

الجمهورية الجزائرية الديمقراطية
الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان -

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER**

En : Télécommunication

Spécialité : Système de télécommunication

Par : GUELLIL Lamia et MAHMOUDI Mohamed Nadir

Sujet

**ÉTUDE ET CONCEPTION D'UN ASSISTANT DE SANTE
PERSONEL**

Devant le jury composé de :

Mr BENADDA Belkacem	Professeur	Université de Tlemcen	Président
Mr BORSALI Ahmed Riad	Professeur	Université de Tlemcen	Encadreur
Mr MERZOUGUI Rachid	Professeur	Université de Tlemcen	Examineur

Remerciement

Avant toutes choses, nous remercions ALLAH, pour avoir donné la force et la patience pour réaliser ce travail.

En tout premier lieu, toute notre gratitude et notre reconnaissance s'adressent aussi à monsieur le professeur Borsali Ahemd Riad qui nous a fait l'honneur accepté de nous encadrer.

Ainsi que pour l'aide compétente nous a apportée, pour sa patience et son encouragement et sa disponibilité. Son œil critique a été très précieux pour structurer le travail et pour améliorer la qualité de ses différentes sections. On veut vraiment vous remercier Professeur.

Nous adressons-nous sincères remerciements à monsieur BENADDA d'avoir accepté de présider le jury.

Nous exprimons vifs remerciements à MERZOUGUI. Pour l'honneur d'examiner et acceptant de faire partie de jury.

Enfin, nous n'oserions oublier de remercier tout le corps professoral de la faculté de technologie surtout le département de télécommunication, pour le travail énorme qu'il effectue pour nous former dans les conditions les plus favorables possible pour le déroulement de nos études.

Dédicace

A ma mère.

A mes frères .Mohcine ,Karim,Otheman

A monsieur KH.Monaim

A ma famille

A mes amies

Lamia

Dédicace

Je dédie ce travail et cet événement marquant de ma vie à:

Mes très chers parents, pour tous leurs sacrifices, leur amour, leur soutien, et leurs prières tout au long de mes études.

Mon petit frère Aness, et mes chères sœurs Houda et Fadwa pour leurs encouragements et surtout Fadwa .

Mes amis, mes camarades et un grand remerciement à Daoud Yasser et son papa Mohamed.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible.

Nadir

Sommaire

SOMMAIRE

Sommaire.....	5
Liste des figures.....	7
Introduction générale.....	9
I-1 Introduction :.....	12
I-2 Historique de la domotique.....	12
I.4. Les objective de la domotique.....	13
I-5 Protocole de la domotique.....	14
1. En Ocean.....	14
2. IO Home control.....	14
3. Z-wave.....	14
4. ZigBee.....	14
5. Les protocoles IP.....	14
6. Universal Plug and Play (Upnp).....	14
7. DPWS.....	14
I-6 L'architecture de la domotique.....	15
I-7 La domotique au service de la santé.....	16
I.7.1. Exemples de la domotique au service de santé.....	17
I.8. La domotique au service des personnes âgées.....	18
I-8-1 Les problèmes d'assistance des personnes âgées.....	18
I.8.3. Quelque innovation de la domotique de santé dédiée au service des seniors.....	19
a. La Smartcane, intelligente et connectée par Nov'in.....	19
d. Fingertips Viktor, le coussin tactile qui permet de se faire livrer des médicaments.....	21
f. LifeInABox, le frigo portatif pour transporter les médicaments.....	22
II-1 Introduction.....	24
II-2 L'historique de la carte arduino.....	24
II-4 Les gammes de la carte arduino.....	25
II-5 Arduino Uno.....	27
II.6 La constitution de la carte Arduino UNO.....	29
II.6.1 Partie matérielle.....	30
II.6.1.1 Le Microcontrôleur ATmega328.....	30
II.6.2 Partie Programmation.....	32
II.6.2.1 L'environnement de programmation.....	32
II.6.2.2 la structure de programmation de la carte Arduino.....	32

II.6.2.2 .1 La fonction setup () :.....	33
II.6.2.2 .2 La fonction loop()	33
II-7 La conclusion :.....	34
3-1-Introduction	36
3-2-Le choix de l'application.....	36
3-3-La description de l'application	36
III-4-Matériels et logiciels déployés	37
III-4-1-Partie matérielle	37
III-4-1-1-le module Bluetooth HC-06	38
III-4-1-2-Accéléromètre ADXL335	39
III-4-2-Partie logiciels	39
III-4-2-1-Windows azure mobile	39
III-4-2-3-L'IDE Arduino	40
III-5-Réalisation.....	40
III-5-1-Principe de détection des chutes avec reconnaissance de posture	40
III-5-2- Conditions préalables.....	41
III-5-3- L'instruction du projet	41
III-5-3-1-Etape 1.....	42
III-5-3-2-Etape 2	43
III-5-3-3-Etape 3	44
III-5-3-4-Etape 4	45
III-5-3-6-Etape 6	46
III-6-Conclusion	48
References Bibliographiques	50
Annexe.....	54

Liste des figures

Chapitre 1 : Introduction a la domotique

1. Figure 1.1 : Exemples de la domestique [2].....	13
1.1 Figure 1.1.1: propriétés structurelles [1.1].....	14
2 Figure 1.2: Réseau domotique à deux niveaux. Internet le protocole est sur le niveau 1, hétérogène sans fil à courte portée les sous-réseaux sont au niveau 2.[4]	16
3 figure 2.3 La Smartcane, intelligente et connectée par Nov'in	20
4 Figure 1.4 Assistance auto reeducation -Ted Orthopedics[15]	20
5 Figure 1.5 BUDDY Robot [16]	21
6 Figure 1.6 Fingertips Viktor[15].....	21
7 Figure 1.8 lumeen.....	22

Chapitre 2 :Systemes Arduino

8 Figure 2.1: La carte arduino [20].....	25
9 Figure 2.2 : arduino Uno WiFi Rev 2 [21].....	25
10 Figure 2.3 : la carte Portenta H7 [22].....	26
11 Figure 2.2 : MKR Vidor 4000 [23]	26
12 Figure 2.5: Carte Arduino mega [25].....	27
13 FFigure 2. 6 : carte Arduino Esplora [26].....	27
14 Figure 2.7 : Arduino Uno [29].....	28
15 Figure 2.8 : constitution de la carte Arduino Uno [31]	29
16 Figure 2.9: spécification de la carte Arduino Uno [32]	30
17 Figure 2.10:Architecture de ATmega 328 [34]	31
18 Figure 2.11 : la Structure de base d'un programme d'Arduino[37].	32
19 Figure 2.12: Exemple de la fonction setup () sous Arduino[38].....	33
20 Figure 2.13 : Exemple de la fonction Loop () sous Arduino [40].....	33
21 Figure 2.24 : Structure d'un programme.[39]	34

Chapitre 3 : Conception et réalisation d'un assistant de santé personnel

22	Figure.3.1 : chute d'une personne âgée [40].....	36
23	Figure.3.2 : Architecture du système [41]	
24	Figure.3.3: Module Bluetooth HC-06 [43]	38
25	Figure.3.4 : Accéléromètre ADXL335	39
26	26 Figure.3.5 : schéma de chute [49]	41
26.1	Figure 3.6.b: commandes AT du micro logiciel hc-06 [50].....	42
26.1.1	Figure 3.6.a : notre propre branchement	42
28	Figure 3.6.1 : notre propre branchement.....	42
29	Figure.3.7: Croquis de la connexion Bluetooth HC-06[52]	44
30	Figure.3.8: Couplage Bluetooth pour Windows 10 mobile	45
31	Figure.3.9: Création d'une table de stockage azure [53]	46
32	Figure.3.10: Ajout des informations de compte	47
33	32 Figure 3.11: Ecran de démarrage de l'application [54]	47

chapitre 2 : Système arduino

	Tableau 2.2 :Chronologie des sorties de cartes Arduino [24]	26
	Tableau 2.2: spécification de la carte Arduino Uno [32].....	30

Chapitre 3 : Conception et réalisation d'un assistant de santé personnel

	Tableau.32 : commandes AT du micro logiciel hc-06 [50].....	43
--	---	----

Introduction générale

Le développement technologique a connu un essor extraordinaire, en fait la technologie moderne a touché tous les secteurs et a changé le mode de vie des personnes. L'apparition de l'internet qui est un réseau informatique de télécommunication à l'échelle du monde, reposant sur le protocole de communication IP (Internet Protocol) depuis le début des années 1990 a changé le dogme des systèmes technologiques et a donné la naissance des objets connectés, petits à petit les objets électroniques utilisent l'internet, afin de profiter d'autres options plus intéressantes notamment l'usage du téléphone intelligent ou smartphone pour contrôler les objets électroniques. Le développement des microcontrôleurs a permis de créer des systèmes intelligents connectés qui peuvent contrôler la santé humaine.

L'arrivée de la domotique a permis l'amélioration de la qualité de vie de plusieurs personnes surtout les seniors et les handicapés, ces technologies ont permis le contrôle et l'automatisation de leurs maisons, en matière de sécurité, confort, gestion de l'énergie, communication, etc...

Parmi les applications de la domotique sur la santé des personnes âgées, c'est le contrôle de leurs paramètres biologiques exemple : pression artérielle, rythme cardiaque, sa température corporelle, son taux de glycémie (pour le cas d'un diabétique), la quantité et la qualité de l'air ect.

Parmi les préoccupations permanentes des gens, la surveillance des personnes âgées vis-à-vis des chutes brusques et le besoin d'intervenir dans ce cas. Ces chutes peuvent entraîner des blessures mortelles, elles sont devenues un grand défi qui ne peut être négligé pour les personnes âgées.

Au cours des dernières années, différents types d'approches ont été proposées dans la zone de détection des chutes, qui peuvent être expliquées et classées en trois types: basées sur les appareils portables, basées sur les capteurs de l'environnement et basées sur la surveillance de la vision.

La réalisation de cette approche nécessite que les appareils portables profitent généralement des capteurs intégrés pour détecter le mouvement et l'emplacement du corps, tels que l'accéléromètre, le magnétomètre et le gyroscope.

Notre projet vise à établir un système de surveillance des personnes âgées basé sur Arduino, Windows 10 et Microsoft Azure, cela va permettre de détecter les chutes en temps réel fournissant ainsi des données de l'accéléromètre au moment de la chute à des personnes intéressées qui surveillent ces personnes âgées. Les données brutes de l'accéléromètre tridimensionnel sont fournies par Arduino avec ADXL345. L'appareil Windows 10 utilise ces informations pour obtenir l'orientation du sujet grâce à des algorithmes efficaces de fusion des données et de détection des chutes. Les services Microsoft Azure et l'application Mobile / PC sont également utilisés pour réaliser un traitement, une analyse, un stockage et une acquisition de données sans interruption à tout moment et en tout lieu, tant qu'ils ont accès à Internet.

Dans le premier chapitre, on va présenter la domotique et certains protocoles ainsi que les secteurs d'application, notamment l'utilisation pour les personnes âgées.

Le deuxième chapitre présente une description générale de la carte Arduino, ses caractéristiques, sa programmation et son principe de fonctionnement.

Dans le troisième chapitre on va présenter une description générale de notre application, avec les différents composants et les logiciels utilisés. Cette application va être décrite étape par étape.

Enfin, on termine avec une conclusion générale et des perspectives.

Chapitre 1 :

Introduction à la domotique

I-1 Introduction :

L'habitat du futur est un souci actuel qui a plusieurs objectifs, dont celui de la sécurité des biens et des personnes, le développement technologique a connu un essor impressionnant rendant ainsi tout objets contrôlables à distance et tout phénomène biologique ou physique peut-être détecter, contrôler ou mesurer. Le vieillissement de la population moderne est un phénomène qui ne peut pas être négligé et qui est en constante augmentation. Le maintien à domicile des personnes âgées est un enjeu et désire vieillir chez elles en logement autonome et la majorité refuse d'être placé dans une maison de retraite pour des raisons financières et morales. .

Les plus grands risques chez ces personnes âgées sont les chutes brutales qui leur coute parfois leurs vies où les rendre handicapés. La domotique peut constituer une solution idéale pour surveiller les personnes âgées lorsqu'ils sont seuls ou la nuit lorsque les personnes qui s'occupent d'eux sont endormies du fait de la disponibilité des capteurs qui peuvent surveiller leur mouvement d'une manière constante et de signaler n'importe chute du a un déséquilibre ou une crise a d'autres personnes proches ou de protection civile pour venir à leurs secours.

I-2 Historique de la domotique

L'histoire de la domotique a commencé en 1975, où a été développé comme le premier protocole de communication sans fil, c'était parmi les premier dispositifs pour la domotique qui utilisaient des lignes électriques pour envoyer des rafales de signaux numériques pour contrôler appareils électroménagers d'une maison. En 1978, des fabricants ont développé plus technologies contrôlées par ordinateur telles que l'éclaircissement interrupteurs et contrôleurs de thermostat. En 1984, les technologies domestiques smart comprenaient les contrôleurs de portes de garage, appareils de divertissement, télécommandes et systèmes de sécurité. En 1998, la domotique gagné en popularité et en technologies différentes émergé et est devenu visiblement intégré avec de nouvelles technologies domestiques, instruments domestiques et mise en réseau. À la fin de 2012, le nombre de maisons intelligentes aux États-Unis ont dépassé 1,5 millions[2] .

I-3 La définition de la domotique

La domotique est la science qui s'intéresse à l'étude des technologies conçues pour améliorer la qualité de vie à la maison et plus généralement dans les environnements anthropisés. Ce domaine fortement interdisciplinaire nécessite la contribution de nombreuses technologies et professions, parmi lesquelles l'ingénierie du bâtiment, l'ingénierie énergétique, l'automatisation, la technologie électrique, l'électronique, les télécommunications et les technologies de l'information et des communications.

La domotique permet de mettre à la disposition des utilisateurs des appareils allant au-delà du «traditionnel» où les équipements et systèmes peuvent réaliser des fonctions partiellement autonomes

(en fonction des réactions aux paramètres environnementaux fixes et préétablies), ou programmées par l'utilisateur, ces appareils sont complètement autonomes (ils agissent comme des réactions aux paramètres environnementaux et sont activés par des programmes dynamiques). Grâce à la domotique, il est possible d'intégrer divers utilisateurs et fonctions domestiques, qui étaient traditionnellement distincts et séparés, en un seul réseau coopératif et interopérable pour pouvoir créer des fonctions dynamiques de plus en plus avancées et intelligentes[3].



Figure 1.1 : Exemples de la domotique [2]

I.4. Les objectifs de la domotique

Parmi les enjeux de cette technologie :

- 1- Le confort : le contrôle des appareils par les individus en utilisation de 'scénario domestique.
- 2- Economie : l'économie au niveau de budget domestique par exemple économie d'énergie non utilisable par les individus pour éviter l'extravagance de consommation ainsi que l'extravagance de l'espace (exemple réduction d'espace du mobilier).
- 3- La sécurité : augmentation sensation de sécurité pour les individus à travers les différentes applications de la domotique dans la vie quotidienne tel que les caméras de surveillance, les alarmes, le détecteur de mouvement.

Presque ou tous les domaines peuvent s'adapter aux les différents principes de la domotique tel que domaine médicale, sécurité, surveillance, industrie

I-5 Protocole de la domotique

Il existe plusieurs protocole de la domotique tel que le wifi et le Bluetooth mais les principaux protocoles Wi-Fi et Bluetooth ne peuvent pas être directement comparés car ce ne sont pas des technologies domotiques en tant que tel, mais fonctionnent dans une couche inférieure [1]. Autre protocoles tel que :

1. En Ocean : Chaque appareil a un ID unique 32 bits comme partie fixe de chaque message transmis pour empêcher l'activation du mauvais Composants.
2. IO Home control : IO-home control est conçu comme un OSI à sept couches la mise en oeuvre. Sur la couche application, divers services, les profils du capteur et de l'actionneur sont définis. Le protocole fournit une authentification bidirectionnelle via un 48 bits défi-réponse.
3. Z-wave : Z-wave demeure une technologie protégée à source unique qui permet au dispositifs pour coopérer uniquement par le biais de l'Alliance Z-wave.
4. ZigBee : Basé sur la norme ouverte IEEE 802.15.4, le Zigbee spécification définit le maillage et diverses applications couches de support.
5. Les protocoles IP : Les protocoles suivants définissent la couche d'application, en fonction sur le protocole Internet et HTTP. Par conséquent, ils peuvent travailler sur haut de n'importe quel protocole compatible IP tel que Wi-Fi ou Bluetooth.
6. Universal Plug and Play (Upnp) : Upnp construit sur TCP/IP en utilisant principalement HTTP, XML(langage de balisage extensible) et SOAP (SOAP (ancien acronyme de Simple Object Access Protocol) est un protocole d'échange d'information structurée dans l'implémentation de services web bâti sur XML).. il est indépendant de savoir si les appareils sont câblés ou sans fil, et la portée et la bande passante sont déterminés par la technologie IP utilisée.
7. DPWS : « Devices Profile for Web Services » Il définit un ensemble minimal de contraintes pour permettre le Web sécurisé messagerie, découverte et saisie d'événements. Il s'appuie sur un certain nombre de normes de service Web, y compris WS-Security(WS Security (Web Services Security) est un protocole de communications qui permet d'appliquer de la sécurité aux services web), qui est basé sur SOAP . [1]

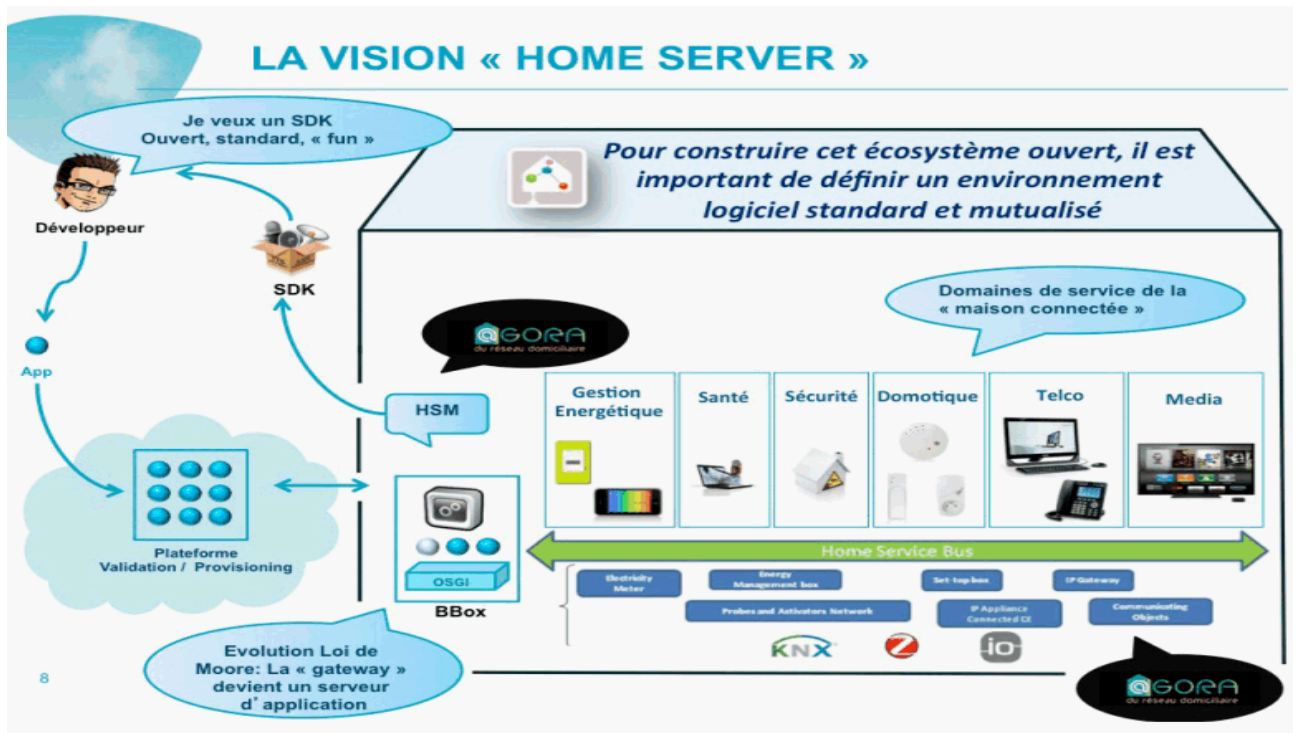


Figure 1.1.1: propriétés structurelles [1.1]

I-6 L'architecture de la domotique

Une plateforme qui prend en charge un réseau hétérogène sécurisé l'interopérabilité pourrait améliorer la sécurité de la domotique. En utilisant des technologies qui offrent des garanties de sécurité très élevées, mais adaptées à la maison. Il devrait ouvrir le réseau sans fil à courte portée SRW (Short Range Wireless) réseaux de périphériques pour une utilisation par tout logiciel dans le réseau, plutôt que un tunnel de toute communication à travers une passerelle de traduction qui nécessiteront des mises à jour sans fin. Il devrait fournir une communication ouverte d'une manière agnostique vers une application ayant des modèles spécifiques, technologies SRW, technologies de contrôle, OS smartphone, applications de serveur cloud, etc. Il doit y avoir une méthode générale d'adaptation des nouveaux types de dispositifs et du SRW réseaux. Et enfin, la sécurité de la plateforme doit être basée sur des technologies fiables et testées telles que SSL. Tout nouveau protocole ou mécanismes mis en place pour compléter ces technologies, la fonctionnalité et la sécurité devraient être formellement vérifiées.

