

# وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار- عنابة

Année : 2020

Faculté : Sciences de l'Ingénierat  
Département : Electronique

## MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :

Etude et réalisation d'un système IoT à faible coût  
pour la mesure et le suivi à distance de la Glycémie

Domaine : Sciences et Technologie

Filière : Télécommunications

Spécialité : Réseaux et Télécommunications

Par : Ines ABDELAZIZ et Sassa BOULILA

DEVANT Le JURY

Président	: Mohamed KADDECHE	Prof	UBMA
Directeur de mémoire	: Noureddine DOGHMANE	Prof	UBMA
Examineur	: Seïf Allah el Mesloul NASRI	MCB	UBMA

## Remerciements

En préambule à ce mémoire en vue de l'obtention du diplôme de Master 2 en Réseaux Et Télécommunications, nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

La première personne que nous tenons à remercier est notre encadreur **Mr NOUREDDINE DOGHMANE**, pour l'orientation, la confiance et la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'elle trouve dans ce travail un hommage vivant à sa haute personnalité.

Nos remerciements les plus vifs s'adressent aussi à **Mr MOHAMED KADDECHE** président de notre jury et **Mr NASRI SEIF ALLAH EL MESLOUL** d'avoir accepté d'examiner et d'évaluer notre travail.

Nous exprimons toute notre gratitude à tous les professeurs et les enseignants qui ont collaboré à notre formation depuis notre premier cycle jusqu'à la fin de nos études universitaires.

Nous n'oublions pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenu et encouragé.

## Dédicaces

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance,

C'est tout simplement que : nous dédions ce mémoire de Master à :

A notre tendre Mère : Tu représentes pour nous la source de Tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager.

Tu as

Fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leurs vies et leurs études.

A notre très cher Père : Aucune dédicace ne saurait exprimer L'amour, l'estime, le dévouement et le respect que nous avons toujours pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour notre éducation et notre bien-être.

Ce travail est le fruit de tes sacrifices que tu as consentis pour notre Éducation et notre formation le long de ces années.

A nos très chères proches amies : vos sacrifices, votre soutien Moral et matériel nous ont permis de réussir nos études.

A nos frères et sœurs

A nos Amis d'études

À nos amis

Cette humble dédicace ne saurait exprimer notre grand respect et notre profonde estime.

A tous les membres de notre promotion.

A tous nos enseignants depuis nos premières années d'études

## Résumé

Dans le monde d'aujourd'hui environ 422 millions de personnes sont atteintes de diabète et ce nombre ne fait qu'augmenter au fil des années. Vers 2040 il doit atteindre 700 millions de personnes. Les avancées spectaculaires dans le domaine des technologies de la communication et des objets connectés (IoT) ont permis de développer des systèmes de surveillance en continue de la glycémie des patients diabétiques surtout atteints de diabète type1. Ces capteurs, à l'instar du FreeStyle d'Abbott sont plus efficaces pour la mesure de l'indice glycémique et sont plus faciles à utiliser que les glucomètres standards basés sur des mesures invasives. Cependant, ces solutions restent encore couteuses et imposent des protocoles à champs proches (NFC : Near Field Communication) qui nécessitent l'utilisation de lecteurs appropriés et sur des distances relativement faibles entre le lecteur et le capteur. Dans notre projet de fin d'études nous nous proposons de relever le défi de mettre en œuvre un système IoT de faible coût qui associé avec le capteur Freestyle va permettre d'extrapoler les possibilités de mesure en continue offertes initialement par ce capteur pour les rendre accessibles en ligne et à distance. Ainsi, en se basant sur des équipements utilisant l'environnement Arduino et offrant souplesse et simplicité, nous proposons de réaliser un système qui garantit la surveillance à travers un Android voire même un serveur web.

## Abstract

In the world today approximately 422 million people have diabetes and this number is only increasing year by year. By 2040 it should reach 700 million people. Spectacular advances in the field of communication technologies and internet of things (IoT) have made it possible to develop continuous monitoring systems for the blood glucose levels of diabetic patients, especially those with type 1 diabetes. These sensors, like Abbott's FreeStyle, are more efficient at measuring the glycemic index and are easier to use than standard glucose monitors based on invasive measurements. However, these solutions are still expensive and require near field protocols (NFC: Near Field Communication) which require the use of appropriate readers and over relatively short distances between the reader and the sensor. In our master thesis, we propose to take up the challenge of implementing a low-cost IoT system which, associated with the Freestyle sensor, will make it possible to extrapolate the continuous measurement possibilities, initially offered by this sensor, to make them accessible online and remotely. Thus, based on equipment using the Arduino environment and offering flexibility and simplicity, we propose to create a system that guarantees monitoring through Android or even a web server.

## ملخص

يوجد في العالم اليوم ما يقرب من 422 مليون شخص مصابون بمرض السكري وقد زاد هذا العدد على مر السنين. بحلول عام 2040 سوف يصل الى 700 مليون شخص. أتاحت التطورات المذهلة في مجال تقنيات الاتصال والأشياء المتصلة (IoT) تطوير أنظمة مراقبة مستمرة لمستويات السكر في الدم لدى مرضى السكري، وخاصة المصابين بداء السكري من النوع 1. هذه المستشعرات، مثل Freestyle من Abbott، أكثر كفاءة في قياس مؤشر نسبة السكر في الدم وأسهل في الاستخدام من أجهزة مراقبة جلوكوز الدم القياسية القائمة على القياسات المجتاحة. ومع ذلك، لا تزال هذه الحلول باهظة الثمن وتتطلب بروتوكولات المجال القريب (NFC: Near Field Communication) والتي تتطلب استخدام أجهزة قراءة مناسبة وعلى مسافات قصيرة نسبيًا بين القارئ والمستشعر. في مشروع نهاية الدراسات الخاص بنا، نقتراح مواجهة التحدي المتمثل في تنفيذ الاتصال والأشياء المتصلة (IoT) منخفض التكلفة، والذي سيتيح، المرتبط بمستشعر Freestyle، إمكانية استقراء إمكانيات القياس المستمرة التي يوفرها هذا المستشعر في البداية لجعلها يمكن الوصول إليها عبر الإنترنت وعن بعد. وبالتالي، بناءً على المعدات التي تستخدم بيئة Arduino وتوفر المرونة والبساطة، نقتراح إنشاء نظام يضمن المراقبة من خلال Android أو حتى خادم الويب.

## Liste des figures

Figure	Page
Figure 1 : Le diabète dans le monde	03
Figure 2 : Production naturelle de l'insuline chez l'être humain	03
Figure 3 : exemple d'un glucomètre	04
Figure 4 : Exemple des valeurs du HbA1c	05
Figure 5 : Glucomètre Freestyle libre	05
Figure 6 : Schéma bloc d'un nœud capteur WBAN	06
Figure 7 : Comparatif des normes (rapport puissance/débit)	06
Figure 8 : Schéma de principe du système de surveillance en continu de la glycémie en ligne	09
Figure 9 : Capteur Abbott	10
Figure 10 : Capteur Abbott et son lecteur NFC	10
Figure 11 : Principe d'un système RFID	10
Figure 12 : Brochage du RFID-RC522 avec Arduino Uno	12
Figure 13 : Module HC-05	13
Figure 14 : Brochage du module HC05 avec arduino Uno	14
Figure 15 : Module DFROBOT Bluno V1.4	15
Figure 16 : Module Arduino Tian	16
Figure 17 : Module Arduino/Genuino 101	16
Figure 18 : Module Arduino Nano 33 IoT	17
Figure 19 : Module Beetle	17
Figure 20 : Les deux modules Simple : 7-GPIO et 29-GPIO	18
Figure 21 : NFC BM-019	18
Figure 22 : Pile Li-po 3.7V 100mA et chargeur Li-po	19
Figure 23 : Schéma électrique du système de surveillance de la glycémie	19
Figure 24 : Schéma synoptique du système de surveillance de la glycémie en continu et en ligne	21
Figure 25 : Principe de fonctionnement du capteur Abbott	23
Figure 26 : Schéma synoptique du CR95HF du lecteur NFC	

Figure 27 : Organigramme simplifié du fonctionnement du système IoT	23
Figure 28 : Les principaux systèmes d'exploitation	24
Figure 29 : L'architecture de la plateforme d'Android.	25
Figure 30 : Le SDK Android.	25
Figure 31 : L'émulateur Android	27
Figure 32 : L'aire de développement Android Studio	28
Figure 33 : Architecture de l'interface d'Android studio	28
Figure 34 : Vue d'ensemble de l'application Android réalisée	29
Figure 35 : Fenêtre "Monitor" de l'application Android réalisée	30
Figure 36 : Fenêtre "Alert" de l'application Android réalisée	30
Figure 37 : Algorithme simplifié de la Connexion Bluetooth dans l'application Android	31
Figure 38 : Algorithme simplifié de l'acquisition et l'affichage dans l'application Android	31
Figure 39 : Fonctionnement d'un site web entièrement statique	31
Figure 40 : Fonctionnement d'un site web entièrement dynamique	32
Figure 41 : la communication entre l'application et le serveur web	32
Figure 42 : L'interface de site web	35
Figure 43 : La réception des valeurs de la glycémie par le site web	36
	36

## Liste des tableaux

<b>Tableau</b>	<b>Page</b>
Tableau 1 : Bandes de fréquences RFID	11
Tableau 2 : Câblage des broches (ou Pin)	12
Tableau 3 : Brochage RFD77201/BM019	19

## Liste des abréviations

ADC : Analog to Digital Converter

API : Application Programming Interface

ARC : Argonaut RISC Core

ARM : Advanced RISC Machines

AT : code d'Attention

BLE : Bluetooth Low Energy

CGM : Continuous Glucose Monitoring

COVID 19 : COronaVirus Disease 2019

CRC-16 : Cyclic Redundancy Check-16

ECG : Electrocardiogramme

FRAM : Ferroelectric Random Access Memory

GND : Ground

GPIO : General Purpose Input/Output

GUI : Graphic User Interface

HF : High frequency

HTML : Hypertext Markup Language

I<sup>2</sup>C : Inter-Integrated Circuit

IDE : Integrated Development Environment

IEC : International Electrotechnical Commission

IMC : Indice de masse corporelle

IMU : Inertial Measurement Unit

ISF : interStitial Fluid

ISM band : Industrial, Scientific and Medical band

IoT : Internet of Things

ISO : International Organization for Standardization

JSON : JavaScript Object Notation

LF : Low Frequency

Li-po : Lithium-polymère

mAh : milliampère-heure

NFC : Near Field Communication

OMS : Organisation Mondiale de la Santé

PC : Personal Computer

PHP : Hypertext Preprocessor (originally Personal Home Page)

PIB : Produit Intérieur Brut

PICC : Proximity Integrated Circuit Card

QoE : Quality of Experience

QoS : Quality of Service

RAM : Random Access Memory

RFID : Radio Frequency IDentification

RISC : Reduced Instruction Set Computer

SPI : Serial Peripheral Interface

SQL : Structured Query Language

SRAM : Static Random Access Memory

SSL : Secure Sockets Layer

TLS : Transport Layer Security

UAC : User Account Control

UART : Universal Asynchronous Receiver Transmitter

UHF : Ultra High Frequency

UID : Unique IDentifer,

USB : Universal Serial Bus

UWB : Ultra Wide Band

WAN : Wide Area Network

WBAN: Wireless Body Area Netwrok

WHO: World Health Organization

WiFi : Wireless Fidelity

WLAN : Wireless Local Area Network

WSN : Wireless Sensor Network

# Sommaire

	<b>Titre</b>	<b>Page</b>
	Introduction générale	01
	Chapitre 1 : Problématique et Fondements de base	02
1.	Introduction	02
2.	Position du problème	02
3.	Définitions de la glycémie et le diabète	03
4.	Surveillance de la glycémie chez la patients diabétiques type 1	04
5.	WBAN	05
5.1	Description des WBAN	06
5.2	Protocoles de communications utilisés avec les WBAN	06
6.	Conclusion	07
	Chapitre 2 : Description de la réalisation matérielle	08
1.	Introduction	08
2.	Descriptif du système à réaliser	08
3.	Caractéristiques du capteur FreeStyle Abbott	09
4.	Les systèmes RFID	10
4.1	Transmission de données	11
4.2	Bandes de fréquences RFID	11
4.3	Module RFID RC522	12
5.	Communication Bluetooth	13
6.	Optimisation de la réalisation	14
6.1	Système IoT	15
	<ul style="list-style-type: none"><li>• Module BLUNO</li><li>• Module Arduino Tian</li><li>• Module IoT Arduino/Genuino 101</li><li>• Module IoT Arduino Nano 33</li><li>• Module Beetle</li><li>• Modules Simblee</li></ul>	15 15 16 16 17 17
6.2	Module NFC Reader (Lecteur NFC)	18
6.3	Alimentation de la réalisation	18
6.4	Réalisation complète	19
7.	Conclusion	20

Chapitre 3 : Description de la partie Software	21
1. Introduction	21
2. Programme dans l'environnement de développement intégré Arduino	22
2.1 Environnement de développement intégré Arduino (IDE)	22
2.2 La structure générale de programme Arduino	22
2.3 Lecture et envoie des données de Freestyle	23
2.4 Organigramme générale du programme de IOT	24
3. Création de l'application Android sous l'environnement SDK	25
3.1 Les principaux systèmes d'exploitation d'un smartphone	25
3.1.1 L'Android	25
3.1.2 L'architecture de la plateforme d'Android	25
• Premier niveau : le noyau Linux	26
• Deuxième niveau :les librairies	26
• Troisième niveau : le module de développement d'application	26
• Quatrième niveau : les applications	26
3.2 Développement sous Android	26
3.2.1 L'environnement de développement sous Android	26
3.2.2 Le JDK (Java Développement Kit)	27
3.2.3 Le SDK (Software Développement Kit) Android	27
3.2.4 L'IDE Eclipse	27
3.2.5 Le pulgin ADT pour Eclipse	27
3.2.6 L'émulateur de téléphone (Android Virtual Device)	27
4. Le développement d'application sous Android studio	28
4.1 Architecture d'un projet Android studio	29
4.2 Notre Application Android « DiapApp »	29
5. Le serveur web locale	32
5.1 Le site web	32
5.1.1 Site web statique	32
5.1.2 Site web dynamique	32
5.2 Les étapes de création d'un site web locale dynamique	33
5.2.1 Créer un serveur et une base de données	33
• Définition de XAMPP	33
• Ou 'est ce qui est inclus	33
• Installation de XAMPP	33
• XAMPP-Control-Panel	34
5.2.2 Un langage pour créer un site dynamique	34

• Installation de Python	34
• Création d'un projet Django	34
• Présentation de l'environnement	34
6. La communication entre notre application et le serveur web	35
6.1 Principe de fonctionnement	35
6.2 Notre site web locale	36
7. Conclusion	37
Conclusion générale et perspectives	38
Références bibliographiques	39

## Introduction générale

Aujourd'hui, dans le monde il y'a plus de 422 millions de diabétiques, un taux de mortalité imputé à cette maladie de presque 2 millions d'individus par année, une prévalence mondiale du diabète de l'ordre de 9% et un coût annuel pour son traitement qui avoisine les 500 Millions de dollars américains. Il s'agit d'une pandémie dont les effets néfastes dépassent de loin les plus affreuses des maladies que l'on a connu jusqu'à ce jour malgré que son traitement médical existe et ceci depuis plusieurs décennies. D'après toutes les estimations cette maladie va évoluer et augmenter de plus en plus ces prochaines décennies et ceci est dû surtout à notre mode de vie beaucoup plus sédentaire et à notre hygiène de vie basée sur une très mauvaise alimentation rapide. En effet, les études ont montré que cette maladie comme bien d'autres pathologies est directement liée à l'obésité.

La surveillance des taux de glycémie en continu des diabétiques est parmi les solutions les plus prisées pour pouvoir contrôler efficacement cette maladie. De nos jours les avancées spectaculaires dans les technologies de la communication et des objets connectés, ont permis le développement de capteurs permettant des mesures en continu non invasives de la glycémie des patients atteints de diabète. Parmi ces capteurs qui ont révolutionné ce domaine vital nous avons le FreeStyle d'Abbott. En utilisant le protocole de communication NFC (Near Field Communication) ce capteur permet, à travers son lecteur, de communiquer les valeurs de la glycémie en continu et sans contact. Néanmoins cette technologie de communication, en l'occurrence NFC, reste encore relativement couteuse et ne garantit pas une communication entre le capteur et son lecteur que sur une distance de quelques cm. D'autre part, ce moyen de communication n'est pas présent dans tous les Smartphones et autres tablettes ce qui nous oblige à s'équiper d'un lecteur spécifique pour relever les mesures obtenues et transmises par le capteur.

L'idée préconisée dans ce travail de fin d'études est de mettre en œuvre un système IoT (Internet of Things) capable de lire les données NFC et dotée d'une liaison Bluetooth de faible énergie (BLE) afin de pouvoir les transmettre en continu vers un Smartphone. L'Android pourra afficher les valeurs mesurées, lancer des alertes dans le cas d'une hypoglycémie ou encore une hyperglycémie. Comme il peut également envoyer les mesures vers un serveur web local ce qui va permettre de stocker les données sur des durées importantes ou encore les consulter en ligne à travers l'internet. Cette réalisation, que nous appellerons système de surveillance en continu de la glycémie en ligne, doit répondre à plusieurs contraintes comme par exemple la miniaturisation, la faible consommation énergétique, la simplicité de mise en œuvre et le faible coût de réalisation. Ces exigences vont nous permettre de bien choisir les différents modules de cette réalisation notamment l'IoT ou encore le lecteur NFC.

Notre mémoire de fin d'études est donc divisé en trois chapitres. Le premier chapitre a été consacré essentiellement à la définition de problématique et aux motivations qui nous ont poussé à s'intéresser à un tel sujet. Ce même premier chapitre rappelle succinctement les fondements de base de cette nouvelle technologie liée aux WBAN (Wireless Body Area Network) et aux IoT d'une manière générale. Le second chapitre de notre mémoire présente la partie matérielle de notre réalisation. Après un tour d'horizon sur les solutions possibles à faible coût adaptables pour notre projet, le choix a été fixé sur l'utilisation de cartes Arduino compte tenu des avantages qu'elles offrent. Une première solution a été proposée en utilisant des équipements standards à l'instar d'un Arduino-Uno classique, un lecteur NFC standard et un module Bluetooth HC-05. Même si cette solution est opérationnelle les contraintes posées initialement ne seront pas satisfaites. Ceci nous a conduit à proposer une optimisation matérielle de la réalisation en adoptant d'autres modules plus adaptés à une telle application. Quant au troisième chapitre il concerne la partie software du système proposé. Plusieurs programmes doivent être réalisés en utilisant différentes plateformes de programmation selon l'équipement ou le module à programmer. En effet, nous avons d'un côté le système IoT qui doit être programmé sous un environnement intégré Arduino pour la lecture en continu des données envoyées par le capteur Abbott via l'interface NFC et ensuite pour l'envoi périodiquement de ces mêmes données vers l'Android via une liaison Bluetooth BLE. D'un autre côté une application sous Android SDK doit être mise en œuvre pour permettre au Smartphone de récupérer les données transmises via BLE, de les afficher sur son écran et les transmettre vers un serveur web local. Finalement, un site web approprié doit être réalisé et hébergé dans le serveur local afin de récupérer les mêmes données envoyées en continu par le Smartphone via une liaison Wi-Fi. Le serveur local pourra être consulté à travers internet pour une éventuelle surveillance à distance.

