La carte STM32F4 dispose de 23 sources d'interruptions ou événements externes. Ils sont divisés en 2 sections, première section d'interruption est pour les broches externes (P0 à P15) (GPIO) sur chaque port, et autre section est pour d'autres événements, comme l'interruption du RTC, Ethernet, USB et autres.

Dans notre projet, nous avons défini les interfaces comme suit :

Les deux entrées digitales Digital_Input_5V, et Digital_Input_24V liées respectivement aux pins PI1, et PB14. Les trois sorties, Digital_Output1_24V, Digital_Output2_24V, et Digital_Output_5V liées respectivement aux pins PB15, PI10, et PA15. Leur configuration dans le logiciel STM32CubeMx est présentée par la Figure (II.8) où les numéros indiqués sont comme suit :

- (1) : Entrées : PC4 & PC5 et sorties PE11, PE12, PE13.
- (2) : Activation du mode interruption des entrées des pinPC4 (line 4) et PC5 (line 5).

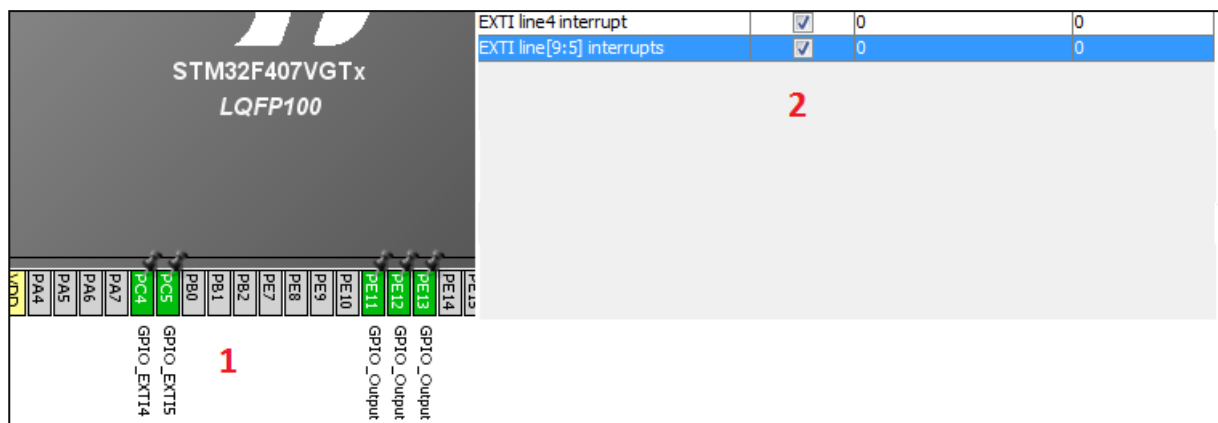


Figure (II.8) : La configuration d'entrée et de sortie et la mode d'interruption en STM32CubeMx

L'initialisation des ports est comme indiquée à la figure (II.9).

```

137 void MX_GPIO_Init(void)
138 {
139     GPIO_InitTypeDef GPIO_InitStructure;
140
141     /* GPIO Ports d'activation d'horloge */
142     __HAL_RCC_GPIOC_CLK_ENABLE();
143     __HAL_RCC_GPIOE_CLK_ENABLE();
144
145     /*Configuration GPIO pins : PC4 PC5 */
146     GPIO_InitStructure.Pin = GPIO_PIN_4|GPIO_PIN_5;
147     GPIO_InitStructure.Mode = GPIO_MODE_IT_RISING;
148     GPIO_InitStructure.Pull = GPIO_NOPULL;
149     HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);
150
151     /*Configuration GPIO pins : PE11 PE12 PE13 */
152     GPIO_InitStructure.Pin = GPIO_PIN_11|GPIO_PIN_12|GPIO_PIN_13;
153     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
154     GPIO_InitStructure.Pull = GPIO_NOPULL;
155     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
156     HAL_GPIO_Init(GPIOE, &GPIO_InitStructure);
157
158     /*Configuration GPIO pin de sorties */
159     HAL_GPIO_WritePin(GPIOE, GPIO_PIN_11|GPIO_PIN_12|GPIO_PIN_13, GPIO_PIN_RESET);
160
161     /* EXTI initialisation des interruptions */
162     HAL_NVIC_SetPriority(EXTI4_IRQn, 0, 0);
163     HAL_NVIC_EnableIRQ(EXTI4_IRQn);
164
165     HAL_NVIC_SetPriority(EXTI9_5_IRQn, 0, 0);
166     HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
167
168 }
169

```

Figure (II.9) : La configuration d'entrée et de sortie et la mode d'interruption en Keil

Les fonctions de base des entrées et des sorties digitales sont indiquées dans le tableau (II.2).

Tableau (II.2) : Les fonctions de lecture d'E /S digitales

Fonction	Description
HAL_GPIO_ReadPin(GPIO(A-E), GPIO_PIN_(0-15));	Lire l'état de pin (A-E)(0-15)
HAL_GPIO_TogglePin (GPIO(A-E),GPIO_PIN_(0-15));	Alternation des broches GPIO spécifiées
HAL_GPIO_WritePin (GPIO(A-E),GPIO_PIN_(0-15), GPIO_PIN_(SET/RESET));	Mise à « 1 » (SET) ou efface(RESET) le bit de port de données sélectionné.

II.3.2 Les sorties analogiques :

Nous avons défini trois entrées analogiques : Analog_Input_420mA, Analog_Input_5V, et Analog_Input_10V, liées respectivement aux pins PA0, PF10, et PF9. Pour l'acquisition, nous utilisons les canaux 3,5, et 7 de l'ADC (premier convertisseur analogique numérique) du STM32.

Pour cela, nous utilisons l'accès direct en mode DMA (Direct Memory Access) pour éviter l'écrasement des données enregistrée dans l'unique registre ADC_DR suite à une conversion rapide par rapport au programme de traitement.

Lorsque la période du cycle de conversion est plus rapide (37 μ s au minimum), il est nécessaire d'utiliser un temporisateur (dans notre cas le TIMER2) pour maîtriser la période de conversion (ici nous avons choisi 50 ms).

La configuration des paramètres et la fonction utilisée dans le temporisateur (Timer2) et du convertisseur (ADC1), sont montrées par les figures (II.10), (II.11) et (II.12).

```
/*1'appel de fonction*/
TIM_HandleTypeDef htim2;

/* initialisation de fonction Timer2 */
void MX_TIM2_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;

    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 16000;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 50; //periode de timer
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    HAL_TIM_Base_Init(&htim2);

    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig);

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig);
}
```

Figure (II.10) : Configuration du Timer2 en keil v5.

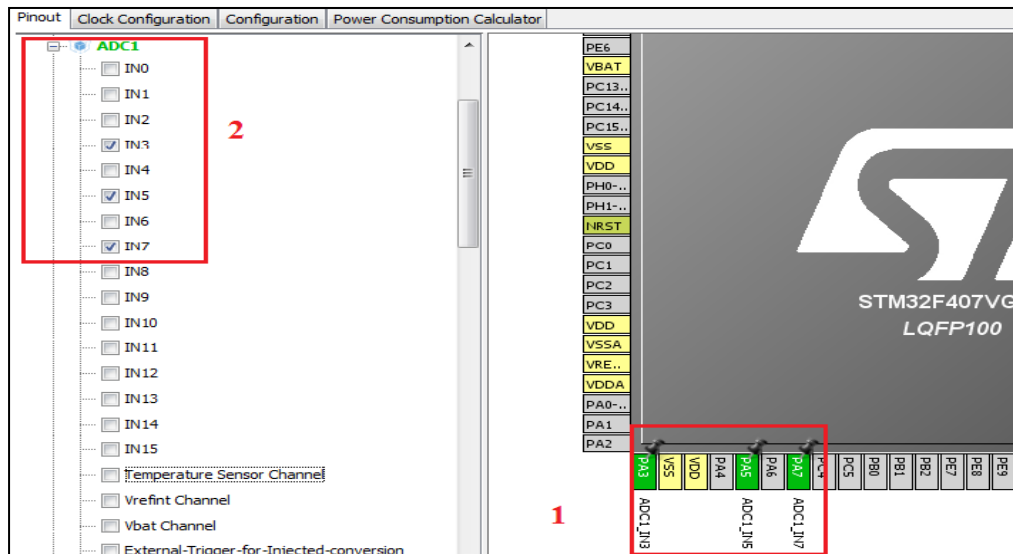


Figure (II.11) : Choix des pins et les canaux de l’ADC1 en STM32 CubeMx.

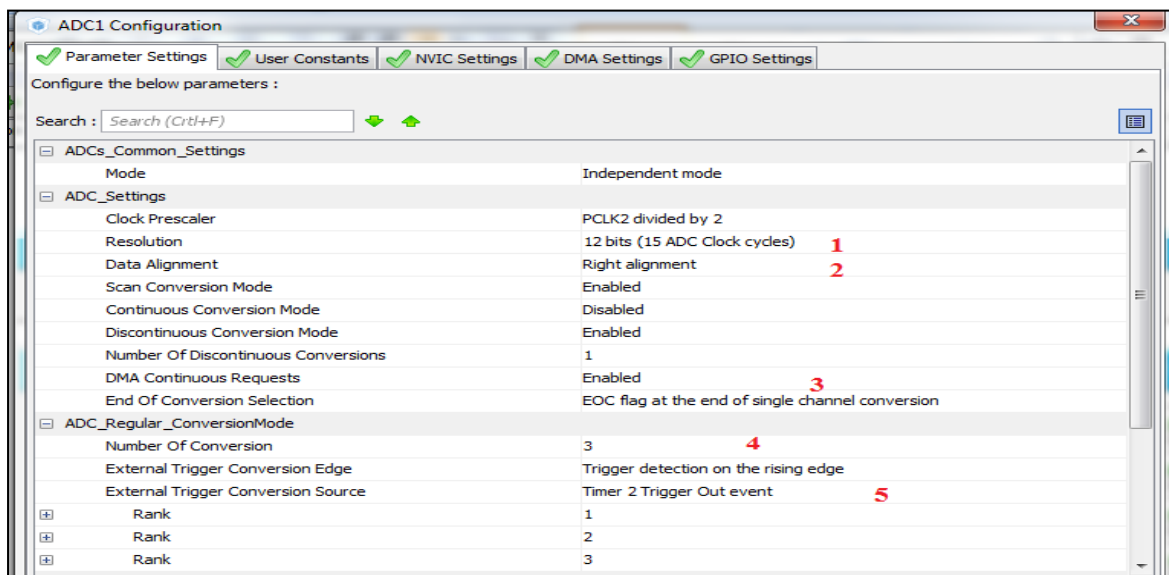


Figure (II.12) : Configuration de l’ADC en STM32 CubeMx.