L'interopérabilité des sous-réseaux SRW est gérée via cette infrastructure d'automatisation

Figure 1.2. Les dispositifs finaux sont services de réseau SRW de niveau 2 qui peuvent être découverts et accessible par les contrôleurs IP de niveau 1. Ce système prend en charge les technologies existantes et les technologies futures, systèmes qui seront développés sans interopérabilité entre les technologies à l'esprit en raison des adaptateurs.

Niveau 1 est le réseau IP soit filaire ou Wi-Fi et/ou les contrôleurs locaux, smartphones, tablettes, ordinateurs portables, ordinateurs de bureau, et d'autres dispositifs de contrôle IP-capable se connecter. Il y a une Passerelle Internet pour chaque segment de réseau physique, qui autorise les connexions entrantes afin de fournir une communication sécurisée.

Le matériel est un routeur domestique standard avec des petits logiciels supplémentaires pour maintenir les coûts faible. Les contrôleurs d'automatisation locaux gèrent certains appareils finaux dans le système et fournis une fonctionnalité continue dans la maison, tandis que les contrôleurs sur les appareils portables ou le nuage les serveurs réagissent à l'entrée de l'utilisateur seulement lorsque il est demandé.

Le niveau 2 se compose de sous-réseaux créés par tous les Technologies hétérogènes SRW. Les capteurs sans fil et les actionneurs et les terminaux participent à ces réseaux et à la communication est relié de manière transparente au réseau IP par Ponts de salle. Un adaptateur de sous-réseau fournit le SRW spécifique communication et sépare la charge utile du message de la technologie SRW afin d'encapsuler ces concepts comme adressage SRW et formats de message de la couches supérieures. Un adaptateur est nécessaire pour connecter chaque SRW. La technologie à portée du pont; ceux-ci peuvent être intégrés dans le pont, mais il peut être préférable de faire, p. ex., un USB externe adaptateurs afin de garder le pont générique.

Les ponts peuvent aussi s'assurer que tous les dispositifs d'extrémité SRW sont repérables et accessible par les contrôleurs de niveau 1 en maintenant le service. [4].

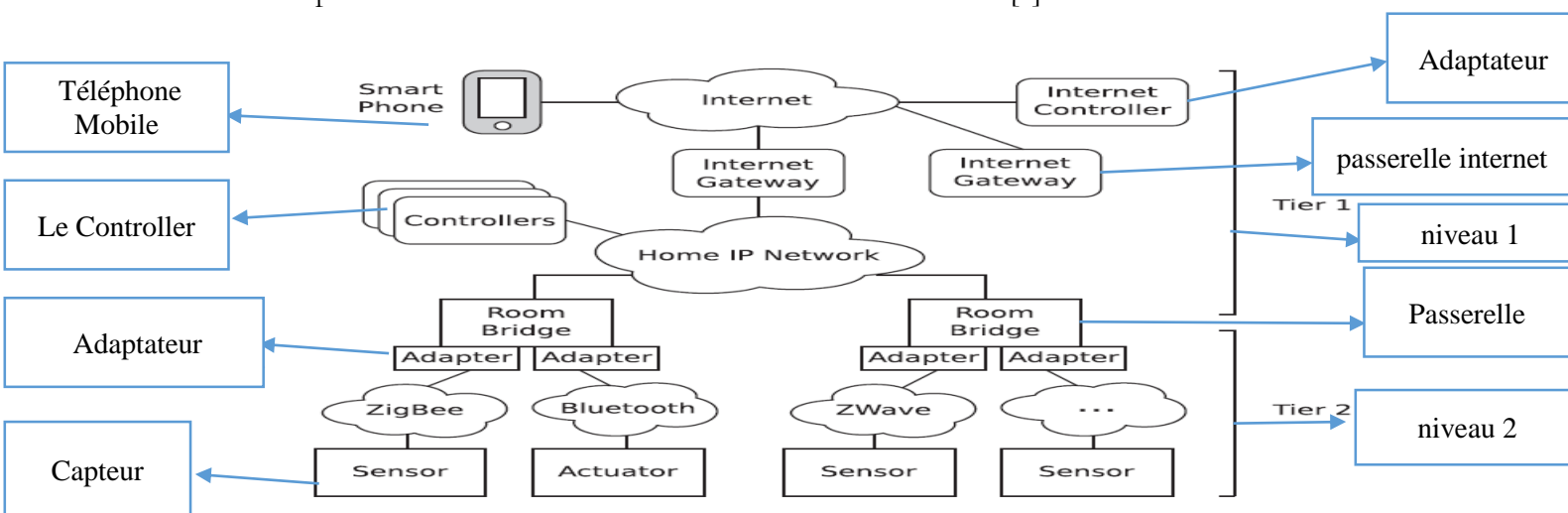


Figure 1.2: Réseau domotique à deux niveaux. Internet le protocole est sur le niveau 1, hétérogène sans fil à courte portée les sous-réseaux sont au niveau 2. [4]

I-7 La domotique au service de la santé

L'estimation que la population mondiale devrait croître d'une estimation de 2,3 milliards d'ici 2050. Leur fournir des soins de santé deviendra beaucoup plus difficile. Pour mieux se préparer à cela, l'industrie actuelle des soins de santé est en transformation [5]. En raison de vastes améliorations de

l'apprentissage automatique et de la révolution de l'apprentissage en profondeur, les ordinateurs apprennent désormais par eux-mêmes, uniquement avec des données en entrée et le temps de traiter les données [6].

Les objectifs de fournir de meilleurs services de santé et d'améliorer la qualité de vie des citoyens conduisent automatiquement à considérer les technologies basées sur l'Internet des objets (IoT). Ces derniers joueront un rôle clé dans la conception, le développement et le maintien de services et de solutions de santé plus intelligentes, connectés et personnalisés. Ces services peuvent permettre une surveillance continue des conditions physiques et leur traitement automatisé [6].

Les dispositifs de soins de santé intelligents prolifèrent dans l'environnement domestique, ce qui introduit l'avantage potentiel d'améliorer certains aspects de la vie autonome. Ces dispositifs sont particulièrement bénéfiques pour ceux qui ont besoin d'aide et d'intervention en matière de soins de santé à domicile [7].

Autres dispositifs et systèmes relevant de ce cadre des dispositifs de santé intelligents sont ceux qui interagissent physiquement avec l'utilisateur pour fournir un soutien préventif et thérapeutique. Par exemple, des stimulateurs de mollet contre les veines profondes thrombose, ou dispositifs fournissant un soutien thérapeutique pour patients qui se remettent d'une maladie, d'un traumatisme ou d'une intervention postopératoire. À titre d'exemple du dispositif orienté vers la thérapeutique. [7].

I.7.1. Exemples de la domotique au service de santé

1. L'un des systèmes d'IA (intelligent artificiel) les plus puissants, IBM Watson, est actuellement testé dans le domaine médical pour aider les médecins à diagnostiquer et à trouver les meilleurs traitements (IBM, 2015). Cela permet de décentraliser les conseils individuels en matière de soins de santé à grande échelle. La télésanté à domicile est une stratégie clé qui combine les soins de santé, les équipements médicaux électroniques et les technologies de communication dans le but de fournir des soins préventifs et des services de santé dans la communauté et les environnements domestiques [6].
2. Plusieurs dispositifs automatisés de distribution de médicaments sont actuellement disponibles dans le commerce, par exemple CompuMed.
3. Le dernier de ces appareils, MD2, peut-être une offre de la fonctionnalité la plus large, car cet appareil particulier peut stocker, organiser, dispenser, surveiller et rappeler aux patients leurs régimes de médication. Si le patient ne prend pas leurs médicaments après plusieurs rappels, le système peut être configuré pour alerter automatiquement un membre de la famille ou un soignant par téléphone.
4. Un système similaire, récemment mis en place développé, est le système MEDICATE. Ce système offre des fonctionnalités similaires au MD2, mais comprend également une

unité portable qui se détache de l'unité à domicile. l'avantage du composant portable supplémentaire est la capacité de fournir un continuum de prestation de services une fois que la personne quitte leur environnement familial.[7]

Il existe plusieurs domaines d'application de la domotique appliquée a la santé tel que la domotique au service des senior et la domotique au service handicap et comme exemple appliqué de la domotique de la santé, c'est le projet que on ait en train de le réalisé.

I.8. La domotique au service des personnes âgées

De nos jours, un grand nombre de personnes âgées dépendantes nécessitant des services de soins de santé doivent quitter leur domicile et déménager dans des centres de soins de santé spéciaux, ce qui n'est souvent pas acceptable pour eux. Les réseaux de communication avancés et les technologies Internet jouent un rôle clé dans le développement des maisons intelligentes. Les maisons intelligentes intégrées aux soins de santé peuvent être utilisées pour permettre aux personnes âgées et aux membres de leur famille de bénéficier d'un environnement familial sûr avec des activités et des services d'accompagnement pour leurs étapes quotidiennes. Ces technologies jouent un rôle clé dans le développement des maisons intelligentes [8].

Les maisons intelligentes sont considérées comme une bonne alternative pour la vie indépendante des personnes âgées et des personnes handicapées. De nombreux appareils intelligents, intégrés dans l'environnement domestique, peuvent fournir au résident d'une maison de retraite ou au niveau de son domestique une aide à la mobilité et surveillance de la santé 24h / 24.

Des statistiques récentes montrent une tendance à la croissance rapide du nombre de personnes handicapées physiques et de personnes âgées qui ont besoin d'une aide extérieure dans leurs tâches quotidiennes de mouvement. Le problème des soins aux personnes âgées et aux personnes handicapées physiques deviendra plus grave dans un avenir proche lorsqu'une partie importante de la population mondiale croissante est dans le groupe des 65 ans ou plus et que le modèle de bien-être actuel n'est pas en mesure de répondre les besoins accrus [9].

I-8-1 Les problèmes d'assistance des personnes âgées

Les personnes âgées souffre de plusieurs problèmes de santé leur rendre ainsi vulnérables et exposés au danger de chutes ou de crises ou de blessures et donc requièrent une assistance permanente, toutefois cette assistance est difficilement réalisable au niveau de leurs maison du fait que beaucoup parmi eux se trouvent seules et exposés au différents danger. Les études et les statistiques montrent que la plupart de ces personnes âgées refusent de résider dans une maison de retraite et préfèrent rester dans leurs domestique à côté de leurs proche cependant, une assistance permanente dans les 24 heures est presque impossible[10].

1.8.2. Problème des chutes chez les personnes âgées

Les chutes chez les personnes âgées sont une préoccupation majeure en matière de soins de santé en raison des blessures et des décès associés aux chutes. Environ un tiers des personnes âgées de plus de 65 ans chutent chaque année et la moitié des sujets âgés de plus de 85 ans font une ou plusieurs chutes par an, Les conséquences physiques (fracture) et psychologiques d'une chute sont nombreuses. Elles peuvent entraîner une diminution de la mobilité, une perte d'autonomie, une perte de la confiance en soi, une limitation des activités quotidiennes et accélérés ainsi le déclin des capacités fonctionnelles. Elles engendrent un nombre important d'hospitalisations, la fracture de la hanche en est ainsi le principal motif. Elles constituent la principale cause de décès par traumatisme dans cette population (12 000 décès par an), et le facteur d'entrée dans la dépendance : 40 % des sujets âgés hospitalisés pour chute sont orientés ensuite vers une institution. Chaque chute est susceptible de provoquer la hantise d'une nouvelle chute avec comme conséquence la crainte et la limitation spontanée de la marche et la spirale de la dépendance. Ainsi, toute chute chez une personne âgée doit être prise en charge non comme un simple accident mais comme une affection potentiellement grave [11].

Les facteurs de risque de chutes sont multiples. Dans la littérature, les études évaluant le risque de chute parmi les sujets âgés distinguent les facteurs de risque intrinsèques et extrinsèques. Les facteurs de risque intrinsèques sont : les antécédents de chute, l'âge avancé , les troubles de la posture et de l'équilibre , la diminution des capacités sensorielles (baisse de l'acuité visuelle, surdité, diminution de la sensibilité proprioceptive) [12], certaines maladies chroniques (maladie de Parkinson, diabète, séquelles d'accident vasculaire cérébral) et les troubles cognitifs . Les facteurs de risque extrinsèques sont : la polymédication, notamment les hypnotiques, l'environnement inadapté, la peur de chuter, le niveau socio-économique faible, les vêtements et les chaussures inadaptés. Le risque de chute augmente avec le nombre de facteurs de risque.

Plusieurs études ont démontré qu'une diminution d'acuité visuelle augmente le risque de chute accidentelle [13] .

I.8.3. Quelques innovations de la domotique de santé dédiée au service des seniors

a. La Smartcane, intelligente et connectée par Nov'in

L'objectif de cette canne est d'éviter la chute et ses conséquences telles que la peur de se mouvoir, sortir seule... Comme l'explique Vincent Gauchard, co-fondateur de Nov'in, « l'objectif est d'éviter les pertes de confiance qui surviennent après ces incidents » et qui peuvent entraîner une fragilité – premier stade vers la perte d'autonomie si cela n'est pas traité rapidement.

Le fonctionnement de la Smartcane est simple, dans un premier temps, grâce à ses capteurs de mouvement, elle enregistre les habitudes de son utilisateur comme l'heure du réveil et les activités

de son quotidien. Ces données vont être les éléments de référence lui permettant instantanément de détecter lorsqu'une situation est inhabituelle. Dans ce cas, une alerte est envoyée aux personnes choisies.[14]



Figure 1.3 La Smartcane, intelligente et connectée par Nov'in

b. Ted Orthopedics

Cette société a développé un assistant d'auto-rééducation connecté dédié aux patients souffrant d'arthrose du genou, une pathologie touchante. Grâce à cette solution, le patient peut s'auto-réduquer à son domicile, tout en bénéficiant d'un suivi à distance par son professionnel de santé. Le prix Med4Age permettra de valider cliniquement une nouvelle génération de soins de traitement de la douleur liée à l'arthrose de genou des seniors afin de lancer un produit spécifique. [15]



Figure 1.4 Assistance auto reeducation –Ted Orthopedics[15]

c. **BUDDY, le compagnon des séniors isolés**

BUDDY est un robot-compagnon qui a pour mission d'assister les séniors qui souhaitent rester plus longtemps chez eux. Plusieurs études s'accordent à dire que la solitude est le facteur le plus associé au risque de décès. [16]



Figure 1.5 BUDDY Robot [16]

d. Fingertips Viktor, le coussin tactile qui permet de se faire livrer des médicaments

Viktor a été sacré « meilleur produit » à la SilverNight 2017. Il s'agit d'un coussin connecté qui s'adapte parfaitement aux problématiques des personnes en situation de handicap, âgées ou fragiles. Il permet en effet de garder contact avec la famille et d'interagir directement avec les autres objets connectés de la maison.



Figure 1.6 Fingertips Viktor[15]

Développe des expériences, basées sur la réalité virtuelle, combinant musicothérapie, sophrologie et environnements calmes (plages paradisiaques, forêts, montagnes, aurores boréales...) pour apaiser et stimuler les résidents seniors anxieux en établissements. Avec l'aide du Prix Med4Age, Lumeen va mettre en place une étude clinique pour évaluer scientifiquement l'efficacité de sa solution sur le bien-être, la qualité de vie et la cognition des personnes âgées (dont celles avec troubles cognitifs légers). Sur une période d'étude de 16 mois, seront organisés 192 ateliers immersifs collectifs avec usage de casques de réalité virtuelle.



Figure 1.8 lumeen [15.1]

f. LifeInABox, le frigo portatif pour transporter les médicaments

Voici un constat simple : les patients atteints de maladies chroniques sont prisonnières de leurs traitements. En effet, certains médicaments doivent être conservés dans un environnement à température optimale. La société LifeIna a souhaité rendre à ces personnes leur liberté. C'est ainsi qu'est né LifeInABox, le plus petit frigo intelligent du monde !

Conçu selon une démarche orientée utilisateur, ce mini frigo permet désormais à tout un chacun de conserver des médicaments à température idéale. Très simple d'utilisation, son poids léger (790g) et son prix abordable (200€) en font la solution idéale pour mieux vivre le quotidien.

1.9. Conclusion

Les maisons intelligentes deviendront plus omniprésentes à l'avenir en ce qui concerne l'automatisation, la durabilité, santé..... Au fil du temps, le besoin d'utiliser les technologies modernes afin de contrôler, surveiller, économiser rappeler, ect... est devenue une nécessité surtout pour certains cas comme celui des personnes âgées, leurs fragilités et expositions aux dangers nécessitent une surveillance en continue. La domotique représente une solution extraordinaire aux séniors pour profiter de la sécurité, surveillance de leurs fonctions biologiques et avertissement de leurs proches en cas de crise ou de chutes brusques.

