



École Supérieure Multinationale des Télécommunications



BÉNIN BURKINA FASO GUINÉE MALI MAURITANIE NIGER SÉNÉGAL

MÉMOIRE FIN DE FORMATION POUR L'OBTENTION DU DIPLÔME DE LICENCE PROFESSIONNELLE EN TIC



OPTION : Systèmes Audiovisuels Numériques (SAN)

THÈME

*Etude et mise en œuvre d'un système intelligent de
surveillance de cheptel*

Sous la direction de

Dr. Boudal **NIANG**

Enseignant - Chercheur

Département ESMT TIC

Présenté et soutenu par

M. Cheikh Ahmad TIDIANE
NDIAYE

Mlle Ndeye Fatou **MBOUP**

Promotion 2015 - 2018

Décembre 2018

EPIGRAPHE

Celui qui applique ce qu'il aura appris.

Allah l'augmente en savoir jusqu'à ce qu'il sache ce qu'il n'a pas appris.

Prophète Muhammad (PSL)

DEDICACE

Nous dédions ce travail à:

- *Nos chers parents, que ce travail soit pour eux le témoignage de notre reconnaissance et de notre affection*
- *Nos familles respectives*
- *Nos amis*
- *Et à tous ceux qui ont pu nous apporter leur soutien*
- *Nos professeurs*
-

REMERCIEMENTS

En préambule à ce mémoire, nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

Nous tenons à remercier **Dr Boudal Niang** enseignant/Chercheur à l'Ecole Supérieure Multinationale des Télécommunications, Monsieur **Omar Ndiaye** notre cher responsable pédagogique & Monsieur **Omar Dabo**. C'est grâce à leur disponibilité, leur écoute, l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer que ce mémoire a vu le jour.

Nous exprimons notre gratitude à tous nos camarades, proches, amis et à tous ceux, qui avec gentillesse, nous ont donné des conseils, soutenu et encouragé au cours de la réalisation de ce mémoire.

Nous exprimons notre reconnaissance envers nos parents et familles pour avoir fait de nous de dignes personnes et pour leur contribution, leur soutien, et leur patience qu'ils nous ont toujours apportée.

Merci à toutes et à tous.

AVANT-PROPOS

Ce mémoire s'inscrit dans le cadre de l'obtention d'un diplôme de Licence Professionnelle en Télécommunications et Informatique (LPTI) après trois (03) années d'études dans les dits domaines.

Notre travail consiste en la conception d'un système intelligent pour la surveillance du cheptel national.

Pour la bonne compréhension de ce document, certaines notions qu'on pourrait affecter au domaine de l'internet des objets ont été brièvement expliquées.

SOMMAIRE

INTRODUCTION.....	
CHAPITRE 1. PRÉSENTATION DU SUJET ET PROBLÉMATIQUE.....	
1.1 Problématique.....	12
1.1.1 Explication du sujet.....	12
1.2 Présentation du sujet.....	12
1.2.1 Explication du sujet.....	13
1.2.2 Objectifs et Délimitation du sujet.....	13
1.2.3 Démarche et Méthodologie.....	14
CHAPITRE 2. FONCTIONNEMENT DU COLLIER CONNECTÉ.....	
2.1 Architecture du système de collier connecté.....	15
2.2 Principe de fonctionnement.....	15
CHAPITRE 3. REALISATION DU COLIER CONNECTÉ.....	
3.1 ETUDE DES CARTES ET MODULES UTILISÉES.....	19
3.1.1 Microcontrôleur ARDUINO.....	19
3.1.2 MODULE GSM/GPRS.....	21
3.1.3 MODULE GPS.....	22
3.1.4 Présentation du module GPS NEO-6M.....	24
3.1.5 LORA GATEWAY ET SON UTILISATION.....	25
3.2 Logiciels utilisé dans la réalisation du système.....	26
3.2.1 LOGICIEL ARDUINO.....	26
3.2.2 De la configuration jusqu'au téléversement en passant par le codage au microcontrôleur.....	40
ASSEMBLAGE ET ELABORATION DU DISPOSITIF.....	58
CHAPITRE 4 : GESTION DU BETAIL (APPLICATION MOBILE).....	
3.3 ETUDE ET CONCEPTION DE L'APPLICATION MOBILE.....	60
ETUDE.....	60
CONCEPTION DE L'APPLICATION MOBILE.....	60
PERSPECTIVES.....	
Modifier le collier de sorte qu'il soit adéquat aux endroits sans connexion.....	67
Minimiser la taille du matériel ayant servi à faire le collier.....	67
Ajouter des fonctionnalités à l'application mobile.....	67
Caméra IP.....	68
CONCLUSION.....	

LISTE DES FIGURES & TABLEAUX

Figure 2.1-1:Architecture du système.....	15
Figure 2.2-1:Envoi de position.....	16
Figure 2.2-2:Déroulement du service.....	16

LISTE DES TABLEAUX

Tableau 1 : Explication du diagramme UML

Tableau 2 : fonctionnalités du GPS

Tableau 3 : fonctionnalités du collier

GLOSSAIRE

GSM: Global System for Mobile
GPS: Global Positioning System
GPRS: General Packet Radio Service
IoT: Internet of Things
LED: Light-Emitting Diode
OS: Operating System
UML: Unified Modeling Language
XML: Extensible Markup Language

1) INTRODUCTION

Estimé à 2 milliards de Francs CFA en 2017, le vol du bétail des éleveurs est un sujet qui tient à cœur le président de la République du Sénégal.

Notre approche est ainsi de sécuriser un élément important tant sur le plan culturel qu'économique : le cheptel national. Une grande partie de la population est concernée depuis l'éleveur jusqu'au consommateur du produit d'origine animale (l'industrie laitière, chaîne de production de viande, ...) en outre les populations rurales sont souvent concernées par des conflits sociaux notamment avec les agriculteurs dans la gestion du trajet des troupeaux lors de leurs transhumance.

C'est pour cette raison, le sujet portant sur la mise en place d'un système intelligent pour la surveillance du cheptel national nous a été proposé.

La réalisation de ce travail passe par une certaine maîtrise des équipements les plus utilisés dans ce domaine. Ceci permet de développer une péremptoire compétence dans ce domaine qui est l'IoT.

Le présent travail, qui permet de faire ressortir la réalisation de ce système jusqu'à la pratique est structuré en trois grands points. Dans un premier temps, nous allons rappeler quelques concepts théoriques de façon générale sur notre système intelligent; ensuite présenter les équipements, outils et logiciels nécessaires à la conception du dispositif avant de terminer par le manuel des procédures des différents travaux pratiques et la mise en place des services.

CHAPITRE 1. PRÉSENTATION DU SUJET ET PROBLÉMATIQUE

1.1 Problématique

1.1.1 Explication du sujet

Les objets connectés, ou internet des objets (IoT) est un domaine en plein essor depuis une vingtaine d'année avec l'arrivée de la technologie Smartphone. Tous ces objets collectent un nombre important de données (environnementales, logistique, santé, etc.). Les domaines d'activités des IoT sont multiples, par exemple l'élevage qui de plus en plus prend de l'ampleur dans l'économie : comme l'indique la racine étymologique du mot « cheptel », le troupeau constitue avant tout une richesse. De ce fait, il convient de considérer les éleveurs, sinon comme des nantis, du moins comme une catégorie d'individus n'appartenant pas aux plus défavorisés. Aussi, l'élevage constitue-t-il, au moins en théorie, un rempart contre la pauvreté, une véritable « richesse » des pauvres, une richesse qu'on se doit de protéger contre la menace qu'elle court le plus qui est le vol de bétail. Le vol de bétail n'est plus un événement passager s'inscrivant dans la rubrique des faits divers, mais un phénomène de plus en plus dangereux qui menace le secteur de l'élevage comme l'indiquent les données du ministère de l'intérieur qui a recensé 778 plaintes relatives à des vols de bétail, au cours du 1er semestre 2016, contre 1876 plaintes pour le même motif en 2017.

Du fait de son caractère extensif, l'élevage reste confronté à des problèmes récurrents de sécurisation du foncier pastoral et de vol de bétail.

La sécurité du cheptel national est devenue un problème d'ordre majeur du fait de la perte économique estimé à 2 milliards de Francs CFA par an. En effet l'élevage, par les revenus qu'il génère et sa contribution à la croissance des activités du secteur primaire, constitue « un secteur clé » de l'économie sénégalaise. C'est pourquoi ce projet est d'une importance capitale pour la facilitation et l'intégration des éleveurs dans le numérique.

1.2 Présentation du sujet

Après avoir fait état de la problématique que soulève notre sujet, il convient de faire sa présentation, de ses objectifs, du périmètre d'étude et de la démarche choisie pour une meilleure compréhension.

1.2.1 Explication du sujet

Ce projet de mémoire consiste en la conception d'un système intelligent pour la surveillance du cheptel national permettant d'assurer une ou plusieurs missions tel que celles susmentionnées dans la problématique. Chaque fonctionnalité nécessite d'embarquer un certain nombre de composants : microcontrôleur ARDUINO, module GSM, module GPS. Dans l'optique de missions en milieu urbain, le dispositif (collier) devra être de petite taille, transportable, discret, capable de répondre aux attentes.

La vocation principale de ce système est la suivie et la surveillance du bétail. Ainsi, notre collier sera accompagné d'une application mobile qui permettra de recueillir en temps réel les bénéfices de la mission. Il sera chargé de l'analyse et l'interprétation des données, leur éventuelle retransmission ainsi que leur enregistrement.

Le système est destiné aux éleveurs de façon générale, aux éleveurs localisés dans les milieux urbains en particulier. Par conséquent, les informations traitées par le système doivent être de bonne qualité c'est-à-dire sûres, fiables, utiles et à jour pour une efficacité de la prise de décision des utilisateurs.

1.2.2 Objectifs et Délimitation du sujet

Conformément à la méthodologie de recherche, nos objectifs seront de deux natures à savoir l'objectif général et quelques objectifs spécifiques.

L'objectif final de ce mémoire est de créer un collier artisanal relié à une application mobile permettant d'obtenir, en temps réel, des informations sur les sujets de l'éleveur, les informations relatives à leur traitement et vaccination (s'ils existent) pour la gestion du cheptel, d'avoir la géolocalisation des sujets disposant du collier, éventuellement toutes les informations disponibles sur la bergerie et ses sujets, etc. L'interface de l'application se fera simple et intuitive pour accéder rapidement aux informations.

Les objectifs spécifiques sont au nombre de cinq (5) étant entendu que leurs réalisations permettront d'atteindre l'objectif général.

- Etudier les notions associées au sujet ;
- Définir les fonctionnalités du système et les données à traiter ;
- Choisir le matériel, les outils et langages de programmation à utiliser ;
- Concevoir des architectures matérielle et logicielle du système, ainsi que son fonctionnement ;
- Réaliser le collier artisanal afin de démontrer les fonctionnalités du système.

Par contre, l'étude conceptuelle et technique du système ne prendra pas en compte certains aspects tels que les questions d'ordre financier et commercial ou encore le business plan du projet. Nous donnerons la priorité aux fonctionnalités essentielles du système et à la réalisation du collier en tant que prototype.

1.2.3 Démarche et Méthodologie

Le travail qui suivra, se décline en quatre grandes parties :

- Dans un premier temps, à travers une étude de l'art, nous allons définir l'ensemble des termes et concepts relatifs à notre thématique, présenter l'historique des colliers, énumérer les différents types de colliers et leurs caractéristiques.
- Dans un deuxième temps, nous aurons à définir les architectures (matérielle et logicielle) du système et son fonctionnement. Il faudra donc faire une analyse fonctionnelle et une étude conceptuelle pour représenter les différents composants, leurs fonctions et leurs interactions.
- En d'autres termes, définir les acteurs sollicitant le système et les fonctionnalités essentielles. Pour ce faire, nous aurons recours aux applications de méthodes conceptuelles à savoir Unified Modeling Language (UML).
- Dans la troisième partie, il s'agira de la réalisation du prototype. Nous allons donc parler de la partie matérielle, ensuite celle logicielle. En effet, au-delà de la représentation physique, nous allons définir les logiciels et programmes utilisés, leur orchestration afin d'obtenir un système sûr, fiable et répondant aux besoins des usagers. Et pour finir nous allons faire des tests pour visualiser le résultat. Le but ultime n'étant pas de présenter une solution totalement prête à l'emploi, mais de prouver la faisabilité du projet d'étude.
- Enfin dans la dernière partie, nous allons concevoir des services qui pourront être implémentés sur le système embarqué avant d'ouvrir des perspectives qui aboutiront à une conclusion générale.

CHAPITRE 2. FONCTIONNEMENT DU COLLIER CONNECTÉ

1.3 Architecture du système de collier connecté

Les systèmes de contrôle et de surveillances de cheptel sont composés des entités suivantes :

- Capteurs (colliers)
- Passerelle
- Réseau (SMS, connexion 3G)
- Serveur applicatif pour le traitement de données

Une vache équipée d'un collier connecté émet sa position à une gateway via un réseau adapté aux objets connectés, c'est-à-dire consommant peu d'énergie. Si l'animal sort de son périmètre de sécurité, une alerte est envoyée à une communauté, sur le smartphone des utilisateurs.

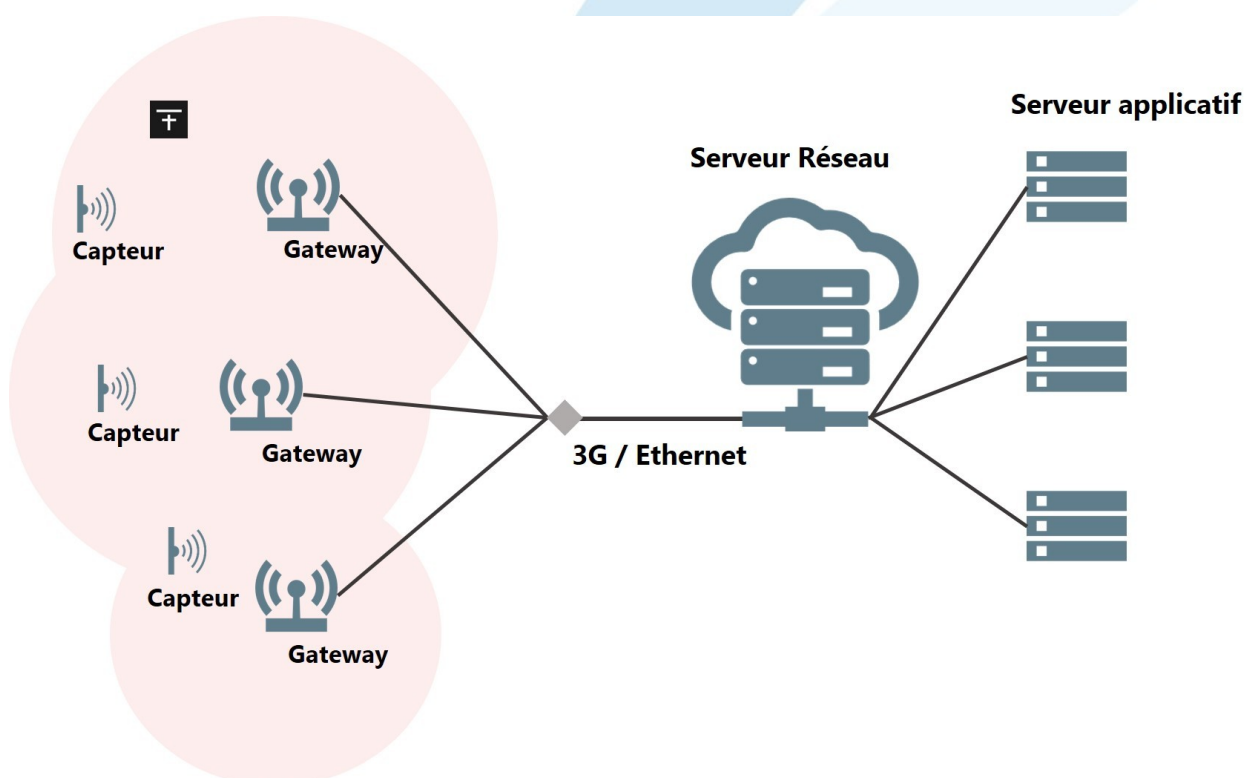


Figure 2-1: Architecture du système

1.4 Principe de fonctionnement

La figure 2.2 illustre un digramme de cas d'utilisation d'envoi d'une position géographique.

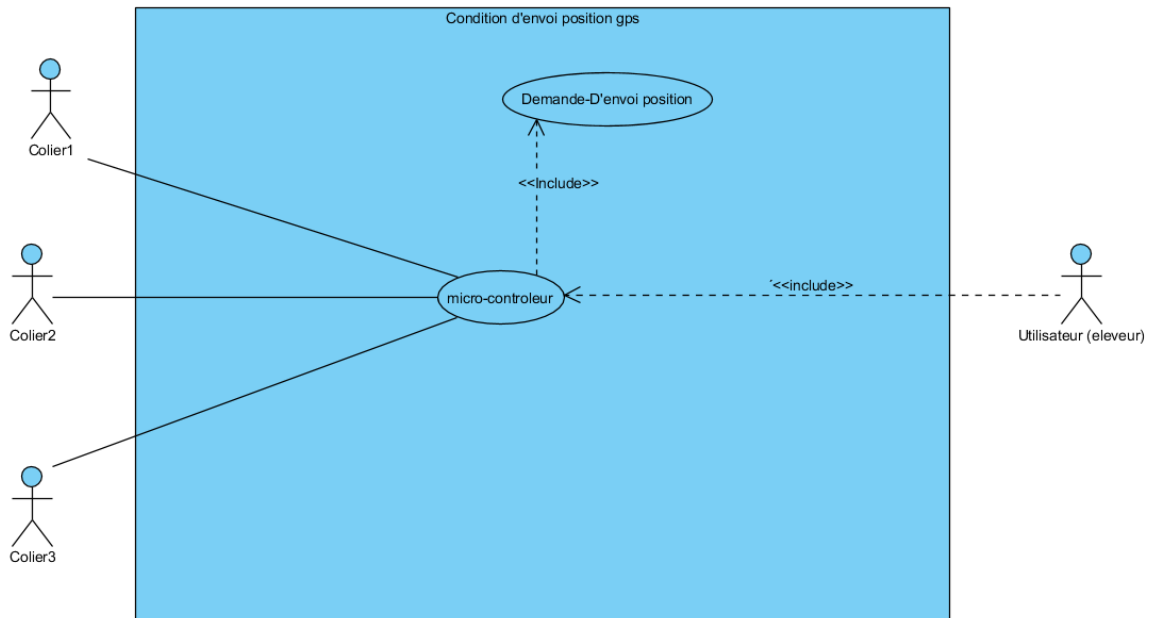


Figure 2-2:Envoi de position

Le déroulement du service se fait comment suit :

- La communauté envoie une demande de positionnement d'un animal
- Le système recherche, effectue des calculs et envoie une réponse contenant la carte ainsi que la position de l'animal recherché
- Une notification est envoyée à la communauté si l'animal s'éloigne de sa position

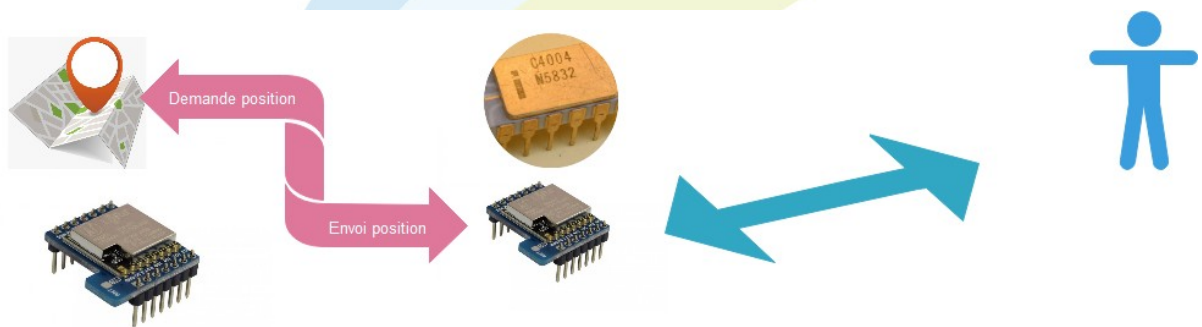


Figure 2-3:Déroulement du service

Le tableau 2.1 contient les données d'explication du diagramme UML (voir la figure 2.1).