# Chapitre 1 : Problématique et Fondements de base

## 1. Introduction

Plus de 422 millions de personnes sont diabétiques dans le monde et ce nombre devrait augmenter au fil des ans. L'Algérie est l'un des pays où cette pandémie se propage le plus ces dernières années. Le coût financier nécessaire pour le traitement de cette pathologie devient prohibitif. Des recherches rigoureuses et approfondies ont conduit au développement de technologies de plus en plus avancées, telle que la surveillance continue de la glycémie. Ces nouvelles technologies sont plus prometteuses et plus efficaces pour le calcul de l'indice glycémique et sont plus faciles à utiliser que la technologie du glucomètre déjà implantée sur le marché depuis de nombreuses décennies. Cependant, les solutions du marché sont souvent très restrictives en raison de leurs coûts. Dans un effort pour relever ce défi, nous proposons alors à travers ce projet de fin d'études d'octroyer davantage de performances au capteur Freestyle d'Abbott, qui a déjà révolutionné ces dernières années les technologies de surveillance en continue de la glycémie, en réalisant un système IoT capable de mettre les mesures en ligne instantanément.

Dans ce premier chapitre nous commençons par positionner la problématique qui nous a incité à choisir cette thématique comme projet de fin d'études. Dans une seconde partie de ce premier chapitre nous avons présenté les fondements de base et le background nécessaires pour comprendre et maîtriser la technologie IoT et les WBAN représentant la pièce maîtresse de ce projet.

## 2. Position du problème

Selon l'organisation mondiale de la santé OMS (en anglais WHO World Health Organization), le nombre de diabétiques dans le monde est passé de 108 millions en 1980 à 422 millions en 2014 et il dépassera certainement la barre des 700 millions vers 2040. La prévalence mondiale du diabète chez les adultes est passée de 4,7 % en 1980 à 8,5 % en 2014 et sûrement plus de 9% aujourd'hui. Autrement dit, une personne sur onze est diabétique et la moitié des diabétiques n'ont pas été diagnostiqués. Le nombre de décès dans le monde chaque année, lié directement au diabète dépasse aujourd'hui 1,6 millions de personnes (2,2 millions de décès en 2012). C'est énorme pour une maladie dont le traitement existe depuis près d'un siècle. L'OMS prévoit qu'en 2030, le diabète sera la septième cause de décès dans le monde. La mortalité prématurée attribuable au diabète n'évolue pas de la même manière dans les différents pays du monde. Si dans les pays à revenu élevé, le taux de mortalité prématurée imputable au diabète a relativement baissé pendant la première décennie du 21<sup>ème</sup> siècle et il a légèrement augmenté pendant la seconde décennie du même siècle. Dans les pays à revenu intermédiaire, le taux de mortalité prématurée attribuable au diabète n'a fait qu'augmenter au cours de ces deux dernières décennies. En Algérie les chiffres sont plus alarmants encore où on compte plus de 10% de la population sont diabétiques, c'est-à-dire environ 4 à 5 millions de diabétiques et ils seront autour de 9 millions d'ici 2040. Selon l'OMS, l'Algérie est parmi les pays du monde où la maladie a le plus augmenté. Les déterminants de cette maladie sont les changements apparus dans les modes de vie des citoyens. Cette maladie, surtout le type 2, est fortement liée, comme bien d'autres maladies (à l'instar des maladies cardiovasculaires, des cancers du foie et du sein), au surpoids et à l'obésité. En effet, la majorité des études montrent qu'il y a une corrélation entre l'évolution rapide de l'indice de masse corporelle (IMC) et la fréquence de la maladie en particulier chez la femme et les personnes âgées des 2 sexes. En effet, en Algérie, le phénomène de l'obésité connaît une recrudescence de manière épidémique, touchant pas moins 30% de la population féminine et près de 14% de la population masculine algérienne. Quant au surpoids, il touche 52 % de la population selon certaines études menées par le ministère de la santé Algérien. A titre de rappel l'indice de masse corporelle (IMC) mesurée en  $\text{kg/m}^2$ , permet d'évaluer rapidement la corpulence d'une personne simplement avec le poids et la taille, quel que soit son sexe.

$$\text{IMC} = \text{poids en kg} / \text{taille}^2 \text{ en m}^2 \quad (1)$$

- $\text{IMC} < 18,5 \text{ kg/m}^2$  : insuffisance pondérale
- $18,5 < \text{IMC} < 24,9$  : poids normal
- $25 < \text{IMC} < 29,9$  : surpoids
- $\text{IMC} > 30$  : obésité

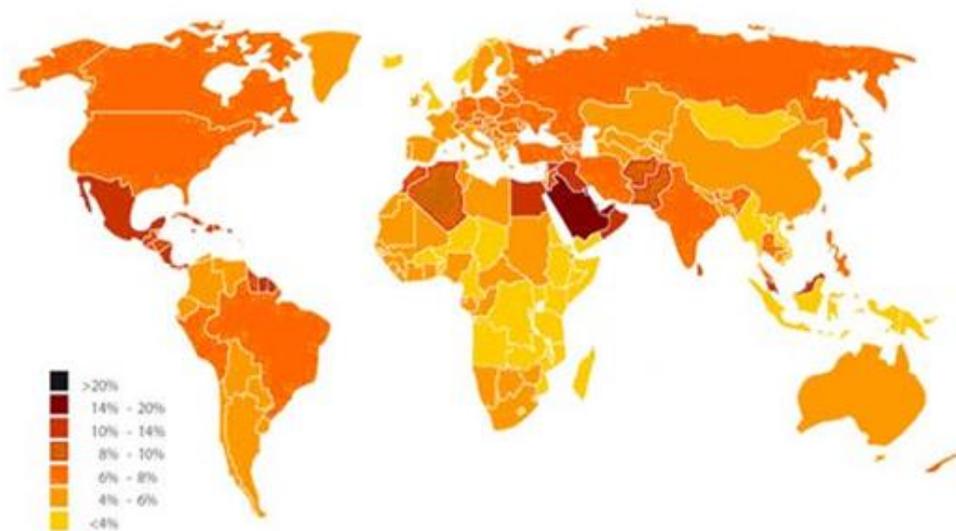


Figure 1 : Le diabète dans le monde

Dans le monde, en 2012, la maladie a coûté 471 milliards de dollars américains, et en 2030, la maladie consommera probablement 595 milliards de dollars américains. En Algérie, le diabète absorbe à lui seul 1,4% du produit intérieur brut (PIB). Le PIB est un indicateur économique permettant de mesurer les richesses produites au sein d'un pays au cours d'une période déterminée.

Il s'agit donc d'une pandémie beaucoup plus grave que la covid-19 et de plus c'est une maladie chronique (qui ne guérit pas mais qui peut être traitée et gérée). A cet effet, plusieurs opérations doivent être menées, aussi bien dans la prévention, le traitement et la surveillance en continue des malades. Ces opérations doivent viser essentiellement à réduire la fréquence de cette maladie, à fournir aux patients un traitement efficace mais aussi à surveiller en continu la variation de la glycémie de ces malades pour pallier aux séquelles très dangereuses de cette pathologie.

### 3. Définitions de la glycémie et le diabète

L'insuline est une hormone qui régule la glycémie (c'est-à-dire le taux de sucre dans le sang) produite normalement par les cellules bêta ( $\beta$ ) (*Beta cell*) des îlots de Langerhans (*Pancreatic islet*) du pancréas, organe situé sous le foie (*Liver*) dans le corps humain. Le diabète, maladie chronique, se caractérise par un défaut de production de cette insuline par le pancréas, soit elle est sécrétée en quantité insuffisante ou encore due à sa mauvaise distribution dans le corps à cause notamment d'un surpoids ou d'une obésité.

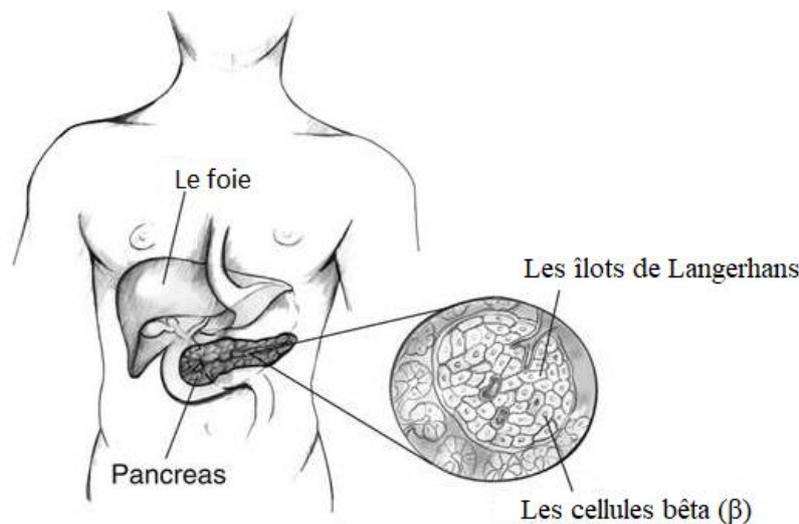


Figure 2 : Production naturelle de l'insuline chez l'être humain

D'après: *National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health*

Il existe donc deux formes de diabète, dues soit à une sécrétion en quantité insuffisante de l'insuline (diabète type 1) ou encore à une mauvaise distribution dans le corps de l'insuline (type 2) :

- **Le diabète de type 1, un diabète insulino-dépendant** se caractérisant par une sécrétion insuffisante d'insuline et entraînant rapidement la mort sans l'administration quotidienne d'insuline. Les symptômes sont, notamment, une sécrétion d'urine et une soif excessive, une faim permanente, une perte de poids, une altération de la vision et une fatigue persistante.
- **Le diabète de type 2, non insulino-dépendant**, est dû à une mauvaise utilisation de l'insuline par l'organisme. **Il représente 90 % des cas de diabète** dans le monde et résulte principalement d'un surpoids et du manque d'exercice physique. Les symptômes peuvent être les mêmes que ceux du diabète de type 1, mais ils sont souvent moins prononcés. De ce fait, il arrive que la maladie ne soit diagnostiquée que plusieurs années après son apparition, alors qu'il existe déjà des complications. Récemment encore, ce type de diabète ne s'observait que chez les adultes, mais maintenant, il touche également les enfants obèses. Cette forme, la plus fréquente du diabète (type 2 : 90% des cas) est favorisée par l'épidémie d'obésité et le manque d'exercice et autrefois essentiellement maladie de l'adulte, est de plus en plus souvent diagnostiquée chez les jeunes.

Concernant le diabète de type 1, le pancréas ne produit pas d'insuline, les patients recourent alors à des injections pour survivre. Quant aux personnes souffrant de diabète de type 2 - représentant 90 % des cas - la synthèse d'insuline se fait normalement mais cette hormone est sécrétée en quantité insuffisante, ou bien elle est utilisée à mauvais escient. Généralement liée au mode de vie, cette dernière affection atteint le plus souvent des patients en surpoids et sédentaires.

#### 4. Surveillance de la glycémie chez les patients diabétiques type 1

La mesure en continue de la glycémie d'un patient atteint d'un diabète d'une manière générale et spécialement le diabétique de type 1 ou insulino-dépendant est très recommandée afin de pouvoir réguler par injection d'insuline leurs taux de glycémies. À cet effet, ils doivent régulièrement contrôler leur glycémie plusieurs fois par jour, s'injecter de l'insuline plusieurs fois par jour, manger de manière équilibrée et pratiquer une activité physique quotidienne. La surveillance de leur glycémie en continue va leur permettre de mieux la contrôler et la gérer. Il existe plusieurs types d'appareils permettant de mesurer les taux de glycémie dans le sang des patients appelés souvent glucomètre.



Figure 3 : exemple d'un glucomètre

Un **glucomètre** ou **lecteur de glycémie** permet de mesurer rapidement le taux de glucose dans le sang (la glycémie), généralement à l'aide d'une goutte du sang capillaire (récupéré sur les bouts de doigts). Mais ces mesures ne peuvent être effectuées que quelques fois par jour et ne peuvent pas montrer la variation journalière de la glycémie du patient. Un autre moyen pour surveiller l'équilibre glycémique des patients diabétiques sur une période beaucoup plus élevée, généralement de trois mois, est l'**hémoglobine glyquée** (fraction **HbA1c**). Il s'agit de la forme glyquée de la molécule d'hémoglobine. Sa valeur biologique permet de déterminer la concentration de glucose dans le sang, la glycémie, sur trois mois. Mais cette mesure s'effectue dans un laboratoire médical et sur des périodes relativement longues (trois mois).

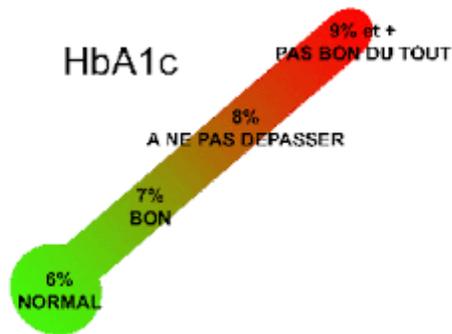


Figure 4 : Exemple des valeurs du HbA1c

De nos jours des méthodes beaucoup moins invasives (sans avoir besoin de récupérer des gouttes de sang à chaque mesure) et en continue sont employées. Elles utilisent des capteurs sans fils souvent de type NFC (Near Field Communication). L'exemple type est le capteur FreeStyle libre d'Abbott. Il s'agit donc de systèmes de mesure de la glycémie interstitielle en continu ou (Continuous Glucose Monitoring CGM) constitués d'un capteur sous-cutané, d'un émetteur et d'un récepteur qui affiche les mesures de glucose recueillies toutes les quelques minutes. Le **liquide interstitiel** : ou **interstitium** a une composition ionique proche de celle du plasma sanguin. Il remplit l'espace entre les capillaires sanguins et les cellules et facilite les échanges de nutriments et de déchets entre ceux-ci.



Figure 5 : Glucomètre Freestyle libre

A travers ces trois exemples de surveillance de la glycémie chez les patients, on remarque clairement l'importante évolution technologique dans ce domaine. La surveillance en continue doit permettre de beaucoup mieux gérer et contrôler cette maladie au quotidien. Ceci, a été possible grâce en particulier aux avancées spectaculaires dans le domaine des capteurs mais aussi et surtout dans le domaine de la communication numérique grâce notamment à la mise en œuvre de des protocoles de communication comme le NFC ou encore BLE. Il s'agit tout simplement d'exemples particuliers de WSN utilisés sur les humains que l'on peut appeler aussi les WBAN.

## 5. WBAN

WBAN pour "Wireless Body Area Network" est un réseau sans fil pour le corps (WBAN) reliant des nœuds indépendants (par exemple des capteurs et des actionneurs) qui sont situés dans les vêtements, sur le corps ou sous la peau d'une personne. Le réseau s'étend généralement sur tout le corps humain et les nœuds sont connectés via un canal de communication sans fil. Selon l'implémentation, ces nœuds sont placés dans une topologie en étoile ou multi-sauts. Un WBAN offre de nombreuses nouvelles applications prometteuses dans le domaine de la télésurveillance de la santé, des soins à domicile / de la santé, de la médecine, du multimédia, des sports et bien d'autres, qui profitent tous de la liberté de mouvement illimitée qu'offre un WBAN. Dans le domaine médical, par exemple, un patient peut être équipé d'un réseau corporel sans fil composé de capteurs qui mesurent en permanence des fonctions biologiques spécifiques, notamment, les constantes physiologiques vitales, telles que la pression artérielle, la fréquence cardiaque, l'électrocardiogramme (ECG), le taux d'oxygénation du sang mais aussi la glycémie, la température, la

respiration, etc. L'avantage est que le patient n'a pas à rester au lit, mais peut se déplacer librement à travers la pièce et même quitter l'hôpital pendant un certain temps. Cela améliore la qualité de vie du patient et réduit les coûts hospitaliers. De plus, les données collectées sur une plus longue période et dans l'environnement naturel du patient, offrent des informations plus utiles, permettant un diagnostic plus précis et parfois encore plus rapide.

### 5.1 Description des WBAN

Les WBAN sont des réseaux reliant essentiellement des capteurs et/ou actionneurs. Ces derniers sont donc au cœur du premier niveau fonctionnel qui correspond à la communication intra-BAN. Le rôle des capteurs consiste à recueillir des signaux analogiques qui correspondent à des activités physiologiques de l'homme ou à des actions du corps. Ce signal analogique est ensuite numérisé puis transmis via un émetteur-récepteur radio. Le nœud capteur, à l'image des WSN (Wireless sensor network) est constitué de plusieurs éléments entre autres la partie captage, un convertisseur Analogique-Numérique, une unité d'alimentation, un processeur, une mémoire et un émetteur opérant selon les normes adoptées par les WBAN. Les nœuds WBAN peuvent être aussi, dans certains cas, composés d'actionneur qui lui un rôle actif vis-à-vis de son porteur, il pourra par exemple délivrer sur commande une substance médicamenteuse. Dans la figure ci-dessous un schéma bloc d'un nœud capteur WBAN.

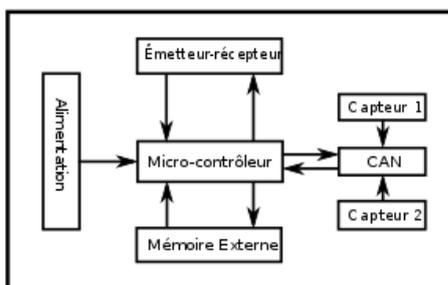


Figure 6 : Schéma bloc d'un nœud capteur WBAN

### 5.2 Protocoles de communications utilisés avec les WBAN

La norme BAN est définie par **IEEE 802.15.6** comme « une norme de communication optimisée pour les appareils à basse consommation et qui fonctionnent sur, dans ou autour du corps humain (mais non limitée aux humains) pour servir une diversité d'applications (y compris médicales), l'électronique grand public, le divertissement et autre ». Ainsi, l'un des objectifs essentiels de cette norme est de garantir une consommation énergétique la plus faible pour assurer une durée de vie des nœuds capteurs WBAN la plus élevée possible.

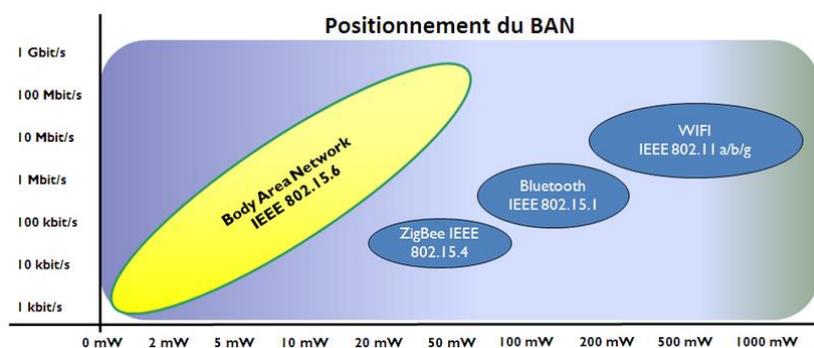


Figure 7 : Comparatif des normes (rapport puissance/débit)

A noter qu'on utilise aussi d'autres normes de communication à faible consommation d'énergie dans les WBAN en particulier le Zigbee, IEEE 802.15.4, Bluetooth et même NFC pour *Near Field Communication* (communication en champ proche).

## 6. Conclusion

Dans ce premier chapitre nous avons commencé par mettre l'accent sur la motivation qui nous a poussé vers ce projet. Il s'agit bien évidemment de l'intérêt qu'on porte de plus en plus aux technologies IoT et aux communications numériques dans le domaine de la santé et plus particulièrement dans la surveillance de certains paramètres physiologiques pertinents. Il s'agit dans ce cas du taux de glycémie permettant ainsi de surveiller les diabétiques surtout de type 1.

D'autre part, nous avons rappelé succinctement et de manière non exhaustive les fondements de base sur les technologies IoT et WBAN. Nous nous sommes surtout intéressés aux moyens de communications utilisés par ces équipements en particulier les interfaces de faibles énergies à l'instar du Bluetooth faible énergie (BLE).