Chapitre 2 :

Le système Arduino

II-1 Introduction

Actuellement le système Arduino est devenu un outil indispensable dans la plupart des domaines grâce à des avantages et des gammes multiples qui dépendent de son application.

Dans ce chapitre on va jeter la lumière sur le système Arduino, on va commencer tout abord par citer l'historique et définir ce fameux système ensuite on va citer les différentes gammes de ce système on va essayer de préciser pourquoi on a choisi le système Arduino Uno. Après sa on va démontrer la constitution de ce système en deux étapes : l'Etape construction et Etape programmation et on finira par une conclusion.

II-2 L'historique de la carte arduino

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea (Italie). En 2003, Hernando Barragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée Wiring, accompagnée d'un environnement de programmation libre et ouvert. Pour ce travail, Hernando Barragan réutilisait les sources du projet Processing. Basée sur un langage de programmation facile d'accès et adaptée aux développements de projets de designers, la carte Wiring a donc inspiré le projet Arduino (2005).

Conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Cuartielles, Massimo Banzì ainsi que Nicholas Zambetti), l'environnement Arduino est particulièrement adapté à la production artistique ainsi qu'au développement de conceptions qui peuvent trouver leurs réalisations dans la production industrielle.

Le nom Arduino trouve son origine dans le nom du bar dans lequel l'équipe avait l'habitude de se retrouver. Arduino est aussi le nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort » [17]

II-3 Définition de la carte arduino :

une carte électronique basée autour d'un microcontrôleur et de composants minimum pour effectuer des fonctions plus ou moins avancées à moindre coût Il a une interface usb pour le programmer. Il s'agit d'une plateforme open source basée sur une simple carte microcontrôleur (de la famille AVR { AVR est le terme utilisé par Atmel pour désigner le cœur du processeur et la famille de microcontrôleurs qui le mettent en œuvre[18]}) et d'un logiciel, véritable environnement de développement intégré, pour l'écriture, la compilation et le transfert du programme sur la carte microcontrôleur.

Arduino peut être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs (création artistiques par exemple), et peut recevoir en entrées une très grande variété de capteurs. Arduino peut aussi contrôler une grande variété d'actionneurs (lumières, moteurs ou toutes autres sorties matériels). Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées préassemblées) ; le logiciel de développement open-source est téléchargeable gratuitement [19].

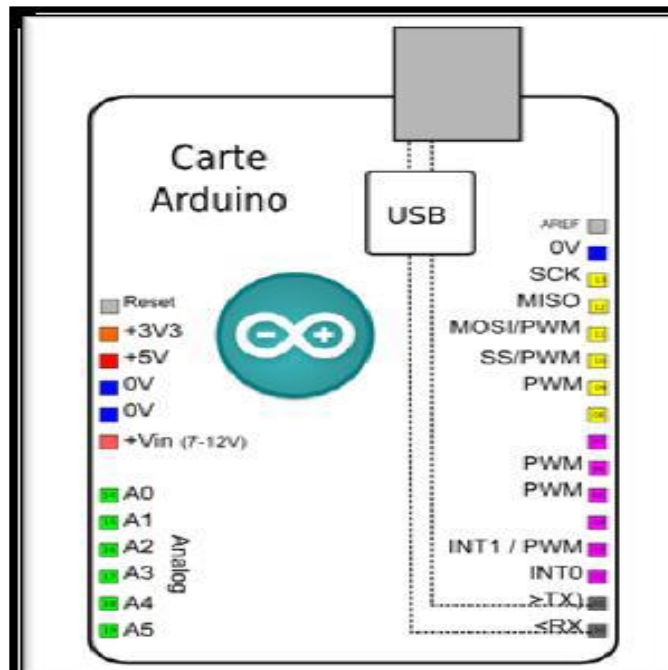


Figure 2.1: La carte arduino [20]

II-4 Les gammes de la carte arduino

De nos jours il existe un très grand nombre de carte arduino et chaque gamme dépend de son domaine application, et son utilisation .Parmi ces cartes, qui on a récemment développer on 02/2020 c'est la carte Portenta H7 (figure 2.2) et l'Arduino MKR Vidor 4000 (figure 2.4) et arduino Uno WiFi Rev 2 (figure 2.3)qui était developper l'an 2018 .



Figure 2.2 : arduino Uno WiFi Rev 2 [21]



Figure 2.3 : la carte Portenta H7 [22]

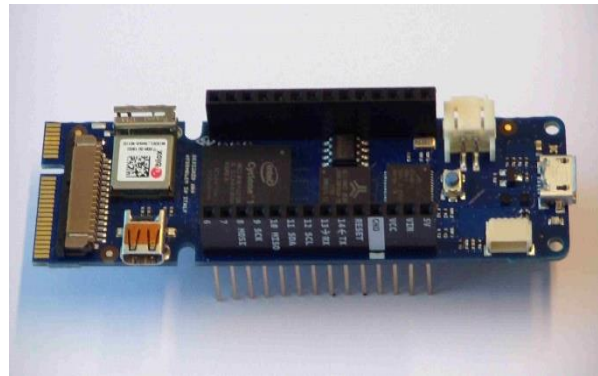


Figure 2.2 : MKR Vidor 4000 [23]

On vas citer maintenant chronologie des sorties de cartes Arduino avec sans microcontrôleur (tableau 2.1).

Tableau 2.1 :Chronologie des sorties de cartes Arduino [24]

Version de carte	Année de sortie	Micro-contrôleur
Diecimila (figure 2.6)	2007	ATmega168V
LilyPad (figure 2.5)	2007	ATmega168V/ATmega328V
Nano	2008	ATmega328/ATmega168
Mini	2008	ATmega168
Mini Pro	2008	ATmega328
Duemilanove	2008	ATmega168/ATmega328
Mega (figure 2.7)	2009	ATmega1280
Fio	2010	ATmega328P
Mega 2560	2010	ATmega2560

Uno	2010	ATmega328P
Ethernet	2011	ATmega328
Mega ADK	2011	ATmega2560
Leonardo	2012	ATmega32U4
Esplora	2012	ATmega32U4
Micro	2012	ATmega32U4
Yún	2013	ATmega32U4 +Linino



Figure 2.5: Carte Arduino mega [25]



Figure 2.6 : carte Arduino Esplora [26]

Notre choix s'est porté sur la carte Arduino Uno.

II-5 Arduino Uno

Le modèle UNO (figure 2.7) de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. L'ATmega328 est un microcontrôleur 8bits de la famille AVR { AVR est le terme utilisé par Atmel pour désigner le cœur du processeur et la famille de microcontrôleurs qui le mettent en œuvre[27]} dont la programmation peut être réalisée en langage C/C++ [28].



Figure 2.7 : Arduino Uno [29]

La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. C'est la plus récente et la plus économique carte à microcontrôleur d'Arduino. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enfiler une série de modules complémentaires. Elle peut se programmer avec le logiciel Arduino. Le contrôleur ATmega328 contient un bootloader qui permet de modifier le programme sans passer par un programmeur. Le logiciel est téléchargeable gratuitement. Cette carte est livrée avec un support en plastique mais sans cordon USB (voir articles conseillés)[30].

II-5-1 Les avantages de la carte Arduino Uno

Il existe beaucoup d'avantages de la carte :

1. La facilité d'utilisation : seulement deux étapes {partie de montage du circuit et puis la programmation} .
2. Le prix : pas cher avec un prix raisonnable.
3. Peut être adapté avec des différents systèmes d'exploitation.
4. Système open source.
5. Programmation très claire, facile et simple.
6. La possibilité d'ajouter plusieurs Shields .
7. Réalisation de programmes complexes.
8. Peut recevoir des informations analogiques et numériques.
9. Stocker un programme et de le faire fonctionner.

II-5-1 -2 Les inconvénients

La carte Arduino Uno a aussi certains inconvénients, et parmi les inconvénients majeurs sont :

1. Tension d'alimentation limitée 6-20V

2. Utilisation câbles Grove

II.6 La constitution de la carte Arduino UNO

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Elle dispose

:

- de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée))
- de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- d'un quartz 16Mhz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP (programmation "in-circuit"),
- et d'un bouton de réinitialisation (reset)

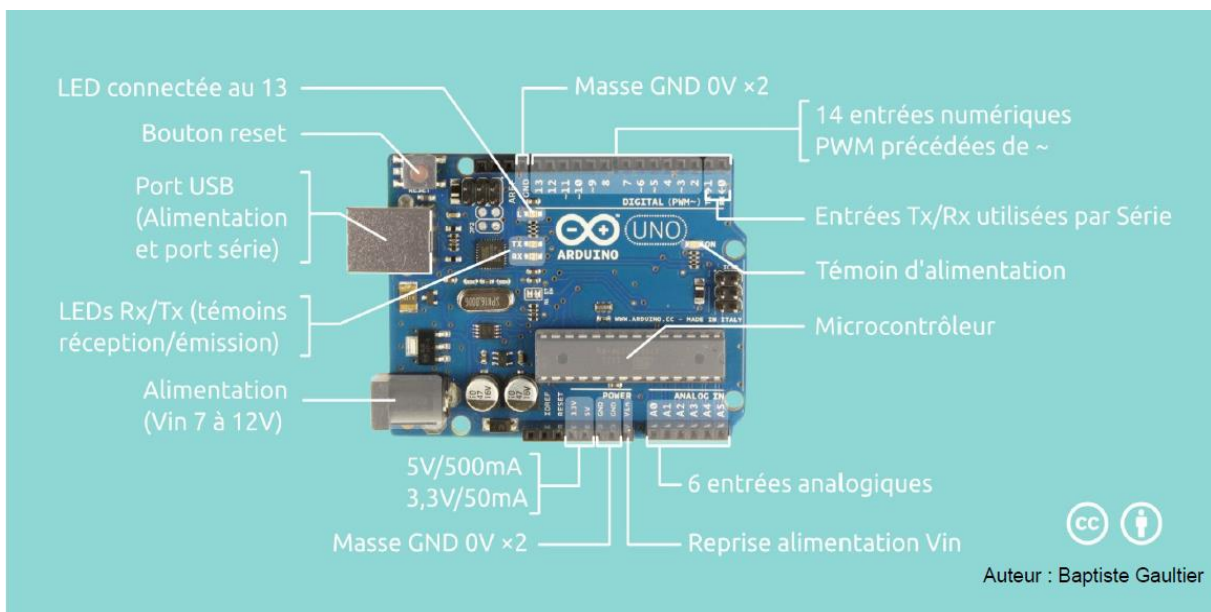


Figure 2.8 : constitution de la carte Arduino Uno [31]

Il existe aussi certain spécification de la carte arduino Uno (figure 2.11) qui sont :

Microcontrôleur	ATmega328
Frequences Horloge	16 Mhz
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM représenter par le symbole tilde)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (Attention 200mA max total entrée/sortie cumulées)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	500 mA max Sur port USB
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader (chargeur d'amorçage)
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Interface	USB (permet d'alimenté la carte et de transféré les programmes vers la carte)
Alimentation externe	Jack
Dimensions	6,86 cm x 5,3 cm

Tableau 2.2: spécification de la carte Arduino Uno [32]

II.6.1 Partie matérielle

Généralement pour réaliser n'importe quelle projet basé sur la carte Arduino il existe deux parties. La partie réalisation de circuit et la partie programmation.

II.6.1.1 Le Microcontrôleur ATmega328

Le microcontrôleur de la carte Arduino UNO est un ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Les principales caractéristiques sont :

- **FLASH** = mémoire programme de 32Ko
- **SRAM** = données (volatiles) 2Ko
- **EEPROM** = données (non volatiles) 1Ko
- **Digital I/O (entrées-sorties Tout Ou Rien)** =3 ports PortB, PortC, PortD (soit 23 broches en tout I/O)
- **Timers/Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits). Chaque timer peut etre utilise pour générer deux signaux PWM. (6 broches OCxA/OCxB)
- **Plusieurs broches multifonctions** : certaines broches peuvent avoir plusieurs fonctions différentes, choisies par programmation. Elles ont alors plusieurs noms sur le brochage (voir ci-avant). Par exemple, les broches PB1, PB2, PB3, PD3, PD5, PD6 peuvent servir de sortie PWM (PulseWidth Modulation), c'est-a-dire des sorties qui joueront le rôle de sorties analogiques. Elles correspondent aux broches des connecteurs 3,5,6,9,10 et 11. Cet autre rôle possible est lie aux timers et ces broches sont alors appelées OCxA ou OcxB dans la

documentation. Ce sont les memes broches, mais pour une autre fonction. Si vous regardez a nouveau le brochage, vous constaterez que toutes les broches sont multifonctions.

- **PWM** = 6 broches **OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB2), OC2A(PB3), OC2B(PD3)** .

Les broches du PORTC peuvent être converties par un convertisseur Analog toDigital.

- **Analog to Digital Converter** (resolution 10bits) = 6 entrees multiplexes ADC0(PC0) a ADC5(PC5).
 - **Gestion bus I2C** (TWI Two Wire Interface) = le bus est exploit via les branches SDA(PC5)/SCL(PC4).
 - **Port série (USART)** = émission/réception série via les broches **TXD(PD1)/RXD(PD0)**
 - **Comparteur Analogique** = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption
- Watchdog Timer programmable [33].**

II.6.1.1.1 Gestion d'interruptions (24 sources possibles (f interrupt vectors)) :

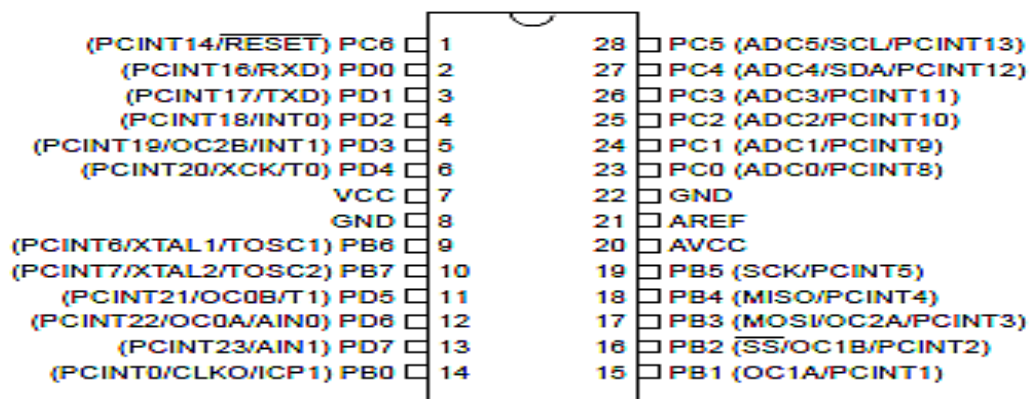


Figure 2.10: Architecture de ATMega 328 [34]

- Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3)
- Interruptions sur changement d'état des broches PCINT0 a PCINT23
- Interruptions liées aux Timer 0, 1 et 2 (plusieurs causes configurables)
- Interruption liée au comparateur analogique
- Interruption de fin de conversion ADC
- Interruptions du port série USART

– Interruption du bus TWI (I2C) [35]

II.6.2 Partie Programmation

L'environnement de programmation est open source et gratuit pour tous les différents système d'exploitation (Windows ,Mac , Os ,).

II.6.2.1 L'environnement de programmation

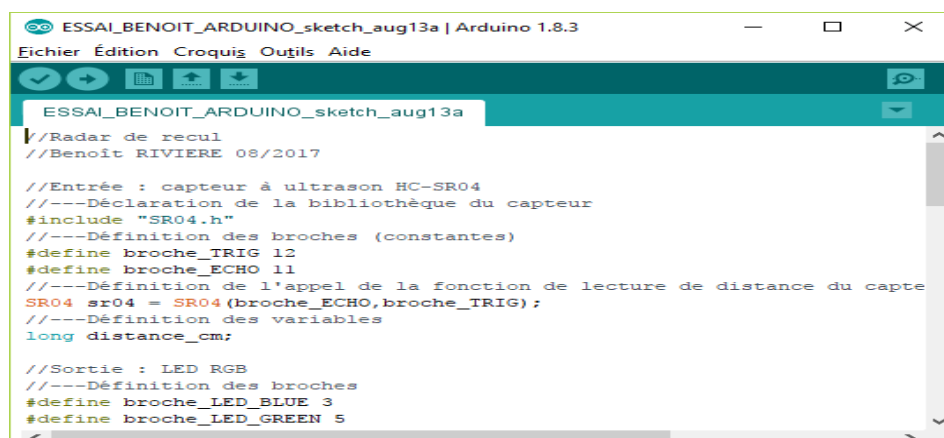
ARDUINO fournit l'environnement programmation (IDE) avec éditeur source, compilation et chargement en mémoire le microcontrôleur est réduit en clics sur les boutons IDE (très simple). La communication entre l'ordinateur et la carte est fait via USB, avec installation pilote personnalisé (fourni par ARDUINO)[36].

II.6.2.2 la structure de programmation de la carte Arduino

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

Il y a trois phases consécutives :

1. La définition des constantes et des variables .
2. La configuration des entrées et sorties : void setup() (figure 2.15 et 2.16).
3. La programmation des interactions et comportements : void loop()(figure 2.14 et 2.16).



```
ESSAI_BENOIT_ARDUINO_sketch_aug13a | Arduino 1.8.3
Eichier Édition Croquis Outils Aide

ESSAI_BENOIT_ARDUINO_sketch_aug13a
//Radar de recul
//Benoit RIVIERE 08/2017

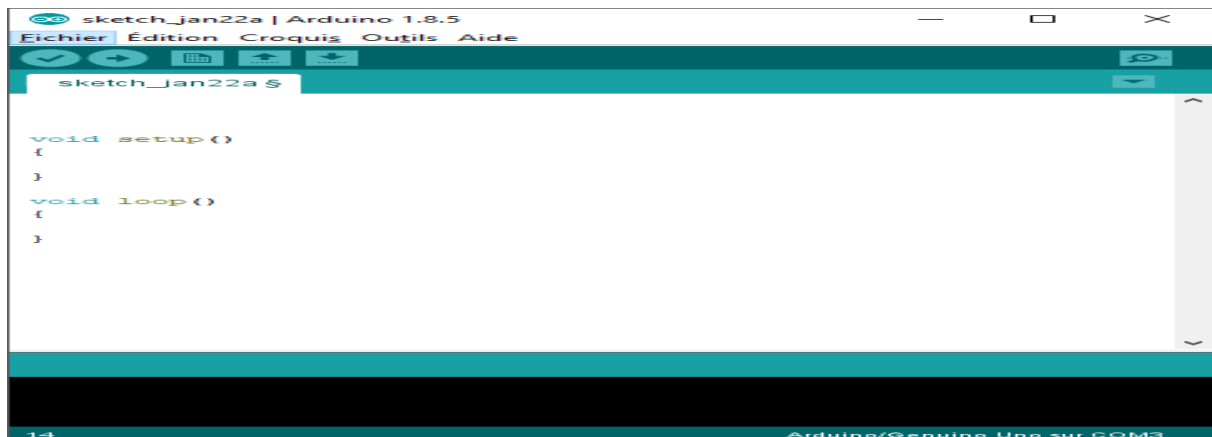
//Entrée : capteur à ultrason HC-SR04
//---Déclaration de la bibliothèque du capteur
#include "SR04.h"
//---Définition des broches (constantes)
#define broche_TRIG 12
#define broche_ECHO 11
//---Définition de l'appel de la fonction de lecture de distance du capte
SR04 sr04 = SR04(broche_ECHO,broche_TRIG);
//---Définition des variables
long distance_cm;

//Sortie : LED RGB
//---Définition des broches
#define broche_LED_BLUE 3
#define broche_LED_GREEN 5
```

Figure 2.11 : la Structure de base d'un programme d'Arduino [37].