Titre	Cas D'envoi de position GPS
Description	Ce cas d'utilisation permet à l'utilisateur de recevoir la position en temps réel du bétail
Acteurs	Eleveur
Préconditions	Il faut au préalable que l'éleveur se connecte au microcontrôleur
Scénario nominal	<p>1- L'utilisateur se connecter à l'application mobile grâce à son smartphone</p> <p>2- Les colliers envoient leur position continuellement au niveau de la base de donnée crée en amont pour sauvegarder ces données.</p> <p>3- L'utilisateur a partir du volet localisation envoi une demande de localisation de ses sujets qu'il recevra en temps réel sur la carte Google Maps intégrée à l'application correspondant aux coordonnées des colliers</p>
Scenario alternatif	Le bétail s'éloigne de la zone de couverture, une alerte est envoyée indiquant le manque d'un ou plusieurs sujets
Post-conditions	Localisation du bétail

Tableau 2.1. Explication du diagramme UML

La technologie GPS est utilisée pour la géolocalisation des animaux en temps réel.

Fonction	Localisation
Objectif	Donner la position de l'animal, suivre ses déplacements
Description	C'est un collier composé de Module GPS. Il permet de suivre les déplacements de l'animal pour éviter qu'il soit volé.
Contraintes/gestion	Avoir une connexion WIFI à tout instant

Tableau 2.2. Technologie GPS

Le tableau 2.3 contient les données d'explication du collier.

Fonction	Protection et support
Objectif	Permettre de mettre le dispositif sur l'animal
Description	Il est composé soit d'un tissu ou d'un fil que l'on utilise pour

	pouvoir mettre le dispositif au cou de l'animal.
Contraintes/gestion	Dès qu'il est enlevé de l'animal on ne peut ne plus le retrouver

Tableau 2.3. Description du collier



CHAPITRE 3. REALISATION DU COLIER CONNECTÉ

1.5 ETUDE DES CARTES ET MODULES UTILISÉES

1.5.1 Microcontrôleur ARDUINO

ARDUINO est une petite carte électronique programmable et un logiciel multiplateforme gratuit, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques.



Figure 3.1: Logo officiel de ARDUINO

Une carte électronique est un support plan, flexible ou rigide, généralement composé d'Époxy ou de fibre de verre. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques. Chaque piste relie tel composant à tel autre, de façon à créer un système électronique qui fonctionne et qui réalise les opérations demandées.

Évidemment, tous les composants d'une carte électronique ne sont pas forcément reliés entre eux. Le câblage des composants suit un plan spécifique à chaque carte électronique, qui se nomme le schéma électronique.

Enfin, avant de passer à la réalisation d'une carte électronique, il est nécessaire de transformer le schéma électronique en un schéma de câblage, appelé typon.

Avec ARDUINO, nous n'aurons pas à fabriquer la carte et encore moins à la concevoir. Elle existe, elle est déjà prête à l'emploi et nous n'avons plus qu'à l'utiliser.

Pourquoi avons-nous choisi ARDUINO?

C'est une carte électronique programmable et un logiciel gratuit. De plus ARDUINO c'est :

- Un prix dérisoire étant donné l'étendue des applications possibles. On comptera 20 euros pour avoir de ses cartes. Le logiciel est fourni gratuitement !

- Une compatibilité sous toutes les plateformes, à savoir : Windows, Linux et Mac OS.
- Une communauté ultra développée ! Des milliers de forums d'entre-aide, de présentations de projets, de propositions de programmes et de bibliothèques, ...
- Une liberté quasi absolue. Elle constitue en elle-même deux choses :
- Le logiciel : gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet.

Cette liberté a une condition : le nom « ARDUINO » ne doit être employé que pour les cartes « officielles ». En somme, vous ne pouvez pas fabriquer votre propre carte sur le modèle ARDUINO et lui assigner le nom « ARDUINO ».

Et enfin, les applications possibles

Voici une liste non exhaustive des applications possibles réalisées grâce à ARDUINO :

- Contrôler des appareils domestiques
- Donner une "intelligence" à un robot (intelligence artificielle)
- Réaliser des jeux de lumières
- Permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs
- Télécommander un appareil mobile (modélisme)
- etc.

Elle possède deux rangées de connecteurs le long de ses extrémités qui sont utilisés pour brancher une vaste gamme de composants électroniques ou des cartes d'extensions (appelées SHIELDS) qui augmentent ses capacités.

Elle est aussi équipée d'une LED intégrée qu'il est possible de commander au travers de vos programmes. Vous pouvez apercevoir cette LED sur l'image ci-dessous, elle est repérable grâce au « L » visible sur la carte.

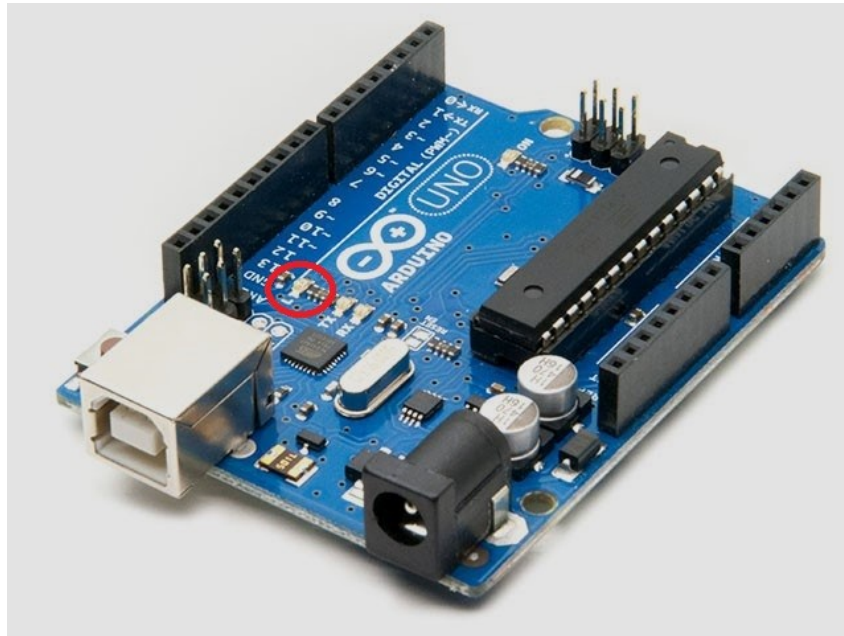


Figure 3.2 Carte ARDUINO UNO

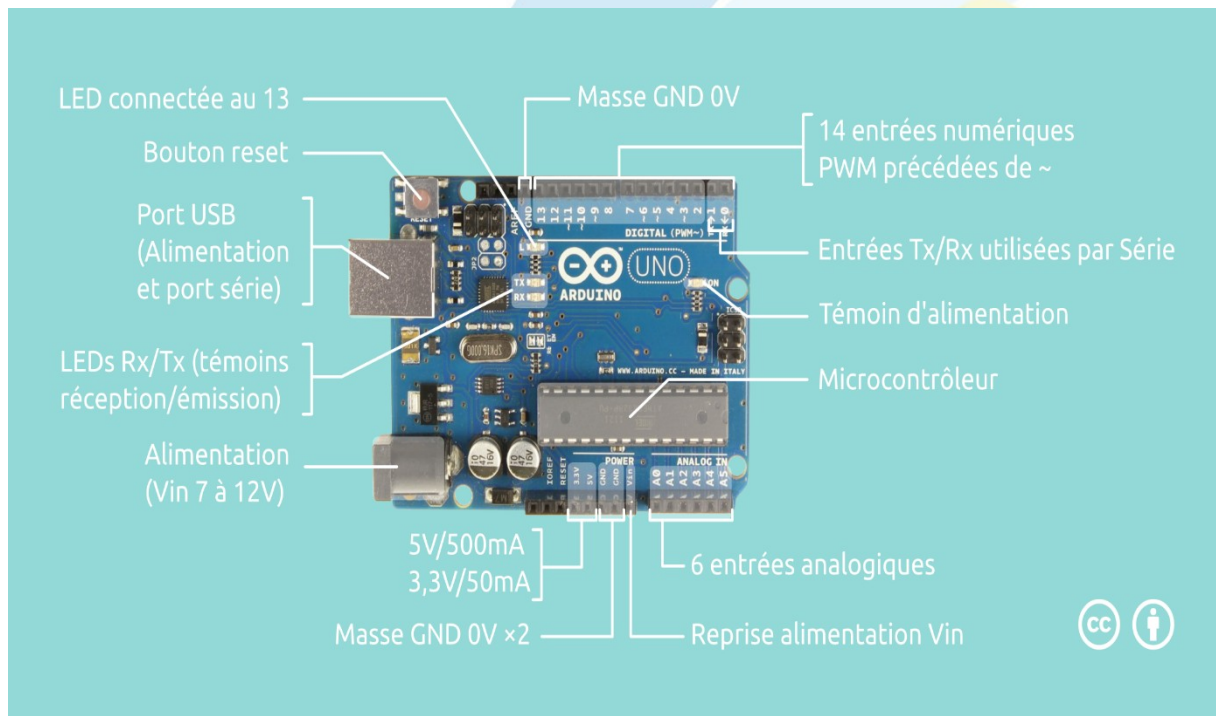


Figure 3.3. Carte ARDUINO UNO annoté

1.5.2 MODULE GSM/GPRS

Ce module GSM-GPRS 2 ARDUINO avec antenne intégrée basé sur le module M10 de QUACTEL est prévu pour ajouter les fonctionnalités de SMS, GSM/GPRS et appel (nécessite l'ajout d'un micro et d'un HP) à vos applications ARDUINO.

Il vous suffit d'ajouter une carte SIM (non incluse) pour appeler, envoyer des SMS et héberger une page web. Ce module est compatible avec les cartes ARDUINO UNO, LEONARDO, MEGA ou compatibles.

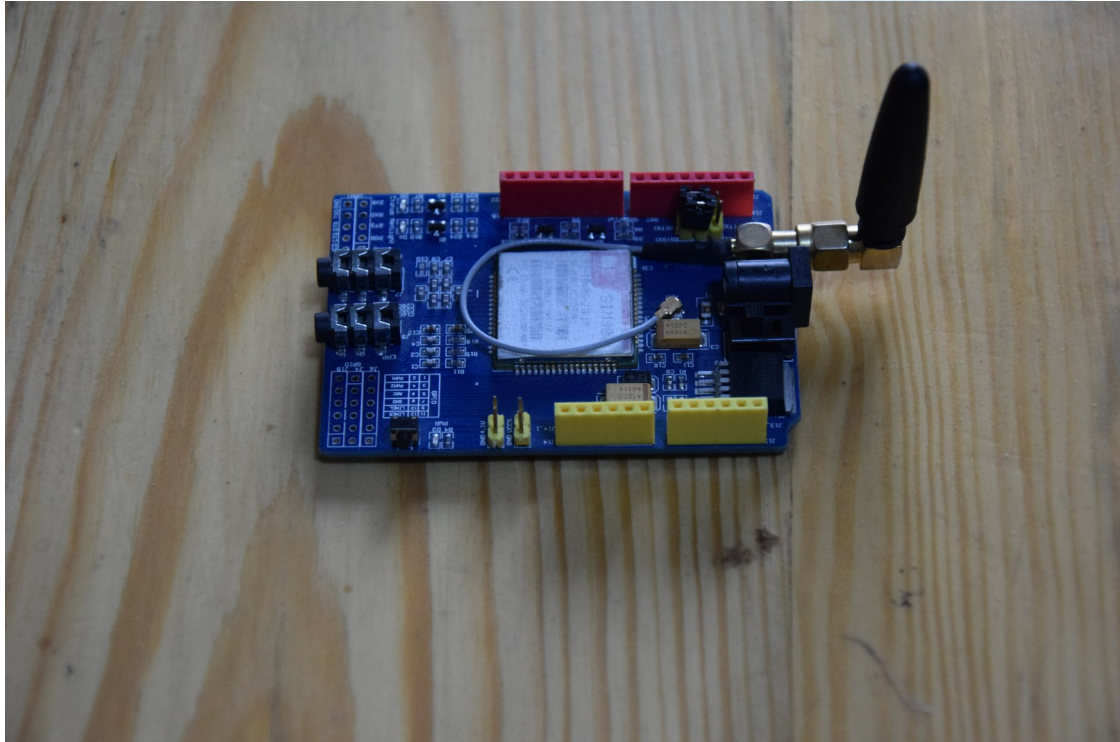


Figure 3.4. Module GSM/GPRS SIM900

1.5.3 MODULE GPS

Ce module GPS est fourni totalement assemblé et ne nécessite pas d'antenne pour fonctionner. Il est compatible avec les cartes ARDUINO UNO, Duemilanove, Leonardo etc. Ce module GPS utilise le récepteur SUP500F de Skytraq. Ce récepteur GPS possède 65 canaux à 10hz ainsi qu'une antenne intelligente intégrée. Une interface série 3V est disponible.

Ce module ARDUINO est réalisé par la société Dexter Industries.

Les caractéristiques techniques de ce module GPS sont :

- Type de récepteur : L1 C/A, Moteur Venus 6 à 65 canaux
- Précision
 - Position : 2,5m CEP
 - Vitesse : 0,1m/sec
 - Temps : 60ns
- Temps de démarrage
 - 1 seconde de démarrage à chaud
 - 29 secondes de démarrage à froid

- Ré acquisition : 1s
- Sensibilité : -161dBm
- Taux de rafraichissement : le récepteur SUP500F support différents taux entre 1 et 10Hz (1Hz par défaut)
- Dynamics : 4G (39.2m/sec²)
- Limites opérationnelles : Altitude < 18000m ou vitesse < 515m/s
- Interface série : 3V LVTTTL
- Protocole : NMEA-0183 V3.01 GPGGA, GPGLL, GPGSA, GPGSV, GPRMC, GPVTG*1
9600 bauds, 8, N, 1
- Tension d'entrée : 3,0V ~ 5,5V DC
- Courant d'entrée : ~33mA
- Plage de température opérationnelle : -40°C ~ +85°C
- Humidité : 5% ~ 95%

Le module GPS modèle NEO-6M-7M est utilisé avec l'ARDUINO pour obtenir des données GPS. GPS est synonyme de système de positionnement global et peut être utilisé pour déterminer la position, le temps et la vitesse.



Figure 3.5. Module GPS

A partir de ce module couplé à une carte ARDUINO on peut obtenir :

- Obtenir des données GPS RAW
- Analyse des données brutes pour obtenir des informations GPS sélectionnées et lisibles
- Obtenir l'emplacement

1.5.4 Présentation du module GPS NEO-6M

Le module GPS NEO-6M-7M est montré dans la figure ci-dessous. Il est livré avec une antenne externe, et viennent avec des broches d'en-tête. On aura juste à souder les broches.



Figure 3.6: Module GPS

Ce module est doté d'une antenne externe et d'une EEPROM intégrée.

- Interface : RS232 TTL
- Alimentation d'énergie : 3V à 5V
- Baud rate par défaut : 9600 bps
- Fonctionne avec des phrases NMEA standard

Le module GPS NEO-6M-7M possède quatre broches : VCC, RX, TX et GND. Le module communique avec l'ARDUINO via la communication série en utilisant les broches TX et RX, de sorte que le câblage ne pouvait pas être plus simple :

- NEO-6M câblage de module GPS à ARDUINO UNO
- VCC 5V
- Broche RX TX définie dans la série logicielle
- TX RX défini dans la série logicielle
- GND
- Obtenir des données RAW GPS

Pour obtenir des données GPS RAW, on aura juste besoin de commencer une communication série avec le module GPS en utilisant Serial Software.

1.5.5 LORA GATEWAY ET SON UTILISATION

LORAWAN est un protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LORA et connectés à l'internet via des passerelles, participant ainsi à l'internet des objets. Ce protocole est utilisé dans le cadre des villes, le monitoring industriel ou encore l'agriculture. La technologie de modulation liée à LORAWAN est LORA, née à la suite de l'acquisition de la startup grenobloise Cycléo par Semtech en 2012. Semtech promeut sa plateforme LORA grâce à la LORA ALLIANCE, dont elle fait partie. Le protocole LORAWAN sur la couche physique LORA permet de connecter des capteurs ou des objets nécessitant une longue autonomie de batterie (comptée en années), dans un volume (taille d'une boîte d'allumettes ou d'un paquet de cigarettes) et un coût réduit.

LORAWAN est l'acronyme de *Long Range Wide-area network* que l'on peut traduire par « réseau étendu à longue portée ».

Relève de compteur électrique en temps réel, monitoring distant de panneaux solaires ou pilotage fin de l'éclairage public constituent quelques exemples de gestion personnalisée de l'énergie associés à l'avènement des objets connectés (IoT).

Économique, puissante et ouverte, la technologie radio LORA (Long Range, « longue portée ») se trouve au cœur de ces nouveaux usages et applications emblématiques des smart Grids et des Smarts Cities.

- **Une technologie bidirectionnelle, à ultra-longue portée et ultra précise**

La technologie LORA appartient aux solutions sans fil bas débit tournées uniquement vers l'internet des objets industriels. Elle se distingue par sa réception et son émission à ultra-longue portée (environ 15 km), dont le spectre autorise la couverture de plusieurs millions d'appareils sans la moindre difficulté. Cet avantage se double d'une excellente pénétration dans les bâtiments (même dans les caves et les sous-sols), avec une liaison 5 à 10 dB plus favorable que les réseaux cellulaires, et d'une géolocalisation précise sans recours au GPS.

- **Avec le protocole LORA, fiabilité et économies maximales**

Les échanges réguliers mais de durée très réduite de messages courts (quelques dizaines d'octets) qui caractérisent le protocole LORA permettent de concilier communication distante de qualité et longue durée de vie des équipements. Ils évitent les phénomènes de collision radio et d'interférences tout en ne sollicitant que faiblement les sources d'électricité des capteurs. Ce renforcement de la fiabilité s'accompagne d'économies opérationnelles très importantes. Le coût

annuel de l'accès, qui n'intègre plus d'abonnement ou de consommation data, se voit divisé par 10 à 12 en comparaison d'un système M2M associant carte SIM et réseau GSM.

- **Une architecture réseau totalement ouverte**

Relayée par plus d'une centaine de partenaires dans le monde, l'architecture LORAWAN met l'accent sur son ouverture. Face à des concurrents comme SIGFOX privilégiant une gestion intégrée de leurs réseaux, elle permet à toute entreprise de déployer rapidement ses propres installations et de les exploiter. La compatibilité étendue de ses passerelles offre en complément une intégration facile et immédiate des données collectées dans les systèmes d'information existants.

