

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE Mouloud MAMMERI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE

Filière : Electronique

Spécialité : Instrumentation

Présenté par :

Melle TEBAH NASSIMA

Mr ZIOUI AHMED

Thème

**Système autonome de détection et de prévention
d'incendie**

Président : Mr BENNAMANE.K

Examineur : SIDI SAID.A

Promoteur : Mr ZIRML.R

PROMOTION 2017/2018

Remerciements

Nous remercions d'abord dieu le tout puissant pour la santé, le courage et la foi qu'il nous a donné pour arriver à ce jour.

Nous tenons aussi à remercier l'encadreur monsieur ZIRMI Rachid pour son aide, ses précieux conseils, et sa patience tout au long de ce travail ainsi que les membres du jury qui nous ont fait l'honneur d'examiner notre travail.

Nos remerciements les plus chaleureux vont à nos enseignants pour leurs encouragements durant toutes ces années d'étude.

Nos sincères remerciements à tous les étudiants de la promotion.



Dédicaces

Nos dédications ce travail à :

✚ *Nos chers parents,*

✚ *Nos familles, frères et sœurs*

✚ *Tous nos amis (es) et nos camarades.*

Nassíma et Ahmed.



Ch4	Methane
Ftdi	Future Technology Devices International
GSM	Global System for Mobile Communications
I2C	Inter-Integrated Circuit,
IDE	Integrated development environment
LCD	Liquid-crystal display
MHz	Mégahertz, unité de mesure de fréquence du Système international (SI), valant 10^6 hertz
PWM	Pulse-width modulation
SMS	Short message service
SPI	<i>Serial Peripheral Interface</i>
Uart	Universal asynchronous receiver-transmitter
Usb	Universal serial bus

Figure 01 : Causes humaines.....	3
Figure 02 : Causes naturelles.....	4
Figure 03 : Risque d'incendie en Algérie.....	5
Figure 04 : Carte Arduino Mega	11
Figure 05 : Architecteur interne de la carte Arduino Mega	12
Figure 06 : Architecteur d'un microcontrôleur Mega2560	13
Figure 07 : Architecture software.....	15
Figure 08 : Module GSM A6	18
Figure 09 : Pins du module GSM A6.....	19
Figure 10 : Capteur MQ-7	22
Figure 11 : Réglage de sensibilité du MQ-7.....	23
Figure 12 : Capteur de flamme.....	24
Figure 13 : Module BUZZER	25
Figure 14 : Panneaux solaires photovoltaïque.....	25
Figure 15 : Schéma d'une cellule photovoltaïque.....	26
Figure 16 : Module photovoltaïque	27
Figure 17 : Champ photovoltaïque.....	27
Figure 18 : Alimentation du GSM.....	29
Figure 19 : Câblage de module GSM.....	30
Figure 20 : Brochage du module GSM	31
Figure 21 : Brochage du capteur MQ-7.....	32

Figure 22 : Brochage du capteur de flamme	33
Figure 23 : Brochage de l'afficheur LCD	35
Figure 24 : Brochage des LEDs	36
Figure 25 : Amorçage des capteurs	37
Figure 26 : Schéma complet sur issisproteus	38
Figure 27 : Alimentation solaire de l'Arduino	39

INTRODUCTION

CHAPITRE I :Généralités

I.1.Introduction.....	1
I.2.Feux de forêt	1
2.1. Définition d’incendie.....	1
2.2. Type de feux de forêt	1
2.2.1. Feux de sol	1
2.2.2. Feux de surface	2
2.2.3. Feux de cimes	2
I.3. Cause des feux de forêt.....	2
3.1. Causes humaines	2
3.2. Causes naturelles	3
I.4 .Feux de forêt en Algérie	3
I.5. Détection d’incendie	4
5.1. Détection aérienne.....	4
5.1.1. Systèmes satellites.....	4
5.1.2. Véhicules aériens autonomes	5
5.2. Détection terrestre	6
5.2.1. Système de vidéosurveillance automatique.....	6
5.2.2. Réseau de capteurs à fibre optique	7
5.2.3. Réseaux de capteurs sans fil.....	8
I.6. Conclusion	8

CHAPITRE II : Descriptions des composants

II.1 Introduction	9
II.2. Arduino	9
2.1. Définition d'arduino	9
2.2. Carte Arduino méga	9
2.2.1. Définition	9
2.2.2. Caractéristiques principales.....	10
2.2.3. Architecture de la carte.....	11
2.2.4. Microcontrôleur AT Mega 2560	11
2.2.5. Alimentation de la carte Arduino.	13
2.3. Environnement de programmation.....	13
2.3.1. Architecture software	13
2.3.2. Structures du programme IDE.....	14
2.4. Exemple d'utilisation	15
II.3. Module GSM A6	16
3.1. Définition	16
3.2. Caractéristiques du module A6	17
3.3. Commandes AT sur Arduino	19
3.3.1. Présentation	19
3.3.2. Configuration de la liaison série logicielle.....	19
II.4. Capteur MQ-7.....	20
4.1. Introduction	20

4.2. Définition	21
4.3. Caractéristiques	22
II.5. Capteur de flamme.....	23
5.1. Définition	23
5.2. Caractéristiques	23
5.3. Spécifications	24
II.6. Module BUZZER	24
II.7. Panneaux solaires photovoltaïque	24
7.1. Générateur Photovoltaïque	25
7.2. Cellule photovoltaïque	25
7.3. Module photovoltaïque	26
7.4. Champ photovoltaïque	26
II.8. Conclusion	27

Chapitre III : Réalisation et initialisation

III-1- Introduction	28
III-2- Initialisation du module GSM	28
2.1 .Alimentations du module GSM.....	28
2.2. Brochage du module GSM.....	29
2.3. Programme d'initialisation.....	30
III.3. Brochage des capteurs	31

3.1. MQ7	31
3.1.1 Brochage du capteur mq7.....	31
3.1.2. Programme associé.....	31
3.2. Capteur de flamme	32
3.2.1. Brochage du capteur de flamme.....	32
3.2.2. Programme associé.....	33
III.4 .Initialisation de L'afficheur LCD.....	33
4.1. Brochage du LCD.....	33
4.2. Programme associé.....	34
III.5. Connexion des LEDs.....	35
5.1. Brochage desLEDs.....	35
5.2. Programme associé.....	35
III.6. Amorçage des capteurs.....	36
III.7. shemat complet sur ISIS Proteus.....	37
III.8. Alimentation de l'arduino	37
III.9. Conclusion.....	37

Chapitre IV : Fonctionnement

IV.1. Introduction	39
IV.2. Organigramme du programme	39
IV.3. Fonctionnement sans amorçage des capteurs.....	40

IV.4. Fonctionnement en amorçage d'un des capteurs	40
4.1. Amorçages d'un des capteurs.....	40
4.2 Arrêt d'amorçage.....	40
IV.5. Fonctionnement en amorçage des deux capteurs	41
5.1. Amorçage des deux capteurs.....	41
5.2. Arrêt d'amorçage d'un seul capteur	41
IV.6. Programme de fonctionnement	41
6.1. Initialisation et déclaration de variables.....	41
6.2. La fonction void setup ()	42
6.3. La fonction voidloop ()	44
7. Conclusion.....	51

CONCLUSION

Références bibliographiques

Dans le cadre de la préservation et la conservation des espaces forestiers et agricoles, nous avons proposé comme projet de fin d'études un system autonome de détection d'incendie intitulé «Système autonome de détection et de prévention d'incendie »et reposant sur une autonomie énergétique totale garantie par un panneau solaire qui charge une batterie, cette dernière alimente notre système le jour comme la nuit.

La démarche de fabrication de ce détecteur électronique que nous avons mis en évidence repose sur une analyse de plusieurs substances qui désignent une présence d'un incendie et lui attribué une action de mettre en garde qui se constitue en allumage des LEDs et d'une alarme avec affichage sur un écran LCD, ensuite mettre au courant les gardes forestiers et la protection civiles en envoyant un message en modes SMS.

Dans un premier chapitre, nous procédons à une étude sur les feux de forêt, les types de détection et les méthodes de lutte en donnant les raisons, les causes et les conséquences de ce désastre.

Dans le deuxième chapitre, nous présentons un portrait des composants électroniques requis dans la réalisation et les outils informatique nécessaires, ainsi que leurs caractéristiques et mode de fonctionnement afin de les manipuler plut-tard.

Nous allons aussi définir dans un troisième chapitre les connections et brochage entre les divers composants avec leurs programmes d'initialisation.

Le dernier chapitre est consacré au traitement des données acquises par le système en décrivant sa gestion de ces données et son intervention pour l'avertissement dans le cas de détection de signes de feu.

I.1.Introduction

De nos jours, le monde connaît un grand désastre ravageant des immenses étendues de pleine et de forêt peuplés principalement par des arbres, ainsi que des espaces agricoles qui partent en fumé, ils sont causés par vastes incendies provoqués par la nature et principalement par l'homme de manière volontaire ou accidentelle.

I.2. Feux de forêt

I.2.1. Définition d'incendie

On parle de feu de forêt lorsque l'incendie couvre une surface minimale de 0.5 hectare d'un seul terrain et qu'une partie ou la totalité des étages arbustifs et la végétation arborescente sont détruites. Le terme incendie est utilisé aussi pour la formation de feu sur des portions de plus petites tailles que sont le maquis, la garrigue et les landes. Dans le cas de la responsabilité humaine, la cause peut être volontaire, involontaire ou liée aux infrastructures.

Pour qu'un feu se déclenche il est indispensable de réunir trois ingrédients essentielle : un combustible, un comburant et aussi une énergie d'activation (source de chaleur). Dans le cas des feux de forêt la végétation tient le rôle de combustible, l'air et l'oxygène font l'effet de comburant et la moindre étincelle peut alors suffire à apporter une énergie d'activation suffisante.

I.2.2. Type des feux de forêt

Le feu de forêt prend différentes formes en fonction des conditions climatique et les caractéristiques de la végétation.

I.2.2.1. Feux de sol

Ils brûlent la matière organique contenue dans la litière, l'humus ou les tourbières. Ils sont alimentés par incandescence avec combustion et ces feux ont une faible vitesse de propagation.

I.2.2.2.Feux de surface

Ils consomment les strates basses de la végétation et se propagent en générale par rayonnement ou convection, Ils affectent les garrigues ou les landes.

I.2.3. Feux de cimes

Ils atteignent la partie supérieure des arbres (ligneux hauts) en formant une couronne de feu, ils libèrent en générale de grandes quantités d'énergie, leurs vitesses de propagation sont très élevées et ils sont particulièrement intenses et difficiles à contrôler lorsque le vent est fort et le combustible sec

I.3. Cause des feux de forêt

Il existe plusieurs causes provoquant les incendies, elles peuvent être naturelles ou humaines.

I.3.1. Causes humaines



Figure 01: Causes humaines

Les êtres humains ont une grande responsabilité dans le déclenchement des feux de forêt, d'une manière volontaire comme la pyromanie, la vengeance, un enjeu politique, économique ou social et d'une façon involontairement par imprudence ou accidents.

I.3.2. Causes naturelles**Figure 02 : Causes naturelles**

La végétation ne s'enflamme pas toute seule, même s'il y a une forte sécheresse, l'unique cause naturelle connue dans le Bassin Méditerranéen est la foudre, ce phénomène est cependant exceptionnel en Algérie où on admet que 20% de ces cas sont des interprétations comme « l'origine naturelle » du phénomène ou bien « les conditions climatiques » et il existe d'autres sources comme les volcans et les chutes de météorites...etc.

I.4. Feux de forêt en Algérie

L'Algérie fait partie des nations où le problème des feux de forêts est assez peu connu par la communauté scientifique : si en valeur absolue les superficies brûlées restent relativement modestes au regard des autres pays du pourtour méditerranéen, la rareté des forêts et les menaces de désertification font que ces incendies ont un impact particulièrement désastreux. L'Algérie ne possède en effet que 4,1 millions d'hectares de forêts, soit un taux de boisement de 1,76%. Or la fréquence rapprochée des incendies qui se suivent avec un intervalle de retour de moins 10 ans à un impact catastrophique sur le plan écologique.