II.6.2.2 .1 La fonction setup () :

Il s'agit de la **fonction d'initialisation** (figure 14 et 16). C'est la première fonction qui va être lancée au démarrage de l'Arduino. C'est donc ici que l'on va paramétrer nos différents objets, nos protocoles de communications, la configuration de nos entrées/sorties, etc...



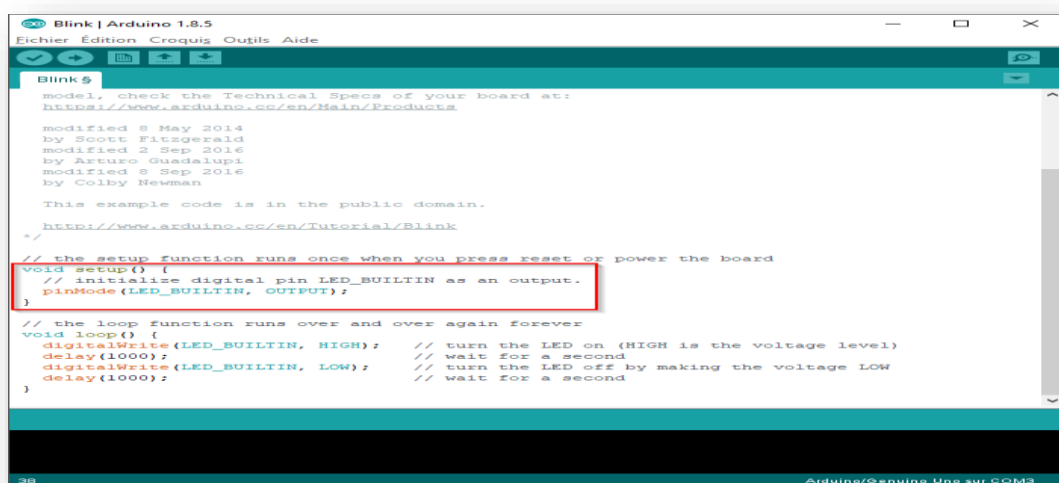
```
sketch_jan22a | Arduino 1.8.5
Fichier Edition Croquis Outils Aide
sketch_jan22a $
void setup()
{
}
void loop()
{
}
```

Figure 2.12: Exemple de la fonction setup () sous Arduino[38].

II.6.2.2 .2 La fonction loop()

La fonction loop() (figure 2.15 et 2.16).est la 2^{ème} fonction a être lancée sur votre Arduino, juste après la fonction setup(). Cette fonction a la particularité, comme son nom l'indique, d'être une boucle (loop en anglais). En effet, cette fonction reboucle sur elle-même. Lorsque la dernière ligne de la fonction loop() sera atteinte par votre programme .

En dehors de ces fonctions, on peut déclarer nos variables, d'autres fonctions, des bibliothèques additionnelles [38] .



```
Blink | Arduino 1.8.5
Fichier Edition Croquis Outils Aide
Blink $
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Figure 2.13 : Exemple de la fonction Loop () sous Arduino [40]

Un exemple de programme qui résume les trois phases de base de la structure de programmation de carte arduino.

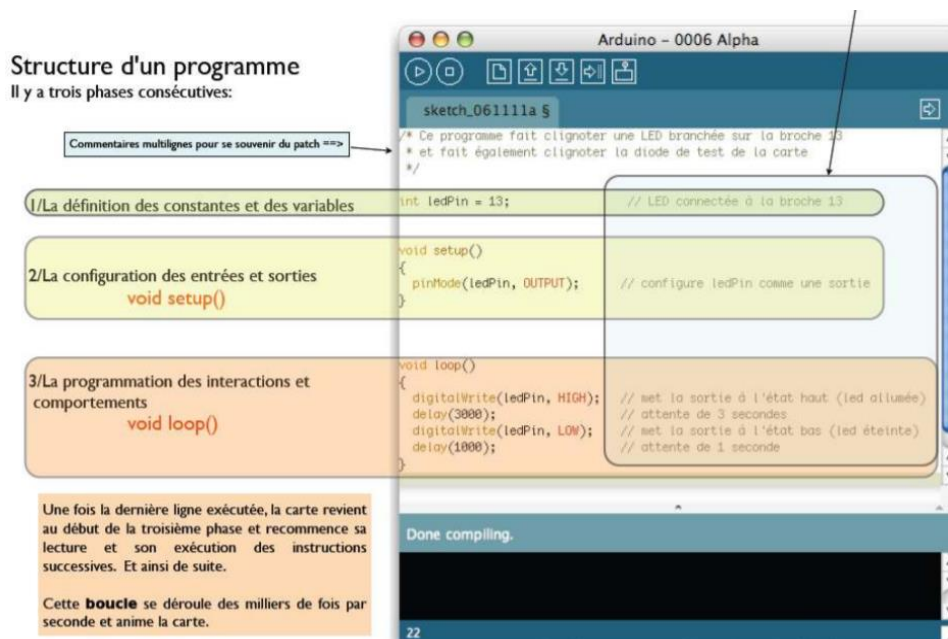


Figure 2.14 : Structure d'un programme.[39]

II-7 La conclusion :

Arduino est devenu indispensable dans divers domaines, comme la médecine, l'instrumentation, l'industrie, l'aéronautique ... Dans ce chapitre, nous avons souligné l'importance de cette carte, expliquant les différents principes qui forment sa base de construction, ainsi que sa programmation.

Chapitre 3 : Conception et réalisation d'un assistant de santé personnel

3-1-Introduction

Dans ce troisième chapitre, nous allons voir comment réaliser notre système (détection de chute). Tout d'abord, on va décrire notre application. Puis, on va essayer de donner les définitions de chaque composant et de chaque logiciel utilisé. Enfin, on va définir les étapes d'exécution et le principe de fonctionnement en détail.

3-2-Le choix de l'application

À nos jours on ne peut pas faire une séparation entre l'homme et la technologie, car elle a été incluse dans notre vie en jouant un rôle très important dans tous les domaines (études, sports, sciences... etc.). Parmi les problèmes dans le domaine de la santé est la chute de personnes âgées [figure.3.1]. Dans ce sens, nous allons proposer une application qui consiste à détecter le nombre de chutes dans la population âgée. Nous avons créé un nouveau système de détection de chute utilisant un accéléromètre (ADXL345). Notre système est basé sur un algorithme. En effet, de nombreux algorithmes ont été développés, mais jusqu'à présent, il est difficile de distinguer les vraies chutes de certaines activités (marchants, debout, assis). On est axé sur la conception d'un nouvel algorithme plus efficace pour distinguer les chutes de non chutes Figure.3.1.



Figure.3.1 : chute d'une personne âgée [40].

3-3-La description de l'application

Au cours des dernières années, différents types d'approches ont été proposées dans le domaine de la détection des chutes, qui peuvent être expliquées et classées en trois types: à base d'appareils portables, à base de capteurs d'ambiance et à base de vision. Tout d'abord, les appareils portables profitent généralement des capteurs intégrés pour détecter le mouvement et l'emplacement du corps, tels que l'accéléromètre, le magnétomètre et le gyroscope. Et le coût de l'approche basée sur les appareils portables est assez faible, ainsi que l'installation et le fonctionnement ne sont pas compliqués pour les personnes âgées.

Deuxièmement, l'approche basée sur l'ambiance utilise toujours des capteurs de pression pour détecter et suivre le corps. Cette solution est également rentable et facile à déployer. Cependant, la possibilité de détecter des objets autres que des corps humains pose un défi remarquable à la précision de détection de cette approche. Enfin, la solution basée sur la vision utilise pleinement les caméras déployées pour surveiller tous les objets à portée, y compris les corps humains. Il y a moins d'intrusion dans la vie quotidienne des gens que les deux approches ci-dessus, alors que l'espace d'observation est limité et qu'une surveillance omniprésente ne peut pas être réalisée.

Dans ce projet, nous avons élaboré un système de surveillance basé sur Arduino, Windows 10 et Microsoft Azure pour la détection de chute en temps réel. Les données brutes de l'accéléromètre tridimensionnel sont fournies par Arduino avec ADXL345. L'appareil Windows 10 utilise ces informations pour obtenir l'orientation du sujet grâce à des algorithmes efficaces de fusion

Les services de plate-forme applicative en nuage Microsoft Azure et l'application Mobile / PC sont également utilisés pour réaliser le traitement, l'analyse, le stockage et l'acquisition de données sans interruption à tout moment et en tout lieu, tant qu'ils ont accès à Internet. L'architecture du système est représentée sur la figure [figure.3.2] :

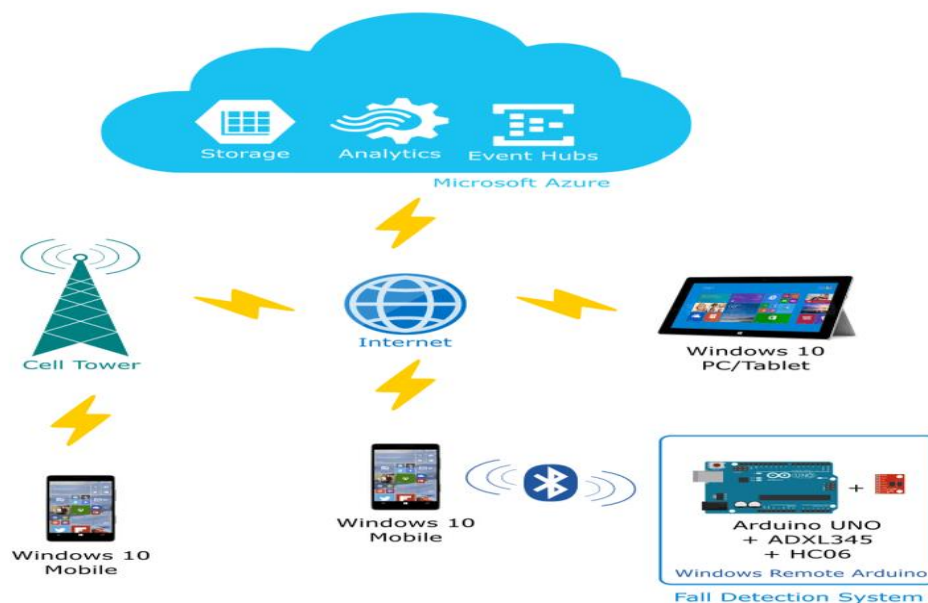


Figure.3.2 : Architecture du système [41]

III-4-Matériels et logiciels déployés

III-4-1-Partie matérielle

Nous avons utilisé trois composants, une carte Arduino et des câbles de liaison (mâle - mâle et mâle- femelle), chaque composant est défini ci-dessous :

III-4-1-1-le module Bluetooth HC-06

Le module Bluetooth HC-06 permet d'établir une liaison Bluetooth (liaison série) entre une carte Arduino et un autre équipement possédant une connexion Bluetooth (Smartphone, tablette, seconde carte Arduino, etc...). Le module HC-06 est un module "esclave" contrairement au module HC-05 qui est "maître". Un module "maître" peut demander à un autre élément Bluetooth de s'appairer avec lui alors qu'un module "esclave" ne peut recevoir que des demandes d'appairage.

Ces deux modules peuvent être configurés grâce à des commandes AT (ou commandes Hayes).

La tension d'alimentation de ces modules doit être comprise entre 3,3 et 5 V, mais la broche RX ne peut recevoir qu'une tension maximale de 3,3 V. Il faudra prévoir un pont diviseur de tension pour ramener la tension 5 V délivrée par la carte Arduino pour ne pas endommager la broche RX du module Bluetooth.

Le module HC-06 (comme les modules XBee) peut communiquer avec la carte Arduino via la liaison série (broches numériques N°0 et N°1) mais le moniteur série ne peut plus être utilisé. C'est pour cette raison qu'il est préférable d'utiliser la bibliothèque SoftwareSerial intégrée d'office dans l'IDE Arduino. Ses principales fonctions sont :

SoftwareSerial HC06 (broche_RX, broche_TX) où broche_RX et broche_TX représente les broches de la liaison série créée. Cette fonction permet de créer l'objet.

HC06.begin (baud) où baud représente la vitesse de transmission. Cette fonction permet d'initialiser l'objet. HC06.available () indique le nombre d'octets reçus. HC06.read () lit l'octet présent dans le buffer de la liaison série.

HC06.print () envoie tout ce qu'il y a entre les parenthèses sous forme d'octets donc en ASCII.

HC06.write (octet) envoie l'octet.

Dans l'univers Arduino, le module Bluetooth HC-06 est principalement utilisé dans des applications nécessitant une communication sans fil. [42] .le HC 06 est présentée dans la figure (figure.3.3) :



Figure.3.3: Module Bluetooth HC-06 [43]

III-4-1-2-Accéléromètre ADXL335

ADXL335 est un accéléromètre avec 3 axes. Fort apprécié, car il est efficace et simple à mettre en œuvre. Depuis qu'AdaFruit lui a ajouté un régulateur 3.3V, cet accéléromètre est devenu compatible avec Arduino ou tout autre microcontrôleur 5 volts. Il est parfait pour mesurer des accélérations statiques avec une grande résolution (comme un senseur de type tilt). Il convient également pour mesurer des accélérations modérées en cours de mouvement, des chocs ou des vibrations. Etant donné que notre planète dispose également d'une accélération gravitationnelle (de 1G), il est possible de Détecter l'orientation du breakout par rapport à la terre avec ce senseur. Un tel accéléromètre peut, par exemple, être utilisé pour détecter l'orientation d'un écran de téléphone. [44].



Figure.3.4 : Accéléromètre ADXL335

III-4-2-Partie logiciels

Dans cette partie on doit travailler sur le système Windows 10 et on va utiliser trois logiciels, définit ci suit.

III-4-2-1-Windows azure mobile

Microsoft Azure (Windows Azure jusqu'en 2014) est la plate-forme applicative en nuage de Microsoft. Son nom évoque le « cloud computing », ou informatique en nuage (l'externalisation des ressources informatiques d'une entreprise vers des datacenters distants). Il s'agit d'une offre d'hébergement (applications et données) et de services (workflow, stockage et synchronisation des données, bus de messages, contacts...). [45]

III-4-2-2-Microsoft Visual studios

Microsoft Visual Studios est une suite de logiciels de développement pour Windows et mac OS conçue par Microsoft. La dernière version s'appelle Visual Studio 2019.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils.

Ce logiciel facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Développeur [46].

III-4-2-3-L'IDE Arduino

Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel Arduino. Il permet : d'éditer un programme : des croquis (sketch en Anglais), de compiler ce programme dans le langage « machine » de l'Arduino, de téléverser le programme dans la mémoire de l'Arduino, de communiquer avec la carte Arduino grâce au terminal (voir annexes) [47]

III-5-Réalisation

III-5-1-Principe de détection des chutes avec reconnaissance de posture

Dans ce projet, l'algorithme de détection de chute est conçu selon l'algorithme de Bharadwaj Sreenivasan [48]. L'algorithme comprend deux modules, la reconnaissance de posture et la détection de chute. Tout d'abord, nous lisons les valeurs x, y et z de l'accéléromètre. Ensuite, nous calculons la norme L2 des x, y et z. Il est utilisé par le module de reconnaissance de posture et de détection de chute.

Dans le module de reconnaissance des postures, les postures des utilisateurs sont classées en trois postures de base: assise, debout et marchant. Les valeurs de "y" sont appliquées un seuil pour connaître l'orientation. En utilisant le taux de variation de la norme L2 par rapport à l'accélération moyenne due à la gravité (9,8) nous classons les données en marchant ou simplement en transition entre les états. Le module de détection de chute recherche un motif particulier dans le signal. La [Fig.3.5] représente un schéma typique de la norme L2 lors d'un événement de chute.

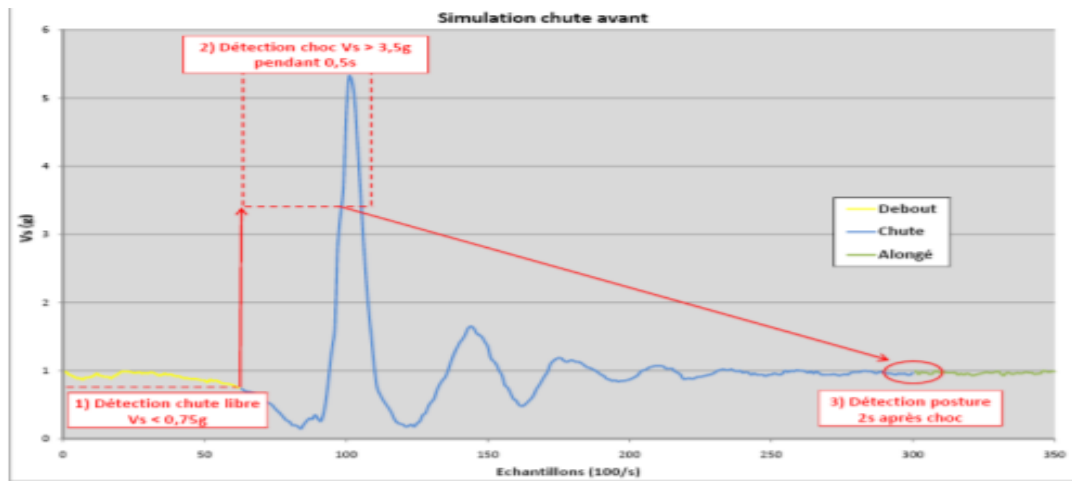


Figure.3.5 : Schéma de chute [49]

Si la différence entre les minima et les maxima consécutifs de l'accélération est supérieure à 2 g (le g : étant l'initiale de « gravité » : est une unité d'accélération correspondant approximativement à l'accélération de la pesanteur à la surface de la Terre. Sa valeur conventionnelle est de $9,80665 \text{ m s}^{-2}$) [50], la sortie est décidée comme une chute. La décision finale sur l'événement d'une chute est basée sur la sortie du module de reconnaissance de posture et de décision de chute. Lorsqu'une chute est détectée, la décision du module de reconnaissance de posture nous indique s'il s'agit d'une fausse alarme. Si l'État marche toujours, la décision d'événement de chute peut être annulée. Le tableau.31 résume la décision finale.

Tableau 3.1: Décision d'automne

La détection des chutes	Reconnaissance de la posture	Décision sur les survenues d'une chute
Chute	En marchant	Faux
Chute	Debout	Faux
Chute	Assit	vrai
Chute	Etat nul	vrai

III-5-2- Conditions préalables

Dans ce projet, pour le périphérique Arduino, la communication Bluetooth est réalisée par HC-06, qui est un petit module compact bon marché. Pour l'appareil Windows 10, si on utilise Surface Pro et Lumia 1520, on n'aura pas besoin du module Bluetooth.

III-5-3- L'instruction du projet

Notre travail se résume dans six étapes :

III-5-3-1-Etape 1

Connectez le module accéléromètre ADXL345 à l'Arduino: L'ADXL345 est un petit accéléromètre MEMS 3. Axes, mince et de faible puissance avec une mesure haute résolution (13 bits) jusqu'à + -16 g. Les données de sortie numériques sont formatées en tant que complément 16 bits deux bits et sont accessibles via une interface numérique SPI (3 ou 4 fils) ou I2C. L'ADXL345 prend en charge SPI et I2C, nous allons simplement parler I2C ici dans ce projet. I2C est une connexion série à 2 fils, nous devons donc connecter les lignes SDA (données) et SCL (horloge) à Arduino UNO pour la communication. Sur Arduino UNO, SDA est sur la broche analogique 4 et SCL sur la broche analogique 5. Assurez-vous que les broches VCC et CS du capteur sont connectées à 3,3 V. Nous connectons CS à 3,3 V pour indiquer au capteur que nous l'utiliserons comme appareil I2C et non comme appareil SPI [figure.3.6].

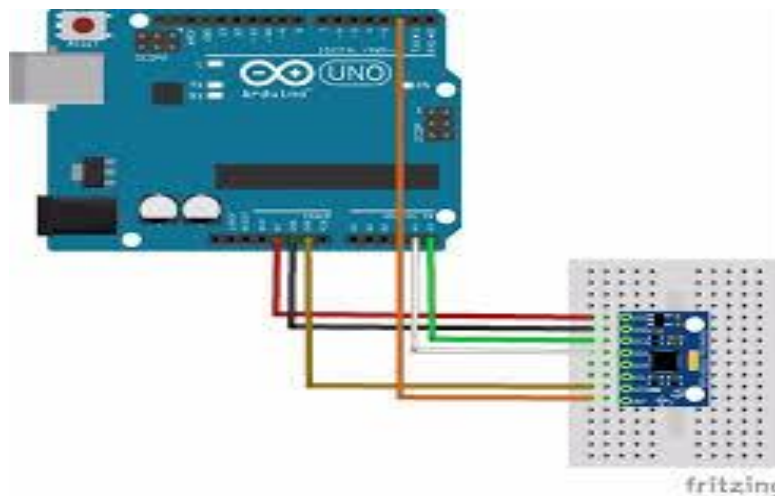


Figure.3.6 : Croquis de la connexion adxl345 [51]

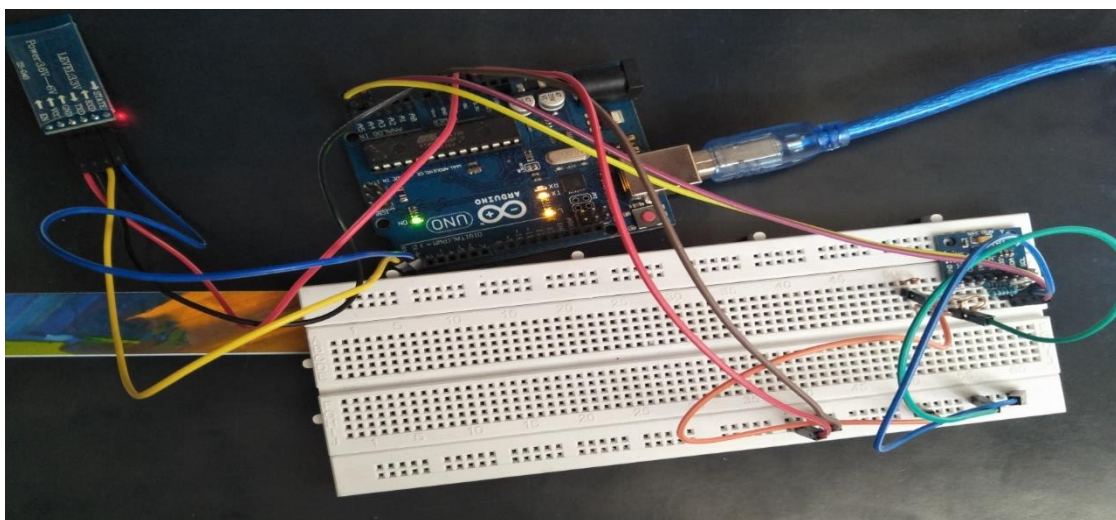
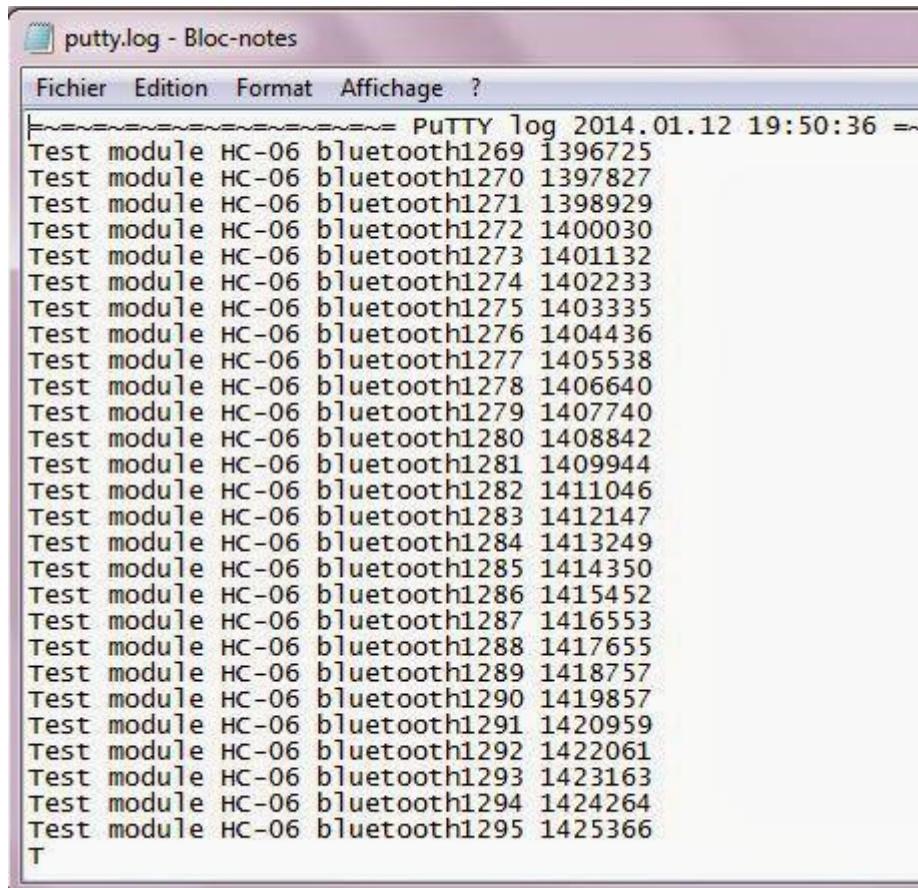


Figure 3.6.a : notre propre branchement

III-5-3-2-Etape 2

Connexion du module Bluetooth. Dans ce projet, nous utilisons HC-06 comme module Bluetooth. De plus, le HC-05 possède le firmware «complet», de nombreuses commandes AT et peut être à la fois un module maître et un module esclave. Le firmware HC-06, en revanche, ne peut être qu'un appareil esclave, avec des commandes AT très limitées. Le tableau ci-dessous (tableau.32) montre toutes les commandes du firmware HC-06 avec la réponse:



```
putty.log - Bloc-notes
Fichier Edition Format Affichage ?
----- PUTTY log 2014.01.12 19:50:36 -----
Test module HC-06 bluetooth1269 1396725
Test module HC-06 bluetooth1270 1397827
Test module HC-06 bluetooth1271 1398929
Test module HC-06 bluetooth1272 1400030
Test module HC-06 bluetooth1273 1401132
Test module HC-06 bluetooth1274 1402233
Test module HC-06 bluetooth1275 1403335
Test module HC-06 bluetooth1276 1404436
Test module HC-06 bluetooth1277 1405538
Test module HC-06 bluetooth1278 1406640
Test module HC-06 bluetooth1279 1407740
Test module HC-06 bluetooth1280 1408842
Test module HC-06 bluetooth1281 1409944
Test module HC-06 bluetooth1282 1411046
Test module HC-06 bluetooth1283 1412147
Test module HC-06 bluetooth1284 1413249
Test module HC-06 bluetooth1285 1414350
Test module HC-06 bluetooth1286 1415452
Test module HC-06 bluetooth1287 1416553
Test module HC-06 bluetooth1288 1417655
Test module HC-06 bluetooth1289 1418757
Test module HC-06 bluetooth1290 1419857
Test module HC-06 bluetooth1291 1420959
Test module HC-06 bluetooth1292 1422061
Test module HC-06 bluetooth1293 1423163
Test module HC-06 bluetooth1294 1424264
Test module HC-06 bluetooth1295 1425366
T
```

Figure 3.6.b: commandes AT du micro logiciel hc-06 [50]

Tout d'abord, on branche le module Bluetooth sur la maquette pour faciliter les connexions. Pour alimenter le module, on exécute un cavalier entre le rail d'alimentation 5 V de l'Arduino et la broche Vcc du module. Et on exécute un cavalier mâle-mâle de la broche GND du module à l'une des broches GND de l'Arduino. Pour établir les communications série, on a connecté un cavalier mâle-mâle de la broche TX du module Bluetooth à la broche RX de l'Arduino (broche numérique 0) et un autre de la broche RX du module à la broche TX de l'Arduino (broche numérique 1). Remarque: on ne peut pas télécharger d'esquisses de l'IDE. Déconnectez-les lorsque vous devez télécharger une esquisse et reconnectez-les une fois que l'esquisse a été téléchargée avec succès. Après la connexion du Bluetooth avec l'accéléromètre, on aura l'affichage suivant (figure.3.7) :

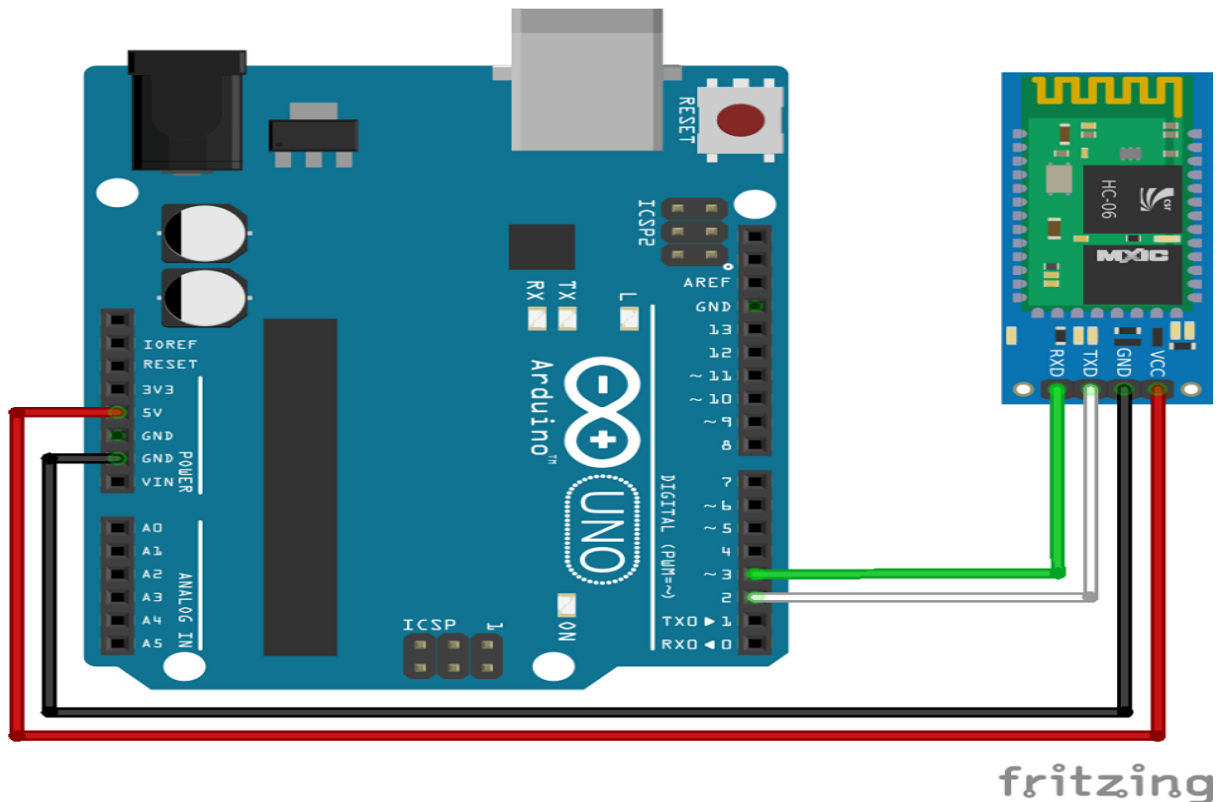


Figure.3.7: Croquis de la connexion Bluetooth HC-06 [52]

III-5-3-3-Etape 3

Le module Bluetooth exécute le protocole SPP (Serial Protocol over Bluetooth). Ainsi, tout appareil prenant en charge SPP peut s'y connecter. Sur un appareil Windows 10, cela ressemble à un port COM virtuel. Avant de vous connecter, assurez-vous que le module est sous tension et prêt à être couplé. La LED rouge sur le module indique l'état: clignotant signifie «prêt à coupler», tandis que fixe allumé signifie «couplé». Sur Windows 10 pour PC, accédez simplement à Paramètres-> Périphériques-> Bluetooth, ouvrez Bluetooth, puis le module «HC-06» s'affichera à l'écran. Appuyez sur pour coupler l'appareil et entrez le code d'appariement de l'appareil. N'oubliez pas que le code de couplage par défaut est 1234 et que la vitesse du port série par défaut est de 9600 bauds.

Pour Windows 10 pour mobile, accédez à Paramètres-> Appareils-> Bluetooth et suivez les mêmes étapes décrites dans la section ci-dessus. Et la page de l'appareil jumelé est présentée comme suit [figure.3.8] :

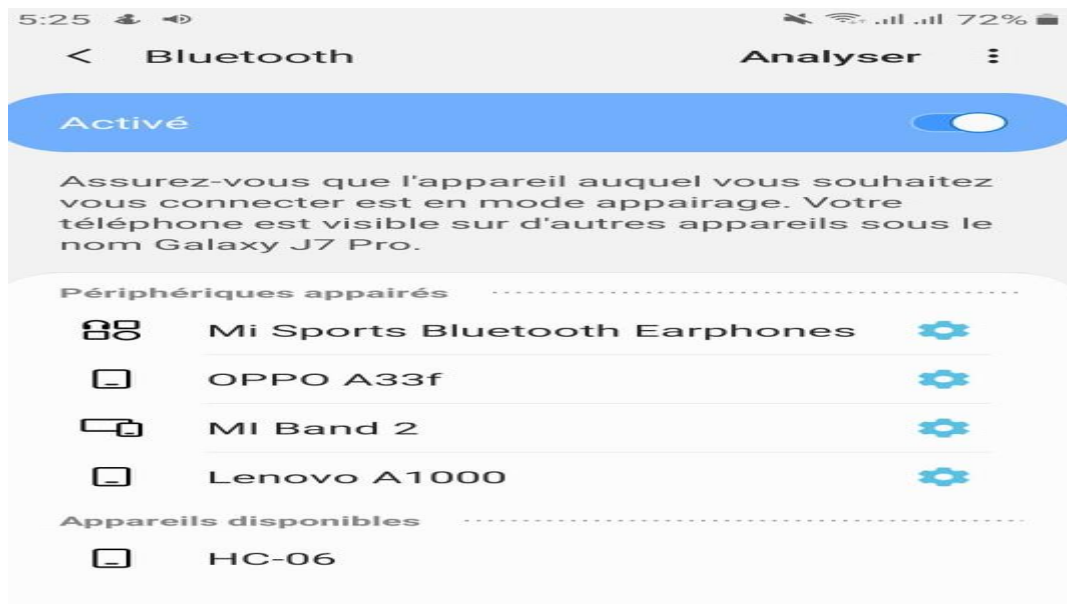


Figure.3.8: Couplage Bluetooth pour Windows 10 mobile

III-5-3-4-Etape 4

Créer le compte de stockage azure :

Dans ce projet, nous aurons besoin d'Azure Storage Table pour stocker les données de l'appareil Windows 10 IoT Core. Donc, nous pouvons le créer comme suit.

- (1) Dans le portail de gestion Azure (<https://manage.windowsazure.com>), cliquez sur l'icône "STOCKAGE" à gauche pour afficher vos comptes de stockage existants (le cas échéant), puis cliquez sur le bouton "+ NOUVEAU" dans le coin inférieur gauche.
- (2) Dans le panneau "NOUVEAU", sélectionnez "SERVICES DE DONNÉES" | "STOCKAGE" | "CRÉATION RAPIDE". Remplissez les champs «URL, groupe d'emplacement / affinité et réplication», puis cliquez sur le bouton «CRÉER UN COMPTE DE STOCKAGE».
- (3) Attendez que l'état du nouveau compte de stockage s'affiche comme "En ligne".
- (4) Sélectionnez le compte de stockage nouvellement créé, puis cliquez sur GÉRER LES CLÉS D'ACCÈS en bas de la page. Copiez le nom du compte de stockage et l'une des clés d'accès.

III-5-3-5-Etape 5

Nous utilisons Azure Storage Explorer (C'est un outil open source tiers disponible sur codeplex: <https://azurestorageexplorer.codeplex.com>) pour créer la table de stockage Azure.

- (1) Allez pour ajouter un compte et remplissez le nom du compte et la clé d'accès. Une fois le compte établi, vous pouvez accéder au stockage qui affiche les conteneurs d'objets blob, les files d'attente et les tables.

- (2) (2) Sélectionnez le tableau et cliquez sur le nouveau bouton et créez un nouveau tableau nommé «Tableau d'accéléromètre» comme suit (figure.3.9)

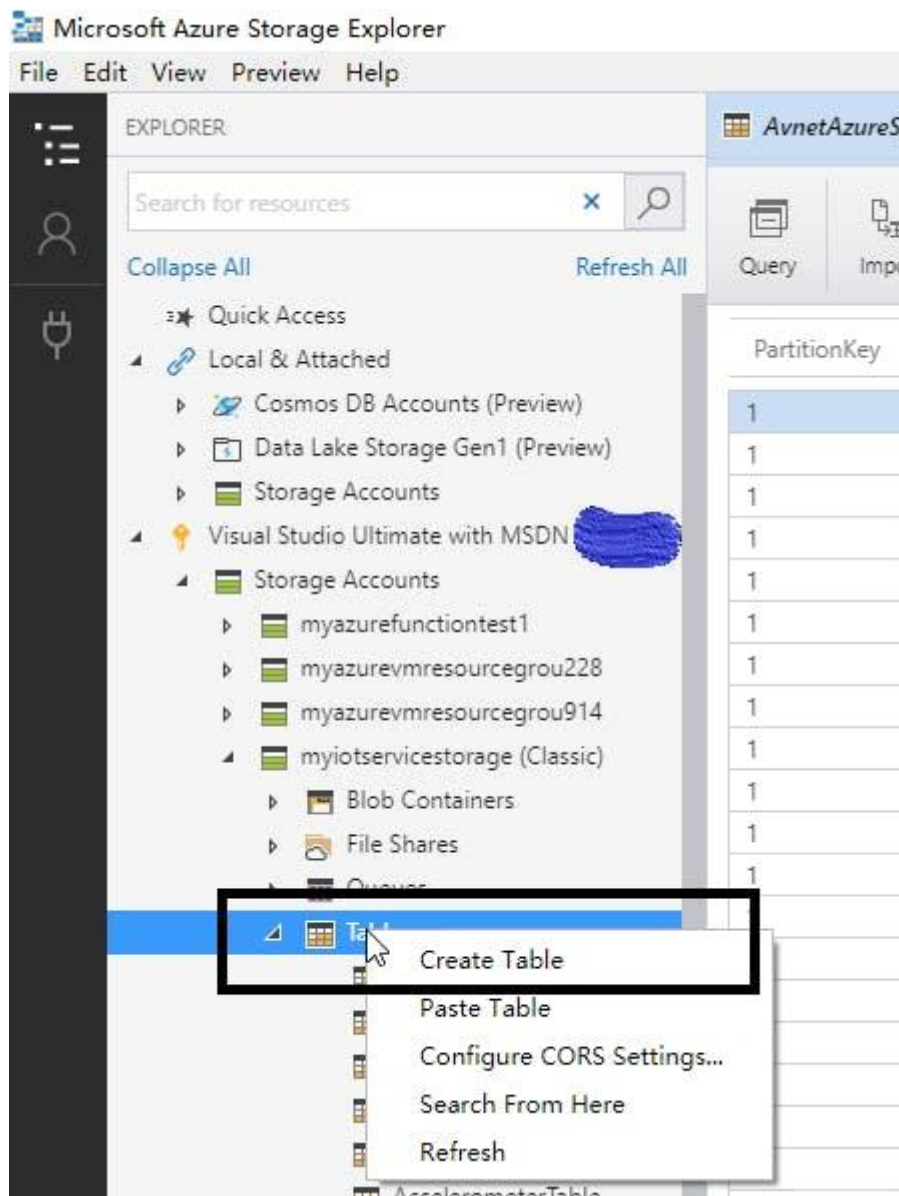


Figure.3.9: Création d'une table de stockage azure [53]

III-5-3-6-Etape 6

On Télécharge et déployez le code : Tout d'abord, maintenant que nous avons tout câblé pour Arduino, on charge le dans l'IDE Arduino sur notre PC. On a débranché les fils de connexion TX / RX du module HC-06. On a reconnecté les fils de cavalier TX / RX aux broches numérique 0 et numérique 1 de l'Arduino. On a Branché l'alimentation et l'application fonctionnera sur Arduino Uno Deuxièmement, on a téléchargé le projet «GenericBluetoothSerialUWApp» depuis github.

On Utilise Visual Studio Community 2019 avec la mise à jour pour charger «GenericBluetoothSerialUWApp».

On exécute l'application UWP (Windows Universal Platform) notre sur PC, si pour ça choisisse x86 ou x64 avec débogage de périphérique local. Si vous souhaitez que cette application UWP s'exécute sur Windows 10 Mobile, choisissez ARM avec le débogage de l'appareil. Ensuite, on ouvre MainPage.xaml.cs dans l'Explorateur de solutions et recherchez le «dataTransferTick». On ajoute le nom de notre compte de stockage et la clé d'accès enregistrée dans la section 4. voir la figure (figure.3.10)

