

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 : Electronique

Spécialité : Instrumentation

Par : Salah Omar Abdelhak
Rahaoui Ayoub

Sujet

Conception et Réalisation des séquences d'animations à distance depuis un Smartphone

Soutenu publiquement, le 29/09 /2022 , devant le jury composé de :

Mr. Bennada Belkacem

Mme. Kaddouri Nadera

Mr. Massoum Noureddine

Université de Tlemcen

Université de Tlemcen

Université de Tlemcen

Président

Examineur

Encadreur

Année universitaire : 2021/2022

Dédicace :

Tout d'abord, je tiens à remercier DIEU De m'avoir donné la force et le courage de mener à bien ce modeste travail.

Je tiens à dédier cet humble travail à: A ma mère et mon très cher père..

A mes sœurs, A mon binôme : Omar Abdelhak Salah

A mes meilleurs Amis et tous ceux qui m'aiment et que j'aime.

Rahaoui Ayoub

Dédicace :

Je dédie Ce modeste travail à : A mes chers parents, que nulle dédicace ne puisse exprimer mes sincères sentiments, pour leur patience illimitée, leur encouragement continu, leur aide, en témoignage de mon profond amour et respect pour leurs grand sacrifices.

À mes chères sœurs.

À mes frères.

À Mon chère binôme « Rahaoui Ayoub » Et à tous ceux qui ont contribué de près ou de loin pour que ce travail soit possible, je vous dis merci...

Omar Abdelhak Salah

Remerciements :

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrions témoigner toute nos gratitude.

Nous voudrions tout d'abord adresser toutes nos reconnaissances à notre encadreur de ce mémoire, Monsieur MASSOUM. NOUREDDINE, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter nos réflexions.

Nous remercions les Enseignants de la faculté de technologie, qui nous ont fourni les outils nécessaires à la réussite de nos études universitaires.

Je tiens à remercier spécialement Mr le professeur Bennada Belkacem qui a accepté de presider cette soutenance et Madame KADDOURI NADERA de l'examiner.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Un grand merci à Mr Mohamed Amine Breksi, et Madame Bouazza Guen Ahlem pour leur confiance et leur soutien inestimable.

Table des matières

Remerciements

Dédicaces

Table des matières	I
Liste de figures.....	IV
Liste de tableaux.....	VI

Introduction générale.....	1
----------------------------	---

Chapitre 1: Généralités sur les enseignes lumineuses

1.1 Introduction.....	2
1.2 Définition d'enseigne lumineuse	2
1.3 Types d'enseignes lumineuses	3
1.4 Pourquoi choisir une enseigne lumineuse.....	4
1.5 Utilisation.....	5
1.6 Conclusion	6

Chapitre 2: Technologie de composants

2.1 Introduction.....	7
2.2 Les Transistors	7
2.2.1 Les transistors bipolaires (rappels)	7
2.2.2 Les transistors à effet de champs	10
2.2.3 Les transistors IRFZ44N	11
2.2.3.1 Brochage de l'IRFZ44N	11
2.3 Diodes électroluminescentes.....	12
2.3.1 C'est quoi une LED RGB ?	12
2.3.2 Comment brancher une LED RGB	12
2.4 Les régulateurs de tension.....	13
2.4.1 Quel est le rôle de la régulation.....	14
2.5 Arduino	14
2.5.1 Qu'est-ce que Arduino	14

2.5.2 Architecture de L'Arduino.....	15
2.5.3 Arduino Uno	16
2.5.4 Arduino Nano	17
2.5.5 Caractéristiques de l'Arduino Nano.....	18
2.5.5.1 Brochage de l'Arduino Nano.....	19
2.5.6 Arduino Mega 2560.....	20
2.5.6.1 Spécifications techniques	21
2.6 Bluetooth.....	22
2.6.1 Comment se connecter avec le Bluetooth.....	22
2.6.2 Bluetooth HC05	23
2.7 Conclusion	26

Chapitre 3: Outils logiciels et Réalisation

3.1 Logiciel MikroC	27
3.1.1 Qu'est-ce qu'un MikroC PRO pour PIC	27
3.1.2 Comment programmer un microcontrôleur image avec MikroC ?.....	27
3.1.3 Qu'est-ce que PIC SimLab	27
3.1.4 Pourquoi utiliser le compilateur MikroC.....	27
3.1.5 Créer un nouveau projet avec MikroC	28
3.1.6 Définir les bits de configuration.....	30
3.1.7 Utilisation du Langage MikroC	32
3.1.8 Structure d'un programme en mikroC.....	32
3.1.9 Exemple de code MikroC	33
3.1.10 Compiler le code avec MikroC.....	33
3.2 Logiciel Proteus.....	33
3.2.1 Proteus Simulation	34
3.2.2 Utilisation de l'application ISIS.....	36
3.3 MIT App Inventor	37
3.3.1 C'est quoi App Inventor	37
3.3.2 Comment créer une application mobile gratuitement.....	38
3.3.3 Création d'un exemple simple.....	40
3.3.4 Quel logiciel utiliser pour créer une application	43
3.4 Partie pratique	44
3.4.1 Code de PIC	44

3.4.2 Schéma du circuit	49
3.4.3 Réalisation pratique	51
3.5 Conclusion	52
Conclusion Générale	53
Références Bibliographiques	54

Liste de figures

Chapitre 1:

Figure 1.1 : Une Enseigne lumineuse.	2
Figure 1.2 : Des états différents de enseignes lumineuses.....	3
Figure 1.3 : Une Enseigne lumineuse.	4
Figure 1.4 : Une Enseigne publicitaire.....	5
Figure 1.5 : Une Enseigne de magasin.	5

Chapitre 2

Figure 2.1: Le transistor NPN et PNP.....	8
Figure 2.2: Les trois regions d'un transistor.....	10
Figure 2.3: Les symboles d'un transistor.....	10
Figure 2.4: Deux circuits similaires (un BJT et un FET).....	11
Figure 2.5 : MOSFET IRFZ44N.....	11
Figure 2.6 : LED RGB.....	12
Figure 2.7 : Carte Arduino.....	15
Figure 2.8 : Architecture de L'Arduino.....	16
Figure 2.9 : Architecture d'Arduino Uno.....	16
Figure 2.10 : Composants d'un Arduino.	17
Figure 2.11 : Arduino Nano.	18
Figure 2.12 : Brochage de l'Arduino Nano.....	19
Figure 2.13 : Brochage de l'Arduino Mega 2560.	21
Figure 2.14 : Icon Bluetooth.....	22
Figure 2.15 : Broche de Bluetooth HC05.....	23
Figure 2.16 : Connexion hardware entre le module Bluetooth HC-05 et Arduino UNO.	25
Figure 2.17 : Transfert de données entre Arduino UNO et PC via des dispositifs Bluetooth.....	26

Chapitre 3

Figure 3.1 : L'interface du mikroC.....	28
Figure 3.2 : Création un nouveau projet sur mikroC.....	29
Figure 3.3 : Choix de PIC et fréquence sur mikroC.....	29
Figure 3.4 : Zone de code sur mikroC.....	30
Figure 3.5 : Paramètres du projet sur mikroC.	31
Figure 3.6 : Fenêtre de modification du projet sur mikroC.	32
Figure 3.7 : Examinons le fichier LEDc du code-source.	32
Figure 3.8 : Code source sur mikroC.....	33
Figure 3.9 : Capture d'écran de logiciel proteus.....	34
Figure 3.10 : Schéma électronique du PIC sur logiciel proteus.....	35
Figure 3.11 : Fenêtre de PIC pour ajouter le code.....	35
Figure 3.12 : Boîte de simulation sur proteus.....	36
Figure 3.13 : Simulation du circuit électronique sur proteus.	36
Figure 3.14 : Schéma de MIT app inventor.....	38
Figure 3.15 : MIT application sur play store.	39
Figure 3.16 : L'interface de MIT inventor.	40
Figure 3.17 : Code QR MIT inventor.....	40
Figure 3.18 : Création d'un projet sur MIT inventor.	41
Figure 3.19 : Nom du projet sur MIT inventor.....	41
Figure 3.20 : Ecriture de code sur MIT inventor.....	42
Figure 3.21 : Affichage de Hello World ! sur MIT inventor.....	43
Figure 3.22 : Connexion par code QR sur MIT inventor.....	43
Figure 3.23 : Affichage de Hello World ! sur l'écran de samrtphone.....	44
Figure 3.24 : Composants électroniques utilisés sur ISIS Proteus.	51
Figure 3.25 : Schéma du circuit d'enseigne lumineuse sur ISIS Proteus.	51
Figure 3.26 : Circuit électronique réalisé.	53
Figure 3.27 : Schéma du circuit de commande d'enseigne lumineuse avec arduino nano et HC05 sur ISIS Proteus	53

Liste de tableaux

Chapitre 2

Tableau 2.1 : Description de chaque broche de MOSFET.....	11
Tableau 2.2 : Spécification technique d'Arduino Mega 2560.	22
Tableau 2.3 : Fonction et description de chaque broche de HC05.....	24

Chapitre 3

Tableau 3.1 : Les valeurs de type de variable signed et unsigned.	27
--	----

Introduction Générale

Introduction Générale

De nos jours nous assistons à une forte évolution de la technologie, plus particulièrement dans le domaine de la communication notamment les « smartphones ».

La communication homme-machine ou machine-machine peut être considérée comme étant un nouveau type de dialogue possible. En effet, depuis une dizaine d'années les appareils deviennent intelligents, agissent selon le profil des utilisateurs et sont capables de prendre des décisions de manière autonome. Cet outil « smartphones » nous apporte donc beaucoup plus de fonctionnalité, en dehors de son utilisation pour les appels téléphoniques. On peut les utiliser pour renforcer notre sécurité et notre confort de près comme à une petite ou grande distance via Bluetooth qui est déjà intégré dans le smartphone ou un Module GSM.

Ceci Donne la possibilité via l'envoi et la réception des SMS de commander et contrôler un système/processus quelconque.

Sur Ce, l'objet de notre mémoire est de commander des enseignes lumineuses (l'éclairage couleur ou monochrome, avec des séquences d'animations) par un smartphone via Bluetooth.

- Trois (3) points nous ont motivés pour le choix de ce projet qui sont :

Commander à distance une enseigne lumineuse.

Visualiser les différentes séquences d'animation monochromes et couleurs.

Le confort de l'utilisateur.

- Objectifs du projet :

- Regrouper suffisamment d'informations sur la carte développement Arduino: son langage de programmation, son Principe de fonctionnement.

- Comprendre le Principe de fonctionnement d'une communication serie depuis un Bluetooth

- Comment se fait l'envoi et la lecture d'un SMS

- Réaliser un système capable d'exécuter une action entre un smartphone, un Module Bluetooth et une carte

Arduino en expliquant les différents blocs.

Chapitre 1 :

Généralités sur les enseignes lumineuses

1.1 Introduction

Dans notre vie quotidienne, on remarque que les enseignes lumineuses sont très utilisées pour la publicité grâce à ses caractéristiques visuelles.

Ces enseignes offrent plusieurs d'avantages dans le coté de cout, consommation d'énergie, fabrication et la facilité de l'installer.

1.2 Définition d'enseigne lumineuse

L'enseigne lumineuse est l'un des supports les plus efficaces de la communication publicitaire. Quelle que soit la technique d'éclairage, elle est toujours équipée d'un système d'éclairage soit à LED, soit à néon. Tout comme l'enseigne de magasin, elle se compose de toutes les inscriptions, formes ou images lumineuses pouvant être apposées sur la façade d'un immeuble où une activité est exercée. L'enseigne, qu'elle soit lumineuse ou non, est un moyen d'affichage qui permet :

- D'attirer la clientèle.
- De favoriser l'image.
- De transmettre un message.



Figure 1.1 : Une Enseigne lumineuse.

Les enseignes lumineuses sont éteintes entre 1 heure et 6 heures, lorsque l'activité a cessé. Si l'activité cesse ou commence entre minuit et 7h, les enseignes sont éteintes 1 heure après la cessation d'activité, et peuvent être allumées 1 heure avant la reprise.

Dérogation possible lors d'événements exceptionnels définis par arrêté municipal ou préfectoral. Enseignes clignotantes interdites, sauf pharmacie et service d'urgence.

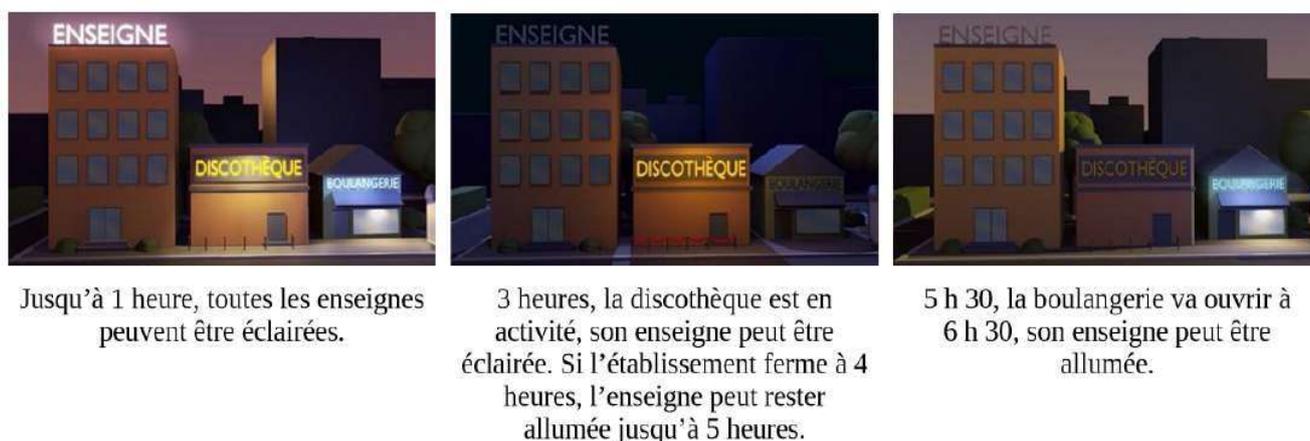


Figure 1.2 : Des états différents de enseignes lumineuses.

1.3 Types d'enseignes lumineuses

Les enseignes lumineuses permettent de valoriser son entreprise de jour comme de nuit, et se déclinent sous différents modèles :

- **Les caissons lumineux**

Le caisson lumineux est un dispositif visuel qui permet d'exposer une affiche. Totalement ou partiellement transparent, il dispose d'une installation électrique lumineuse (rétro-éclairage par LED ou néon) qui vous permet de rehausser votre affichage.

- **Les enseignes à LED**

L'enseigne lumineuse à LED dispose de diodes électroluminescentes pour son éclairage. Qu'importe le modèle ou la marque de l'enseigne, le procédé reste identique : ces diodes

permettent à l'électricité de traverser la matière et l'énergie qui se libère permet un éclairage vers l'extérieur.

- **Les enseignes de néon**

Pour être vue, l'enseigne à néon est l'un des supports les plus efficaces. Les enseignes à néon en verre : il s'agit d'enseignes lumineuses éclairées par des tubes fluorescents. Ces tubes éclairent, par en-dessous, un caisson qui est équipé d'une face en verre. Les enseignes à néon flexibles : elles utilisent les dernières technologies en matière d'éclairage afin de pallier les inconvénients relatifs aux enseignes à néon classiques.

1.4 Pourquoi choisir une enseigne lumineuse

L'enseigne lumineuse est un visuel de l'entreprise qui favorise le contact avec le client. Apposer une enseigne lumineuse sur son entreprise ou son magasin permet :

- De se démarquer de la concurrence.
- De capter le regard du client.
- D'être visible de jour comme de nuit.



Figure 1.3 : Une Enseigne lumineuse.

1.5 Utilisation

L'enseigne lumineuse est d'une utilité capitale pour un magasin ou une entreprise, qu'elle soit placée dans une rue animée ou dans une zone peu commerciale. On retrouve cette utilité dans :

- **Les enseignes publicitaires**

L'enseigne publicitaire est disposée à proximité immédiate de l'immeuble où s'exerce l'activité dont elle fait la promotion. Il s'agit d'un dispositif de signalétique extérieure placé pour mettre en valeur votre image, votre activité ou vos produits et services proposés à la vente.



Figure 1.4 : Une Enseigne publicitaire.

- **Les Enseignes de magasin**

Une enseigne de magasin est un des éléments de communication les plus efficaces : elle vous permet d'améliorer la notoriété de votre magasin grâce aux différents messages que vous pouvez véhiculer à travers elle. L'enseigne d'un magasin est le prolongement de son nom commercial : il s'agit d'un visuel informatif, publicitaire et décoratif qui reflète l'image de votre magasin au public. Elle est le facteur-clé de l'identification de votre commerce.



Figure 1.5 : Une Enseigne de magasin.

1.6 Conclusion

Dans ce chapitre, nous avons présenté quelques généralités sur les enseignes lumineuses.

Ces enseignes sont très utilisées grâce à ses avantages qui l'offrent pour attirer les clients afin de développer le commerce.

Chapitre 2 :

Technologie de composants

2.1 Introduction

Un composant est dit passif quand il ne permet pas d'augmenter la puissance d'un signal, il peut même parfois par effet Joule réduire la puissance disponible en sortie. Tel est le cas des résistances, condensateurs, bobines et transformateurs.

A l'opposé, un composant électronique est nommé actif quand il permet d'amplifier ou de transformer la puissance d'un signal en tension et/ou en courant. La puissance ainsi générée est récupérée au travers d'une alimentation. Les composants actifs sont des composants capables d'introduire de l'énergie dans le circuit auquel il contribue.

Les composants actifs dans la grande majorité sont des semi-conducteurs, classés en deux grandes catégories : les composants discrets et les circuits intégrés. Les composants actifs discrets réalisent une seule fonction, le cas des diodes à effet tunnel, transistors et tubes électroniques. Parmi les composants actifs, on distingue ceux spécifiques utilisés en électronique de puissance : diode, thyristor, transistor. Ces composants permettent la conversion statique de l'énergie électrique.

Les composants électroniques actifs et passifs, assemblés en modules puis interconnectés dans le but de réaliser une ou plusieurs fonctions électroniques, constituent un circuit intégré.

Les circuits intégrés plus ou moins complexes intègrent plusieurs types de composants électroniques, ils sont conçus pour réaliser une ou plusieurs fonctions électroniques, le cas des microprocesseurs, amplificateurs opérationnels, fonctions combinatoires, convertisseurs analogique-numérique.