1.6 Logiciels utilisé dans la réalisation du système

1.6.1 LOGICIEL ARDUINO

On parle de *carte électronique programmable* ; c'est dû au fait que ARDUINO est programmable. Cela signifie qu'elle a besoin d'un programme pour fonctionner.

Un programme est une liste d'instructions qui est exécutée par un système. Par exemple votre navigateur internet est un programme.

```
1 // définition de la broche 2 de la carte en tant que variable
2 const int led_rouge = 2;
3
4 // fonction d'initialisation de la carte
5 void setup()
6 {
7     // initialisation de la broche 2 comme étant une sortie
8     pinMode(led_rouge, OUTPUT);
9 }
10
11 void loop()
12 {
13     // allume la LED
14     digitalWrite(led_rouge, LOW);
15     // fait une pause de 1 seconde
16     delay(1000);
17     // éteint la LED
18     digitalWrite(led_rouge, HIGH);
19     // fait une pause de 1 seconde
20     delay(1000);
21 }
```

Figure15: Un exemple de programme

Sur le plan des logiciels on utilisera le logiciel ARDUINO. Ce n'est autre qu'un programme informatique exécuté sur un ordinateur. Car pour programmer la carte ARDUINO, nous allons utiliser un programme et en fait, il va s'agir d'un **compilateur** :

En informatique, ce terme désigne un logiciel qui est capable de traduire un langage informatique, ou plutôt un programme utilisant un langage informatique, vers un langage plus approprié afin que la machine qui va le lire puisse le comprendre. Ce langage utilise uniquement des 0 et des 1. Cela pourrait être imagé de la façon suivante :



Figure 16: Procédure de conversion binaire

Donc, pour traduire le langage texte vers le langage machine (avec des 0 et des 1), nous aurons besoin de ce compilateur. Et pas n'importe lequel, il faut celui qui soit capable de traduire le *langage texte ARDUINO* vers le *langage machine ARDUINO*, sinon rien ne va fonctionner.

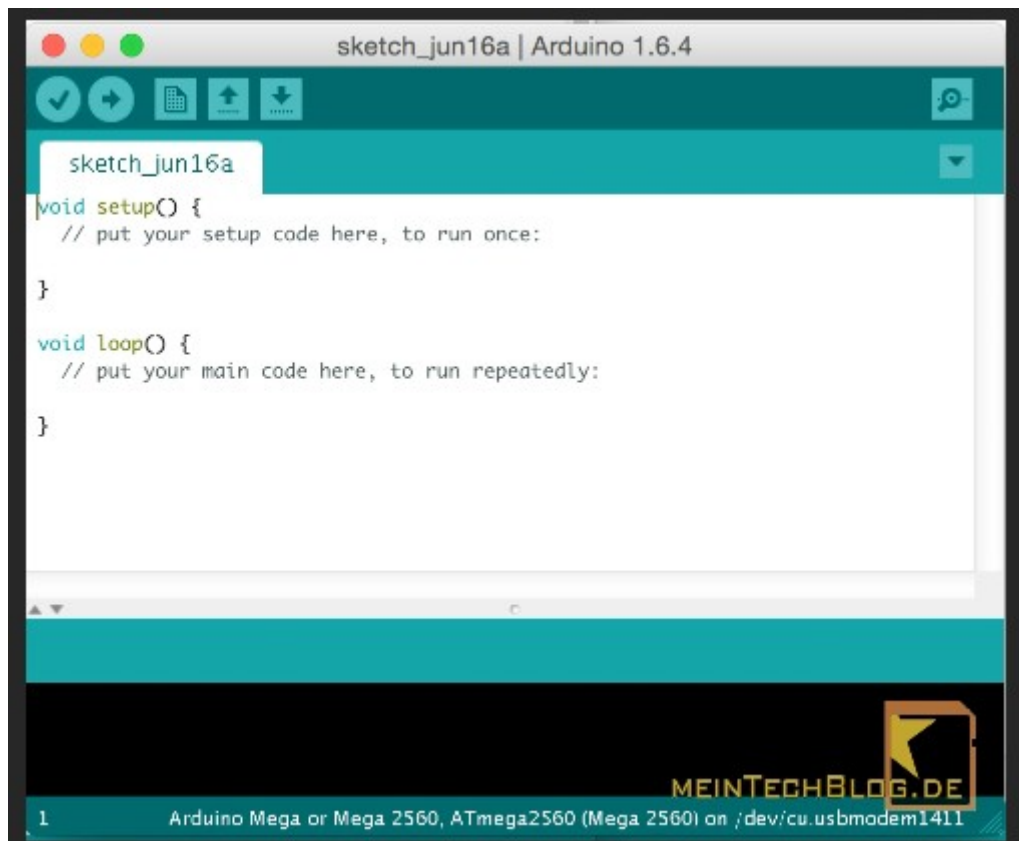


Figure 17: Fenêtre de départ

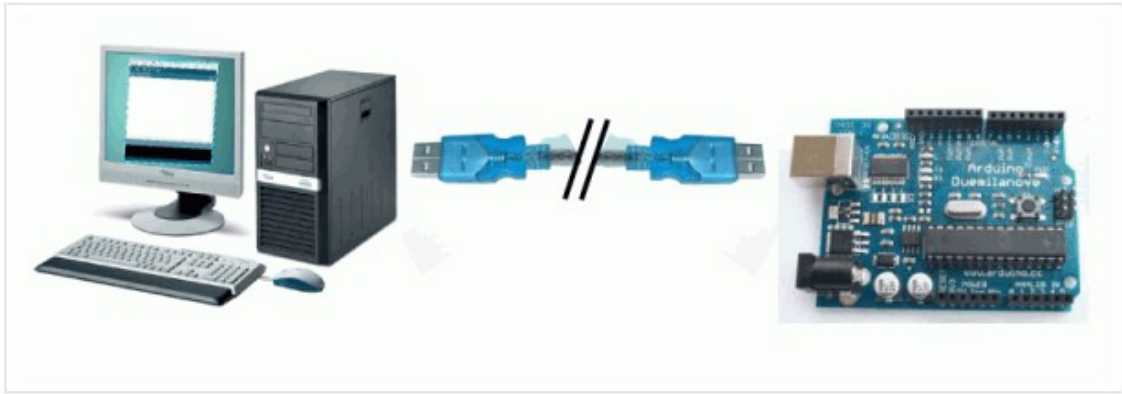
Ceci est la fenêtre de départ. Elle contient l'ossature du code commune et obligatoire. Pour installer le logiciel depuis un fichier zip veuillez suivre les instructions suivantes

1 Débuter avec ARDUINO sous Windows : Installation du logiciel ARDUINO et du driver USB

1. Télécharger le logiciel ARDUINO Téléchargez la dernière version du logiciel ARDUINO sur cette page

2. Connecter la carte ARDUINO à l'ordinateur. A présent, connectez votre carte ARDUINO à votre ordinateur en utilisant votre câble USB. La LED verte d'alimentation (notée PWR) devrait s'allumer. Vous pouvez à présent vérifier que les drivers ont bien été installés en ouvrant le **Panneau de Configuration > Système > Gestionnaire de Périphériques**.

Vous devriez retrouver dans la section Ports LPT et COM un "USB Serial Port" : c'est le port USB de la carte ARDUINO. Noter au passage le numéro du port.



k

Figure 18: Connection avec un Hardware

3. Lancer le logiciel ARDUINO : A présent, lancez le logiciel ARDUINO en double-cliquant 2 fois sur l'icône ARDUINO disponible sur votre bureau ou dans la liste des programmes.

4. Installation du plugin ARDUBLOCK : Il faut déjà avoir installé le logiciel ARDUINO sur son ordinateur. Installer ArduBlock : Le télécharger sur le site : pour la version stable, et jar/download pour la version bêta du 28/8/2014. Créer les dossiers nécessaires où il faut et y placer le fichier "ardublock-all.jar". Pour Windows, **C:\Users\abu\Documents\ARDUINO\tools\ArduBlockTool\tool\ardublock-all.jar** Attention au nom du répertoire \ArduBlockTool, la casse est importante. La vérification de la mise en place du programme Ardublock est visible en lançant le programme ARDUINO, puis aller dans le menu **Outils**, vous devez voir apparaître un menu **Ardublock**.

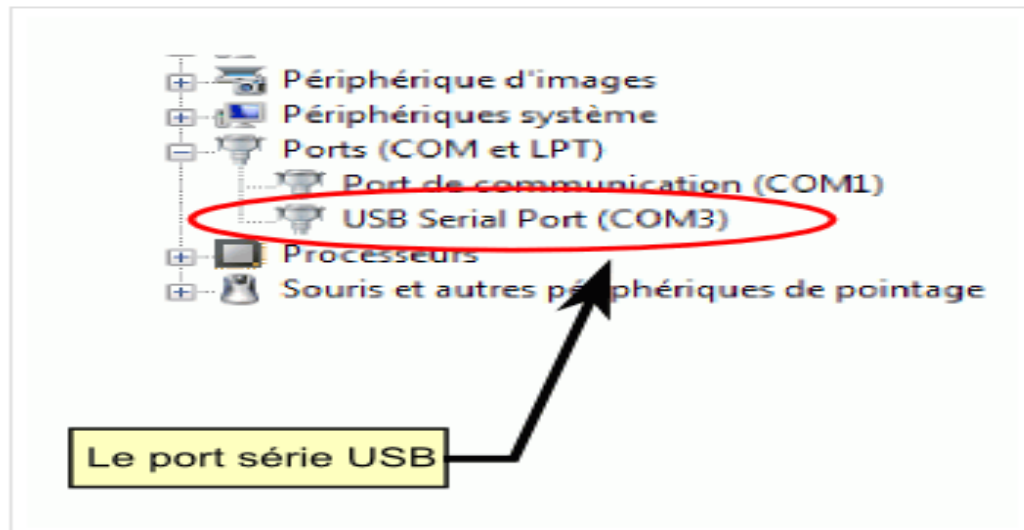


Figure 19:

Le Logiciel ARDUINO : Espace de développement Intégré (EDI) ARDUINO Sur cette page... (Masquer) Description Principe général d'utilisation Description de la barre des boutons Description des menus Sketchbook ("Livre de programmes") Onglets, fichiers multiples et compilation Transfert des programmes vers la carte ARDUINO Saisir votre programme et vérifier le code Sélectionner le bon port série (et la bonne carte ARDUINO...).

Clic sur le bouton UPLOAD (Transfert du programme) Le programme transféré avec succès se lance Bibliothèques Le Moniteur Série Preferences Third-Party Hardware Boards Commentaires utilisateurs 1. Description Le logiciel ARDUINO a pour fonctions principales : de pouvoir écrire et compiler des programmes pour la carte ARDUINO de se connecter avec la carte ARDUINO pour y transférer les programmes de communiquer avec la carte ARDUINO

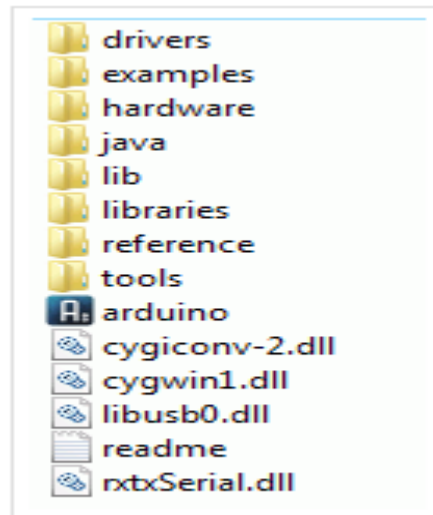


Figure 20: livre de programmes

5 Cet espace de développement intégré (EDI) dédié au langage ARDUINO et à la programmation des cartes ARDUINO comporte : une BARRE DE MENUS comme pour tout logiciel une interface graphique (GUI), une BARRE DE BOUTONS qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation, un EDITEUR (à coloration syntaxique) pour écrire le code de vos programme, avec onglets de navigation, une ZONE DE MESSAGES qui affiche indique l'état des actions en cours, une CONSOLE TEXTE qui affiche les messages concernant le résultat de la compilation du programme Le logiciel ARDUINO intègre également : un TERMINAL SERIE (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte ARDUINO et d'envoyer des caractères vers la carte ARDUINO. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.

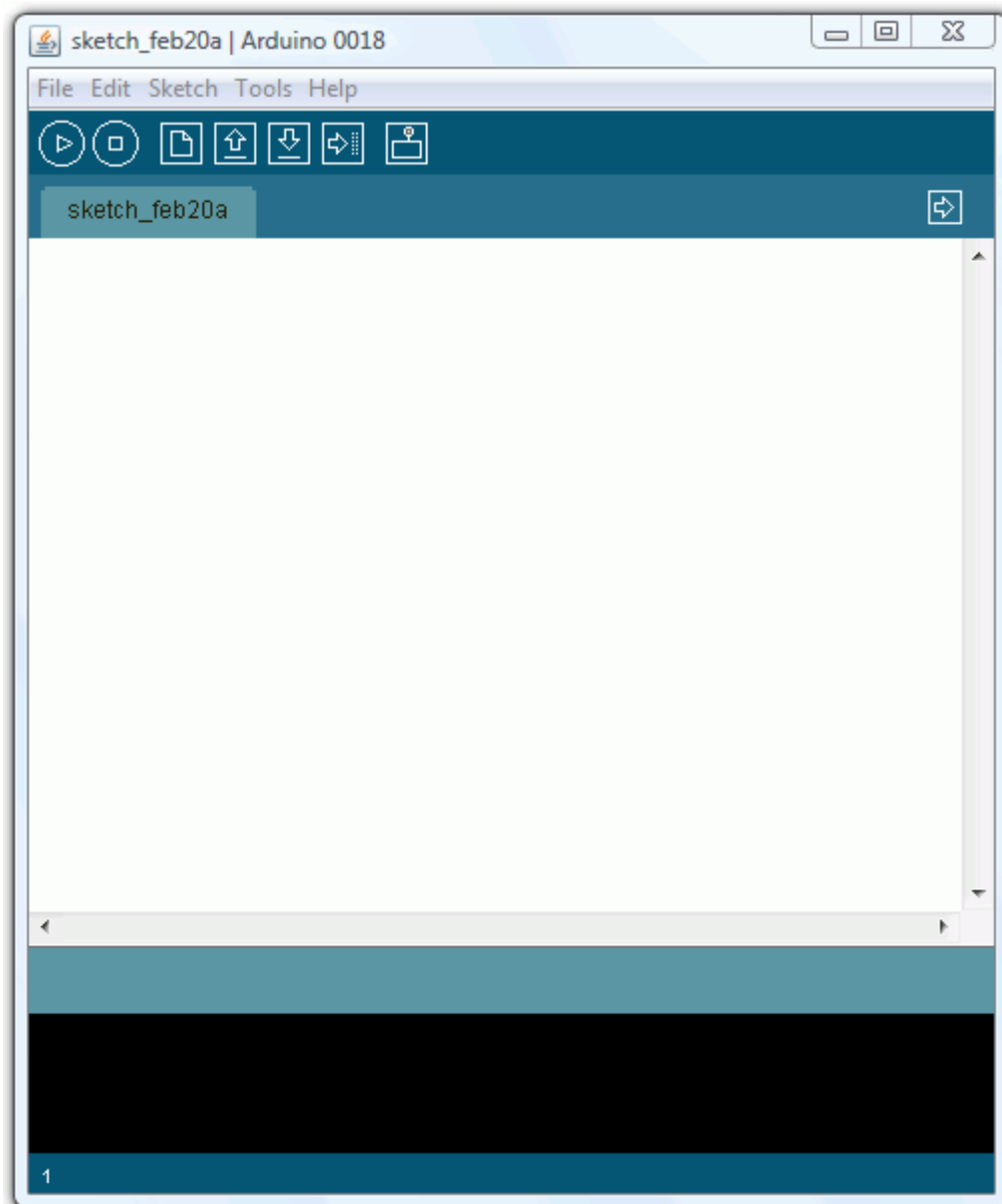


Figure 21: Fenêtre type

Principe général d'utilisation Le code écrit avec le logiciel ARDUINO est appelé un programme (ou une séquence - sketch en anglais) : Ces programmes sont écrits dans l'éditeur de texte. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte. la zone de messages donne l'état de l'opération en cours lors des sauvegardes, des exportation et affiche également les erreurs. La console texte affiche les messages produits par le logiciel ARDUINO incluant des messages d'erreur détaillés et autres informations utiles. La barre de boutons vous permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauver votre code, et ouvrir le moniteur série. la barre des menus vous permet d'accéder à toutes les fonctionnalités du logiciel ARDUINO. 3. Description de la barre des boutons Vérifier/compiler : Vérifie le code

à la recherche d'erreur. Stop : Stoppe le moniteur série ou les autres boutons activés. Nouveau : Crée un nouveau code (ouvre une fenêtre d'édition vide)

6Ouvrir : Ouvre la liste de tous les programmes dans votre "livre de programmes". Cliquer sur l'un des programmes l'ouvre dans la fenêtre courante. Note : en raison d'un bug dans Java, ce menu ne défile pas. Si vous avez besoin d'ouvrir un programme loin dans la liste utiliser plutôt le menu File > Sketchbook. Sauver : Enregistre votre programme. Transférer vers la carte : Compile votre code et le transfère vers la carte ARDUINO. Voir ci-dessous "Transférer les programmes" pour les détails. Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.

4. Description des menus Des commandes complémentaires sont disponibles dans cinq menus : File (Fichier), Edit (Editer), Sketch (Programme ou Séquence), Tools (Outils), Help (Aide). Le menu est sensible au contexte ce qui signifie que seulement les items correspondant au travail en cours sont disponibles.

