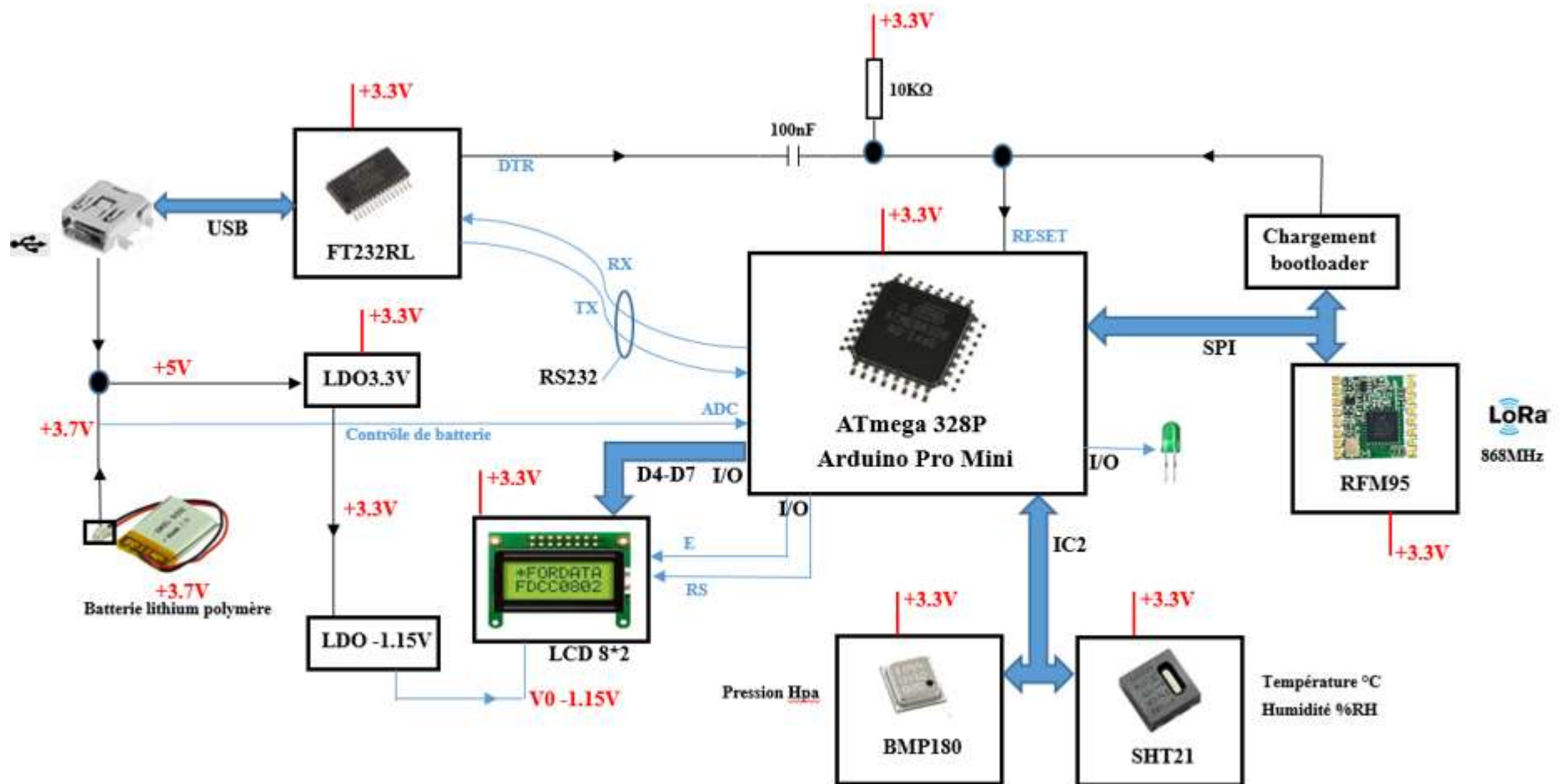


SA2- Prototypage électronique

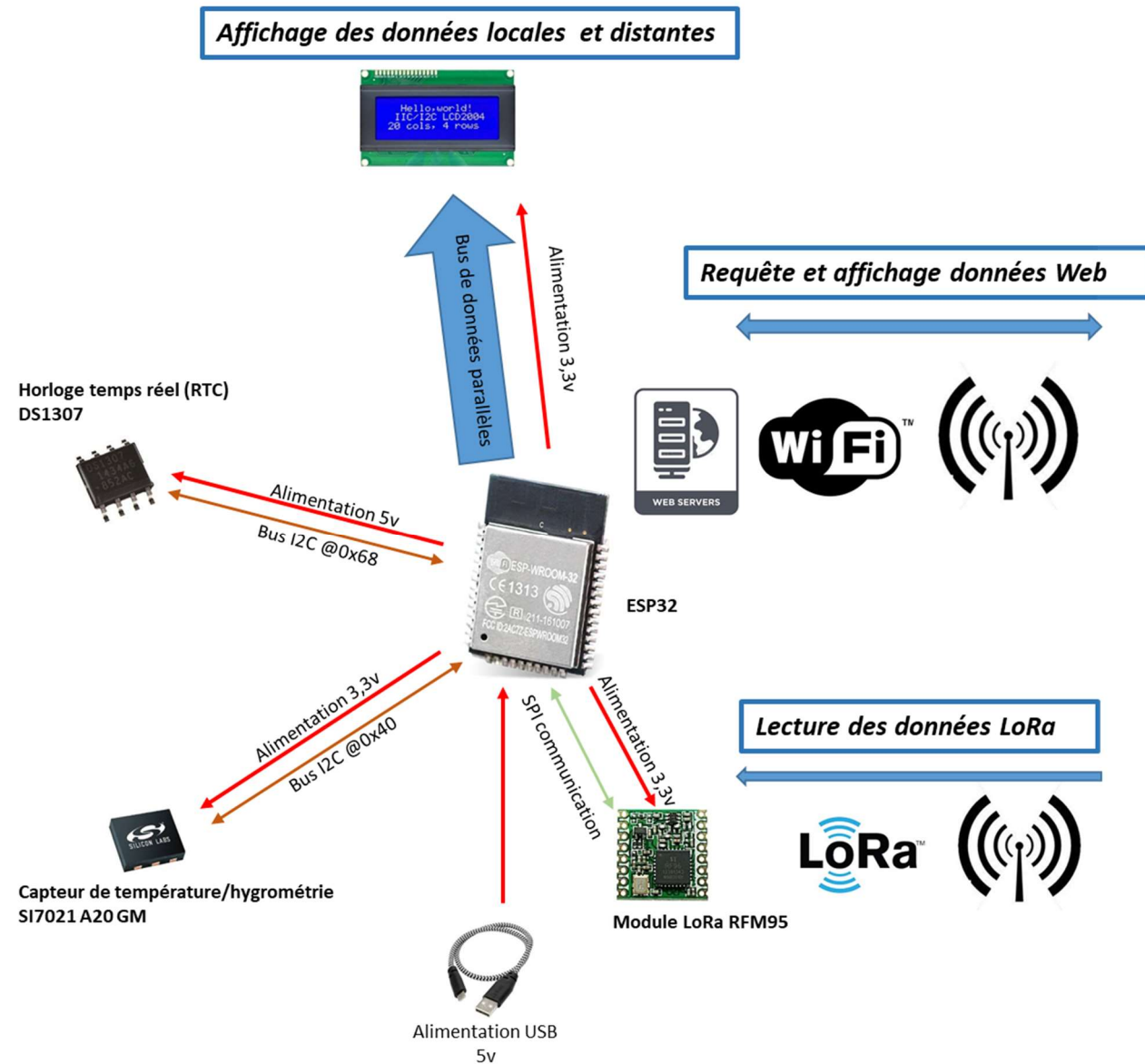


Station météo connectée

Synoptique du module capteurs :



Synoptique du module récepteur



La station météo connectée

Le projet proposé ici est une station météo connectée. Nous allons réaliser un ensemble de cartes électroniques composées des éléments permettant de recevoir des données de température et hygrométrie d'ensembles distants et de mettre ces informations à disposition sur internet. Ces cartes seront composées d'éléments prédéfinies vous permettant d'arriver à la fonction finale.

On retrouvera donc un module capteur composé d'un capteur de température (BMP180), d'un capteur d'hygrométrie (SHT21), d'un afficheur LCD 8x2, d'un module de transmission LoRa (RFM95), d'un port USB de programmation, d'une batterie et d'un cœur de système basé sur un ATmega 328P (tous les composants sont simulables sous Proteus, sauf le module LoRa).