Les constructeurs doivent créer davantage des composants électroniques, à un meilleur coût et de meilleure qualité afin de pouvoir couvrir au mieux la demande du cahier de charge qui est en évolution rapide et intense. [1]

2.2 Les Transistors

2.2.1 Les transistors bipolaires (rappels)

Un transistor bipolaire est un dispositif semi-conducteur couramment utilisé pour l'amplification. Ce dispositif peut amplifier des signaux analogiques ou numériques. Il peut également commuter le courant continu ou fonctionner comme un oscillateur.

Physiquement, un transistor bipolaire amplifie le courant, mais il peut être connecté dans des circuits conçus pour amplifier la tension ou la puissance. [2]

Il est constitué de trois couches de semiconducteurs extrinsèques. On distingue deux types de transistors bipolaires : les transistors NPN et les transistors PNP. Le transistor NPN est constitué par :

- Une couche N fortement dopée constituant l'émetteur.
- Une couche P très mince et faiblement dopée constitue la base. Une couche N faiblement dopée constituant le collecteur.

Le transistor PNP est constitué par :

- Une couche P fortement dopée constituant l'émetteur.
- Une couche N très mince et faiblement dopée constitue la base.
- Une couche P faiblement dopée constituant le collecteur.

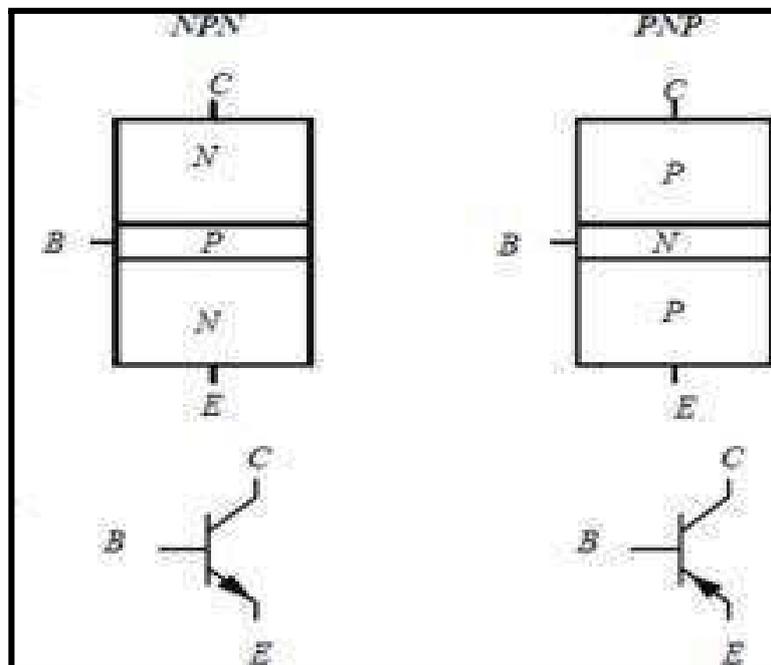


Figure 2.1: Le transistor NPN et PNP.

Les transistors sont des dispositifs actifs à trois bornes fabriquées à partir de différents matériaux semi-conducteurs qui peuvent agir comme un isolant ou un conducteur en appliquant une petite tension de signal. Isolant ou conducteur par l'application d'une petite tension de signal. La capacité du transistor à passer de l'un à l'autre de ces deux états lui permet d'avoir deux fonctions de base : l'isolation et le conducteur.

Deux états lui permettent d'avoir deux fonctions de base : commutation (électronique numérique) ou amplification (électronique analogique). Ensuite, les transistors bipolaires ont la capacité de fonctionner dans trois régions différentes :

- La région active - le transistor fonctionne comme un amplificateur et $I_c = \beta \cdot I_b$
- La saturation - le transistor est "complètement actif" et fonctionne comme un interrupteur, $I_c = I(\text{saturation})$.
- Coupure - le transistor est "complètement ouvert", il fonctionne comme un interrupteur et $I_c = 0$.

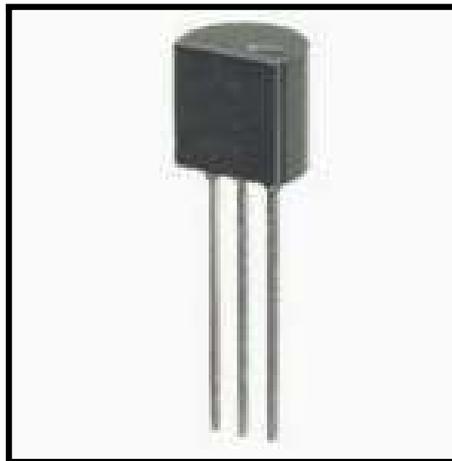


Figure 2.2: Les trois regions d'un transistor.

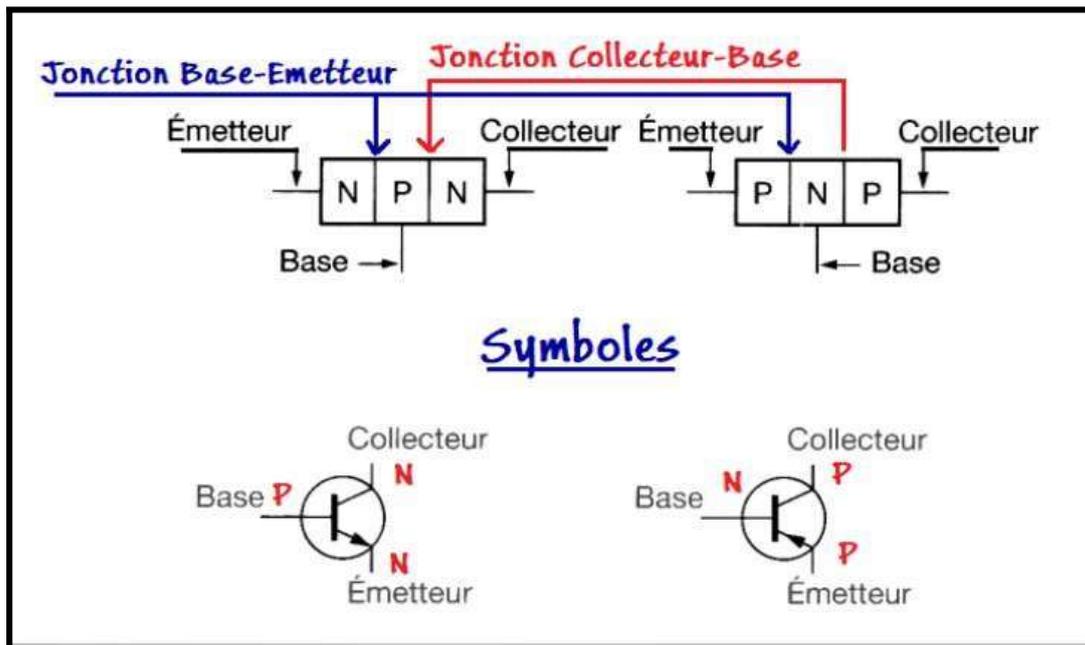


Figure 2.3: Les symboles d'un transistor.

2.2.2 Les transistors à effet de champs

Les FET (Field Effect Transistors) sont similaires aux BJT (Bipolar Junction Transistors) que nous avons utilisés, sauf que le courant de sortie est contrôlé par la tension de grille au lieu du courant de base. Pour ce FET à canal N, lorsque la tension de grille est élevée (par rapport à la tension de source), la résistance entre le drain et la source diminue, ce qui permet au courant de circuler du drain vers la source. L'exemple ci-dessous montre deux circuits similaires (un BJT et un FET). [3]

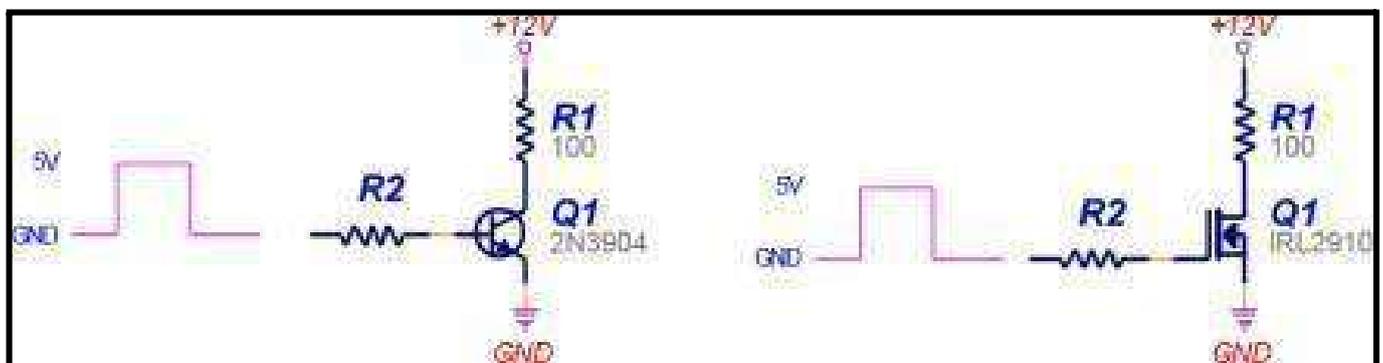


Figure 2.4: Deux circuits similaires (un BJT et un FET).

Lorsque 5V est appliqué à la grille du FET, il s'allume et permet au courant de circuler du drain vers la source. Lorsque la grille est à 0V, le FET est éteint (résistance haute) et seul un petit courant de fuite circule.

2.2.3 Les transistors IRFZ44N

IRFZ44N est un MOSFET à canal N qui a un courant de drain de 49 A et une valeur R_{ds} de 17,5 m. Il a également un seuil extrêmement bas de seulement 4 V, auquel le MOSFET commence à conduire. Il possède également un seuil extrêmement bas de seulement 4 V, à partir duquel le MOSFET commence à conduire. C'est pourquoi il est souvent utilisé avec des microcontrôleurs qui fonctionnent avec 5V. Mais un circuit de commande sera nécessaire lorsque le MOSFET doit être complètement activé. Les dispositifs sont disponibles dans une variété d'emballages de type trou traversant et montage en surface.

2.2.3.1 Brochage de l'IRFZ44N

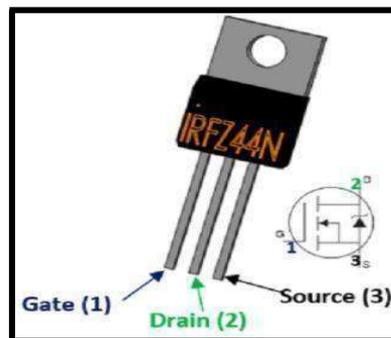


Figure 2.5 : MOSFET IRFZ44N.

Broche	Description
Gate (1)	Elle contrôle la polarisation du MOSFET.
Drain (2)	Le courant entre par le Drain
Source (3)	Le courant sort par la Source

Tableau 2.1 : Description de chaque broche de MOSFET.

2.3 Diodes électroluminescentes

2.3.1 C'est quoi une LED RGB ?

RGB LED signifie LED rouge, bleue et verte. Les produits RGB LED combinent ces trois couleurs pour produire plus de 16 millions de nuances de lumière.

Notez que toutes les couleurs ne sont pas possibles. Certaines couleurs sont "en dehors" du triangle formé par les LED RVB. De même, les couleurs de pigments comme le brun ou le rose sont difficiles, voire impossibles à obtenir. [4]

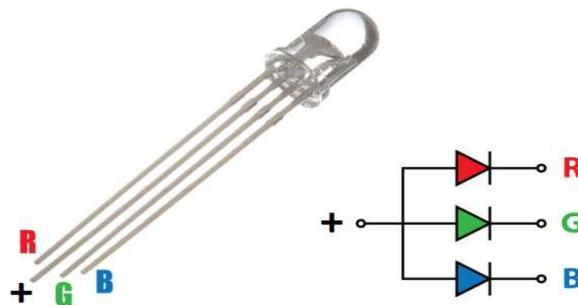


Figure 2.6 : LED RGB.

2.3.2 Comment brancher une LED RGB

- Choix du repère de coupe.

Avant de vous expliquer comment utiliser un connecteur rapide, petit rappel sur la méthode pour couper un ruban LED RGB. Rien de compliqué, il vous suffit simplement d'une paire de ciseaux (une pince coupante peut également faire l'affaire) et de choisir le repère de coupe qui se rapproche le plus de la longueur souhaitée. Les traits de coupe sont situés toutes les 3 LED.

- Résultat une fois le ruban LED coupé.

Voilà le résultat une fois la bande LED RGB coupée !

- Visualisation des points de contact

Quand nous regardons bien votre connecteur rapide, vous pouvez apercevoir sur la face avant le signe + (flèches rouges). il faut de bons yeux ! Nous retrouvons la même indication sur notre bande LED RGB, à proximité des points de contact (les plots cuivrés). Généralement, vous aurez les indications suivantes : « +DC12V », « +12V » ou « + ».

- Mise en place du premier ruban LED RGB

Ouvrez le premier clapet du connecteur. Insérez la première bande LED dans les encoches prévues à cet effet. Enfoncez-la jusqu'à ce que les pattes du connecteur touchent les points de contact (plots cuivrés) de la bande LED. Ils doivent se placer sous les pattes du connecteur.

- Fermeture du premier clapet

Une fois ceci fait, vous pouvez refermer le premier clapet de votre connecteur.

- Mise en place du second ruban LED RGB

Il ne reste plus qu'à répéter l'opération avec la seconde chute. N'oubliez de faire correspondre les polarités entre les rubans LED.

- Fermeture du second clapet Fermez le deuxième clapet du connecteur

Voilà, vous avez raccordé en un temps record et sans outils, les morceaux coupés de votre bande LED RGB.

- Fin du raccordement.

Si cela ne fonctionne pas, assurez-vous que les +12V correspondent entre votre bande LED multicolore et le connecteur. Ensuite, vérifiez bien la connexion entre les points de contact de la bande LED et les pattes du connecteur.

2.4 Les régulateurs de tension

Tous les appareils électroniques sont conçus pour fonctionner à des puissances nominales prédéterminées, c'est-à-dire la tension et le courant. Alors que la consommation de courant est dynamique et dépend de la charge de l'appareil, la tension d'alimentation est fixe et idéalement constante pour le bon fonctionnement de l'appareil. Un régulateur de tension est chargé de maintenir cette tension idéale nécessaire à l'appareil. Votre ordinateur portable, votre chargeur mural et votre télévision sont tous équipés de régulateurs de tension.

2.4.1 Quel est le rôle de la régulation

L'objectif d'un régulateur de tension est de maintenir la tension d'un circuit relativement proche de la valeur souhaitée. Les régulateurs de tension sont l'un des composants électroniques les plus courants, car une alimentation électrique produit fréquemment un courant brut qui, autrement, endommagerait l'un des composants du circuit. Les régulateurs de tension ont une variété de fonctions spécifiques, en fonction de leur application particulière.

2.5 Arduino

2.5.1 Qu'est-ce que Arduino

Arduino est une plateforme électronique open-source basée sur du matériel et des logiciels faciles à utiliser pour réaliser des projets électroniques. Toutes les cartes Arduino ont une chose en commun, à savoir un microcontrôleur. Un microcontrôleur est en fait un très petit ordinateur.

Avec l'Arduino, vous pouvez concevoir et construire des appareils qui peuvent interagir avec votre environnement. Les cartes Arduino sont essentiellement un outil pour contrôler l'électronique. Elles sont capables de lire les entrées à l'aide de leur microcontrôleur intégré (par exemple, la lumière sur un capteur, un objet à proximité d'un capteur) et de les transformer en une sortie (actionner un moteur, faire sonner une alarme, allumer une LED, afficher des informations sur un écran LCD). [5]

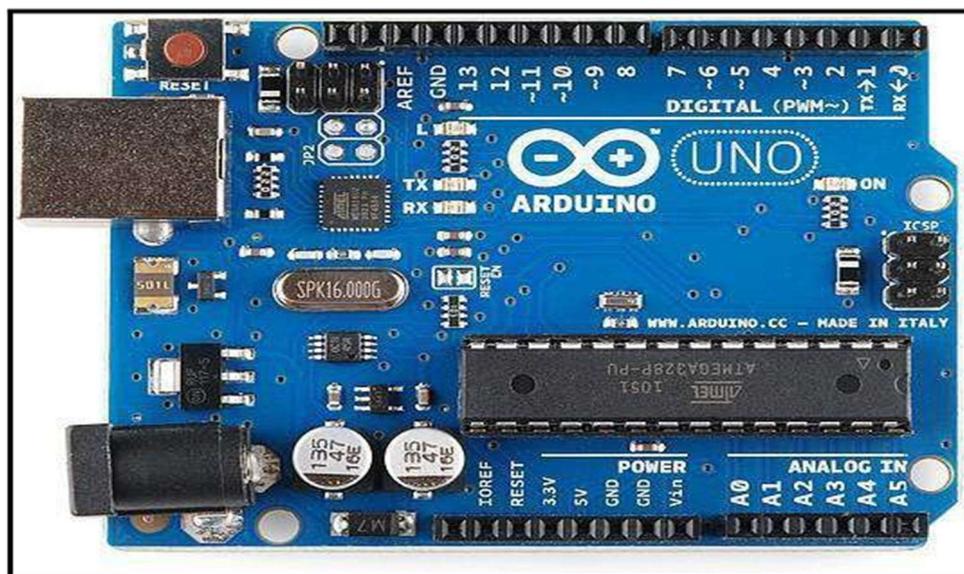


Figure 2.7 : Carte Arduino.

2.5.2 Architecture de L'Arduino

Le processeur d'Arduino utilise fondamentalement l'architecture Harvard où le code de programme et les données de programme ont une mémoire séparée. Le code est stocké dans la mémoire flash du programme, tandis que les données sont stockées dans la mémoire des données.

L'Atmega328 possède 32 Ko de mémoire flash pour le stockage du code (dont 0,5 Ko est utilisé pour le chargeur de démarrage), 2 Ko de SRAM et 1 Ko d'EEPROM et fonctionne avec une vitesse d'horloge de 16 MHz.