Les indications dans la figure (II.14) sont comme suit :

(1) : Choix de la résolution de 12 bits (maximum). (2) : Alignement à droite de la donnée de conversion (3) : Activation de la DMA. (4) : Le nombre de conversion. (5) : Géré l’ADC1 par Timer2.

La figure(II.13) montre les fonctions d'initialisation et de récupération de la donnée de conversion. Les indications sont comme suit :

(1) : l'appel des fonctions (DMA, ADC1). (2) : l'initialisation des fonctions. (3) : démarrage de la conversion avec DMA.

```

43  ADC_HandleTypeDef hadc1;
44  DMA_HandleTypeDef hdma_adc1;      1
45
46
--
127  MX_DMA_Init();
128  MX_ADC1_Init();                  2
129
132  HAL_ADC_Start_DMA(&hadc1, (uint32_t *)valeur_convertir,3);      3

```

Figure (II.13) : Les fonctions d'initialisation et d'appel de l'ADC1 en Keil v5.

II.3.3 Mise à l'échelle des entrées analogique

Les valeurs des entrées analogiques acquises doivent être mise à l'échelle réel de leurs grandeurs physique correspondantes selon l'échelle établit au niveau des instruments de mesures. Cela facilite la manipulation de ces données et leurs affichage sur l'interface homme machine à développer.

Pour ce faire, nous proposons de créer une fonction linéaire pour chaque entrée. Nous considérons la fonction que nous appellerons "alpha"(voir la figure (II.14)) pour calculer la valeur physique (notée valeur_phy) à partir de la valeur acquise (notée valeur_convertir).

Nous montrons que cette fonction est donnée en fonction des valeurs maximum et minimum des variables valeur_phy et valeur_convertir comme suit :

$$\text{valeur_phy} = \alpha * \text{valeur_convertir} + \text{beta}$$

$$\text{Où } \alpha = \frac{\text{valeur_phy_max} - \text{valeur_phy_min}}{\text{valeur_convertir_max} - \text{valeur_convertir_min}}$$

$$\text{Et } \text{beta} = \text{valeur_phy_min} - \alpha * \text{valeur_convertir_min}$$

```
1 void alphabeta(float valeur_phy_min,float valeur_phy_max,float valeur_convert_min,
2 float valeur_convert_max,float *alpha, float *beta){
3 alpha [0]=(valeur_phy_max-valeur_phy_min)/(valeur_convert_max-valeur_convert_min);
4 beta [0]= (valeur_phy_min)-((*alpha)*(valeur_convert_min));
5 }
6 void valeur_phy(float *valeur_phy1,float valeur_convert,float alpha,float beta){
7 valeur_phy1[0]=alpha* valeur_convert+beta;
8
9 }
10
```

Figure (II.14) : Fonction de mise à l'échelle

II.4 Conclusion :

Dans ce chapitre, nous avons développé la section des entrées et des sorties.

Le bloc d'alimentation d'un PC assure divers niveaux de tension avec une puissance suffisante pour alimenter la carte d'interfaçage. Le programme du STM32F7 assure l'acquisition des entrées digitales et analogiques avec mise à l'échelle de ces derniers.

Chapitre III :

Conception de la section IHM et communication

III.1 Introduction

La configuration et l'exploitation de la carte de contrôle nécessite l'introduction de plusieurs valeurs de paramétrage et de fonctionnement qui doivent être sauvegardées en mémoire non volatile.

En effet, l'utilisateur doit configurer les entrées et les sorties à utiliser, les adresses, les noms des variables, les plages des échelles des variables analogiques...etc. Pour simplifier l'interaction avec le programme, nous développerons l'Interface Homme Machine (IHM) sur l'afficheur associé à la carte en option.

D'autre part, la carte de contrôle développée devra pouvoir échanger des données de contrôle avec les instruments intelligents local ou distants. Ces instruments, souvent connectés en réseau local, utiliseront le protocole très connue Modbus/TCP.

Dans ce cas, nous saurons en mesure d'intégrer notre solution dans un réseau d'information déjà existant (exemple internet).

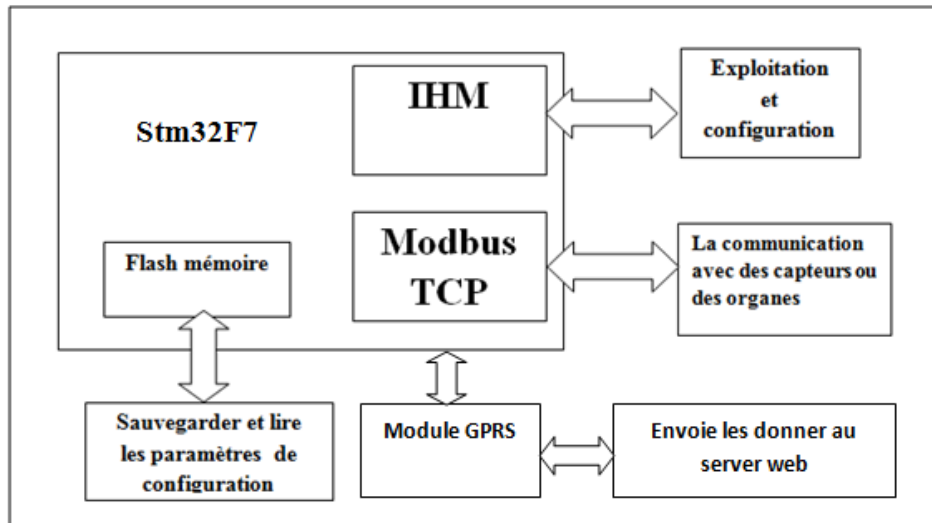
Quant à l'envoi des données et leur sauvegarde dans des bases distantes, cette carte de contrôle doit assurer la fonction de communication via le réseau GSM/GPRS. Pour cela le module de communication Sim900a sera intégré.

Les données envoyées seront insérées dans une base de données déjà existante.

Dans ce présent chapitre, nous allons développer une interface Homme Machine « IHM » qui nous permettra d'introduire les paramètres de fonctionnement et d'afficher les données du procédé.

Aussi, nous allons concevoir l'interface de communication locale en protocole Modbus/TCP à travers le port Ethernet et distante via le module Sim900a.

III.2 Schéma de visualisation et de communication



Figure(III.1) : Schéma de visualisation et de communication.

La figure (III.1) montre le schéma de principe de la section IHM et communication. L'opérateur utilise le module IHM pour la configuration et l'exploitation de la carte. C'est une liaison bidirectionnelle.

Les données concernées sont enregistrées dans la mémoire Flash (flash memory). De même, la communication avec les capteurs et les organes de contrôle s'effectue à travers le module Modbus/TCP.

Une liaison bidirectionnelle est réalisée en liaison Rj45. Une communication GSM/GPRS avec un serveur web distant est mise en œuvre par le biais du module

III.3 Le Module IHM

Cette interface est matérialisée dans un écran couleur de ROCKTECH, de 4,3 pouces 480x272 LCD-TFT avec écran tactile capacitif. Elle est connectée à l'interface LCD RGB de la carte STM32F7 [6].

Nous avons considéré selon le type de la variable entrée ou sortie, digital ou analogique les propriétés suivante : nom, unité, à transférer en Modbus/TCP ou non, valeur maximale et valeur finale. Le tableau (III.1) donne un exemple de cette configuration.

Tableau (III.1) : La configuration des paramètres des entrées et de sorties.

	Paramètres				
Entrées analogiques	Nom	Unite	Modbus TCP	Valeur physique min	Valeur physique max
Analog_Input_420mA	Température de four	C°	Oui /non	0 C°	500 C°
Analog_Input_5V	Pression de la vapeur	Bar	Oui /non	0 bar	10 bar
Analog_Input_10V	Débit de réservoir 1	Litre	Oui /non	0 Litre	5000 Litre
Entrées digitales	Nom	Unite	Modbus TCP	Valeur physique min	Valeur physique max
Digital_Input_5V	Capteur 1	-	Oui /non	-	-
Digital_Input_24V	Capteur 2	-	Oui /non	-	-
Sorties digitales	Nom	Unite	Modbus TCP	Valeur physique min	Valeur physique max
Digital_Output_5V	Actionneur1	-	Oui /non	-	-
Digital_Output1_24V	Actionneur2	-	Oui/ non	-	-
Digital_Output2_24V	Actionneur3	-	Oui/non	-	-

Le module IHM permet ainsi l'exploitation des données. Et ceci à travers l'affichage des grandeurs physique dans des graphes et des variables digitaux comme des leds rouge/vert.

III.3.1 Mise en œuvre Matériel

La carte STM32F746NG-DISCO contient un contrôleur d'affichage (LCD-TFT Controller (LTDC)).

Ce contrôleur fournit les signaux RGB (Rouge, Vert, Bleu) et des signaux horizontal, synchronisation verticale, l'horloge pixel et des données activées comme sortie via interface liée directement aux panneaux TFT (Thin Film Transistor), avec les caractéristiques suivantes [6]:

Le tableau (III.2) ci-dessous résume les pins et les signaux d'interface de LTDC.

Tableau (III.2) : les signaux d'interface de LTDC.