# Chapitre 2 : Description de la réalisation matérielle

## 1. Introduction

Dans ce second chapitre, nous nous sommes focalisés sur la partie matérielle de notre réalisation. Nous avons donc effectué un tour d'horizon sur les solutions possibles à faible coût. Notre intérêt s'est porté sur l'utilisation de cartes Arduino compte tenu des avantages qu'elles offrent comme par exemple : l'environnement de programmation basé sur C++, sa flexibilité, sa disponibilité ...etc. Ainsi, une première solution pour réaliser le système préconisé est alors proposée et réalisée. L'idée de base est d'utiliser un Arduino uno classique muni d'un lecteur NFC standard pour lire en continu les données fournies par le capteur FreeStyle d'Abbott. Ensuite, l'Arduino Uno à l'aide d'une connexion Bluetooth envoie périodiquement ces données vers un Smartphone. A l'aide d'une application appropriée le Smartphone reçoit ces données, les affiche, les surveille et les envoie également vers un Serveur Web local pour que ces données soient visibles sur le net. Dans une seconde partie de notre réalisation nous avons travaillé dans le sens d'optimiser la réalisation proposée de point de vue uniquement matériel. Dès lors, nous avons posé plusieurs contraintes à satisfaire pour le choix des composants dans le but d'optimiser cette réalisation comme par exemple :

- Economie de la consommation énergétique,
- Miniaturisation de la réalisation,
- Garder toujours un environnement de programmation Arduino,
- Réduction du coût de la réalisation.

## 2. Descriptif du système à réaliser

Notre projet de fin d'études est basé sur la réalisation pratique d'un système embarqué à faible coût, à l'instar d'un Arduino ou tout autre microcontrôleur répondant aux exigences de cette réalisation surtout de point de vue faible consommation énergétique et taille réduite. Comme moyen de transmission nous préconisons le protocole '*Bluetooth low energy*' ou BLE. Le système embarqué muni d'un capteur de glycémie mesure instantanément et en continu ce paramètre physiologique (Continuous Glucose Monitoring CGM) et le transmet avec une cadence fixe et choisie d'une manière adéquate vers un Smartphone. Ce dernier, via une application réalisée sous Android, va recevoir ces valeurs pour les afficher sous forme de courbes tout en surveillant d'éventuels dépassement de seuils indiquant une hypoglycémie (valeur de la glycémie  $< 0,7\text{g/l}$  à jeun) ou une hyperglycémie (valeur de la glycémie  $> 1,26\text{g/l}$  à jeun). Ces dépassements seront alors indiqués sous forme d'alarmes pour que le patient puisse prendre les précautions nécessaires. Les valeurs journalières, hebdomadaires voir même mensuelles, seront stockées sous forme d'un historique, par exemple à l'aide d'un site web dédié, permettant au médecin un suivi plus rigoureux et surtout en temps réel de n'importe quel endroit (via internet). Nous souhaitons donc lire ces informations envoyées par le capteur freestyle en NFC et l'envoyer à travers le Wi-Fi vers un serveur (ordinateur ou Smartphone) connecté à internet. En effet, le plus gros handicap d'une NFC est sa très courte portée généralement d'une distance de 4 cm ou moins. Autrement dit, il établit une connexion de très courte portée pour pouvoir lire les données envoyées par le capteur. Nous serons donc amenés d'abord à capter à l'aide d'un lecteur NFC le code envoyé par ce capteur, le traduire ensuite l'envoyer via une connexion wifi vers le serveur. Nous proposons d'utiliser un tout petit système embarqué (le plus petit possible pour réduire la consommation énergétique) muni d'un module BM019 (Serial to Near Field Communication NFC) ou autre équivalent (Wi-Fi par exemple). Ensuite, une mise en œuvre d'un site web pouvant afficher les informations en continu et rendre possible la création d'un système d'administration d'insuline à l'aide d'un algorithme de dosage. Ce système peut, pour le moment fonctionner en « boucle ouverte » en faisant des recommandations à l'utilisateur en définissant automatiquement le débit basal temporaire recommandé.

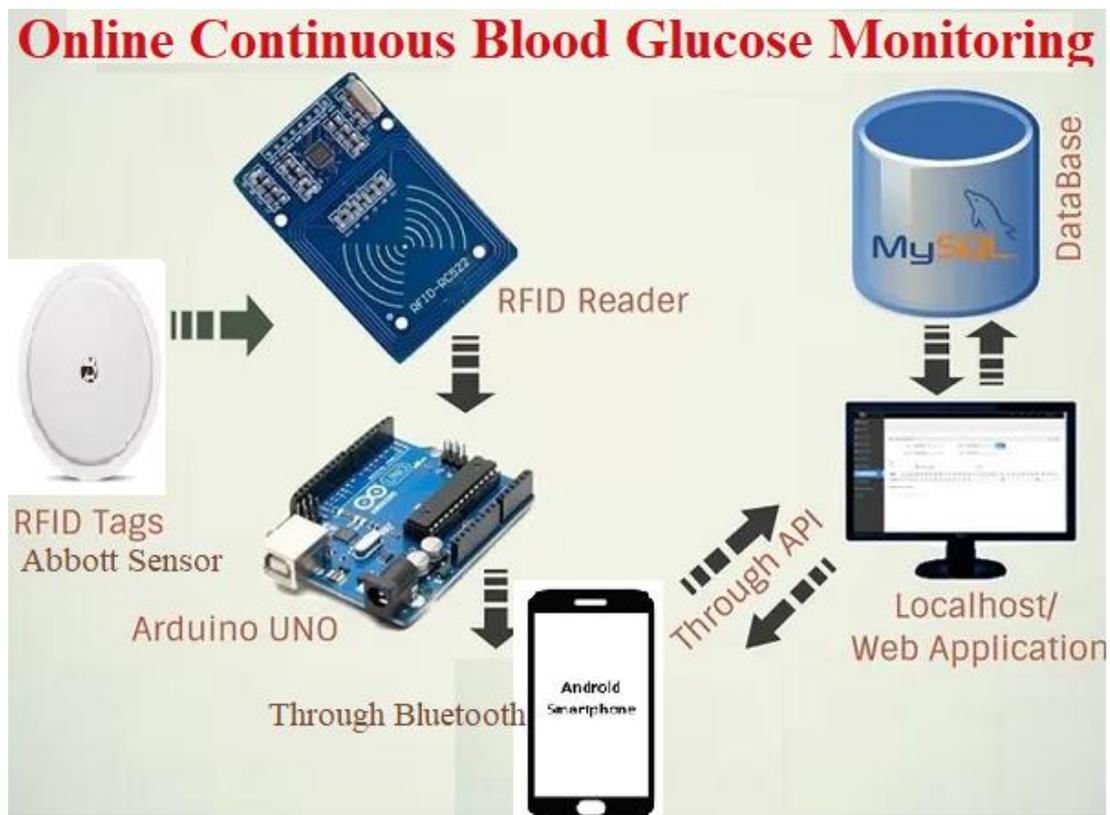


Figure 8 : Schéma de principe du système de surveillance en continu de la glycémie en ligne

### 3. Caractéristiques du capteur FreeStyle Abbott

Le capteur Abbott est équipé d'une puce RFID 13,56 MHz du "Texas Instruments" (RF430). Le dispositif RF430FRL15xH est une puce de transpondeur 13,56 MHz avec un microcontrôleur programmable 16 bits MSP430™ basse consommation. Cet équipement dispose d'une mémoire non volatile FRAM universelle intégrée pour le stockage du code de programme ou des données utilisateur telles que les données d'étalonnage et de mesure. La puce RF430TAL152H contient 244 blocs de 8 octets chacun de mémoire à accès aléatoire ferroélectrique non volatile (FRAM), qui peut être lu via NFC en utilisant une commande de lecture de bloc normale, comme à partir d'un téléphone portable. La FRAM est triée en sections comprenant plusieurs blocs, et chaque bloc est protégé par un CRC-16 dans les deux premiers octets.

Le RF430FRL15xH prend en charge la communication, le paramétrage et la configuration via l'interface RFID conforme :

- A la norme ISO/IEC15693 des Cartes et dispositifs de sécurité pour l'identification personnelle et Objets sans contact de voisinage,
- A la norme ISO/IEC18000-3 Technologies de l'information — Identification par radiofréquence (RFID) pour la gestion d'objets,
- À l'Interface SPI ou I<sup>2</sup>C. Une liaison SPI (pour Serial Peripheral Interface) est un bus de données série synchrone travaillant en mode full-duplex. Les circuits communiquent selon un schéma maître-esclave, où le maître contrôle la communication avec plusieurs esclaves. Quant à la liaison I<sup>2</sup>C (signifie : Inter-Integrated Circuit, en anglais) est également un bus de communication permettant de relier facilement un microprocesseur et différents circuits, etc.



Figure 9 : Capteur Abbott

Les mesures du capteur sont prises en charge par le capteur de température interne et le convertisseur analogique-numérique (ADC) 14 bit sigma-delta embarqué, et le balayage numérique du capteur doit être connecté via SPI ou I2C. L'appareil RF430FRL15xH est optimisé pour un fonctionnement en mode entièrement passif (sans batterie) ou alimenté par batterie à une seule cellule (semi-actif) pour prolonger la durée de vie de la batterie dans les applications de détection portables et sans fil. FRAM est une mémoire non volatile qui combine la vitesse, la flexibilité et l'endurance de la SRAM avec la stabilité et la fiabilité de la mémoire flash, le tout pour une consommation d'énergie totale inférieure.



Figure 10 : Capteur Abbott et son lecteur NFC

#### 4. Les systèmes RFID

RFID (*Radio Frequency Identification*) ou identification par radiofréquence et est un protocole de communication radiofréquence utilisé dans divers domaines notamment pour identifier des objets. Dans une configuration RFID générale, il y a un lecteur RFID et une étiquette (transpondeur) qui doivent être identifiés.

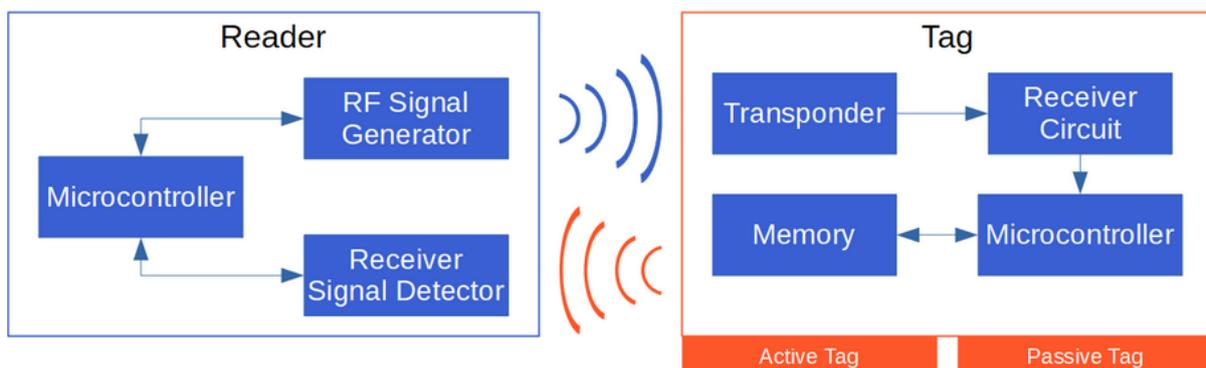


Figure 11 : Principe d'un système RFID

Le lecteur (Reader) se compose de trois parties. Le générateur de signal radiofréquence génère le champ électromagnétique à travers une bobine qui est envoyée aux étiquettes de puissance. Il existe également un récepteur et un détecteur de signal pour lire la réponse de l'étiquette. Le microcontrôleur est la tête du lecteur et contrôle les autres composants et traite en outre les informations. L'étiquette a un transpondeur qui reçoit les ondes radio du lecteur et renvoie également des données au lecteur. Le circuit récepteur stocke l'énergie dans l'étiquette pour alimenter le microcontrôleur. Le microcontrôleur lui-même est connecté à une mémoire où sont stockées les informations qui peuvent être transmises au lecteur. Il existe deux types fondamentaux de systèmes RFID :

- Système passif où les étiquettes n'ont pas de source d'alimentation interne et sont donc alimentées par le lecteur RFID par ondes radio.
- Système actif où les étiquettes sont alimentées par une batterie et peuvent être lues sur une plus longue distance jusqu'à des centaines de mètres.

#### 4.1 Transmission de données

Une fois que l'étiquette (Tag) est alimentée par la tension d'induction du lecteur RFID, l'étiquette envoie des données au lecteur via une manipulation de charge. La charge est activée et désactivée sur l'étiquette. Du fait que le lecteur et l'étiquette sont couplés par induction, une modification de la charge au niveau de l'étiquette entraîne une modification de la consommation d'énergie de l'antenne du lecteur. Ce changement de consommation est représenté par une chute de tension et interprété comme 1 et 0. Une autre approche de transmission de données est la transmission de données via un couplage rétrodiffusé. L'étiquette crée un deuxième champ électromagnétique avec la puissance à travers le premier champ. Ce deuxième champ est reçu par le lecteur et par des changements dans le deuxième champ électromagnétique, des données sont transmises. Normalement, les premières données transmises de l'étiquette au lecteur sont l'UID et le type PICC (Proximity Integrated Circuit Card) qui identifie l'étiquette. UID : L'UID est l'identifiant unique de l'étiquette par exemple : «BD 31 15 2B ». L'UID est enregistré dans le premier bloc de mémoire de la balise et en lecture seule. Par conséquent, l'identifiant ne peut pas être modifié.

#### 4.2 Bandes de fréquences RFID

En général, il existe différentes bandes de fréquences dans lesquelles l'application RFID peut fonctionner. Le tableau suivant montre les différentes bandes de fréquences.

<b>Bands</b>	<b>Regulations</b>	<b>Range</b>	<b>Data Speed</b>	<b>Remarks</b>
120–150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collection
13.56 MHz (HF)	ISM band worldwide	10 cm–1 m	Low to moderate	Smart cards, ISO-non-compliant memory cards, ISO-compatible microprocessor cards
433 MHz (UHF)	Short range devices	1–100 m	Moderate	Defense applications, with active tags
865–868 MHz (Europe)	ISM band	1–12 m	Moderate to high	EAN, various standards; used by railroads
902–928 MHz (North America) UHF				
2450–5800 MHz	ISM band	1–2 m	High	802.11 WLAN, Bluetooth standards
3.1–10 GHz	UWB Ultra wide band	Up to 200 m	High	Requires semi-active or active tags

Tableau 1 : Bandes de fréquences RFID [https://en.wikipedia.org/wiki/Radio-frequency\\_identification](https://en.wikipedia.org/wiki/Radio-frequency_identification)

### 4.3 Module RFID RC522

Dans ce projet, nous utilisons dans un premier cas le module RC522 comme lecteur et également différentes balises compatibles avec ce lecteur. Il est important que la fréquence corresponde et que le lecteur prenne en charge le format de l'étiquette. Le module RC522 est basé sur la carte Philips MF522-AN-on et prend en charge une fréquence RFID de 13,56 MHz ainsi que les formats d'étiquettes suivantes : S50, S70 Ultralight, Pro, DESFire. Le RC522 est connecté au microcontrôleur Arduino ou ESP8266 via SPI ce qui permet une vitesse de transmission entre lecteur et microcontrôleur jusqu'à 10 Mbit / s.

La consommation de courant est très faible entre 80  $\mu$ A et 26 mA :

- Transmission de données : 13... 26 mA
- État de repos : 10... 13 mA
- État de veille : 80  $\mu$ A

La balise est un transpondeur Mifare avec 1 Ko de mémoire qui est divisé en 16 secteurs, chaque secteur en 4 blocs et chaque bloc peut stocker 2 octets de données. Chaque secteur a deux touches (A / B) qui peuvent être attribuées avec des droits de lecture et d'écriture de blocs de mémoire. Le dernier bloc contient les clés et les droits (conditions d'accès).

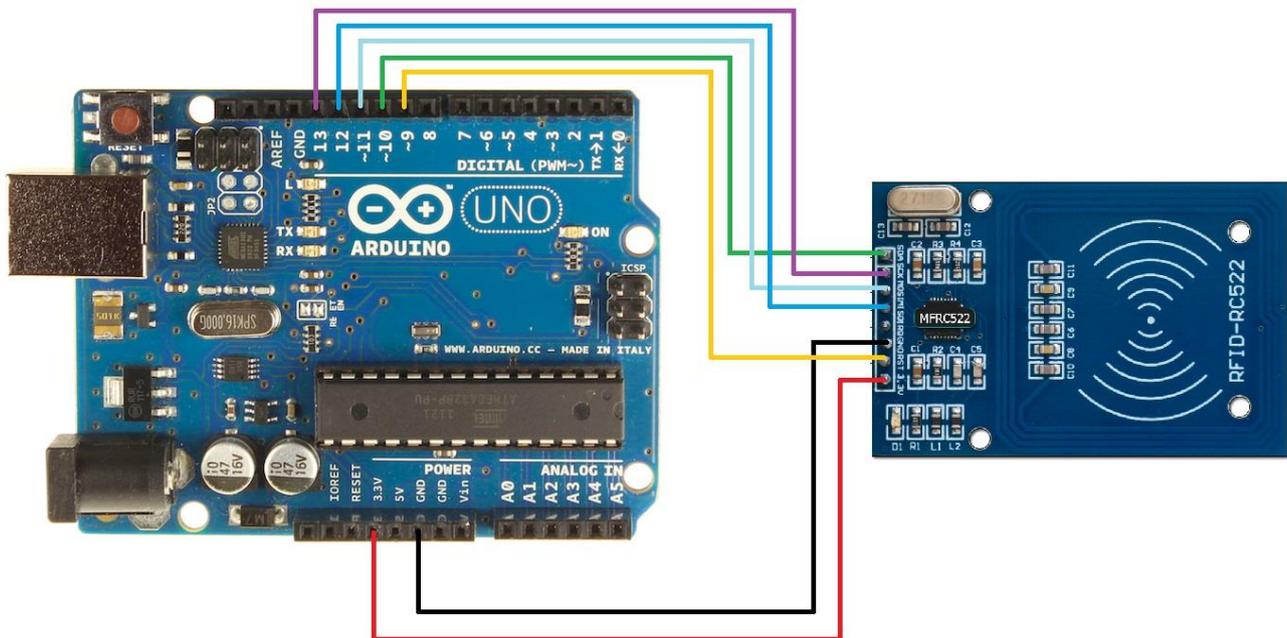


Figure 12 : Brochage du RFID-RC522 avec Arduino Uno

Pin du RC522 reader	Câblage vers l'Arduino Uno
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	Non connecté
GND	GND
RST	Digital 9
3.3V	3.3V

Tableau 2 : Câblage des broches (ou Pin)

## 5. Communication Bluetooth

Le Bluetooth d'une manière générale est un standard de communication permettant l'échange de données bidirectionnel à très courte distance en utilisant des ondes radio UHF (Ultra Haute Fréquence). Le module Bluetooth HC05 permet de doter l'Arduino Uno, ou tout autre microcontrôleur, d'une liaison Bluetooth. La connexion entre le microcontrôleur et le module Bluetooth est une communication série TTL. Le HC-05 possède les caractéristiques suivantes :

- Alimentation : 3,6 à 6 Vcc (régulateur 3 Vcc intégré)
- Intensité 30 mA en fonctionnement, 8mA pour établir la communication
- Bluetooth : 2.0
- Portée : jusqu'à 10 mètres
- Liaison série : 4800 à 1382400 bauds
- Antenne : 2,4 GHz intégrée
- Sorties: Key, Vcc, Gnd, TXD, RXD et State.
- Configuration via commandes AT
- Longueur du cordon : environ 20 cm
- Dimensions : 35 x 16 x 12 mm

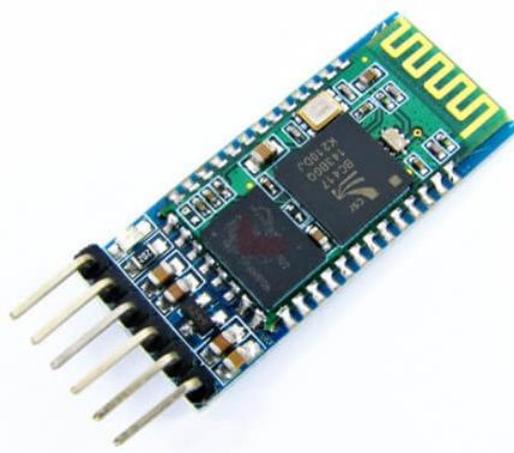


Figure 13 : Module HC-05

Contrairement au module HC-06 qui ne peut fonctionner qu'en mode esclave, le HC-05 peut travailler aussi bien en mode maître (master) qu'en mode esclave (slave). En mode maître le HC-05 peut ainsi se connecter de manière autonome à un autre module Bluetooth esclave équipant une autre carte arduino, un PV via UART (Universal Asynchronous Receiver Transmitter), ou même un Smartphone. L'un ou l'autre de ces deux modes de fonctionnement, comme bien d'autres paramètres de configuration à l'instar de la modification du débit de transmission en baud ou encore ou encore code d'appareillage ...etc., peut être configuré en utilisant les commandes AT (Code d'attention). A noter, que toutes les paramètres du HC-05, pouvant être configurés par les commandes AT, ont des configurations par défauts. Par exemple par défaut le HC-05 est configuré en maître, avec une vitesse de transmission 38 400 bauds et un code d'appareillage correspondant à 1234 ...etc.