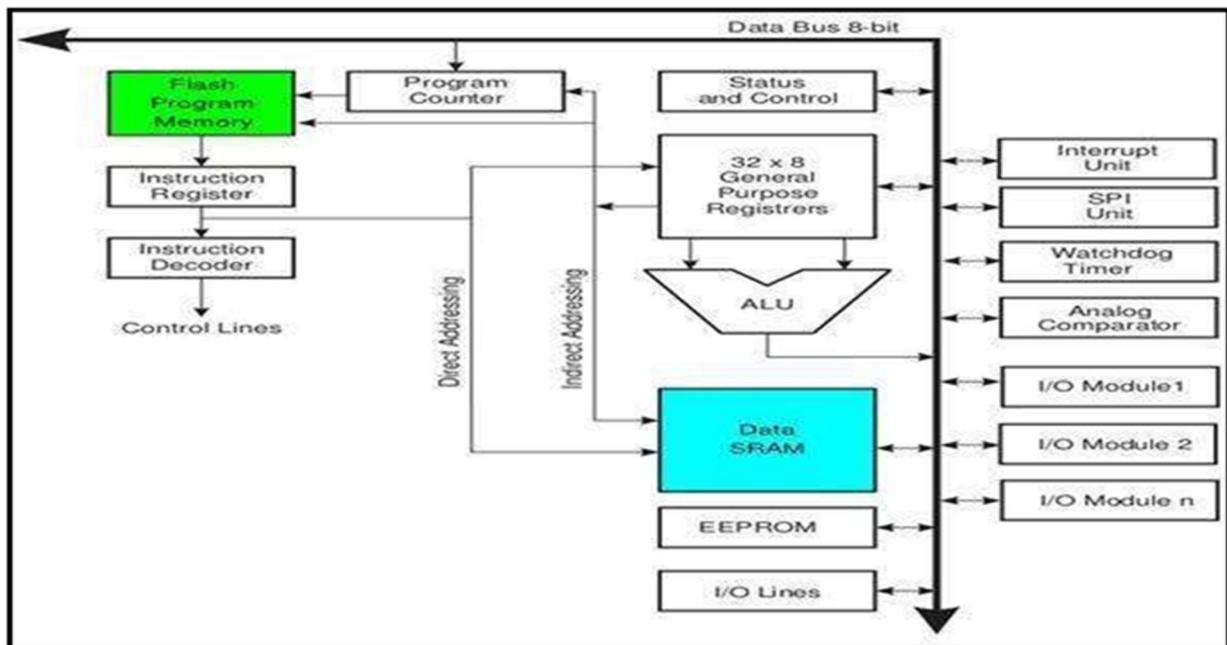


Figure 2.8 : Architecture de L'Arduino.

Un exemple typique de carte Arduino est l'Arduino Uno. Elle se compose d'un microcontrôleur de 28 broches, le ATmega328.

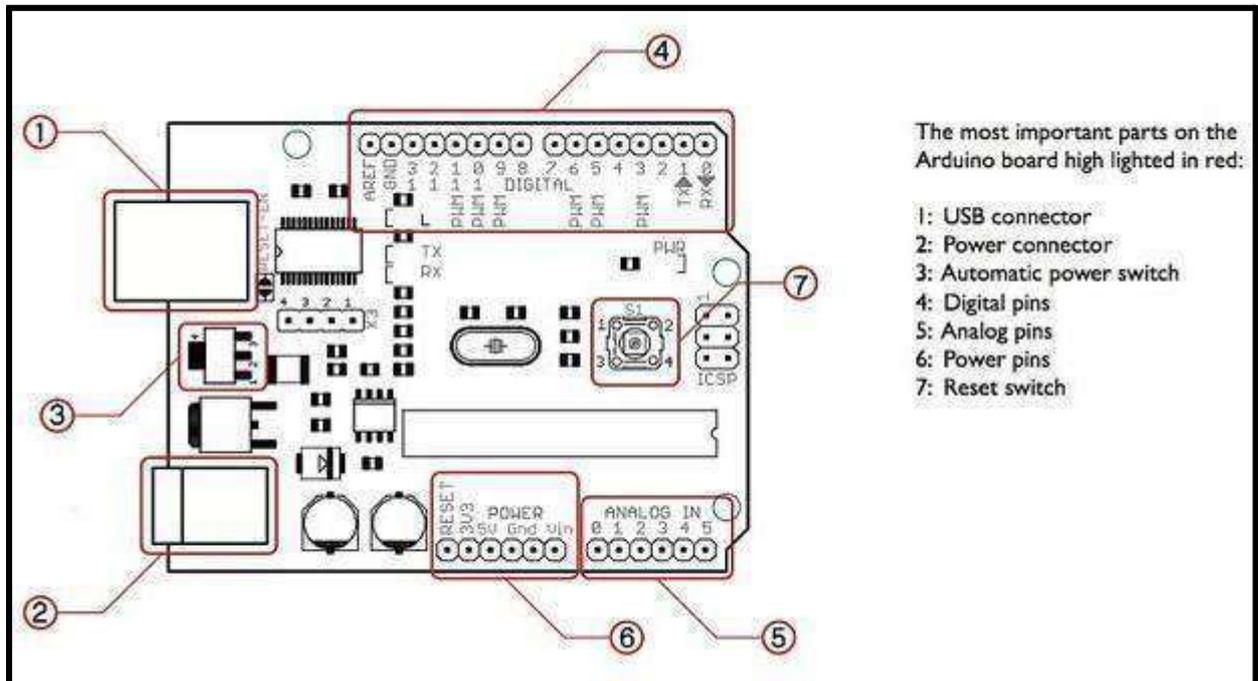


Figure 2.9 : Architecture d'Arduino Uno.

2.5.3 Arduino Uno

Arduino UNO est basé sur un microcontrôleur ATmega328P. Elle est facile à utiliser par rapport à d'autres cartes, telles que la carte Arduino Mega, etc. La carte est composée de broches d'entrée/sortie (E/S) numériques et analogiques, de blindages et d'autres circuits.

L'Arduino UNO comprend 6 entrées analogiques, 14 broches numériques, un connecteur USB, une prise d'alimentation et un connecteur ICSP (In-Circuit Serial Programming). Il est programmé à l'aide d'un IDE (Integrated Development Environment). Il peut fonctionner sur des plateformes en ligne et hors ligne.

L'IDE est commune à toutes les cartes disponibles d'Arduino.

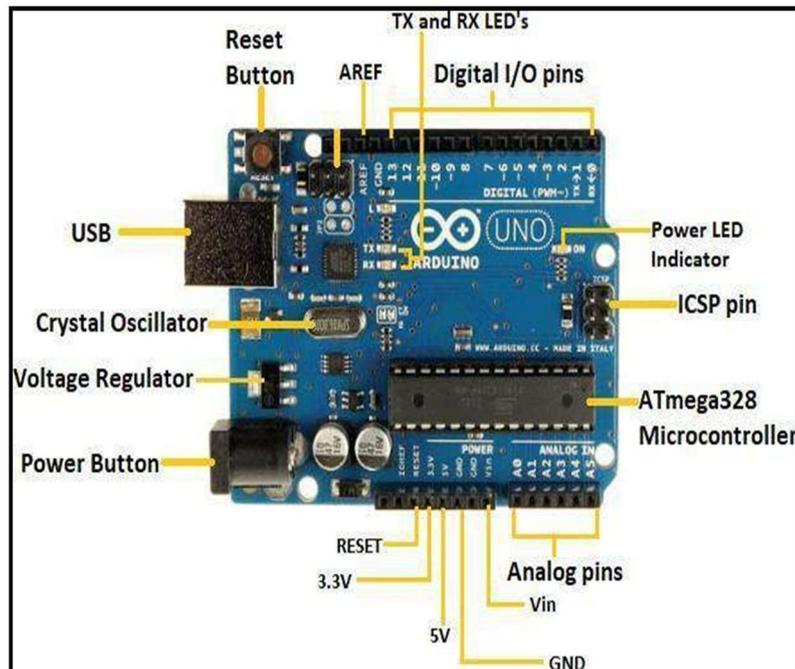


Figure 2.10 : Composants d'un Arduino.

2.5.4 Arduino Nano

Arduino Nano est un type de carte à microcontrôleur, et il est conçu par Arduino.cc. Elle peut être construite avec un microcontrôleur comme Atmega328. Ce microcontrôleur est également utilisé dans l'Arduino UNO. C'est une carte de petite taille et flexible avec une grande variété d'applications. Les autres cartes Arduino comprennent principalement Arduino Mega, Arduino Pro Mini, Arduino UNO, Arduino YUN, Arduino Lilypad, Arduino Leonardo, et Arduino Due. Et d'autres cartes de développement sont la carte de développement AVR, la carte développement PIC, le Raspberry Pi, l'Intel Edison, le Launchpad MSP430 et la carte ESP32. [6]

Cette carte possède de nombreuses fonctions et caractéristiques comme une carte Arduino Duemilanove. Cependant, cette carte Nano est différente dans son emballage. Elle n'a pas de prise DC de sorte que l'alimentation peut être fournie en utilisant un petit port USB ou en se connectant directement aux broches comme VCC et GND. Cette carte peut être alimentée avec 6 à 20 volts en utilisant un mini port USB sur la carte.

2.5.5 Caractéristiques de l'Arduino Nano

Les caractéristiques d'un Arduino nano sont principalement les suivantes.

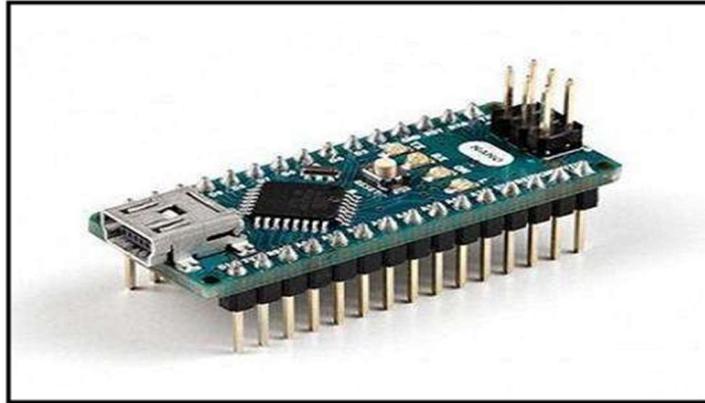


Figure 2.11 : Arduino Nano.

- Le microcontrôleur ATmega328P fait partie de la famille AVR 8 bits.
- La tension de fonctionnement est de 5V
- Tension d'entrée (V_{in}) de 7V à 12V
- Les broches d'entrée/sortie sont au nombre de 22
- Les broches d'entrée/sortie analogiques sont au nombre de 6, de A0 à A5.
- Les broches numériques sont au nombre de 14
- La consommation électrique est de 19 mA
- Le courant continu des broches d'E/S est de 40 mA
- Mémoire flash de 32 Ko
- SRAM : 2 Ko
- EEPROM : 1 Ko
- La vitesse du CLK est de 16 MHz
- Poids : 7 g
- La taille de la carte de circuit imprimé est de 18 x 45 mm.
- Supporte trois communications comme SPI, IIC, et USART
- (“Arduino Docs | Arduino Documentation”)
-

2.5.5.1 Brochage de l'Arduino Nano

La configuration des broches de l'Arduino Nano est présentée ci-dessous et la fonctionnalité de chaque broche est expliquée ci-dessous.

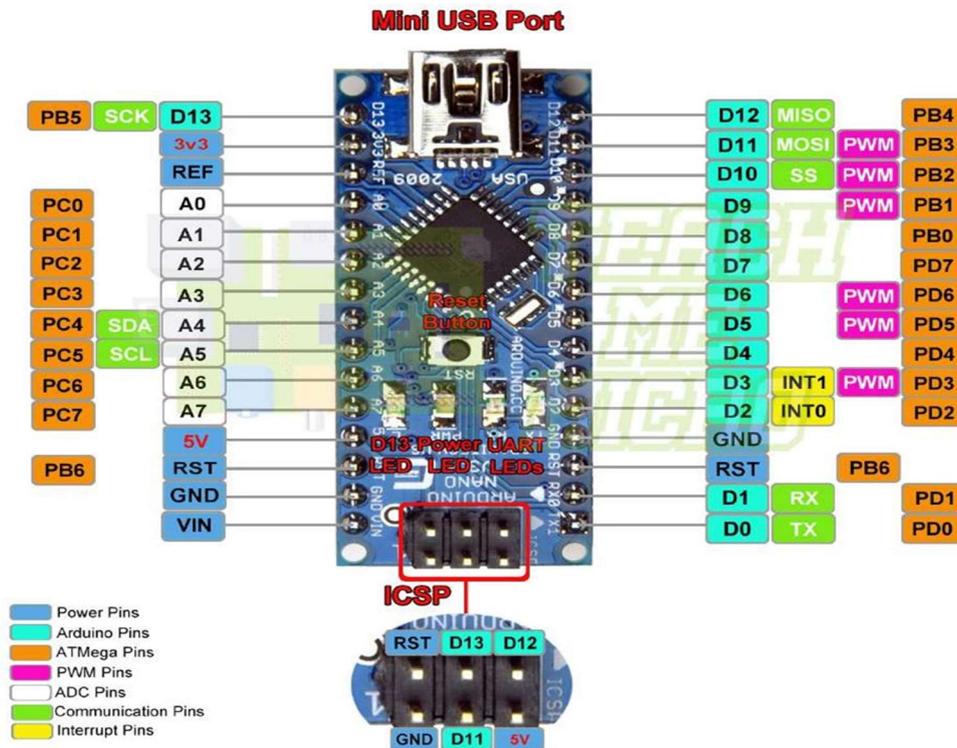


Figure 2.12 : Brochage de l'Arduino Nano.

- Broche d'alimentation (Vin, 3.3V, 5V, GND) : Ces broches sont des broches d'alimentation
- Vin est la tension d'entrée de la carte, et elle est utilisée lorsqu'une source d'alimentation externe de 7V à 12V est utilisée.
- 5V est la tension d'alimentation régulière de la nano carte et elle est utilisée pour alimenter la carte ainsi que les composants.
- 3.3V est la tension minimale générée par le régulateur de tension de la carte. GND est la broche de terre de la carte.
- Broche RST (Reset) : Cette broche est utilisée pour réinitialiser le microcontrôleur.
- Broches analogiques (A0-A7) : Ces broches sont utilisées pour calculer la tension analogique de la carte dans la gamme de 0V à 5V.
- Broches E/S (broches numériques de D0 à D13) : Ces broches sont utilisées comme broches i/p ou o/p. 0V et 5V

- Broches série (Tx, Rx) : Ces broches sont utilisées pour transmettre et recevoir des données série TTL.
- Interruptions externes (2, 3) : Ces broches sont utilisées pour activer une interruption.
- PWM (3, 5, 6, 9, 11) : Ces broches sont utilisées pour fournir 8-bit de sortie PWM.
- SPI (10, 11, 12, & 13) : Ces broches sont utilisées pour supporter la communication SPI.
- LED intégré (13) : Cette broche est utilisée pour activer la LED.
- IIC (A4, A5) : Ces broches sont utilisées pour supporter la communication TWI.
- AREF : Cette broche est utilisée pour donner une tension de référence à la tension d'entrée. ("ElProCus")

2.5.6 Arduino Mega 2560

L'Arduino Mega 2560 est une carte microcontrôleur basée sur l'ATmega2560. Elle possède 54 broches d'entrée/sortie numériques (dont 14 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UARTs (ports série matériels), un oscillateur à cristal de 16 MHz, une connexion USB, une prise d'alimentation, un circuit intégré et une prise de courant. une prise d'alimentation, un connecteur ICSP et un bouton de réinitialisation. Il contient tout ce qui est nécessaire pour supporter le microcontrôleur ; il suffit de le connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur CA/CC. L'alimenter avec un adaptateur CA-CC ou une batterie pour commencer. Le Mega est compatible avec la plupart des shields conçus pour l'Arduino Duemilanove ou Diecimila.[7]

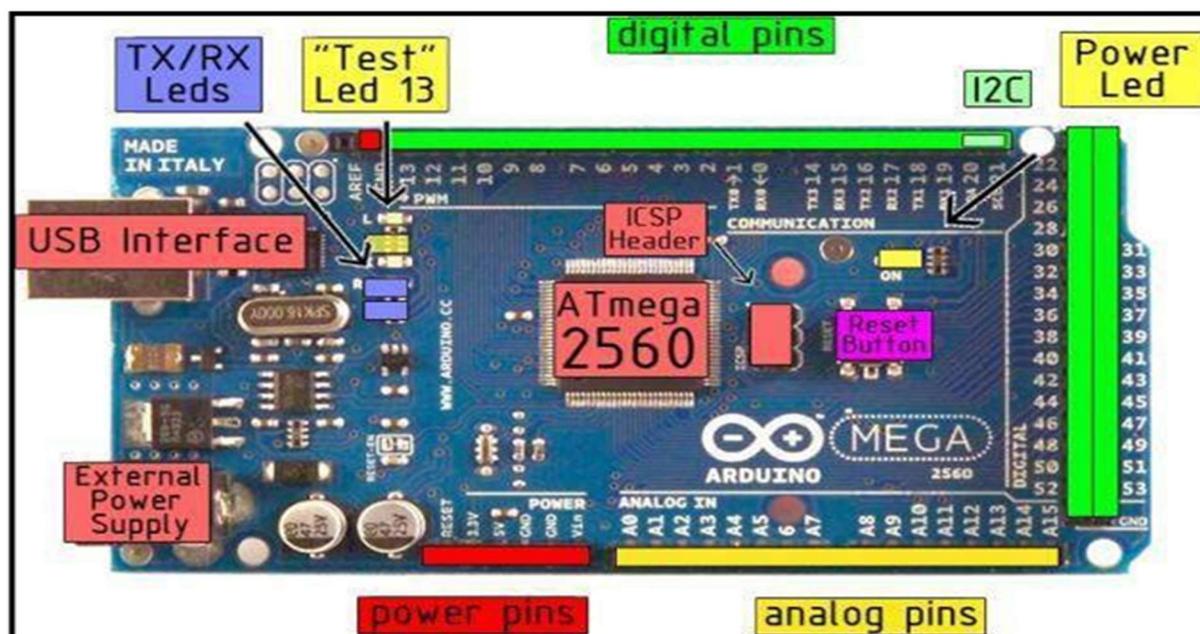


Figure 2.13 : Brochage de l'Arduino Mega 2560.

2.5.6.1 Spécifications techniques

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limites)	6-20V
Broches d'E/S numériques	54 (dont 14 fournissent une sortie PWM)
Broches d'entrée analogique	16
Courant continu par broche E/S	40 mA
Courant continu pour une broche 3.3V	50 mA
Mémoire Flash	256 KB dont 8 KB utilisés par le bootloader
SRAM	8 KB
EEPROM	4 KB
Vitesse d'horloge	16 MHz

Tableau 2.2 : Spécification technique d'Arduino Mega 2560.