7Menu File (Fichier) : Propose toutes les fonctionnalités usuelles pour gérer les fichiers, Sketchbook (Programme) : Fonctionnalité vous permettant d'avoir accès directement à tous vos programmes dans votre répertoire de travail. Voir ci-dessous pour les détails. Exemples (Exemples) : Cet item vous propose un menu déroulant vers toute une série de programmes d'exemples disponibles. Menu Edit (Editer): Copy for forum (copier pour le forum) : Copie le code du programme dans le presse-papier dans un format approprié pour poster sur le forum, avec coloration syntaxique complète. Copy as HTML (copier en tant qu'html) : Copie le code de votre programme dans le presse-papier en tant qu'html, au format adapté pour être intégré dans des pages web. Menu Sketch (Programme): Verify/Compile (Vérifier/compiler) : Vérifie le code à la recherche d'erreurs Import Library (Importer la librairie) : Ajoute une librairie à votre programme en insérant l'instruction #include dans votre code. Pour plus de détails, voir "librairies" ci-dessous. Show Sketch Folder (Montrer le répertoire du programme) : Ouvre le répertoire courant du programme sur votre bureau

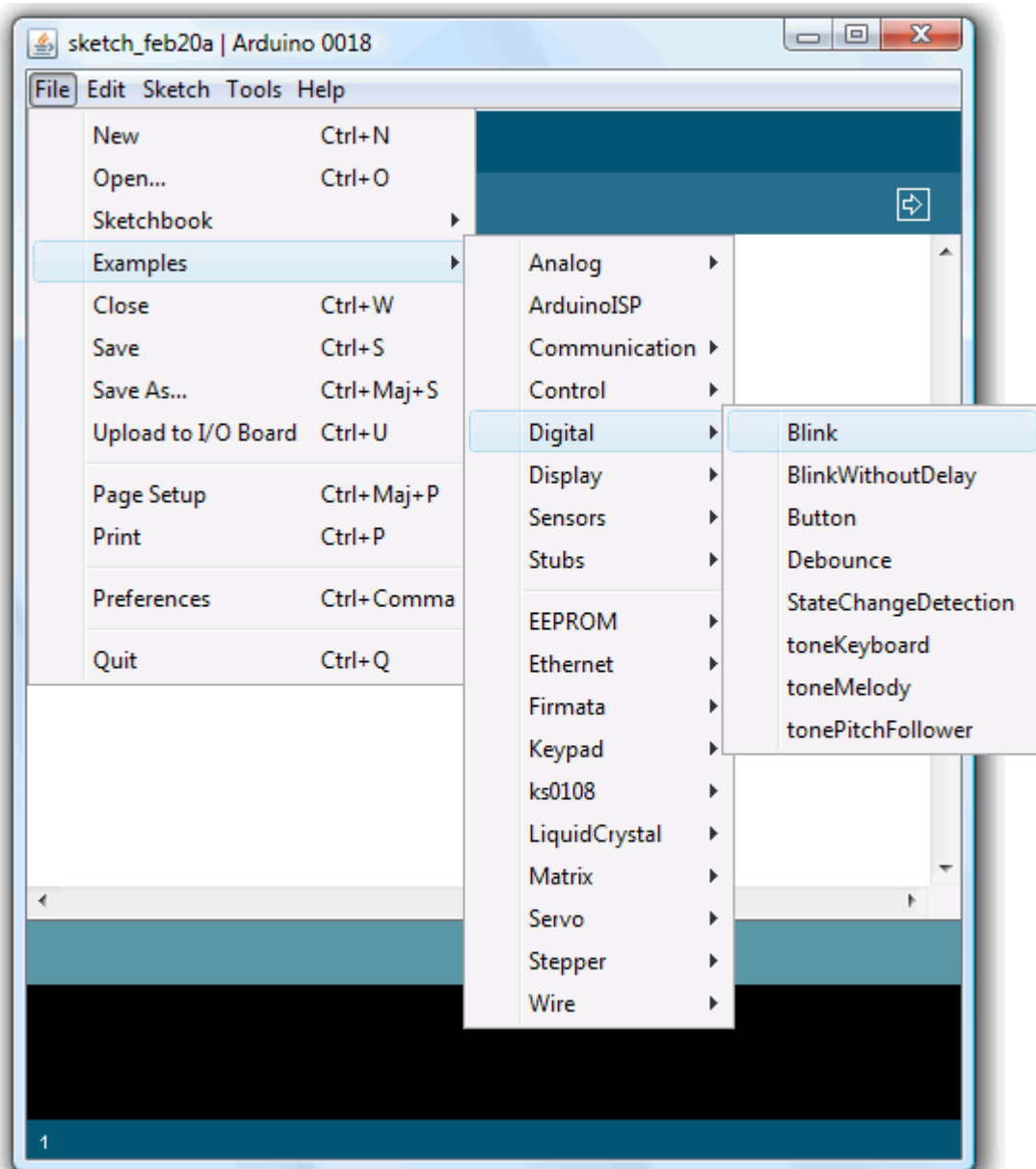


Figure22: chemin vers blink

8 Add File... (Ajouter un fichier): Ajoute un fichier source à votre programme (il sera copier à partir de sa localisation courante. Le nouveau fichier apparaît dans un nouvel onglet dans la fenêtre d'édition. Les fichiers peuvent être retirés du programme en utilisant le menu "tab". Menu Tools (Outils) : Auto Format (Mise en forme Automatique) : Cette fonction formate votre code joliment : c'est à dire ajuste le code de façon à ce que les accolades soient alignées et que ce que les instructions entre les accolades soient davantage décalées. Board (Carte) : Sélectionne la carte ARDUINO que vous utilisez. Voir ci-dessous pour la description des différentes cartes. Serial Port (Port Série) : Ce menu contient tous les ports séries (réels ou virtuels) présents sur votre ordinateur. Il est automatiquement mis à jour à chaque fois que vous ouvrez le niveau supérieur

du menu outil. Burn Bootloader (Graver le bootloader) : Cette fonctionnalité vous permet de graver le bootloader dans le microcontrôleur sur une carte ARDUINO. Ceci n'est pas nécessaire pour une utilisation normale de votre carte ARDUINO (le bootloader est déjà gravé dans votre carte quand vous l'achetez) mais peut être utile si vous achetez un nouvel ATmega (qui sera normalement livré sans bootloader). Assurez vous que vous avez sélectionné la carte correcte dans le menu Boards avant de graver le bootloader. Si vous utilisez un AVR ISP, vous devez sélectionner l'item correspondant à votre programmeur dans le menu Serial Port. Menu Help Propose tout ce qui peut être utile pour être aidé lors de l'écriture des programmes, notamment un lien vers la référence du langage en anglais. 5. Sketchbook ("Livre de programmes")

9 Le logiciel ARDUINO intègre le concept d'un "sketchbook" (livre de programme) : un endroit réservé pour stocker vos programmes. Les programmes que vous mettez dans votre "sketchbook" pourront être ouvert directement depuis le menu File > Sketchbook ou à l'aide du bouton Open (Ouvrir) dans la barre d'outils. La première fois que vous démarrer le logiciel ARDUINO, un chemin automatique sera créé pour votre "sketchbook". Vous pouvez voir ou modifier cette localisation depuis le menu File > Preferences. 6. Onglets, fichiers multiples et compilation Vous permet de gérer les programmes avec plus d'un fichier (chaque fichier apparaissant dans son propre onglet). Ces fichiers doivent être des fichiers ARDUINO normaux (no extension), des fichiers C (extension.c), des fichiers C++ (.cpp) ou des fichiers d'entête (.h). 7. Transfert des programmes vers la carte ARDUINO Saisir votre programme et vérifier le code. On suppose ici qu'un programme correctement écrit se trouve dans la fenêtre éditeur. Pour votre première programmation de la carte, aller dans le menu File>Examples>Digital>Blink : un programme s'ouvre avec du code dans la fenêtre éditeur. Appuyez alors sur le bouton Verify de la barre d'outils pour lancer la vérification du code : Si tout va bien, aucun message d'erreur ne doit apparaître dans la console et la zone de message doit afficher Done Compiling attestant que la vérification s'est bien déroulée.

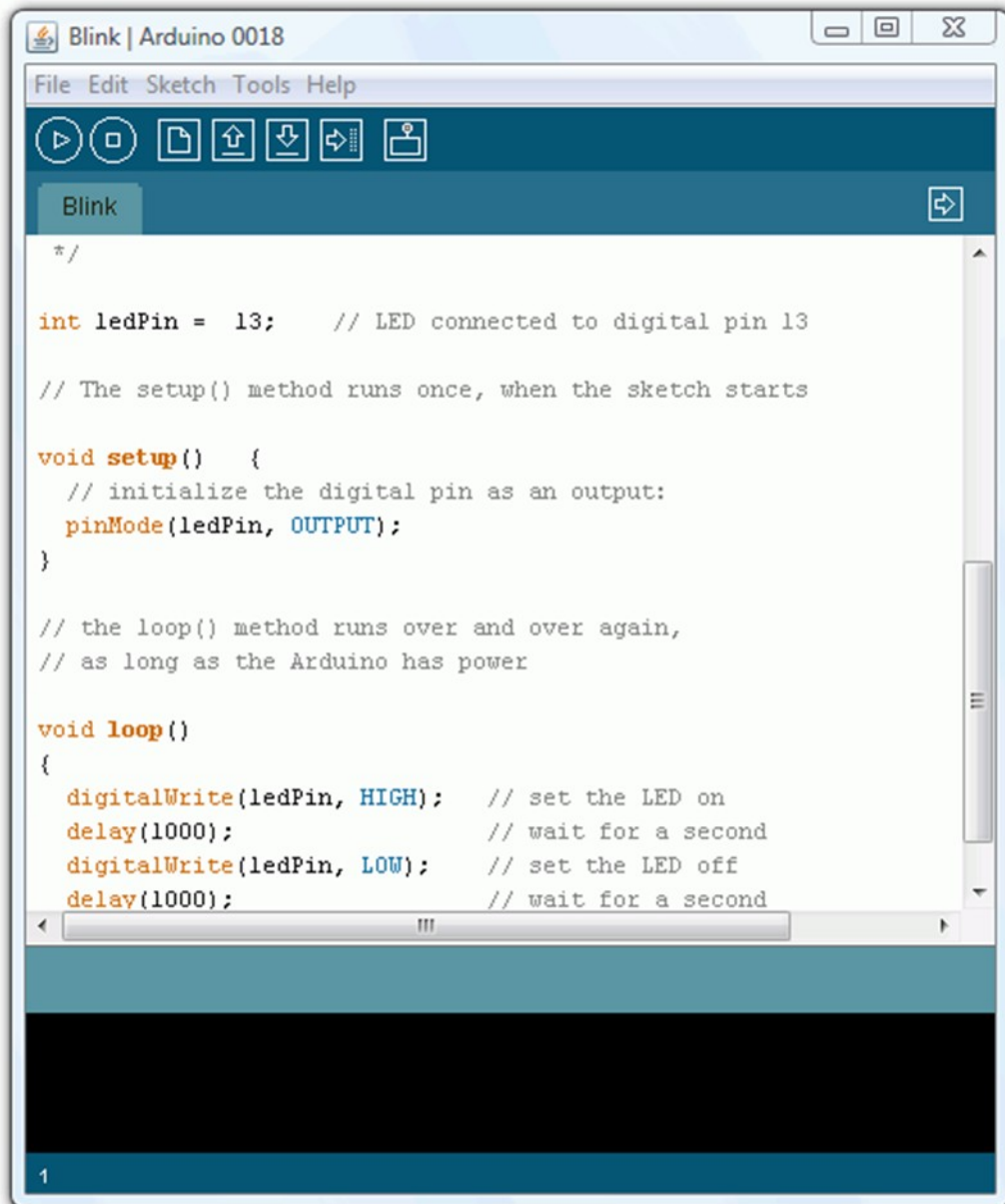


Figure 23: compilation

10 Sélectionner le bon port série (et la bonne carte ARDUINO...). Avant de transférer votre programme vers la carte ARDUINO, vous devez vérifier que vous avez bien sélectionné la bonne carte ARDUINO depuis le menu **Tools>Board** (Outils>Carte). Les cartes sont décrites ci-dessous. Votre carte doit évidemment être connectée à l'ordinateur via un câble USB. Vous devez également sélectionner le bon port série depuis le menu **Tools > Serial Port** (Outils > Port Série) : Sur un Mac, le port série ressemble probablement à quelque chose comme `/dev/tty.usbserial-1b1` (pour une carte USB), ou `/dev/tty.usa19qw1b1p1.1` (pour une carte série connectée avec un adaptateur USB-vers- Série). Sous Windows, c'est probablement COM1 ou COM2 (pour une carte série) ou COM4, COM5, COM7 ou supérieur (pour une carte USB) - pour trouver le bon, vous pouvez chercher dans la rubrique des ports série USB dans la section

des ports du panneau de configuration ou du gestionnaire de périphériques. Sous Linux, ça devrait être /dev/ttyusb0, /dev/ttyusb1 ou équivalent.

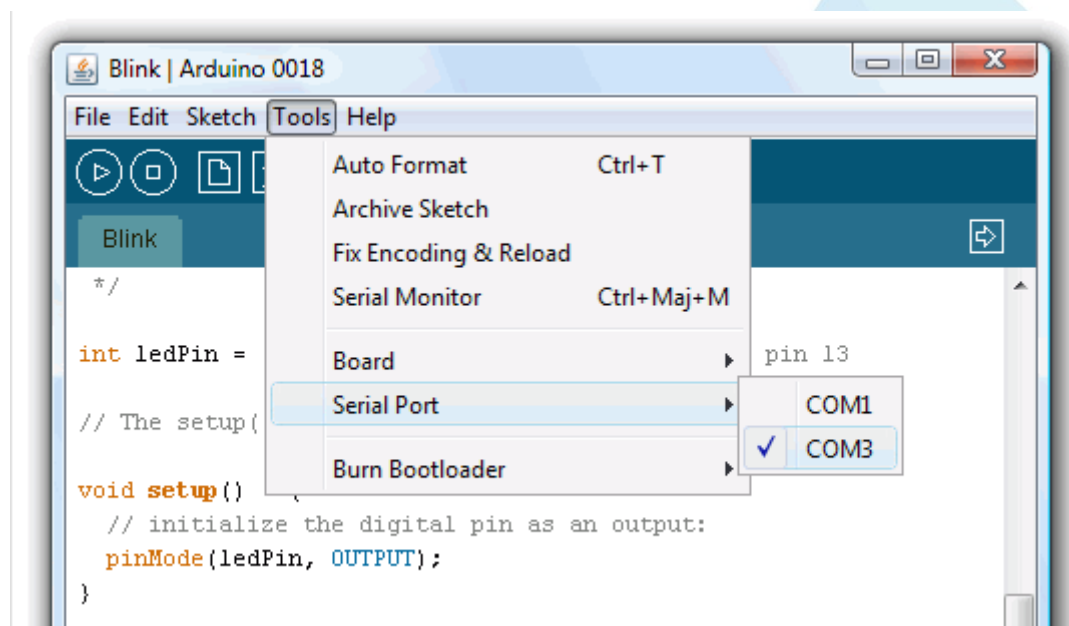


Figure 24: Choix du port

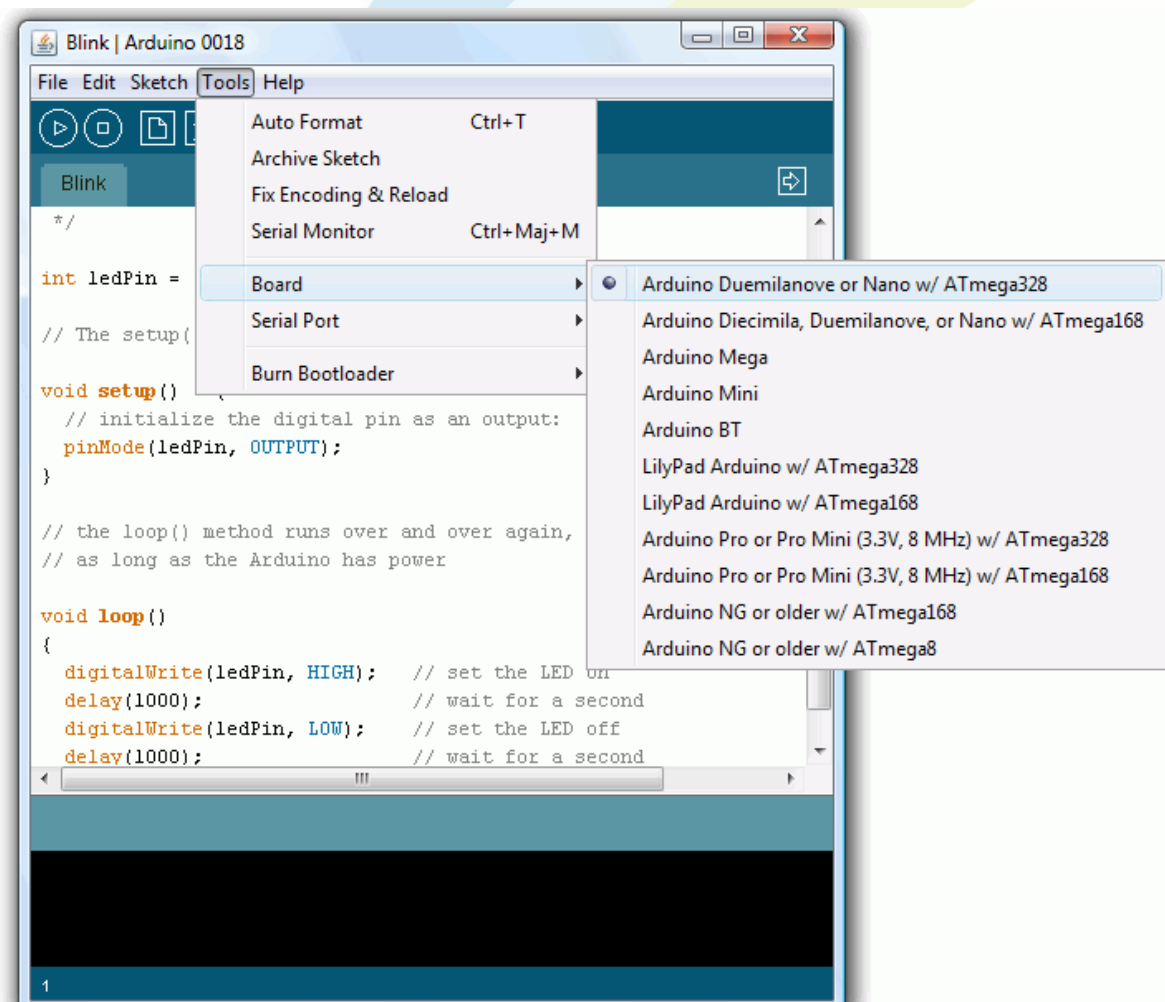


Figure 25: Choix de ARDUINO (dépend du type de carte)

11 Clic sur le bouton UPLOAD (Transfert du programme) Une fois que vous avez sélectionné le bon port série et la bonne carte ARDUINO, cliquez sur le bouton UPLOAD (Transfert vers la carte) dans la barre d'outils, ou bien sélectionner le menu File>Upload to I/O board (Fichier > Transférer vers la carte). Avec les versions récentes (Duemilanove notamment), la carte ARDUINO va alors automatiquement se réinitialiser et démarrer le transfert. Avec les versions précédentes qui ne sont pas équipées de l'auto-réinitialisation, vous devez appuyer sur le bouton "reset" de la carte juste avant de démarrer le transfert. A ce moment précis ; Sur la plupart des cartes, vous devez voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert, le bouton devient jaune et le logiciel ARDUINO affiche un message indiquant que le transfert est en cours :

12 Une fois le transfert terminé, le logiciel ARDUINO doit afficher un message indiquant que le transfert est bien réalisé, ou montrer des messages d'erreurs... (Reprendre dans ce cas la procédure, et si le problème persiste, voir la page "Dépannage" Le programme transféré avec succès se lance. Quand vous transférez un programme, en utilisant le bootloader ARDUINO, un petit programme (code binaire) a été chargé dans le microcontrôleur sur votre carte ARDUINO. Cette technique vous permet comme vous avez pu le voir de transférer votre programme sans aucun matériel externe. Une fois le transfert terminé, le bootloader est actif une petite seconde ("écoute" pour voir si un nouveau programme arrive...) une fois que la carte est réinitialisée à la fin du transfert ; puis le dernier programme programmé dans la carte s'exécute. Note : Le bootloader fait clignoter la LED de la carte (broche 13) quand il démarre (c'est à dire quand la carte est réinitialisée).