```

1 reference
private async void dataTransferTick(ThreadPoolTimer timer)
{
    try
    {
        CloudStorageAccount storageAccount = CloudStorageAccount.Parse("DefaultEndpointsProtocol=https;AccountName=myiotservei

        // Create the table client.
        CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

        // Create the CloudTable object that represents the " AccelerometerTable " table.
        CloudTable table = tableClient.GetTableReference("AccelerometerTable");
        await table.CreateIfNotExistsAsync();
    }
}

```

Figure.3.10: Ajout des informations de compte

Après avoir couplé le HC-06 avec notre appareil Windows 10, on appuyant sur F5 à partir de Visual Studio pour lancer l'application Universel Windows Platform. Regardez la figure suivant [figure.3.11]

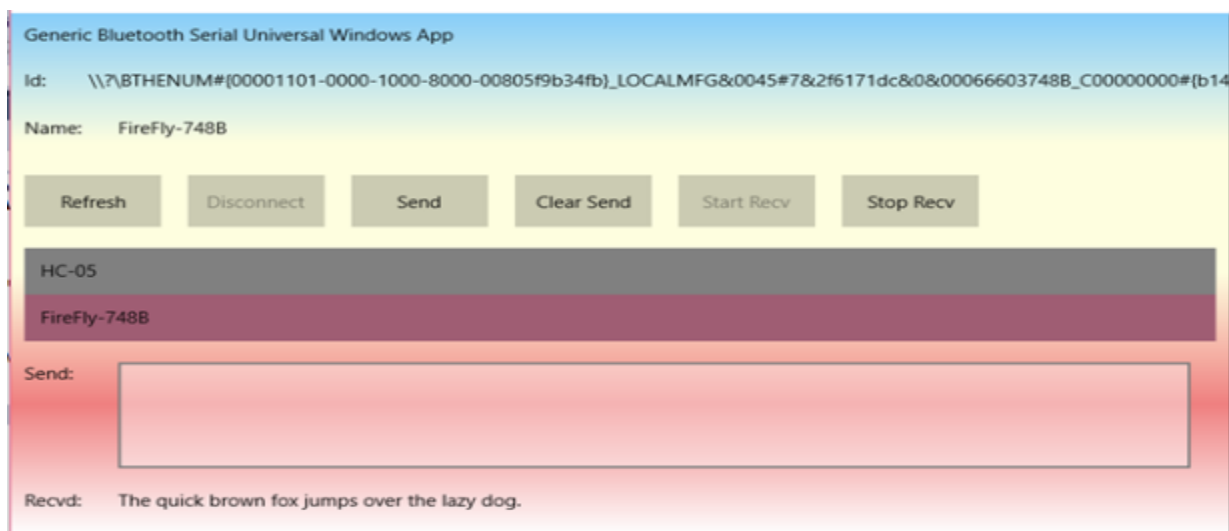


Figure.3.11: Ecran de démarrage de l'application [54]

On appuie deux fois sur le symbole «HC-06», l'affichage de l'identifiant et du nom ce trouve tout en haut de l'application .

Bientôt, l'état de la LED sur HC-06 passera de clignotant à stable. Et le bouton «Start Rev» est activé dès que le canal de communication du module Bluetooth est établi.

On clique sur Démarrer l'enregistrement pour activer la réception de données. Ensuite, vous verrez le bouton Arrêter l'enregistrement, Traiter les données et Télécharger les données sont activés.

On clique simplement sur le bouton Traiter les données pour exécuter l'algorithme de détection de chute. Vous verrez les données sur Recvd TextBlock et l'état de la personne qui porte l'appareil.

Les données seront affichées dans les fenêtres de débogage de Visual Studio. Ensuite, on clique sur le bouton pour télécharger les données vers Azure. Les données seront transmises toutes les 3 secondes. Bientôt, les données sont affichées dans AccelerometerTable :

Troisièmement, on a téléchargé le «FallDetectionClient» depuis github. On utilise Visual Studio Community 2019 avec Update 1 pour charger ce projet. Cette application est construite avec Windows Universal Template, vous pouvez donc l'exécuter sur Windows 10 PC / Mobile. Ensuite, on ouvre StorageSensor.cs dans l'Explorateur de solutions et on recherche «_accountName» et «_key».

On ajoute le nom de notre compte de stockage et la clé d'accès enregistrée dans la section 4.

Maintenant on appuie sur F5 à partir de Visual Studio: l'application se déploiera et Démarrera, et vous devriez le voir sur la sortie de l'appareil. .

Finalement dans le cas où une personne tombe, l'appareil Windows 10 mobile reçoit des informations (les trois dimensions de la personne X, Y, Z) de l'accéléromètre, grâce à la fusion de ces données et un algorithme, la chute sera détectée. Cette détection est envoyée à Windows 10 pour une analyse récapitulatif dans un tableau (tableau 31) des trois positions. Si la chute est détectée, il y aurait une alerte et un envoi d'un message vers un appareil Windows 10 mobile (réception). Le message reçu inclus les trois dimensions X,Y,Z ,et la situation de la personne.(figure.316).**III-6-Conclusion**

Dans ce chapitre nous avons présenté l'application de détection de chute en détail. Nous avons présenté la partie matérielle puis la partie logicielle de l'application, et aussi les différentes étapes de réalisation. Enfin, nous avons parlé des résultats obtenus avec cette application.

Conclusion Générale

Pour le confort de l'utilisateur il existe plusieurs solutions techniques : gestion d'énergie, optimisation de l'éclairage et du chauffage..., de la **sécurité** (alarmes, vidéo surveillance...) et de **communication** (commandes à distance, signaux visuels ou sonores...) ces technologies sont appelé également « la domotique » on peut les trouver dans les maisons, les hôtels, les entreprises, les commerces, les lieux publics et aussi dans le domaine médical.

Il existe une multitude de possibilités d'utilisation avec le système l'Arduino du fait qui il est utilisé avec des capteurs et des actionneurs, mais également il existe plusieurs façons de communiquer avec ce système, c'est là qu'intervient plusieurs applications chacune définit un protocole standardisé ce qui permet une utilisation parfaite de ce système dans le domaine de la domotique.

Dans ce projet, un système de surveillance basé sur Arduino, Windows 10 et Microsoft Azure pour la détection de chutes de personnes agès en temps réel a été proposé.

La première partie du travail consiste à relier la carte arduino à un accéléromètre MEMS 3 axes : ADXL345 pour détecter l'activité ou l'absence de mouvement. Ensuite nous avons couplé entre le module Bluetooth HC-05 et un ordinateur fonctionnant avec Windows 10.

La deuxième partie consiste a développé une application virtuelle dans la Microsoft Visual studio à travers un algorithme qui nous permet de savoir s'il existe ou non une chute. Cette application permet aussi de communiquer avec un assistant d'usage pour cette application pour lui transmettre les données.

Les résultats obtenus avec cette application permettent d'ouvrir la porte à d'autres perspectives dans le domaine médical, parmi lesquelles on peut citer :

1. Réalisation un système détection de chute basée sur la vision.
2. Réalisation un système prévention de chute.
3. Réaliser un système de vidéo surveillance pour les seniors.
4. Réalisation d'un système de détection des vertiges qui précèdent les chutes seniors.
5. Réalisation une assistance complète des personnes âgées à distance.

References Bibliographiques

Chapitre 1

- [1]R. Torbensen, K.M. Hansen, T.S. Hjorth, My Home Is My Bazaar-A Taxonomy And Classification Of Current Wireless Home Network Protocols, In: 2011 Second Eastern European Regional Conference On The Engineering Of Computer Based Systems, Ieee, 2011, Pp. 35-43.
- [1.1] https://www.maison-et-domotique.com/wp-content/uploads/2012/04/2012-04-24_123618.png
- [2]<http://www.habitatnews.fr/2019/01/10/comment-domotique-aider-faire-economies-energie/> [3] D. Russo, Brief Introduction To Domotics, Robotics, And To A Their Possible Evolution, In, 2018.
- [4]T.S. Hjorth, R. Torbensen, Trusted Domain: A Security Platform For Home Automation, Computers & Security, 31 (2012) 940-955.
- [5]S.K. Datta, C. Bonnet, A. Gyrard, R.P.F. Da Costa, K. Boudaoud, Applying Internet Of Things For Personalized Healthcare In Smart Homes, In: 2015 24th Wireless And Optical Communication Conference (Wocc), Ieee, 2015, Pp. 164-169.
- [6] J. Branger, Z. Pang, From Automated Home To Sustainable, Healthy And Manufacturing Home: A New Story Enabled By The Internet-Of-Things And Industry 4.0, Journal Of Management Analytics, 2 (2015) 314-332.
- [7]C.D. Nugent, D.D. Finlay, P. Fiorini, Y. Tsumaki, E. Prassler, Editorial Home Automation As A Means Of Independent Living, Ieee Transactions On Automation Science And Engineering, 5 (2008) 1-9.
- [8]S. Nourizadeh, C. Deroussent, Y.-Q. Song, J.-P. Thomesse, Medical And Home Automation Sensor Networks For Senior Citizens Telehomecare, In: 2009 Ieee International Conference On Communications Workshops, Ieee, 2009, Pp. 1-5.
- [9]D.H. Stefanov, Z. Bien, W.-C. Bang, The Smart House For Older Persons And Persons With Physical Disabilities: Structure, Technology Arrangements, And Perspectives, Ieee Transactions On Neural Systems And Rehabilitation Engineering, 12 (2004) 228-250.
- [10]M. Bal, W. Shen, Q. Hao, H. Xue, Collaborative Smart Home Technologies For Senior Independent Living: A Review, In: Proceedings Of The 2011 15th International Conference On Computer Supported Cooperative Work In Design (Cscwd), Ieee, 2011, Pp. 481-488.
- [11]T.H.C. Tran, D. Nguyen Van Nuoi, H. Baiz, G. Baglin, J.J. Leduc, H. Bulkaen, Déficit Visuel Chez Les Sujets Agés Chuteurs, Journal Français D'ophtalmologie, 34 (2011) 723-728.
- [12]J. Kulmala, A. Viljanen, S. Sipilä, S. Pajala, O. Pärssinen, M. Kauppinen, M. Koskenvuo, J. Kaprio, T. Rantanen, Poor Vision Accompanied With Other Sensory Impairments As A Predictor Of Falls In Older Women, Age And Ageing, 38 (2009) 162-167.

[13]A.L. Coleman, K. Stone, S.K. Ewing, M. Nevitt, S. Cummings, J.A. Cauley, K.E. Ensrud, E.L. Harris, M.C. Hochberg, C.M. Mangione, Higher Risk Of Multiple Falls Among Elderly Women Who Lose Visual Acuity, Ophthalmology, 111 (2004) 857-862.

[15]<https://www.tedorthopedics.com/>

[15.1]<https://lumeen.com/>

Chapitre 2

[17]Cour :Carte Arduino Pour L3 Gbe, Université L'université Abou Bekr Belkaid Tlemcen,

Auteur :M.Belgacem Nourdinne

[18]Dossier La Carte Arduino Lycee Condorcet Bac Sen Eie Montreuil 93 Younsi A. Christian Tavernier, Arduino Maîtriser Sa Programmation Et Ses Cartes D'interface (Shields), Paris, 2011, Dunod (Page 2)

[19]https://fr.wikipedia.org/wiki/Atmel_Avr

[20]Dossier La Carte Arduino Lycee Condorcet Bac Sen Eie Montreuil 93 Younsi A. Christian Tavernier, Arduino Maîtriser Sa Programmation Et Ses Cartes D'interface (Shields), Paris, 2011, Dunod (Page 2)

[21]<https://twitter.com/Hashtag/Portentah7>

[22]<https://systemes-embarques.fr/wp/archives/arduino-mkr-vidor-4000-presentation-et-mise-en-route/>

[23]<https://elmwoodelectronics.ca/products/arduino-uno-wifi-rev2>

[24]https://ar.m.wikipedia.org/wiki/%D9%85%D9%84%D9%81:Arduino_Diecimila.Jpg

[25]<https://www.amazon.fr/sparkfun-carte-lilypad-arduino-simplifi%C3%A9e/dp/B008as3wqs>

[26]<https://www.arduino-france.com/review/arduino-mega-utilisation-et-fonctionnement/>

[27]<https://refrobot.wordpress.com/2014/01/24/arduino-esplora-arduino/>

[28]https://www.positronlibre.com/electronique/arduino/arduino.php?fbclid=Iwar3mojq623qzbzp4daty5xzfsytg1ihey6ioqtaaw1oem7dl3_H42i%20figure%204%20jsca%208%20%20https://www.positronlibre.com/electronique/arduino/arduino.php?fbclid=Iwar3mojq623qzbzp4daty5xzfsytg1ihey6ioqtaaw1oem7dl3_H42i

[29]Microcontrôleurs Ei3 Agi 2016-2017 Istia ,B. Cottenceau B31,Istia,Bertrand.Cottenceau@Univ-Angers.Fr ,Page 3

[30]<https://www.gotronic.fr/art-carte-arduino-uno-12420.html>

[31]Auteur : Guy Sinnig : Guy@Sinnig.Fr ,Diaporama Présentation ,Découverte De La Carte Arduino Page 10 .

[32][Http://Technolab.Fr/Wp-Content/Uploads/2017/10/Description-De-La-Carte-Arduino-Uno.Pdf](http://Technolab.Fr/Wp-Content/Uploads/2017/10/Description-De-La-Carte-Arduino-Uno.Pdf)

[33] ,[34],[35] Microcontrôleurs Ei3 Agi 2016-2017 ,Istia ,Microcontrôleurs 1/29

Carte Arduino Uno ,Microcontrôleur Atmega328 ,B. Cottenceau ,B. Cottenceau B311 Istia, Bertrand.Cottenceau@Univ-Angers.Fr ,Page 4

[36][Https://Bentek.Fr/4-Structure-Programme-Arduino/](https://Bentek.Fr/4-Structure-Programme-Arduino/)

[37] [38][Https://Bentek.Fr/4-Structure-Programme-Arduino/](https://Bentek.Fr/4-Structure-Programme-Arduino/)

[39][Https://Wiki.Mdl29.Net/Lib/Exe/Fetch.Php?Media=Elec:Arduino_Dossier_Ressource.Pdf](https://Wiki.Mdl29.Net/Lib/Exe/Fetch.Php?Media=Elec:Arduino_Dossier_Ressource.Pdf)

[40][Https://Www.Pinterest.Fr/Pin/376543218839841125/](https://Www.Pinterest.Fr/Pin/376543218839841125/)

[41][Https://Www.Virtualizationhowto.Com/2019/11/What-Is-Azure-Arc-Management-And-Governance/](https://Www.Virtualizationhowto.Com/2019/11/What-Is-Azure-Arc-Management-And-Governance/)

[42][Https://Pecquery.Wixsite.Com/Arduino-Passion/Copie-De-Le-Detecteur-A-Ultrasons-H-1](https://Pecquery.Wixsite.Com/Arduino-Passion/Copie-De-Le-Detecteur-A-Ultrasons-H-1)

[43][Https://Www.Amazon.Fr/Bomcomi-Broches-Bluetooth-Transeiver-Remplacement/Dp/B07rz5zthk](https://Www.Amazon.Fr/Bomcomi-Broches-Bluetooth-Transeiver-Remplacement/Dp/B07rz5zthk)

[44][Https://Arduino103.Blogspot.Com/2013/06/Tutoriel-Sur-Laccelerometre-Adxl335-3.Html?M=1](https://Arduino103.Blogspot.Com/2013/06/Tutoriel-Sur-Laccelerometre-Adxl335-3.Html?M=1)

[45][Https://Fr.M.Wikipedia.Org/Wiki/Microsoft_Azure](https://Fr.M.Wikipedia.Org/Wiki/Microsoft_Azure)

[46][Https://Fr.M.Wikipedia.Org/Wiki/Microsoft_Visual_Studio](https://Fr.M.Wikipedia.Org/Wiki/Microsoft_Visual_Studio)

[47] [Https://Arduino.Blaisepascal.Fr/Presentation/Logiciel-Ide-Arduino/](https://Arduino.Blaisepascal.Fr/Presentation/Logiciel-Ide-Arduino/)

[48][Https://En.M.Wikipedia.Org/Wiki/M._B._Sreenivasan](https://En.M.Wikipedia.Org/Wiki/M._B._Sreenivasan)

[49]Article :Mise En Œuvre D'une Plateforme De Suivi De L'actimétrie Associée A Un Système D'identification Yoann Charlon*,** – Walid Bourennane*,** – Eric Campo*,** Pag 09

Chapitre 3

[50][Https://Fr.M.Wikipedia.Org/Wiki/G_\(Acc%C3%A9l%C3%A9ration\)](https://Fr.M.Wikipedia.Org/Wiki/G_(Acc%C3%A9l%C3%A9ration))

[51][Https://Www.Tubefr.Com/Mpu6050-Arduino-6-Axes-Accelerometre-Gyro-Gy-521-Test-Simulation-3d_2.Html](https://Www.Tubefr.Com/Mpu6050-Arduino-6-Axes-Accelerometre-Gyro-Gy-521-Test-Simulation-3d_2.Html)

[52][Https://Www.Google.Com/Search?Q=Commandes+At+Du+Micro+Logiciel+Hc06&Tbm=isch&Ved=2ahukewid_Qm3tpzpahvwlwoukhsewb6kq2ccegqjabaa&Oq=Commandes+At+Du+Micro+Logiciel+Hc06&Gs_Lcp=Cgnpbwccqa1cqbliysafgy7mbaabwahgagaghaygb6aesaqmwljeyaqcgaqqg](https://Www.Google.Com/Search?Q=Commandes+At+Du+Micro+Logiciel+Hc06&Tbm=isch&Ved=2ahukewid_Qm3tpzpahvwlwoukhsewb6kq2ccegqjabaa&Oq=Commandes+At+Du+Micro+Logiciel+Hc06&Gs_Lcp=Cgnpbwccqa1cqbliysafgy7mbaabwahgagaghaygb6aesaqmwljeyaqcgaqqg)

aqtn3mtd2l6lwltzw&Sclient=Img&Ei=Wibjxp2elcuelwsh4jzicg&Bih=722&Biw=1536&Rlz=1c1rl
ns_Frdz823dz823#Imgrc=Pmrybs8bnbwvm

[53] https://www.google.com/search?q=Table+Accelerometre+Microsoft+Azure&Tbm=isch&ved=2ahukewjmgorbtfzpahux_Xokhvmodciq2ccegqiaab&Oq=Table+Accelerometre+Microsoft+Azure&Gs_Lcp=Cgnpbwcqa1aawabgrqloahaaeacaadebiahrazibazitmzgbakobc2d3cy13axotaw1n&Sclient=Img&Ei=Gyjjxqaqljfa9ocsjac&Bih=722&Biw=1536&Rlz=1c1rlns_Frdz823dz823#Imgrc=Dgeama-Xrcf1qm

[54] https://www.google.com/search?q=Genetic+Bluetooth+Serial+Windows+App&Rlz=1c1rlns_Frdz823dz823&Sxsr=Alekk02285szt6zmtjnjhfnwyzifampmfw:1591970275518&Source=Lnms&Tbm=isch&Sa=X&ved=2ahukewjcmo-2t_Zpahvscgmbhsondksq_Auoaxoecasqaw&Biw=1536&Bih=722#Imgrc=Aicxmln9pb0k0m

Annexe

Code pour ADXL345 Accelerometer + HC-06 Bluetooth Module