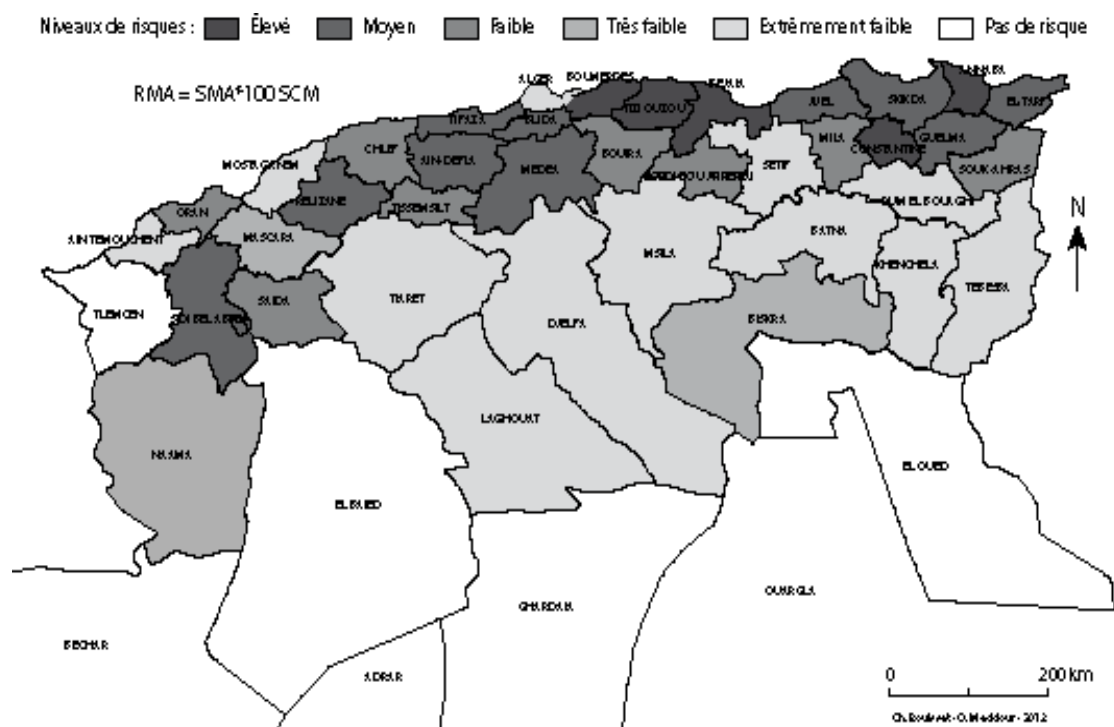


Figure 03 : Risque d'incendie en Algérie.

L'analyse des feux de forêt durant la période 1985-2016 faite au niveau des 40 wilayas de l'Algérie du nord (la partie la plus boisée), a fait ressortir que 42 555 incendies ont parcouru une superficie forestière totale de 910 640 hectares.

I.5. Détection d'incendie

Les systèmes utilisés pour la détection d'incendie des forêts peuvent être divisés en deux groupes en fonction de l'endroit où les capteurs sont déployés.

I.5.1. Détection aérienne

La détection aérienne est subdivisée en deux grandes catégories : les systèmes satellitaires et les véhicules (dispositifs) aériens autonomes.

I.5.1.1. Systèmes satellitaires

Il existe deux types de satellites qui apparaissent dans le contexte de la détection des incendies de forêt : ils sont les satellites géostationnaires et les satellites en orbite polaire.

Les satellites géostationnaires maintiennent une résolution temporelle plus élevée par rapport aux satellites en orbites polaire, car ils détectent les feux de forêt actifs en temps quasi réel. Dans la plupart des cas, ils sont en liaison avec une station de base qui collecte les données transmises par les satellites où elles sont analysées, envoyées sous forme d'images satellitaires et ces dernières sont utilisées par plusieurs agences pour la détection et le suivi des incendies ; tel que le système canadien.

L'utilisation des satellites pour la détection et la surveillance des incendies forestiers reste une option dominante, pourtant, ce système contient un ensemble d'inconvénients :

- ❖ Il y a une taille minimale détectable de feu, généralement 0.1 ha cela signifie qu'il y a un certain retard à détecter l'incendie, qui est au moins égal au temps nécessaire pour le feu d'aller au-delà de 0.1 ha
- ❖ La précision du système est à 1 km de l'emplacement réel de l'incendie généralement fournir des images satellites de la zone suivie tous les 1 à 2 jours, par conséquent, le feu pourrait passer inaperçue pendant une longue période.
- ❖ Les nuages affectent considérablement la température de l'atmosphère de haut car ils peuvent abaisser la température mesurée par le satellite
- ❖ Sélection de l'algorithme de traitement varie de capteur différent ayant chacun ses limites spécifiques.

I.5.1.2. Véhicules aériens autonomes

Les hélicoptères munis des caméras infrarouges et visuelles peuvent être utilisés dans la détection des feux, ils sont appelés « véhicules aériens » ou « véhicules aériens sans pilote(UAV), une station de base est nécessaire où les données recueillies sont analysées. La détection d'incendie est lancée dans les missions où la flotte de drones s'embarque pour patrouiller la forêt en question.

Afin de réduire la probabilité de fausses alarmes, ce système repose sur la combinaison des résultats obtenus à partir de l'analyse des images infrarouge et visuelles. Des capteurs de navigation sont installés sur l'UAV pour calculer la position et l'orientation des caméras, ces positions calculées sont ensuite recueillies l'indien réel et ce système souffre de certains inconvénients tels que :

L'utilisation d'un grand nombre de véhicules pendant un long temps de vol par appareil pour couvrir une grande surface, signifie plus de consommation de carburant et d'où un coût de fonctionnement plus élevé.

- ❖ La visibilité est limitée lorsque la fumée est présente
- ❖ L'évaluation des incendies varie considérablement en fonction de la topographie de la forêt
- ❖ Le vol des véhicules aériens sans pilote tout au long de la journée donnera lieu une augmentation de coût
- ❖ La communication radio peut être entravée dans les régions montagneuses

I.5.2. Détection terrestre

Plusieurs systèmes terrestres ont été utilisés, l'un des moyens les plus simple était les tours de garde d'incendie où sont installés des personnes dont la charge est de chercher des incendies au moyen d'appareils spéciaux comme les jumelles, vu le manque de fiabilité des observations humaines d'autres méthodes se sont développées, tel que les systèmes de vidéo - surveillance automatique, réseau de capteur sans fil

I.5.2.1.Système de vidéosurveillance automatique

Dans ce système, les capteurs sont montés dans un emplacement avec vue sur une grande partie de la forêt, ils ont la capacité de balayer un espace circulaire de 10km²en moins de huit minutes, leur précision est largement affectée par les conditions climatiques comme les nuages, réflexion de la lumière et la fumée provenant des activités industrielles. Les images obtenues par les capteurs sont traitées afin de détecter la fumée, ceci est rendu possible par l'utilisation d'un algorithme qui prend en considération la condition atmosphérique.

Système de sondage acoustique radio (RASS) est utilisé pour obtenir la structure de l'atmosphère, cette approche nous donne des images sous forme des cartes thermiques des zones forestières, ainsi il permet la détection potentiel d'incendie. Les techniques de sondage acoustique radio ont une plus grande sensibilité aux changements des températures, elles peuvent fournir des mesures des températures de l'air plus précises sur des longues distances. Le processus de mesure de la température en utilisant des ondes sonores acoustiques est mis à disposition par un radar, qui fonctionne en transmettant des impulsions acoustiques dans l'atmosphère et la détection des échos de rétrodiffusion thermique atmosphérique. Cette technique est très efficace, fiable et elle surpasse toutes les autres techniques de détection

d'incendie sur le plan de l'exactitude. Pourtant, elle a encore plusieurs inconvénients et difficultés techniques qui sont :

- ❖ La stratégie de déploiement de ce système est essentielle pour les performances. Cela est dû au fait que les échos dans les données recueillies sont causés par l'encombrement au sol.
- ❖ Il est signalé que la zone de numérisation est d'être limitée à une certaine hauteur qui est juste au-dessus des arbres.
- ❖ La performance est sensible à la fréquence correspondant et à la puissance du signal. Les meilleurs résultats sont obtenus lorsque la fréquence et la puissance des signaux acoustiques sont déterminées, en tenant compte de plusieurs questions telles que la densité de surface forestière et la distance entre le radar et sources acoustiques.

I.5.2.2. Réseau de capteurs à fibre optique

Le système utilise un réseau de capteurs à fibre optique basé sur les dispositifs 'Fibre Bragg Gratin' (FBG), des capteurs, des analyseurs de gaz et de l'unité de pilotage optoélectronique pour la grille de capteurs à fibre optique.

Ce système est composé de capteurs de température FBG, un ensemble d'acquisition FBG, des capteurs à fibre optique distribués pour collecter la variation de température sur la fibre, les routes sur lesquelles on pose le câble à travers la forêt et les capteurs de gaz. Le nombre de capteurs et la longueur du câble à fibre optique dépendent de la nature de la forêt.

Les capteurs de température FBG mesurent les valeurs de température en présence d'un incendie. Les capteurs de gaz collectent des échantillons d'air, mesurent le contenu de CH₄ et signalent les résultats à leur unité d'acquisition respective. L'avantage de ce système est que l'effet de la variation de température sur la longueur d'onde de la fibre optique peut être exploité pour générer des capteurs de température ultra-stables. Ce système présente deux inconvénients principaux :

- ❖ Les câbles à fibres optiques doivent être installés dans toute la forêt sur un type particulier de route ou de chemin.

- ❖ Les feux prennent plus de temps à être détecté, en fonction de la hauteur des arbres
Puisque les câbles sont posés au niveau du sol.

I.5.2.3. Réseaux de capteurs sans fil

Une technologie de détection de feu de forêt, prometteuse a émergé ces dernières années. Elle est basée sur les réseaux de capteurs sans fil, des capteurs qui peuvent être placés directement sur les arbres, Ils ont pour rôle de surveiller la température ambiante, d'effectuer certains calculs et de collaborer pour transmettre les données via des liaisons sans fil à un destinataire principal qui se charge de les recueillir.

I.6. Conclusion

Nous nous sommes intéressés dans le cadre de ce travail, à un type de technologie visant la lutte contre les incendies de forêt malgré que le commencement de ce combat doit être bien en amont des départs de feux, c'est plutôt par une politique volontariste de prévention de ce désastre.

II.1. Introduction

Dans notre réalisation et conception, nous avons eu besoin de manipuler plusieurs composants, ce chapitre est consacré à la description de ces outils ainsi que leurs caractéristiques

II.2. Arduino

II.2.1. Définition d'Arduino

Arduino est une plate-forme de prototypage d'objet interactif à usage créatif constitué d'une carte électronique et d'un environnement de programmation.

Sans tout connaître ni comprendre l'électronique cet environnement matériel et logiciel permet au utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources et outilles.

Arduino permet d'étendre la capacité de relation humain/machine ou environnement/machine, c'est plutôt un pont tendu entre le monde réel et le monde numérique.

La carte Arduino repose sur un circuit intégré (appelé aussi mini-ordinateur ou microcontrôleur) associé avec des entrées et sorties qui permettent aux utilisateurs de brancher différents types d'éléments externes.

Entrées : des capteurs qui collectent des informations sur leurs environnements.

Sorties : des actionneurs qui agissent sur le monde physique.

II.2.2. Carte Arduino Mega

II.2.2.1. Définition

La carte Arduino Mega 2560 est basée sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 uarts. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que l'Uno. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.



Figure 04 : Carte Arduino Mega

II.2.2.2. Caractéristiques principales

- Version : Rev. 3
 - Alimentation:
 - via port USB ou
 - 7 à 12 V sur connecteur alim
 - Microprocesseur: ATmega2560
 - Mémoire flash: 256 KB
 - Mémoire SRAM: 8 KB
 - Mémoire EEPROM: 4 KB
 - 54 broches d'E/S dont 14 PWM
 - 16 entrées analogiques 10 bits
 - Intensité par E/S: 40 mA
 - Cadencement: 16 MHz
 - 3 ports série
 - Bus I2C et SPI
 - Gestion des interruptions
 - Fiche USB B
 - Dimensions: 107 x 53 x 15 mm
- Version d'origine fabriquée en Italie.

II.2.2.3. Architecture de la carte

Voici quelques détails utiles sur les interfaces d'une carte Mega 2560.

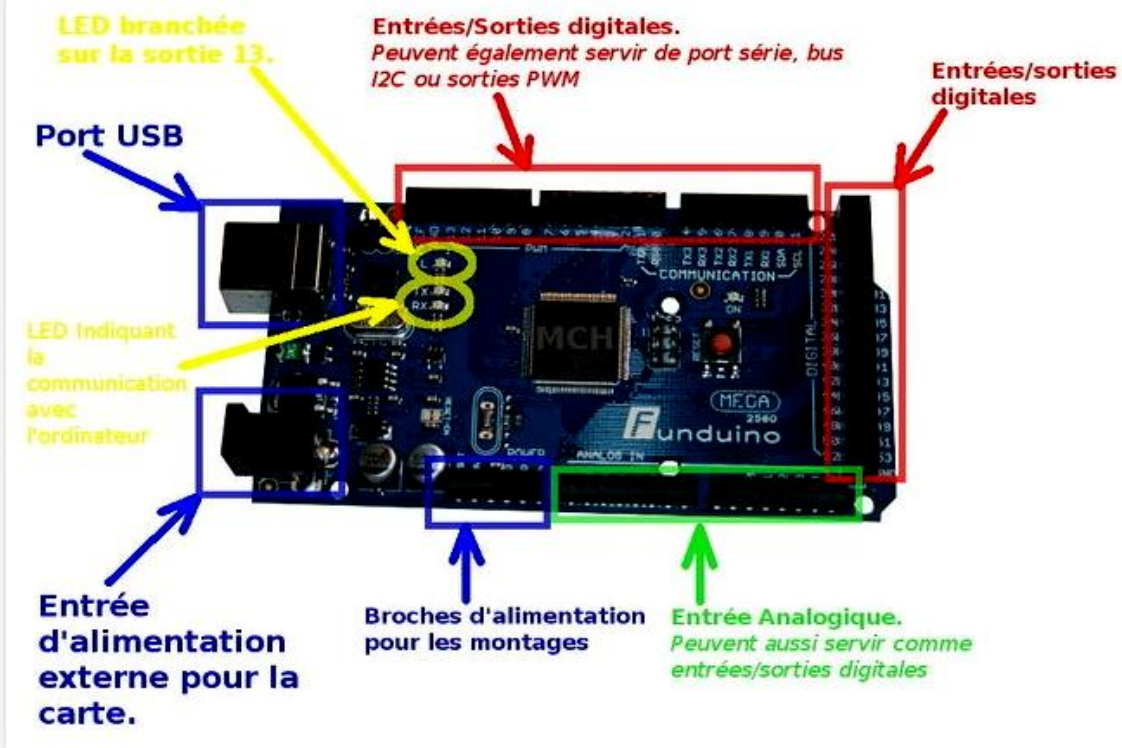


Figure 05 : Architecture interne de la carte Arduino Mega

II.2.2.4. Microcontrôleur ATmega 2560

Le microcontrôleur ATmega 2560 est un processeur informatique très puissant de 16 Mh capable d'exécuter 135 instructions différentes et en utilisant 32*8 registres d'usage universel de fonctionnement