8. Bibliothèques Les bibliothèques fournissent des fonctions nouvelles que vous pouvez utiliser dans vos programmes, par exemple pour utiliser un matériel précis (un afficheur LCD...) ou manipuler des données. Pour utiliser une bibliothèque, la sélectionner depuis le menu Sketch > Import Library (Programme > Importer

13 ou plusieurs instructions **#include** au début de votre programme et compilera la bibliothèque avec votre programme. Puisque les bibliothèques sont transférées dans la carte avec votre programme, elle augmente la quantité de mémoire utilisée. Si un programme ne nécessite plus d'une bibliothèque, effacer simplement l'instruction **#include** correspondante au début de votre code. Il y a toute une

liste de bibliothèques dans la référence du langage ARDUINO. Certaines bibliothèques sont incluses dans le logiciel ARDUINO (la bibliothèque Serial pour les communications série notamment). D'autres bibliothèques peuvent être téléchargées depuis différentes sources (voir notamment : Pour installer ces bibliothèques provenant de tiers, créer un répertoire appelé "libraries" dans votre répertoire "sketchbook". A ce niveau, dézipper les bibliothèques téléchargées. Par exemple, pour installer la bibliothèque DateTime, le fichier devrait être dans le dossier : /libraries/datetime de votre "sketchbook". 9. Le Moniteur Série Le logiciel ARDUINO intègre également un TERMINAL SERIE (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte ARDUINO et d'envoyer des caractères vers la carte ARDUINO. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes. Sur votre ordinateur, il faut ouvrir la fenêtre terminale de l'ide ARDUINO : pour ce faire, un simple clic sur le bouton « Sérial Monitor ».

14 La fenêtre « Terminal » s'ouvre alors : Il faut alors régler le débit de communication sur la même valeur que celle utilisée par le programme avec lequel nous allons programmer la carte ARDUINO :

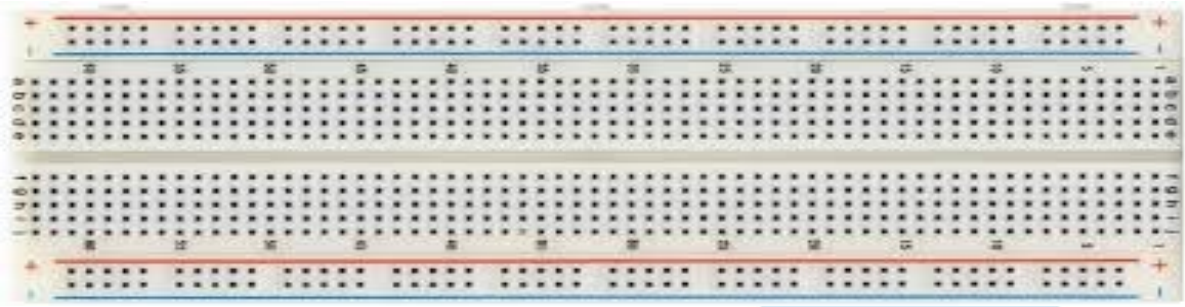
15 Pour envoyer des données vers la carte, saisir le texte dans le champ de saisie et cliquer le bouton "send" (envoi) ou appuyer sur enter. Noter que sur Mac ou Linux, la carte ARDUINO se réinitialisera quand vous vous connecterez avec le moniteur Série. Vous pouvez également communiquer avec la carte ARDUINO depuis l'interface graphique programmable Processing, Flash, MaxMSP, etc..



Figure 26: Icone de téléversement

1.6.2 De la configuration jusqu'au téléversement en passant par le codage au microcontrôleur

On va maintenant vous présenter un outil très pratique lorsque l'on fait ses débuts en électronique ou lorsque l'on veut tester rapidement ou facilement un montage. Cet accessoire s'appelle une breadboard (littéralement : Plaque à pain, techniquement : plaque d'essai sans soudure).



Principe de la breadboard :

Certes la plaque est pleine de trous, mais pas de manière innocente. En effet, la plupart d'entre eux sont reliés. Voici un petit schéma rapide qui va aider à la compréhension.

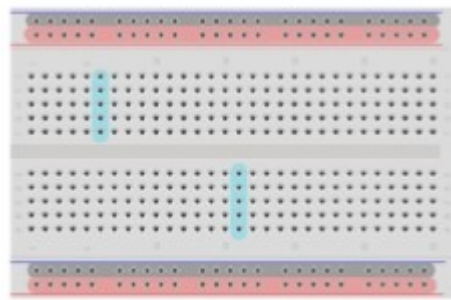


Figure 27: aperçu du breadbord

Comme vous pouvez le voir sur l'image, on a dessiné des zones. Les zones rouges et noires correspondent à l'alimentation. Souvent, on retrouve deux lignes comme celles-ci permettant de relier vos composants aux alimentations nécessaires. Par convention, le noir représente la masse et le rouge est l'alimentation (+5V , +12V , -5V ... ce que vous voulez y amener). Habituellement tous les trous d'une même ligne sont reliés sur cette zone. Ainsi, vous avez une ligne d'alimentation parcourant tout le long de la carte.

Ensuite, on peut voir des zones en bleu. Ces zones sont reliées entre elles par colonne. Ainsi, tous les trous sur une même colonne sont reliés entre eux. En revanche, chaque colonne est distincte. En faisant chevaucher des composants sur plusieurs colonnes vous pouvez les connecter entre eux.

Dernier point, vous pouvez remarquer un espace coupant la carte en deux de manière symétrique. Cette espace coupe aussi la liaison des colonnes. Ainsi, sur le dessin ci-dessus on peut voir que chaque colonne possède 5 trous reliés entre eux. Cet espace au milieu est normalisé et doit faire la largeur des circuits intégrés standards. En posant un circuit intégré à cheval au milieu, chaque patte de ce dernier se retrouve donc sur une colonne, isolée de la précédente et de la suivante.

Si vous voulez voir plus concrètement ce fonctionnement, je vous conseille d'essayer le logiciel Fritzing, qui permet de faire des circuits de manière assez simple et intuitive. Vous verrez ainsi comment les colonnes sont séparées les unes des autres. De plus, ce logiciel sera utilisé pour le reste de la présentation du système pour les captures d'écrans des schémas électroniques.

Nous nous allons créer des programmes, ou bien programmer.

Voici quelques exemples de programmes informatiques : Votre navigateur Web (Internet Explorer, Firefox, Chrome, ...) Votre lecteur multimédia (VLC, Windows Media Player, ...) Votre antivirus (avast!, antivira, ...)

Créer un programme informatique

Ecrire un programme informatique ne s'improvise pas, Il faut d'abord savoir en quel langage il s'écrit et apprendre la syntaxe de ce langage.

Qu'est-ce qu'un langage informatique ?

Un langage informatique est un langage qui va vous permettre de « parler » à votre ordinateur. Nous écrivons le programme avec un langage informatique, c'est-à-dire avec un langage que l'ordinateur peut comprendre.

Il existe, comme pour les langues, une diversité assez impressionnante de langage informatique. Heureusement, nous ne devons en apprendre qu'un seul. Le langage que nous devons apprendre s'appelle le langage ARDUINO.

Le compilateur

La programmation en électronique

Au jour d'aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi d'électronique embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Comment programmer de l'électronique ? Pour faire de l'électronique programmée, il faut un ordinateur et un composant programmable. Il existe tout plein de variétés différentes de

composants programmables, à noter : les microcontrôleurs, les circuits logiques programmables, ... Nous, nous allons programmer des microcontrôleurs.

A ce propos qu'est-ce que c'est qu'un microcontrôleur ?

Le microcontrôleur est un composant électronique programmable. On le programme par le biais d'un ordinateur grâce à un langage informatique, souvent propre au type de microcontrôleur utilisé. Nous n'entrerons pas dans l'utilisation poussée de ces derniers.

Voici la photo d'un microcontrôleur :



Figure 28: Photo de microcontrôleur

Composition des éléments internes d'un microcontrôleur

Un microcontrôleur est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Si on veut, c'est un ordinateur (sans écran, sans disque dur, sans lecteur de disque) dans un espace très restreints.

Ces différents éléments qui composent un microcontrôleur typique et uniquement ceux qui vont nous être utiles.

La mémoire Il en possède 4 types : La mémoire Flash : C'est celle qui contiendra le programme à exécuter (celui que nous allons créer). Cette mémoire est effaçable et réinscriptible (c'est la même qu'une clé USB par exemple) RAM : c'est la mémoire dite "vive", elle va contenir les

variables de votre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur (comme sur un ordinateur). EEPROM : C'est le disque dur du microcontrôleur. Vous pourrez y enregistrer des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. Les registres : c'est un type de mémoire utilisé par le processeur. Nous n'en parlerons pas tout de suite. La mémoire cache : c'est une mémoire qui fait la liaison entre les registres et la RAM. Nous n'en parlerons également pas tout de suite.

Le processeur C'est le composant principal du microcontrôleur. C'est lui qui va exécuter le programme que nous lui donnerons à traiter. On le nomme souvent le CPU.

Fonctionnement

Avant tout, pour que le microcontrôleur fonctionne, il lui faut une alimentation. Cette alimentation se fait en générale par du +5V. D'autres ont besoin d'une tension plus faible, du +3,3V.

Interface du logiciel : Lancement du logiciel

Lançons le logiciel en double-cliquant sur l'icône avec le symbole "infinie" en vert. C'est l'exécutable du logiciel.

Après un léger temps de réflexion, une image s'affiche :

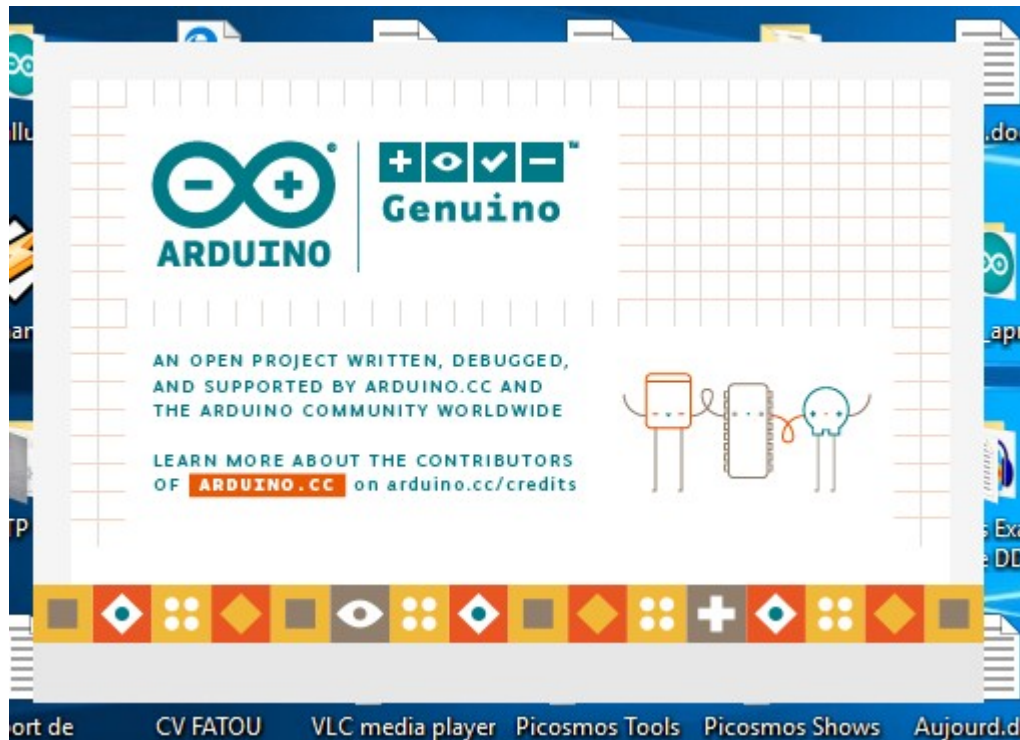


Figure 29 : lancement du logiciel ARDUINO

Cette fois, après quelques secondes, le logiciel s'ouvre. Une fenêtre se présente à nous :

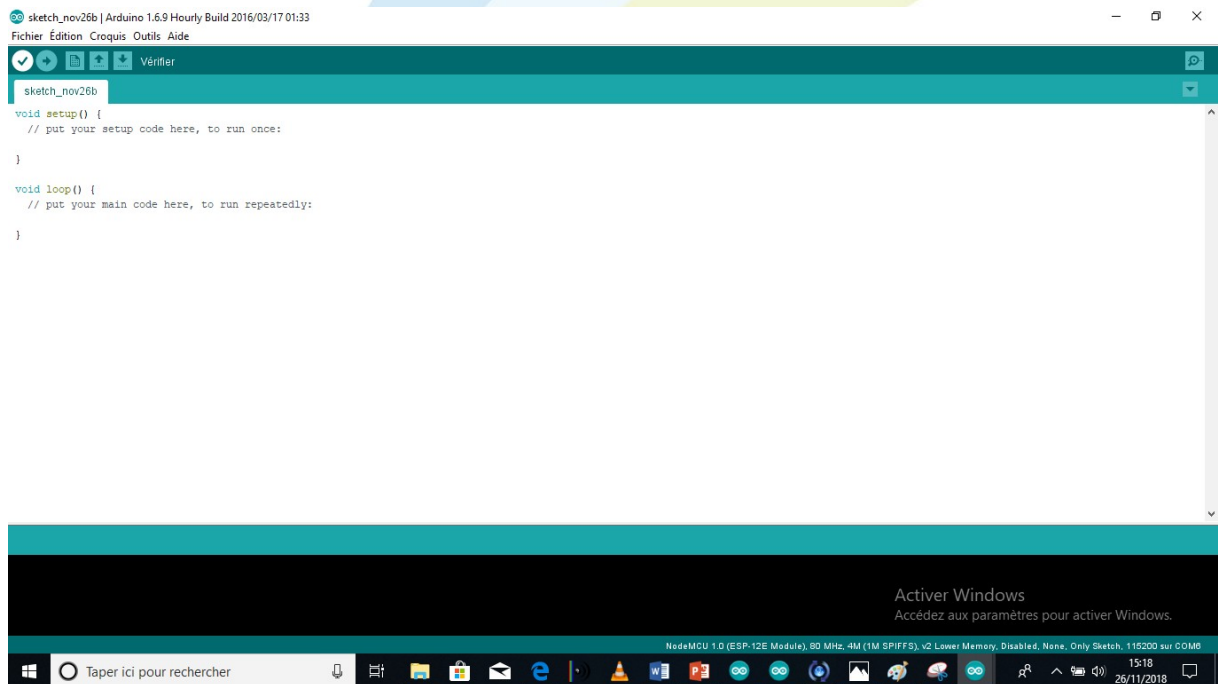


Figure 30 : fenêtre du logiciel ARDUINO

Ce qui saute aux yeux en premier, c'est la clarté de présentation du logiciel. On voit tout de suite son interface intuitive. Voyons comment se compose cette interface.

Présentation du logiciel

L'image précédente en plusieurs parties :

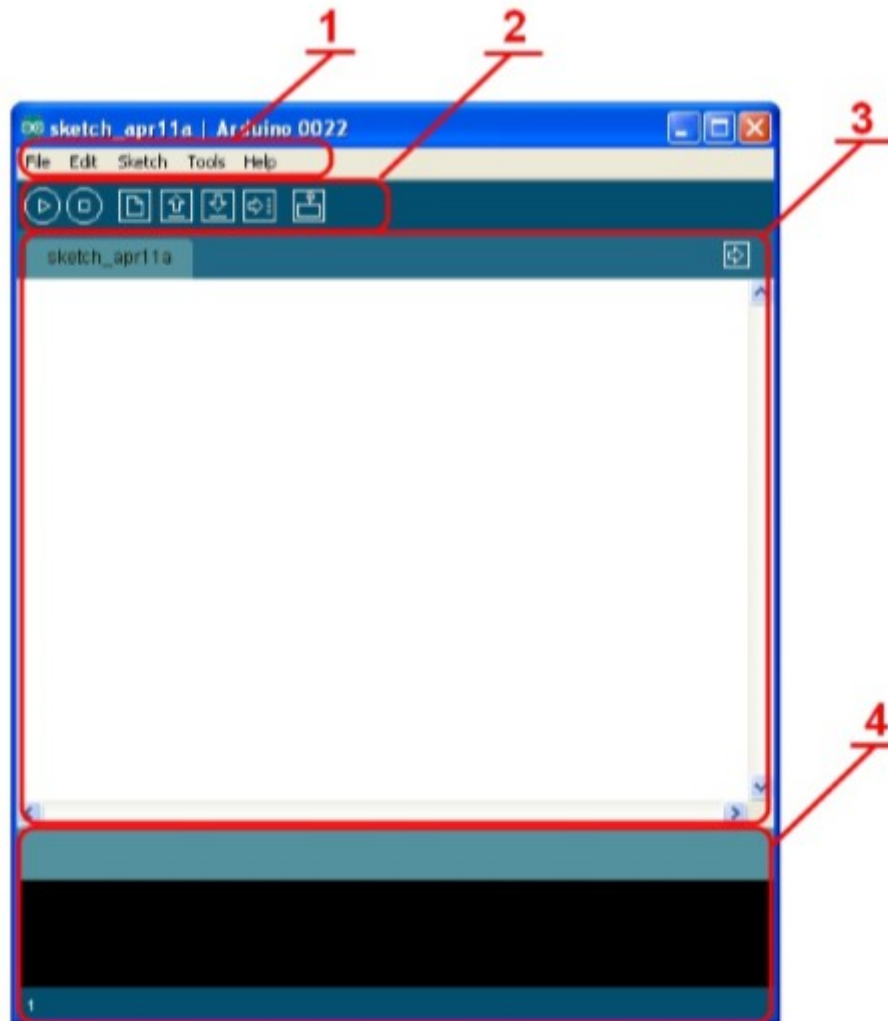


Figure 31: Présentation des parties principales du logiciel

Correspondance

Le cadre numéro **1** : ce sont les options de configuration du logiciel Le cadre numéro **2** : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes Le cadre numéro **3** : ce bloc va contenir le programme que nous allons créer Le cadre numéro **4** : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur. Approche et utilisation du logiciel Attaquons-nous plus sérieusement à l'utilisation du logiciel. La barre des menus est entourée en rouge et numérotée par le chiffre 1.

Le menu File

C'est principalement ce menu que l'on va utiliser le plus. Il dispose d'un certain nombre de choses qui vont nous être très utiles :

New (nouveau) : va permettre de créer un nouveau programme. Quand on appuie sur ce bouton, une nouvelle fenêtre, identique à celle-ci, s'affiche à l'écran Open... (ouvrir) : avec cette commande, nous allons pouvoir ouvrir un programme existant Save / Save as... (enregistrer / enregistrer sous...) : enregistre le document en cours / demande où enregistrer le document en cours Examples (exemples) : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existants ; avec ça, on pourra créer nos propres programmes.

Les boutons

Voyons à présent à quoi servent les boutons, encadrés en rouge et numérotés par le chiffre 2.



Figure 32 : Présentation des boutons

Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme

- Bouton 2 : Créer un nouveau fichier
- Bouton 3 : Sauvegarder le programme en cours
- Bouton 4 : On n'y touche pas pour l'instant
- Bouton 5 : Stoppe la vérification
- Bouton 6 : Charger un programme existant
- Bouton 7 : Compiler et envoyer le programme vers la carte Enfin, on va pouvoir s'occuper du matériel que vous devriez tous posséder en ce moment même : la carte ARDUINO !

Il faudra disposer du présent matériel pour pouvoir avoir accès aux interfaces. Les points importants de la carte ont été représentés en rouge.