Le second module sera basé sur un kit de développement ESP32 auquel sera connecté un capteur de température et d'hygrométrie (SI7021), une horloge temps réelle (DS1307), un afficheur LCD 20x4 et d'un module LoRa (RFM95) (les deux capteurs sont simulables sous Proteus, il n'y a pas de modèle pour la carte ESP32).

Le travail s'effectuera par binôme. Un compte rendu sera établi et rendu en fin de session sous forme d'un rapport d'étude. Ce rapport comprendra les plans, les schémas, les codes ainsi que les explications permettant de comprendre les choix techniques et le fonctionnement du système.

1. Travail demandé :

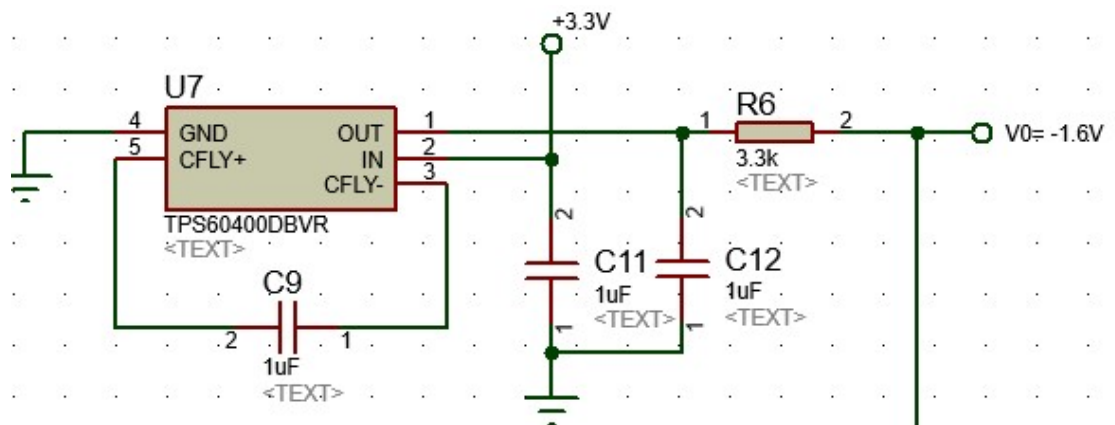
1) Le module capteur

Vous réaliserez sous Proteus un schéma du module capteur. La taille de la carte est imposée et devra avoir la dimension de 40mm x 47mm.