2.6 Bluetooth

26.1 Comment se connecter avec le Bluetooth

La première étape pour se connecter en Bluetooth est d'activer le protocole sur son smartphone. L'icône du Bluetooth s'affiche alors en haut de l'écran. Ensuite, il faut télécharger l'application de l'objet connecté à commander et l'appairer. Une notification précise quand les deux appareils sont appairés. Le smartphone peut à ce moment prendre le contrôle sur l'autre périphérique.

La plupart du temps, les deux appareils échangent automatiquement un code de reconnaissance, qui est mémorisé pour les échanges ultérieurs. Pour les produits ne disposant pas du Bluetooth, il est possible de les raccorder au réseau grâce à un adaptateur USB, qui leur octroie cette fonctionnalité.

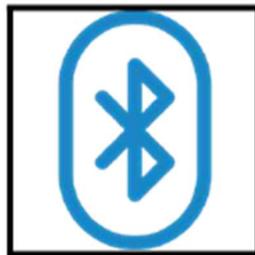


Figure 2.14 : Icon Bluetooth

Le Bluetooth définit un standard de communication développé en 1994 par le fabricant suédois Ericsson. Cette technologie, basée sur l'utilisation d'ondes radio UHF, permet une connexion entre plusieurs périphériques et l'échange bidirectionnel de données et de fichiers sur une très courte distance. Il fonctionne sur les fréquences comprises entre 2.4 GHz et 2.483 GHz.

Le principal avantage du Bluetooth réside dans le fait de pouvoir réaliser une connexion entre deux appareils sans aucune liaison filaire. Très vite, le Bluetooth a investi le monde de la téléphonie mobile et de l'informatique.

2.6.1 Bluetooth HC05

Le module Bluetooth HC-05 est un module SPP (Serial Port Protocol) Bluetooth facile à utiliser, conçu pour la mise en place d'une connexion sans fil transparente. Sa communication se fait via une communication ce qui permet une interface facile avec un contrôleur ou un PC.

Le module Bluetooth HC-05 HC-05 offre un mode de commutation entre le mode maître et le mode esclave, ce qui signifie qu'il est capable de n'utiliser ni recevoir ni transmettre des données. [8]

Spécification :

- Modèle : HC-05
- Tension d'entrée : DC 5V
- Méthode de communication : Communication en série.
- Le mode maître et esclave peut être changé.

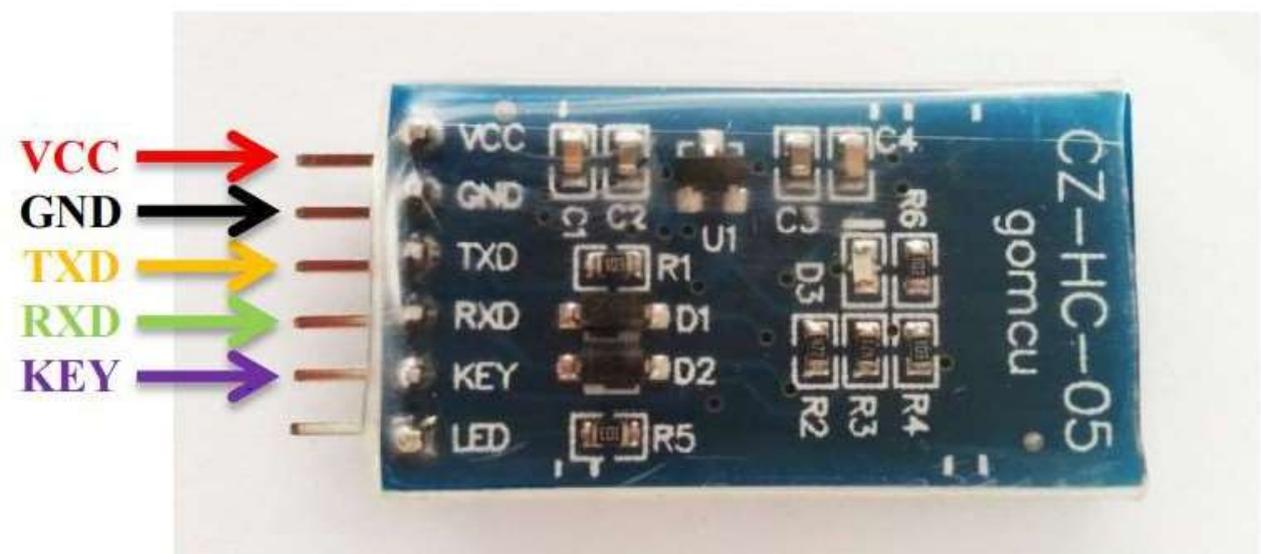


Figure 2.15 : Broche de Bluetooth HC05.

La broche (PIN)	Description	Fonction
VCC	+5V	Connecter à +5V
GND	Terre (Ground)	Connecter à la terre
TXD	UART_TXD, Signal série Bluetooth envoi de signaux PIN	Connexion avec le MCU (Microcontrôleur et autres) PIN RXD.
RXD	UART_RXD, Réception du signal réception de signaux PIN	Connexion avec le MCU (Microcontrôleur et autres) PIN TXD.
KEY	Interrupteur de mode entrée	S'il s'agit d'un niveau d'entrée bas ou d'une connexion à l'air, le module est en mode appairage ou communication. Si le niveau d'entrée est élevé le module passe en mode AT

Tableau 2.3 : Fonction et description de chaque broche de HC05.

Voici un exemple d'interface entre Arduino UNO et un PC via le module Bluetooth HC-05. Dans cet exemple, le mode de communication est utilisé.

- VCC → Arduino 5V
- GND → Arduino GND
- TXD → Arduino Pin RX
- RXD → Arduino Pin TX
- KEY → Se connecter à l'air pour le mode de communication

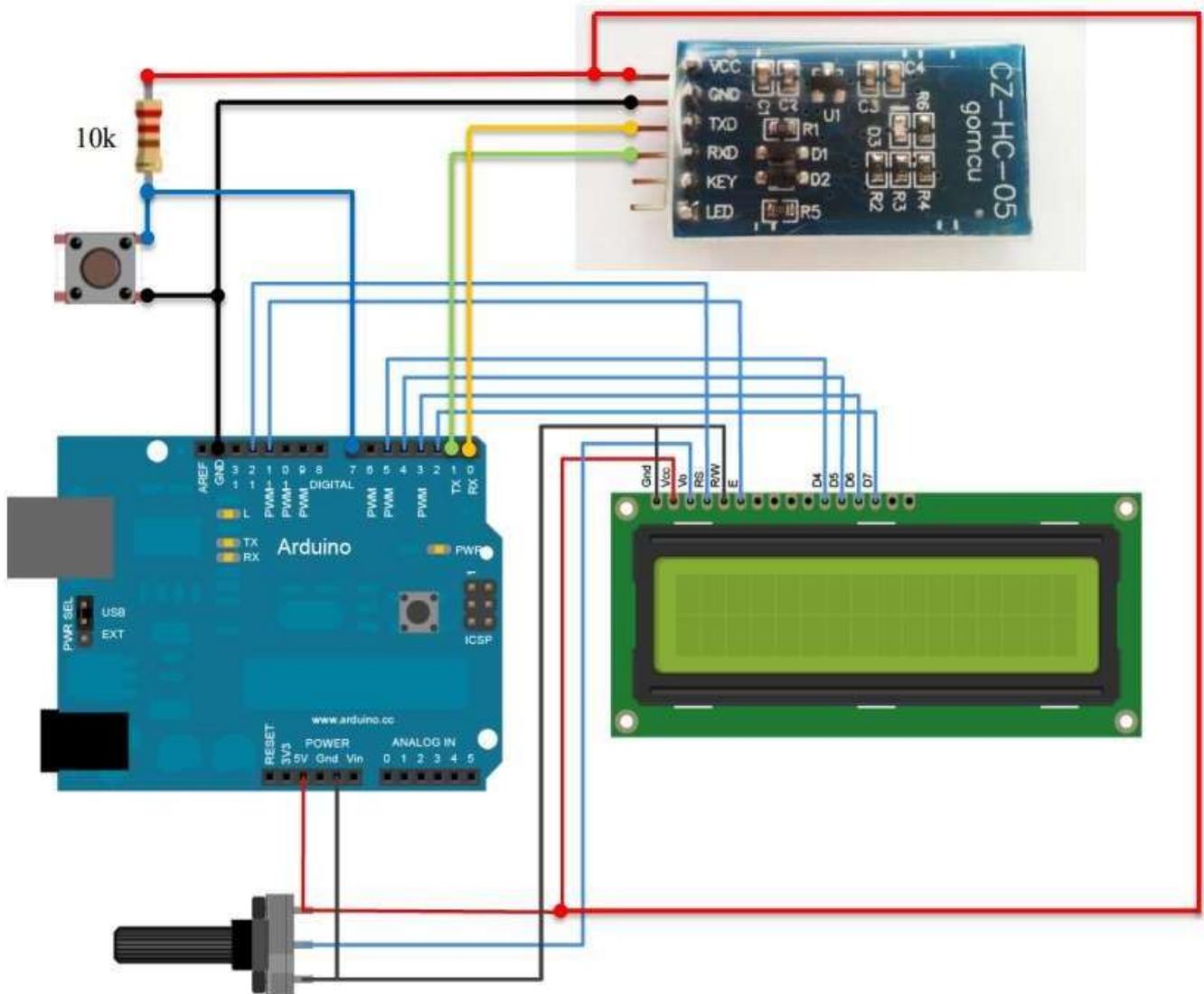


Figure 2.16 : Connexion hardware entre le module Bluetooth HC-05 et Arduino UNO.

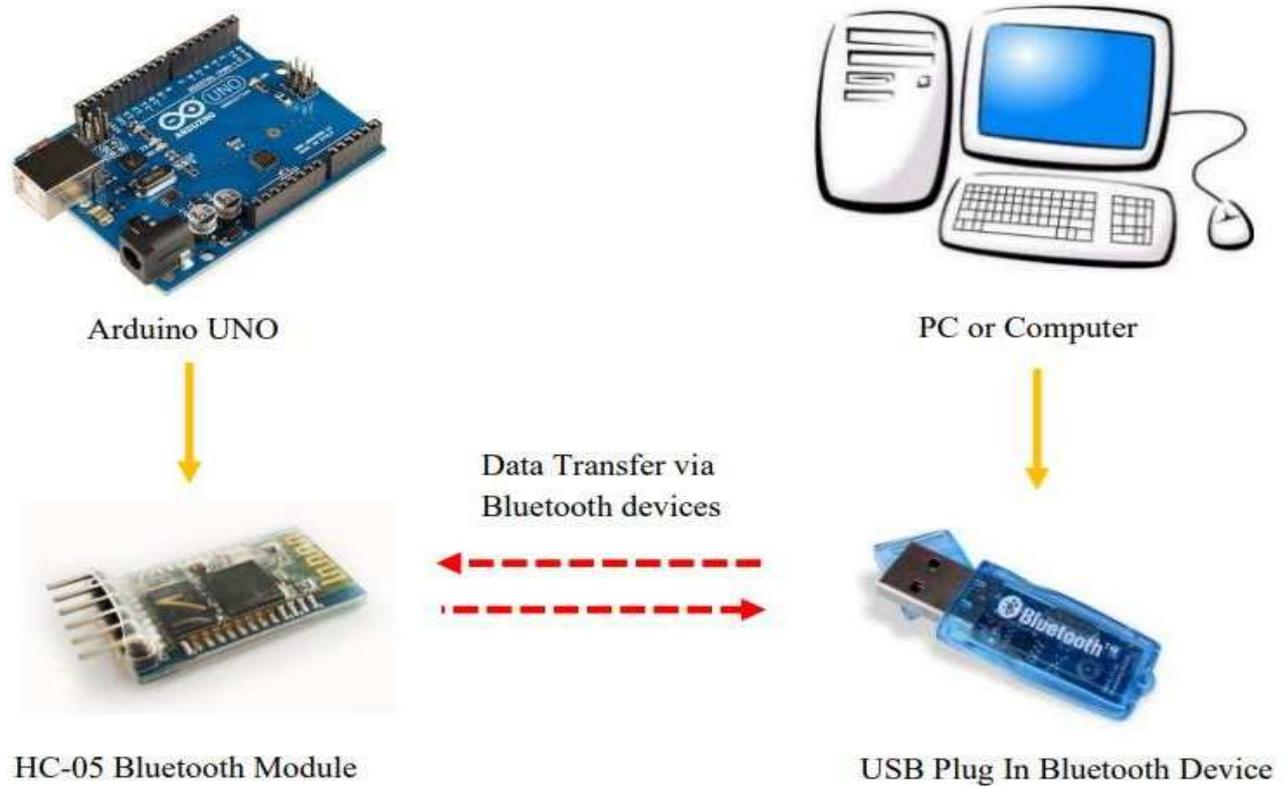


Figure 2.17 : transfert de données entre Arduino UNO et PC via des dispositifs Bluetooth

2.7 Conclusion

La description de tous les composants électroniques nécessaires à l'élaboration de notre projet a été effectuée dans ce chapitre.

Chapitre 3 :

Outils logiciels et Réalisation

3.1 Logiciel MikroC

3.1.1 Qu'est-ce qu'un MikroC PRO pour PIC

MikroC Pro est un compilateur Embedded-C qui vous permet de convertir un code écrit en langage C en langage machine. MikroC Pro pour Pic est utilisé pour la programmation des microcontrôleurs pic en c. Avant d'apprendre comment programmer un microcontrôleur PIC en C.

Il est nécessaire de connaître une introduction de base sur les microcontrôleurs et les microcontrôleurs pic. Ensuite il sera facile de comprendre la programmation des microcontrôleurs PIC.

3.1.2 Comment programmer un microcontrôleur image avec MikroC ?

MikroC Pro est un compilateur Embedded-C qui vous permet de convertir un code écrit en langage C en langage machine. MikroC Pro pour Pic est utilisé pour la programmation des microcontrôleurs PIC en c. Compilateur MikroC Pro pour la programmation des microcontrôleurs PIC.

Type	Taille en octets	Range
signed int	2	-32768 à 32767
unsigned int	2	0 à 65535

Tableau 3.1 : Les valeurs de type de variable signed et unsigned.

3.1.3 Qu'est-ce que PIC SimLab

PIC SimLab est un émulateur en temps réel de cartes de développement avec débogueur MPLABX/avr-gdb intégré. PIC SimLab supporte certains microcontrôleurs de picsim, certains de simard et certains de uCsim. PIC SimLab à une intégration avec MPLABX/Arduino IDE pour la programmation des microcontrôleurs des cartes.

3.1.4 Pourquoi utiliser le compilateur MikroC

C'est l'un des compilateurs les plus populaires utilisés pour la programmation des microcontrôleurs Pic. Sa popularité auprès des étudiants et des amateurs s'explique par un certain nombre de raisons. Tout d'abord, il offre une interface conviviale pour les débutants. Deuxièmement, contrairement au compilateur MPLAB, seules quelques étapes simples sont nécessaires pour créer un nouveau projet. Troisièmement, lorsque vous pouvez créer un nouveau projet, il inclut automatiquement toutes les ressources requises pour un microcontrôleur sélectionné. En plus de cela, il a beaucoup de bibliothèques intégrées pour tous les modules et périphériques de Pic MCUs tels que ADC, SPI, I2C, UART, et bien d'autres.

3.1.5 Créer un nouveau projet avec MikroC

Après avoir téléchargé le fichier d'installation depuis le site officiel, installez le logiciel en cliquant sur le fichier d'installation. Le processus est assez simple.

- Après l'installation, lancez le logiciel à partir de l'icône du bureau qui ressemble à celle-ci. Double-cliquez sur ce lien et le logiciel s'ouvrira.
- Une fois que le compilateur MikroC démarre, vous verrez un écran comme celui donné ci-dessous.

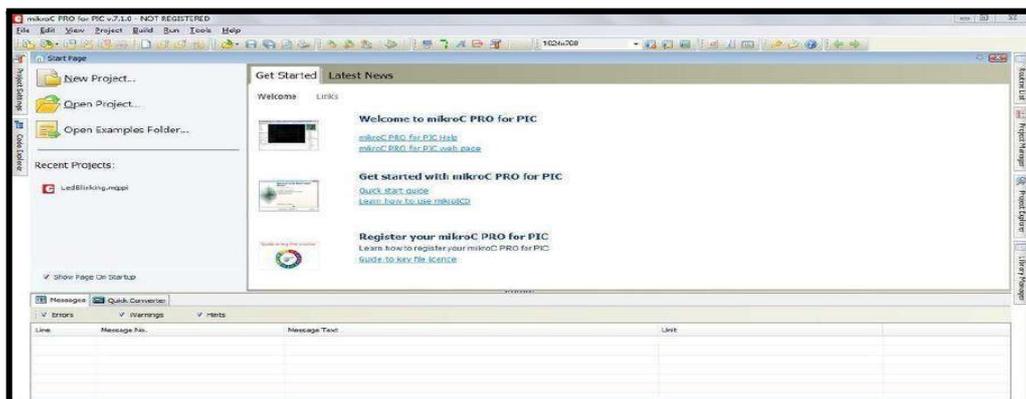


Figure 3.1 : L'interface du mikroC.

- Pour créer votre premier projet, cliquez sur "Nouveau projet". Vous pouvez également créer un nouveau projet à partir de la barre de menu en allant sur Projet>>Nouveau projet.