Quant au module Arduino-Uno, il dispose d'une seule interface de communication série asynchrone avec les broches étiquetées TX et RX qui sont respectivement la ligne d'émission série (T pour Transmit, émettre) et la ligne de réception série (R pour receive, recevoir). Cette interface est principalement employée pour dialoguer avec l'ordinateur hôte. Des fonctions permettent d'afficher des messages dans une fenêtre de l'ordinateur hôte et de lire le clavier de l'ordinateur hôte. Mais elle est également employée pour dialoguer avec certains modules comme les modules radio XBee ou certains modules Wi-Fi ou encore les modules Bluetooth HC-05 ou HC-06. Notons, que nous pouvons utiliser d'autres broches, étiquetées digital

input/output de l'Arduino pour assurer la communication avec ces modules de communications à conditions de les configurer dans le programme de l'Arduino en intégrant certaines bibliothèques.

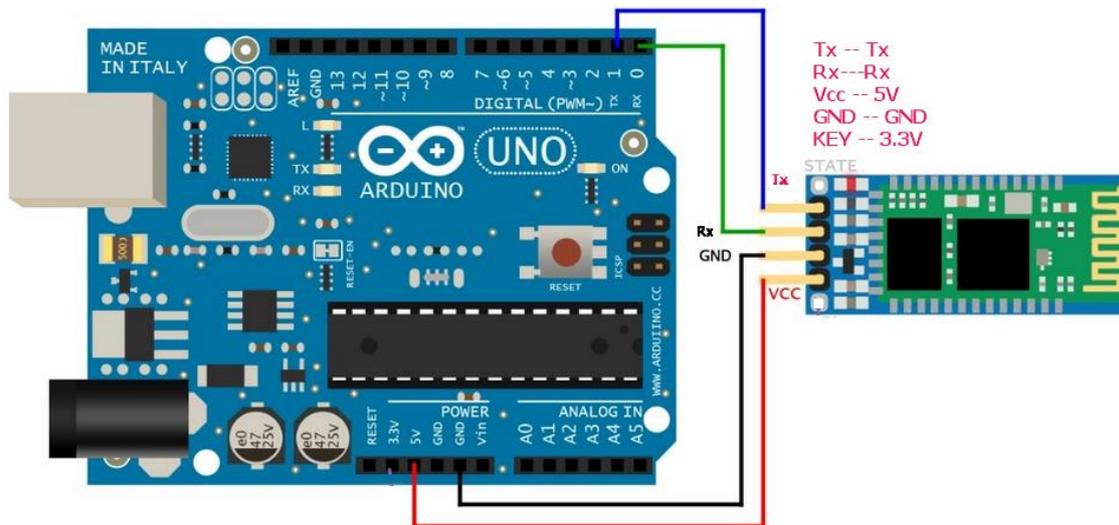


Figure 14 : Brochage du module HC05 avec arduino Uno

L'Arduino-Uno doté d'un module HC-05, convenablement configuré et programmé, pourra communiquer avec un Smartphone via une application appropriée. Le principe final de la réalisation est que le module Arduino acquiert périodiquement les mesures de taux de glycémie par le capteur Abbott en utilisant une liaison NFC. Ensuite, l'Arduino transmet ces mesures via une liaison Bluetooth vers le Smartphone qui pourra ainsi afficher les mesures sous forme de courbes et lancer des alarmes en cas d'hypoglycémie ou hyperglycémie. Le Smartphone doit aussi envoyer ces données vers un site web

Cependant, cette réalisation présente deux inconvénients majeurs à savoir :

1. Les dimensions des différents modules utilisés dans cette réalisation, à l'instar de l'Arduino-uno, le HC-05 et le NFC reader, rendent son exploitation réelle impossible. En effet, il s'agit d'un circuit électronique qui doit être miniaturisé au maximum afin de pouvoir le porter sans aucune gêne au niveau par exemple du bras.
2. L'importante consommation électrique des différents modules utilisés, notamment l'Arduino Uno et même le module Bluetooth HC-05, ne permet pas l'utilisation d'une batterie électrique miniature pour une autonomie énergétique acceptable.

## 6. Optimisation de la réalisation

Ceci, nous a donc incité à améliorer notre réalisation en essayant d'utiliser d'autres modules dont la taille et la consommation électrique sont acceptables pour une telle réalisation. Nous nous sommes alors dirigés vers l'utilisation de modules IoT possédant déjà une liaison Bluetooth à faible énergie (BLE). En effet, l'Internet des objets (IoT) sont souvent des petits équipements accessibles via Internet via plusieurs types d'interfaces à l'instar du Wi-Fi, 3G/4G/5G, du BLE, utilisés dans diverses applications aussi différentes les unes que les autres.

Les contraintes que nous nous sommes imposés pour le choix de notre nœud IoT peuvent être résumées sur quatre critères :

- L'IoT choisit doit intégrer, entre autres, une connexion BLE appelée aussi Bluetooth 4.0. D'abord le Bluetooth est la communication qui doit être utilisée pour envoyer les données de l'IoT vers le Smartphone. D'autre part, nous devons utiliser la variante BLE (Bluetooth Low Energy) pour réduire au maximum la consommation énergétique du nœud et ainsi augmenter l'autonomie de sa batterie. Comme le BLE. Comme rappel, le BLE, tout autant que le Bluetooth

classique, utilise une bande de liaison radio de 2,4GHz et avec un débit de l'ordre de 1Mbps. La grande différence entre ces deux interfaces est que la consommation énergétique du BLE est 10 fois moins importante que le Bluetooth classique.

- L'IoT qui doit s'adapter à notre réalisation doit avoir une dimension parmi la plus petite possible. Il va jouer le rôle d'un WBAN fixé au niveau du bras du patient et par ailleurs il ne doit créer aucune gêne.
- Nous souhaitons opter pour un IoT basé sur l'Arduino. Ceci, est dû à la relative maîtrise de ce type de plateforme aussi bien de point matériel que logiciel. Comme nous avons déjà mis au point les programmes pour notre première application, nous aimerions alors pouvoir rapidement et facilement les utiliser pour cette dernière proposition. Notons, que nous pouvons aussi garder un module Arduino-uno et lui rajouter un module BLE.
- Le coût final de la réalisation doit être abordable. En effet, il s'agit avant tout d'un système à faible coût.

## 6.1 Modules IoT

Notre modeste recherche nous a permis de trouver plusieurs cartes IoT qui répondent plus ou moins à nos exigences et nos contraintes. Parmi ces modules IoT nous avons :

- **Module BLUNO** : Appelé aussi Dfrobot avec des dimensions de l'ordre de 60×53 mm. Il s'agit tout simplement d'un module Arduino avec les mêmes caractéristiques que l'Arduino-uno où on a intégré dedans une interface BLE. A noter que nous trouvons également d'autres arduino classiques avec l'extension BLE à l'instar de Bluno-Nano ou encore Bluno-Mega 2560 ...etc. Ce qui est intéressant le plus dans la version Bluno-Nano par rapport aux deux autres versions Bluno citées ci-dessus est surtout les dimensions de l'ordre de 53×19×12 mm pour un poids qui ne dépasse pas les 20g.

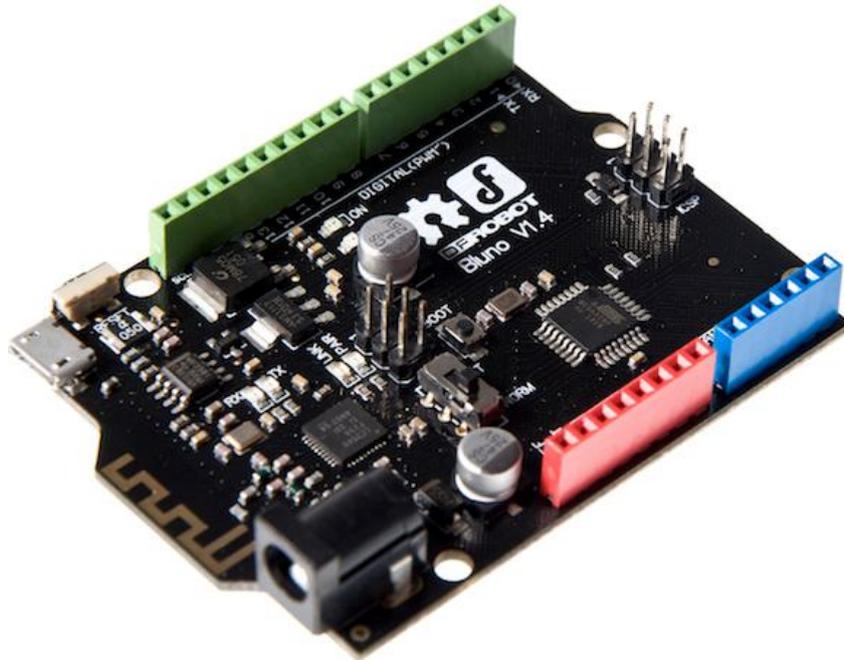


Figure 15 : Module DFROBOT Bluno V1.4

- **Module Arduino Tian** : Possédant des caractéristiques meilleures que le module Bluno Uno. Ses dimensions sont par contre assez comparables au précédent de l'ordre de 53×68 mm. De plus il dispose de connexion Wi-Fi et Ethernet.



Figure 16 : Module Arduino Tian

- **Module IoT Arduino/Genuino 101** : La carte Arduino/Genuino 101 affiche les mêmes dimensions et les mêmes périphériques que la célèbre carte Arduino UNO (68,6 × 53,4 mm) mais elle dispose en sus et en standard de la connectivité Bluetooth Low Energy et d'un accéléromètre/gyroscope 6 axes. Des fonctionnalités qui sont en fait intégrées au sein même du module Curie embarqué, bâti autour d'un processeur Quark à architecture x86 et d'un cœur ARC, tous deux cadencés à 32 MHz.

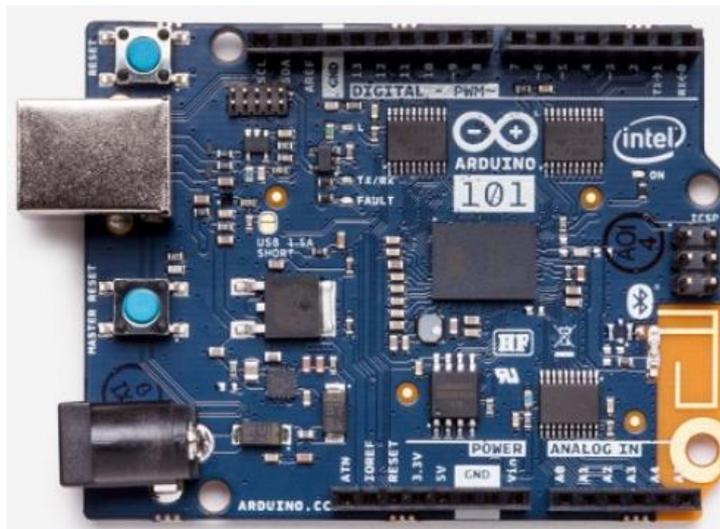


Figure 17 : Module Arduino/Genuino 101

- **Module IoT Arduino Nano 33** : Ce petite module carte robuste et puissante dispose d'une connectivité Wi-Fi et Bluetooth BLE qui, combiné à son architecture basse consommation, en fait une solution pratique et rentable pour les projets connectés. Arduino Nano 33 IoT est entièrement compatible avec l'Arduino IoT Cloud et prend en charge le transport sécurisé TLS (Transport Layer Security) complet (TLS est un protocole successeur du SSL ou "Secure Sockets Layer"). Arduino Nano 33 IoT héberge un processeur Arm Cortex-M0 + SAMD21, un module Wi-Fi et Bluetooth basé sur ESP32, une unité de mesure inertielle (IMU) 6 axes et une puce cryptographique qui peut stocker en toute sécurité certificats et clés pré partagées.

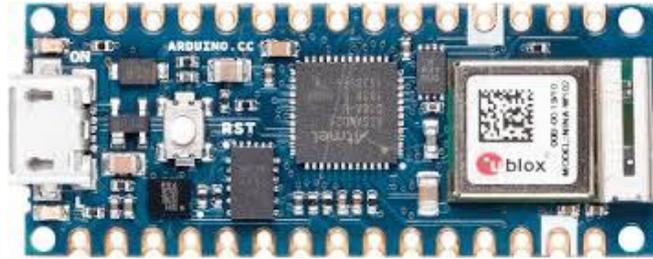


Figure 18 : Module Arduino Nano 33 IoT

- **Module Beetle** : Avec des dimensions de l'ordre de 28,8×33,1mm et poids qui ne dépasse pas les 10g, le module Bluno Beetle Bluetooth 4.0 BLE est l'un des plus petits modules Bluetooth au monde, idéal pour créer les appareils portables telles que la nôtre. Il présente aussi des fonctionnalités extrêmement similaires à une carte UNO standard,

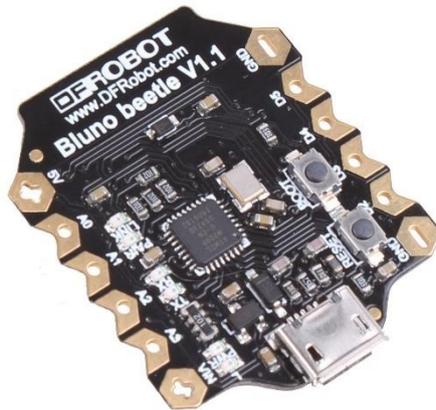


Figure 19 : Module Beetle

- **Modules Simblee** : Il s'agit d'une nouvelle technologie permettant de créer des IoT. Les utilisateurs peuvent créer rapidement et facilement des interfaces utilisateur graphiques (GUI), surtout avec des Smartphones et des tablettes, pour contrôler ces appareils. Deux principaux modules Simblee sont proposés le RFD77201 et RFD77203 désignés respectivement par 7-GPIO et 29-GPIO. GPIO pour "General Purpose Input/Output" (Entrée/Sortie à usage général) désigne le nombre d'entrée/sorties disponibles pour chacun. Le module Simblee contient deux fonctions principales : un processeur ARM Cortex-M0 32 bits (avec 128 Ko de Flash et 24 Ko de RAM - sans EEPROM - fonctionnant à 16 MHz) et une Radio Bluetooth "intelligente". Le module RFD77201 ou 7-GPIO bénéficie de dimensions très avantageuses de l'ordre de 7×10×2,2 mm. Un autre avantage de taille pour ces modules est la possibilité de créer les programmes à exécuter à l'aide de l'environnement de développement intégré Arduino (IDE).

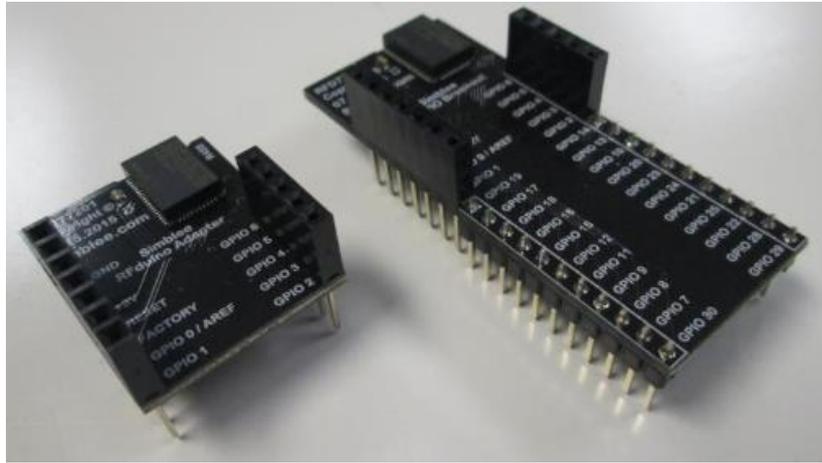


Figure 20 : Les deux modules Simple : 7-GPIO et 29-GPIO

Compte tenu de cette recherche relativement exhaustive que nous avons pu mener et en tenant compte des contraintes que nous nous sommes imposées, le module Simplee RFD77201 se prescrit comme un choix judicieux, justifié et incontournable dans ce cas. En effet, il joint à la fois la très faible consommation énergétique, la miniaturisation et la programmation sous l'environnement de développement Arduino.

### 6.2 Module NFC Reader (Lecteur NFC)

Ce module, qu'il faut lier avec l'IOT est indispensable pour pouvoir lire les données du capteurs FreeStyle d'Abbott. Il existe beaucoup de modules de ce type. Néanmoins, le choix d'un exemple particulier de lecteur NFC doit répondre à plusieurs contraintes notamment la faible dimension et aussi sa compatibilité avec les capteurs FreeStyle. Rappelons que ce capteur Abbott utilise la norme NFC ISO / IEC 15693. Nombreux sont les lecteurs NFC qui sont compatibles avec cette norme, reste alors les dimensions du lecteur qui vont limiter drastiquement notre choix. A cet effet, le module NFC BM-019 paraît le meilleur choix possible pour cette application compte tenu surtout de ses dimensions acceptables 38,10×27,9 mm.



Figure 21 : NFC BM-019

### 6.3 L'alimentation de la réalisation

Le module IoT RFD77201 et le lecteur NFC BM-019 ont besoin d'être alimentés avec 3,6V DC. Une pile rechargeable de 3,7 VDC doit pouvoir assurer cette tâche. Elle doit être à la fois petite et avec une puissance en mAh (milliampère-heure) relativement intéressantes pour assurer une large autonomie d'une journée par exemple. Ceci peut être facilement assuré grâce à la technologie Li-po sous forme d'un accumulateur lithium-ion polymère où l'électrolyte utilisé est un polymère gélifié. Le choix est vite obtenu avec une Pile ACCU Batterie Li-po 3.7V 100mAh. Quant à sa recharge, elle peut être effectuée grâce un micro chargeur Li-po possédant un connecteur Micro-USB.



Figure 22 : Pile Li-po 3.7V 100mA et chargeur Li-po

#### 6.4 Réalisation complète

Après avoir choisi tous les modules nécessaires à la réalisation, nous proposons ci-dessous la connexion électrique complète de tout le dispositif. Comme nous pouvons le voir, la partie électrique est très simple. En effet, cinq pins GPIO (Entrées/Sorties) seulement sont nécessaires pour connecter le module IoT au lecteur NFC, en plus des deux fils indispensables pour l'alimentation de l'ensemble à savoir un fil pour raccorder le +3VDC et un autre fil pour la liaison de la masse Gnd.