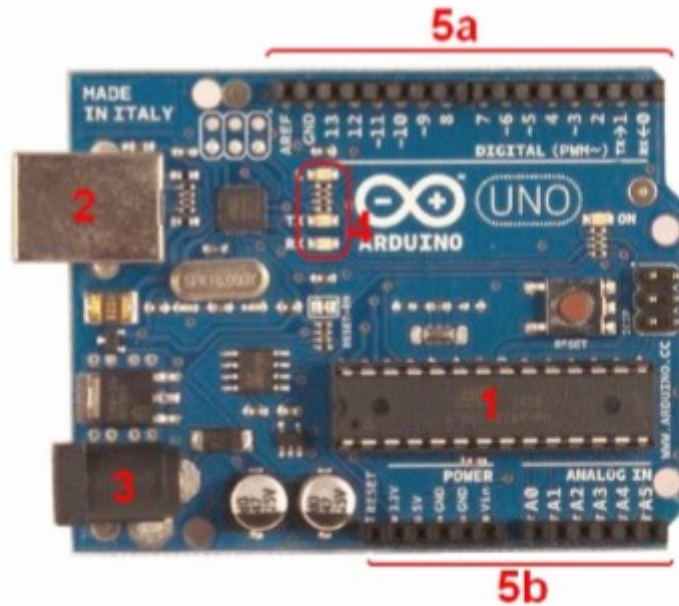


Figure 33: Présentation de la carte ARDUINO

Constitution de la carte

Voyons quels sont ces points importants et à quoi ils servent.

Le microcontrôleur

Voilà le cerveau de notre carte (en 1). C'est lui qui va recevoir le programme que nous aurons créé et qui va le stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur, ...

Alimentation

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en 2) ou bien par une alimentation externe (en 3) qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte. Pas de danger de tout griller donc. Veuillez seulement à respecter l'intervalle de 7V à 15V (même si le régulateur peut supporter plus, pas la peine de le retrancher dans ses limites)

Visualisation

Les trois "points blancs" entourés en rouge (4) sont en fait des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel. Nota : Quand on branche la carte au PC, elle clignote quelques secondes. Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

La connectique

La carte ARDUINO ne possédant pas de composants qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte (en 5a et 5b).

Par exemple, on veut connecter une LED sur une sortie du microcontrôleur. Il suffit juste de la connecter, avec une résistance en série, à la carte, sur les fiches de connections de la carte.

Cette connectique est importante et a un brochage qu'il faudra respecter. C'est avec cette connectique que la carte est "extensible", car l'on peut y brancher tous types de montages et modules ! Par exemple, la carte ARDUINO Uno peut être étendue avec des shields (ou modules), comme le « Shield Ethernet » qui permet de connecter cette dernière à internet.



Figure 34 : Une carte ARDUINO étendue avec un Ethernet Shield

Installation Afin d'utiliser la carte, il faut l'installer. Normalement, les drivers sont déjà installés sous GNU/Linux. Sous mac, il suffit de double cliquer sur le fichier .mkpg inclus dans le téléchargement de l'application ARDUINO et l'installation des drivers s'exécute de façon automatique. Sous Windows

Lorsque vous connectez la carte à votre ordinateur sur le port USB, un petit message en bas de l'écran apparaît. Théoriquement, la carte que vous utilisez doit s'installer toute seule. Cependant, si vous êtes sous Windows 7, il se peut que ça ne marche pas du premier coup. Dans ce cas, laissez la carte branchée puis ensuite allez dans le panneau de configuration. Une fois-là, cliquez

sur "**système**" puis dans le panneau de gauche sélectionnez "**gestionnaire de périphériques**". Une fois ce menu ouvert, vous devriez voir un composant avec un panneau "attention" jaune. Faites un clic droit sur le composant et cliquez sur "**Mettre à jour les pilotes**". Dans le nouveau menu, sélectionnez l'option "**Rechercher le pilote moi-même**". Enfin, il ne vous reste plus qu'à aller sélectionner le bon dossier contenant le driver. Il se trouve dans le dossier d'ARDUINO que vous avez dû décompresser un peu plus tôt et se nomme "**drivers**" (attention, ne descendez pas jusqu'au dossier "FTDI"). Par exemple, pour moi le chemin sera : **[le-chemin-jusqu'au-dossier]\ARDUINO-0022\ARDUINO-0022\drivers**

Il semblerait qu'il y est des problèmes en utilisant la version française d'ARDUINO (les drivers sont absents du dossier). Si c'est le cas, il vous faudra télécharger la version originale (anglaise) pour pouvoir installer les drivers.

Après l'installation et une suite de clignotement sur les micro-LED de la carte, celle-ci devrait être fonctionnelle; une petite LED verte témoigne de la bonne alimentation de la carte :



Figure 35 : carte connectée et alimentée

Tester son matériel

Avant de commencer à programmer la tête baissée, il faut, avant toutes choses, tester le bon fonctionnement de la carte. Nous allons tester notre matériel en chargeant un programme qui fonctionne dans la carte.

Le logiciel ARDUINO contient des exemples de programmes. Et bien ce sont ces exemples que nous allons utiliser pour tester la carte.

1ère étape : ouvrir un programme Nous allons choisir un exemple tout simple qui consiste à faire clignoter une LED. Son nom est Blink et vous le trouverez dans la catégorie Basics :

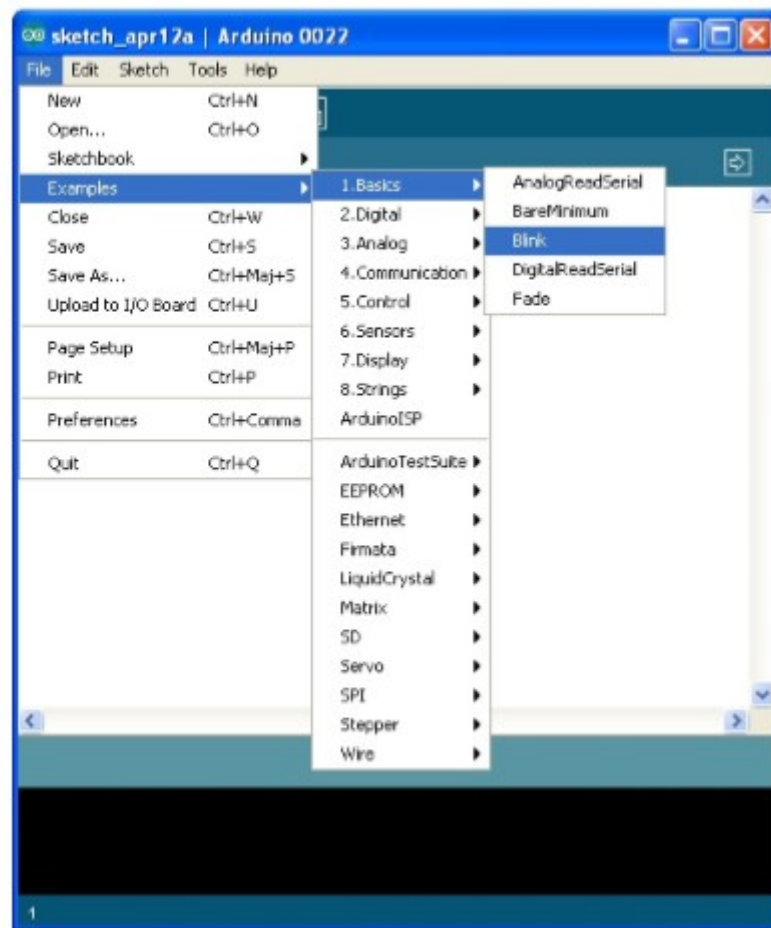


Figure 36 : Ouvrir le programme Blink

Une fois que vous avez cliqué sur Blink, une nouvelle fenêtre va apparaître. Elle va contenir le programme Blink. Vous pouvez fermer l'ancienne fenêtre qui va ne nous servir plus à rien.

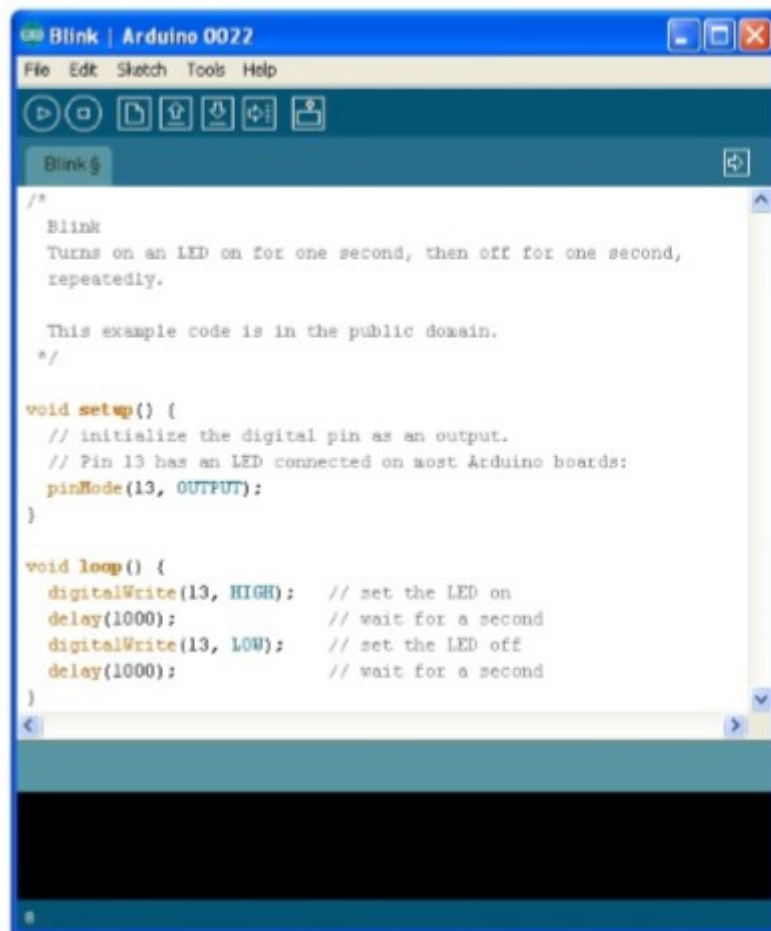


Figure 37: Contenu du programme Blink

2e étape Avant d'envoyer le programme Blink vers la carte, il faut dire au logiciel quel est le nom de la carte et sur quel port elle est branchée.

Choisir la carte que l'on va programmer. Ce n'est pas très compliqué, le nom de votre carte est indiqué sur elle. Pour nous, il s'agit de la carte "Uno". Allez dans le menu "Tools" ("outils" en français) puis dans "Board" ("carte" en français). Vérifiez que c'est bien le nom "ARDUINO Uno" qui est coché. Si ce n'est pas le cas, cochez-le.

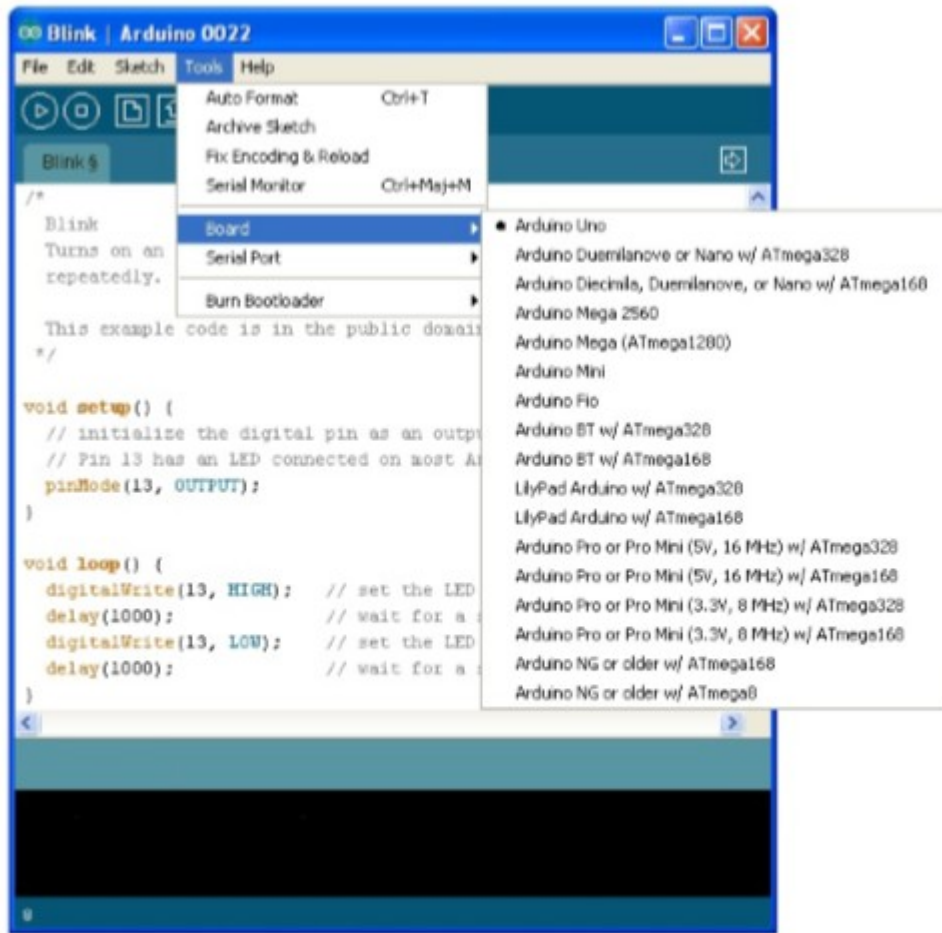


Figure 38 : Choix de la carte ARDUINO

Choisissez le port de connexion de la carte. Allez dans le menu Tools, puis Serial port. Là, vous choisissez le port COMX, X étant le numéro du port qui est affiché. Ne choisissez pas COM1 car il n'est quasiment jamais connecté à la carte. Dans mon cas, il s'agit de COM5 :

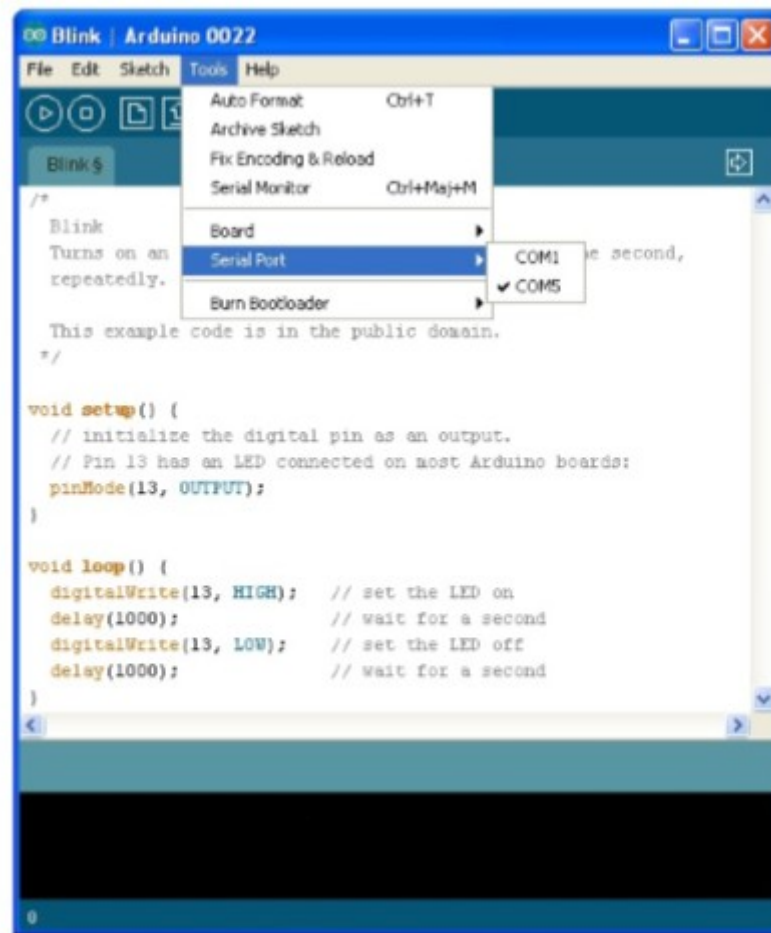


Figure 39 : Choix du port de connexion de la carte

Pour trouver le port de connexion de la carte, vous pouvez aller dans le gestionnaire de périphérique qui se trouve dans le panneau de configuration. Regardez à la ligne Ports (COM et LPT) et là, vous devriez avoir ARDUINO Uno (COMX).

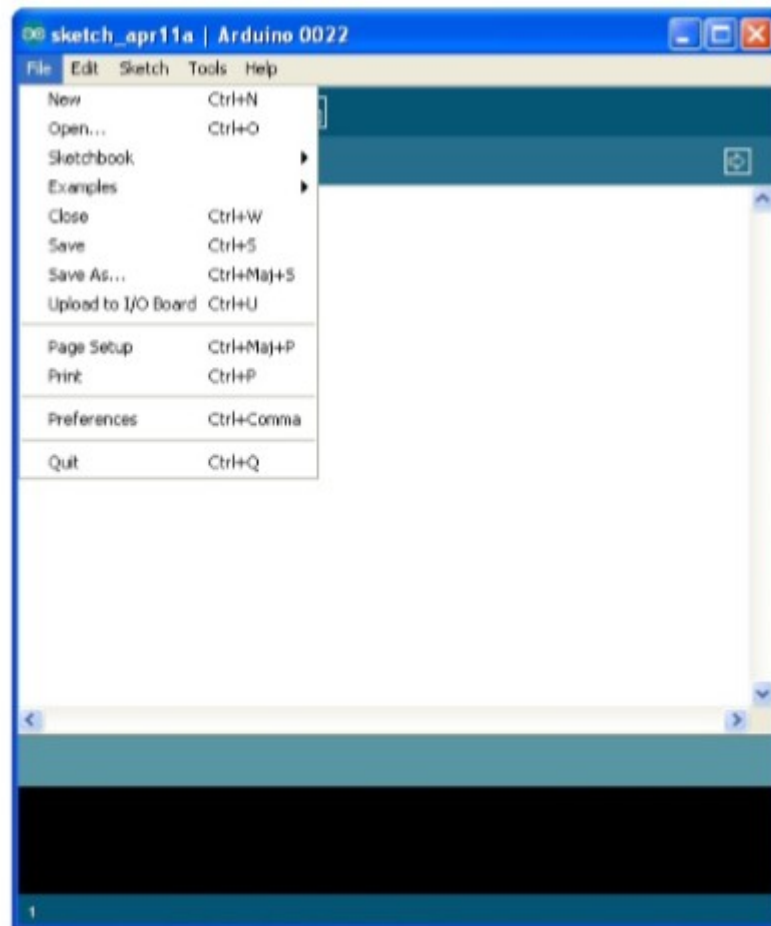


Figure 40 : Recherche du port de communication de la carte

Dernière étape Très bien. Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton Upload (ou "Télécharger" en Français), en jaune-orangé sur la photo :