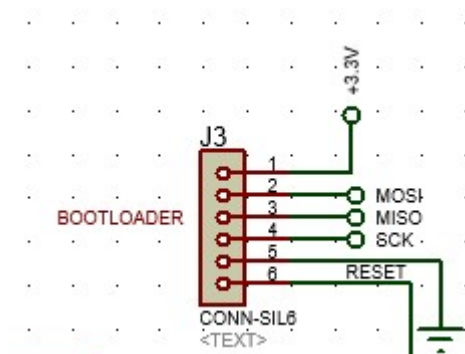
Instructions pour le routage de la carte :

- Créer un compte sur SAMACSYS (pour récupérer des empreintes de composant sous PROTEUS)
- Tous les composants résistances, condensateurs sont en boîtier CMS 1206
- La référence du connecteur micro USB type B est ZX62B5PA33
- Le boîtier du régulateur 3.3V est le SOT223-3
- Pour le capteur de pression BMP180 on utilisera une carte de développement (le composant est trop dur à souder), on utilisera comme empreinte un CONN-SIL4 (respecter le brochage, voir plan de la carte)
- L'empreinte du module LoRa RFM95 est à créer par rapport à la datasheet (créer une pastille CMS 1.2mm/2.2mm)

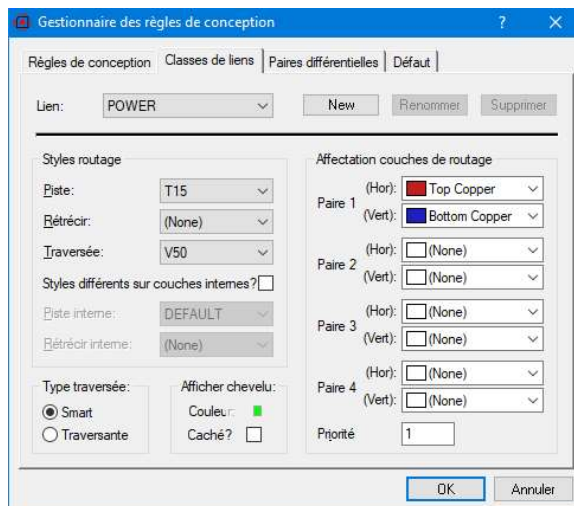
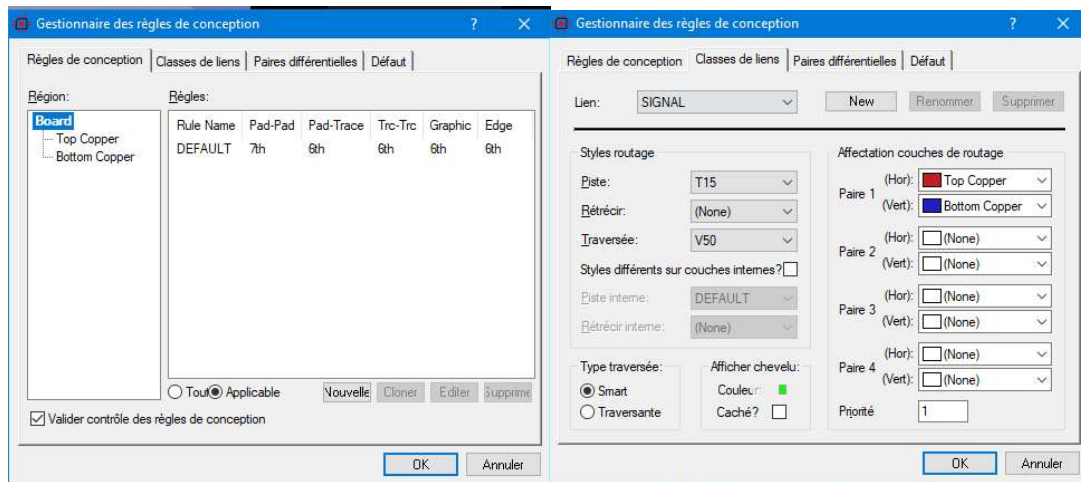
- L'antenne du module LoRa RFM95 est une pastille CONN-SIL1 (pour une antenne sur un fil, dont la longueur sera calculée) et rajouter en plus l'empreinte U.FL-R-SMT-1(10) pour connecter une antenne directement.
- L'entrée RST (reset) du module LoRa RFM95 est à relier sur la sortie PB1 de l'ATmega328P
- Le boîtier de l'ATmega328P est le QFP80P900*900...
- Le boîtier du FT232RL est SOP65P778*200-28N
- L'empreinte de l'afficheur LCD 8*2 est le boîtier CONN-DIL16 (le nombre et le pas correspond, par contre il faut vérifier le brochage par rapport à la datasheet !!)
- La tension négative permettant le réglage du contraste de l'afficheur se fera au travers d'un régulateur inverseur et d'une résistance qui permet d'abaisser la tension présente sur l'afficheur



- Le connecteur pour charger le bootloader est du type CONN-SIL6, le brochage est le suivant