```
#include <Wire.h>
#include <String.h>

#define Register_ID 0
#define Register_2D 0x2D
#define Register_X0 0x32
#define Register_X1 0x33
#define Register_Y0 0x34
#define Register_Y1 0x35
#define Register_Z0 0x36
#define Register_Z1 0x37

int ADXAddress = 0xA7 >> 1;
int reading = 0;
int val = 0;
int X0, X1, X_out;
int Y0, Y1, Y_out;
int Z1, Z0, Z_out;
double Xg, Yg, Zg;

void setup()
{
  Serial.begin(9600);
  Wire.begin();
  delay(100);
  Wire.beginTransmission(ADXAddress);
  Wire.write(Register_2D);
  Wire.write(8);
  Wire.endTransmission();
}

void loop() {
  double tempx = 0;
  char strx[5];
  double tempy = 0;
  char stry[5];
  double tempz = 0;
  char strz[5];

  String gy;

  Wire.beginTransmission(ADXAddress);
  Wire.write(Register_X0);
  Wire.write(Register_X1);
  Wire.endTransmission();
  Wire.requestFrom(ADXAddress, 2);
  if (Wire.available() <= 2);
  {
    X0 = Wire.read();
```

```

    X1 = Wire.read();
    X1 = X1 << 8;
    X_out = X0 + X1;
}

Wire.beginTransmission(ADXAddress);
Wire.write(Register_Y0);
Wire.write(Register_Y1);
Wire.endTransmission();
Wire.requestFrom(ADXAddress, 2);
if (Wire.available() <= 2);
{
    Y0 = Wire.read();
    Y1 = Wire.read();
    Y1 = Y1 << 8;
    Y_out = Y0 + Y1;
}

Wire.beginTransmission(ADXAddress);
Wire.write(Register_Z0);
Wire.write(Register_Z1);
Wire.endTransmission();
Wire.requestFrom(ADXAddress, 2);
if (Wire.available() <= 2);
{
    Z0 = Wire.read();
    Z1 = Wire.read();
    Z1 = Z1 << 8;
    Z_out = Z0 + Z1;
}

Xg = X_out / 256.00;
Yg = Y_out / 256.00;
Zg = Z_out / 256.00;

if (Xg > 0)
{
    gy = String("") + "x=" + Xg;
}
else
{
    gy = String("") + "x=" + Xg;
}

if (Yg > 0)
{
    gy += String("") + "y=" + Yg;
}
else
{
    gy += String("") + "y=" + Yg;
}

```

```

if (Zg > 0)
{
    gy += String("") + "z=" + Zg + ",";
}
else
{
    gy += String("") + "z=" + Zg + ",";
}

Serial.print(gy);
}

```

Le code de l'application Virtual (Language de programmation c # , examl ,cs) GeneticBluetoothSerialUWApp.Sln

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|ARM'">
    <DeviceId>30F105C9-681E-420b-A277-7C086EAD8A4E</DeviceId>
    <UseEmulator>>false</UseEmulator>
    <RemoteDebugMachine>192.168.0.104</RemoteDebugMachine>
    <AuthenticationMode>Universal</AuthenticationMode>
    <RemoteDebugEnabled>>true</RemoteDebugEnabled>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'">
    <AuthenticationMode>Windows</AuthenticationMode>
  </PropertyGroup>
</Project>

```

MainPage.Examel.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;
using Windows.Devices.Enumeration;
using Windows.Devices.Bluetooth.Rfcomm;
using Windows.Networking.Sockets;
using System.Collections.ObjectModel;
using Windows.Storage.Streams;
using Windows.UI.Popups;
using System.Threading.Tasks;
using System.Threading;
using System.Text;
using Windows.System.Threading;

```



```

using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Auth;
using Microsoft.WindowsAzure.Storage.Table;
using System.Net.Http;
using Windows.ApplicationModel.Background;

// The Blank Page item template is documented at
http://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

namespace GenericBluetoothSerialUWApp
{
    public class AccelerometerEntity : TableEntity
    {
        public AccelerometerEntity()
        {
            this.PartitionKey = "ADXL345";
            this.RowKey = Guid.NewGuid().ToString();
            MeasurementTime = DateTime.Now;
            AccelerometerX = 0.0;
            AccelerometerY = 0.0;
            AccelerometerZ = 0.0;
            State = null;
        }
        public DateTime MeasurementTime { get; set; }
        public double AccelerometerX { get; set; }
        public double AccelerometerY { get; set; }
        public double AccelerometerZ { get; set; }
        public string State { get; set; }
    }

    /// <summary>
    /// The Main Page for the app
    /// </summary>
    public sealed partial class MainPage : Page
    {
        string Title = "Generic Bluetooth Serial Universal Windows App";
        private Windows.Devices.Bluetooth.Rfcomm.RfcommDeviceService _service;
        private StreamSocket _socket;
        private DataWriter dataWriterObject;
        private DataReader dataReaderObject;
        ObservableCollection<PairedDeviceInfo> _pairedDevices;
        private CancellationTokenSource ReadCancellationTokenSource;

        //process and transmit timer
        public double tempAccelerometerX = 0;
        public double tempAccelerometerY = 0;
        public double tempAccelerometerZ = 0;

        private static ThreadPoolTimer timerDataProcess;
        private static ThreadPoolTimer timerDataTransfer;
    }
}

```

```

string strMessage;

//for fall detection
public double xtemp, ytemp, ztemp;
public double a_norm;
public int i = 0;
static int BUFF_SIZE = 50;
static public double[] window = new double[BUFF_SIZE];
double sigma = 0.5, th = 10, th1 = 5, th2 = 2;
public static String curr_state, prev_state;

public MainPage()
{
    this.InitializeComponent();
    MyTitle.Text = Title;
    InitializeRfcommDeviceService();
    for (int i = 0; i < BUFF_SIZE; i++)
    {
        window[i] = 0;
    }
    curr_state = null;
    prev_state = null;
}
async void InitializeRfcommDeviceService()
{
    try
    {
        DeviceInformationCollection DeviceInfoCollection = await
DeviceInformation.FindAllAsync(RfcommDeviceService.GetDeviceSelector(RfcommServiceId.SerialPort));

        var numDevices = DeviceInfoCollection.Count();

        // By clearing the backing data, we are effectively clearing the ListBox
        _pairedDevices = new ObservableCollection<PairedDeviceInfo>();
        _pairedDevices.Clear();

        if (numDevices == 0)
        {
            //MessageDialog md = new MessageDialog("No paired devices found", "Title");
            //await md.ShowAsync();
            System.Diagnostics.Debug.WriteLine("InitializeRfcommDeviceService: No paired
devices found.");
        }
        else
        {
            // Found paired devices.
            foreach (var deviceInfo in DeviceInfoCollection)
            {
                _pairedDevices.Add(new PairedDeviceInfo(deviceInfo));
            }
        }
        PairedDevices.Source = _pairedDevices;
    }
}

```

```

    }
    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine("InitializeRfcommDeviceService: " + ex.Message);
    }
}

async private void ConnectDevice_Click(object sender, RoutedEventArgs e)
{
    //Revision: No need to requery for Device Information as we alraedy have it:
    DeviceInformation DeviceInfo; // = await
DeviceInformation.CreateFromIdAsync(this.TxtBlock_SelectedID.Text);
    PairedDeviceInfo pairedDevice = (PairedDeviceInfo)ConnectDevices.SelectedItem;
    DeviceInfo = pairedDevice.DeviceInfo;

    bool success = true;
    try
    {
        _service = await RfcommDeviceService.FromIdAsync(DeviceInfo.Id);

        if (_socket != null)
        {
            // Disposing the socket with close it and release all resources associated with the socket
            _socket.Dispose();
        }

        _socket = new StreamSocket();
        try
        {
            // Note: If either parameter is null or empty, the call will throw an exception
            await _socket.ConnectAsync(_service.ConnectionHostName,
_service.ConnectionServiceName);
        }
        catch (Exception ex)
        {
            success = false;
            System.Diagnostics.Debug.WriteLine("Connect:" + ex.Message);
        }
        // If the connection was successful, the RemoteAddress field will be populated
        if (success)
        {
            this.buttonDisconnect.IsEnabled = true;
            this.buttonSend.IsEnabled = true;
            this.buttonStartRecv.IsEnabled = true;
            this.buttonStopRecv.IsEnabled = false;
            string msg = String.Format("Connected to {0}!",
_socket.Information.RemoteAddress.DisplayName);
            //MessageDialog md = new MessageDialog(msg, Title);
            System.Diagnostics.Debug.WriteLine(msg);
            //await md.ShowAsync();
        }
    }
    catch (Exception ex)

```

```

    {
        System.Diagnostics.Debug.WriteLine("Overall Connect: " + ex.Message);
        _socket.Dispose();
        _socket = null;
    }
}
private void ConnectDevices_DoubleTapped(object sender, DoubleTappedRoutedEventArgs
e)
{
    PairedDeviceInfo pairedDevice = (PairedDeviceInfo)ConnectDevices.SelectedItem;
    this.TxtBlock_SelectedID.Text = pairedDevice.ID;
    this.textBlockBTName.Text = pairedDevice.Name;
    ConnectDevice_Click(sender, e);
}

//Windows.Storage.Streams.Buffer InBuff;
//Windows.Storage.Streams.Buffer OutBuff;
//private StreamSocket _socket;
private async void button_Click(object sender, RoutedEventArgs e)
{
    //OutBuff = new Windows.Storage.Streams.Buffer(100);
    Button button = (Button)sender;
    if (button != null)
    {
        switch ((string)button.Content)
        {
            case "Disconnect":
                await this._socket.CancelIOAsync();
                _socket.Dispose();
                _socket = null;
                this.textBlockBTName.Text = "";
                this.TxtBlock_SelectedID.Text = "";
                this.buttonDisconnect.IsEnabled = false;
                this.buttonSend.IsEnabled = false;
                this.buttonStartRecv.IsEnabled = false;
                this.buttonStopRecv.IsEnabled = false;
                this.ProcessButton.IsEnabled = false;
                this.AzureButton.IsEnabled = false;
                break;
            case "Send":
                //await _socket.OutputStream.WriteAsync(OutBuff);
                Send(this.textBoxSendText.Text);
                this.textBoxSendText.Text = "";
                break;
            case "Clear Send":
                this.textBoxRecvText.Text = "";
                break;
            case "Start Recv":
                this.buttonStartRecv.IsEnabled = false;
                this.buttonStopRecv.IsEnabled = true;
                Listen();
                break;
            case "Stop Recv":

```

```

        this.buttonStartRecv.IsEnabled = false;
        this.buttonStopRecv.IsEnabled = false;
        CancelReadTask();
        break;
    case "Refresh":
        InitializeRfcommDeviceService();
        break;
    }
}
}
public async void Send(string msg)
{
    try
    {
        if (_socket.OutputStream != null)
        {
            // Create the DataWriter object and attach to OutputStream
            dataWriterObject = new DataWriter(_socket.OutputStream);

            //Launch the WriteAsync task to perform the write
            await WriteAsync(msg);
        }
        else
        {
            //status.Text = "Select a device and connect";
        }
    }
    catch (Exception ex)
    {
        //status.Text = "Send(): " + ex.Message;
        System.Diagnostics.Debug.WriteLine("Send(): " + ex.Message);
    }
    finally
    {
        // Cleanup once complete
        if (dataWriterObject != null)
        {
            dataWriterObject.DetachStream();
            dataWriterObject = null;
        }
    }
}
/// <summary>
/// WriteAsync: Task that asynchronously writes data from the input text box 'sendText' to the
OutputStream
/// </summary>
/// <returns></returns>
private async Task WriteAsync(string msg)
{
    Task<UInt32> storeAsyncTask;

    if (msg == "")
        msg = "none";// sendText.Text;
}

```

```

if (msg.Length != 0)
//if (msg.sendText.Text.Length != 0)
{
    // Load the text from the sendText input text box to the dataWriter object
    dataWriterObject.WriteString(msg);

    // Launch an async task to complete the write operation
    storeAsyncTask = dataWriterObject.StoreAsync().AsTask();

    UInt32 bytesWritten = await storeAsyncTask;
    if (bytesWritten > 0)
    {
        string status_Text = msg + ", ";
        status_Text += bytesWritten.ToString();
        status_Text += " bytes written successfully!";
        System.Diagnostics.Debug.WriteLine(status_Text);
    }
}
else
{
    string status_Text2 = "Enter the text you want to write and then click on 'WRITE'";
    System.Diagnostics.Debug.WriteLine(status_Text2);
}
}

/// <summary>
/// - Create a DataReader object
/// - Create an async task to read from the SerialDevice InputStream
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private async void Listen()
{
    try
    {
        ReadCancellationTokenSource = new Cancellation-tokenSource();
        if (_socket.InputStream != null)
        {
            dataReaderObject = new DataReader(_socket.InputStream);
            this.buttonStopRecv.IsEnabled = true;
            this.ProcessButton.IsEnabled = true;
            this.AzureButton.IsEnabled = true;
            this.buttonDisconnect.IsEnabled = false;
            // keep reading the serial input
            while (true)
            {
                await ReadAsync(ReadCancellationTokenSource.Token);
            }
        }
    }
    catch (Exception ex)
    {
        this.buttonStopRecv.IsEnabled = false;
    }
}

```

```

        this.buttonStartRecv.IsEnabled = false;
        this.buttonSend.IsEnabled = false;
        this.buttonDisconnect.IsEnabled = false;
        this.textBlockBTName.Text = "";
        this.TxtBlock_SelectedID.Text = "";
        if (ex.GetType().Name == "TaskCanceledException")
        {
            System.Diagnostics.Debug.WriteLine("Listen: Reading task was cancelled, closing device
and cleaning up");
        }
        else
        {
            System.Diagnostics.Debug.WriteLine("Listen: " + ex.Message);
        }
    }
    finally
    {
        // Cleanup once complete
        if (dataReaderObject != null)
        {
            dataReaderObject.DetachStream();
            dataReaderObject = null;
        }
    }
}
/// <summary>
/// ReadAsync: Task that waits on data and reads asynchronously from the serial device
InputStream
/// </summary>
/// <param name="cancellationToken"></param>
/// <returns></returns>
private async Task ReadAsync(CancellationTokentoken cancellationToken)
{
    Task<UInt32> loadAsyncTask;

    uint ReadBufferLength = 1024;

    // If task cancellation was requested, comply
    cancellationToken.ThrowIfCancellationRequested();

    // Set InputStreamOptions to complete the asynchronous read operation when one or more
bytes is available
    dataReaderObject.InputStreamOptions = InputStreamOptions.Partial;