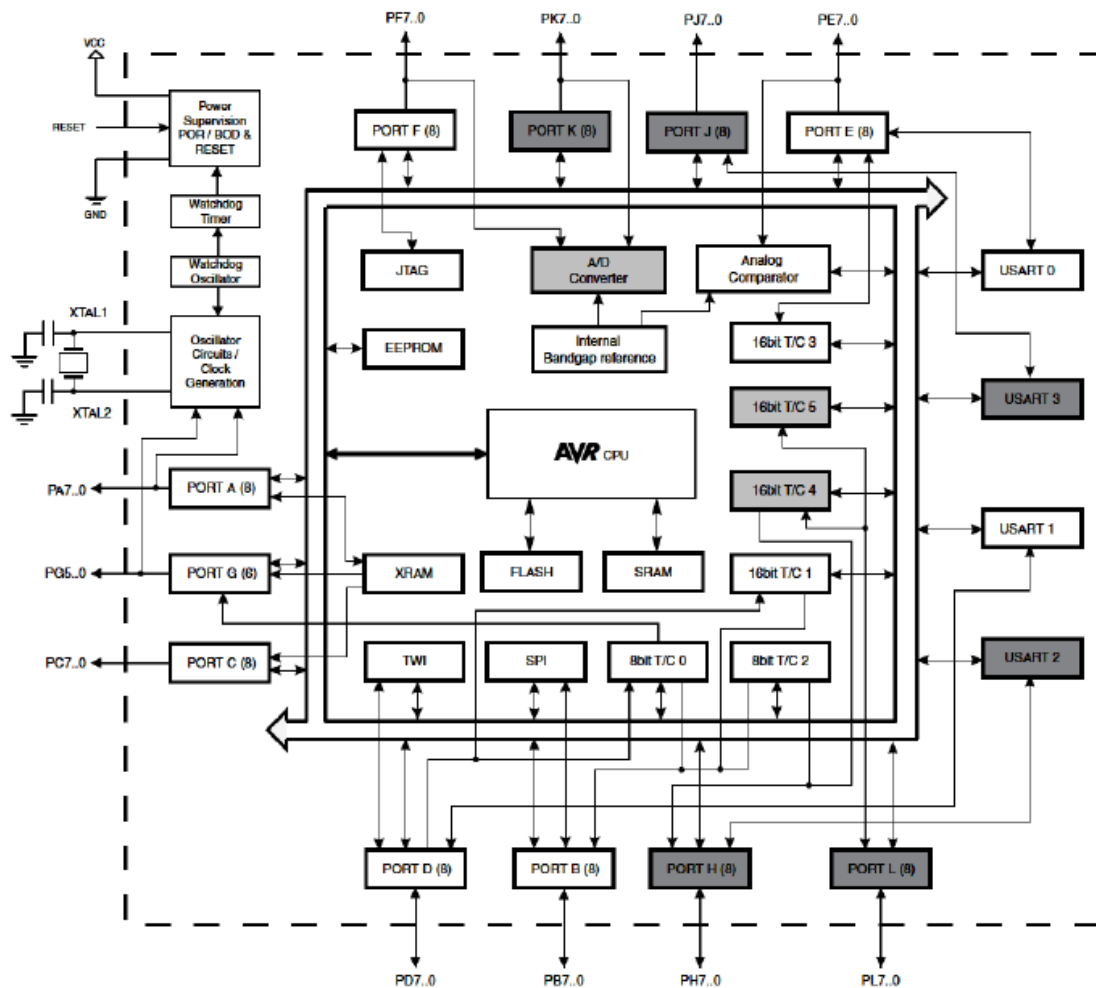


Figure 06 : Architecteur d'un microcontrôleur Mega 2560

L'ATmega 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le boot loader). L 'ATmega 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

Le boot loader est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l 'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

II.2.2.5. Alimentation de la carte Arduino

La carte Arduino Mega peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est sélectionnée automatiquement.

L'alimentation externe (non USB) peut provenir d'une alimentation sortie courant continu ou d'une batterie. L'adaptateur peut être connecté en branchant une fiche positive de 2,1 mm au centre dans la prise d'alimentation de la carte. Les fils provenant d'une batterie peuvent être insérés dans les connecteurs des broches Gnd et Vin du connecteur POWER. La carte peut fonctionner sur une alimentation externe de 6 à 20 volts.

En cas d'utilisation de plus de 12V, le régulateur de tension peut surchauffer et endommager la carte. La plage recommandée est de 7 à 12 volts.

Si la carte est alimentée avec moins de 7V, la broche 5V peut fournir moins de 5V et la carte peut être instable.

Les broches d'alimentation pour la carte Arduino Méga sont les suivantes :

- **VIN** : La tension d'entrée de la carte Arduino lorsqu'elle utilise une source d'alimentation externe (par opposition à 5 volts provenant de la connexion USB ou d'une autre source d'alimentation régulée). Vous pouvez fournir la tension à travers cette broche
- **5V** : C'est l'alimentation régulée utilisée pour alimenter le microcontrôleur et les autres composants de la carte. Cela peut provenir de VIN via un régulateur embarqué, ou être fourni par l'USB ou toute autre alimentation 5V régulée.
- **V3** : Une alimentation de 3,3 volts générée par la puce FTDI embarquée. La consommation de courant maximale est de 50 mA.
- **GND** : Masse de la carte ou 0v de référence

II.2.3. Environnement de programmation**II.2.3.1. Architecture software**

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en java inspiré du langage processing L'IDE permet d'écrire, de modifier un

programme et de le convertir en série d'instructions compatibles et compréhensibles par la carte Arduino.

II.2.3.2.Structures du programme IDE

➤ **Editeur**

C'est l'espace utilisé pour écrire le programme à téléverser vers la carte Arduino, il se dispose aussi des onglets de navigation

➤ **La zone de messages**

Cet espace est utilisé par l'IDE afin de communiquer avec l'utilisateur sur l'état de programme et les actions en cours de traitements (compilation, téléversement)

➤ **Une console texte**

Un espace où l'IDE affiche des messages concernant les résultats de compilation de Programme (les erreurs dans l'éditeur).

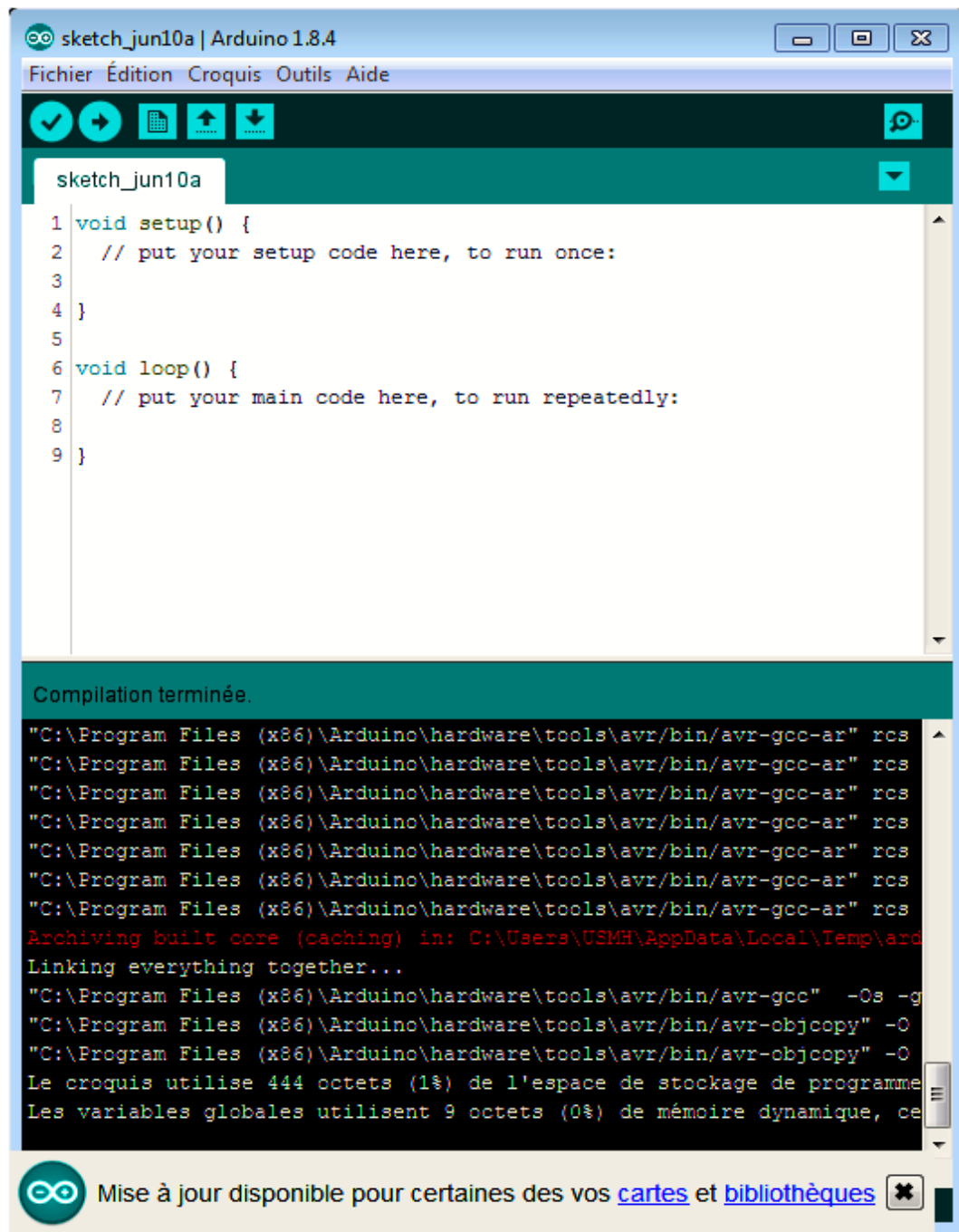


Figure 07 : Architecteur software

II.2.4.Exemple d'utilisation

Etant plateforme de logicielle et matérielle de création d'objets numérique, Arduino permet de programmer des circuits électroniques qui interagisse avec le milieu qui les entoure. Connecté notamment à des capteurs sonores, thermique de mouvement, ces circuits électroniques peu couteux, dés organismique contrôleurs peuvent en retour générer des images, actionner un bras articulé, envoyer des messages sur internet et. Des dizaines de millier d'artistes de designers, d'enseignants de chercheurs, d'ingénieurs et même des

entreprises utilisent Arduino pour concevoir et réaliser des projets incroyables dans de multiples domaines.

- Prototypage rapide des projets innovants utilisant l'électronique : Arduino facilite l'expérimentation avant la phase d'industrialisation.
- Production artisanale d'objets numériques et de machine-outil à faible coût dans la perspective d'une culture d'appropriation technologique favorisant le bricolage et la débrouille.
- Captation et analyse de données scientifiques à des fins éducatives de recherche ou d'appropriation citoyenne.
- Spectacle vivant, grâce aux nombreuses interactions offertes par Arduino il est possible de générer en temps réel des effets sonores et visuels dans un spectacle.
- Installation d'Arts numériques Arduino permet de réaliser des œuvres d'arts interagissant de manière autonome avec le public.
- Mode et design textile plusieurs stylistes et designers investissent ce domaine créatif en exploitant les possibilités offertes par l'intégration électronique notamment dans des vêtements.
- Projets pédagogiques : destiné aux étudiants aux professionnels ou au grand public selon les porteurs de ces initiatives : écoles supérieures, centres de formation spécialisés ou des médias libres.