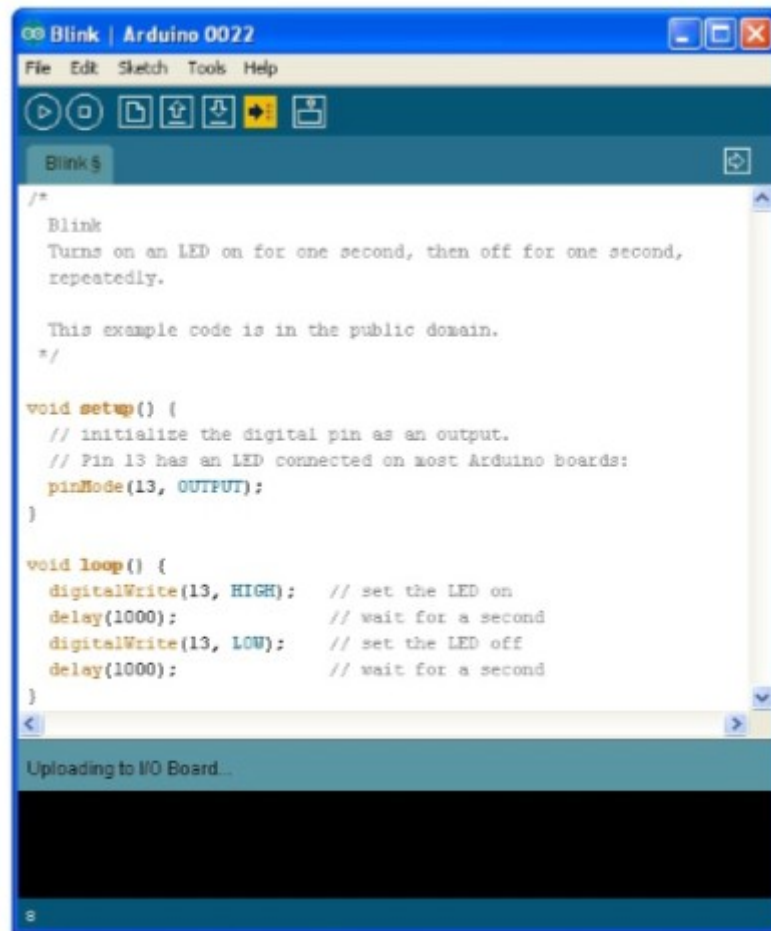


Figure 41: Envoi du programme Blink

En bas dans l'image, vous voyez le texte : "Uploading to I/O Board...", cela signifie que le logiciel est en train d'envoyer le programme dans la carte. Une fois qu'il a fini, il affiche un autre message :

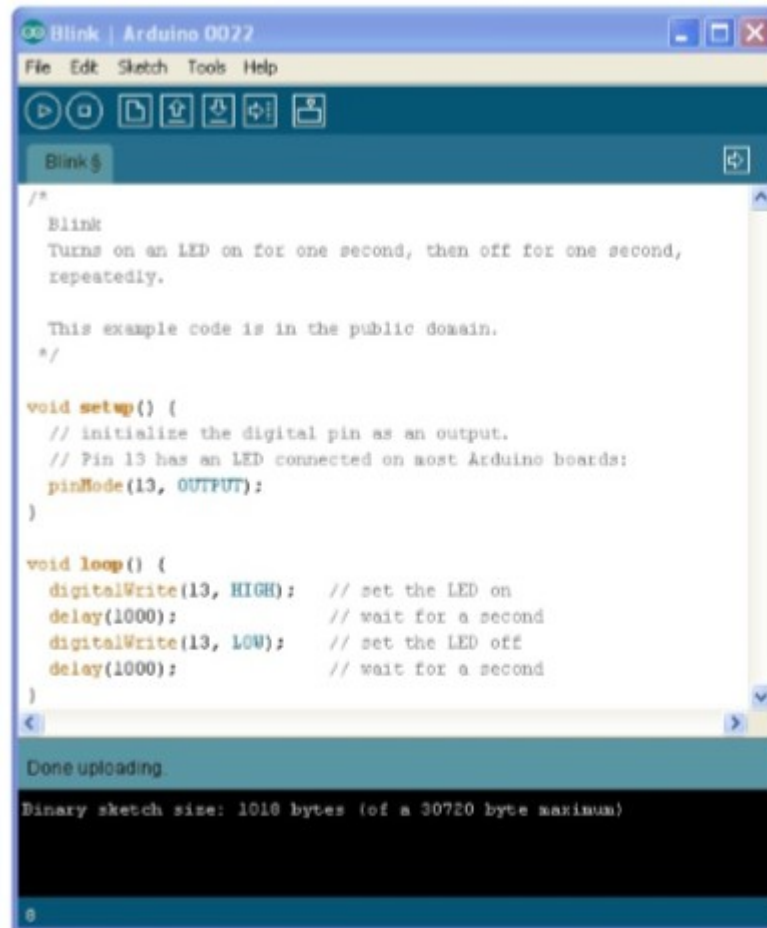


Figure 42: fin de l'upload

Le message afficher : "Done uploading" signale que le programme à bien été chargé dans la carte. Si votre matériel fonctionne, vous devriez avoir une LED sur la carte qui clignote : Si vous n'obtenez pas ce message mais plutôt une notification en rouge, pas d'inquiétude, le matériel n'est pas forcément défectueux.

En effet, plusieurs erreurs sont possibles: - l'IDE recompile avant d'envoyer le code, vérifier la présence d'erreur - La voie série est peut-être mal choisi, vérifier les branchements et le choix de la voie série - l'IDE est codé en JAVA, il peut-être capricieux et bugger de temps en temps (surtout avec la voie série...) : réessayez l'envoi!



Figure 43 : LED sur la carte qui clignote Toutes ces étapes, vous devrez les faire avant d'utiliser la carte pour vérifier son bon fonctionnement. C'est très important

1.1.1 ASSEMBLAGE ET ELABORATION DU DISPOSITIF

Après avoir rassemblé tout le matériel requis on passe à l'assemblage du collier connecté suivant le schéma définit.

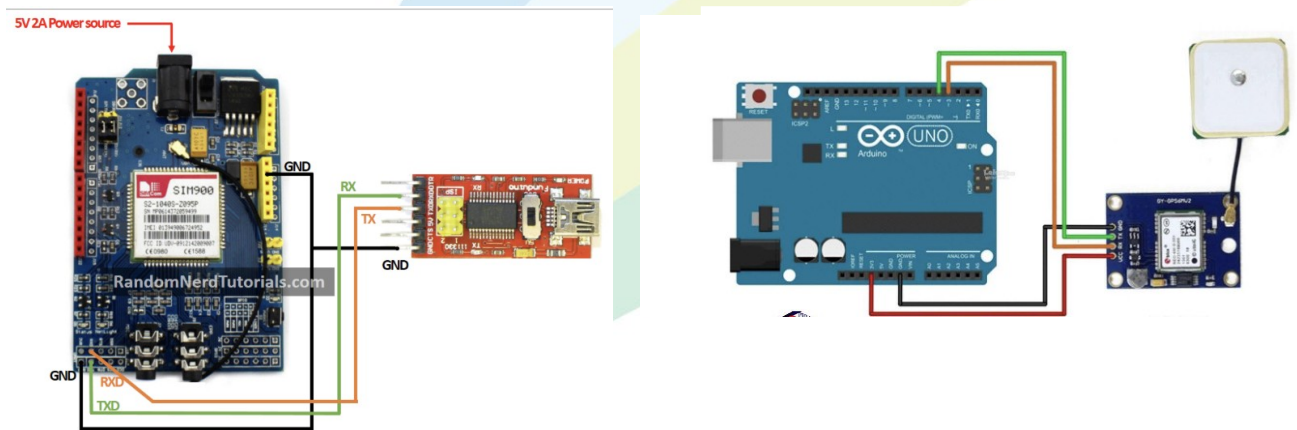


Figure 44: Schéma des circuits du dispositif

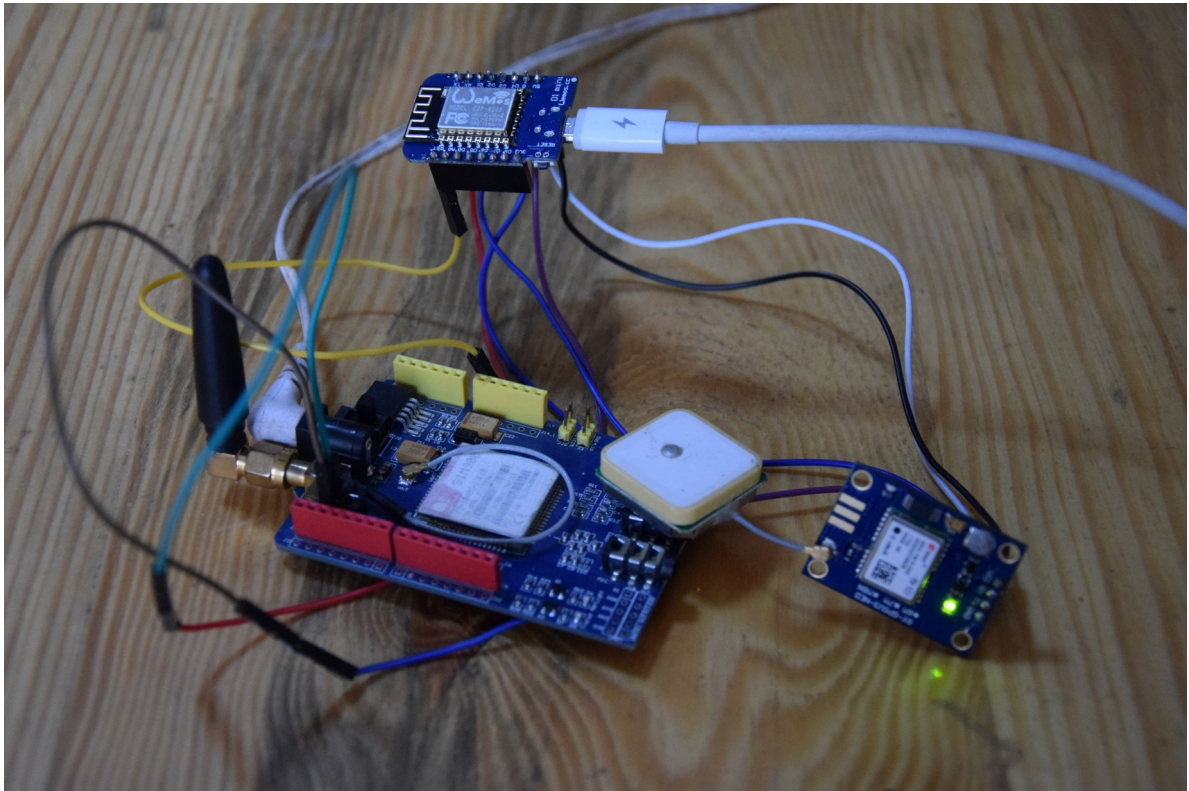


Figure 45: montage final du dispositif

Ensuite on UPLOAD notre code comme susmentionné et expliqué dans les lignes précédentes grâce à l'application de ARDUINO.

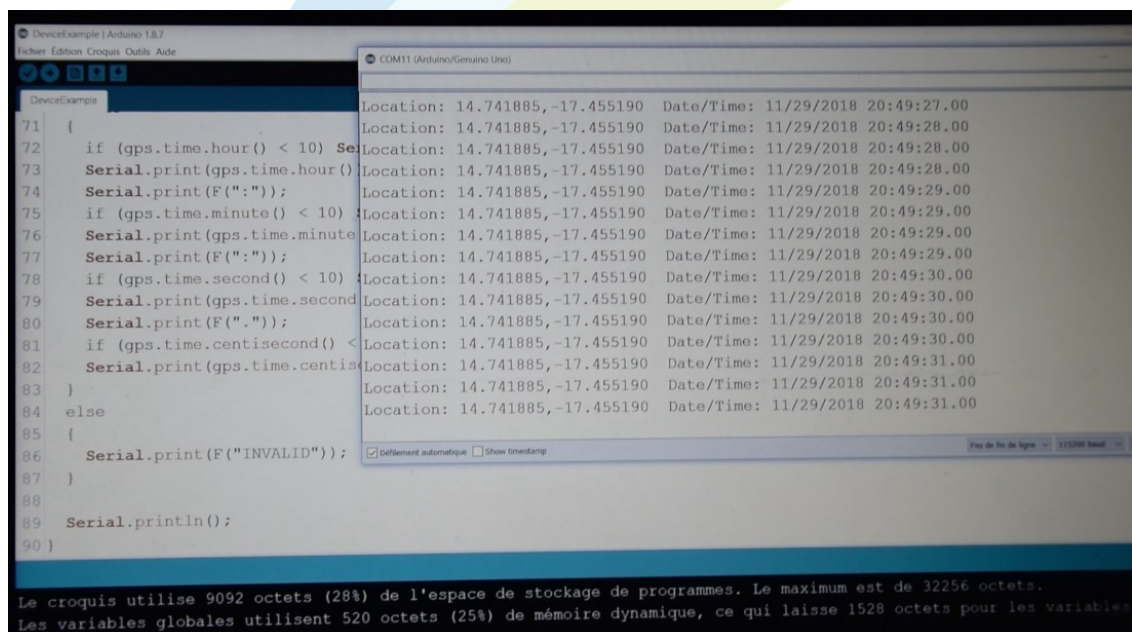


Figure 46: capture de l'interface lors de l'upload après montage du dispositif

2) CHAPITRE 4 : GESTION DU BETAIL (APPLICATION MOBILE)

1.7 ETUDE ET CONCEPTION DE L'APPLICATION MOBILE

2.1.1 ETUDE

A la suite d'une étude menée aux niveaux de certaines bergeries on a su élaborer les besoins fonctionnels dont devrait avoir notre application. Cette dernière devrait résoudre beaucoup de contrainte, tel n'est pas le cas malheureusement car toutes les contraintes des éleveurs ne pourraient être résolues avec une seule application. Cependant l'application conçue s'est limitée pour le moment aux contraintes majeures que rencontraient les éleveurs à savoir :

- ❖ Le suivi complet d'un sujet du point de vue descendance et race (naissance, paternité, etc.)
- ❖ Le rappel des traitements des sujets ainsi que de leurs vaccins et des sujets prêts pour la lutte.
- ❖ Le suivi des dates de saillie, les répétitions de chaleur et les prédictions des mises bas et des naissances.
- ❖ Et également la géolocalisation du cheptel pour les détenteurs du collier connecté.

2.1.2 CONCEPTION DE L'APPLICATION MOBILE

1) OUTILS DE CONCEPTION ET BASE DE DONNEES

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, MacOS et Linux.

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.

2.1.2.1.1 Langage de programmation et environnement

Comme outil de conception nous avons choisi Android Studio pour concevoir une application Android pouvant répondre aux exigences.

2.1.2.1.2 Outils Base de données

Pour la base de données vu qu'elle devra recueillir des données (coordonnées GPS et autres) venant du collier connecté et donc jouer office d'intermédiaire entre le collier et l'application mobile nous avons configuré une base de données en ligne à partir d'un hébergeur gratuit nommée ALWAYS DATA. Ce dernier offre un nom de domaine gratuit avec une possibilité d'avoir derrière une base de données de 500 mégaoctets suffisant pour sauvegarder nos données et les recueillir au moment voulu.

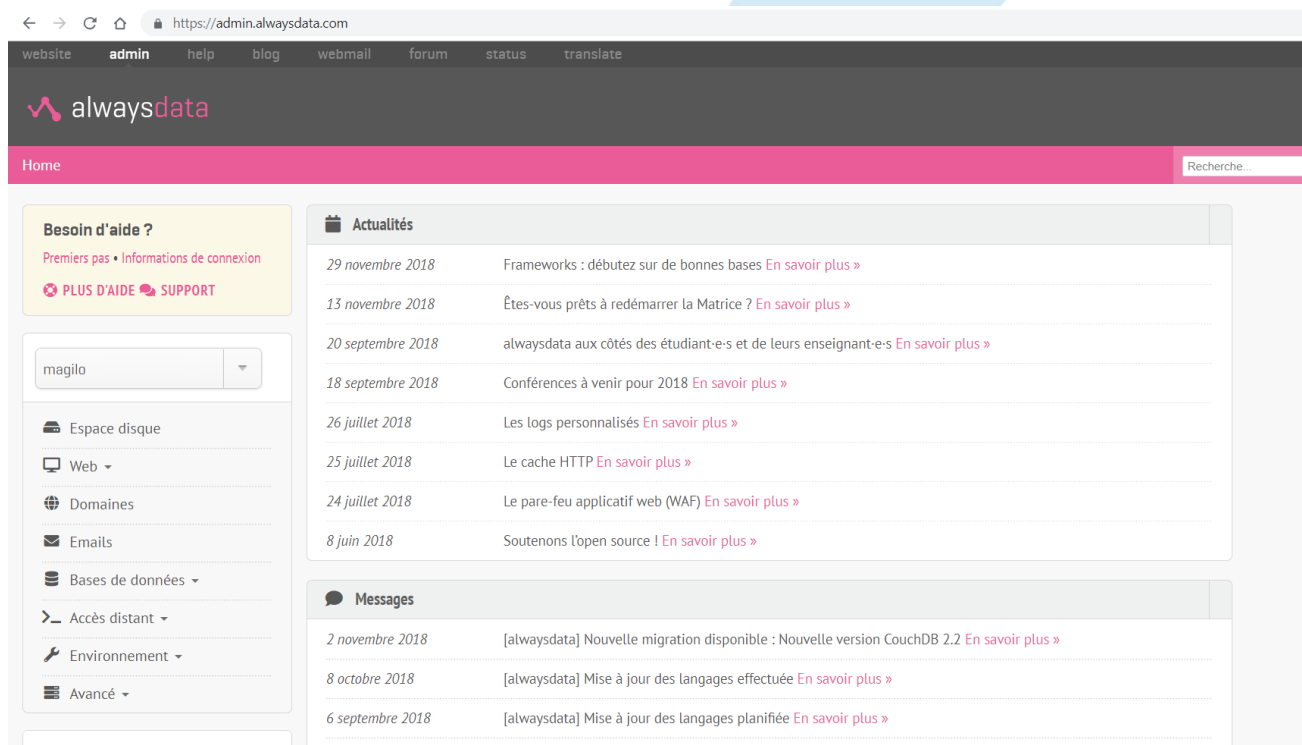


Figure 47: aperçu de la plateforme de ALWAYS DATA

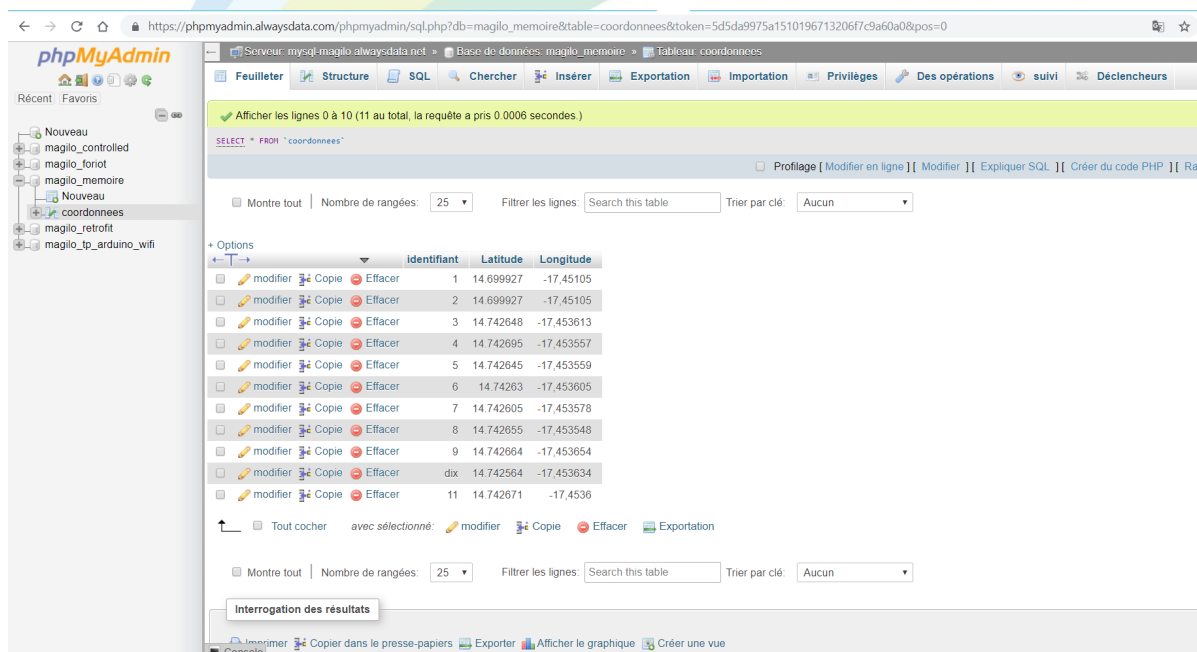


Figure 48: Aperçu de la base de données créée pour le système avec les coordonnées émises par le collier