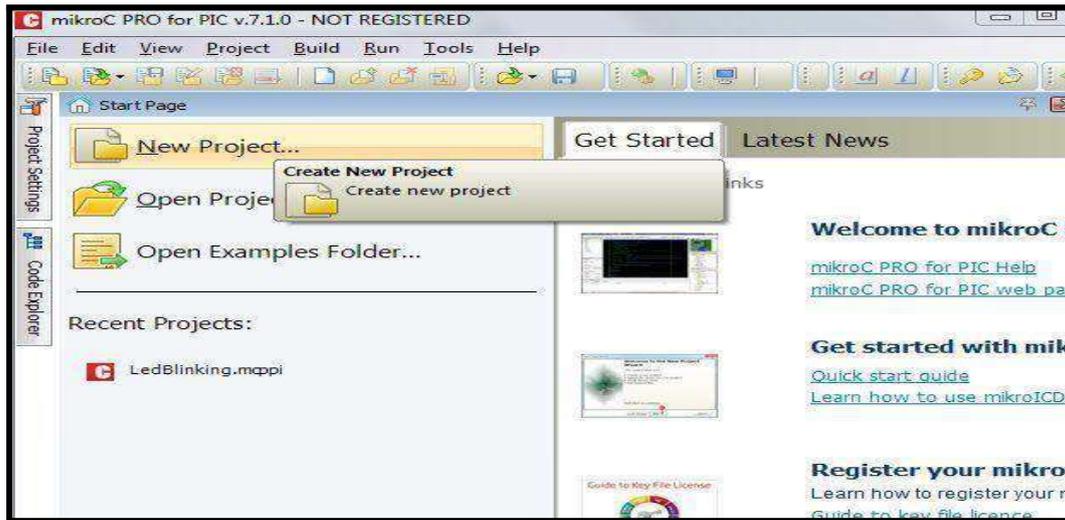


Figure 3.2 : Création un nouveau projet sur mikroC.

- La fenêtre "Assistant nouveau projet" s'ouvre. Sélectionnez un projet standard et cliquez sur suivant. Nous pouvons également créer des projets basés sur TFT avec le compilateur MikroC. Mais dans ce tutoriel, nous allons créer un projet standard uniquement. Ensuite, cliquez sur le bouton "Next".
- Tapez le nom du projet, le dossier où vous voulez sauvegarder votre projet, votre microcontrôleur PIC, pour cet exemple, c'est le P18F46K22 (pour la simulation, nous pouvons utiliser n'importe lequel mais nous devons le sélectionner dans Proteus) et l'horloge du dispositif (réglée sur 32MHz). Cliquez ensuite sur Next. Avant de sélectionner la fréquence.

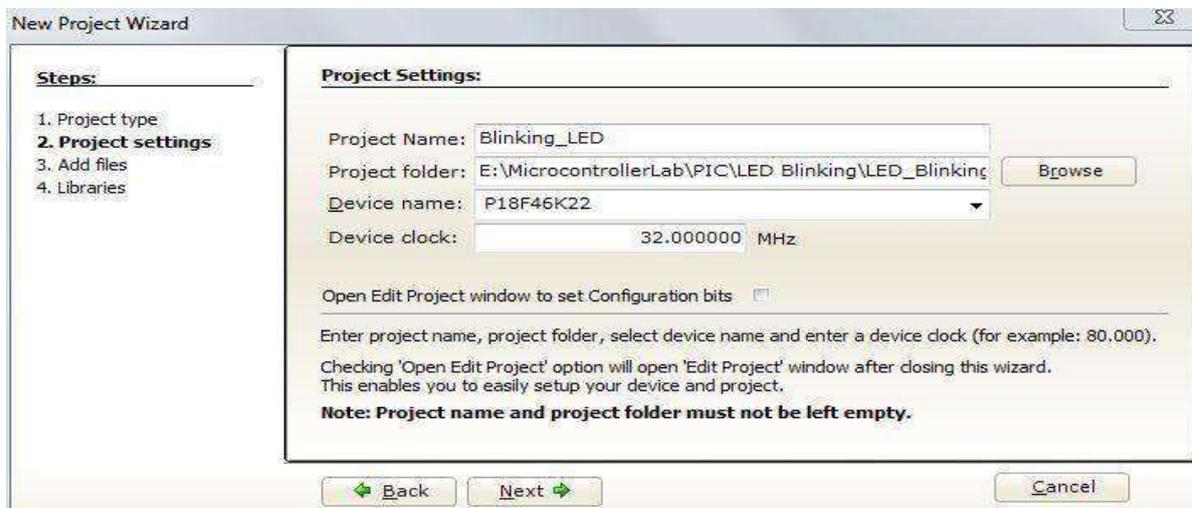


Figure 3.3 : Choix de PIC et fréquence sur mikroC.

- Cliquez à nouveau sur suivant. Nous n'avons pas besoin d'ajouter de fichiers. Cette option est utilisée pour ajouter des fichiers de programme c existants à notre projet. Par exemple, nous avons déjà un code de microcontrôleur pic que nous voulons inclure dans notre nouveau projet. Depuis cette fenêtre, nous pouvons ajouter ce fichier au nouveau projet. Mais pour l'instant, laissez-le tel quel et passez à la suite.
- Comme nous l'avons mentionné précédemment, le compilateur MikroC supporte de nombreuses bibliothèques intégrées. En utilisant cette option, nous pouvons sélectionner les bibliothèques ou inclure toutes les bibliothèques dans le projet. Par exemple, si nous utilisons uniquement les modules de communication ADC et UART dans le projet, pour l'instant, sélectionnez "Inclure toutes" les bibliothèques et cliquez sur Terminer.
- Une fenêtre comme celle présentée ci-dessous apparaît. C'est l'endroit où nous allons écrire notre programme. Nous écrivons tout dans cette fenêtre principale du programme.

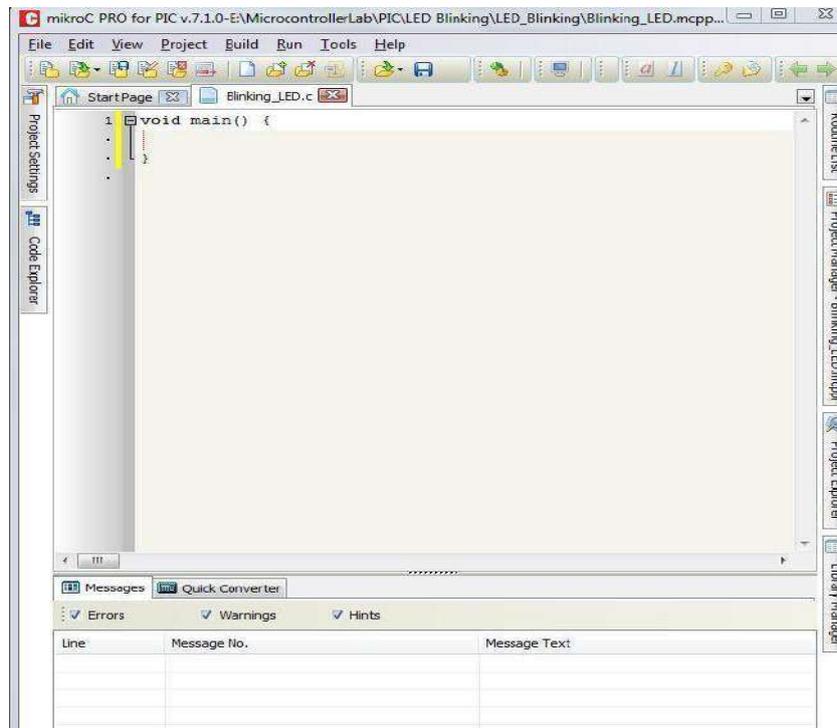


Figure 3.4 : Zone de code sur mikroC.

3.1.6 Définir les bits de configuration

Dans le compilateur mikroC, il est très facile de définir les bits de configuration. Pour définir les bits de configuration. Allez dans les projets dans la barre de menu et sélectionnez l'option Edit Project comme indiqué dans la figure ci-dessous :

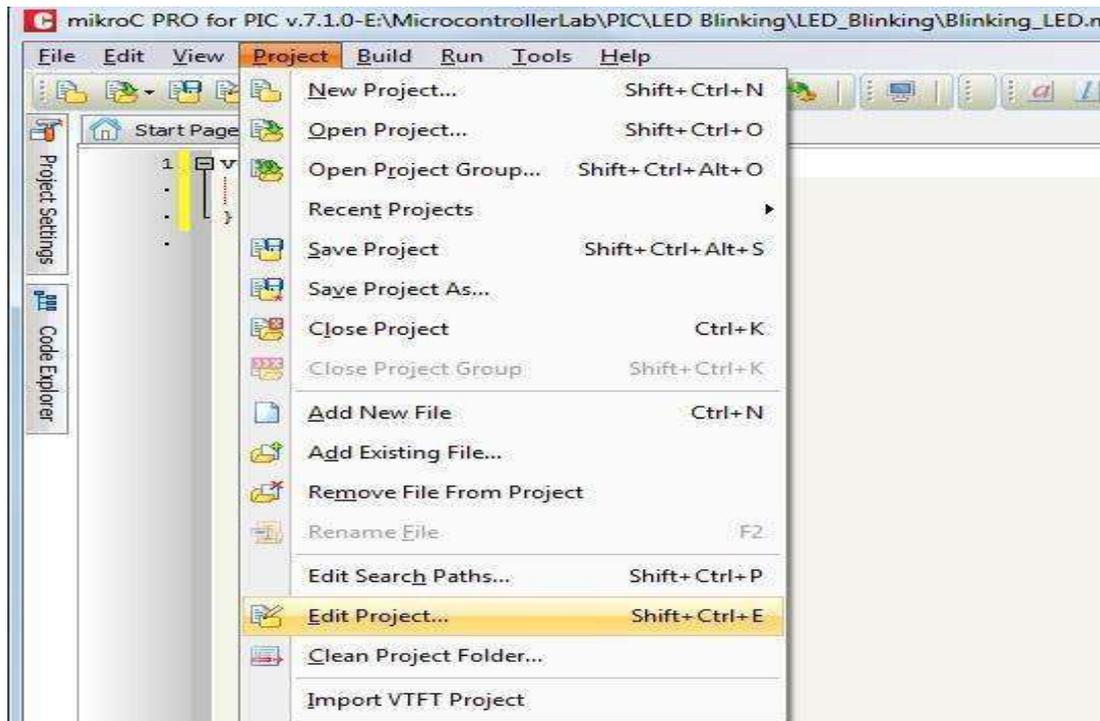


Figure 3.5 : Paramètres du projet sur mikroC.

Dans la fenêtre ouverte, sélectionnez votre MCU (nom du MicroContrôleur) et la fréquence d'horloge du MCU. En outre, sur le panneau de gauche, nous pouvons sélectionner les paramètres pour d'autres bits de configuration tels que la sélection de l'oscillateur, 4x PLL, Brown out Reset et la broche MCLR. Vous devez définir ces paramètres dans le menu déroulant en fonction de vos besoins, puis cliquer sur OK.

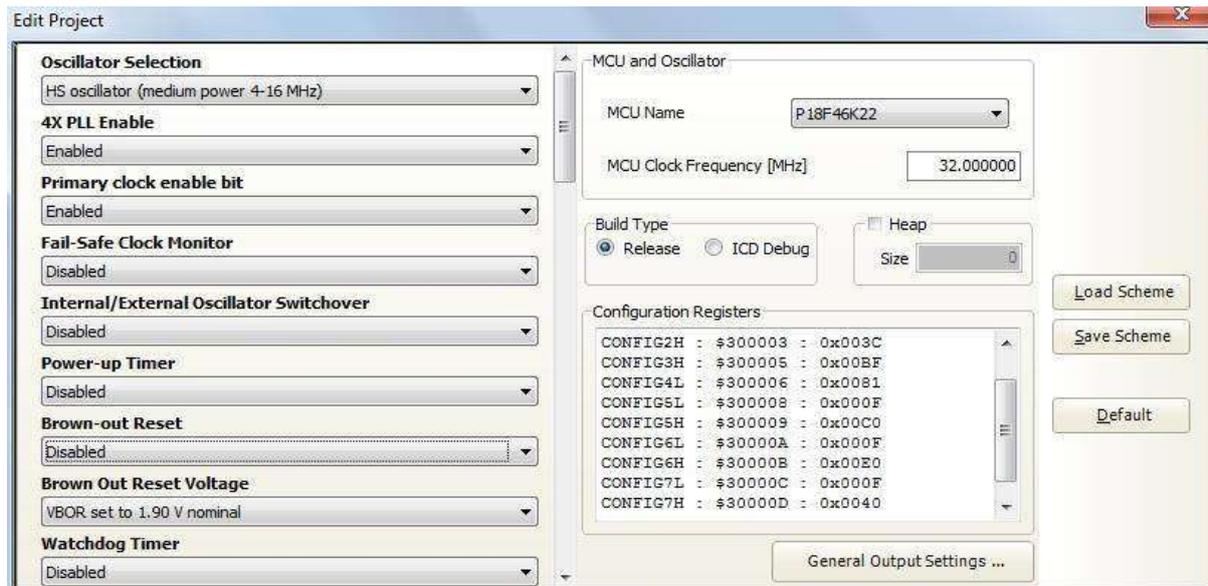


Figure 3.6 : Fenêtre de modification du projet sur mikroC.

3.1.7 Utilisation du Langage MikroC

Le langage mikroC pour PIC a trouvé une large application pour le développement de systèmes embarqués sur la base de microcontrôleurs. Il assure une combinaison de l'environnement de programmation avancée IDE (Integrated Development Environment), et d'un vaste ensemble de bibliothèques pour le matériel, de la documentation complète et d'un grand nombre d'exemples.

Après la compilation, le fichier source est sauvegardé avec l'extension .C. Une fois que le programme source est figé, le compilateur mikroC donne une facilité du programmer le pic directement en transformer le programme source en un programme exécutable "binaire".

L'extension du fichier sera alors. HEX (hexadécimal)

3.1.8 Structure d'un programme en mikroC

La structure la plus simple d'un programme en mikroC, c'est le programme représenté dans le code source (figure 17), qui nous permettra de faire clignoter une LED connectée au PORTB (par exemple bit 0 du PORTB) du microcontrôleur PIC avec une période de 2 secondes (1 seconde allumée et une seconde éteinte).

```

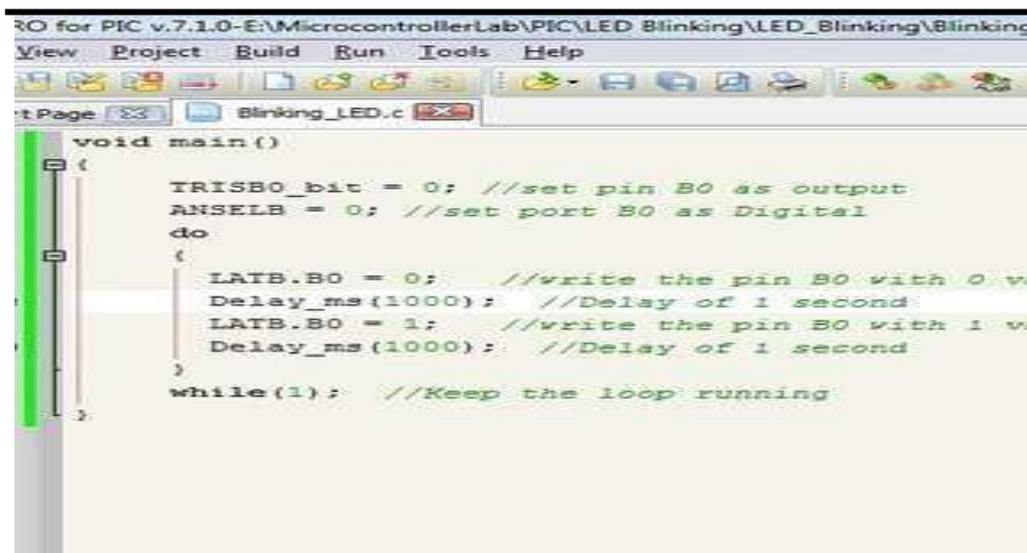
/*****
LED clignotante
=====
Ce programme fait clignoter une LED connectée à la broche RB0 du PIC
 16F84A
*****/
void main( )
{
for( ; ; ) // Boucle sans fin
{
TRISB = 0; // Configuration du PORTB en sortie
PORTB.F0 = 0; // RB0 = 0
Delay_Ms(1000); // Pause d'une seconde
PORTB.F0 = 1; // RB0 = 1
Delay_Ms(1000); // Pause d'une seconde
} // Fin de la boucle
}

```

Figure 3.7 : Examinons le fichier LED.c du code-source.

3.1.9 Exemple de code MikroC

Ecrivez le code suivant dans le logiciel



```

void main()
{
    TRISB0_bit = 0; //set pin B0 as output
    ANSELB = 0; //set port B0 as Digital
    do
    {
        LATB.B0 = 0; //write the pin B0 with 0 w
        Delay_ms(1000); //Delay of 1 second
        LATB.B0 = 1; //write the pin B0 with 1 w
        Delay_ms(1000); //Delay of 1 second
    }
    while(1); //Keep the loop running
}

```

Figure 3.8 : Code source sur mikroC.

3.1.10 Compiler le code avec MikroC

Après avoir écrit le code, il faut le compiler à partir de la barre de menu. Si tout est correct, il affichera un message de fin de compilation dans la fenêtre ci-dessous. Après cela, allez dans le dossier du projet où vous avez enregistré votre projet et localisez le fichier hex. Car nous allons utiliser ce fichier hex pour programmer le microcontrôleur PIC.

Après avoir obtenu le fichier hex, vous pouvez le télécharger dans le microcontrôleur PIC18F46K22 ou l'utiliser dans Proteus. Si vous voulez le télécharger vers le matériel actuel

3.2 Logiciel Proteus

Proteus est un outil EDA (Exploratory Data Analysis) qui permet de simuler des microcontrôleurs et des périphériques. Proteus peut véritablement transformer une conception complète, du concept au produit. Proteus peut programmer en se basant sur le prototype virtuel directement.

Avec les dispositifs d'affichage et de sortie, les entrées et les sorties peuvent être vues après l'exécution des programmes. Proteus constitue un environnement complet de conception et de développement électronique.

Il est utilisé pour la conception de différents circuits sur PCB (printed circuit board) et la simulation de différents circuits. L'utilisation de Proteus pour tout projet de circuit électronique rend ce projet rentable et moins erroné en raison de la construction schématique sur le Proteus.