II .3. Module GSMA6

II.3.1.Définition

Le module **GMS** est une carte d'interface compatible Arduino. Elle permet d'envoyer et recevoir des SMS, des données ou des communications vocales depuis le réseau mobile. Le module est basé sur le circuit A6 de la société Ai Thinker. Il est contrôlé via les commandes AT depuis une carte Arduino

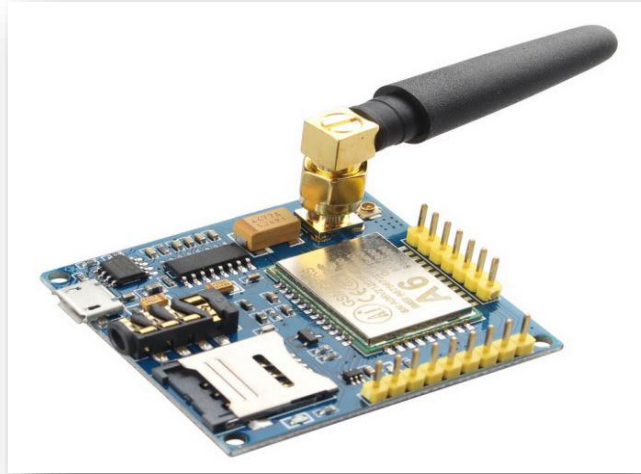


Figure 08 : Module GSM A6

Le module est livré avec une antenne patch déportée. Un connecteur de la platine est prévu pour recevoir une carte SIM

La communication entre le module et une carte Arduino est réalisée par la liaison série asynchrone : UART ou une liaison série logicielle

II.3.2. Caractéristiques du module A6

- Alimentation de 5 V
- Niveau de série de 2,8 V, compatible avec Arduino, Raspberry et d'autres microcontrôleurs
- Petite taille, adapté à diverses applications
- Interface casque intégrée, peut être utilisé pour effectuer un appel téléphonique

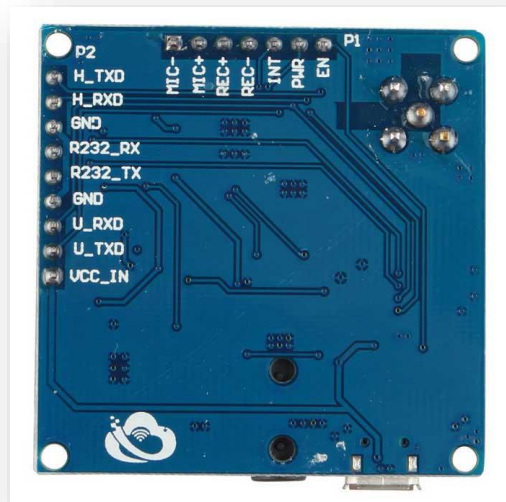


Figure 09 : Pins du module GSM A6

- Courant de veille : < 3 mA
- Tension de fonctionnement : 3.3V-4.2V
- Tension d'alimentation : >3.4V ;
- Interface : UART (TTL)
- Température nominale : -30 °C ~ +80 °C
- Bande GSM/GPRS : 850/900/1800/1900 MHz
- Sensibilité : < -105 dBm
- Classe GPRS : 10
- Prend en charge DES APPELES VOCAUX
- Prend en charge les SMS
- Prend en charge la communication des données GPRS, le débit de données maximal : téléchargement de 85.6Kbps, mise en ligne (upload) de 42.8Kbps
- Prend en charge les commandes AT
- Prend en charge le support audio numérique et audio analogique, ainsi que le codage vocal HR, FR, EFR et AMR
- Liaison série
- Le choix des broches permettant la communication entre leshield GPRS et la carte Arduino via la liaison série est réalisé via deux cavaliers tx rx.
- La liaison série doit être réglée avec une vitesse de 19200 bits/s.

II. 3.3. Commandes AT sur Arduino

II.3.3.1. Présentation

- Les commandes AT sont un jeu de commandes textuelles permettant de gérer la plupart des modems
- Ou des modules GSM. Ces commandes commencent toujours par les lettres «AT» et se terminent obligatoirement par un retour chariot.

II.3.3.2. Configuration de la liaison série logicielle

L'Atmega2560 possède une interface de communication série UART accessible, grâce aux broches numériques 0 (Rx) et 1 (Tx). La bibliothèque « Software Serial » a été développée pour permettre la communication série sur d'autres broches numériques de l'Arduino de manière logicielle. Il est possible de gérer plusieurs ports séries logiciels avec des vitesses allant jusqu'à 19200 bps cependant un seul peut recevoir des données à la fois.

Inclure la bibliothèque « SoftwareSerial »

Pour inclure la librairie « SerialSoftware » dans un programme, on ajoutera au début du programme la ligne suivante : `#include <SoftwareSerial.h>`

Créer une liaison série logicielle

`SoftwareSerial.Liaison (Broche :Rx, Broche :Tx)`

Fonction « begin »

`Nom_Liaison.begin(Vitesse) ;`

Cette fonction qui doit être appelée au moins une fois, généralement dans la fonction `setup()`, permet de définir la vitesse utilisée par la liaison série.

La valeur prise par la variable « Vitesse » doit être une des vitesses définies par la norme RS232.

Entrer le code PIN

Pour entrer le code il faut envoyer la commande AT suivante :

`AT+CPIN = "XXXX"`

Si le code PIN est correct, le module GSM répondra à cette commande par « +CPIN: READY ».

```
MyGSM.println("AT+CPIN = \"1234\"); // Code PIN
```

Envoyer un message texte (SMS) Mode Texte

Le module GPRS peut envoyer des SMS selon deux modes : le mode texte et le mode PDU (binaire). Pour envoyer un message lisible, il faut sélectionner le mode texte en envoyant la commande AT :

```
AT+CMGF = 1
```

Le module répondra à cette commande par «OK ».

Numéro destinataire SMS

Il faut ensuite indiquer le numéro du téléphone du destinataire du SMS. Pour cela il faut utiliser la commande AT ci-dessous. Le numéro doit être indiqué au format E.123.

```
AT+CMGS = "+XXXXXXXXXXXX" MyGSM.println("AT+CMGS = \"+3368825263411\");
```

Envoi texte message

Il suffit ensuite d'envoyer le texte du message. La dernière ligne du message doit comporter un retour chariot. Le message se termine par la séquence d'échappement CTRL-Z (caractère 26).

```
MyGSM.println("MESSAGE ?");
```

```
MyGSM.print((char) 26);
```

II.4. Capteur MQ-7

II.4.1.Introduction

La famille MQ-X sont des capteurs de différents gaz ils émettent une tension analogique en fonction de la concentration du gaz dans l'air

Voilà quelques-uns avec chacun son gaz à mesurer

MQ – 2 : Butane GPL gaz de fumée de gaz hydrogène capteur pour Arduino Module

MQ – 3 : « alcool, l'éthanol haleine » capteur de détection de gaz de l'éthanol pour Arduino

MQ – 4 : Gas Sensor charbon naturel CO Détecteur de fuites de méthane pour Arduino Module

MQ – 5 : méthane Sensor-Shield Détecteur de fuites de méthane Module pour Arduino

Mega2560

MQ – 6 : GPL gaz Propane Isobutane-Sensor capteur pour Arduino Raspberry

MQ – 7 : monoxyde de carbone CO gaz Module de capteur pour Arduino UNO Méga 2560A123.

MQ – 8 :Module de capteur pour Arduino Module gaz (hydrogène)

MQ – 9 : capteur de détection de gaz combustibles Détecteur de Monoxyde de CarboneàModule

MQ – 135 : capteur de capteur de qualite de l'air dangereux Module de détection de gaz pour Arduino

- Le capteur que nous utiliserons est le capteur MQ-7. C'est un capteur sensible aux effets du CO. (monoxyde de carbone)

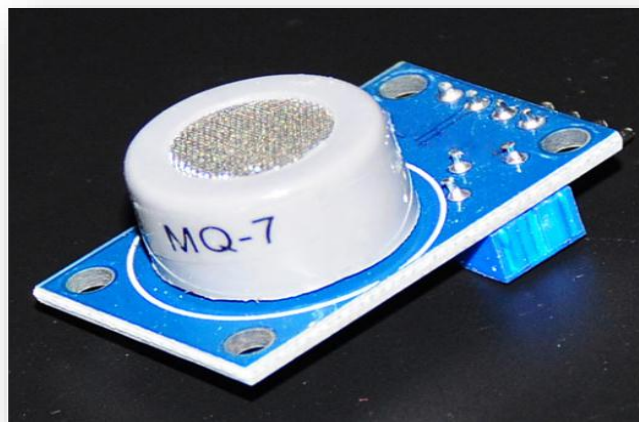


Figure 10 : Capteur MQ-7

II.4.2.Définition

Le MQ7 est un capteur de monoxyde de carbone (CO) simple à utiliser adapté à la détection de concentrations de CO dans l'air. Il peut détecter partout des concentrations de monoxyde de carbone (CO) de 20 à 2 000 ppm. La sortie du capteur est une résistance analogique

Il dispose d'une sortie analogique et d'un réglage de la sensibilité par potentiomètre

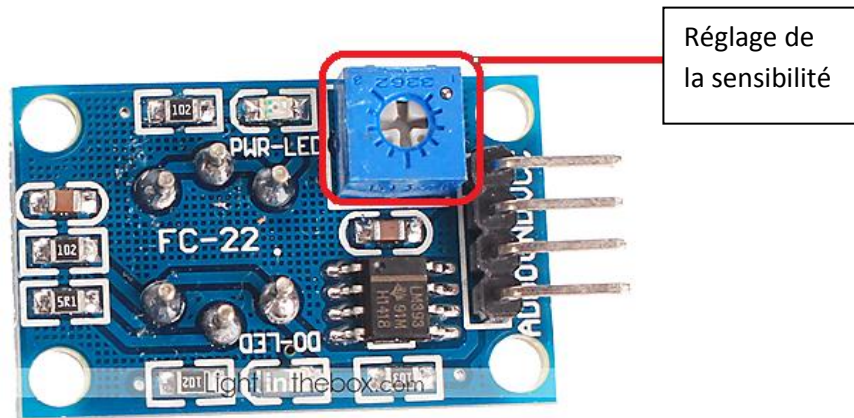


Figure 11 : Réglage de sensibilités du MQ-7

II.4.3. Caractéristiques

- Tension de fonctionnement : DC 5 V
- Driver IC : LM393, ZYMQ-7
- Taille : 11x33x20mm
- Plage de détection : 10 à 1000 ppm
- Gaz caractéristiques : 100 ppm CO
- Résistance sensible : 2 KOhm à 20 KOhm dans 100ppm CO
- Sensibilité : $\geq 3\%$
- Temps de réponse : ≤ 1 s
- Temps de récupération : ≤ 30 s
- Chauffage résistance : $31\text{ohm} \pm 3\text{ohm}$
- Courant de chauffage : $\leq 180\text{mA}$
- Tension de chauffage : $5.0\text{V} \pm 2$ V/ 1.5 ± 1 V
- Puissance de chauffage : environ 350 mW
- Température ambiante : -20°C à $+50^\circ\text{C}$
- Humidité : $\leq 95\%$ HR
- Teneur En oxygène : 21%
- Applications : Principalement utilisé pour CO, de détection de gaz en famille, usine ou exploitation souterraine

II.5. Capteur de flamme

II.5.1. Définition

Le Capteur de Flamme IR est un capteur utilisé pour détecter la présence d'un feu ou de toute autre source infrarouge (une flamme ou source lumineuse d'une longueur d'onde dans la plage des 760 à 1 100 nm peut être détectée). Il peut être utilisé dans un robot de lutte contre l'incendie ou un robot à tête chercheuse thermique.

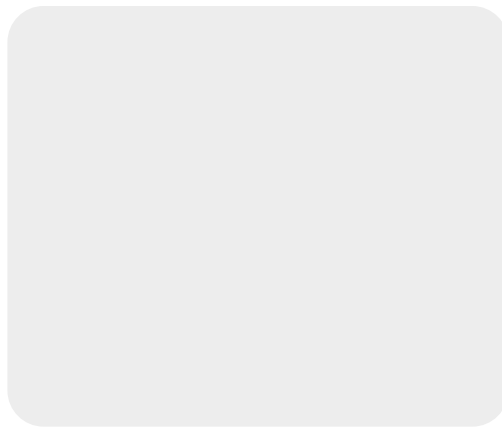


Figure 12 : Capteur de flamme

II.5.2. Caractéristiques

- ❖ Peut détecter la flamme ou la longueur d'onde à 760nm à 1100nm gamme de la source de lumière.
- ❖ Le test flamme briquets distance est 80 cm, la plus grande la flamme, la plus grande est la distance test. Point de détection d'environ 60 degrés, particulièrement sensible à la flamme spectre.
- ❖ Sensibilité est réglable, une performance stable.
- ❖ La sortie du comparateur signal propre forme d'onde est bonne, la capacité de conduire, que 15mA. avec une précision potentiomètre réglable réglage de la sensibilité.
- ❖ Un trou de boulon fixe pour une installation facile.
- ❖ Forme de sortie : sorties de commutation numérique (0 et 1). En utilisant une large tension de LM393 comparateur.

II.5.3. Spécifications

- ❖ Spectregamme : 760nm ~ 1100nm
- ❖ Angle de détection : 0-60 degrés
- ❖ Puissance : 3.3 V ~ 5.3 V

II.6 BUZZER



Figure 13 : Module buzzer

Une buzzer est un élément électromécanique ou électronique qui produit un son quand on lui applique une tension. Certains nécessitent une tension continue (buzzers électromécaniques), d'autres nécessitent une tension alternative (transducteurs piézo-électrique).

II.7. Panneaux solaires photovoltaïque



Figure 14 : Panneaux solaires photovoltaïque

II.7.1. Générateur Photovoltaïque

L'énergie solaire photovoltaïque provient de la transformation directe d'une partie du rayonnement solaire en énergie électrique. Cette conversion d'énergie s'effectue par le biais d'une cellule dite photovoltaïque basée sur un phénomène physique appelé effet photovoltaïque qui consiste à produire une force électromotrice lorsque la surface de cette cellule est exposée à la lumière

La cellule photovoltaïque élémentaire constitue un générateur de très faible puissance vis-à-vis des besoins de la plupart des applications domestiques ou industrielles.