Par la pratique avec l'IDE Android Studio, On obtient une application avec de modestes interfaces dont un SPLASH SCREEN (ou interface d'accueil) avec une animation à l'ouverture de l'application. (Voir capture ci-dessous)

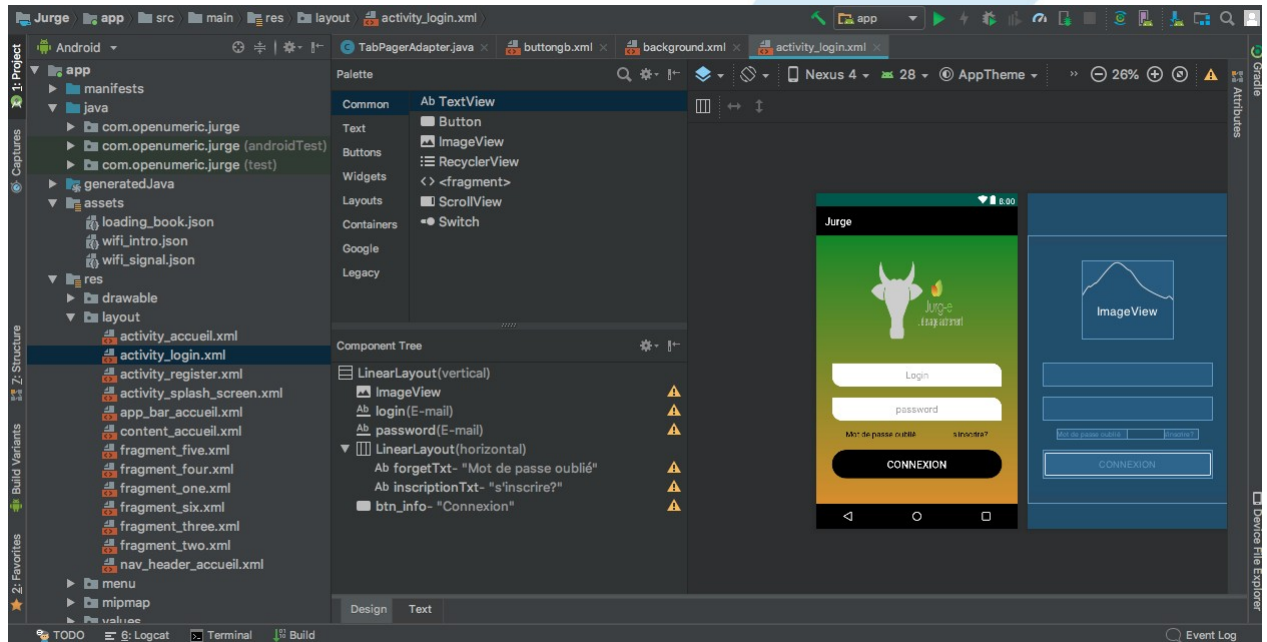


Figure 49: capture de Android studio pendant la conception

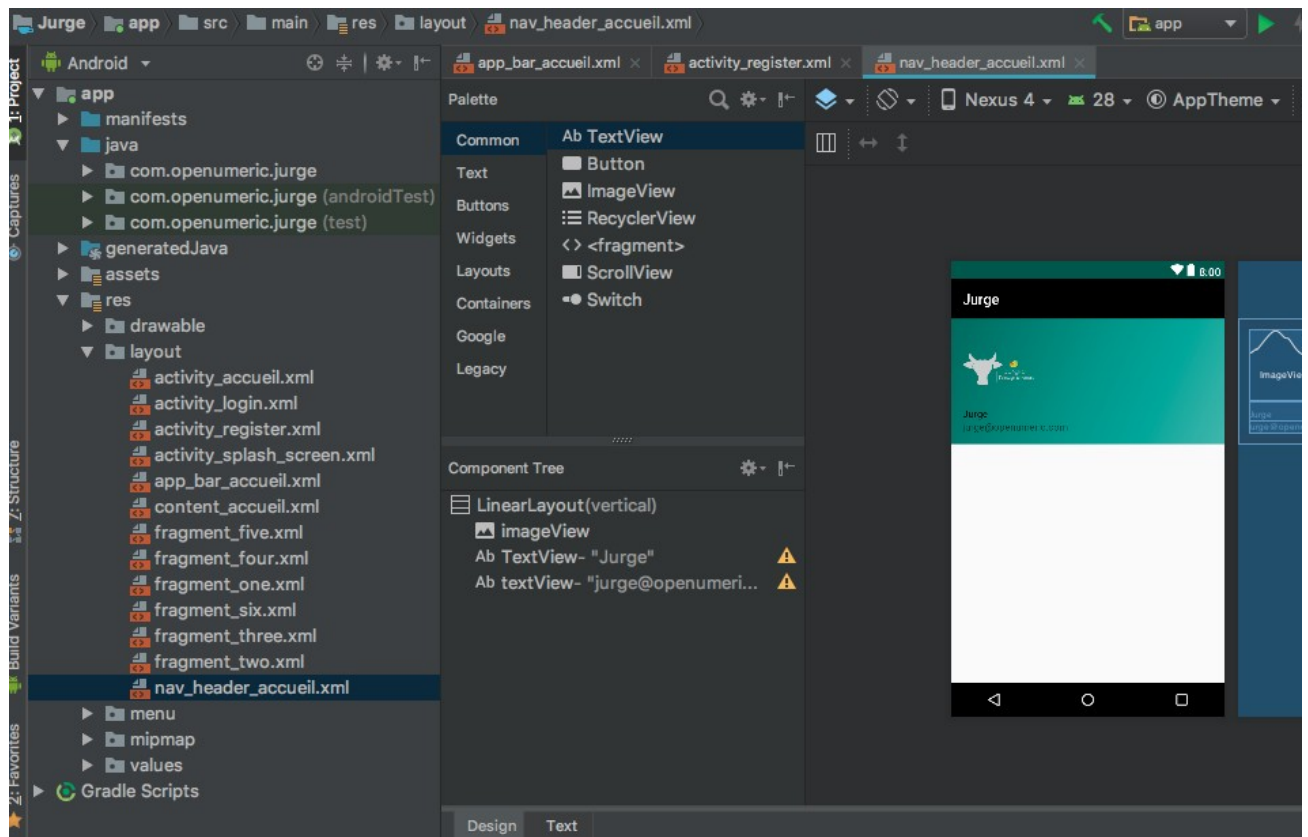
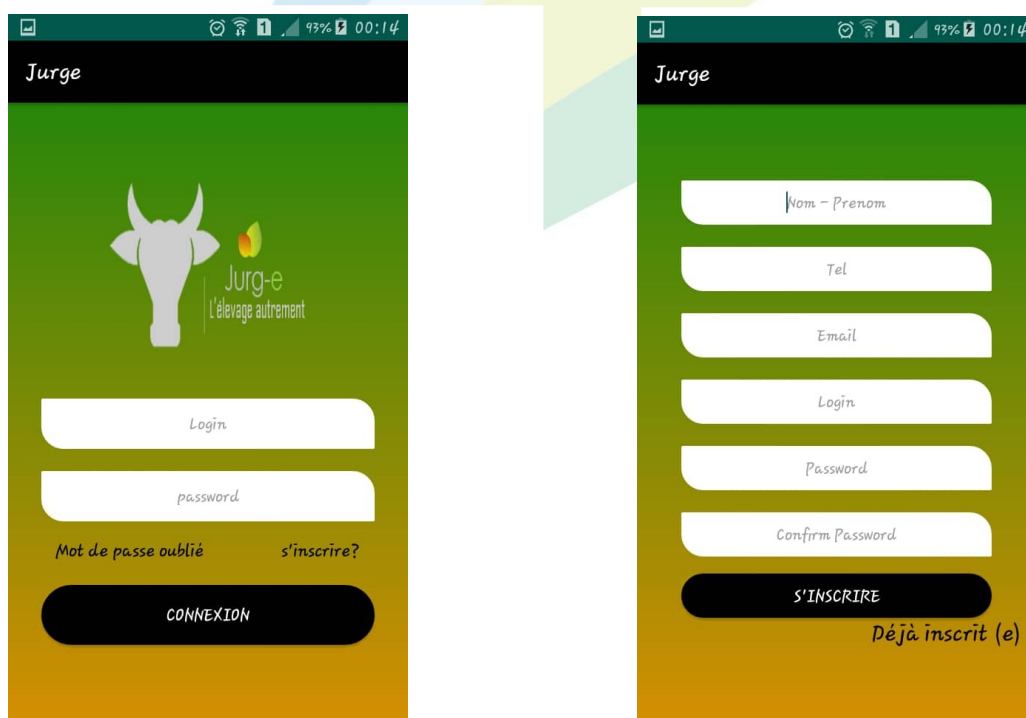


Figure 50: capture de Android studio pendant la conception(2)



Figure 51: page d'accueil

Ensuite on arrive sur la page d'authentification, on s'identifie lorsqu'on a déjà un compte sinon on s'inscrit en appuyant sur le bouton « s'inscrire » qui nous redirige vers la fenêtre



d'inscription.

Figure 52: pages d'authentification

Une fois authentifiée on accède a une interface aux allures de tableau de bord avec des onglets et également un NAV HEADER qui regroupe les fonctionnalités de l'application tel que l'onglet « traitements », « vaccins », « gestion saillie », « mes infos », « statistiques » et « localiser ».

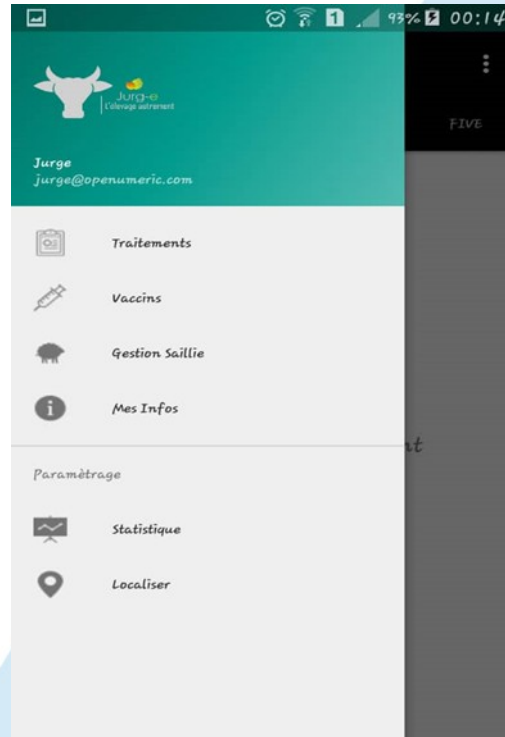


Figure 53: informations

Et on accède à la localisation sur une carte Google Maps intégrée à l'application lorsqu'on appuie sur le bouton «localiser». Ces coordonnées viennent des coordonnées émises par le collier connecté et qui sont stockées directement et en temps réel sur la base de données.

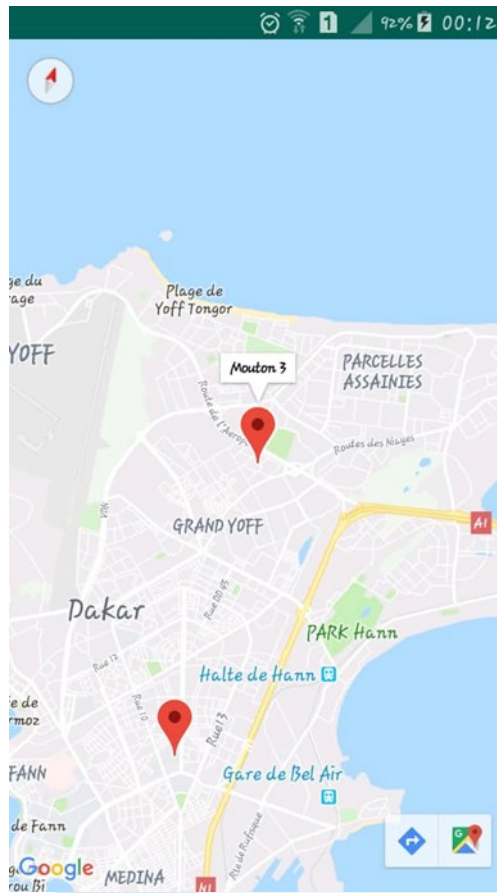


Figure 54: géolocalisation des sujets

3) PERSPECTIVES

3.1.1 Modifier le collier de sorte qu'il soit adéquat aux endroits sans connexion

Nous utiliserons certaines applications GPS hors ligne et déploierons également la technologie LORA dans les zones isolées pour faciliter la communication de nos capteurs mais aussi la géolocalisation des têtes de bétail sans connexion.

Abstraction faite de Google MAPS nouvellement arrivé dans cette sélection, il est clair que HERE MAPS reste peut-être la vraie seule référence par rapport à la foultitude de fonctionnalités disponible, mais des applications comme Locus MAPS ou MAVERICK sont tout à fait pertinentes lorsque vous vous trouvez en pleine nature, là où le réseau peut venir à manquer ou alors que votre forfait data est épuisé. Et ces applications nous serviront de modèles pour faire celui qui nous sera adéquat.

3.1.2 Minimiser la taille du matériel ayant servi à faire le collier

Afin de ne pas attirer le voleur nous serons amenés à revoir la taille du collier tout en sachant qu'il reste néanmoins d'autres fonctionnalités à tenir en compte comme la batterie etc.... qui seront incorporés au fur et à mesure qu'on effectue les mises à jour.

Un **GPS** de petite taille sera toujours meilleur **pour** un sujet, mais le problème avec les petits **colliers GPS**, c'est que la durée de vie de la batterie tend à diminuer rapidement avec la taille. Tant bien qu'il vaut la peine d'opter **pour** un **collier** légèrement plus gros, mais avec un dispositif plus discret afin de réduire le nombre de rechargement de batterie que vous devrez faire.

3.1.3 Ajouter des fonctionnalités à l'application mobile

- Nous définirons une barrière géographique à ne pas franchir, facile à mettre en place depuis notre application.
- Nous rajouterons également d'autres onglets à savoir la possibilité d'instaurer un réseau de partage entre bergerie (ou éleveurs) via un système de chat intégré à l'application.
- Revoir les fonctionnalités concernant les informations sur les sujets. Instaurer à l'avenir une application personnalisable par l'utilisateur selon les fonctionnalités désirées et améliorer la version de sorte que les versions diffèrent selon les fonctionnalités embarquées.
- Un meilleur suivi des statistiques, la vente de données aux autres bergerie ou organismes intéressés tout en respectant les normes juridiques et éthiques.

3.1.4 Caméra IP

Un système de vidéosurveillance IP fait également partie des perspectives

La vidéosurveillance IP comporte de nombreux avantages liés à ses fonctionnalités. Grâce au réseau et via Internet :

- Les images peuvent être visualisées en temps réel et consultées à n'importe quel moment sur n'importe quel support connecté au réseau ;
- les caméras peuvent être gérées à distance : paramétrage, maintenance, zoom...
- les images sont accessibles sur un PDA ou un GSM.

Les caméras IP, très sophistiquées, sont dotées d'un grand nombre de fonctions : système d'alarme, capteurs de mouvement, de chaleur, comptage, détection de silhouette...

Cette technologie permet également de stocker un très grand nombre d'images sans perte de qualité et une meilleure surveillance.

Voici un tableau récapitulatif des différents aspects de ce type de vidéosurveillance :

--	Avantages	Inconvénients
Accessibilité	Bénéficie de toutes les fonctionnalités d'Internet	Peu de choix de caméras (taille, forme...)
Fonctionnalités	Multiples grâce à la technologie réseau : la vidéosurveillance IP peut s'intégrer à d'autres technologies qui relèvent de l'IP (système de sécurité, visioconférence...)	--
Installation	<p>Peut s'installer sur un réseau informatique qui existe déjà (économie)</p> <p>Compatible avec tous types de câblage : IP, coaxial... ou sans fil (WIFI)</p> <p>Pas de gros travaux à prévoir</p> <p>Tout passe par un ordinateur : pas besoin de moniteur, d'enregistreur...</p> <p>Grande flexibilité d'installation des caméras : possibilité d'en ajouter facilement ou de les changer de place</p>	Complexe : nombreux réglages, présence d'un informaticien indispensable
Visualisation des images	Sur les ordinateurs du réseau ou via Internet (PC, PDA, GSM)	Images de moins bonne qualité qu'avec une

	Visualisation des images en temps réel Réglage possible des images (dimension, zoom...) Les caméras peuvent être commandées à distance (selon les modèles)	installation analogique
Maintenance	Pas de contrat de maintenance nécessaire : peut se faire à distance par le service informatique	Informaticien
Sécurité	Verrouillage de l'accès aux images par mot de passe Peut se coupler avec un système de sécurité (alarme...)	--
Coût	Coût des caméras IP moins onéreux que les caméras analogiques	Important s'il faut prévoir l'installation du réseau

4) CONCLUSION

Les conséquences qu'engendre le vol de bétail sont non seulement imprévisibles au plan sécuritaire, mais inacceptables au plan économique ; car toute activité criminelle a un effet multiplicateur qui fait qu'elle se nourrit de son propre élan en se donnant comme exemple de raccourci pour un enrichissement à pas accéléré.

La voie royale pour accroître sensiblement le revenu en monde rural, passe inévitablement par l'élevage ; car à la différence de l'agriculture, la productivité y est très élevée et les risques d'avarie beaucoup moindres.

Ne pas être dans les conditions propres à la pratiquer convenablement ouvre la voie à l'exode rural pour ces masses paysannes, qui risqueront en ville l'altération de leur culture et la perte de leurs valeurs morales et éthiques, face à l'inadéquation de leurs moyens avec les exigences de vie de leur nouveau milieu.

Eradiquer le vol de bétail est une belle opportunité pour faire baisser le chômage, assurer le retour des populations dans le monde rural, et même d'attirer vers ce milieu une nouvelle frange de la population urbaine, tels les retraités et certains hommes d'affaires du milieu agroalimentaire, etc.

La disparition de ce fléau, alliée à une bonne réorganisation des circuits économiques, pourrait sans nul doute avoir un effet éminemment bénéfique sur le taux de croissance de l'économie, la baisse du taux de chômage et même sur la balance commerciale.

La solution est à votre portée, la solution c'est JURG-E !