Signal LCD-TFT	E/S	Description	Les pins
LCD_CLK	S	La sortie de l'horloge	PI14
LCD_HSYNC	S	La synchronisation horizontale	PI10
LCD_VSYNC	S	La synchronisation verticale	PI9
LCD_DE	S	Pas de données activées	PK7
LCD_R [0 :7]	S	Données: 8 bits de données Rouges	[PI15, PJ0, PJ1, PJ2, PJ3, PJ4, PJ5, PJ6]
LCD_G [0 :7]	S	Données: 8 bits de données Vert	[PJ7, PJ8, PJ9, PJ10, PJ11, PK0, PK1, PK2]
LCD_B [0 :7]	S	Données: 8 bits de données Blue	[PE4, PJ13, PJ14, PJ15, PG12, PK4, PK5, PK6]

III.3.2 Mise en œuvre Logiciel

L'interface IHM utilisée s'intègre directement sur la carte STM32F7. Les différents vus d'exploitation et de configuration sont créés en suivant les étapes suivantes :

- 1-Création des images statiques avant l'action et après l'action.
- 2-Affecter des événements aux différentes images d'action. Ces fonctions s'exécuteront sur la carte dès que l'événement se produit (par exemple un click).

Un cadre logiciel (software framework) d'exploitation de l'interface est mis à la disposition de l'utilisateur. Seulement, elle facilite les tâches de communication entre la carte et l'IHM et non l'édition des images et des programmes associés.

III.3.2.1 Le cadre de Logiciel (Touch GFX)

Touch GFX est un cadre logiciel unique qui déverrouille l'interface utilisateur graphique (GUI) les performances de votre matériel à faible ressource.

Les fonctionnalités intégrées du matériel ST, les deux dimensions transferts DMA et l'accélération graphique Chrom-ART, assurent une charge de MCU très faible - même les plus

complexes des transitions d'écran les complète avec massif alpha-blending et le rendu du texte.

La combinaison de microcontrôleurs STM32 et logiciels Touch GFX facilite la conduite et une création facile de solutions d'affichage TFT qui vivent aux normes des appareils modernes intelligents qui exigeraient normalement processeurs d'applications complexes et étendues [7], dans les figure (III.2), une exemple de programme qui permet nous l'affichage d'une entrées digitale.

```

61 void EditorView::handleTickEvent(){
62   GPIO_PinState mystate;
63   if(bpressed == 1){
64     mystate = HAL_GPIO_ReadPin(GPIOI,GPIO_PIN_11);
65     if (mystate == GPIO_PIN_RESET){
66       d_in1.setBitmap(Bitmap(BITMAP_DIGITAL_P_ID)); 2
67       d_in1.invalidate();
68     }
69     if (mystate == GPIO_PIN_SET)
70     {
71       d_in1.setBitmap(Bitmap(BITMAP_DIGITAL_ID)); 3
72       d_in1.invalidate();
73     }
74     bpressed=0;
75   }
76 }
77
78 extern "C"{
79 void EXTI15_10_IRQHandler(void)
80 {
81     static uint32_t debounce_time = 0;
82
83
84     bpressed=1;
85
86     /* USER CODE END EXTI15_10_IRQn 0 */
87     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_11); 1
88     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
89
90

```

Figure (III.2) : programme d'affichage d'entre digitale.

L'affichage de l'entre digitale, sous les étapes suivantes :

- (1): Faire une interception.
- (2) et (3) : Si l'état de pin est '0' ou bien '1' on va afficher l'image qui présente cette état.

III.4 Module de Communication Modbus /TCP

Ce module assure la communication via le protocole Modbus/TCP avec d'autres instruments du même réseau. Dans la suite, nous allons introduire tout d'abords le protocole Modbus ensuite sa version Ethernet Modbus/TCP.

III.4.1 Protocole Modbus

Le protocole Modbus (marque déposée par la société Modicon récupérée par Schneider electric) est un protocole de dialogue basé sur une structure hiérarchisée entre un client unique et plusieurs serveurs. Il se situe à la septième couche dans le modèle OSI. Il utilise la communication maître/esclave entre des appareils connectés même sur différents types de bus ou de réseaux. Ce protocole consiste à la définition de trames d'échange ;[5]

- ✓ Le maître envoie une demande et attend une réponse.
- ✓ Deux esclaves ne peuvent dialoguer ensemble.
- ✓ Le dialogue maître/esclave peut être schématisé sous une forme successive de liaisons point à point comme indique la figure (III.2)

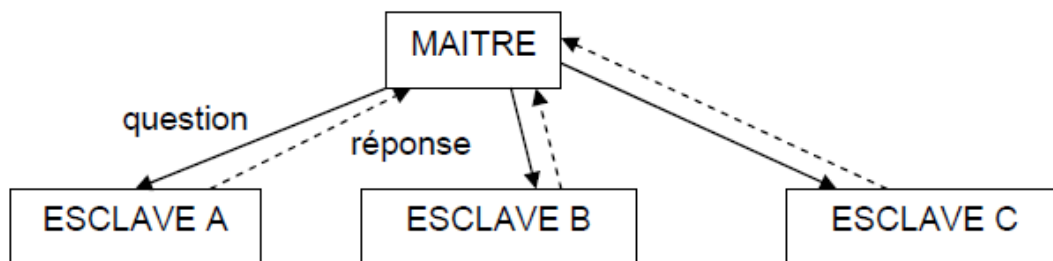


Figure (III.3) : Principe de communication Client/ serveur.

Les abonnés du bus sont identifiés par des adresses attribuées par l'utilisateur. L'adresse de chaque abonné est indépendante de son emplacement physique. Les adresses vont de 1 à 64 et ne doivent pas obligatoirement être attribuées de manière séquentielle. Deux abonnés ne peuvent avoir la même adresse.

a- Trame d'échange requête/réponse

La requête contient un code de fonction indiquant à l'esclave adressé quel type d'action est demandé telle que montre la figure (III.3). Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction. Le champ octets de contrôle permet à l'esclave de s'assurer de l'intégralité du contenu de la question.

N° d'esclave	Code fonction	Information spécifique concernant la demande	Mot de contrôle
1 octet	1 octet	n octets	2 octets

Figure (III.4) : Trame de requête Modbus.

L'esclave répond par la trame de la figure (III.4). Le code de la fonction est retourné avec les données nécessaires.

N° d'esclave	Code fonction	Données reçues	Mot de contrôle
1 octet	1 octet	n octets	2 octets

Figure (III.5) : Trame de réponse Modbus.

Chaque octet composant une trame est codé sur 2 caractères hexadécimaux (2 fois 4 bits). La taille maximale des données est de 256 octets.

b- Fonctions principales du Modbus

Le protocole MODBUS offre jusqu'à 19 fonctions différentes de manipulations de bits et de registres. Le tableau (III.3) montre quelques fonctions principales. Elles se caractérisent par un code fonction sur un octet (en hexadécimal). Tous les équipements ne supportent pas tous les codes fonction.

Tableau (III.4) : Les fonctions de Modbus.

Code	Nature des fonctions Modbus
H'01'	Lecture de n bits de sortie consécutifs
H'02'	Lecture de n bits de sortie consécutifs
H'03'	Lecture de n mots de sortie consécutifs
H'04'	Lecture de n mots consécutifs d'entrée
H'05'	Ecriture de 1 bit sortie
H'06'	Ecriture de 1 mot sortie
H'07'	Lecture du statut d'exception
H'0F'	Ecriture de n bits sortie
H'10'	Ecriture de n mots sortie

III.4.2. Protocole Modbus/TCP

Modbus TCP / IP utilise le protocole TCP / IP et Ethernet pour transporter les données de la structure de message Modbus entre des appareils compatibles. Modbus TCP / IP combine un réseau physique (Ethernet), avec une mise en réseau standard (TCP / IP), et le protocole

Modbus. Essentiellement, le message Modbus TCP/IP est simplement une communication Modbus encapsulé dans un réseau Ethernet TCP / IP.

Dans la pratique, Modbus/TCP intègre une trame de données standard Modbus TCP dans un cadre, sans la somme de contrôle Modbus, comme le montre le schéma ci-dessous.[8]

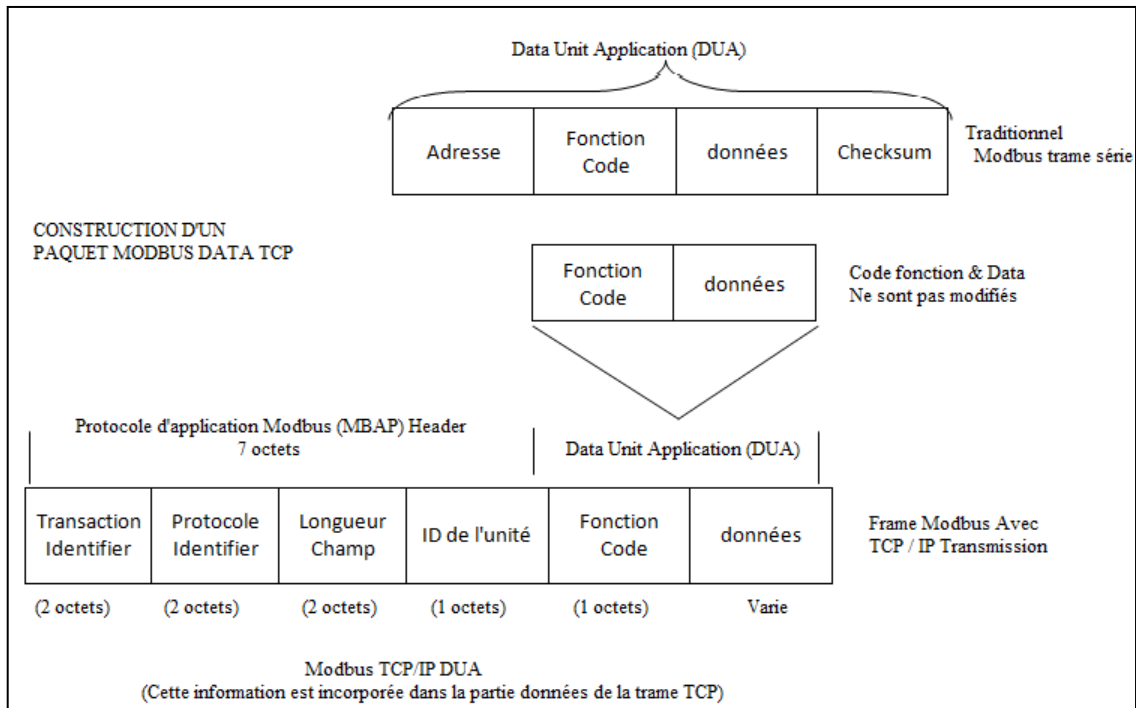


Figure (III.6) : Trame MODBUS/TCP

Le tableau ci-dessous illustre les différentes fonctions du schéma bloc de la figure précédente.