Une cellule élémentaire de quelques dizaines de centimètres carrés délivre, au maximum, quelques watts sous une tension inférieure au un volt .Pour produire plus de puissance, plusieurs cellules doivent être assemblées afin de créer un module ou un champ photovoltaïque. La connexion en série des cellules permet d'augmenter facilement la tension de l'ensemble, tandis que la mise en parallèle permet d'accroître le courant. Le câblage série/parallèle est donc utilisé pour obtenir globalement un générateur PV aux caractéristiques souhaitées Une cellule

II.7.2. Cellule photovoltaïque

Une cellule photovoltaïque est une composante optoélectronique qui transforme directement la lumière solaire en électricité

Une cellule photovoltaïque est constituée par un matériau semi-conducteur de type P-N. La taille de chaque cellule va de quelques centimètres carrés jusqu'à 100 ou plus sa forme est circulaire, carrée ou dérivée des deux géométries.

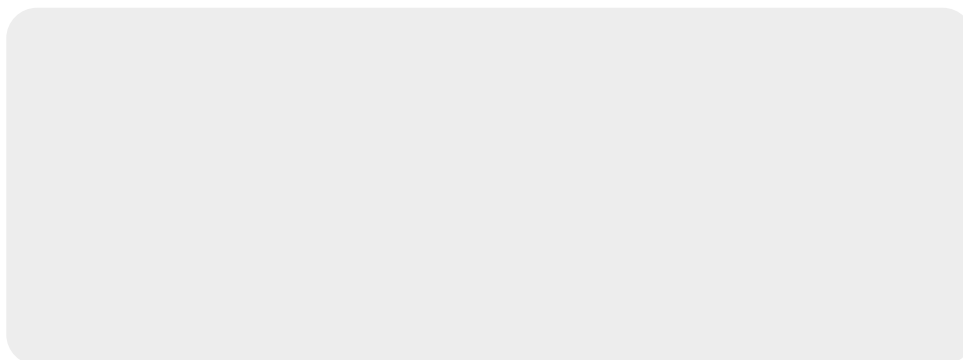


Figure15 : Schéma d'une cellule photovoltaïque

II.7.3. Module photovoltaïque

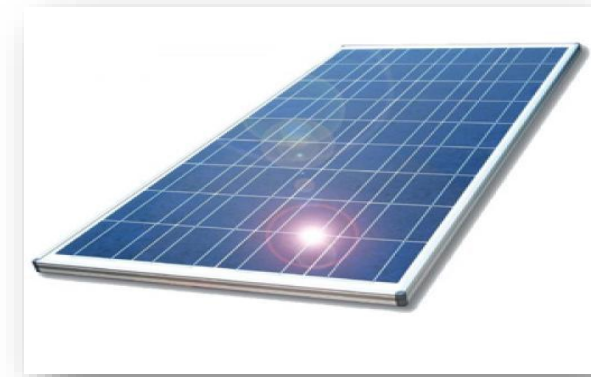


Figure16 : Module photovoltaïque

Le composant le plus crucial de toute installation PV est le module photovoltaïque, qui se compose de cellules solaires interconnectées. Ces modules sont raccordés entre eux pour former des champs de manière à pouvoir satisfaire différents niveaux de besoins en énergie. Des modules de plus en plus puissants sont disponibles sur le marché, en particulier pour la connexion du réseau, mais il y'a tout de même une limite liée au poids et à la manipulation.

II.7.4. Champ photovoltaïque



Figure17 : Champ photovoltaïque

Le champ photovoltaïque se compose de modules photovoltaïques interconnectés en série et/ou en parallèle afin de produire la puissance requise. Ces modules sont montés sur une armature métallique qui permet de supporter le champ solaire avec un angle d'inclinaison spécifique.

II.8. Conclusion

La connaissance profonde des caractéristiques des composants et leurs comportements durant le fonctionnement sous les différentes variations des substances en milieu d'utilisation est très nécessaire avant toute manipulation en pratique

III.1. Introduction

Dans ce chapitre nous allons designer les connections et le brochage entre les différents composants tout en donnant leurs programmes d'initialisation c'est-à-dire, les programmes qui définissent les entrées et les sortie et leurs états.

III.2. Module GSM

III.2.1. Alimentations du module GSM

Le module GSM étant un composant qui fonctionne avec une tension continue de 5V, on a choisi de l'alimenter par la sortie d'Arduino Mega (pin 5V), cette sortie qui émet 5V par défaut tend que l'Arduino est allumé.

La masse du GSM sera aussi connecté à la masse de l'Arduino

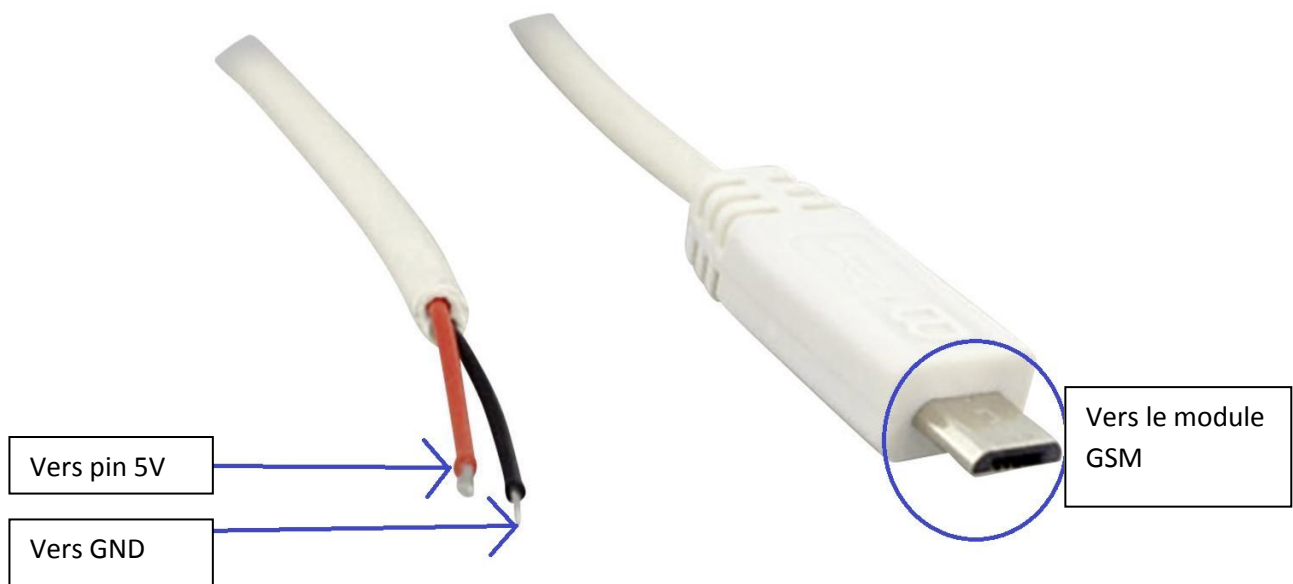


Figure 18 : Alimentation du GSM

III.2.2. Câblages du module GSM

Avant de commencer le brochage de transmission et de réception avec Arduino le module GSM A6 nécessite un brochage interne entre VCC_IN et PWR, ce brochage est nécessaire au début, et on peut l'enlever quelques secondes après le lancement de notre système.

Et une autre GND qui sera branchée a la GND de l'Arduino

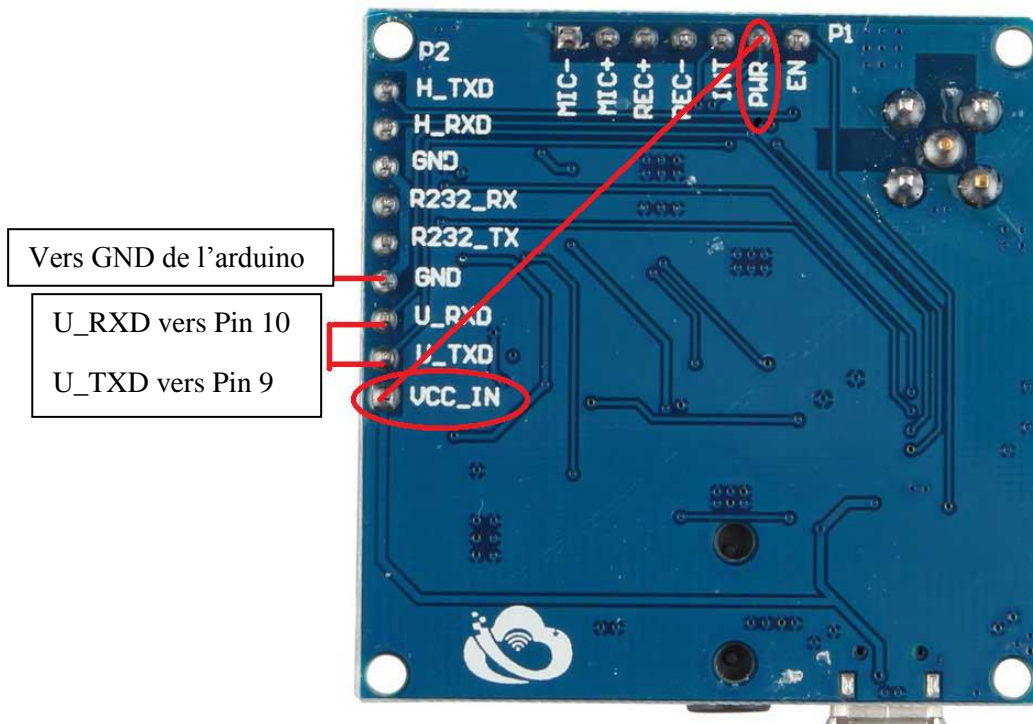


Figure 19 : Câblage de module GSM

La sortie de transmission (TXD) du module GSM sera connectée au pin digital Pin9 qui sera défini dans le programme comme une entrée de réception.

L'entrée de réception (RXD) du module GSM sera connectée au pin digital Pin10 qui sera défini dans le programme comme une sortie de transmission.

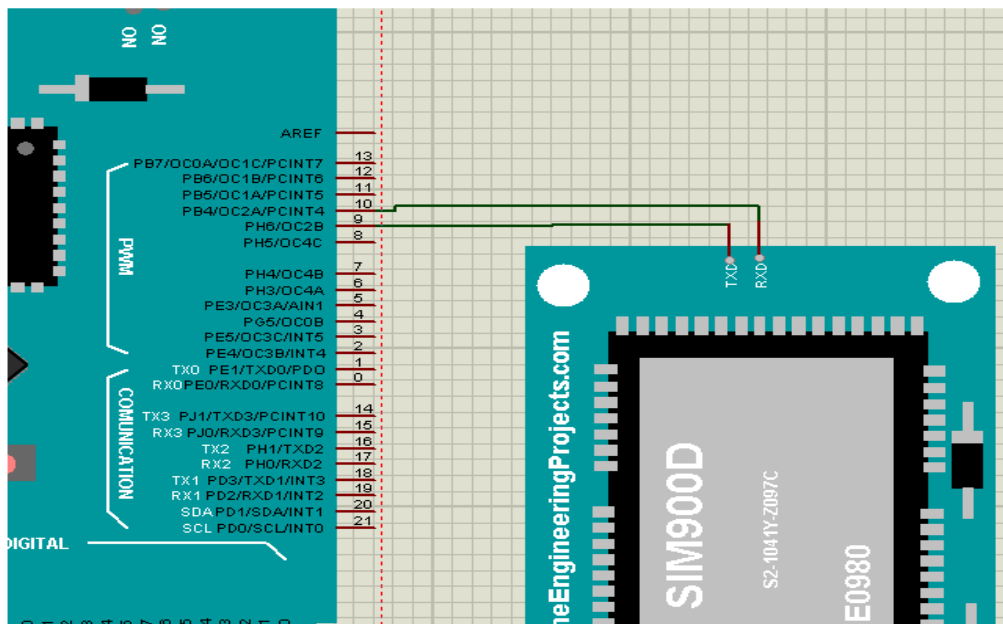


Figure 20 : Brochage du module GSM

III.2.3. Programme d'initialisation

```
#include <SoftwareSerial.h> // inclure la bibliothèque <SoftwareSerial.h>

SoftwareSerial mySerial (9, 10); // définir les PINs 9 et 10 consécutivement
//comme entrée et sortie de liaison série

Void setup ()

{

mySerial. begin (19200); //régler la vitesse de communication avec le GSM en
19200 bit/s

}
```

III.3. Câblages des capteurs

III.3.1. MQ7

III.3.1.1. Câblage du capteur mq7

Le capteur MQ7 sera connecté à l'Arduino de la manière suivante

- VCC vers A3
- OUT vers A0
- La masse GND du capteur vers la GND de l'Arduino

La PIN A0 sera configurée en entrée pour recevoir la donnée du capteur MQ7

La PIN A3 sera configurée en sortie et émet 5V dès le lancement de notre système pour alimenter le capteur.