- Le connecteur pour le branchement de la batterie est du type CONN-SIL2
- L'empreinte du quartz 8Mhz est ABLS
- Pour le routage de la carte, le circuit doit être en double face, vous pouvez mettre des composants des deux côtés. La taille des pistes doit être du T15 au minimum, les via en V50. Si vous utilisez des plans de masse Frontière en T15, Relief en T15 et isolement en 10th. Il ne faut pas de via sous le capteur de température
- Règle de routage :



- Le connecteur USB doit être en bord de carte, ainsi que les connecteurs d'antenne du module LoRa, le connecteur du bootloader, le connecteur de la batterie et de l'afficheur

Les fonctions du module capteurs seront :

- lecture, affichage des paramètres T° , H% et Hpa et transmission des paramètres en LoRa
- lecture de la tension batterie. Détecter la « batterie faible » ou « batterie OK » (batterie faible pour 3.6V). Transmission de l'information en LoRa

Les informations de température, humidité et pression seront affichées sur l'afficheur LCD 8x2 puis transmises vers le Lora. Pour cela, vous devrez effectuer la simulation puis le routage de la carte.

Génération du code

A partir de l'IDE ARDUINO, vous allez générer le code de l'ATmega.

Le code permettant la gestion des capteurs de pression et de température/humidité n'utilisera pas les bibliothèques disponibles.

Pour le LoRa et l'afficheur LCD la gestion pourra se faire grâce aux bibliothèques ARDUINO disponible :

(LoRa.h) disponible sous l'appellation « arduino-LoRa library by sandeep mistry ».

L'ATmega se programme directement sous l'IDE ARDUINO (via une interface FTDI USB/RS232), auparavant il faut charger un bootloader sur le circuit.

Pour cela, vous devez prévoir sur votre carte un connecteur du type CONSIL6 afin de permettre le chargement du bootloader via le port SPI.

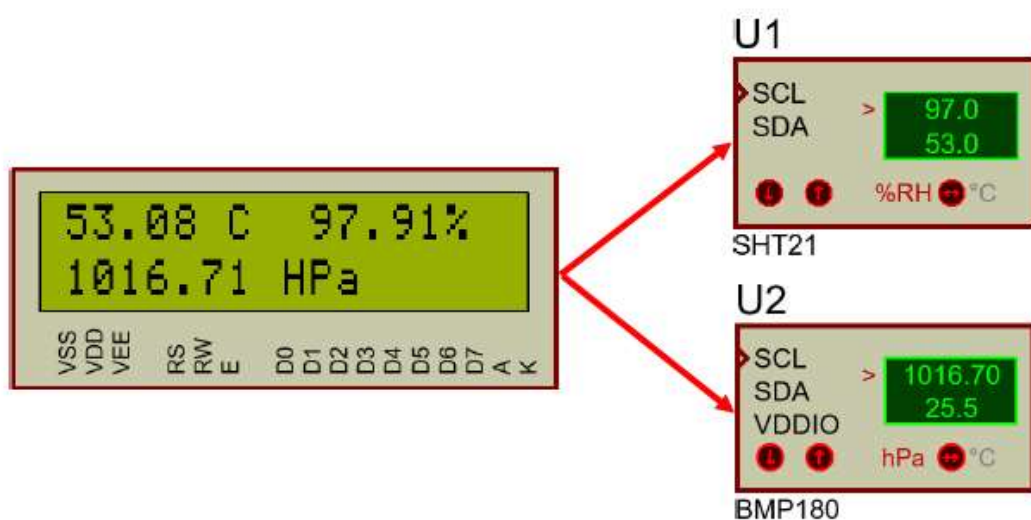
Routage et simulation

Avant cela, il vous faudra simuler le code généré sur Proteus. Cela vous permettra de consolider votre architecture matérielle et logicielle à l'aide des éléments disponibles sous Proteus.

Le code sera simulé par un composant ATmega dont le code exécutable aura été téléchargé sous Proteus.

Une fois votre architecture matérielle terminée, vous fournirez un routage afin que celui-ci soit amené en fabrication.

Tous les composants sont simulables sous Proteus à l'exception du composant LoRa. Celui-ci ne pourra être testé qu'une fois le module émission et réception terminé.



Réalisation et montage

Dès le retour de votre carte de chez le fabricant, vous aurez à réaliser le montage du module, à tester et implanter votre programme.

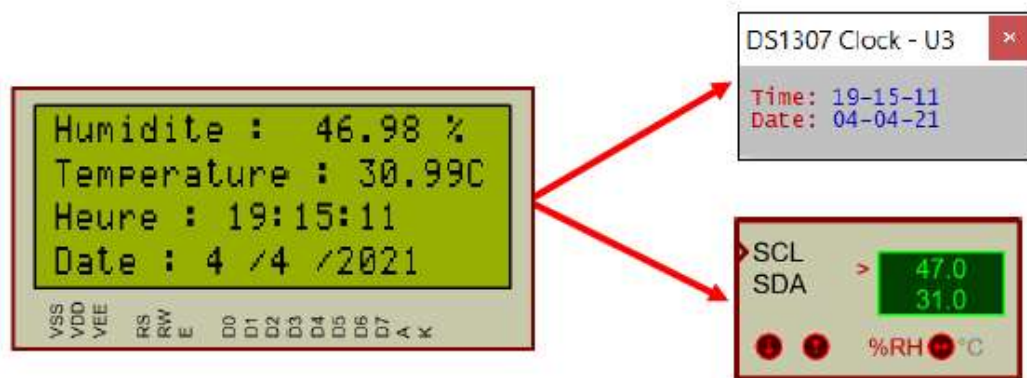
2) Le module récepteur

La carte permettant le montage du module récepteur à base d'ESP32 DEVKIT vous sera fournie.