Tableau (III.5) : Encapsulation de la trame MODBUS TCP/IP.

Bloc	Fonction
Identificateur de transaction	Pour la synchronisation entre les messages du serveur et client
Identificateur de protocole	Zero pour Modbus /TCP
Champ Longueur	Nombre d'octets restants dans ce cadre
Adresse identifiant	Adresse esclave (255 si non utilise)
Code de fonction	Codes de fonction comme dans d'autres variantes
Octets de données	Données de la réponse ou des commandes

III.4.3. Mise en œuvre Matériel

Le périphérique Ethernet sur la carte STM32F7 permet de transmettre et de recevoir des données via Ethernet en conformité avec la norme IEEE 802,3 à 2.002.

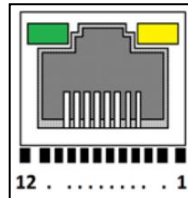


Figure (III.7) : Connecteur RJ45 Ethernet CN9 (vue de face).

L'interface Ethernet prend en charge, selon la configuration, deux interfaces standard de l'industrie : par défaut, MII (Media Independent Interface) définies dans les spécifications IEEE 802.3 et RMI (Reduced Media Independent Interface). Les pins du port Ethernet sont illustrés par la figure (III.6) tandis que leurs attributions aux pins de la carte sont indiquées dans le Tableau (III.6)[7].

Tableau (III.6) : Attribution des pins d'Ethernet TCP sur la carte STM32F7

Fonction	Pin
RMII_REF_CLK	PA1
RMII_MDIO	PA2
RMII_MDC	PC1
RMII_MII_CRSDV	PA7
RMII_MII_RXD0	PC4
RMII_MII_RXD1	PC5
RMII_MII_TX_RXER	PG2
RMII_MII_TX_EN	PG11
RMII_MII_RXD0	PG13
RMII_MII_RXD1	PG14

III.4.4 Mise en œuvre logiciel

L'exploitation du port Ethernet nécessite de trouver une plateforme logiciel permettant de fournir à la fois les fonctionnalités du Modbus et du protocole de transfert TCP.

Pour cela, nous proposons d'utiliser des bibliothèques dédiés.

a- Bibliothèque Modbus (Master MODBUS)

L'intégration du protocole Modbus/TCP dans une application industrielle embarquée nous demandera un effort initiale important pour définir et programmer les différentes fonctionnalités tant au niveau de Modbus comme au niveau du protocole de transfert TCP.

Une fois les fonctions sont prédéfinies, nous devons configurer, programmer et synchroniser les taches de communications entre les éléments du réseau.

Dans notre travail, nous avons recherché les bibliothèques des fonctions Modbus/TCP.

En C avant d'optimiser l'effort et le temps de programmation. Plusieurs bibliothèque sont disponibles mais sous différentes contraintes concernant particulièrement le système d'exploitation, maitre ou esclave.

Pour notre projet, nous avons choisi la bibliothèque **free Modbus V3.0** Master sous le système d'exploitation Free RTOS puisque la carte STM32F7 fonctionnera en tant que maitre (Master)[9].

b- Bibliothèque TCP (lwIP)

Le **light weight Internet Protocol** (lwIP) est une petite application indépendante de la suite de protocole de réseau.

Elle offre les fonctions de base de transfert de donnée en TCP/IP.

La bibliothèque lwIP réduit la mémoire d'utilisation des ressources tout en ayant une connexion TCP à grande échelle. Cela rend lwIP adapté pour une utilisation dans les systèmes embarqués avec des dizaines de kilo-octets.[10]

Pour configure TCP en utilise les fonctions présentées dans le tableau (III.7).

Tableau (III.7) : Quelques fonctions utilisées dans le programme de Modbus TCP.

Fonction	Description
tcpip_init();	Création d'un objet TCP/IP
Netif_Config() ;	Cite les adresse IP, mask, et adresse
netif_add	Créer une nouvelle interface réseau TCP/IP
ethernetif_init () ;	Initialise l'interface réseau TCP IP
low_level_init () ;	Initialise le matériel (hardware)
low_level_input () ;	Echange le paquet
eMBMTCPFrameInit () ;	Initialise la trame
eMBMTCPFrameSend () ;	Envoi la trame vers l'esclave.
eMBMTCPFrameReceive () ;	Reçois la trame de l'esclave.

Tableau (III.8) : les paramètres de configuration.

Paramètre	Description
ip_addraddr	Adresse ip '192.168.0.11'
ip_addrnetmask	Adresse mask '2050.250.250.0'
ip_addrgw	Adresse getway
EthHandle.Init.Speed	La vitesse de transmission (100M/s)
EthHandle.Init.DuplexMode	Le mode de transmission
EthHandle.Init.RxMode	Mode d'interruption
EthHandle.Init.PhyAddress	L'adresse physique 'LAN8742A'

La figure (III.7) illustre le diagramme de communication Modbus TCP ;

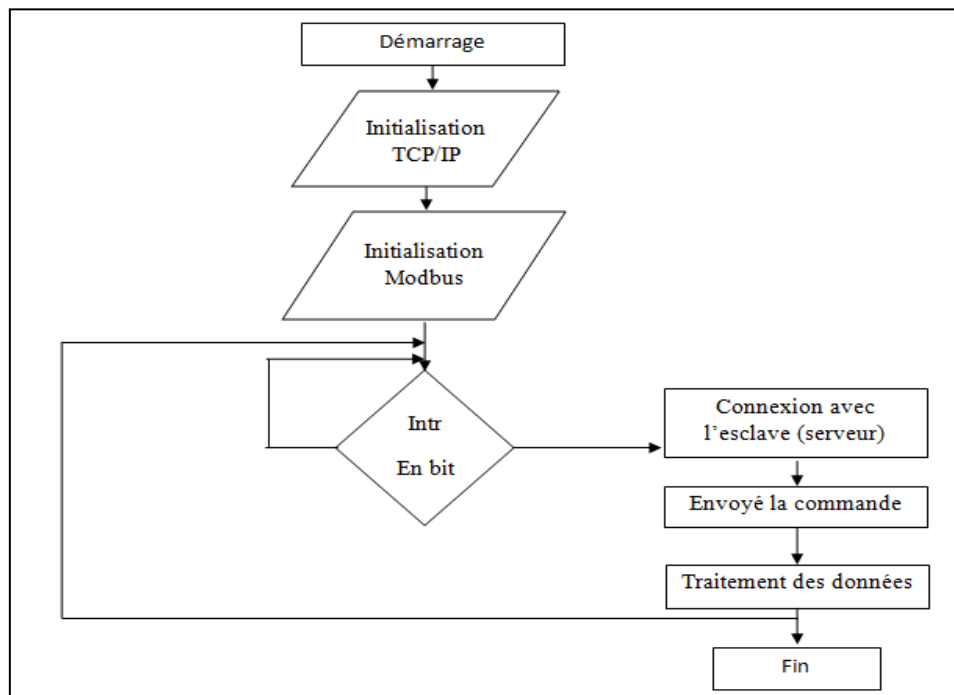


Figure (III.8) : Diagramme de communication Modbus TCP.

III.5. Module de Mémoire

Dans les systèmes embarqués, il est très important de disposer d'une mémoire permettant de stocker les informations.

Microcontrôleur utilise la mémoire flash qui peut être effacé et reprogrammé. Nous sauvegardons les données et les variables initiales de configuration pour éviter la perte en cas de redémarrage du système

III.5.1 Mise en œuvre Matériel

La carte STM32F7 à un microcontrôleur qui a une capacité de mémoire flash 1 mégaoctet. La mémoire flash du microcontrôleur a été divisée en 12 secteurs.

Certains secteurs sont de petite taille et autres de grande taille. L'espace de mémoire flash est présenté dans le tableau de la page suivante.

La division de l'espace de mémoire flash de STM32F746NG-DISCO est suivante :

Tableau (III.9) : Espace de la mémoire flash de la carte STM32F7.[2]

Bloc	Le nom	Les adresses de base	La taille
Mémoire principale	Secteur 0	0×0800 0000 – 0×0800 3FFF	16 Kb
	Secteur 1	0×0800 4000 – 0×0800 7FFF	16 Kb
	Secteur 2	0×0800 8000 – 0×0800 BFFF	16 Kb
	Secteur 3	0×0800 C000 – 0×0800 FFFF	16 Kb
	Secteur 4	0×0801 0000 – 0×0801 FFFF	64 Kb
	Secteur 5	0×0802 0000 – 0×0803 FFFF	128 Kb
	Secteur 6	0×0804 0000 – 0×0805 FFFF	128 Kb
	.	.	.
	.	.	.
	.	.	.
	Secteur 11	0×080E 0000 – 0×080F FFFF	128 Kb
Système de mémoire		0×1FFF 0000 – 0×1FFF 77FF	30 Kb
OTP		0×1FFF 7800 – 0×1FFF 7A0F	528 Kb
Les options des octets		0×1FFF C000 – 0×1FFF C00F	16 Kb