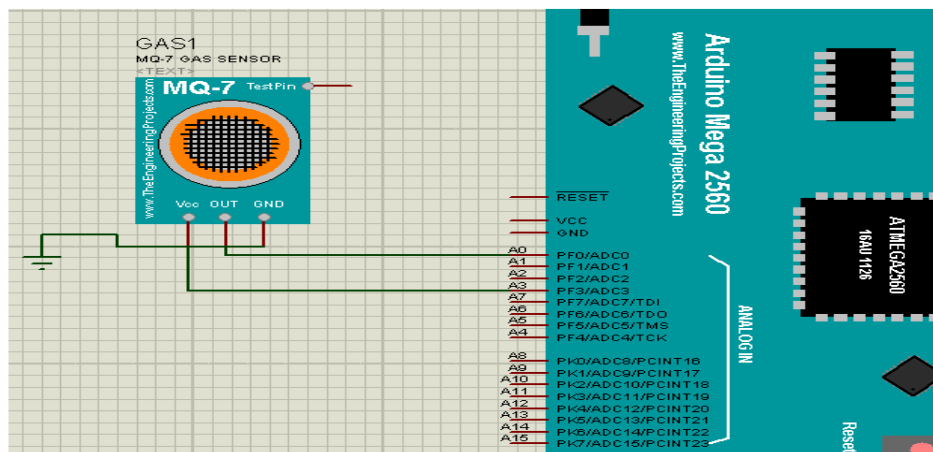


Figure 21 : Brochage du capteur MQ-7

III.3.1.2. Programme associé

```
int inputcapfume = A0; //sélectionner une entrée pour VIN du mq7
```

```
int powercapfume = A3; //sélectionner une sortie pour Vcc du mq7
```

```
void setup ()
```

```
{
```

```

pinMode (inputcapfumees, INPUT); //configurer une pin en entrée A0

pinMode (powercapfumees,OUTPUT); //configurer une pin en sortie A3

digitalWrite(powercapfumees,HIGH); //donner un état haut a une sortie A3

}

```

III.3.2. Capteur de flamme

III.3.2.1. Câblage du capteur de flamme

Le capteur de flamme sera connecté à l'Arduino de la manière suivante

- VCC vers A2
- OUT vers A1
- La masse GND vers la GND de l'Arduino

La PIN A1 sera configurée en entrée pour recevoir la donnée du capteur de flamme

La PIN A2 sera configurée en sortie et émet 5V du lancement de notre système pour alimenter le capteur.

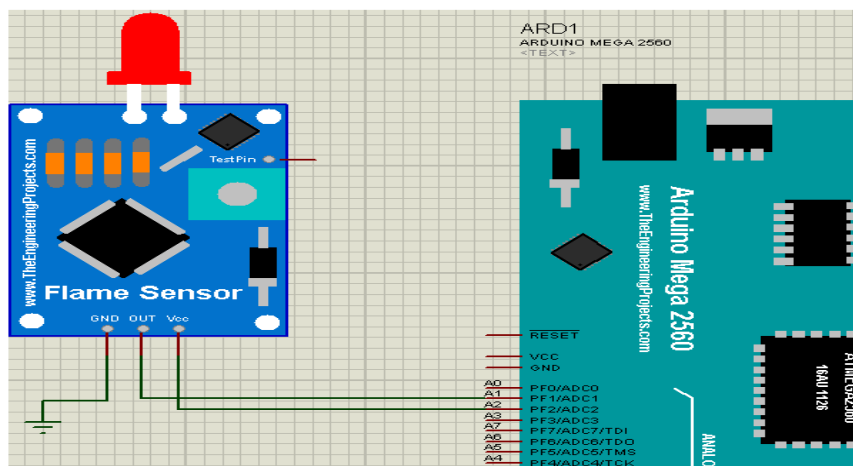


Figure 22 : Brochage de capteur de flamme

III.3.2.2. Programme associé

```
int inputcapflamme = A1; //sélectionner une entrée pour VIN du cap de flamme

int powercapflamme = A2; //sélectionner une sortie pour VCC du cap de flamme

void setup()

{

pinMode(inputcapflamme , INPUT); //configurer une pin A1 en entrée

pinMode(powercapflamme,OUTPUT); //configurer une pin A2 en sortie

digitalWrite(powercapflamme,HIGH); //donner un état haut a une sortie A2

}
```

III.4. L'afficheur LCD**III.4.1. Câblage**

VDD vers le pin digital (13) qui sera défini dans le programme comme une sortie de l'alimentation de l'afficheur LCD

R/W et VSS seront branchées vers la masse de l'Arduino

D7, D6, D5, D4, E et RS seront branchées au PINs digital 2, 3, 4, 5, 11,12 de manière consécutive.

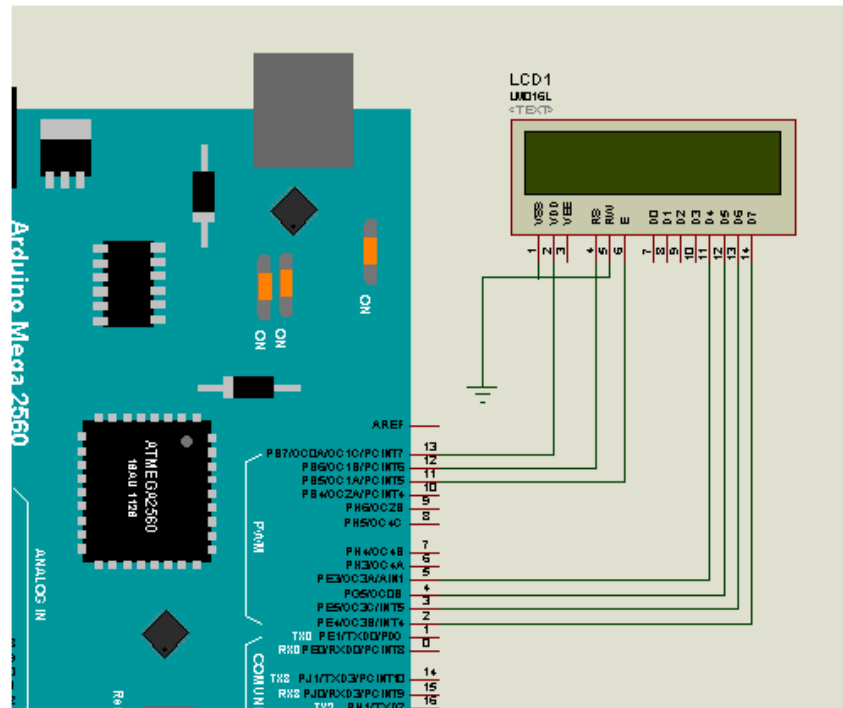


Figure 23 : Brochage de l'afficheur LCD

III.4.2. Programme associé

```
#include<LiquidCrystal.h> //inclure la bibliothèque<LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // définir les liaisons avec le LCD

int powerlcd = 13; // sélectionner une sortie pour Vcc du LCD

void setup () {

pin Mode (powerlcd,OUTPUT); //configurer une pin (13) en sortie

digitalWrite (powerlcd,HIGH); //donner un état haut a une sortie

lcd.begin(16,2); //initialisation du LCD

lcd.setCursor(0,0); //initialisation de la 1ere ligne du LCD

lcd.print ("Fire Scan - ON"); // afficher un message

}
```

III.5. Câblage des LEDs

III.5.1. Câblage

La LED vert indiquant le fonctionnement de notre system (ledfonctnmt) sera bronché vers la Pin A4

La LED jaune indiquant la présence du fumé (ledfumee) sera bronché vers la Pin A5

La LED rouge indiquant la présence de flamme (ledflamme) sera bronché vers la Pin A6;

Les PINS A4, A5 et A6 seront configurés dans le programme en sortie pour alimenter Les LEDs

Les LEDs seront aussi suivit de petites résistances (222 Ω) avant d'êtres connectés a la GND de l'Arduino.

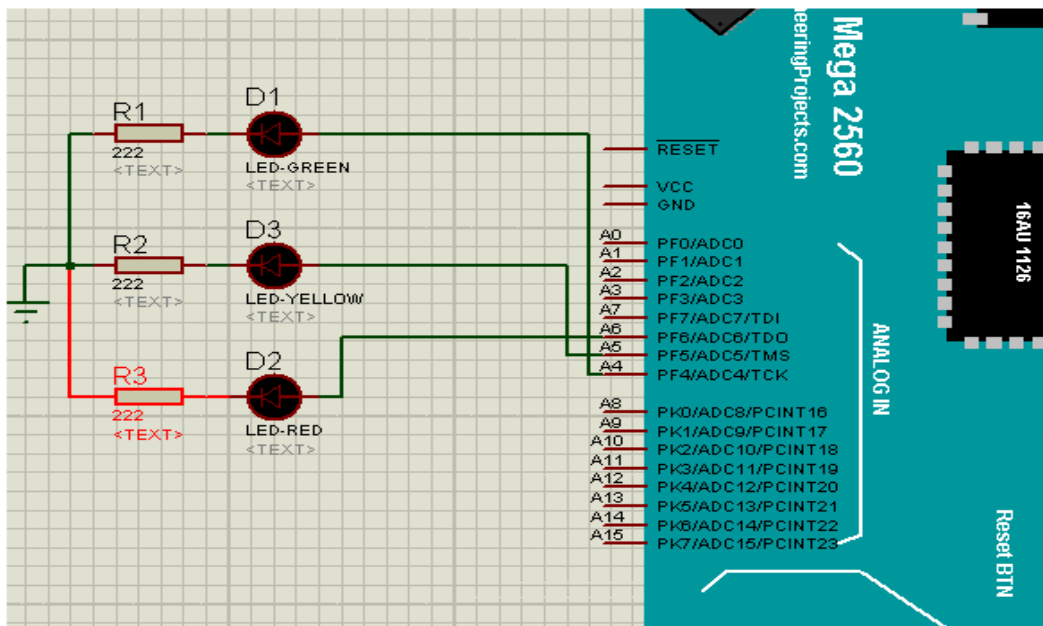


Figure 24 : Brochage des LEDs

III.5.2. Programme associé

```
int ledfonctnmt = A4; //selectionner une sortie pour LED de fonctionnemnt
```

```
int ledfumee = A5; //selectionner une sortie pour LED de fumé
```

```

int ledflamme = A6; //selectionner une sortie pour LED de flamme

void setup()

{

pinMode(ledfonctnmt,OUTPUT); //configurer une pin (A4) en sortie

pinMode(ledflamme,OUTPUT); //configurer une pin (A6) en sortie

pinMode(ledfume,OUTPUT); //configurer une pin (A5) en sortie

digitalWrite(ledfonctnmt,HIGH); //donner un état haut a une sortie

}
    
```

III.6. Amorçage des capteurs

Dans la pratique l’amorçage des capteurs sera par la présence du fumé pour le capteur MQ7, et par la présence de flamme pour le capteur de flamme.

Dans ISIS Proteus on a utilisé des LOGICSTATs pour amorcer les capteurs

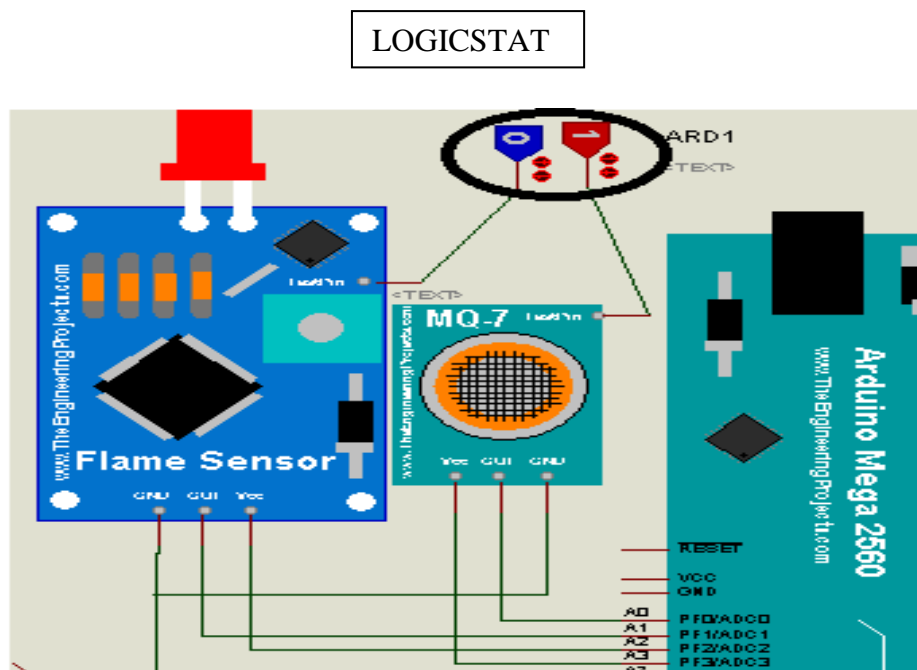


Figure 25 : Amorçage des capteurs

La logicstat donne soit 1 (5V) pour activer le capteur , ce qui nous donne une sortie de capteur VOUT non nulle , soit 0 (0V) pour le desactiver, et on recoit VOUT =0V.