Les informations reçues par le LoRa ainsi que l'heure, la température et l'hygrométrie locale seront visualisées sur l'afficheur LCD 20x4. Ces informations seront mises à disposition sur un serveur Web embarqué dans l'ESP32, que l'on pourra consulter via une requête HTML à partir d'une connexion wifi extérieure (téléphone portable par exemple).

Génération du code

A partir de l'IDE ARDUINO, vous allez pouvoir générer le code pour l'ESP32. A l'identique du module d'émission, la gestion du capteur de température/hygrométrie, de l'horloge temps réelle se fera sans l'utilisation des bibliothèques disponibles sous l'IDE ARDUINO. Vous avez la possibilité de simuler votre code sous Proteus, en utilisant l'ATmega 328P comme modèle (votre code sera compatible pour l'ESP32).



La gestion du wifi, du NTP, LoRa, de l'afficheur LCD pourra se faire à partir des bibliothèques.

Réalisation et montage

La carte réception (Serveur) n'est pas à router, elle vous est fournie. Vous devez souder les composants. Vous avez la possibilité de charger un programme de test sur l'ESP32, pour vérifier le bon fonctionnement de la plateforme (le fichier est fourni sur moodle).

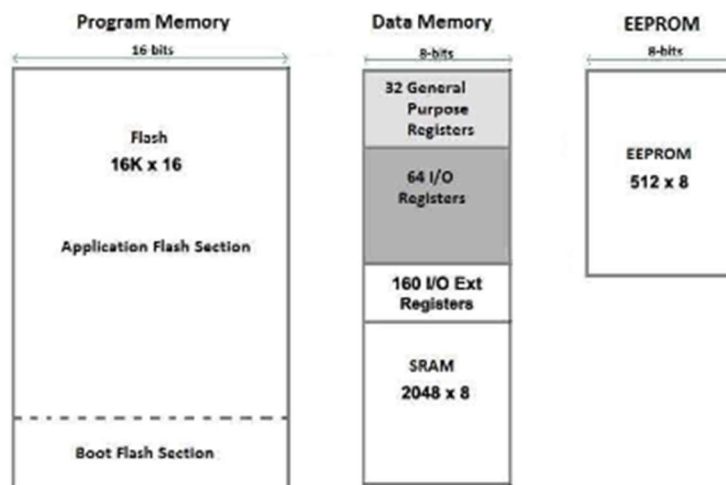
Arduino / ATmega : programmation ISP

<https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoISP>

Pourquoi faire ?

Afin de pouvoir télécharger (uploader) vos programmes, directement dans la mémoire flash de votre processeur, à partir de votre ordinateur et de l'IDE ARDUINO, il faut pouvoir connecter votre carte par le port USB et donc installer un bootloader.

Ce bout de code est situé à la fin de la zone d'adresse mémoire programme.



Lors de chaque reset, le microcontrôleur exécutera un code spécial de prise en compte de la connexion.

Boot loader

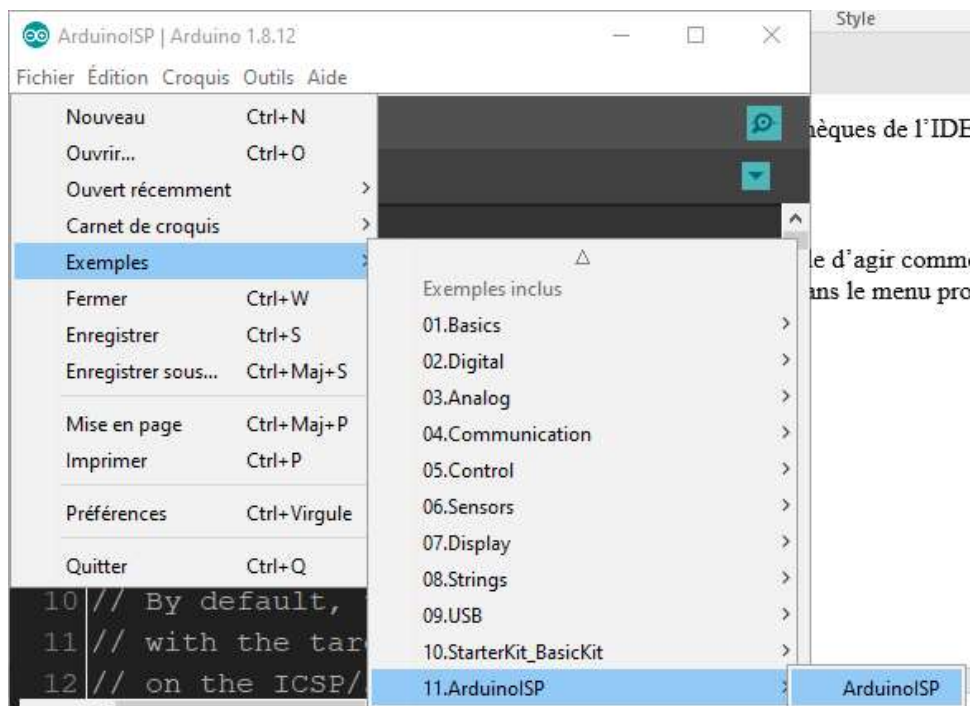
Le bootloader est un genre de BIOS pour microcontrôleur. Il gère les paramètres bas-niveau de la puce, comme sa fréquence, son voltage, etc. Chaque microcontrôleur a sa propre version du bootloader. Le bootloader d'un ATmega8 ne sera pas compatible avec un ATmega328, par exemple.