III.5.2 Mise en œuvre Logiciel

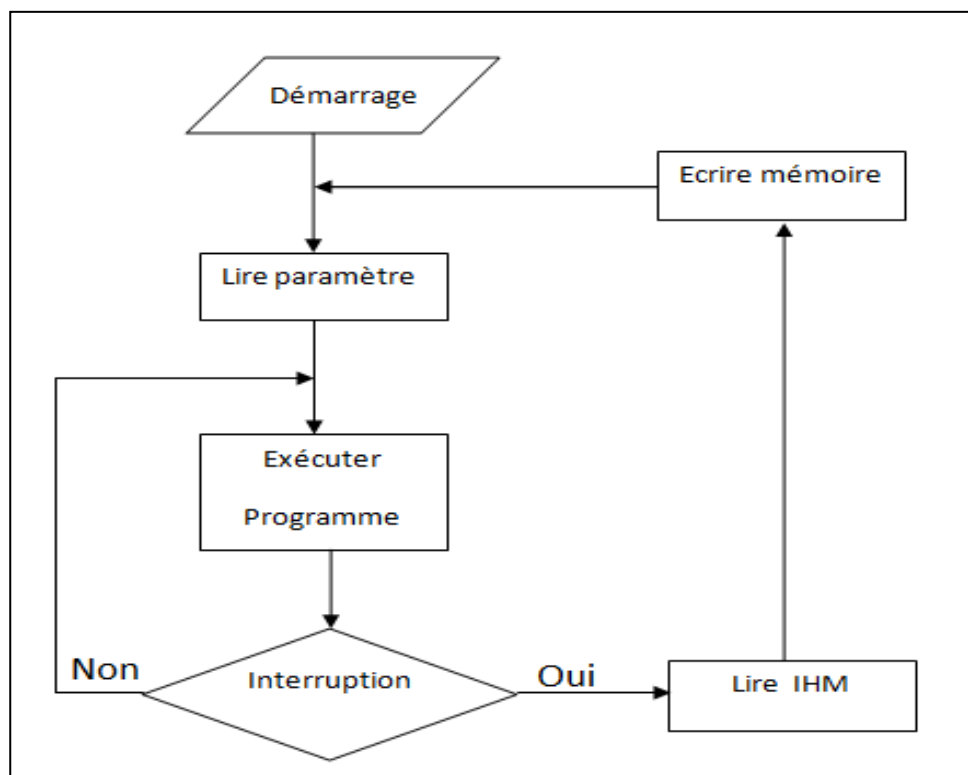
Le pilote de la couche d'abstraction matériel HAL accède à la mémoire flash et Il met en œuvre les opérations d'effacement, de la lecture, d'écriture des mécanismes de protection. L'HAL accélère l'exécution de code avec un système d'instruction préfecture et cache lignes.

Ce pilote fournit des fonctions et macros pour configurer et programmer la mémoire FLASH. Les fonctions de lecture et d'écriture dans le mémoire flash sont présentées dans le tableau suivant ;

Tableau (III.10) : Quelques fonctions du pilote HAL associé à la mémoire flash.

Fonction	Description
HAL_FLASH_Unlock();	Déverrouiller l'accès au registre de commande FLASH
HAL_FLASH_CLEAR_FLAG	Effacer le flag de flash
FLASH_Erase_Sector	Effacer le secteur
HAL_FLASH_Program	écrire, octet, demi-mot, mot ou double mot à une adresse spécifiée avec interruption activée.
HAL_FLASH_Lock	Bloque l'accès au registre de contrôle de FLASH.
Data = *(__IOuint32_t*)(Address);	Lire, octet, demi-mot, mot ou double mot à une adresse spécifiée

La figure suivante représente diagramme d'interaction avec la mémoire flash ;

**Figure (III.9) : diagramme d'interaction avec la mémoire flash.**

III.6 La Communication par le module Sim900a:

La supervision à distance des données se fait par le module GPRS SIM900A avec des commandes AT spécifiées pour configurer ce dernier et pour lui envoyer des données. Ce module, envoie les données reçues de la carte STM32F7 vers un serveur de données distant.

Dans notre travail, nous avons créé une page dynamique sur un serveur déjà existant pour de telles applications.

Parmi les méthodes utilisées pour transférer ces données, la méthode GET qui ajoute les noms des variables des données et leurs valeurs dans l'URL. Elle rend les données visibles dans la barre d'adresse du navigateur, cette donnée définie dans le programme sous forme de vecteur.

Les données sont transmises au serveur web à travers le protocole HTTP ((HyperText Transfer Protocol). Nous avons développé le programme de la figure (III.10) qui permet d'assurer, à travers la liaison série, le transfert des requêtes vers le modules Sim900a en langage AT. Le module Sim900a transmis à son tour l'instruction en HTTP vers le destinataire selon l'adresse incluse.

```

52
53     TM_USART_Puts(USART2,"AT+CGATT?\r\n");//la vérification de la connexion GPRS//
54     HAL_Delay(1000);//delai de 1sec
55     TM_USART_Puts(USART2,"AT+SAPBR=3,1,\"CTYPE\","GPRS"\r\n");//La configuration de mode GPRS//
56     HAL_Delay(1000);//delai de 1sec
57     TM_USART_Puts(USART2,"AT+SAPBR=3,1,\"APN\","djezzy.internet"\r\n");//La configuration d'APN//
58     HAL_Delay(1000);//delai de 1sec
59     TM_USART_Puts(USART2,"AT+SAPBR=1,1\r\n");//La fin de configuration de la mode GPRS//
60     HAL_Delay(4000);//delai de 4sec
61     TM_USART_Puts(USART2,"AT+HTTIPINIT\r\n");// initialisation du protocole HTTP//
62     HAL_Delay(1000);//delai de 1sec
63     TM_USART_Puts(USART2,"AT+HTTIPARA=\"URL\","http://ec2-52-26-70-156.us-west-2.compute.amazonaws.com/
64     add.php?valeur_phy=&valeur_phy?valeur_D=&valeur_D"\r\n");//définit les paramètres du protocole HTTP
65     HAL_Delay(1000);//delai de 1sec
66     TM_USART_Puts(USART2,"AT+HTTIPACTION=0\r\n");//choix la mrthode de l'envoye " GET " //
67

```

Figure (III.10) : Programme de transmission des données vers le serveur web.

Pour établir la communication et l'envoi des données, nous avons réalisé les étapes suivantes :

- 1- Exécuter la commande de vérification de la connexion GPRS du modem sim900a.
- 2- Configurer le mode GPRS0
- 3- Etablir la configuration d'APN, dans ce cas l'APN de Djezzy.
- 4- Initialiser le protocole http.

- 5- Définir et insérer les pointeurs vers les vecteurs des données dans l'URL.
- 6- définir la méthode d'envoi (méthode GET)

L'organigramme général de la supervision est illustré par la figure (III.10) ;

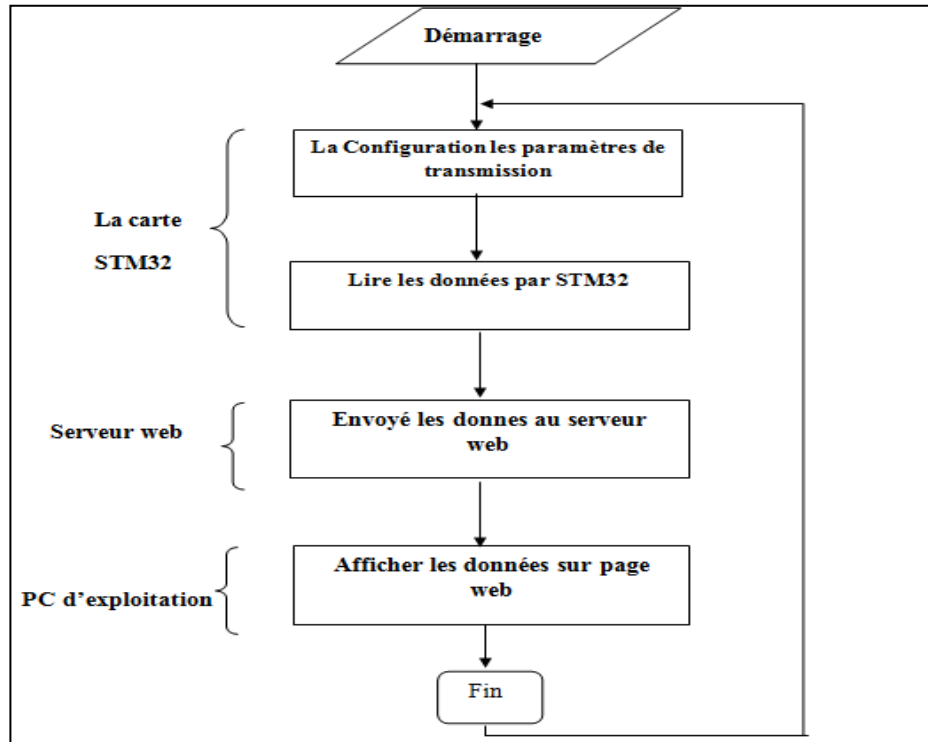


Figure (III.11) : Organigramme générale de la supervision

Au niveau du serveur, les données sont récupérées par le fichier 'add.php' pour être insérer dans le fichier de donnée 'table1.php'.

Pour visualiser les données reçues, nous avons créé une page dynamique 'myapp.php' sous forme de tableau. La figure (III.11) illustre le principe d'échange et d'intégration des données dans le serveur.

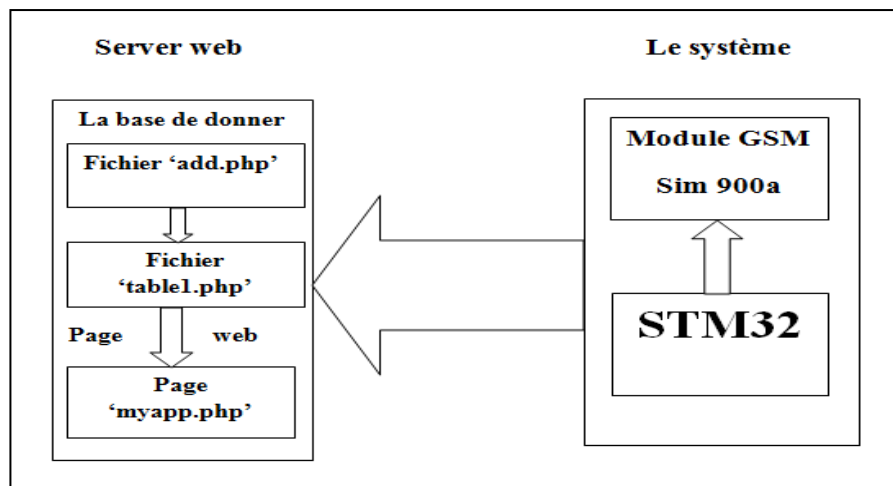


Figure (III.12) : Schéma de transfert et d'intégration des données dans le serveur.

III.7. Conclusion

Dans ce chapitre, nous avons présenté les différentes fonctionnalités liées à la visualisation et la communication. L'IHM offre à l'utilisateur un environnement de configuration et de programmation souple et interactif.