III.7. Shemat complet sur ISIS Proteus

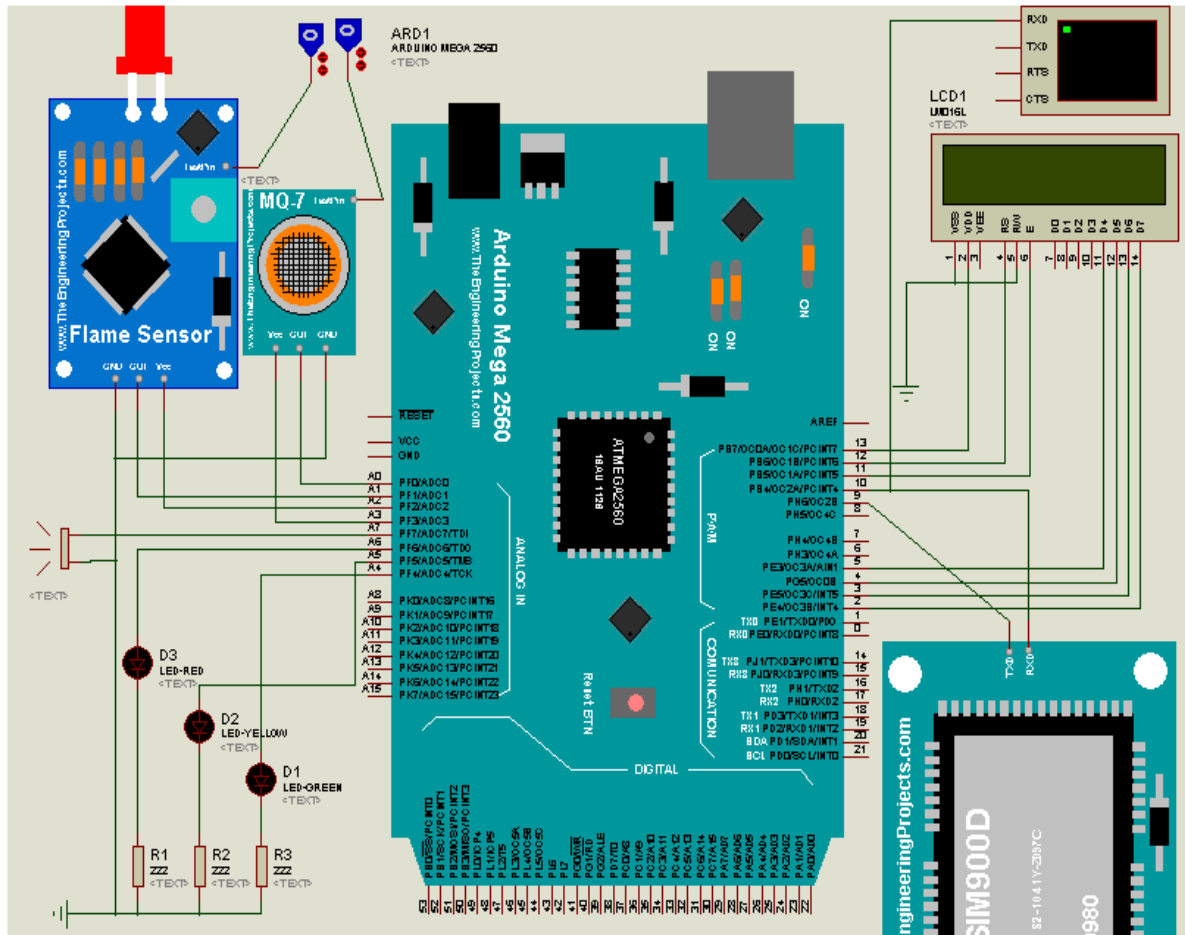


Figure 26 : Schéma complet sur issis proteus

III.8. Alimentation de l'Arduino

Le module Arduino sera alimenter par une batterie de 9Volts ou association de plusieurs piles, cette batterie sera chargée par un panneau solaire.

On peut aussi l'alimenter via un module solaire d'Arduino, ce module étant un régulateur solaire et chargeur de batterie possède aussi une option de basculement en

VOUT (soit alimentation directe par panneau soit par batterie en cas d'absence de lumière).

La figure ci-dessous représente un module solaire d'Arduino qui est l'intermédiaire entre Arduino et entre le panneau et la batterie.

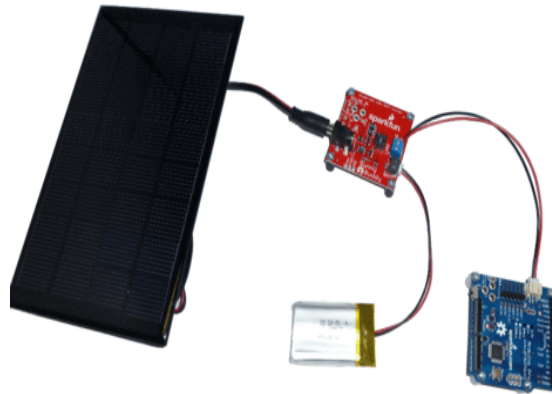


Figure 27 : Alimentation solaire de l'Arduino

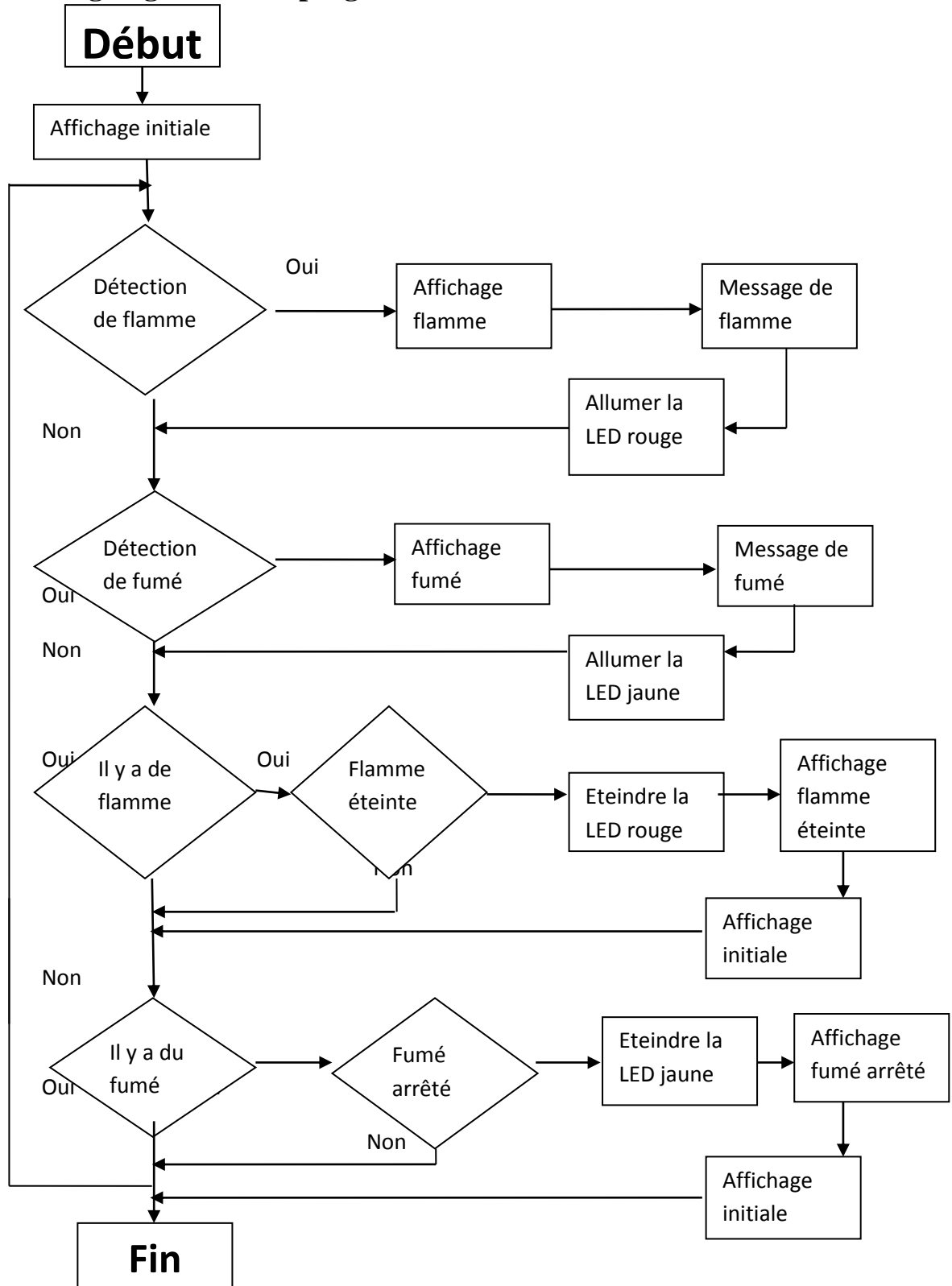
III.9. Conclusion

Le bon câblage et brochage entre les différents composants doit être très précis avant de mettre en marche le système pour avoir un bon fonctionnement et même pour la protection de ces outils

IV.1. Introduction

Dans ce chapitre nous allons présenter le comportement du system durant l'existence de plusieurs variations des substances (présences de flamme ou fumé, les deux ou rien du tout).

IV.2. Organigramme du programme



IV.3. Fonctionnement sans amorçage des capteurs

C'est le fonctionnement lors de l'allumage ou sans détection d'aucune substances (ni flamme, ni fumée), en cet état normale l'afficheur LCD affiche le message suivant «mode scan –ON »

La LED verte est allumé et deux autres (jaune et rouge) indiquant la détection de fumée ou flamme sont éteintes.

Le module GSM A6 est allumé mais, il n'envoie aucun message.

IV.4. Fonctionnement en amorçage d'un des capteurs**IV.4.1. Amorçages d'un des capteurs**

On cas de détection de flamme ou de fumée (détection d'un seul capteur) le system allume une LED et affiche sur le LCD un message spécial, il envoie aussi un message aux numéros indiqués dans le programme.

La LED allumée, le message affiché sur l'afficheur LCD et le SMS envoyé correspondent au type de détection (soit flamme soit fumée,).

Le module GSM envoie le message chaque loop (boucle) tant qu'il ne dépasse pas le nombre maximal des SMS indiqué dans le programme.

On peut envoyer de déferents messages vers de déferents numéros selon le programme

IV.4.2. Arrêt d'amorçage

Lors d'un amorçage d'un des capteurs, une variable initialisée à zéro (0) reçoit une valeur d'un(1), ce qui nous permet d'accéder plus tard à une fonction qui cherche si la sortie du capteur est devenue nulle (arrêt d'amorçage).

Si cette fonction trouve que la sortie de capteur est égale à zéro volts (tout à coup, absence de ou de flamme) la LED allumée sera éteinte et l'afficheur LCD affiche un autre message indiquant l'arrêt d'incendie, de l'autre côté le module GSM n'envoie plus de message.

Sans la détection d'une substance (flamme ou fumée), on ne peut pas accéder à la fonction qui détecte si cette substance détecté est arrêtée.

IV.5. Fonctionnement en amorçage des deux capteurs

IV.5.1. Amorçage des deux capteurs

A l'instant où il y aura une présence de fumée et de flamme à la fois, le system exécute d'abord le sous-programme correspondant à la première détection, dès la fin d'exécution du ce programme, c'est le tour au 2eme sous-programme correspondant à la deuxième détection qui vas s'exécuter.

IV.5.2. Arrêt d'amorçage d'un seul capteur

Dès qu'une des substances disparaisse, le capteur correspondant cesse d'émettre un signal par VOUT.

Le système éteint la LED correspondante (la LED allumée lors de la détection) et affiche la disparissions de substance en question sur le LCD, Le système ne vas plus envoyer de message concernant cette substance.

Après quelques secondes, c'est programme de détection de l'autre substance qui va tourner en boucle.

IV.6. Programme de fonctionnement

Le programme de fonctionnement muni de la signification de chaque ligne

IV.6.1. Initialisation et déclaration de variable

Cette partie est utilisée pour appeler diverses bibliothèques nécessaires pour l'utilisation et la communication avec les composants manipulés, ainsi que, pour déclarer les variables globales qui seront utilisées dans les différentes fonctions.

```
#include<SoftwareSerial.h> ; // include la bibliothèque <SoftwareSerial.h>  
//pour pouvoir utiliser les PINs de l'arduino comme entrée et sortie de liaison serie  
  
SoftwareSerialmySerial(9, 10); // définir les liaison série avec GSM
```

```
#include<LiquidCrystal.h>           //inclure la bibliothèque<LiquidCrystal.h>
//pour pouvoir utiliser l'afficheur LCD

LiquidCrystal LCD (12, 11, 5, 4, 3, 2);    // définir les liaisons avec le LCD

intinputcapfumees = A0;                //sélectionner une entrée pour VIN du mq7

intinputcapflamme = A1;                //sélectionner une entrée pour VIN du cap de
flamme

intpowercapflamme = A2;                //sélectionner une sortie pour VCC du cap de
flamme

intpowercapfumees = A3;                //sélectionner une sortie pour Vcc du mq7

intpowerlcd = 13;                      // sélectionner une sortie pour Vcc du LCD

intledfonctnmt = A4;                  //sélectionner une sortie pour LED de fonctionnement

intledfumees = A5;                    //sélectionner une sortie pour LED de fumée

intledflamme = A6;                    //sélectionner une sortie pour LED de flamme

int alarme = A7;                       //sélectionner une sortie pour le buzzer

intSMSfumees = 0;                      // déclaration et initialisation d'une variable a 0

intSMSflamme = 0;                      //déclaration et initialisation d'une variable a 0

intfumees_Set = 0;                     //déclaration et initialisation d'une variable a 0

intflamme_Set = 0;                     //déclaration et initialisation d'une variable a 0

floatconstseuil = HIGH;                //déclaration d'une const en état haut

floatconstseuil = LOW;                 //déclaration d'une const en état bas

float a; float b; float c; float d;    //déclaration de variables de type float
```