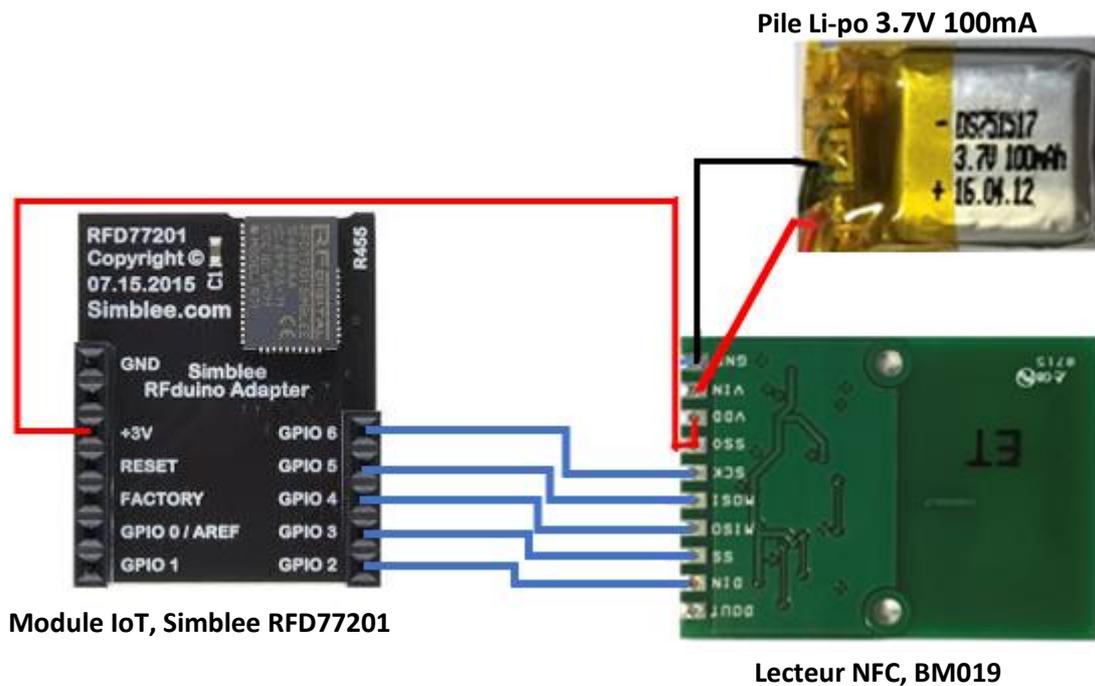


Figure 23 : Schéma électrique du système de surveillance de la glycémie

Lecteur NFC, BM09	Module IoT, Simblee RFD77201
Pin2 : DIN	GPIO Pin 2 : IRQ
Pin3 : SS	GPIO Pin 3 : SS
Pin4 : MISO	GPIO Pin 4 : MISO
Pin 5 : MOSI	GPIO Pin 5 : MOSI
Pin6 : SCK	GPIO Pin 6 : SCK
Pin 7+Pin8 : SS0 + VDD	Pin +3V
Pin 9 : VIN to + Li-po	
Pin 10 : Gnd	Pin Gnd

Tableau 3 : Brochage RFD77201/BM019

## 7. Conclusion

Ce deuxième chapitre très important nous a permis de mettre en œuvre notre réalisation de point de vue matériel. Le choix des différents composants et équipements nécessaires à notre réalisation ont été choisis et adoptés d'une manière judicieuse selon plusieurs critères objectifs. À vrai dire la réalisation a été effectuée en deux temps. Dans un premier temps nous avons utilisé les équipements à faibles coûts disponibles sans aucune contrainte. Après avoir validé le bon fonctionnement de la réalisation et en tenant compte des bons résultats des tests effectués, une optimisation matérielle s'est avérée indispensable. En effet, le système proposé est censé être porté en continu par le patient. Il est donc nécessaire qu'il ne crée aucune gêne physique au patient et aussi augmenter au maximum son autonomie énergétique. Ceci ne peut être assuré que grâce à l'utilisation d'un module IoT de très faible dimension et muni d'une connexion BLE. Une autre contrainte que nous avons choisie également de vérifier est que l'IoT à adopter doit pouvoir travailler sous l'environnement Arduino. Ceci doit nous permettre de garder le même programme mis en œuvre (voir dans le troisième chapitre) pour la première réalisation. Après avoir étudié et passé en revue un bon nombre de modules IoT disponibles et à faible coût, nous avons adopté les équipements les plus pertinents pour notre réalisation selon les contraintes évoquées précédemment.

# Chapitre 3 : Description de la partie software

## 1. Introduction

Dans ce troisième chapitre nous allons décrire la réalisation software du système proposé. Le système que nous proposons, comme nous l'avons décrit dans le chapitre précédent, s'articule sur quatre systèmes indépendants qui communiquent entre eux via trois types de liaisons à savoir NFC, BLE et Wi-Fi :

- Le capteurs FreeStyle d'Abbott qui utilise le protocole NFC pour l'envoi des mesures
- Un système IoT muni d'un lecteur NFC pour récolter les données fournies par le capteur et qui les envoie périodiquement à travers une liaison BLE vers un Smartphone
- Le Smartphone, utilisant une application appropriée, récupère ces données via la liaison Bluetooth, pour les afficher sur son écran sous forme de courbes, pouvant générer des alertes et la transmission de ces données vers un serveur web local
- Le serveur web local connecté à l'Internet a pour rôle le stockage de ces données sur des durées importantes et la possibilité de permettre une surveillance à distance de ces mesures.

Il est donc indispensable de mettre en œuvre trois parties logicielles permettant de faire fonctionner respectivement le module IoT en utilisant l'environnement Arduino, l'application au niveau de Smartphone à l'aide du kit de développement Android SDK et enfin le site web qui est sensé récupérer les données mesurées envoyées par le Smartphone pour les mettre en ligne sur le net.

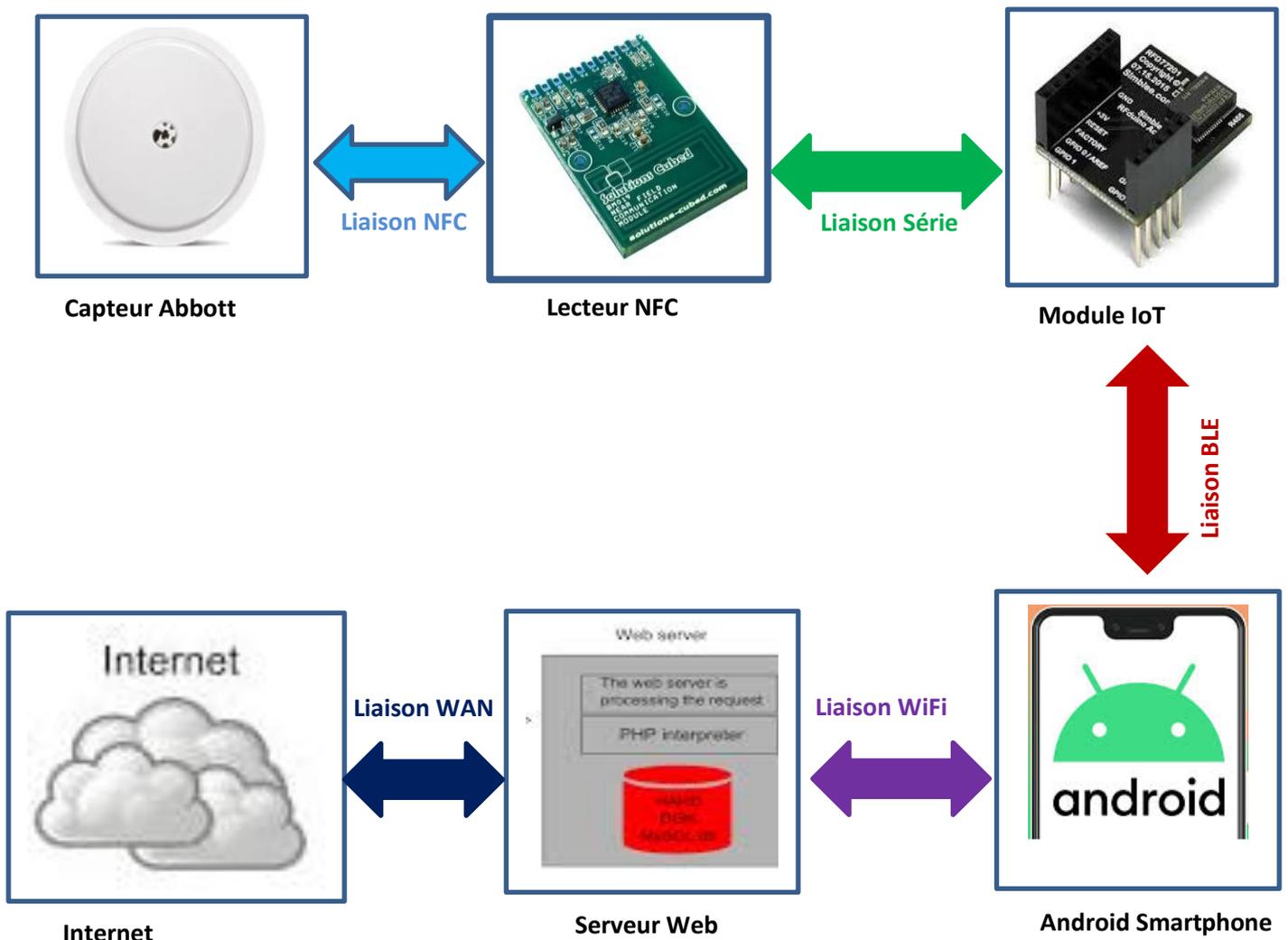


Figure 24 : Schéma synoptique du système de surveillance de la glycémie en continu et en ligne

## **2. Programme dans l'environnement de développement intégré Arduino**

### **2.1 Environnement de développement intégré Arduino (IDE)**

Les modules Arduino sont des cartes de prototypage programmables. Les programmes, appelés croquis (sketch en anglais), de ces cartes sont écrits dans un environnement de développement intégré Arduino connu sous le sigle IDE Arduino. Les programmes Arduino, écrits dans un langage basé sur le langage C, comportent trois parties essentielles à savoir : les déclarations des variables et des constantes globales, la fonction Setup () exécutée une seule fois au démarrage ou lorsqu'on appuie sur Reset et la fonction Loop () exécutée en boucle sans interruption tant que le module Arduino est alimenté. Après avoir terminé le programme nous avons la possibilité de le vérifier, pour ensuite le téléverser (le charger) dans le module Arduino qui doit être connecté au PC via une liaison USB (Com).

### **2.2 La structure générale du programme Arduino**

Le module IoT, en l'occurrence le RFD77201, encapsulera les données collectées par le module NFC et les transmettra à l'application Android via le BLE. Le patient choisira la fréquence de collecte. Le paquet de données encapsulé contiendra les informations collectées par le module NFC, son identifiant de série et la température de la carte. Par conséquent le programme que nous devons développer et implanter au niveau de l'IoT doit satisfaire ces exigences à savoir la lecture des données fournies par le capteur via le lecteur NFC, l'encapsulation de ces données et leurs transmissions via l'interface Bluetooth. Ceci ne peut être effectué qu'après que le module IoT soit connecté avec l'Android par Bluetooth.

## 2.3 Lecture et envoie des données du FreeStyle

Rappelons tout d'abord que le capteur FreeStyle d'Abbott mesure non pas le glucose dans le sang (obtenu par exemple à l'aide d'une piqure utilisée avec les glucomètres classiques) mais plutôt le glucose dans le liquide interstitiel, autrement dit le liquide entre les cellules du corps ISF (interStitial Fluid). En effet, Le glucose dans l'ISF s'est avéré être un indicateur fiable des niveaux de glucose dans le sang, car le glucose diffuse librement des capillaires vers l'espace interstitiel. Lorsque les taux de glucose dans le sang sont stables, les mesures de glucose sanguin (la mesure réelle dans le sang) et du capteur (à travers le liquide interstitiel) sont similaires. Si les niveaux de glucose augmentent ou diminuent, les deux lectures peuvent être un peu différentes car la diffusion du glucose entre le capillaire et l'ISF a un court décalage de 5 à 10 minutes.

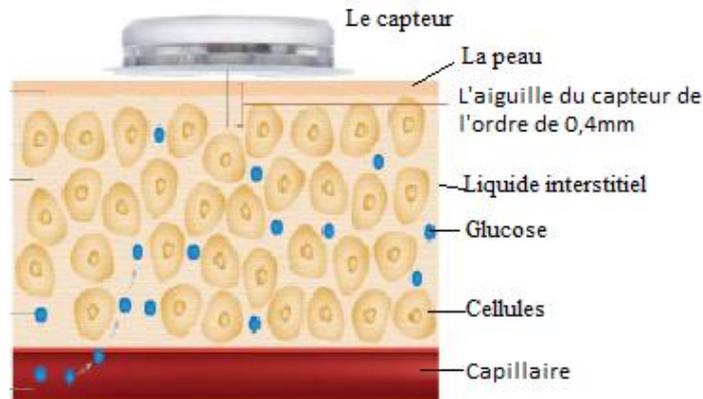


Figure 25 : Principe de fonctionnement du capteur Abbott

Comme nous l'avons indiquée plus haut le capteur Abbott dispose, en plus de la puce RFID 13,56 MHz le RF430 de Texas Instrument, d'un microcontrôleur 16 bits MSP430 également de Texas Instrument et d'une mémoire non volatile FRAM universelle. Cependant, il est très probable que ce capteur d'Abbott utilise une version personnalisée de ce contrôleur. Autrement dit, Texas Instrument aurait fabriqué ce microcontrôleur avec des spécificités particulières demandées par Abbott. Ce qui rend difficile la compréhension de la structure des données récupérées de ce capteur. Cependant, les données peuvent être lues grâce notamment à l'utilisation d'un lecteur NFC (NFC reader) de type NFC BM019. Il dispose d'un module de conversion série ayant une puce d'interface NFC ST Micro CR95HF.

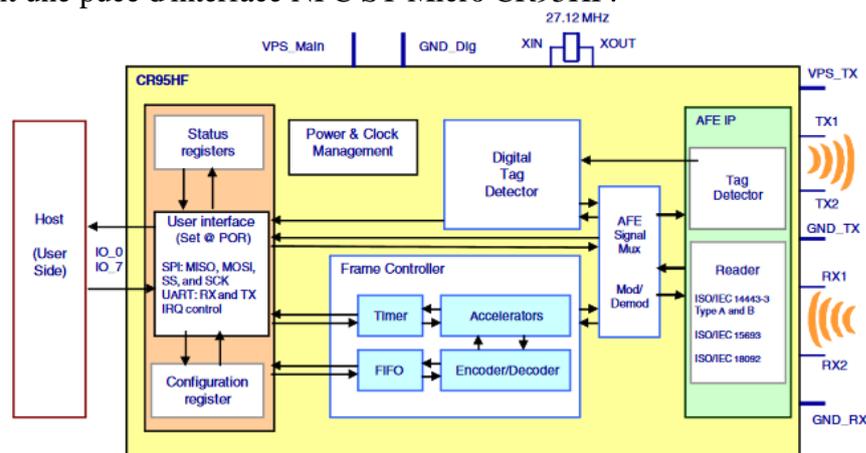


Figure 26 : Schéma synoptique du CR95HF du lecteur NFC

Dès lors nos recherches sur le sujet nous ont permis de trouver une application open source développée et maintenue par Uwe Petersen (<https://github.com/UPetersen/LibreMonitor>) appelée Libre Monitor permettant de lire toutes les deux minutes le capteur et de transférer les 32 valeurs d'historique, en l'occurrence les mesures de taux de glucose, mesures de taux de glucose des huit dernières heures, les 16 valeurs de tendance temporelles actuelles et les valeurs des 15 minutes précédentes. Nous avons donc utilisé cette application pour ne permettre de lire les mesures fournies par le capteur FreeStyle.

## 2.4 Organigramme général du programme de IoT

Une connexion doit être assurée avec le partenaire désirée en l'occurrence le Smartphone. Puis, chaque deux minutes l'IoT va devoir lire les données NFC du capteur chaque deux minutes. Enfin, un transfert par Bluetooth sera effectué à chaque lecture NFC. Ce transfert comportera outre la mesure actuelle du glucose, 32 autres valeurs historiques. Dans le schéma synoptique, ci-dessous, nous représentons d'une manière très simplifié le fonctionnement du système IoT.

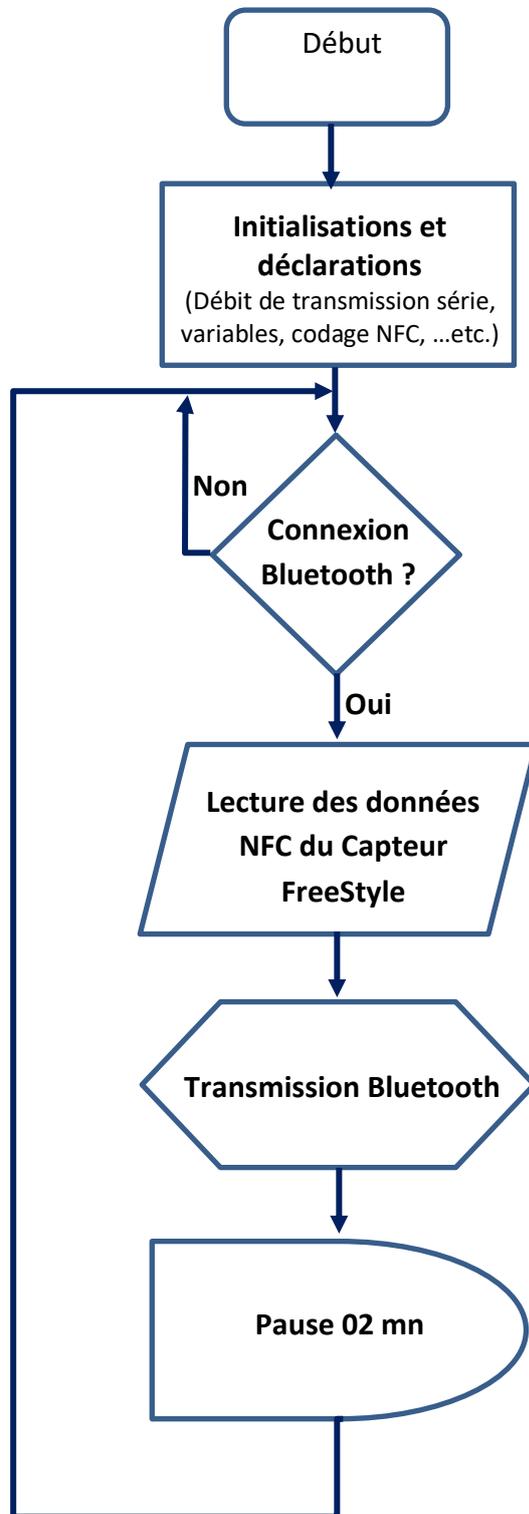


Figure 27 : Organigramme simplifié du fonctionnement du système IoT

### 3. Création de l'application Android sous l'environnement SDK

Dans cette partie, nous avons envisagé de profiter de la puissance du Smartphone afin de réaliser une simple application « développée sous Android studio » dédiée aux à la surveillance en continu et en ligne du taux de glycémie des personnes diabétiques type 1. Ceci doit leur permettre de mieux gérer et contrôler leur maladie. Cette application a pour rôle essentiel de réceptionner les mesures de la glycémie en continu, d'enregistrer certaines de ces données qui sont nécessaires, non seulement pour les diabétiques, mais aussi pour le médecin traitant.

#### 3.1 Les principaux systèmes d'exploitation d'un Smartphone

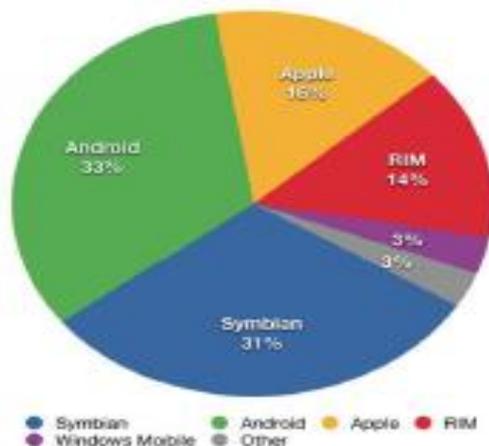


Figure 28 : Les principaux systèmes d'exploitation

##### 3.1.1 L'Android

Android est un système d'exploitation ouvert (Open Source) pour tablettes tactiles, terminaux mobiles, Smartphones et TV connectées. Il a été conçu en 2007, par la société Android, une start-up rachetée par Google. C'est un système d'exploitation fondé sur un noyau Linux disponible grâce à une licence Apache. Ce système inclut tous les utilitaires requis par un constructeur pour le mettre en œuvre dans un téléphone portable. Il est proposé à tous les fabricants de téléphones mobiles, pour faciliter son adoption. Cependant, Android est modifiable par les constructeurs. Donc, les versions varient largement d'un constructeur à l'autre.