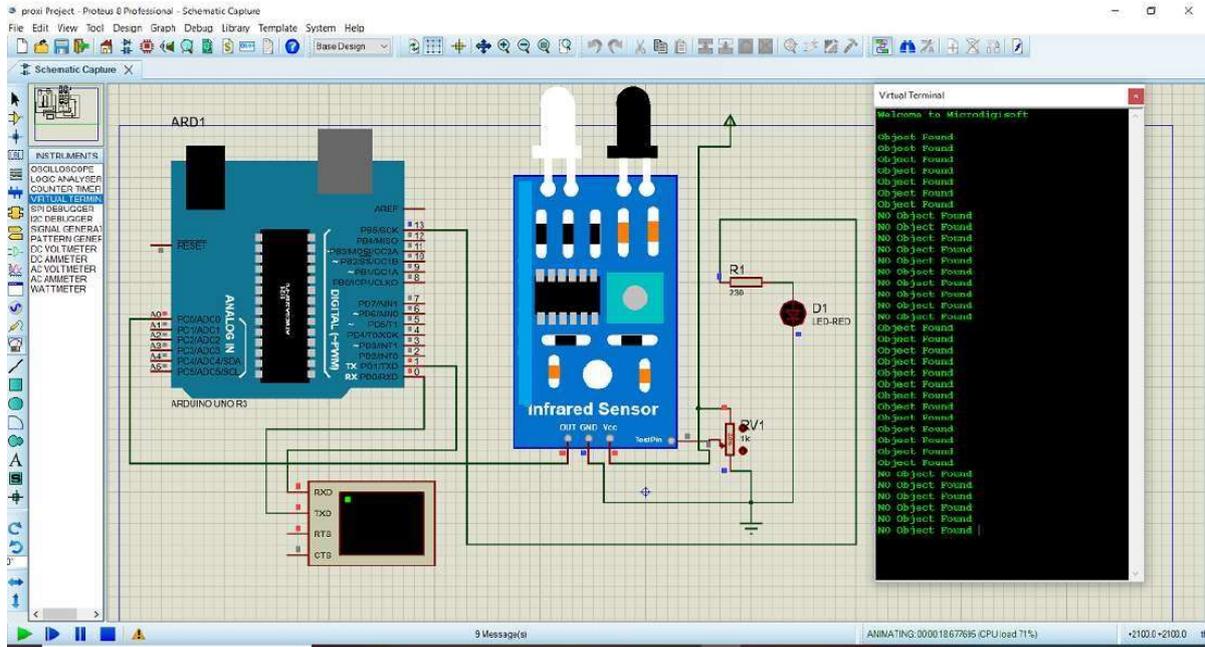


Figure 3.9 : Capture d'écran de logiciel proteus.

3.2.1 Proteus Simulation

Dessinez le circuit dans Proteus comme indiqué (utilisez le même contrôleur que vous avez sélectionné dans le logiciel précédemment). Réglez la fréquence du cristal à 8 MHz en double cliquant dessus et en entrant la fréquence dans la boîte de dialogue qui apparaît.

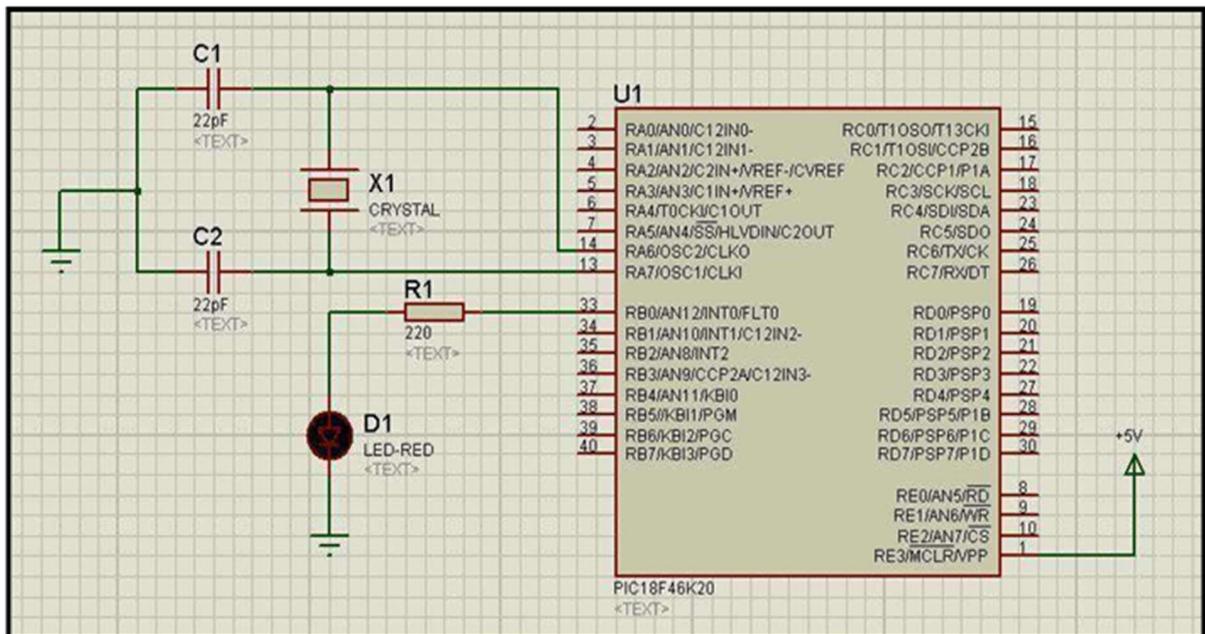


Figure 3.10 : Schéma électronique du PIC sur logiciel proteus.

Double-cliquez sur le MikroController dans Proteus, une boîte de dialogue apparaît comme suit.

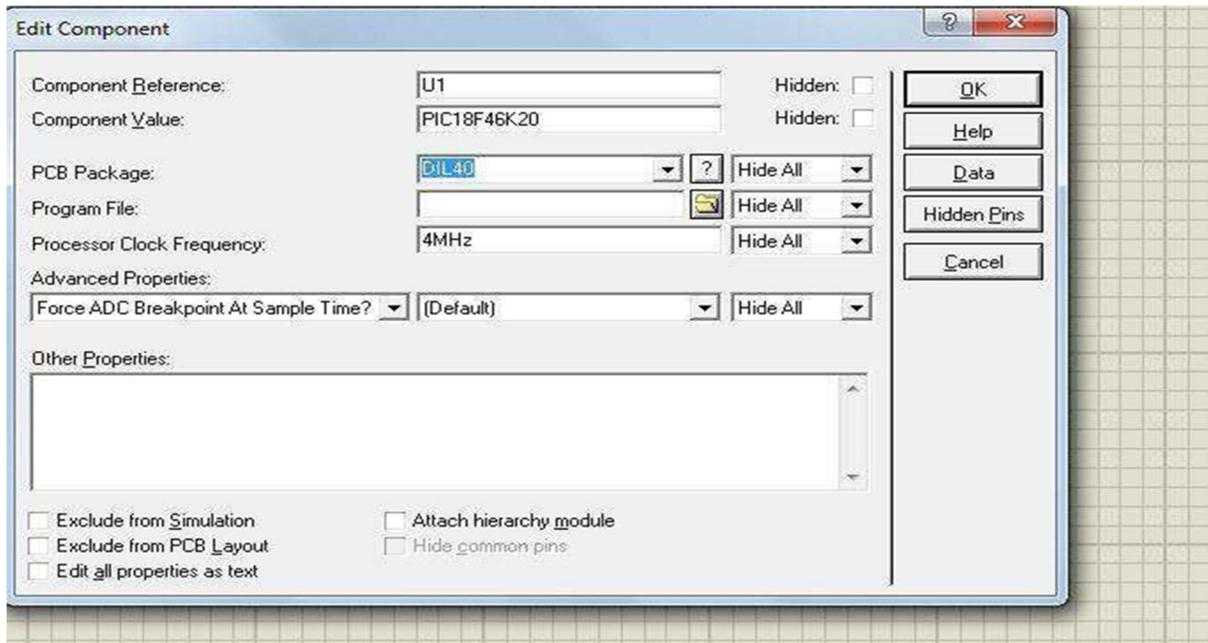


Figure 3.11 : Fenêtre de PIC pour ajouter le code.

Cherchez le fichier du programme dans la boîte et indiquez le chemin du fichier .hex créé après avoir construit le programme dans le logiciel MikroC. Il sera dans le même dossier, où vous avez enregistré le fichier .c. Changez également la fréquence d'horloge du processeur à 32MHz.

Cliquez sur le bouton de lecture en bas de Proteus.

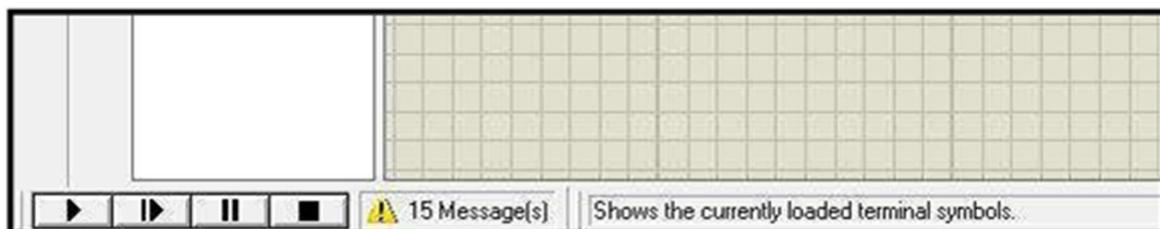


Figure 3.12 : Boite de simulation sur proteus.

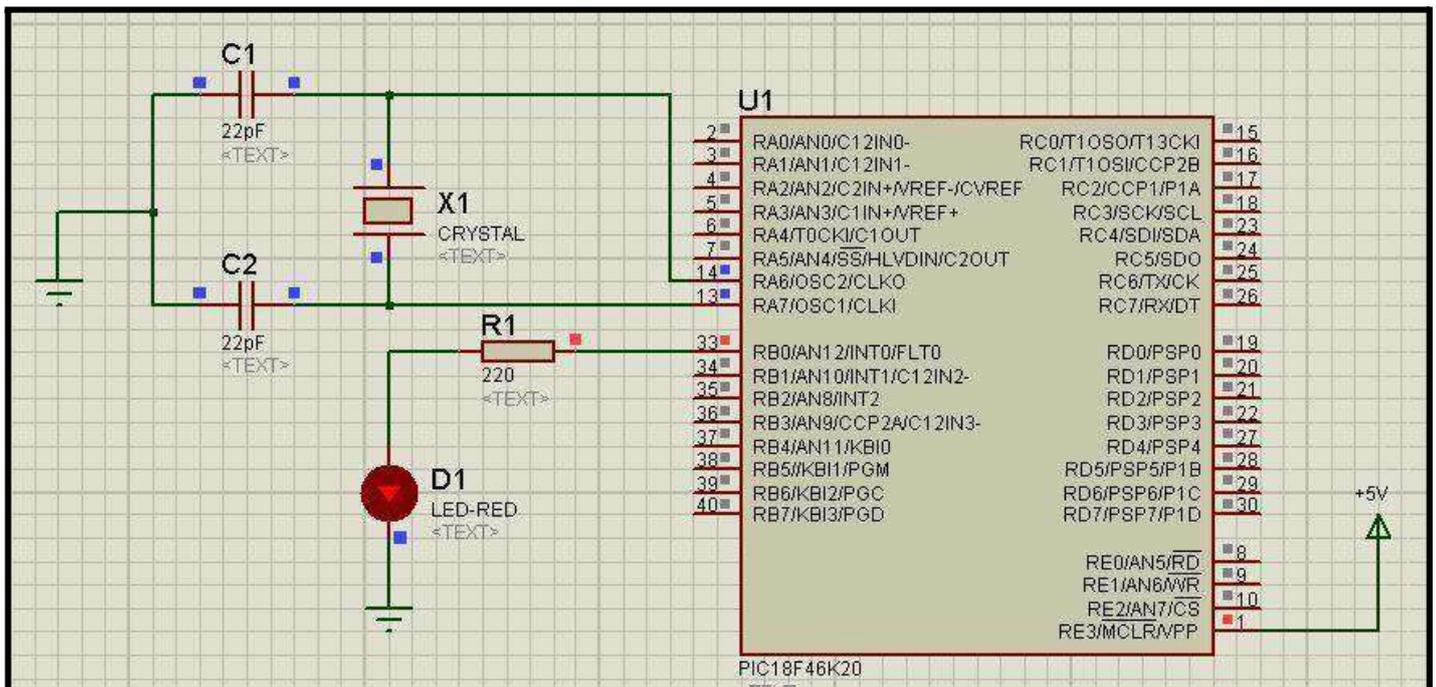


Figure 3.13 : Simulation du circuit électronique sur proteus.

3.2.2 Utilisation de l'application ISIS

Proteus Isis est un logiciel de développement et de simulation d'application via un environnement graphique simple et interactif.

Proteus est une suite logicielle de CAO électronique éditée par la société Labcenter Electronics. ISIS est la composante de Proteus qui permet la création de schémas et la simulation électrique. La grande force de ISIS est de pouvoir simuler le comportement d'un microcontrôleur (PIC, Atmel, 8051, ARM, HC11...) et de son interaction avec les composants qui l'entoure.

- Etape 1 : Ouvrir MPLAB : Démarrez MPLAB IDE en faisant Démarrer > Programme > Microchip MPLAB IDE>MPLAB IDE. On obtient alors la fenêtre ci-dessous.
- Etape 2 : Lancement du «Wizard» : Démarrez le « Project Wizard » en faisant Project>Project Wizard, vous obtenez alors la fenêtre ci dessous.

Pour continuer cliquez sur Suivant.

- Etape 3 : Choisissez le microcontrôleur que vous allez utiliser. Dans notre cas choisissez le PIC 16F84A.
- Etape 4 : Choisissez le langage de programmation que vous allez utiliser.

Dans notre cas on va programmer en assembleur donc choisissez le MPASM Assembler.

Pour continuer cliquez sur Suivant.

- Etape 5 : Nommez le projet : La fenêtre suivante vous permet de donner le nom « TP1 » à votre projet et d'indiquer dans quel répertoire vous voulez le ranger en cliquant sur Browse.

Note : Le nom du projet « TP1 » peut être différent du nom du dossier de travail « dossier TP1 ». Pour continuer cliquez sur Suivant.

- Etape 6 : Ajoutez les fichiers que vous allez utiliser dans votre projet Dans notre cas ne rien choisir, on va ajouter les fichiers manuellement.
- Etape 7 : Validez le projet : La fenêtre qui apparaît résume le projet que vous venez de créer et vous permet de vous assurer que vous n'avez pas fait d'erreurs.

Remarque :

- Le projet ainsi créé est vide, par conséquent il est nécessaire d'ajouter en particulier des fichiers sources (en assembleur). Il est toujours préférable que fichier source et projet soient dans le même répertoire.

- Une fois le fichier source (.asm) est édité utiliser (project add files to project) pour l'ajouter au projet.

- Pour la sauvegarde du projet utiliser (project save project)

3.3 MIT App Inventor

3.3.1 C'est quoi App Inventor

App Inventor est une plateforme en ligne conçue pour former aux concepts de la pensée computationnelle par le biais du développement d'applications mobiles. Les étudiants créent des applications en faisant glisser et en déposant des composants dans une vue de conception et en utilisant un langage de blocs visuels pour programmer le comportement de l'application.

- Site web pour réaliser des applications simples.
- Utile, facile à utiliser et gratuite !
- Langages : Java, Kawa, Scheme
- Excellent pour : faire comprendre, enseigner les bases, le dépannage et la persistance de la programmation.
- Possibilité d'utiliser votre propre application sur un android ou un émulateur.

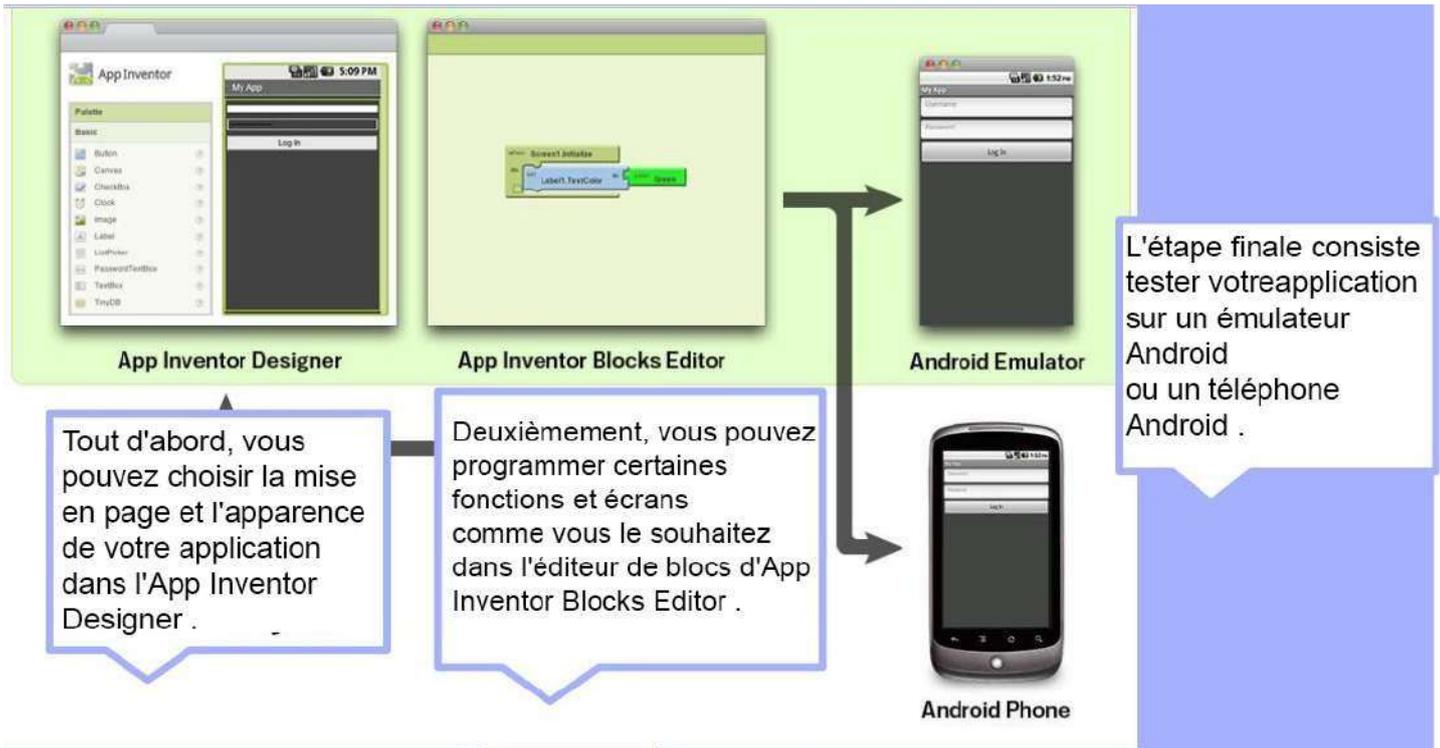


Figure 3.14 : Schéma de MIT app inventor.