Quand vous voudrez programmer une puce, commencez par vous assurer d'avoir le bootloader requis, et l'ajuster selon vos besoins.

Le bootloader que nous allons utiliser fait partie des bibliothèques de l'IDE Arduino.

Le sketch Arduino

Notre Arduino a besoin d'un sketch spécial pour être capable d'agir comme un programmeur. Le sketch se trouve dans Fichier -- Exemples -- 11.ArduinoISP— ArduinoISP.



Nous chargerons le bootloader sur notre carte cible (il faut souder évidemment l'ATmega328P et le quartz et les deux condensateurs) à l'aide d'une carte UNO, il faut programmer notre carte UNO.

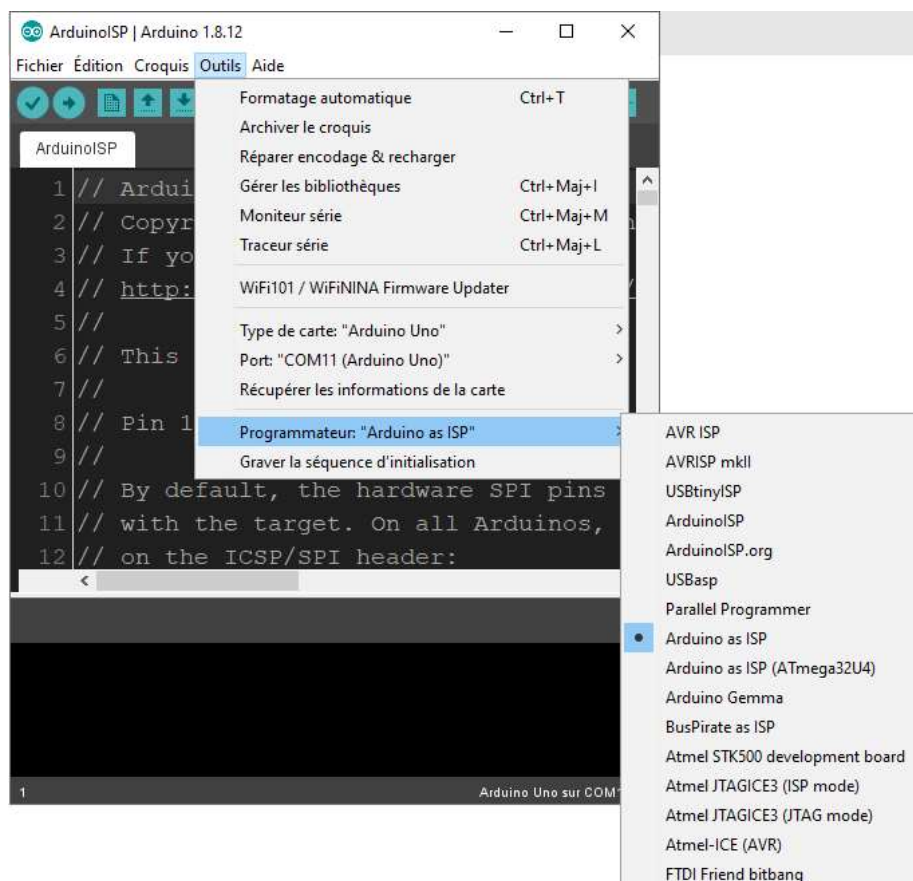


Diagram illustrating the connection of an ATMEGA328P microcontroller to an Arduino Uno R3 in ISP mode.

The ATMEGA328P (U2) is connected to the Arduino Uno R3 (ARDUINO UNO EN MODE ISP) as follows:

- VCC (Pin 20) is connected to the VCC header (Pin 6).
- GND (Pin 18) is connected to the GND header (Pin 5).
- MOSI (Pin 16) is connected to the MOSI/D11 header (Pin 15).
- MISO (Pin 17) is connected to the MISO/D12 header (Pin 16).
- SCK (Pin 18) is connected to the SCK/D13 header (Pin 17).
- RESET (Pin 29) is connected to the RESET/D10 header (Pin 1).

The ATMEGA328P is labeled U2 and ATMEGA328P (QFP80P) cible.

The screenshot shows the Arduino IDE interface. The 'Outils' (Tools) menu is open, displaying various options for configuring the development environment. The 'Type de carte: "Arduino Pro or Pro Mini"' option is highlighted. To the right, the 'Gestionnaire de carte' (Board Manager) window is open, showing a list of available boards. The 'Arduino Pro or Pro Mini' board is selected, indicated by a blue dot next to its name.