##### 3.1.2 L'architecture de la plateforme d'Android :

L'image ci-dessous décrit l'architecture complète d'Android :

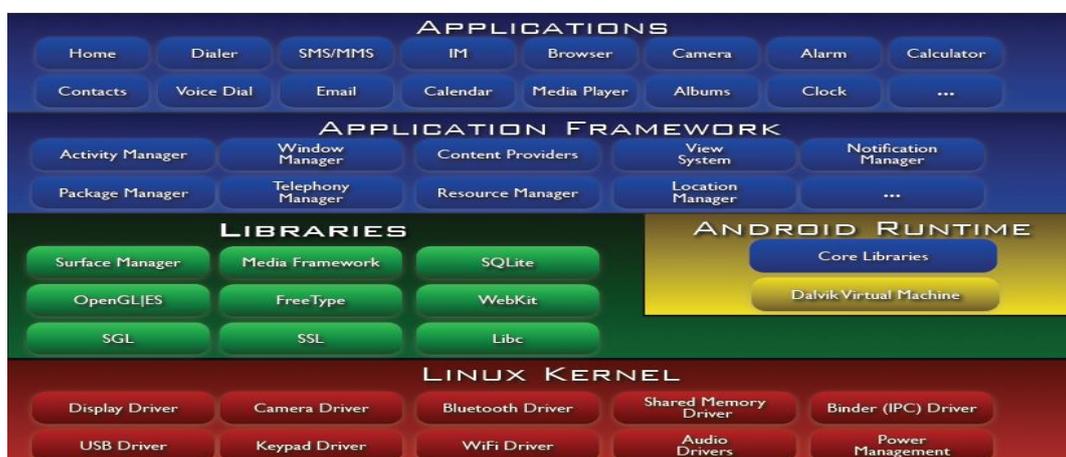


Figure 29 : L'architecture de la plateforme d'Android.

## ▪ Premier niveau : Le noyau Linux

Android s'appuie sur un noyau Linux 2.6 qui agit également comme une couche d'abstraction entre le matériel et le reste de la pile logicielle sur laquelle vient s'agréger différents services tels que la sécurité, le gestionnaire de mémoire, le gestionnaire des processus et la pile réseau.

## ▪ Deuxième niveau : Les librairies

Les librairies natives sont écrites en langage C et C++.

- Le Surface Manager est chargé de la composition des items sur l'écran, de la gestion du dispositif d'affichage. Il permet de s'assurer que les pixels s'affichent bien à l'écran.
- Open GL/ES quant à lui gère le graphisme en 3D tandis que SGL gère l'affichage en 2D. Ainsi une même application peut combiner du 2D avec du 3D.
- Le Media Framework contient des codecs audio et média (MPEG 4, H.264, AAC, MP3...).
- Le Free type est une bibliothèque logicielle open source qui implémente un moteur de rendu de police de caractère.
- Le SQLite est une bibliothèque open source écrite en C permettant d'implémenter un moteur de base de données relationnelle.

Niveau adjacent : l'environnement d'exécution

- Le Dalvik Virtual Machine exécute des fichiers de type «. Dex » qui sont en fait le résultat en bytecodes de la conversion de fichier «.class » et «.jar ». Il permet un usage de la mémoire, un partage entre processus plus efficaces. C'est un interpréteur de bytecode optimisé. Il est possible d'avoir plusieurs instances de DVM s'exécutant au même moment.
- Les « Core Librairies » écrit en java est un ensemble de collection, de classes, d'utilitaires d'entrée/ sortie.

## ▪ Troisième niveau : le module de développement d'application

Un Framework fournit un ensemble de fonctions facilitant la création de tout ou d'une partie d'un système logiciel, ainsi qu'un guide architectural en partitionnant le domaine visé en module.

L'Application Framework développée en java contient un certain nombre d'applications dédiées (application téléphonique, des applications écrites par Google ou par un tiers). Toutes les applications peuvent utiliser le même API.

« Activity Manager » permet de gérer le cycle de vie d'une application (application en tâche de fond par exemple).

□ Le « paquet manager » garde une trace des applications installées dans l'équipement. Si l'on télécharge une nouvelle application par exemple, le paquet manager informe sur la capacité du système.

Le « Windows manager » s'occupe de gérer la fenêtre d'affichage.

□ Le « Téléphonie manager » contient des API pour la construction d'une application téléphonique.

□ Le « Content provider » permet le partage de données, l'interaction avec d'autres applications (répertoire, numéro de téléphone, dont on a besoin les autres applications).

□ Le « Ressource Manager » quant à lui stocke les bitmaps locaux.

## ▪ Quatrième niveau : les applications

Niveau d'abstraction dans lequel on peut trouver toutes les applications spécifiques au fonctionnement d'un Smartphone (téléphone, répertoire, navigateur web...)

### 3.2 Développement sous Android

#### 3.2.1 L'environnement de développement sous Android

Afin de développer des applications sous Android, un ensemble d'outils est nécessaire. Vu que les procédures d'installation de ces outils sont assez longues et fastidieuses, alors les décrire, pas à pas, risquerai

de prendre énormément de place et ainsi beaucoup de pages (Nous laisserons le lien de téléchargement et l'installation dans les références). Alors, on se contentera juste d'évoquer les outils et leur intérêt.

### 3.2.2 Le JDK (Java Développement Kit)

Les applications développées pour Android étant essentiellement écrites en langage java ; un langage de programmation orienté objet qui a la particularité d'être très portable. Cela signifie qu'un programme Java, fonctionnant sur Windows (par exemple), pourra facilement tourner sur Mac ou GNU/Linux.

Cette petite prouesse vient du fait que Java s'appuie sur une machine virtuelle pour s'exécuter (appelée la JVM). Pour avoir une JVM sur votre ordinateur, il vous faut télécharger le JRE. Ce dernier contient, en plus de la JVM, des bibliothèques Java standards.

La JVM ne lit pas directement le code Java. Elle lit un code compilé (le bytecode). Pour passer du code Java, que le développeur écrit, au code compilé, lu par la JVM, des outils spéciaux sont nécessaires. Ces outils sont inclus dans le JDK. De plus, le JDK contient le JRE (et donc la machine virtuelle), ce qui est bien pratique. Pour résumer, on dira que :

- ✓ Pour un simple utilisateur de Java : il doit avoir le JRE ;
- ✓ Pour un développeur : il aura besoin des outils du JDK

### 3.2.3. Le SDK (Software Development Kit) Android

Un SDK, c'est-à-dire un kit de développement logiciel, est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Le SDK Android permet, donc, de développer des applications pour Android et uniquement pour Android.

Au premier lancement du SDK, un écran semblable à la Fig. S'affichera :

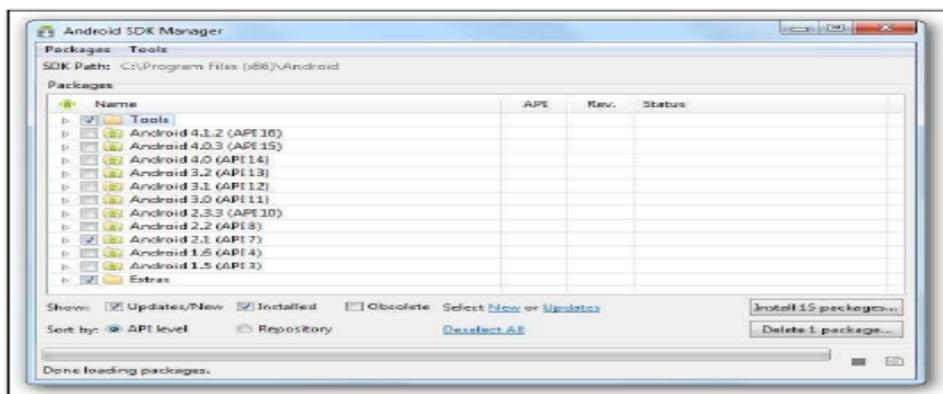


Figure 30 : Le SDK Android.

### 3.2.4. L'IDE Eclipse

Eclipse est un environnement de développement intégré. C'est un logiciel qui permet d'écrire un programme beaucoup plus facilement qu'avec le simple Bloc-notes. Outre la coloration du code, il permet d'apporter des outils très pratiques pour compiler vos programmes, les déboguer, etc. Il peut être utilisé pour programmer avec n'importe quel type de langage, mais nous l'utiliserons pour faire du Java.

De plus, Eclipse est conçu pour pouvoir être complété avec des plugins (extension). Ainsi, il existe un plugin pour développer des applications Android que nous verrons dans la partie suivante.

### 3.2.5. Le plugin ADT pour Eclipse

Google fournit un plugin pour Eclipse, nommé ADT (Android Development Tools). La fonction principale de ce plugin est de créer un pont entre Eclipse et le SDK Android.

### 3.2.6. L'émulateur de téléphone : Android Virtual Device

L'Android Virtual Device, aussi appelé AVD, est un émulateur de terminal sous Android, c'est-à-dire que c'est un logiciel qui fait croire à l'ordinateur qu'il est un appareil sous Android. C'est la raison pour laquelle nous n'avons pas besoin d'un périphérique sous Android pour développer et tester la plupart de nos applications.

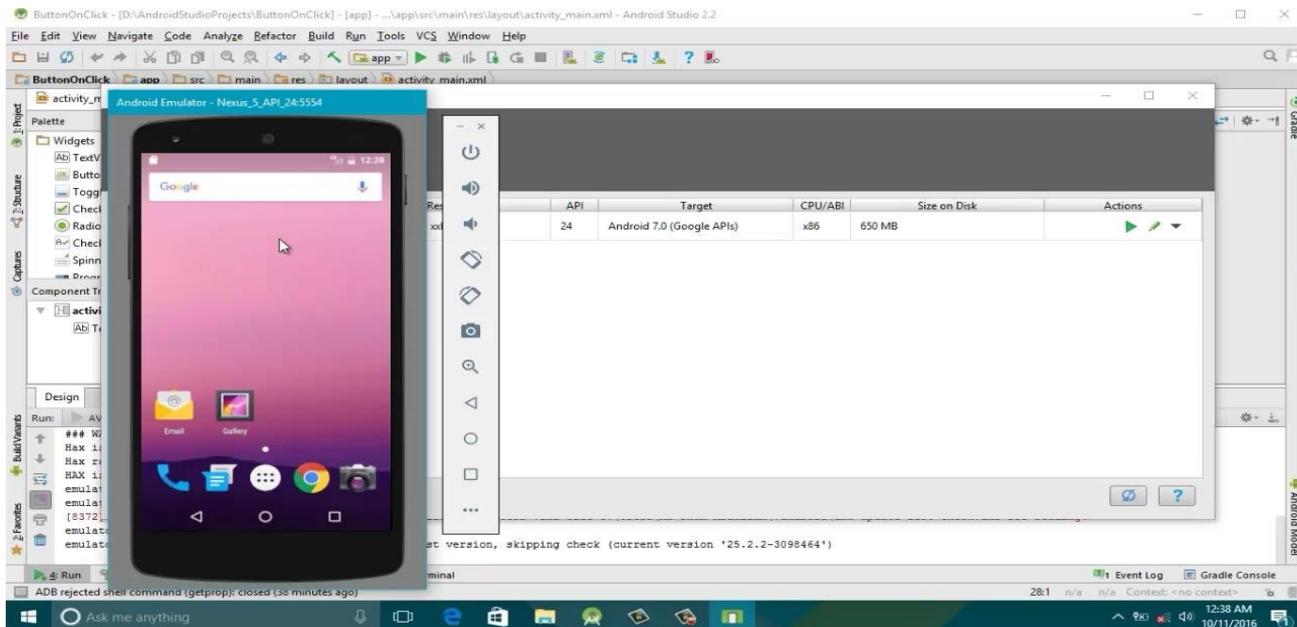


Figure 31 : L'émulateur Android

#### 4. Le développement d'applications sous Android studio

Comme nous avons déjà entamé le langage de programmation « java », utilisé pour créer des applications Android, Android Studio est l'un des logiciels qui utilise lui aussi ce langage et qu'on utilise pour le développement de notre application.

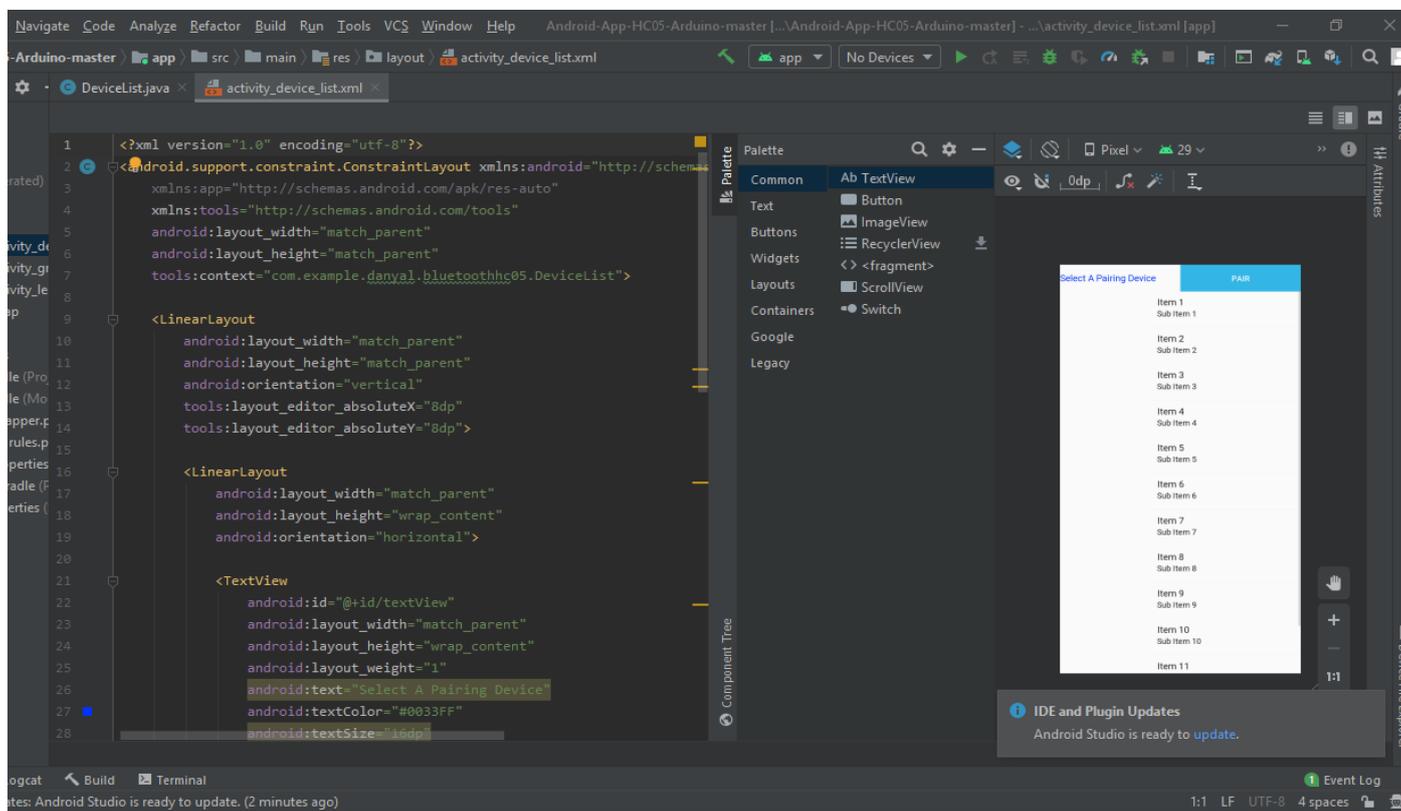


Figure 32 : L'aire de développement Android Studio

## 4.1. Architecture d'un projet Android studio

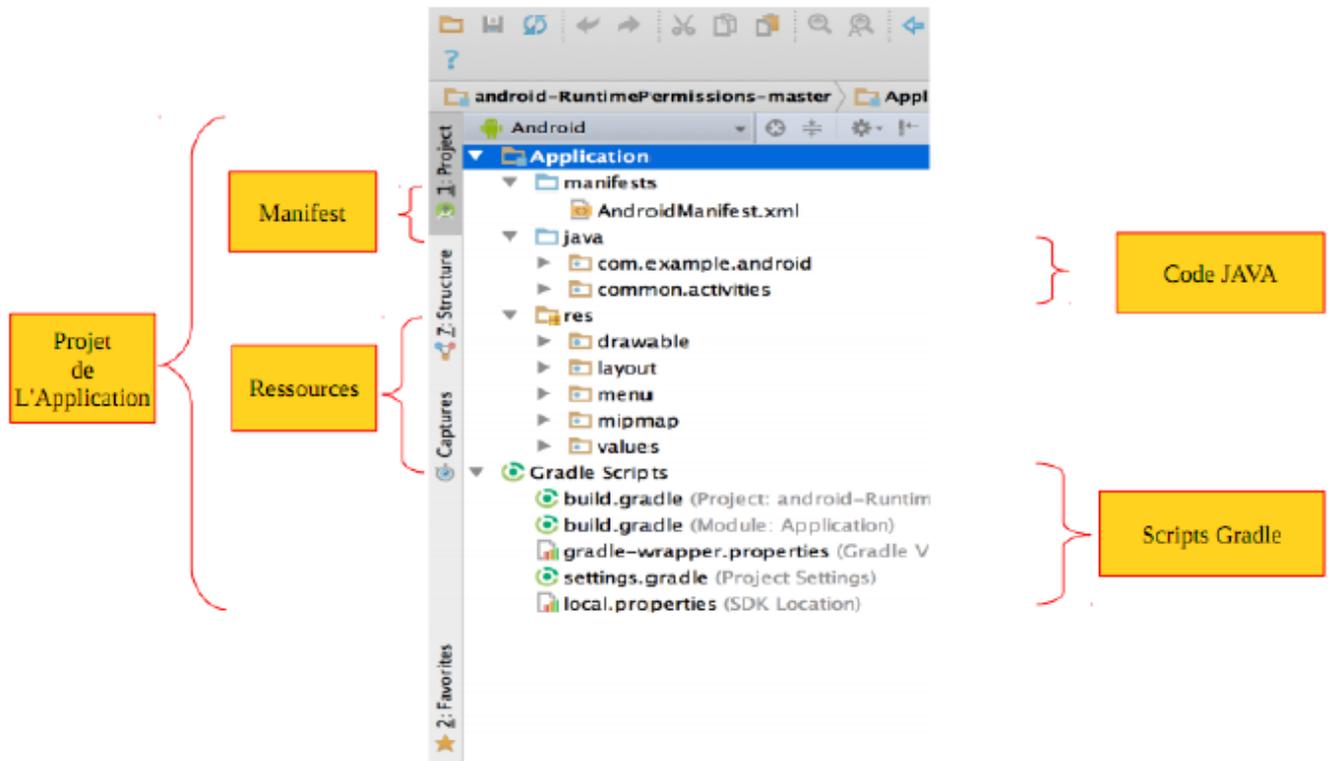


Figure 33 : Architecture de l'interface d'Android studio

**Java** : Contient le code source écrit en java classé par paquet. Dans notre projet, le paquet une activity se nommant « MainActivity ».

**Res** : Contient toutes les ressources de notre application c'est à dire les images, les couleurs, les textes dans toutes les langues, les thèmes... et surtout les interfaces graphiques.

Dans **res** il y a les répertoires :

**Drawable** : Ils sont nombreux car il y en a un par type d'écran. On y met par exemple les images de votre application adaptée à chaque type d'appareil.