3.3.2 Comment créer une application mobile gratuitement

Étape 1 : Ouvrez un compte Gmail si vous n'en avez pas.

Étape 2 : Ouvrez le lien <https://appinventor.mit.edu/> et connectez-vous à votre compte Gmail.

Étape 3 : Vous devez installer l'App Inventor Companion App (MIT AI2 Companion) sur notre appareil mobile qui aide à tester notre application en direct.

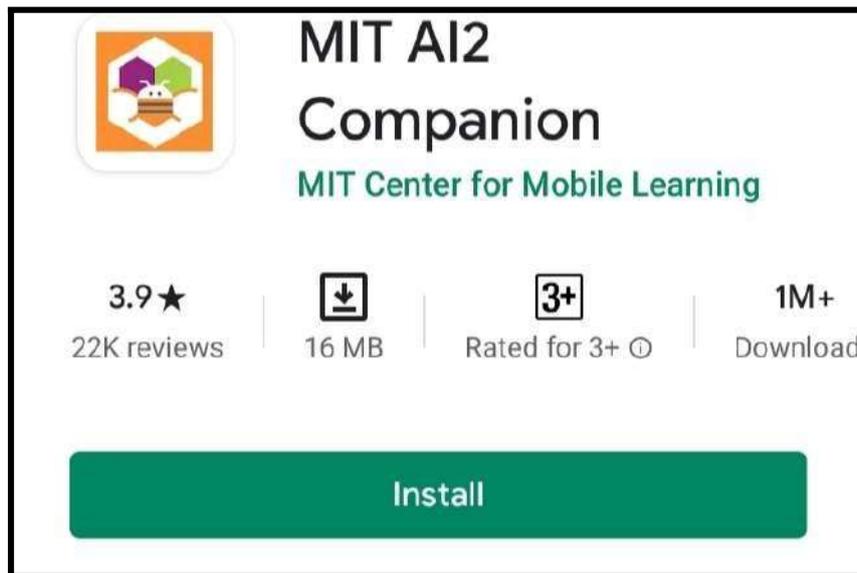


Figure 3.15 : MIT application sur play store.

Étape 4 : Nous devons connecter les deux appareils mobiles et les ordinateurs portables/de bureau doivent être connectés au même réseau WiFi.

Étape 5 : Pour commencer la construction de l'application, cliquez sur "Start New Project".

Étape 6 : Pour connecter votre appareil mobile, choisissez "Connecter" et "AI Companion" dans le menu supérieur.

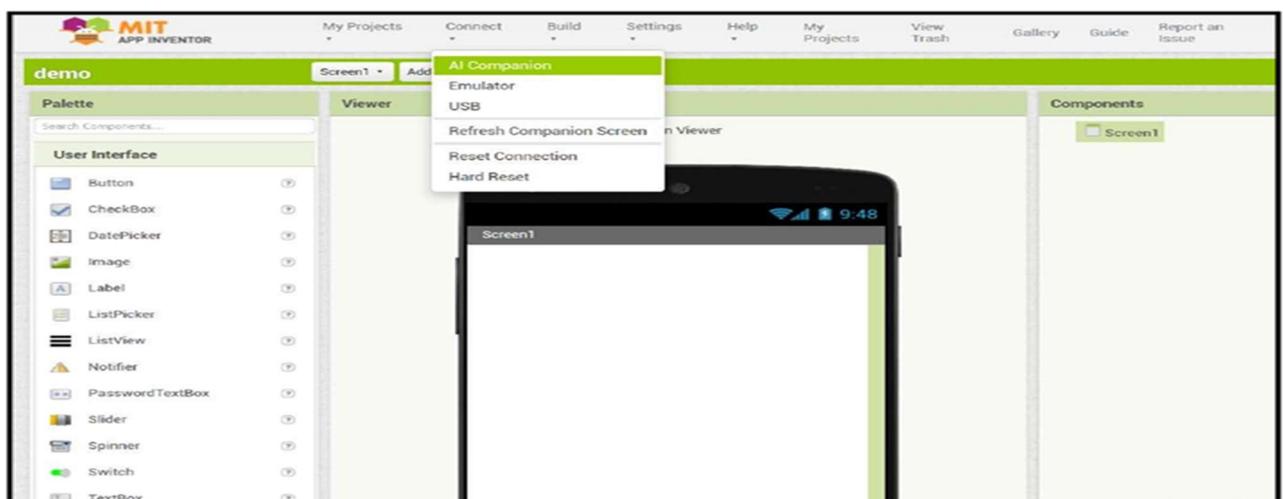


Figure 3.16 : L'interface de MIT inventor.

Étape 7 : Maintenant, pour connecter l'application MIT AI2 sur votre appareil et votre ordinateur de bureau/portable, scannez le code QR ou tapez le code à 6 chiffres qui apparaît sur l'écran de votre PC.

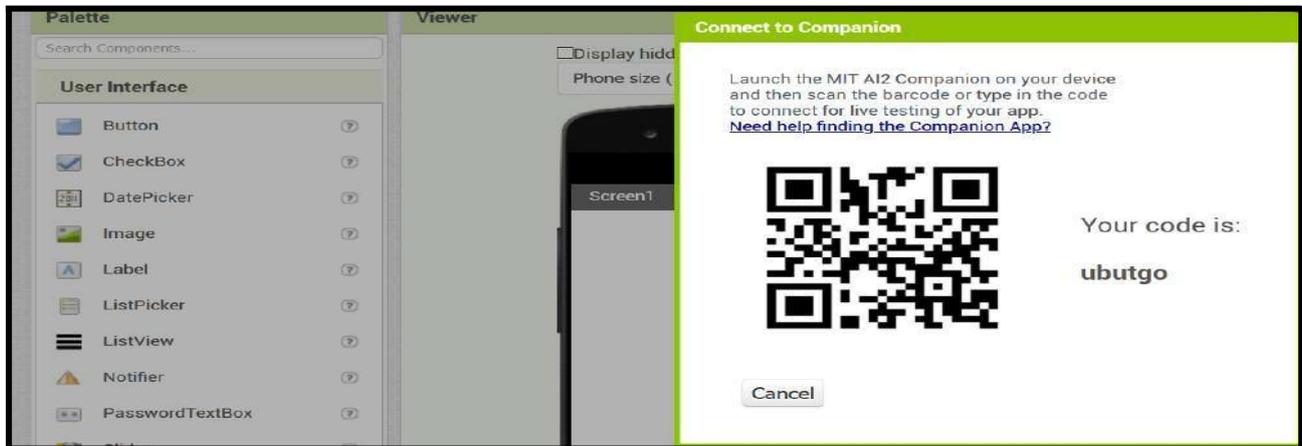


Figure 3.17 : Code QR MIT inventor.

Étape 8 : Maintenant vous pouvez voir l'application que vous construisez sur votre appareil.

3.3.3 Création d'un exemple simple

- Cliquer sur "Start new project".

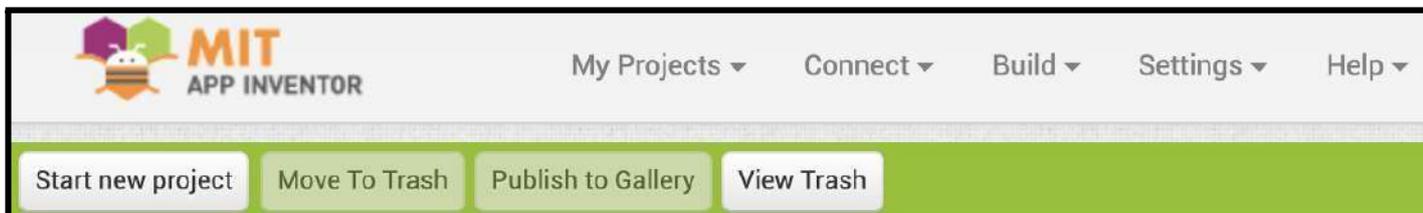


Figure 3.18 : Création d'un projet sur MIT inventor.

- Nommez le projet “HelloWorld”



Figure 3.19 : Nom du projet sur MIT inventor.

- Écriture de l'application HelloWorld

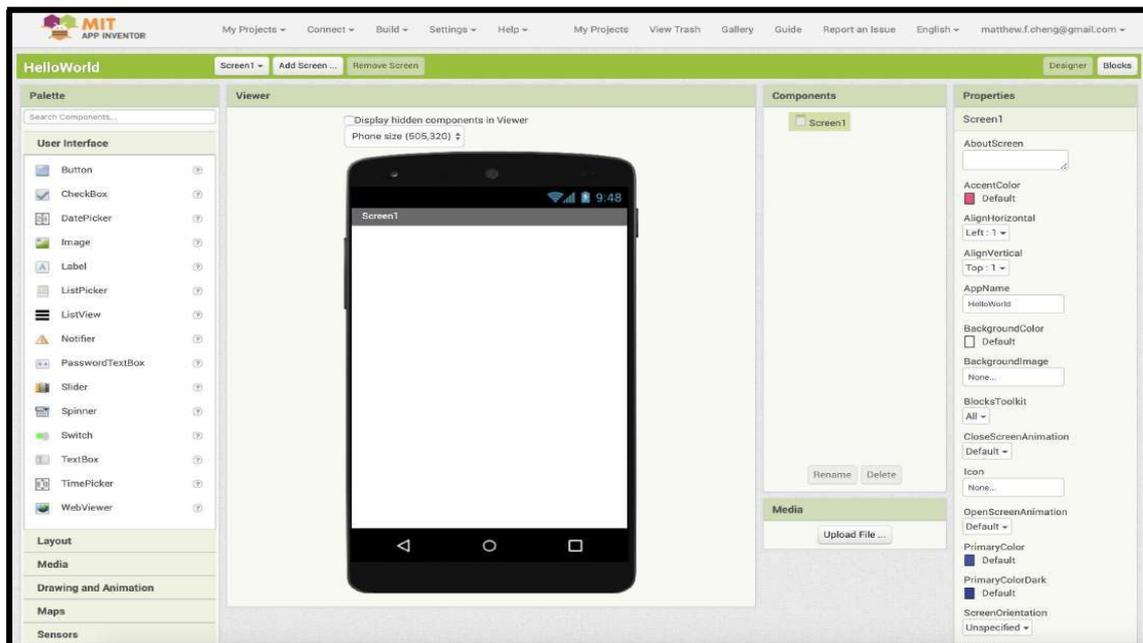


Figure 3.20 : Ecriture de code sur MIT inventor.

- Faire glisser une étiquette sur l'écran
- Changez le texte en Hello World !
- Vous remarquerez que le texte à l'écran se transforme en Hello World !

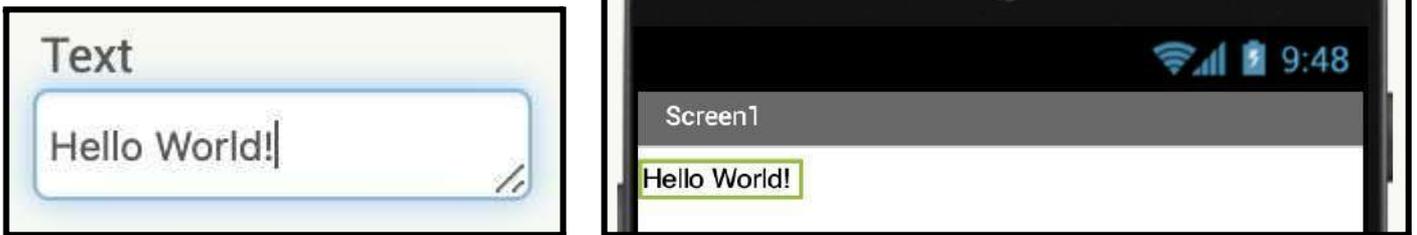


Figure 3.21 : Affichage de Hello World ! sur MIT inventor.

- Cliquez sur Connecter et sélectionnez AI Companion.
- Une fenêtre popup apparaît avec un code QR et un code de 6 lettres pour connecter votre application à votre appareil. Sur votre appareil, vous pouvez soit scanner le QR, soit saisir le code. Scanner le code est beaucoup plus rapide que d'essayer de le taper.

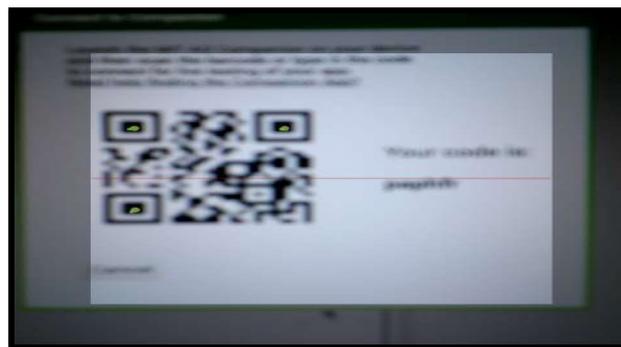


Figure 3.22 : Connexion par code QR sur MIT inventor.

Si tout se passe bien, vous verrez Hello World ! sur l'écran de votre appareil ! Tous les changements ou mises à jour de votre programme que vous effectuez sur le site web du MIT App Inventor apparaîtront instantanément sur votre appareil.



Figure 3.23 : Affichage de Hello World ! sur l'écran de smartphone.

3.3.4 Quel logiciel utiliser pour créer une application

Android Studio est l'IDE (Integrated Development Environment) officiel de Google pour les développeurs Android.

Le travail d'Android Studio est de fournir l'interface pour que vous puissiez créer vos applications et de gérer une grande partie de la gestion compliquée des fichiers dans les coulisses. Le langage de programmation que vous utiliserez est Java et il est installé séparément sur votre machine. Android Studio est simplement l'endroit où vous écrirez, éditez et enregistrez vos projets et les fichiers qui les composent.

3.4 Partie pratique

Pour la réalisation de notre projet, nous avons trois étapes à démontrer qui sont :

- 1- Le code de PIC en utilisant Arduino IDE.
- 2- Schéma du circuit sur ISIS Proteus.
- 3- Réalisation pratique du circuit électronique.

3.4.1 Code de PIC

```
#include <avr/interrupt.h>

#include <Adafruit_NeoPixel.h>

#include <EEPROM.h>

#define PIN          9

#define deli 500

int t1[]={1,0,0,1,1,0,1,0,1,0,0,1,1,0,1,0};
int t2[]={0,1,0,1,0,1,1,0,0,1,0,1,0,1,1,0};
int t3[]={0,0,1,0,1,1,1,0,0,0,1,0,1,1,1,0};
unsigned char lim0[11][8]=
{
{0x00,0x00,0x00,0xFF,0xFF,0x01,0x00,0xFF}, //int co,int inte,int
vit,int nbf;//unif
{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0x00}, //int co1,int co2,int
inte,int sens,int vit,int nbf;//merg
{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0xFF}, //int co1,int co2,int
inte,int sens,int vit,int nbf;//centr
{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0x00}, //int nbd,int inte,int
sens,int vit,int nbf;//multic
```

```

{0x00,0xFF,0x00,0x00,0xFF,0x01,0x00,0xFF}, //int co,int intel,int
inte2,int vit,int nbf;//pulsar

{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0xFF}, //int co,int intel,int
inte2,int vit,int nbf;//remp

{0x00,0x00,0x00,0xFF,0xFF,0x01,0x00,0x01}, //balan

{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0x01} ,//empil

{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0x01} ,//empil2

{0x00,0x00,0x00,0xFF,0x00,0x01,0x00,0x01}, //empil3

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}};//ofof

unsigned char lim1[11][8]=

{

{0x07,0x07,0x05,0xFF,0xFF,0x32,0x32,0xFF}, //int co,int inte,int
vit,int nbf;//unif

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0x50}, //int col,int co2,int
inte,int sens,int vit,int nbf;//merg

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0xFF}, //int col,int co2,int
inte,int sens,int vit,int nbf;//centr

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0x32}, //int nbd,int inte,int
sens,int vit,int nbf;//multic

{0x07,0xFF,0x0a,0x0a,0xFF,0x32,0x32,0xFF}, //int co,int intel,int
inte2,int vit,int nbf;//pulsar

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0xFF}, //int co,int intel,int
inte2,int vit,int nbf;//remp

{0x07,0x07,0x05,0xFF,0xFF,0x32,0x32,0x32}, //balan

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0x32}, //empil

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0x32}, //empil2

{0x07,0x07,0x05,0xFF,0x01,0x32,0x32,0x32}, //empil3

{0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01}};//ofof

unsigned char par[11][8]=

```

```

{
  {0x01,0x07,0x01,0xFF,0xFF,0x0a,0x02,0xFF}, //unif
  {0x01,0x04,0x01,0xFF,0x01,0x02,0x02,0x05}, //merg
  {0x01,0x02,0x01,0xFF,0x00,0x0a,0x02,0xFF}, //centr
  {0x00,0x04,0x01,0xFF,0x00,0x06,0x02,0x05}, //multic
  {0x00,0xFF,0x01,0x04,0xFF,0x0a,0x02,0xFF}, //pulsar
  {0x00,0x01,0x01,0xFF,0x00,0x05,0x02,0xFF}, //remp
  {0x00,0x01,0x01,0xFF,0xFF,0x05,0x02,0x20}, //balan
  {0x00,0x04,0x01,0xFF,0x00,0x05,0x02,0x01}, //empil
  {0x07,0x00,0x01,0xFF,0x00,0x05,0x02,0x01}, //empil2
  {0x07,0x00,0x01,0xFF,0x00,0x05,0x02,0x01}, //empil3
  {0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}}; //ofof

//col co2  int1 int2 sens vit  nbf  nbd

unsigned int trp=0;

unsigned char npx=1;

int scr=0;

int m=0,i,j=0,k,p,r=2,nn;

//char mm,nn0=0;

String inpstr = "",ss;

bool strFin = false;

int co,inte,vit,nbf;

int nbd;

int intel,inte2;

int addr = 0;

Adafruit_NeoPixel pixels(npx, PIN, NEO_GRB + NEO_KHZ800);

void sauve()

```

```
{  
  
addr=4;//1;  
  
for (i=0;i<=10;i++)  
  
{  
  
for (j=0;j<8;j++)  
  
{  
  
EEPROM.write(addr,par[i][j]);  
  
addr = addr + 1;  
  
delay(100);  
  
}  
  
}  
  
EEPROM.write(addr,scr);  
  
delay(100);  
  
}  
  
  
void lect()  
  
{  
  
addr=4;//1;  
  
for (i=0;i<=10;i++)  
  
{  
  
for (j=0;j<8;j++)  
  
{  
  
par[i][j]=EEPROM.read(addr);  
  
addr = addr + 1;  
  
delay(100);  
  
}  
  
}  
  
scr=EEPROM.read(addr);
```

```
    delay(100);
}

/*
void lect0()
{
    addr=100;
    for (i=0;i<=10;i++)
    {
        for (j=0;j<8;j++)
        {
            par[i][j]=EEPROM.read(addr);
            addr = addr + 1;
            delay(100);
        }
    }
    scr=EEPROM.read(addr);
    delay(100);
}
*/

// col  co2  intel  inte2  sens  vit  nbf  nbd
// 0x01,0x07,0x01,0xFF,0xFF,0x32,0x32,0xFF}, //unif
void unif()
{
    int b;

    nbf=par[0][6];

    b=par[0][2]*51;
```

```
if (b==51) b=5;

for(k=0;k<nbf;k++)
{
if (par[0][0]<=par[0][1])
{
for (co=par[0][0];co<=par[0][1];co++)
{
vit=5*par[0][5];
for (i=0;i<vit;i++)
{
for (p=0;p<=npv;p++)
{
pixels.setPixelColor(p,
pixels.Color(t1[co]*b,t2[co]*b,t3[co]*b));
if (strFin)
{
strFin=false;
return 1;
}
}
pixels.show();
}
}
}
else
{
for (co=par[0][0];co>=par[0][1];co--)
```

```
vit=5*par[0][5];  
  
for (i=0;i<vit;i++)  
{  
    for (p=0;p<=npX;p++)  
    {  
        pixels.setPixelColor(p,  
pixels.Color(t1[co]*b,t2[co]*b,t3[co]*b));  
  
        if (strFin)  
        {  
            strFin=false;  
            return 1;  
        }  
    }  
    pixels.show();  
}  
}  
}
```

3.4.2 Schéma du circuit

On utilise les composants électroniques suivants pour dessiner le schéma du circuit sur ISIS Proteus :

- Régulateur de tension de 5V (7805)
- Condensateur (CAP)
- Résistance (RES)
- Diode (DIODE)

- Transistors IRFZ44N (IRLZ44N)
- Des afficheurs de LED (LED-BARGRAPH)
- Microcontrôleur (PIC16F628A)

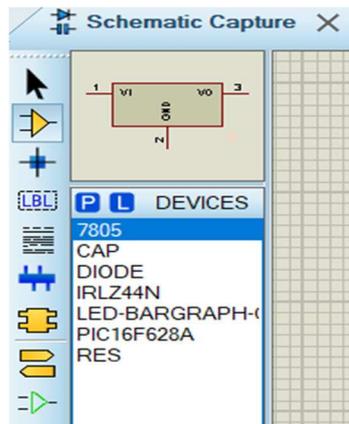


Figure 3.24 : Composants électroniques utilisés sur ISIS Proteus.