L'intégration de la communication en Modbus/TCP permettra à la carte de s'insérer en toute conformité dans un réseau Ethernet et de communiquer avec les instruments dotés de ce protocole. Les bibliothèques free Modbus et lwIP ont simplifié relativement la tâche d'intégration du protocole. La supervision à distance, en particulier sur internet, est réalisée grâce à l'intégration du module de communication Sim900a.

Chapitre IV :

Réalisation et Tests

IV.1 Introduction

Dans les chapitres précédents, nous avons développé la conception des circuits d'interface ainsi que les différents programmes nécessaires. La réalisation de la carte reste une étape importante dans le projet. En effet, la solution ne sera considérée réalisable et seulement si des résultats expérimentaux de tests ont confirmé sa validité et son efficacité.

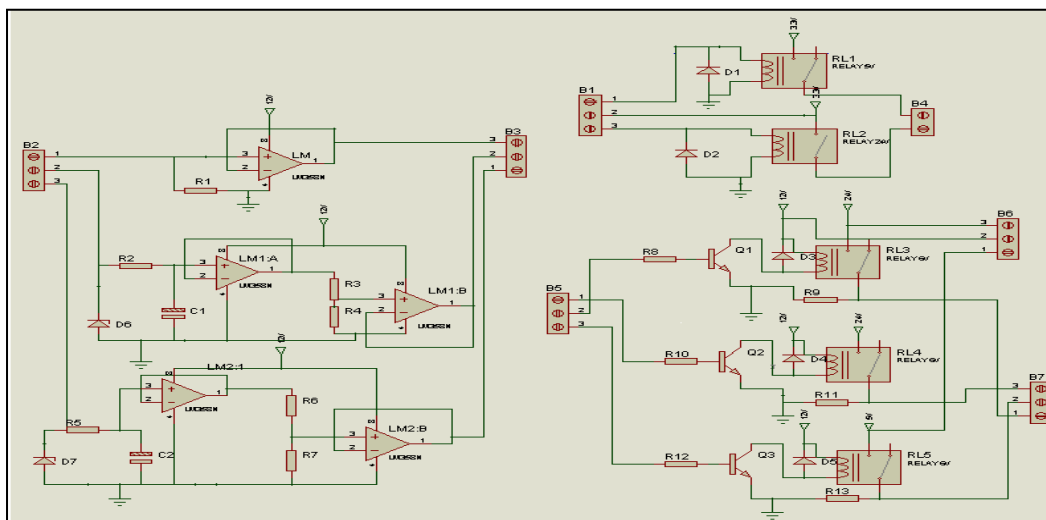
Dans ce chapitre, nous allons présenter la réalisation du circuit imprimé et l'intégration des composants et des modules. Par la suite, nous allons tester les principales fonctionnalités de la carte de contrôle à savoir :

- ✓ L'acquisition des signaux ;
- ✓ L'interfaçage graphique d'IHM ;
- ✓ La transmission à distance à travers le module GSM ;
- ✓ La communication Modbus/TCP

IV.2 La réalisation de la carte

IV.2.1 Réalisation du circuit

Avant de passer de la réalisation pratique, nous avons utilisé Proteus ISIS qui permet de dessiner des schémas électroniques, simuler et réaliser de circuit imprimé correspondant. Ce dernier, [11] nous permet de schématiser notre carte électronique comme le montre la figure(IV.1). Les différents circuits d'adaptation des entrées et des sorties sont illustrés dans cette figure.



Figure(IV.1) : La carte réalisée sur ISIS Proteus

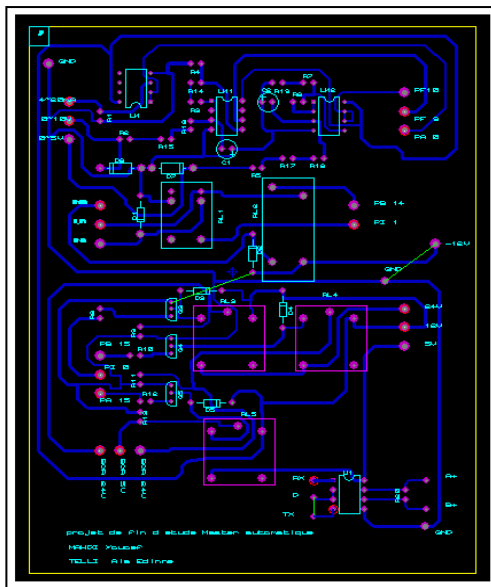
Pour une telle réalisation, nous avons assemblé les différents composants suivants :

- ✓ 5 relies (XM4100 (5V), S0NGLE (6V), DG1U (24V)) ;
- ✓ 13résistances (100 Ω ,165 Ω ,200 Ω ,470 Ω ,1K Ω ,6.5K Ω) ;
- ✓ 2 condensations (4.7 μ F) ;
- ✓ 3 amplifications opérationnels (LM 358N) ;
- ✓ 7 diodes (simple, Zener) ;
- ✓ 3 transistors (BD 177) ;
- ✓ 8 LEDs.

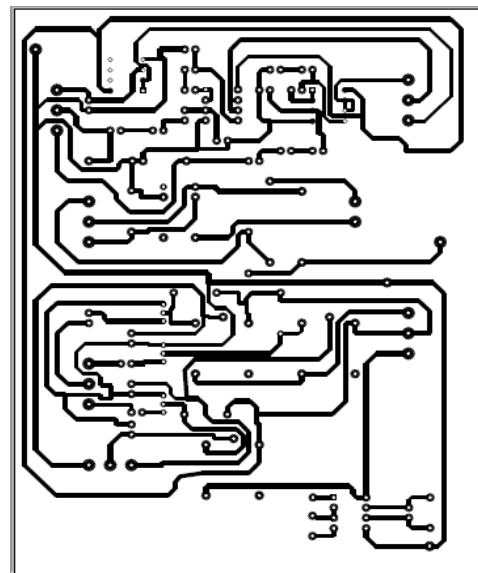
IV.2.2Réalisation du typon (PCB)

Le typon est en effet le négatif de partie cuivrée du circuit imprimé. Il s'agit donc de créer les liaisons entre les différents composants utilisés, avec des pistes en cuivre sur le circuit imprimé.

Proteus ARES nous permet de dessiner des schémas de haute qualité avec la possibilité de contrôle parfaitement l'apparence du dessin tout au long du processus d'édition : largeurs de lignes, style de remplissage, couleurs et polices. Dans notre travail, nous utilisons le logiciel Proteus Ares pour tracer le typon montré par la figure(IV.2).



A : Typon négatif



B : Typon positif

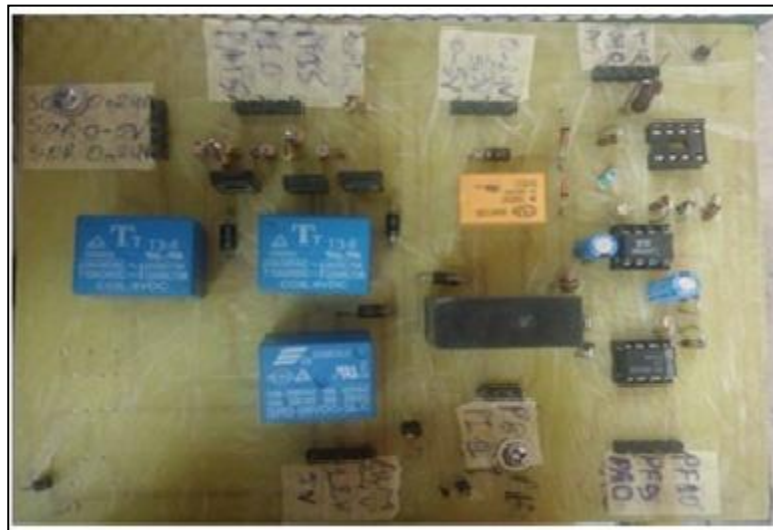
Figure(IV.2) : Typon du circuit imprimé

IV.2.3 Réalisation de circuit et intégration des composants

Pour réaliser une carte électronique nous avons suivi les étapes suivantes :

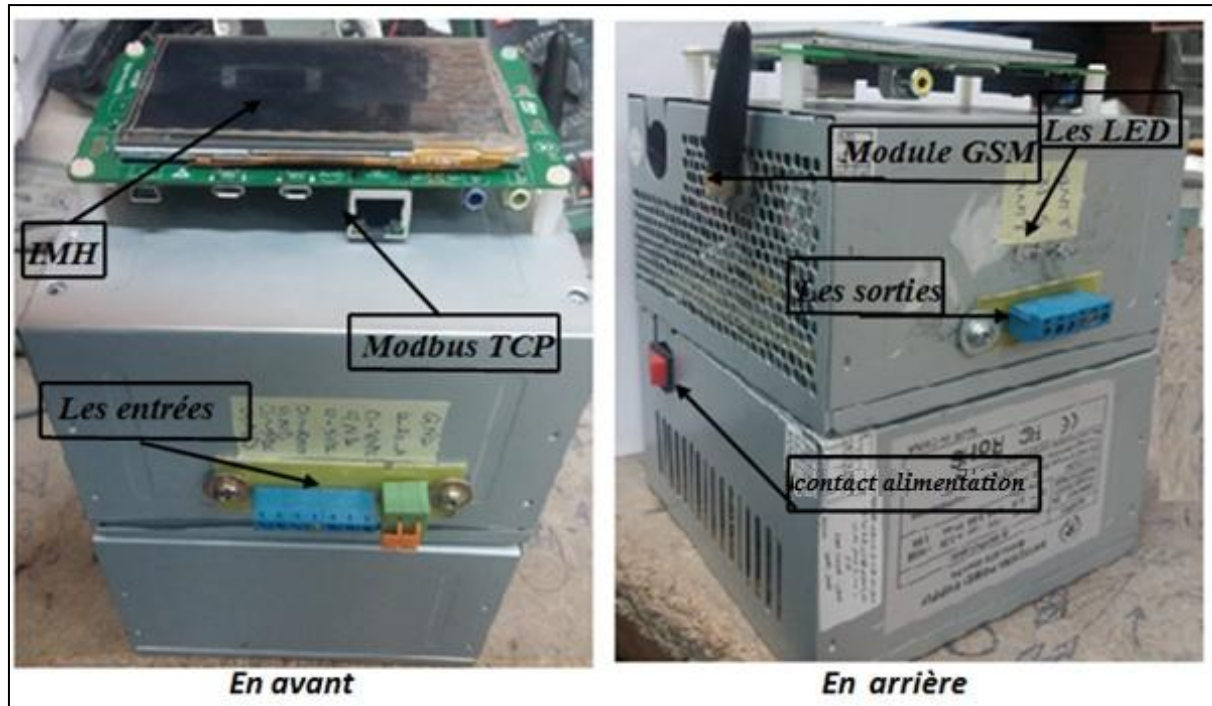
1. Découper la plaque Epoxy/Cuivre aux dimensions du futur circuit imprimé « 11x14.5 cm » ;
2. Mettre le typon dans l'insoleuse en vérifiant bien son sens ;
3. Pour que l'insolation puisse s'effectuer, il faut retirer l'adhésif opaque qui protège la couche photosensible des rayons ultra-violets de la lumière ambiante ;
4. Placer la plaque pré-sensibilisée dans l'insoleuse avec le typon coté cuivre une fois l'adhésif retiré ;
5. L'étape qui suit présente « Une gravure chimique » dans laquelle est utilisé le produit attaquant le cuivre « le perchlorure de fer » ;
6. Enfin, on finalise notre réalisation par deux opérations « perçage + soudage » est cela après le perforage des via ou on implante les différents composants sur leurs emplacements.