**Layout** : C'est le répertoire par défaut pour les interfaces graphiques.

**Menu** : Il permet de définir les menus qui sont proposés par les différentes activity.

**Values** : Ce répertoire contient par défaut les fichiers dimens.xml pour centraliser des informations de dimensionnement comme les marges et autres espacements entre les éléments de l'interface graphique, strings.xml qui contient les textes de votre application (texte des menus, texte des boutons...), et styles.xml qui permet de définir le style de votre application (Android propose des styles pour débiter facilement).

## 4.2 Notre Application Android « Diabtapp »

Cette application développée sous Android studio consiste à réceptionner la donnée émise de la carte Arduino via le module Bluetooth. Elle comporte, entre autres :

- ✓ Un bouton pour la connexion Bluetooth,
- ✓ Un champ d'affichage pour la liste des appareils Bluetooth connecté,
- ✓ Un champ d'affichage de la valeur lue instantanément.

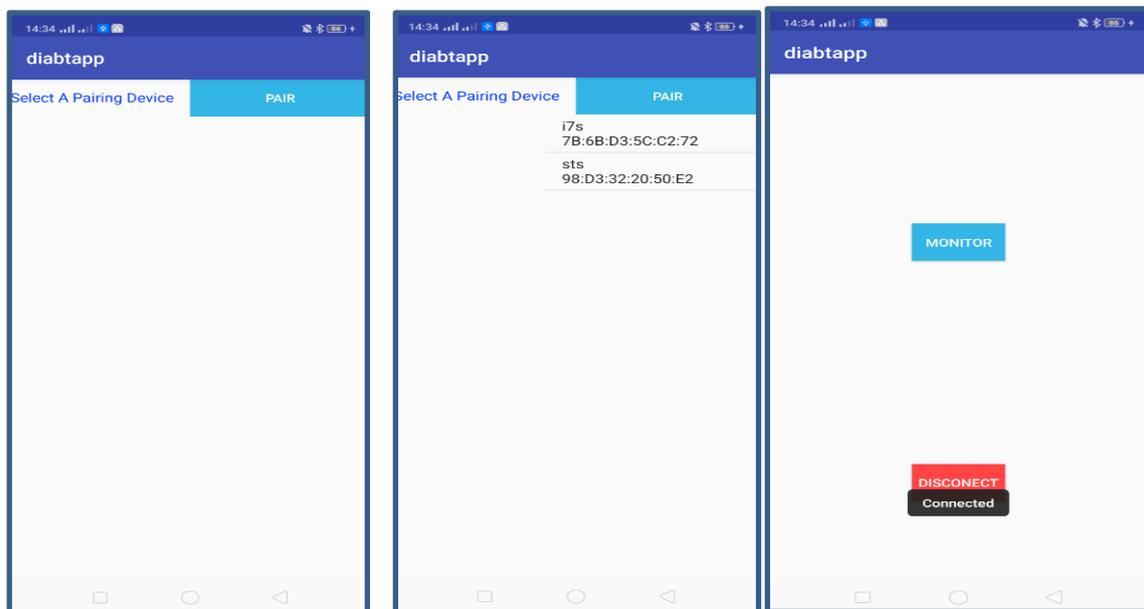


Figure 34 : Vue d'ensemble de l'application Android réalisée

Une fois que le Bluetooth est associé au Smartphone « connexion établie » et en cliquant sur le bouton « Monitor », on aura l'affichage des valeurs du taux de glucose en temps réel sous forme d'une courbe, acquise depuis la carte Arduino.

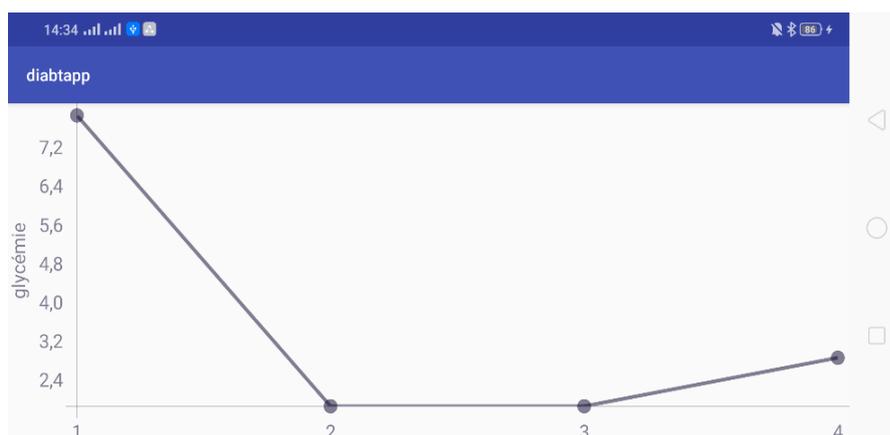


Figure 35 : Fenêtre 'Monitor' de l'application Android réalisée

Une alerte est déclenchée si la valeur de taux de glucose est dépassée. Il s'agit alors d'une hypoglycémie si le taux  $< 0,72$  g/L (l'équivalent de 4 mmol/L) ou d'une hyperglycémie lorsque le taux dépasse 1,44 g/L (l'équivalent de 8 mmol/L). Nous rappelons que la conversion des g/l en mmol/l, dans le cas de la glycémie, est obtenue en multipliant les g/l par 5,5.

$$\text{g/l} \times 5,55 = \text{mmol/l}$$

(2)

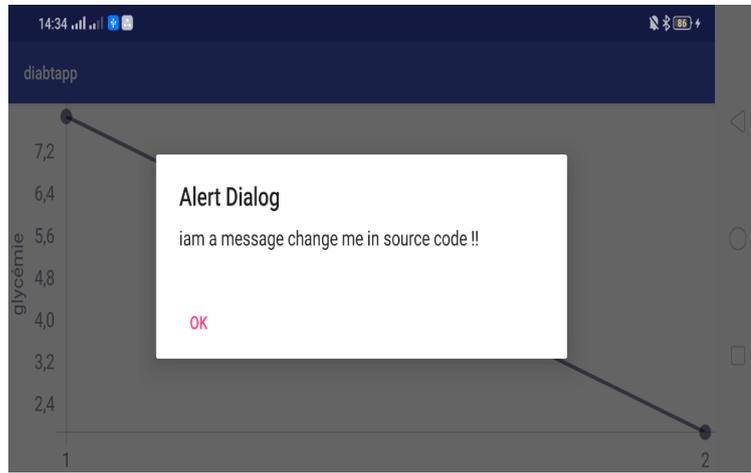


Figure 36 : Fenêtre "Alert" de l'application Android réalisée

L'organigramme simplifié de l'application sous Android peut être décrite par le schéma suivant :

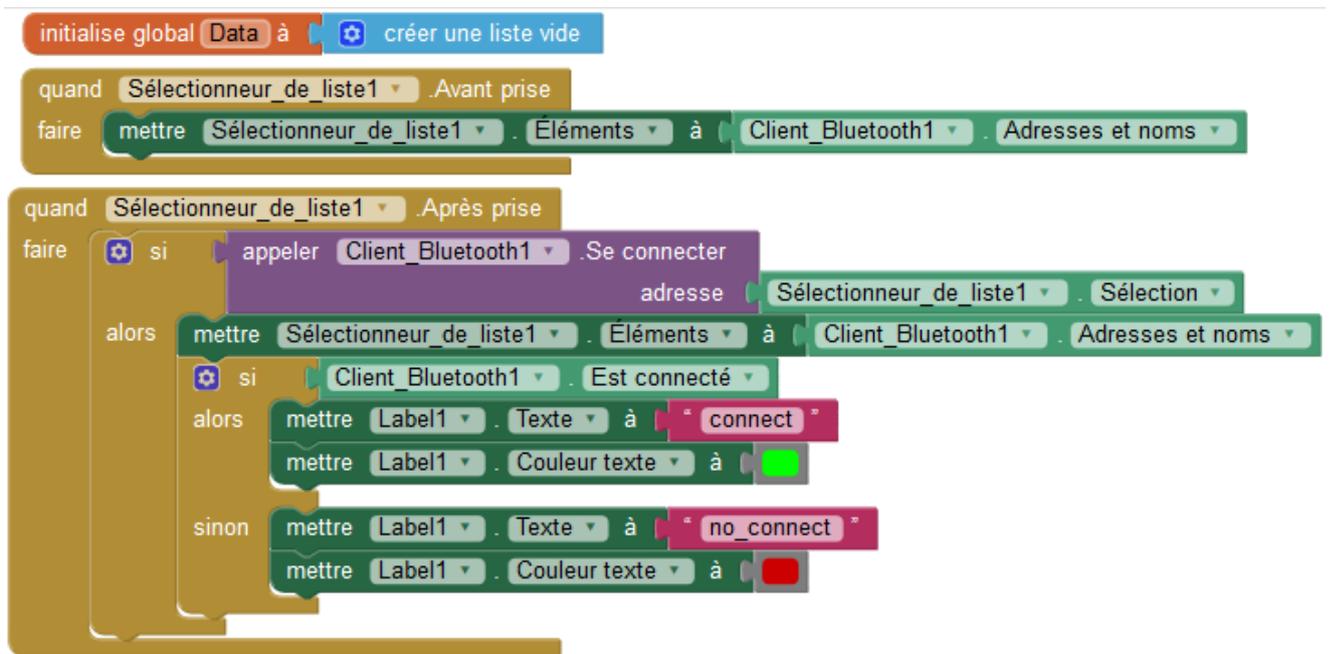


Figure 37 : Algorithme simplifié de la Connexion Bluetooth dans l'application Android

Une fois la connexion Bluetooth est validée, le Smartphone récupère les données envoyées par l'IoT selon l'organigramme simplifié ci-dessous :

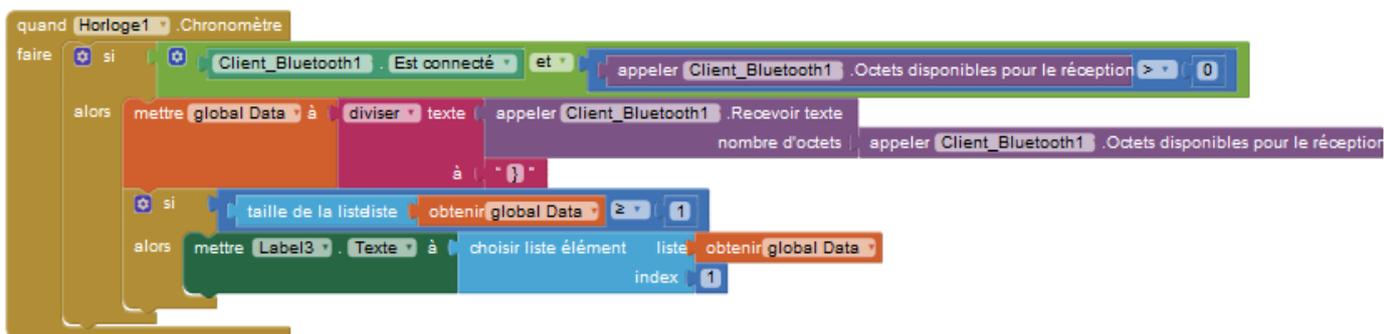


Figure 38 : Algorithme simplifié de l'acquisition et l'affichage dans l'application Android

## 5. Le serveur web local

Comme nous l'avons mentionné plus haut, la finalité de ce projet est de pouvoir récupérer les données mesurées, périodiquement au niveau d'un serveur web qui pourra ensuite les stocker et les mettre en ligne à travers internet. Dans cette dernière partie de notre projet nous souhaitons donc mettre en œuvre ce serveur. A vrai dire il s'agit de deux étapes à savoir la conception de notre site web et la configuration du serveur qui doit héberger ce site et le faire fonctionner sur un réseau local et sur internet.

### 5.1 Site web

Avant de créer notre site il faut d'abord choisir de quel à savoir un site web statique ou bien dynamique et quelle est alors la différence les deux ?

#### 5.1.1 Site web statique

Un site web statique est un site internet composé de pages web statiques dont le contenu est fixe. - Lorsque, le client demande au serveur à voir une page web, le serveur lui répond en lui envoyant la page demandée.

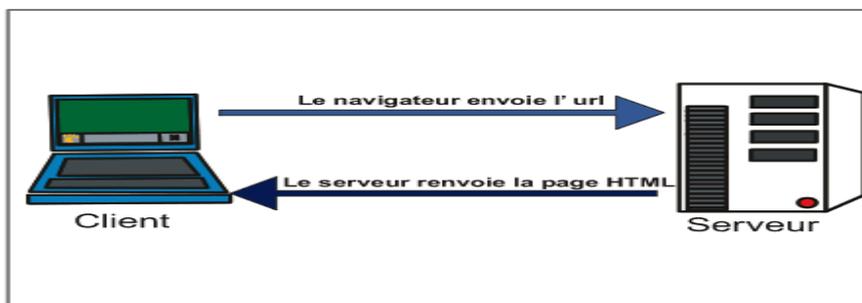


Figure 39 : Fonctionnement d'un site web entièrement statique

#### 5.1.2 Site web dynamique

Un site web dynamique est un site web relié à une base de données, dont les pages se construisent à la demande de l'internaute. Celui-ci envoie des requêtes que le serveur WEB traite afin de retransmettre les résultats au navigateur de l'internaute

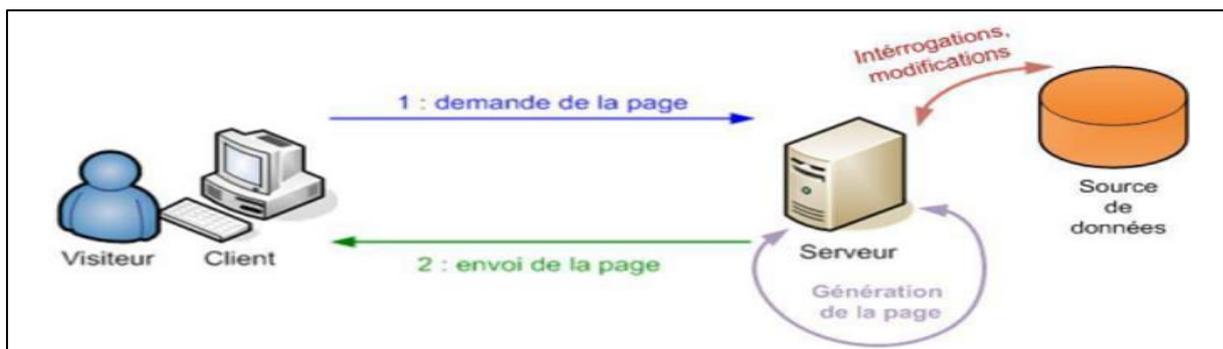


Figure 40 : Fonctionnement d'un site web entièrement dynamique

## 5.2 Les étapes de création d'un site web locale dynamique

### 5.2.1 créer un serveur et une base de données

Dans notre projet nous allons utiliser le logiciel XAMPP qui nous permet de créer facilement un serveur web local et une base de données.

#### a- Définition de XAMPP :

XAMPP est un ensemble de logiciels libres. Le nom est un acronyme venant des initiales de tous les composants de cette suite. Ce dernier réunit donc le serveur Web Apache, la base de données relationnelle et système d'exploitation MySQL ou MariaDB ainsi que les langages scripts Perl et PHP. L'initiale X représente tous les systèmes d'exploitation possibles, à savoir Linux, Windows et Mac OS X.

#### b- Qu'est ce qui est inclus :

- ✓ **Apache** : le serveur Web open source Apache est utilisé mondialement et permet de délivrer des contenus Web. L'application de serveur est mise à disposition en open source par l'*Apache Software Foundation*.
- ✓ **MySQL/MariaDB** : avec MySQL, XAMPP se compose de l'un des systèmes de gestion de base de données relationnelle les plus populaires au monde. En combinaison avec le serveur Web Apache et le langage script PHP, MySQL sert à l'enregistrement de données pour des services Web. Les versions actuelles de XAMPP favorisaient MariaDB à l'insu de MySQL comme gestionnaire de base de données, marquant un détachement avec ce dernier.
- ✓ **PHP** : Il s'agit d'un langage script côté serveur permettant de créer des pages Web ou applications dynamiques. PHP peut être mis en place sur toutes les plateformes possibles et est compatible avec divers systèmes de base de données.
- ✓ **Perl** : le langage script Perl est utilisé pour l'administration système, le développement Web et la programmation en réseau. De plus, des applications Web dynamiques peuvent être programmées de la même manière que PHP.

En dehors des composants principaux, la distribution gratuite d'Apache comprend divers outils selon chaque système d'exploitation comme le serveur Mail Mercury, l'application d'administration de base de données **PhpMyAdmin**, le logiciel d'analyse de données **Webalizer**, **OpenSSL**, **ApacheTomcat** ainsi que **FileZilla** ou **ProFTPd**.

#### c- Installation de XAMPP :

- ✓ Téléchargement <https://www.apachefriends.org/fr/download.html>.
- ✓ Lancer un fichier .exe,
- ✓ Une fois le logiciel téléchargé, lancez l'installation avec un double-clic sur le fichier avec la terminaison .exe,
- ✓ Désactiver les programmes antivirus temporairement le temps d'installer XAMPP,
- ✓ Désactiver l'UAC. Le contrôle du compte utilisateur (UAC pour User Account Control) peut aussi gêner l'installation de XAMPP car les droits d'écriture du lecteur C : sont limités. Le site de l'[assistance Windows](#) vous indique comment procéder.
- ✓ *Lancer l'assistant ou Setup-Wizard*. Une fois le fichier .exe exécuté et les programmes de sécurité gênant l'installation écartés, l'écran de démarrage du Setup-Wizard XAMPP apparaît. Cliquez sur suivant pour configurer les paramètres d'installation.
- ✓ Choix des composants logiciels. Ceci va nous permettre de sélectionner certains composants de l'ensemble XAMPP lors de l'installation

#### d- XAMPP-Control-Panel :

XAMPP Control Panel nous permet de contrôler les composants de votre serveur de test. L'**interface claire** recueille toutes les actions et vous permet de démarrer ou d'arrêter certains modules en un clic. Cette interface propose des actions diverses :

- **Config** : permet la configuration de votre XAMPP ainsi que des composants partiels
- **Netstat**: affiche les processus courants sur l'ordinateur local
- **Shell** : ouvre une cellule UNIX
- **Explorer** : ouvre le dossier XAMPP dans Windows Explorer
- **Services** : affiche les services fonctionnant en arrière-plan
- **Aide** : propose des liens vers des forums pour utilisateurs
- **Terminer** : ferme le XAMPP-Control-Panel

#### 5.2.2 Un langage pour crée un site dynamique :

##### a- Installation de python :

- ✓ Téléchargement <https://www.python.org/downloads/>,
- ✓ Après avoir lancé l'installateur on coche add path,
- ✓ Terminer,
- ✓ Installation de Visual studio,
- ✓ Ajout de l'extension python à Visual studio

##### b- Création d'un projet Django :

**Définition de Django** : est un cadre de développement web *open source* en Python. Il a pour but de rendre le développement web simple et rapide

##### Étapes de création du projet :

- ✓ Création un environnement virtuel,
- ✓ Installer Django,
- ✓ Lancer le projet

c- **Présentation de l'environnement** : Dans notre projet nous avons 4 fichiers nécessaires à programmer :

**Setting.py** : est un fichier core dans les projets **Django**. Il contient toutes les valeurs de configuration dont votre application Web a besoin pour fonctionner ; les paramètres de la base de données, la configuration de la journalisation, où trouver les fichiers statiques, les clés d'API si vous travaillez avec des API externes et bien d'autres

**Models.py** : est une classe qui représente une table ou une collection dans notre base de données, et où chaque attribut de la classe est un champ de la table ou de la collection.

**View.py** : sont un composant clé des applications créées avec le Framework. À leur plus simple expression, il s'agit d'une fonction ou d'une classe Python qui prend une requête Web et renvoie une réponse Web. Vues sont utilisés pour faire des choses comme aller chercher des objets de la base de données, modifier ces objets si nécessaires, rendre les formes, le retour HTML, et bien plus encore.