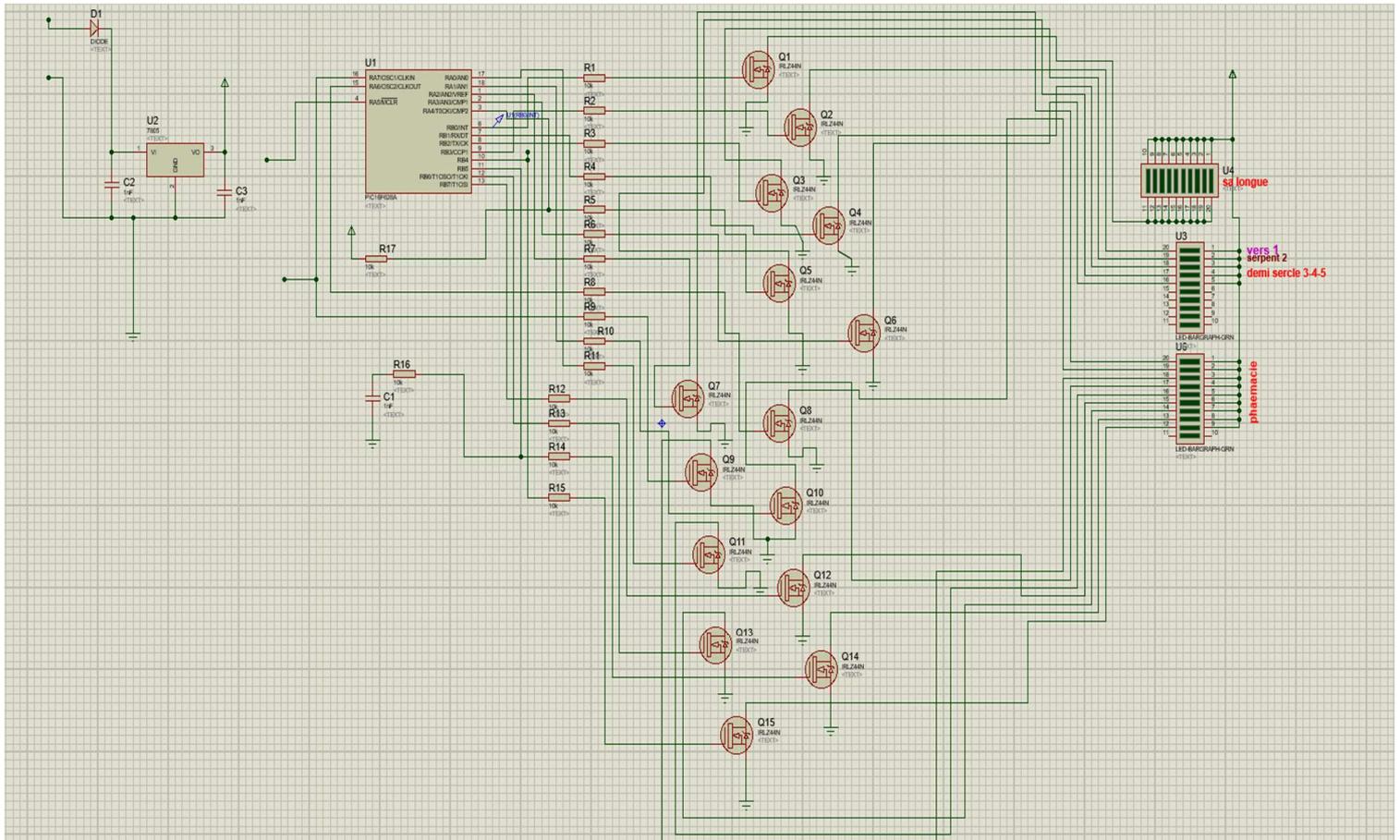


Figure 3.25 : Schéma du circuit d’enseigne lumineuse sur ISIS Proteus.

3.4.3 Réalisation pratique

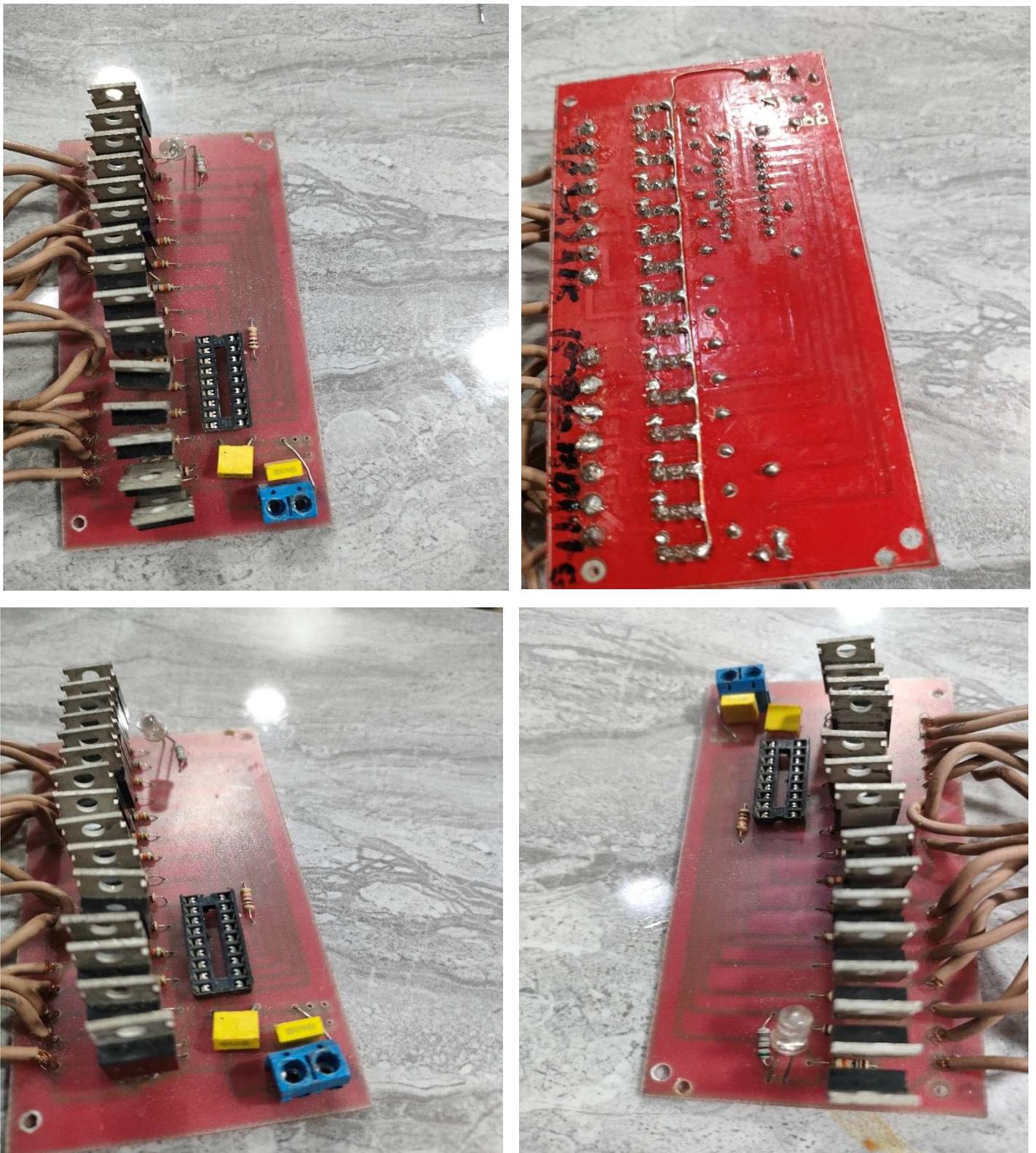


Figure 3.26 : Circuit électronique réalisé.

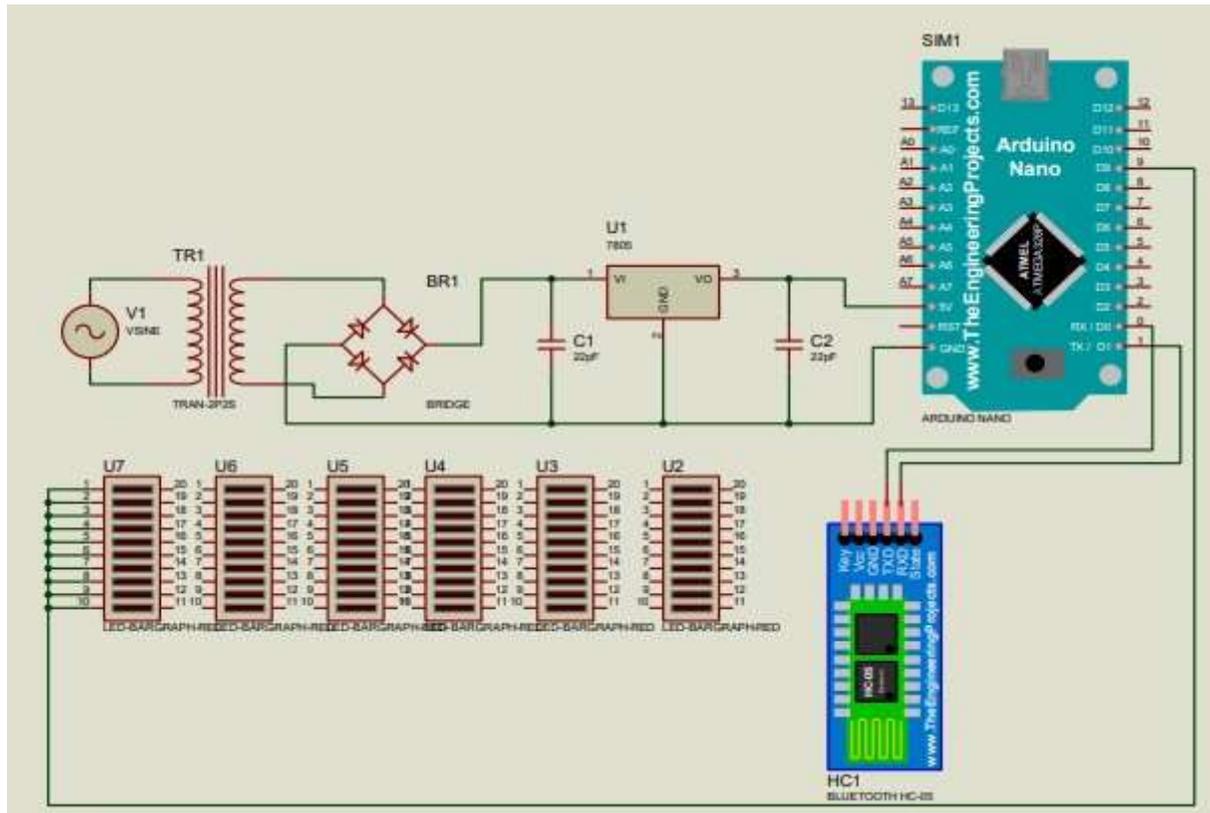


Figure 3.27 : Schéma du circuit de commande d’enseigne lumineuse avec arduino nano et HC05 sur ISIS Proteus.

3.5 Conclusion

La commande d’enseigne lumineuse a été réalisée par un Arduino nano qui sert à contrôler l’affichage, le clignotement et d’autres fonctions. Ce dernier est attaché avec un module de bluetooth HC05 qui permet la communication avec le smartphone à fin de commander l’enseigne lumineuse.

Conclusion Générale

Conclusion Générale

Grâce au développement de ce projet, nous avons tous les deux acquis une grande quantité de connaissances sur une variété de technologies qui ont servi au développement de ce projet. Audépart, Les Transistors nous était étranger car c'était quelque chose que nous n'avions utilisé qu'une seule fois auparavant.

Il a été très stimulant mais aussi très enrichissant d'apprendre de nouveaux concepts technologiques tels que Arduino/Bluetooth, Logiciel MikroC.

MIT App Inventor était un autre outil avec lequel nous pensions que nos connaissances étaient suffisantes. et l'apprentissage de ces deux outils, combiné à l'utilisation de nombreuses autres technologies, s'est avéré extrêmement utile, qui ont été utilisées dans le développement du projet en travaillant dans une équipe de développement, nous pensons qu'il sera utile d'apprendre à utiliser ces deux outils.

Références Bibliographiques

Références Bibliographiques

1. Technologies des composants électroniques.” Technologies des composants électroniques.
2. ingénieurs.com, ingénieurs.com.
3. “Arduino Docs | Arduino Documentation.” Arduino Docs | Arduino Documentation, <https://docs.arduino.cc/>. Accessed 15 September 2022.
4. “The Arduino Mega 2560.” Repository UNAIR, <https://repository.unair.ac.id/55279/4/FV.OSI.42-16%20Bah%20%20r-3.pdf>. Accessed 15 September 2022.
5. “Arduino Nano Board: Features, Pinout, Differences and Its Applications.” ElProCus, <https://www.elprocus.com/an-overview-of-arduino-nano-board/>. Accessed 15 September 2022.
6. “ElProCus.” ElProCus - Electronic Projects for Engineering Students - ElProCus is an educational website on electronic projects for ECE and EEE students., <http://www.elprocus.com>. Accessed 15 September 2022.
7. “Enseigne lumineuse : avantages - Ooreka.” Enseigne, <https://enseigne.ooreka.fr/comprendre/enseigne-lumineuse>. Accessed 15 September 2022.
8. “INTRODUCTION AU LANGAGE MIKROC.” Microcontrôleurs.
9. “Introduction to mikroC PRO for PIC.” Introduction to mikroC PRO for PIC, 19 December 2012, https://download.mikroe.com/documents/compilers/mikroc/pic/help/introduction_to_mikroc_pro_for_pic.htm. Accessed 15 September 2022.
10. KHAN, ASIF. “Introduction to PROTEUS.” PROTEUS, pp. 1-5.
11. “Les transistors bipolaires.” Ingénieurs, 20 February 2013, <https://www.ingenieurs.com/documents/cours/les-transistors-bipolaires-316.php>. Accessed 15 September 2022.
12. “MIT App Inventor.” MIT App Inventor, <https://appinventor.mit.edu/>. Accessed 15 September 2022.
13. “Régulateur de tension électrique : principe et types - Ooreka.” Installation électrique, <https://installation-electrique.ooreka.fr/astuce/voir/728075/regulateur-de-tension-electrique>. Accessed 15 September 2022.
14. “Sensors Modules Bluetooth Module Hc 05 | Sensors Modules.” ElectronicWings, <https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05->. Accessed 15 September 2022.

Références Bibliographiques

15. “What does RGB LED mean?” Philips lighting, <https://www.lighting.philips.com/main/support/support/faqs/white-light-and-colour/what-does-rgb-led-mean>. Accessed 15 September 2022.
16. “What is a FET: Field Effect Transistor: Types, Technology.” Electronics Notes, https://www.electronics-notes.com/articles/electronic_components/fet-field-effect-transistor/what-is-a-fet-types-overview.php. Accessed 15 September 2022.

Resumé :

Notre projet consiste à réaliser un circuit électronique basé sur une plate forme arduino pour commander une enseigne lumineuse à distance via le Bluetooth HC-5 depuis une application sur un smartphone. Nous transmettrons des commandes pour réaliser des sequences d'animations couleurs et monochromes.

Abstract :

Our project consists of creating an electronic circuit based on an Arduino platform to control a remote illuminated sign via Bluetooth HC-5 from an application on a smartphone. We will send commands to make sequences of color and monochrome animations.

ملخص:

للتحكم في علامة مضيئة عن بعد عبر تطبيق على اردوينو. يتكون مشروعنا من انشاء دارة الكترونية تعتمد على الهاتف الذكي عن طريق ارسال سلسلة من الرسوم والوان بمساعدة البلوتوت