# Cahier des charges

La serre connectée doit recevoir des informations sur un microcontrôleur MSP430G2553 par l’intermédiaire d’un module bluetooth RN-42 interfacé en UART. Ces informations doivent être interprétées et transmises à un deuxième microcontrôleur MSP430G2553 via un bus SPI. Ce deuxième microcontrôleur doit, selon la commande reçue

* Lire la valeur d’un capteur UV sur un de ses ports analogiques
* Lire la valeur d’humidité et de température d’un capteur via un interfaçage I2C
* Commander un servomoteur, variant entre 3 positions

Le programme doit être réalisé en C, et devra respecter certaines normes de codage, dérivées du MISRA 2012.

# Etude préalable

À la suite d’un problème au niveau des microcontrôleurs reçus lors de la distribution du matériel, nous avons dû faire le projet avec deux MSP430G2553.

## Schéma

Ici, tout est alimenté en 3.3V via l’USB du PC. Toutes les masses sont communes.

## Découpage fonctionnel

Master :

* UART : Fichier gérant la configuration de l’UART et son fonctionnement
  + **void init\_UART( void )** : initialisation de l’UART
  + **void send\_UART(unsigned char \*msg)** : envoie une chaîne de caractères via l’UART
* Interpreteur : Interpreteur de commande : transfert via SPI ou répond via UART selon la commande reçue via UART
  + **void interpreteur( void )**: renvoie un message d’acknowledge via UART et transmet la commande au slave via SPI si besoin de transmettre.
* SPI\_Master : Fichier gérant la configuration du SPI et son fonctionnement
  + **void init\_SPI( void )**: Initialisation du SPI en master
  + **void Send\_SPI(unsigned char carac)**: envoie d’un seul caractère au slave via SPI
* Init\_LP : Fichier d’initialisation de la launchpad
  + **void init\_LP( void )** : Initlialisation de la launchpad
* 2553\_MASTER : fichier contenant la fonction main du MASTER
  + **void main( void )**: Comportement du projet

Slave :

* ADC : Bibliothèque gérant l’ADC (fournie par l’ESIGELEC donc non-détaillée ci-dessous)
* UV\_sensor : Fichier gérant l’initialisation et la récupération des données du capteurs UV
  + **void init\_UV( void )** : Paramétrage de l’ADC sur le pin du capteur UV
  + **unsigned int Read\_UV(void)**: Lecture de la valeur de l’ADC et conversion en indice UV pour la valeur retournée
* Init\_LP :
  + **void init\_LP\_Slave( void ) :** Initialisation de la launchpad
* Servomoteur**:** Initialisation et gestion de la PWM pour le servomoteur
  + **void init\_pwm( void )** : Initialisation de la PWM pour le servomoteur
  + **void set\_servo (unsigned char pos)**: Modification de la PWM en fonction de la position demandée
* SPI\_Slave : Fichier gérant la configuration du SPI
  + **void init\_SPI\_Slave ( void )**: Initialisation du SPI en slave
* 2553\_SLAVE : fichier contenant la fonction main du MASTER
  + **void main( void )**: Comportement du projet
* I2C : Fichier gérant la configuration et la communication I2C
  + **Void init\_I2C (void)**: initialisation et configuration de l’I2C