    // Create a task object to wait for data on the serialPort.InputStream
loadAsyncTask
dataReaderObject.LoadAsync(ReadBufferLength).AsTask(cancellationToken);

    // Launch the task and wait
    UInt32 bytesRead = await loadAsyncTask;
    if (bytesRead > 0)
    {
        try

```

```

    {
        string recvdtxt = dataReaderObject.ReadString(bytesRead);
        System.Diagnostics.Debug.WriteLine(recvdtxt);
        this.textBoxRecvText.Text += recvdtxt;
        strMessage += recvdtxt;
        //strMessage.Append(recvdtxt);
        /*if (_Mode == Mode.JustConnected)
        {
            if (recvdtxt[0] == ArduinoLCDDisplay.keypad.BUTTON_SELECT_CHAR)
            {
                _Mode = Mode.Connected;

                //Reset back to Cmd = Read sensor and First Sensor
                await Globals.MP.UpdateText("@");
                //LCD Display: Fist sensor and first comamnd
                string lcdMsg = "~C" + Commands.Sensors[0];
                lcdMsg += "~" + ArduinoLCDDisplay.LCD.CMD_DISPLAY_LINE_2_CH +
Commands.CommandActions[1] + " ";
                Send(lcdMsg);

                backButton_Click(null, null);
            }
        }
        else if (_Mode == Mode.Connected)
        {
            await Globals.MP.UpdateText(recvdtxt);
            recvText.Text = "";
            status.Text = "bytes read successfully!";
        }*/
    }
    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine("ReadAsync: " + ex.Message);
    }
}

}

/// <summary>
/// CancelReadTask:
/// - Uses the ReadCancellationTokenSource to cancel read operations
/// </summary>
private void CancelReadTask()
{
    if (ReadCancellationTokenSource != null)
    {
        if (!ReadCancellationTokenSource.IsCancellationRequested)
        {
            ReadCancellationTokenSource.Cancel();
        }
    }
}
}

```



```

/// <summary>
/// Class to hold all paired device information
/// </summary>
public class PairedDeviceInfo
{
    internal PairedDeviceInfo(DeviceInformation deviceInfo)
    {
        this.DeviceInfo = deviceInfo;
        this.ID = this.DeviceInfo.Id;
        this.Name = this.DeviceInfo.Name;
    }

    public string Name { get; private set; }
    public string ID { get; private set; }
    public DeviceInformation DeviceInfo { get; private set; }
}

private void AzureButton_Click(object sender, RoutedEventArgs e)
{
    timerDataTransfer = ThreadPoolTimer.CreatePeriodicTimer(dataTransferTick,
    TimeSpan.FromMilliseconds(Convert.ToInt32(3000)));
}

private async void dataTransferTick(ThreadPoolTimer timer)
{
    try
    {
        CloudStorageAccount storageAccount =
        CloudStorageAccount.Parse("DefaultEndpointsProtocol=https;AccountName=;AccountKey=");

        // Create the table client.
        CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

        // Create the CloudTable object that represents the " AccelerometerTable " table.
        CloudTable table = tableClient.GetTableReference("AccelerometerTable");
        await table.CreateIfNotExistsAsync();

        // Create a new customer entity.
        AccelerometerEntity ent = new AccelerometerEntity();

        ent.MeasurementTime = DateTime.Now;
        ent.AccelerometerX = tempAccelerometerX;
        ent.AccelerometerY = tempAccelerometerY;
        ent.AccelerometerZ = tempAccelerometerZ;
        ent.State = curr_state;
        // Create the TableOperation that inserts the customer entity.
        TableOperation insertOperation = TableOperation.Insert(ent);
        // Execute the insert operation.
        await table.ExecuteAsync(insertOperation);
    }
    catch (Exception ex)
    {
        MessageDialog dialog = new MessageDialog("Error sending to Azure: " + ex.Message);
    }
}

```

```

        dialog.ShowDialog();
    }
}

private void ProcessButton_Click(object sender, RoutedEventArgs e)
{
    timerDataProcess = ThreadPoolTimer.CreatePeriodicTimer(dataProcessTick,
    TimeSpan.FromMilliseconds(Convert.ToInt32(300)));
}

private async void dataProcessTick(ThreadPoolTimer timer)
{
    string[] arr = strMessage.Split(',');
    string process = null;
    int n = arr.Length;
    int i = 0;
    for (i = n - 1; i >= 0; i--)
    {
        if (arr[i].Length == 21)
        {
            process = arr[i];
            break;
        }
    }
    //string process = arr[0];
    double xtemp, ytemp, ztemp;
    if (process.Substring(2, 1) == "+")
        xtemp = Convert.ToDouble(process.Substring(3, 4));
    else
        xtemp = Convert.ToDouble(process.Substring(2, 5));

    if (process.Substring(9, 1) == "+")
        ytemp = Convert.ToDouble(process.Substring(10, 4));
    else
        ytemp = Convert.ToDouble(process.Substring(9, 5));

    if (process.Substring(16, 1) == "+")
        ztemp = Convert.ToDouble(process.Substring(17, 4));
    else
        ztemp = Convert.ToDouble(process.Substring(16, 5));

    AddData(xtemp, ytemp, ztemp);

    tempAccelerometerX = xtemp;
    tempAccelerometerY = ytemp;
    tempAccelerometerZ = ztemp;

    posture_recognition(window, ytemp);
    SystemState(curr_state, prev_state);

    if (prev_state != curr_state)
    {
        prev_state = curr_state;
    }
}

```

```

    }

    await
textBoxRecvdText.Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.Normal, () =>
    {
        textBoxRecvdText.Text = process;
    }
);

    strMessage = "";
}

private void posture_recognition(double[] window2, double ay2)
{
    // TODO Auto-generated method stub
    int zrc = compute_zrc(window2);
    if (zrc == 0)
    {
        if (Math.Abs(ay2) < th1)
        {
            curr_state = "sitting";
        }
        else {
            curr_state = "standing";
        }
    }
    else
    {
        if (zrc > th2)
        {
            curr_state = "walking";
        }
        else {
            curr_state = "none";
        }
    }
}

private int compute_zrc(double[] window2)
{
    // TODO Auto-generated method stub
    int count = 0;
    for (i = 1; i <= BUFF_SIZE - 1; i++)
    {
        if ((window2[i] - th) < sigma && (window2[i - 1] - th) > sigma)
        {
            count = count + 1;
        }
    }
    return count;
}

private async void SystemState(String curr_state1, String prev_state1)

```

```

    {
        //Fall !!
        if (prev_state1!=curr_state1)
        {
            await
textStateText.Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.Normal, () =>
            {
                textStateText.Text = curr_state1;
            }
        );
    }
}

private void AddData(double ax2, double ay2, double az2)
{
    // TODO Auto-generated method stub
    a_norm = Math.Sqrt(ax2 * ax2 + ay2 * ay2 + az2 * az2);
    for (i = 0; i <= BUFF_SIZE - 2; i++)
    {
        window[i] = window[i + 1];
    }
    window[BUFF_SIZE - 1] = a_norm;
}

}
}

```

App.Xaml

```

<Application
  x:Class="GenericBluetoothSerialUWApp.App"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:GenericBluetoothSerialUWApp"
  RequestedTheme="Light">

</Application>

```

App.Examel.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.ApplicationModel;
using Windows.ApplicationModel.Activation;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

namespace GenericBluetoothSerialUWApp

```

```

{
    /// <summary>
    /// Provides application-specific behavior to supplement the default Application class.
    /// </summary>
    sealed partial class App : Application
    {
        /// <summary>
        /// Initializes the singleton application object. This is the first line of authored code
        /// executed, and as such is the logical equivalent of main() or WinMain().
        /// </summary>
        public App()
        {
            this.InitializeComponent();
            this.Suspending += OnSuspending;
        }

        /// <summary>
        /// Invoked when the application is launched normally by the end user. Other entry points
        /// will be used such as when the application is launched to open a specific file.
        /// </summary>
        /// <param name="e">Details about the launch request and process.</param>
        protected override void OnLaunched(LaunchActivatedEventArgs e)
        {
            // #if DEBUG
            //     if (System.Diagnostics.Debugger.IsAttached)
            //     {
            //         this.DebugSettings.EnableFrameRateCounter = true;
            //     }
            // #endif

            Frame rootFrame = Window.Current.Content as Frame;

            // Do not repeat app initialization when the Window already has content,
            // just ensure that the window is active
            if (rootFrame == null)
            {
                // Create a Frame to act as the navigation context and navigate to the first page
                rootFrame = new Frame();

                rootFrame.NavigationFailed += OnNavigationFailed;

                if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)
                {
                    // TODO: Load state from previously suspended application
                }

                // Place the frame in the current Window
                Window.Current.Content = rootFrame;
            }

            if (rootFrame.Content == null)
            {
                // When the navigation stack isn't restored navigate to the first page,
                // configuring the new page by passing required information as a navigation
                // parameter
                rootFrame.Navigate(typeof(MainPage), e.Arguments);
            }
            // Ensure the current window is active
            Window.Current.Activate();
        }

        /// <summary>
        /// Invoked when Navigation to a certain page fails
        /// </summary>
        /// <param name="sender">The Frame which failed navigation</param>
        /// <param name="e">Details about the navigation failure</param>
    }
}

```

```

void OnNavigationFailed(object sender, NavigationFailedEventArgs e)
{
    throw new Exception("Failed to load Page " + e.SourcePageType.FullName);
}

/// <summary>
/// Invoked when application execution is being suspended. Application state is saved
/// without knowing whether the application will be terminated or resumed with the contents
/// of memory still intact.
/// </summary>
/// <param name="sender">The source of the suspend request.</param>
/// <param name="e">Details about the suspend request.</param>
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    //TODO: Save application state and stop any background activity
    deferral.Complete();
}
}
}

```

Genetic BluetoothSerialUWApp.csproj

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
    Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props') />
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''>Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''>x86</Platform>
    <ProjectGuid>{CCD27A13-68E5-4F71-B606-C3C14B294042}</ProjectGuid>
    <OutputType>AppContainerExe</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>GenericBluetoothSerialUWApp</RootNamespace>
    <AssemblyName>GenericBluetoothSerialUWApp</AssemblyName>
    <DefaultLanguage>en-US</DefaultLanguage>
    <TargetPlatformIdentifier>UAP</TargetPlatformIdentifier>
    <TargetPlatformVersion>10.0.10240.0</TargetPlatformVersion>
    <TargetPlatformMinVersion>10.0.10240.0</TargetPlatformMinVersion>
    <MinimumVisualStudioVersion>14</MinimumVisualStudioVersion>
    <EnableDotNetNativeCompatibleProfile>true</EnableDotNetNativeCompatibleProfile>
    <FileAlignment>512</FileAlignment>
    <ProjectTypeGuids>{A5A43C5B-DE2A-4C0C-9213-0A381AF9435A};{FAE04EC0-301F-11D3-BF4B-
00C04F79EFBC}</ProjectTypeGuids>
    <PackageCertificateKeyFile>Bluetooth2_TemporaryKey.pfx</PackageCertificateKeyFile>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Debug | ARM'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\ARM\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE;NETFX_CORE;WINDOWS_UWP</DefineConstants>
    <NoWarn>;2008</NoWarn>
    <DebugType>full</DebugType>
    <PlatformTarget>ARM</PlatformTarget>
    <UseVSHostingProcess>>false</UseVSHostingProcess>
    <ErrorReport>prompt</ErrorReport>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Release | ARM'">
    <OutputPath>bin\ARM\Release\</OutputPath>
    <DefineConstants>TRACE;NETFX_CORE;WINDOWS_UWP</DefineConstants>
    <Optimize>true</Optimize>
    <NoWarn>;2008</NoWarn>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>ARM</PlatformTarget>
    <UseVSHostingProcess>>false</UseVSHostingProcess>
    <ErrorReport>prompt</ErrorReport>
  </PropertyGroup>

```

```

    <Prefer32Bit>true</Prefer32Bit>
    <UseDotNetNativeToolchain>true</UseDotNetNativeToolchain>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Debug|x64'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\x64\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE;NETFX_CORE;WINDOWS_UWP</DefineConstants>
    <NoWarn>;2008</NoWarn>
    <DebugType>full</DebugType>
    <PlatformTarget>x64</PlatformTarget>
    <UseVSHostingProcess>>false</UseVSHostingProcess>
    <ErrorReport>prompt</ErrorReport>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Release|x64'">
    <OutputPath>bin\x64\Release\</OutputPath>
    <DefineConstants>TRACE;NETFX_CORE;WINDOWS_UWP</DefineConstants>
    <Optimize>true</Optimize>
    <NoWarn>;2008</NoWarn>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>x64</PlatformTarget>
    <UseVSHostingProcess>>false</UseVSHostingProcess>
    <ErrorReport>prompt</ErrorReport>
    <Prefer32Bit>true</Prefer32Bit>
    <UseDotNetNativeToolchain>true</UseDotNetNativeToolchain>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Debug|x86'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\x86\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE;NETFX_CORE;WINDOWS_UWP</DefineConstants>
    <NoWarn>;2008</NoWarn>
    <DebugType>full</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <UseVSHostingProcess>>false</UseVSHostingProcess>
    <ErrorReport>prompt</ErrorReport>
    <Prefer32Bit>true</Prefer32Bit>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Release|x86'">
    <OutputPath>bin\x86\Release\</OutputPath>
    <DefineConstants>TRACE;NETFX_CORE;WINDOWS_UWP</DefineConstants>
    <Optimize>true</Optimize>
    <NoWarn>;2008</NoWarn>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <UseVSHostingProcess>>false</UseVSHostingProcess>
    <ErrorReport>prompt</ErrorReport>
    <Prefer32Bit>true</Prefer32Bit>
    <UseDotNetNativeToolchain>true</UseDotNetNativeToolchain>
  </PropertyGroup>
  <ItemGroup>
    <!-- A reference to the entire .Net Framework and Windows SDK are automatically included -->
    <None Include="project.json" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="App.xaml.cs">
      <DependentUpon>App.xaml</DependentUpon>
    </Compile>
    <Compile Include="MainPage.xaml.cs">
      <DependentUpon>MainPage.xaml</DependentUpon>
    </Compile>
    <Compile Include="Properties\AssemblyInfo.cs" />
  </ItemGroup>
  <ItemGroup>
    <AppxManifest Include="Package.appxmanifest">
      <SubType>Designer</SubType>
    </AppxManifest>
    <None Include="Bluetooth2_TemporaryKey.pfx" />
  </ItemGroup>

```

```

</ItemGroup>
<ItemGroup>
  <Content Include="Properties\Default.rd.xml" />
  <Content Include="Assets\LockScreenLogo.scale-200.png" />
  <Content Include="Assets\SplashScreen.scale-200.png" />
  <Content Include="Assets\Square150x150Logo.scale-200.png" />
  <Content Include="Assets\Square44x44Logo.scale-200.png" />
  <Content Include="Assets\Square44x44Logo.targetsize-24_altform-unplated.png" />
  <Content Include="Assets\StoreLogo.png" />
  <Content Include="Assets\Wide310x150Logo.scale-200.png" />
</ItemGroup>
<ItemGroup>
  <ApplicationDefinition Include="App.xaml">
    <Generator>MSBuild:Compile</Generator>
    <SubType>Designer</SubType>
  </ApplicationDefinition>
  <Page Include="MainPage.xaml">
    <Generator>MSBuild:Compile</Generator>
    <SubType>Designer</SubType>
  </Page>
</ItemGroup>
<PropertyGroup Condition="'$(VisualStudioVersion)' == " or '$(VisualStudioVersion)' &lt; '14.0'">
  <VisualStudioVersion>14.0</VisualStudioVersion>
</PropertyGroup>
<Import
Project="$(MSBuildExtensionsPath)\Microsoft\WindowsXaml\v$(VisualStudioVersion)\Microsoft.Windows.UI.Xaml.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and uncomment it.
      Other similar extension points exist, see Microsoft.Common.targets.
-->
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

Genetic BluthoothSerialUWApp.csproj.User

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Debug|ARM'">
    <DeviceId>30F105C9-681E-420b-A277-7C086EAD8A4E</DeviceId>
    <UseEmulator>>false</UseEmulator>
    <RemoteDebugMachine>192.168.0.104</RemoteDebugMachine>
    <AuthenticationMode>Universal</AuthenticationMode>
    <RemoteDebugEnabled>>true</RemoteDebugEnabled>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration) | $(Platform)' == 'Debug|x86'">
    <AuthenticationMode>Windows</AuthenticationMode>
  </PropertyGroup>
</Project>

```


Résumé

Ce projet aborde la conception et l'intégration d'un système de surveillance des personnes âgées basé sur Arduino, Windows 10 et Microsoft Azure, de qui permet de détecter les chutes en temps réel et fournissant ainsi des données de l'accéléromètre au moment de la chute. Les données brutes de l'accéléromètre tridimensionnel sont fournies par Arduino avec ADXL345. L'appareil Windows 10 utilise ces informations pour obtenir l'orientation du sujet grâce à des algorithmes efficaces de fusion des données et de détection des chutes. Les services Microsoft Azure et l'application Mobile / PC sont également utilisés pour réaliser un traitement, une analyse, un stockage et une acquisition de données sans interruption à tout moment et en tout lieu, tant qu'ils ont accès à Internet.

Mot clés : personnes âgées, Arduino , chutes, l'accéléromètre , Microsoft Azure , Windows 10.

Abstract

This Project addresses the design and integration of senior monitoring system based on Arduino, Windows 10 and Microsoft Azure, which can detect falls in real time and thus provide accelerometer data at the time of the fall. The three-dimensional accelerometer raw, data is provided by Arduino with ADXL345. Windows 10 device uses this information to get subject orientation through efficient data fusion and fall detection algorithms. Microsoft azure services and the mobile/PC app are also used to perform uninterrupted data processing, analysis, storage and acquisition at any time and location, as long as they have internet access.

Keywords: seniors, Arduino , falls, accelerometer , Microsoft Azure , Windows 10

ملخص

يتناول هذا المشروع تصميم ودمج نظام مراقبة كبار السن يعتمد على Arduino و Windows 10 و Microsoft Azure ، والذي يمكنه اكتشاف السقوط في الوقت الفعلي وبالتالي توفير بيانات مقياس التسارع في وقت السقوط. يتم توفير البيانات الأولية من مقياس التسارع ثلاثي الأبعاد بواسطة Arduino مع ADXL345. يستخدم جهاز Windows 10 هذه المعلومات للحصول على اتجاه الشخص المسن من خلال خوارزميات فعالة مستخلصة من دمج هي البيانات والكشف عن السقوط. تُستخدم أيضًا خدمات Microsoft Azure وتطبيق Mobile / PC لإجراء معالجة البيانات وتحليلها وتخزينها واكتسابها دون انقطاع في أي وقت وفي أي مكان ، طالما كان لديهم إمكانية الوصول إلى الإنترنت.

الكلمات الدالة : كبار السن ، Arduino ، مقياس التسارع ، Windows 10. Microsoft Azure ، يتبعون