ArduinoISP | Arduino 1.8.12

Fichier Édition Croquis **Outils** Aide

Formatage automatique Ctrl+T
 Archiver le croquis
 Réparer encodage & recharger
 Gérer les bibliothèques Ctrl+Maj+I
 Moniteur série Ctrl+Maj+M
 Traceur série Ctrl+Maj+L
 WiFi101 / Wi-FiNINA Firmware Updater
 Type de carte: "Arduino Pro or Pro Mini"
 Processeur: "ATmega328P (3.3V, 8 MHz)"
 Port: "COM11 (Arduino Uno)"
 Récupérer les informations de la carte
 Programmeur: "Arduino as ISP"
 Graver la séquence d'initialisation

1 // Ardui
 2 // Copyr
 3 // If yo
 4 // http:
 5 //
 6 // This
 7 //
 8 // Pin 1
 9 //
 10 // By de
 11 // with the target. On all Arduinos, t
 12 // on the ICSP/SPI header:

1

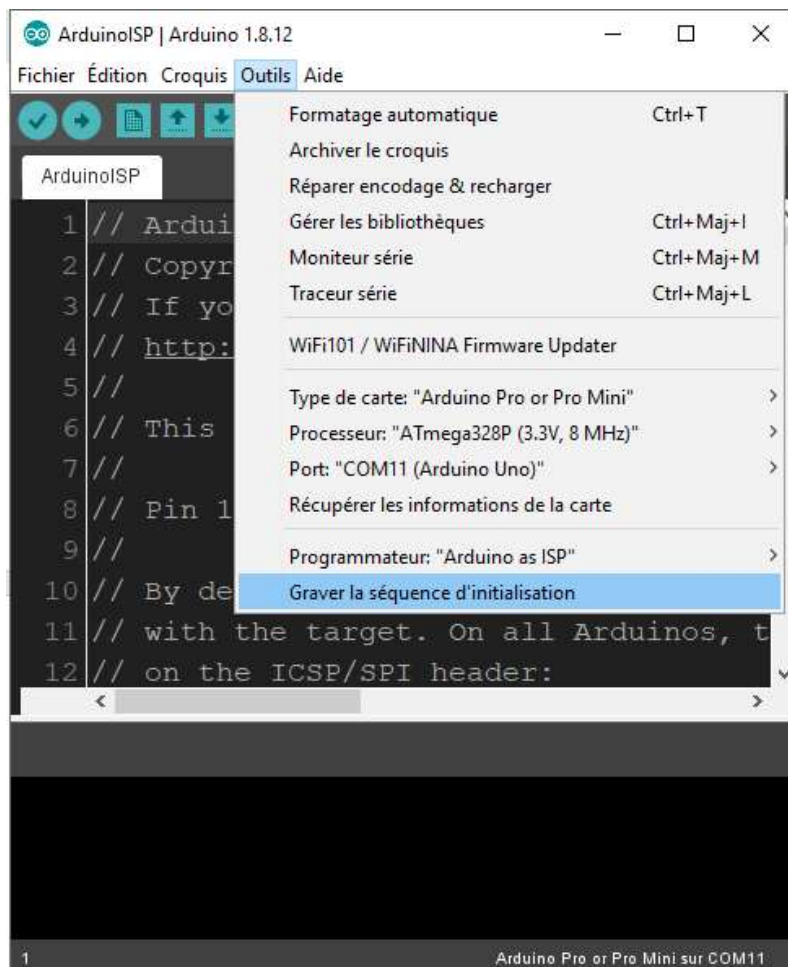
Arduino Pro or Pro Mini sur COM11

Gestionnaire de carte

Cartes Arduino AVR

Arduino Yún
 Arduino Uno
 Arduino Duemilanove or Diecimila
 Arduino Nano
 Arduino Mega or Mega 2560
 Arduino Mega ADK
 Arduino Leonardo
 Arduino Leonardo ETH
 Arduino Micro
 Arduino Esplora
 Arduino Mini
 Arduino Ethernet
 Arduino Fio
 Arduino BT
 LilyPad Arduino USB
 LilyPad Arduino
 • Arduino Pro or Pro Mini
 Arduino NG or older

Il ne reste plus qu'à charger le bootloader, dans le menu Outils, sélectionner Graver la séquence d'initialisation.



Après quelques secondes d'activité sur l'Arduino, l'IDE devrait vous indiquer :