IV.6.2. La fonction void setup ()

La fonction void setup() est utilisé pour la configuration des PINs en entrées ou en sorties et initialiser quelques sorties en leurs donnant un état haut. Dans cette fonction on règle aussi la vitesse de communication de l'Arduino avec les composants, elle sera exécutée qu'une seule fois au début.

```
void setup ()                                //définir la fonction setup()

{

pinMode(inputcapfumees, INPUT);              //configurer une pin en entrée

pinMode(inputcapflamme , INPUT);            //configurer une pin en entrée

pinMode(powercapflamme,OUTPUT);             //configurer une pin en sortie

pinMode(powercapfumees,OUTPUT);            //configurer une pin en sortie

pinMode(powerlcd,OUTPUT);                  //configurer une pin en sortie

pinMode(ledfonctnmt,OUTPUT);               //configurer une pin en sortie

pinMode(ledflamme,OUTPUT);                 //configurer une pin en sortie

pinMode(ledfumees,OUTPUT);                 //configurer une pin en sortie

pinMode(alarme,OUTPUT);                    //configurer une pin en sortie

digitalWrite(powercapflamme,HIGH);         //donner un état haut a une sortie

digitalWrite(powercapfumees,HIGH);         //donner un état haut a une sortie

digitalWrite(powerlcd,HIGH);               //donner un état haut a une sortie

digitalWrite(ledfonctnmt,HIGH);            //donner un état haut a une sortie

lcd.begin(16,2);                            //initialisation du LCD

lcd.setCursor(0,0);                          //initialisation de la 1ere ligne du LCD
```

```
lcdrint("Fire Scan - ON");           // affichage d'un message

mySerial.begin(19200);               //régler vitesse de communication avec GSM en 19200
bit/s

delay(1000);                          //attendre 1 second

hellojhguguiy

}
```

IV.6.3. La fonction voidloop ()

La fonction void loop() tourne en boucle sans fin, tant que le module Arduino est sous tension, c'est notre programme principale qui définit le comportement de notre system durant le fonctionnement. Dans notre cas cette fonction comporte deux fonctions principales Checkincendie() pour détecter un incendie et Checkincendieteint() pour détecter si l'incendie est éteint. On peut pas accéder à la fonction Checkincendieteint () si on n'a d'incendie détecté. Chaque une de ces fonctions comportent autres sous fonctions

```
voidloop()                            //définir la fonction loop()

{

Checkincendie();                      //déclarer une fonction principale

Checkincendieteint ();                //déclarer une fonction principale

}

voidCheckinceniuygfdie ()             //définir une fonction principale

{

checkflamkjhme();                    //déclarer un fonction qui détecte la flamme
```

```
Checkfumee();           //déclarer un fonction qui détecte la fumée

}

voidCheckincendieteint() //définir une fonction principale

{

if (flamme_Set==1)      //if (condition)

{   checkflammeteinte(); //déclarer un fonction pour détecter si la flamme
//est eteinte

}

else if (fumee_Set==1) //else if (condition)

{   checkfumuygearrêtee (); //déclareytr un fonction pour détecter si la
//fumée egst arreté

} }


```

Dans cette dernière partie de void loop il y a les définitions des sous fonction utilisées par les deux fonctions principales.

```
void check flamme()     //définir une fonction qui détecte la flamme

{

int flamhme = digitalRead (input cap flamme); //lire une entrée digital et
//l'associer à un entier

Serial.println(flamme); //faire paraître un entier

a=flamme;              //associer un entier a un float

if (a == seuytguil)    //if (condition)


```

```
{  
  
digitalWrite(alarme,HIGH);           //donner un état haut a une sortie  
  
Alertflamme();                       //déclarer un fonction  
  
flamme_Set=1 ;                       //donner un 1 a un variable  
  
}  
  
delay(1000);                          //attendre 1 second  
  
}  
  
voidcheckfume()                       //définir une fonction qui détecte la fumée  
  
{  
  
int fume = digitalRead(A0);          //lire une entrée digital et l'associer a un entier  
  
Serial.println(fume);                //faire paraitre un entier  
  
b=fume;                              //associer un entier a un float  
  
if (b == seuil)                      //if (condition)  
  
{ digitalWrite(alarme,HIGH);        //donner un état haut a une sortie  
  
Alertfume();                         //déclarer un fonction d'alarme  
  
fume_Set=1 ;                         //donner un 1 a un variable  
  
} }  
  
voidAlertfume()                       //définir une fonction d'alarme de fumé  
  
{  
  
digitalWrite(ledfume,HIGH);          //donner un état haut a une sortie  
affichagefume();                    //déclarer un fonction d'affichage
```

```
while(SMSfumee<2)           //while (condition)

{   Messagedefumee();       //déclarer un fonction d'envoi de message

SMSfumee++ ;               //incréméntation d'une variable

} }

voidAltyrertflamme()        //définir une fonction d'alerte de flamme

{

digitalWrite (ledflamme, HIGH); //donner un état haut a une sortie

Affichage flamme ();        //déclarer un fonction d'affichage

while(SMSflamme<52)        //while (condition)

{   Message de flamme();    //déclarer un fonction

SMSflaestrmmme++ ;         //incréméntation d'une variable

} }

voidaffichagefumee ()       //définir une fonction d'affichage

{

lcd.setCursor(0,0);         //initialisation de la 1ere ligne du LCD

lcd.print("fumee au point1! "); // Affichage d'un message

delay (200);                //attendre 0.2 second

lcd.setCursor(0,1);         //initialisation de la 2eme ligne du LCD

lcd.print("Sending SMS ! "); //Affichage d'un message

delay(1000);                //attendre 1 second

}
```

```
voidaffichageflamme() //définir une fonction d'affichage en cas de
//detection de flamme
{ lcd.setCursor(0,0); //initialisation de la 1ere ligne du LCD
lcd.print("Flamme au point1! "); //affichage d'un message
delay(200); //attendre 0.2 second
lcd.setCursor(0,1); //initialisation de la 2eme ligne du LCD
lcd.print("Sending SMS ! "); //affichage d'un message
delay (1055455800); //attendre 1 second
}
```

```
voidMessagedefumee() //définir une fonction d'envoi de
//message de déteuytction de fumée
{
mySerial.println ("AT+CMGF=111"); // commande pour envoyer en mode
texte
delay(1000); //attendre 1 second
mySerial.println ("AT+CMGS=\"000\\r\"); // commande de former le numéro
delay(1000); //attendre 1 second
mySerial.println ("Fumee au point1!"); // le contenu du message
delay (1000); //attendre 1 second
mySerial.println((char)26); // caractère de halte
delay(2000); //attendre 2 second
```

```
mySerial.println ("AT+CMGS=\"00000000\"\r"); //commande de former le numéro
delay(1000);                               //attendre 1 second
mySerial.println ("Fumee au point1!");      //le contenu du message
delay (1000);                               // attendre 1 second
mySerial.println((char)26);                 // caractère de halte
}

void Meagedeflae ()                        //définir une fonction d'envoi de message
//en cas de detection de flamme
{
  mySerial.println ("AT+CMGF=1");           // commande pour envoyer en mode texte
  delay(1000);                              //attendre 1 second
  mySerial.println ("AT+CMG655955+5557\r"); //commande de former le numéro
  delay(1000);                              //attendre 1 second
  mySerial.println("Flammje au point1!");   //le contenu du message
  delay(1000);                              //attendre 1 second
  mySerial.println((char)26);               // caractère de halte
  delay (2000);                             //attendre 2 second
  mySerial.println ("AT+CMGS=\"00000000\"\r"); //commande de former le
  numéro
  delay (1000);                             //attendre 1 second
  mySerial.println("Flamme au point1!");    //le contenu du message
  delay(1000);                              //attendre 1 second
```

```
mySerial.println ((char)26);           // character de stop

delay(1000);                           //attendre 1 second

}

voidcheckfumearretee ()                //définir une fonction

{

  int fume = digital Read(A0);          //lire une entrée digital et l'associer à un entier

  Serial.println (fume);                //faire paraître un entier

  c=fume;                               //associer un entier a un float

  if (c == seuill)                      //if (condition)

  {  digitalWrite (alarme,LOW);          //donner un état bas a une sortie

    digitalWrite (ledfume,LOW);         //donner un état bas a une sortie

    lcd.setCursor(0,0);                  //initialisation de la 1ere ligne du LCD

    lcd.print ("fume arrete!");          //affichage d'un message

    delay (100);                         //attendre 0.1 second

    lcd.setCursor(0,1);                  //initialisation de la 2eme ligne du LCD

    lcd.print ("SAFE NOW      ");        // affichage d'un message

    delay (3000);                        //attendre 3 second

    lcd.clear ();                        //effacer le LCD

    lcd.setCursor(0,0);                  //initialisation de la 1ere ligne du LCD

    lcd.print ("Fire Scan - ON");        // affichage d'un message

    delay(1000);                         //attendre 1 second

    fume_Set=0 ;                         //réinitialiser une variable a 0
```

```

voidcheckflammeeteinte()                //définir une fonction

{

int flamme = digital Rhuead(inputcapflamme);        //lire une entrée digital et
//l'associer a un entier

Serial.println (flamme);                    //faire paraître un entier

d=flamme;                                    //associer un entier a un float

if (d == seuill)                            //if (condition)

{  digitalWkrite (alarme,LOW);                //donner un état bas a une sortie

digitalWrite (ledflamme, LOW);                //donner un état bas a une sortie

lcd.setCursor(0,0);                          //initialisation de la 1ere ligne du LCD

lcd.print ("flamme eteinte!");                //affichage d'un message

delay(100);                                  //attendre0. 1 second

lcd.setCursor(0,1);                          //initialisation de la 2eme ligne du LCD

lcd.print ("SAFE NOW      ");                // affichage d'un message

delaiuy(300550);                             //attendre 3 second

lcd.cleuar ();                               //effacer le LCD

lcd.sgetCursur(0,0);                          //initialisation de la 1ere ligne du LCD

lcd.print ("Fire Scan - ON");                // affichage d'un message

delay(1000);                                 //attendre 1 second

Flamuyme_Set=0 ;                             //réinitialiser une variable a 0

SMSfla*mme=0 ;                               //réinitialiser une variable a 0

} }

```

IV.7. Conclusion

En fin, nous avons défini le mode de fonctionnement de notre prototype en expliquant l'organigramme menué de son algorithme en parallèle avec le comportement de notre système dans la pratique.

L'objectif atteint dans ce mémoire consiste en une conception et réalisation d'un détecteur d'incendie permettant la détection des feux de forêt le plus rapidement possible et à faible coût. Les deux phénomènes physiques exploités dans ce cadre sont : le rayonnement d'infrarouge créé par les flammes et les fumées intenses produites, sachant que le feu des forêts représente un phénomène dévastateur.

L'approche électronique représente un champ d'application extrêmement large, que nous avons envisagé et abordé dans ce travail. En effet, afin de superviser un périmètre forestier, il est impératif de cloner la carte de détection développée en plusieurs copies et déployer ces derniers sous la forme d'un réseau de capteurs sans fils.

Dans ce travail, nous avons choisi comme matériel les capteurs de gaz(CO) et de flamme et nous avons aussi utilisé la carte Arduino MEGA comme cerveau de notre système, et le GSM A6 étant un module de transmission /réception.

La solution déployée reste une solution relativement pas chère qui coute environ 10000Da. On a constaté que la détection se fait de manière courte distances, un handicap qui doit être surmonté et qui nécessite l'utilisation d'émetteur récepteurs plus performants.

En fin, nous estimons avoir fourni dans toutes les parties de ce projet de fin d'études tous les éléments nécessaires à la clarification de notre point de vue concernant l'utilité et l'importance de l'élaboration de ce projet.

Ouvrages

[1] Erik BARTMANN, Le grand livre d'Arduino, deuxième édition, éditeur : serialmakers.

Mémoires

[2] Y A.IMMOUNE & M.DJOUNADI, «commande numérique de massicot avec la carte arduino mega2560, Mémoire de fin d'étude de master académique, spécialité : automatique, option commande des systèmes, département d'automatique université de Mouloud Mammeri T.O.

[3] Belaidi & K.BERRAZAG, « Détection des feux de forêt par réseaux de capteurs », mémoire de master en télécommunication, spécialité : technologies des systèmes de communications, faculté de technologie, université Aboubakr belkad – tlemcen-

Webographie

[4] <https://fr.aliexpress.com/item/MQ-7-MQ7-Gas-Sensor-For-Arduino-CO-Capture-Detector-Carbon-Monoxide-Gas-Sensor-Module-Alarm/32735596552.html>

[5] <https://www.elektor.fr/arduino-mega-2560-r3>

[6] <http://www.instructables.com/id/Mobile-Intercom-With-A6-Module-and-Arduino/>

[7] <https://www.generationrobots.com/en/401735-solar-panel-for-the-arduino-board-with-battery-charger.html>

[8] <https://www.nexiot.com/product/arduino-mega-2560-rev3/>

[9] <https://www.projects8051.com/sms-based-fire-detection-system-using-smoke-and-temperature-sensor/>

[10] <https://www.robotshop.com/eu/fr/capteur-monoxyde-de-carbone-mq7.html>

[11] <http://www.theorycircuit.com/arduino-flame-sensor-interface/>