Une fois la carte du circuit imprimé est obtenue, nous avons procédé au perçage de la carte et aux placements et soudures des composants. La figure (VI.3) montre la carte d'interfaçage ainsi réalisée.



Figure(IV.3) : La carte d'interfaçages finale avec les composants

A la fin, nous avons assemblée la carte précédente, le carte STM32F7, le module Sim900a avec le boîtier d'alimentation. La figure (VI.4) montre le boîtier final du projet



Figure(IV.4) : Assemblage final de la carte de commande et de communication

IV.3 Les tests expérimentaux :

Dans cette section, nous allons effectuer les tests des différentes fonctionnalités.

IV.3.1 Test 01 : Entrée digitale 0~5 VDC :

Nous appliquons une tension à l'entrée dédiée de niveau 05VDC comme montre la figure (IV.5). Nous constatons la variable associé prend la valeur logique « 1 » ce qui aussi visible sur l'IHM et le LED d'état correspondant s'allume. La situation s'inverse lorsque la tension est ramenée à 0VDC (« 0 » logique). Le LED s'éteint et change de valeur à « 0 » logique dans le programme visible aussi sur l'IHM.

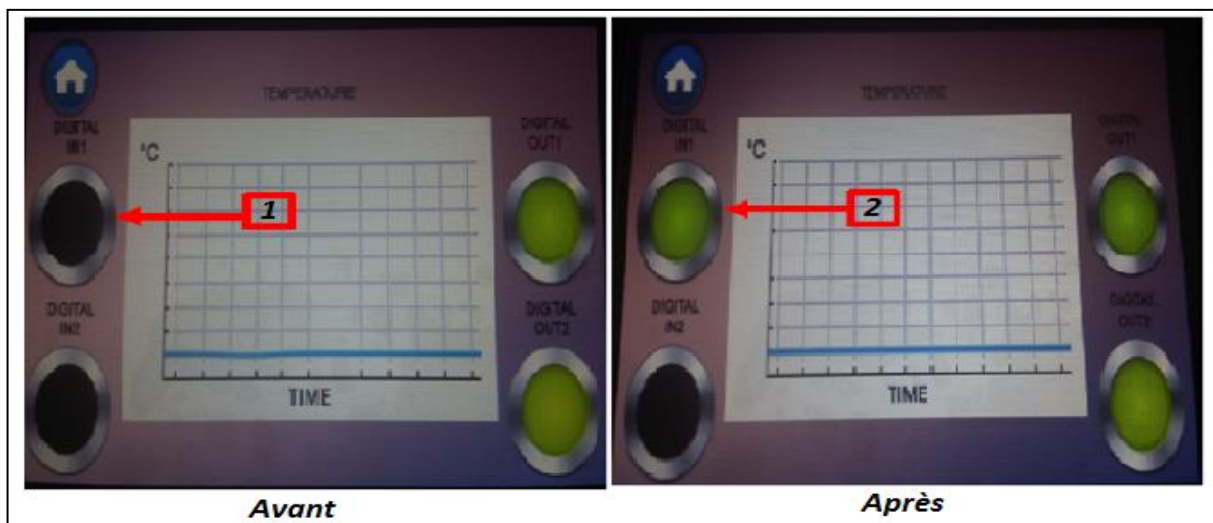


Figure (IV.5) : Test de l'entrée digitale 0~5VDC.

(1) : la lampe éteindre (état 0) (2) : la lampe allume (état 1)

IV.3.2 Test 02 : Entrée analogique (0-5VDC):

L'entrée analogique à testé est représentée sur l'IHM par une courbe qui doit être mise à jour instantanément selon la variation de cette entrée. Le courbe exemple est définie avec un axe de temps en minutes et l'axe des cordonnées en °C afin de simuler une entrée issue de transmetteur de température.

Lorsque nous appliquons à cette entrée une tension de simulation variable entre 0et 5VDC. La figure (IV.6) montre le résultat obtenu, les valeurs acquises s'affiches sur la courbe et varie en temps réels.

Cette entrée est appliquée au circuit d'adaptation, ainsi, après filtrage, un diviseur la ramène au niveau de tension du convertisseur analogique numérique situé au prédéfinie au Pin PA0. Après conversion, elle récupérée par le programme utilisateurs et envoyée vers l'IHM avec une période constante.



Figure (IV.6) : Test de l'entrée analogique 0-5VDC.

IV.3.3 Test 03 : Communication à distance

Dans ce test, nous allons vérifier la liaison et la transmission des données de la carte de contrôle et de communication et un serveur distant sur Internet à travers via le réseau GSM/GPRS. Après la configuration de module GPRS et l'envoi d'un paquet, les données sont récupérées et affichée sous un tableau dans la page du serveur web créé spécifiquement pour cet objectif comme montre la figure (IV.7). La durée totale de la transmission et l'affichage des données est de 04 secondes.

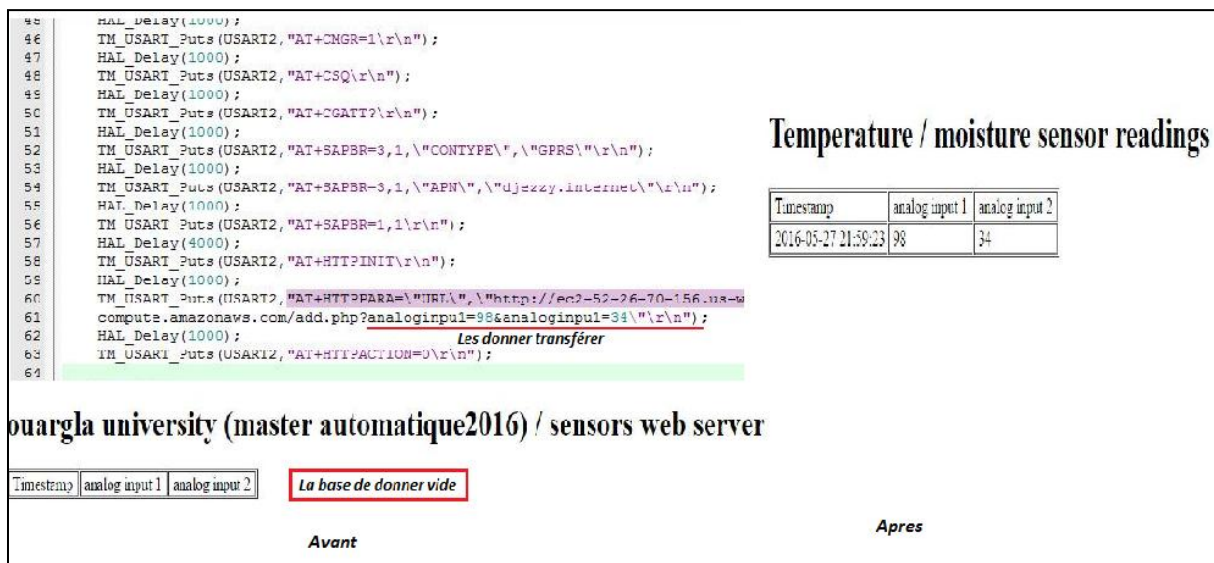


Figure (IV.7) : Test de la transmission à distance

IV.3.4 Test 04 : la communication via Modbus/TCP

Dans ce dernier test, nous allons valider le fonctionnement de la communication via Modbus/TCP. Pour cela, nous devons considérer un esclave Modbus/TCP.

Pour raison de non disponibilité, nous proposons d'utiliser un simulateur de Modbus/TCP esclave.

Au départ, nous configurons l'adresse du maitre dans le logiciel, et l'adresse de l'esclave dans le programme de Modbus. Ensuite, le maitre (la carte STM32F7) envoie une trame ayant l'adresse de l'esclave (1) et le code de fonction (2) qui correspond à l'écriture du bit (3) ou du registre dans ID est (4), qui présentée dans la figure(IV.8).