**url.py** : Le fichier urls.py contient les routes du projet : il dispatché les requêtes vers

Les contrôleurs adéquats

Maintenant après cette configuration Il est temps d'afficher des données, Pour nous aider Django fournit des balises de Template

Django Template : est simplement un document texte ou une chaîne Python balisée à l'aide du langage de Template Django

## 6.La communication entre notre application et le serveur web

### 6.1 Principe de fonctionnement

D'abord pour mieux comprendre la procédure de fonctionnement entre l'Application et le serveur web, nous proposons le schéma de principe suivant :

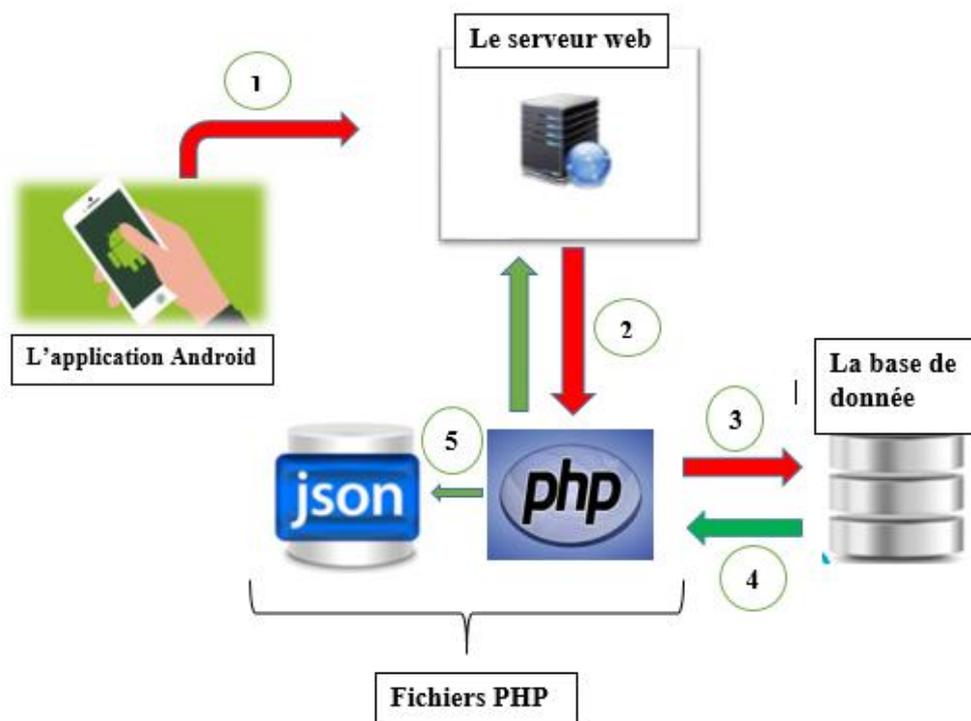


Figure 41 : la communication entre l'application et le serveur web

#### 1 Etablissement de la connexion avec le serveur en utilisant les commandes http client :

Notre application va communiquer avec le serveur en envoyant une requête http, il y a plusieurs méthodes mais on s'intéresse uniquement à 2 commandes : « get » et « post »

La méthode « get » : on l'utilise pour la récupération et la demande des ressources.

La méthode « post » : on l'utilise pour l'insertion et la modification des ressources.

#### 2 Le serveur faire un appel à les fichier PHP :

Seul le serveur qui va comprendre les fichiers PHP qui vont générer le code html et ccs pour les envoyés au client.

#### 3 La connexion à la base de données :

Les fichiers PHP vont envoyer des requêtes SQL a la base de données pour l'établissement de la connexion et va chercher la ressource demandée ensuite il récupère le résultat de ces requêtes.

#### 3 Le renvoi des données en JSON :

Après la récupération des données nous allons les encoder en JSON (*JavaScript Object Notation*) qui est un format de données textuelles, générique, dérivé de la notation des objets du langage JavaScript. Il présente aussi le retour des scripts PHP lorsqu'on les appelle par notre application Android.

### 6.2.1 Notre site web locale

Les deux figures ci-dessous présentent deux fenêtres de l'interface de site web avant la réception des données par l'application Android :

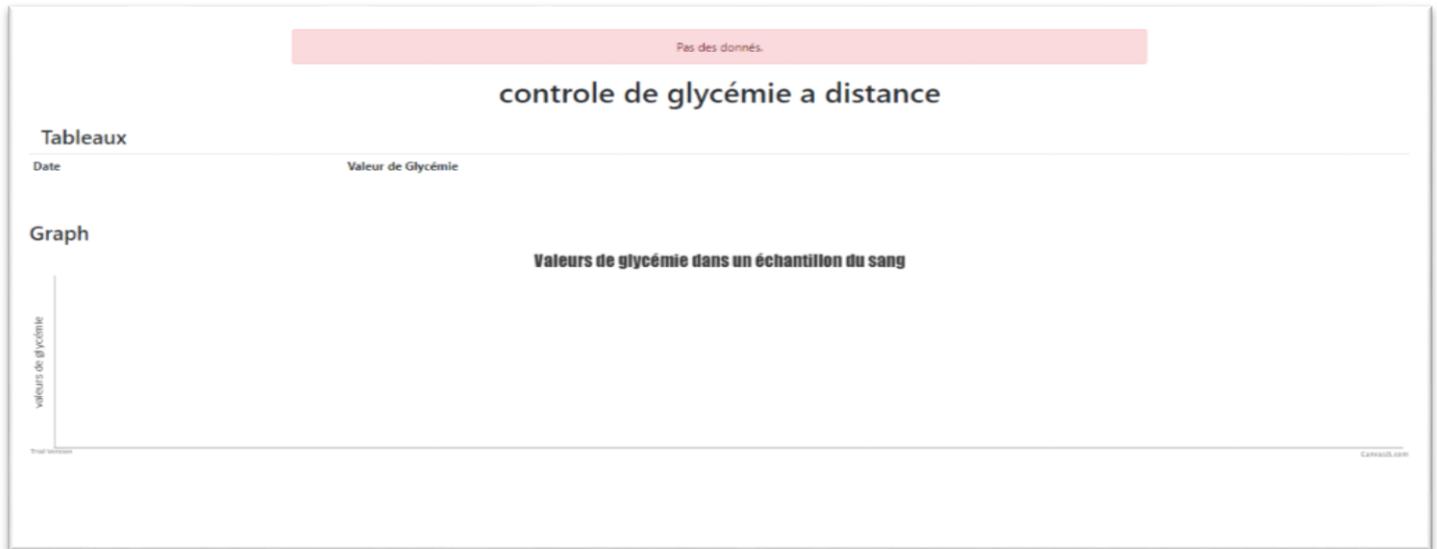


Figure 42 : L'interface de site web

L'image ci-dessous présente l'interface de site web après la réception des données par l'application Android :



Figure 43 : La réception des valeurs de la glycémie par le site web

## 7. Conclusion

Dans ce chapitre nous avons mis l'accent sur la partie logicielle de notre réalisation. Cette partie représente le cœur de notre travail. Elle est divisée en trois parties aussi importantes les unes que les autres :

- ✓ La première partie est dédiée au programme qui sera implémenté dans le système IoT. Ce programme doit gérer à la fois la lecture des données délivrées par le capteur FreeStyle sous forme d'un protocole NFC. Ce même programme assurera la transmission de ces mêmes données via la connexion Bluetooth Low Energy.
- ✓ La seconde partie de la réalisation software est la mise en œuvre d'une application Android. Elle doit garantir la réception des mesures envoyées par le système IoT, l'affichage en continu de ces mesures, l'envoi d'alerte dans le cas d'une hypoglycémie ou hyperglycémie et enfin la communication avec le serveur web via une connexion Wi-Fi
- ✓ La troisième partie est la réalisation d'un serveur web qui doit réceptionner les mesures envoyées par l'Android et leurs mises en lignes sur internet.

## Conclusion Générale et perspectives

Le Diabète est de nos jours une véritable pandémie qui ne cesse de s'accroître de jour en jour. En effet, il y'a presque un demi-milliard de diabétiques dans le monde avec une mortalité de presque deux millions de personnes par an qui est imputée à cette maladie alors que des traitements existent. Dans quelques décennies, à l'horizon de 2050, le nombre de diabétiques dans le monde doit se rapprocher d'un milliard de personnes. La surveillance et le traitement de tous ces patients est un défi qu'il est impératif de relever malgré les retombées économiques qui ne cessent d'augmenter et de devenir très onéreuses voir même un véritable handicap surtout pour les pays en développement. Les avancées spectaculaires connues par les systèmes de communications numériques et les systèmes IoT ont permis de réaliser de nouveaux capteurs capables d'une surveillance du taux de glycémie en continue d'une manière non invasive. Cependant, ces capteurs restent encore difficiles à acquérir compte tenu de leurs prix et présentent aussi certaines limitations liées essentiellement à leurs moyens de communications basées sur les protocoles à champs proches ou NFC. Dans ce contexte et à travers notre projet de fin d'études nous nous sommes intéressés à développer et à réaliser un système de surveillance de la glycémie en continu et en ligne. Sur la base d'un capteur FreeStyle d'Abbott fonctionnement en champs proche NFC, nous avons mis en œuvre un système IoT qui lui sera adapté. Ceci dans le but de rendre la portée de cette mesure en continue non invasive plus longue en utilisant une connexion WiFi. Ainsi, par le biais d'un Android les mesures en continu pourront être captées et affichées instantanément par un Smartphone. Ce qui nous permet aussi de consulter ces mesures en lignes à l'aide d'un serveur web

Dans notre mémoire de fin d'études nous avons commencé par poser le problème qui nous a incité à choisir cette thématique. Nous avons ensuite rappelé les principes de base des systèmes IoT objets de notre réalisation. Le second chapitre de notre mémoire a été consacré à la présentation de la réalisation matérielle. Un choix pertinent du système IoT et de la connexion Bluetooth basse énergie ont été motivés par rapport à plusieurs contraintes imposées pour que le système soit flexible, à faible coût, non gênant à porter et avec une autonomie énergétique intéressante. Dans la dernière partie du mémoire nous nous sommes attaqués à la partie la plus importante de la réalisation où il s'agit des différents softwares à mettre en œuvre. Il est question d'un programme qui doit gérer le système IoT, d'une application Android à implémenter dans le Smartphone et d'un site web qui doit communiquer avec le Smartphone pour récolter et mettre en ligne les mesures instantanées. Cette partie, que nous estimons très pertinentes, très complexe et qui nous a obligé d'apprendre et de comprendre plusieurs points essentiels. Ceci nous a poussé à développer beaucoup plus nos connaissances dans la programmation Android et également la création des sites web surtout dynamiques. Il s'agit de la maîtrise d'un nombre considérable de logiciels et d'outils très vastes et qui se développent rapidement de jour en jour.

Ce travail n'est que le prélude d'un travail beaucoup plus large et plus important que nous espérons pouvoir entreprendre dans un futur proche. Nous nous sommes contentés uniquement, dans le travail actuel, à récupérer des données d'un capteur Abbott et les envoyer vers un Smartphone et ensuite vers un site web. Même s'il s'agit d'un travail relativement difficile qui nécessite beaucoup de connaissances dans plusieurs domaines, que nous n'avions pas initialement et que nous avons pu maîtriser durant ce projet, beaucoup d'autres choses très importantes doivent être développées dans un futur proche pour rendre cette application plus pertinente, plus flexible et plus adaptée. Il s'agit avant tout de surveiller des patients diabétiques et de leurs offrir, à eux mais aussi aux médecins et soignants qui les suivent, en ligne des informations qui peuvent leurs permettre de mieux contrôler leur maladie. Ainsi, de nouvelles options peuvent être mis en œuvre afin de rendre cette application plus adaptée dans un contexte non pas seulement de surveillance mais aussi de contrôle. D'autre part, le capteur Abbott utilise un codage de données très particulier. Il est donc nécessaire de mieux comprendre la structure des "datas" envoyées par ce capteur afin de pouvoir bénéficier complètement et amplement des informations véhiculées, en continu et en ligne. Enfin, une autre perspective à prospecter dans ce sens est relative à la miniaturisation et l'économie de consommation énergétique du système IoT qui doit être porté en permanence par le patient. Même si ce premier système proposé répond assez bien à ces contraintes, nous devons œuvrer encore plus pour rendre le système final le mieux adapté possible.

## Références bibliographiques

- [1] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6068508/>
- [2] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5190964/>
- [3] <https://onlinelibrary.wiley.com/doi/full/10.1002/adma.201706910>
- [4] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4434067/>
- [5] R. Maheswar, G. R. Kanagachidambaresan, R. Jayaparvathy, Sabu M. Thampi. "Body Area Network Challenges and Solutions". Livre Springer
- [6] M. Udin Harun Al Rasyid, Ferry Astika Saputra, Alexander Christian. "Implementation of Blood Glucose Levels Monitoring System Based on Wireless Body Area Network". 2016 International Conference on Consumer Electronics-Taiwan.
- [7] Stepan Ivanov, Christopher Foley, Sasitharan Balasubramaniam, and Dmitri Botvich. "Virtual Groups for Patient WBAN Monitoring in Medical Environments". IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 59, NO. 11, NOVEMBER 2012
- [8] E. Villalba, D. Salvi, M. Ottaviani, I. Peinado, M. T. Arredondo, and A. Akay. "Wearable and Mobile System to Manage Remotely Heart Failure". IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 13, NO. 6, NOVEMBER 2009
- [9] Mashael Saud Bin-Sabbar, Mznah Abdullah Al-Rodhaan. Diabetes Monitoring System Using Mobile Computing Technologies". (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 2, 2013. [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [10] Marti Widya Sari, Muhtar Luthfi. "Design and Analysis of Non-Invasive Blood Glucose Levels Monitoring". 2016 International Seminar on Application for Technology of Information and Communication
- [11] Imed Ben Dhaou3, Amir M. Rahm. "IoT-based continuous glucose monitoring system: A feasibility study Tuan Nguyen Gia, Mai Ali". 8th International Conference on Ambient Systems, Networks and Technologies (ANT-2017)
- [12] Miriam Garcia Yanez. Glucose Meter Fundamentals and Design. Freescale Semiconductor Document Number:AN4364. Application Note Rev. 1, 01/2013
- [13] [http://e2e.ti.com/blogs\\_/b/process/archive/2018/09/07/designing-a-blood-glucose-meter-and-pulse-oximeter-with-the-msp430-mcu-smart-analog-combo](http://e2e.ti.com/blogs_/b/process/archive/2018/09/07/designing-a-blood-glucose-meter-and-pulse-oximeter-with-the-msp430-mcu-smart-analog-combo)
- [14] <https://loopkit.github.io/loopdocs/faqs/rileylink-faqs/>
- [15] <https://loopkit.github.io/loopdocs/build/step5/>
- [16] <https://loopkit.github.io/loopdocs/faqs/rileylink-faqs/#what-happens-if-i-walk-away-from-my-rileylink>
- [17] <https://www.medtronicdiabetes.com/loop-blog/introducing-convenient-mobile-access-pump-cgm-data-minimed-connect/>
- [18] <https://diatribe.org/how-i-loop-two-years-using-iphone-app-automate-my-insulin-delivery>
- [19] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6705487/>

- [20] [http://www.stronglink-rfid.com/en/rfid-readers/sl600.html?gclid=EAIaIQobChMIIm8mg9da-5gIVWbvVCh1J0wPsEAAYASAAEgIZNvD\\_BwE](http://www.stronglink-rfid.com/en/rfid-readers/sl600.html?gclid=EAIaIQobChMIIm8mg9da-5gIVWbvVCh1J0wPsEAAYASAAEgIZNvD_BwE)
- [21] <http://www.solutions-cubed.com/bm019/>
- [22] <https://github.com/vicktor/FreeStyleLibre-NFC-Reader/commit/c67681555d92b00f371ddc93dda5a279c8996512>
- [23] <https://github.com/vicktor/FreeStyleLibre-NFC-Reader/blob/master/app/FreeStyleLibre/app/build/outputs/apk/app-debug.apk>
- [24] <https://github.com/vicktor/FreeStyleLibre-NFC-Reader/issues/2>
- [25] <https://www.freestylelibre.de/libre/>
- [26] <https://github.com/arifC/freestyle-sensor-reader/tree/master/.idea/codeStyles>
- [27] <https://github.com/arifC/freestyle-sensor-reader/tree/master/.idea>
- [28] <https://www.frontiersin.org/articles/10.3389/fendo.2019.00697/full>
- [29] Cotton, S. L., R. D’Errico, and C. Oestges. (2014), A review of radio channel models for body centric communications, *RadioSci.*, 49, 371–388, doi:10.1002/2013RS005319.
- [30] Jun-ich Naganawa, Karma Wangchuk, Minseok Kim, Takahiro Aoyagi, and Jun-ichi Takada, Simulation-based Scenario-Specific Channel Modeling for WBAN Cooperative Transmission Schemes. 2014, *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS*
- [31] <https://onlinelibrary.wiley.com/doi/10.1002/adhm.201701150>
- [32] R P Jenie, J Iskandar, A Kurniawan, E Rustami, H Syafutra, N M Nurdin, T Handoyo, J Prabowo, R Febryarto, M S K Rahayu, E Damayanthi, Rimbawan, D Sukandar, Y Suryana, Irzaman, and H Alatas. “Proposed Application of Fast Fourier Transform in Near Infra-Red Based Non Invasive Blood Glucose Monitoring System”. *IOP Conf. Series: Earth and Environmental Science* 58 (2017) 012011. <https://iopscience.iop.org/article/10.1088/1755-1315/58/1/012011>
- [33] Shihong Zou, Yanhong Xu, Honggang Wang, Zhuzhou Li, Shanzhi Chen, and Bo Hu. “A Survey on Secure Wireless Body Area Networks”. *Wiley Hindawi, Security and Communication Networks*. Volume 2017
- [34] José Jorge Rodrigues BARATA, Roberto Munoz, Rafael D. De Carvalho Silva, Joel J. P. C. Rodrigues, Victor Hugo C. De Albuquerque. “Internet of Things Based on Electronic and Mobile Health Systems for Blood Glucose Continuous Monitoring and Management”. *IEEE Access*. Vol 7, 2019
- [35] J. J. P. C. Rodrigues, D. B. De Rezende Segundo, H. A. Junqueira, M. H. Sabino, R. M. Prince, J. Al-Muhtadi, and V. H. C. De Albuquerque, “Enabling technologies for the Internet of health things,” *IEEE Access*, vol. 6, pp. 13129–13141, 2018
- [36] D. Bruen, C. Delaney, L. Florea, and D. Diamond, “Glucose sensing for diabetes monitoring: Recent developments,” *Sensors*, vol. 17, no. 8, p. 1866, Aug. 2017
- [37] P. V. Vandevenne. (2014). Some Thoughts and Experiments on Using the Freestyle Libre as a CGM. [Online]. Available: <http://type1tennis.blogspot.com/2015/02/some-thoughts-on-using-freestyle-libre.html>
- [38] Uwe Petersen. Available : <https://github.com/UPetersen/LibreMonitor>
- [39] <http://www.ti.com/lit/ds/slas834c/slas834c.pdf>
- [40] <https://hal.archives-ouvertes.fr/cel-01241650v2/document>

[41] <https://www.youtube.com/watch?v=M6pi6jXpRrs>

[42] [http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi\\_hal.html](http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi_hal.html)

[43] <https://openclassrooms.com/fr/courses/4425076-decouvrez-le-framework-django/4630835-creez-un-nouveau-projet>

[44] <https://www.windows8facile.fr/xampp-autoriser-connexion-distance-phpmyadmin/>

[45] <https://waytolearnx.com/2018/09/difference-entre-les-pages-web-statiques-et-dynamiques.html>

[46] <https://www.ionos.fr/digitalguide/serveur/outils/tutoriel-xampp-creer-un-serveur-de-test-local/>