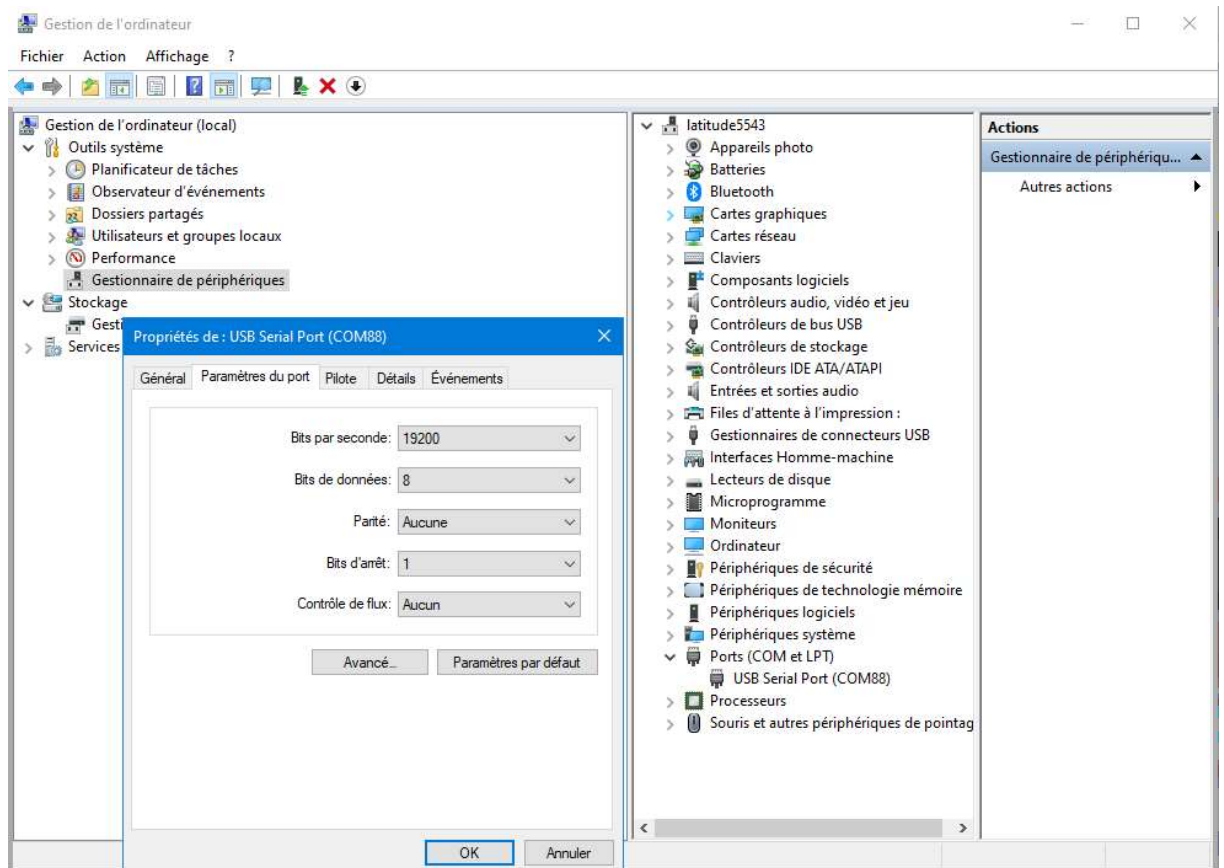
```
avrdude: verifying ...  
avrdude: 1 bytes of lock verified  
avrdude: Send: Q [51] [20]  
avrdude: Recv: . [14]  
avrdude: Recv: . [10]  
  
avrdude done. Thank you.
```

C'est terminé pour le bootloader.

Maintenant, vous pouvez uploader votre sketch de la manière habituelle, en utilisant le bouton « Téléverser » avec un programme présent dans vos Fichiers.

N'oubliez pas de configurer correctement l'IDE ARDUINO pour la carte Arduino Pro or Pro Mini, en *ATmega328P (3.3v, 8Mhz)*.

Sous Windows, pensez à vérifier la vitesse du port série et la paramétrer à 19200 bauds.



I. Travail à réaliser

<u>Carte capteurs</u>	
<p>Réaliser le circuit électronique de la carte module capteurs sous PROTEUS.</p> <p>Fonction : Mesurer la tension de batterie, mesurer la température, l'hygrométrie, la pression et afficher les informations.</p>	
Réaliser le routage de la carte module capteurs sous PROTEUS (voir les instructions ce dessus pour le routage)	
Réaliser le programme du module capteurs et la simulation sous PROTEUS. Le programme de gestion des capteurs I2C (BMP180, SHT21) doit être réalisé sans bibliothèque intégrée sous ARDUINO (c'est à vous de réaliser votre propre bibliothèque). Les autres composants (module LoRa , écran LCD) peuvent intégrer des bibliothèques ARDUINO.	
Tester la carte module capteurs via le terminal ARDUINO IDE	

<u>Carte récepteur</u>	
Le schéma du module récepteur, serveur est donné sous PROTEUS. Le programme de gestion des capteurs I2C (SI7021, DS1307) doit être réalisé sans bibliothèque intégrée sous ARDUINO (c'est à vous de réaliser votre propre bibliothèque). Les autres composants (module LoRa , écran LCD) peuvent intégrer des bibliothèques ARDUINO. Pour réaliser les programmes, vous pouvez les simuler sous Proteus avec un ATmega 328 sous Arduino(votre code sera compatible pour l'ESP32).	
La carte module récepteur, serveur est déjà réaliser, vous devez souder les différents composants puis tester votre carte électriquement et implanter vos programmes de gestion des capteurs.	
Réaliser la connexion WIFI et se synchroniser sur l'heure NTP. Synchroniser le DS1307 avec l'heure NTP.	
Récupération des données LoRa	
Afficher les données récepteurs (T°, H%, heure et date) et capteurs (T°, H%, Hpa et état batterie) sur l'afficheur	
Mettre à place les données récepteurs (T°, H%, heure et date) et capteurs (T°, H%, Hpa et état batterie) sur le serveur Web	
Se connecter sur le serveur Web via le wifi et afficher les données récepteurs (T°, H%, heure et date) et capteurs (T°, H%, Hpa et état batterie) sur votre client Web.	
<u>Rapport</u>	
Fournir du schéma du module capteurs	
Fournir le routage du module capteurs	
Fournir les codes commentés des deux modules	
Fournir les explications des choix techniques, une synthèse du projet	