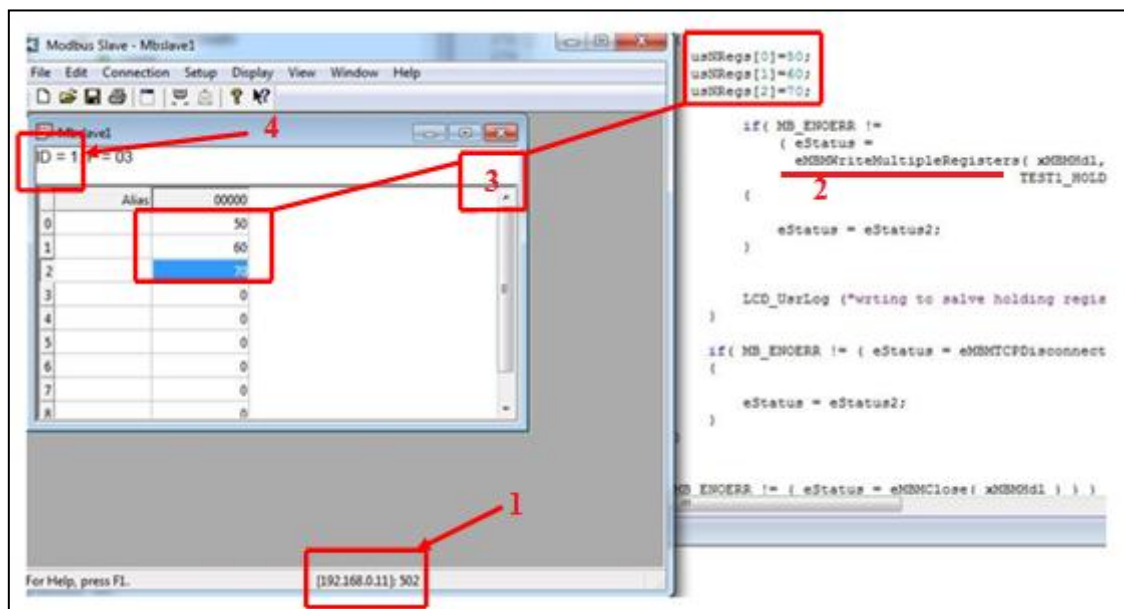


Figure (IV.8) : Test de la communication en Modbus TCP/ écriture.

Comme indique-la figure (IV.9), nous constatons que la fonction est exécutée par l'esclave et le maitre a bien récupérée la donnée dont la valeur est (1).

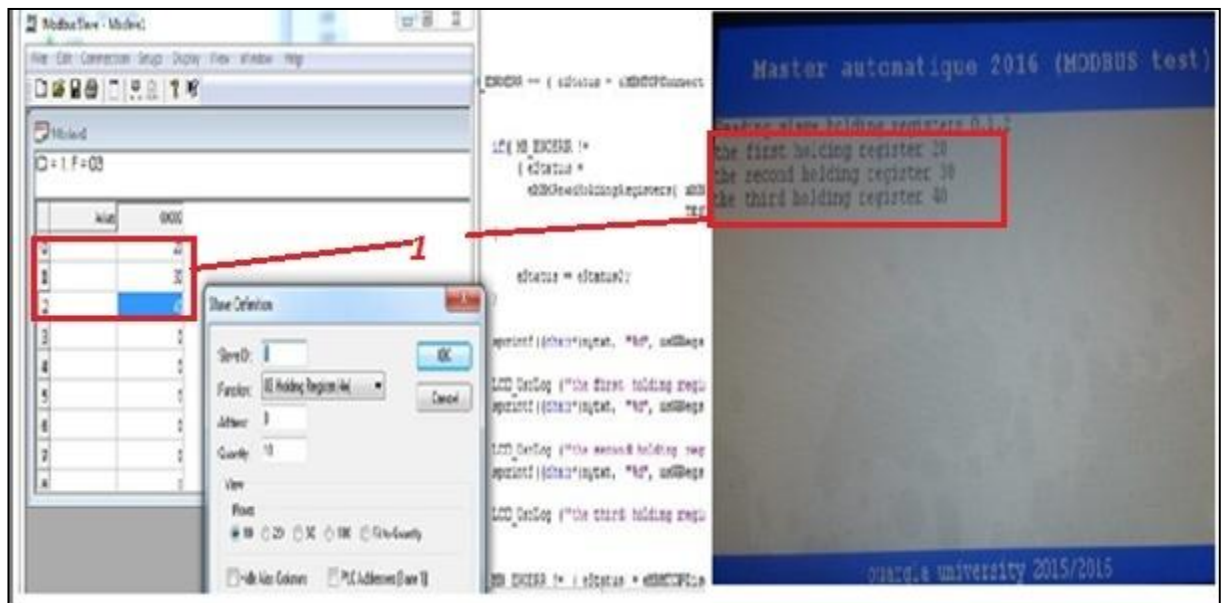


Figure (IV.9) : Test de la communication en Modbus TCP.

IV.4 Conclusion

Les différents tests des entrées et des communications distante et en Modbus/TCP ont montré la validité de l'architecture proposé et de la réalisation effectuée. Grâce à l'IHM les données ont été visualisés ce qui facilitera la tâche de suivi des différentes variables.

Conclusion générale

Conclusion générale

Dans ce projet, nous avons élaboré la conception et la réalisation d'une carte de contrôle et de communication à base de la carte STM32F7.

Dans l'étude de la section contrôle, nous avons développé un circuit d'adaptation des entrées et des sorties pour qu'elles soient compatibles avec les niveaux des tensions de la carte STM32F7. De même, le programme d'acquisition et de traitement a été élaboré. Le programme de gestion de l'IHM, de la communication en Modbus/TCP et en GSM/GPRS a été aussi développé.

La mise en œuvre du programme nécessitait la lecture et l'intégration de plusieurs bibliothèques de fonctions (FreeModbus et lwIP) ainsi que celle de l'implantation de l'IHM. La fonctionnalité de communication à distance a été réalisés en utilisant le langage AT mais après configuration du serveur de données distant.

Les tests expérimentaux ont mis en valeur la validité de la conception et la réalisation. De l'acquisition des entées analogiques et leurs affichage sur l'IHM après conversion, des entrées digitales et visualisation de leurs états via les LED dédiées.

La carte de contrôle développée après configuration s'est intégrée dans une communication en Modbus/TCP.

Enfin, la carte de contrôle développée s'est avérée efficace lors de transfert de données à distance. En effet la liaison de communication a été établie, la trame a été traitée par le serveur et les données envoyées sont insérées dans la base de données.

La carte a été réalisée et placé dans le boîtier dédié tout en ayant accès facile au bornier d'interfaçage (entrées, sorties, Ethernet).

Nous considérons ce travail, une première phase pour la réalisation d'une solution de contrôle et de communication à cout réduit. Dès que les différentes fonctionnalités de bases sont mises à la disposition de l'exploitant, nous proposons dans l'avenir de finaliser la conception de l'interface IHM et de l'enrichir par des fonctions de configurations. Cela, simplifiera l'exploitation de la carte développée.

Bibliographies

Bibliographies

- [1] http://www.st.com/content/st_com/en.html Site de fabricant STM32F7
- [2] RM0090/Reference manual/STM32F407/417
- [3] www.keil.com/ Site de logiciel de programmation
- [4] www.simcomm2m.com Site de fabricant module GSM 900a
- [5] <https://en.wikipedia.org/wiki/STM32>
- [6] UM1907/User manual/Discovery kit for STM32F7 Series with STM32F746NG MCU
- [7] <http://www.tech-gfx.net/forum/> Site de bibliothèque de la programmation graphique
- [8] <http://www.rtaautomation.com/technologies/modbus-rtu/> Site protocol Modbus rtu
- [9] <http://www.modbus.org/tech.php> Site Modbus TCP
- [10] <http://savannah.nongnu.org/projects/lwip/> Bibliothèque Modbus
- [11] Adjerid Hassem-eddine, Tabouchount Ihad; Etude, conception et réalisation de la partie commande d'une tournette (Spin coater) dédiée à la micro-fabrication; Université M'Hamed BOUGARA, Boumerdes (2015) .

الملخص:

يشهد المجال الصناعي عبر العالم تطورا سريعا في أنظمة التحكم و المراقبة، و في هذا المشروع قمنا بدراسة و تصميم نظام مراقبة يعتمد على بطاقة ذكية **DTM32F746NG-DISCO** يقوم بالاتصال السلبي بواسطة بروتوكول **Modbus tcp**، و اتصال لا سلبي مع سرفر و اب عبر مودم **GPRS**. يحتوي هذا النظام على شاشة تفاعلية تسهل من التواصل بين الإنسان و الآلة، بحيث يتم استقبال و معالجة الإشارات التماثلية و الرقمية من مختلف الحساسات الموجودة في النظام و ترجمتها إلى بيانات على الشاشة التفاعلية للبطاقة الذكية، كما يمكن هذا النظام المستخدم من إرسال أوامر تحكم و كذا مراقبة و متابعة حالة النظام من على موقع انترنت صمم خصيصا لهذا الغرض، و النتائج التجريبية المتحصل عليها أثبتت فاعليته في المراقبة و التحكم.

الكلمات المفتاحية : نظام، اتصال، مراقبة، تحكم، **Modbus TCP**، **GPRS**، **STM32F746NG-DISCO**.

Abstract:

Industrial sector had known an increasing development in control systems. In this perspective, the aim of this work is to realize control and communication card based on the microcontroller card STM32F746NG-DISCO. It will ensure both Modbus/TCP protocol and GPRS communication. This card enclose HMI interface that simplify human machine interaction in digital and analog signal acquisition and treatment from various system sensors to present them on HMI. Furthermore, this card will allow the user to send controls and to supervise from web site created to this purpose. The experimental results had shown the effectiveness of the proposed solution.

Abbreviation: system, control, HMI, Modbus TCP, GPRS, STM32F746NG-DISCO.

Résumé :

Le domaine industriel a connu dans le monde entier un développement rapide en systèmes de commande et contrôle. Dans ce contexte, cette étude vise à réaliser une carte de contrôle se basant sur une carte à base de microcontrôleur STM32F746NG-DISCO qui fait la communication avec le protocole Modbus/TCP et la communication à distance avec module GPRS. Ce système renferme un IHM qui facilite l'interaction entre l'homme et la machine, ce qui permet l'acquisition et le traitement des signaux analogiques et digitaux des différents capteurs et leur présentation graphique sur l'IHM. En plus, ce système permet à l'utilisateur d'envoyer des commandes vers les organes et le contrôle ainsi que la supervision à distance par l'utilisation d'un site web créé spécifiquement pour cet objectif. Les résultats expérimentaux obtenus ont montré l'efficacité de ce système dans le contrôle.

Les mots clé : système, contrôle, commande, Modbus TCP, GPRS, IHM, serveur web, STM32F746NG-DISCO