

Systèmes interconnectés – SINT

Projet SmartGreenHouse — supervision d'une serre

Centrale Nantes

P.-E. Hladik, pehladik@ec-nantes.fr

Version bêta (9 mai 2022)

“ Selon différentes expérimentations menées aux quatre coins du monde, l’Internet des objets pourrait nous aider à relever les deux grands défis d’une agriculture plus intelligente : augmenter les rendements tout en intégrant harmonieusement la production agricole dans les systèmes naturels. Pour nourrir les 9 milliards d’humains que comptera la planète à l’horizon 2050, la FAO (Organisation des Nations unies pour l’alimentation et l’agriculture) estime que la production vivrière mondiale devra augmenter de 60%. Sachant que l’agriculture est déjà responsable de 20% des émissions de gaz à effet de serre, comment nourrira-t-on une population plus nombreuse sans compromettre la biodiversité ? [<https://hellofuture.orange.com>] ”

1 Organisation du travail

Le travail se fait par groupe de 6 (ou 5). La séquence pédagogique dure 14h avec 2h de TD, 11h de TP et 1h de démonstration. L’objectif est de réaliser un dispositif de surveillance et de contrôle environnemental d’une serre. Pour cela vous allez :

- mettre en place les drivers pour échanger les données entre les composants,
- définir les messages et les mécanismes permettant leur échange.

Chaque groupe aura à sa disposition l’ensemble des composants du système. Vous devrez vous auto-organiser en vous répartissant le travail afin d’atteindre les objectifs attendus. Pendant le premier TD vous commencerez par une phase d’analyse et de conception du système en identifiant clairement les fonctionnalités attendues et la messagerie à mettre en place. Lors des séances suivantes, il est conseillé de vous répartir le travail par sous-système en prenant en compte la charge de travail estimée (cette répartition pourra être revue à chaque début de séance). Une estimation au doigt mouillé de la charge en fonction des sous-systèmes est fournie dans le tableau 1.

Une archive avec les documentations techniques est disponible sur Hippocampus.

La dernière heure sera consacrée à des démonstrations. Chaque groupe devra présenter en 10 min les résultats obtenus devant les encadrants et l’ensemble de la promo. Aucun compte-rendu ne sera demandé, mais des explications précises sont attendues lors de la démonstration.

TAB. 1 – Estimation de la charge de travail en fonction des sous-systèmes

Sous-système	Charge estimée
Feather M0 WiFi Superviseur	Faible
stm32-1 Thermique	Moyenne
stm32-2 THP	Moyenne
stm32-2 CO2	Forte
stm32-3 Humidificateur	Très Faible
Basys3 Leds	Très forte
Superviseur PC	non obligatoire

La section suivante du document est consacrée à la vue d'ensemble du système et des sous-systèmes. La section 3 présente les cas d'utilisation du système. La section 4 présente plus en détail les composants du système (cartes, capteurs et actionneurs). La section 5 reprend chacun des sous-systèmes et précise les attendus pour chacun d'eux. La section 6 précise les attendus de la séance de TD et les modalités de la démonstration.

2 Vue globale du système de supervision d'une serre

Le système a pour objectif de superviser et contrôler les conditions environnementales d'une serre. Il est constitué de sous-systèmes interconnectés via un bus CAN (voir figure 1). Chaque sous-système est lui-même composé d'une unité de contrôle (micro-contrôleur ou FPGA) et d'un ou plusieurs circuits de type capteur ou actionneur. Un sous-système est dédié à la communication vers un système extérieur de type PC.

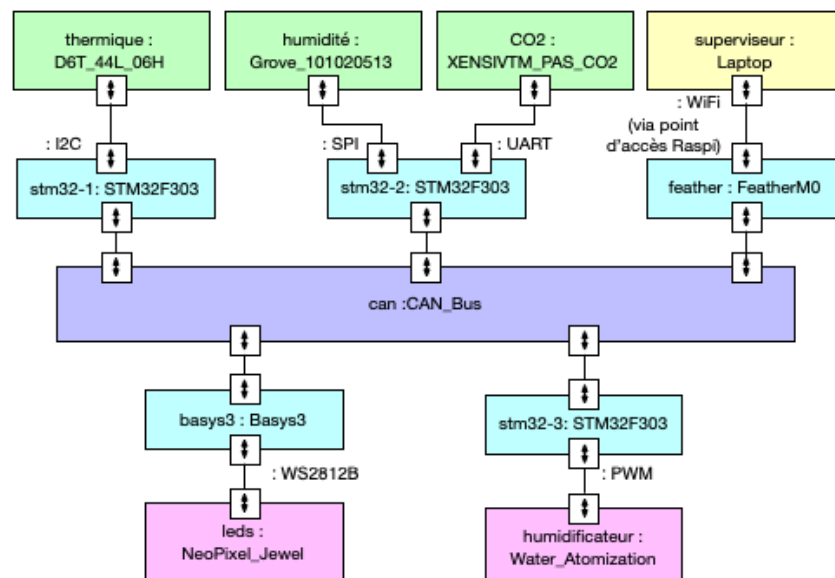


FIG. 1 – Diagramme d'interconnexion des éléments du système

3 Cas d'utilisation

Les différents cas d'utilisation identifiés sont décrits sur la figure 2 :

- le superviseur doit pouvoir suivre l'état de l'environnement de la serre : données thermiques, température, humidité, pression et CO₂. La dynamique du système étant lente une périodicité de l'ordre de la seconde est largement suffisante,
- le superviseur doit pouvoir piloter les actionneurs : leds et humidificateurs,
- le superviseur doit pouvoir activer un mode de contrôle automatique de contrôle de l'environnement de la serre.

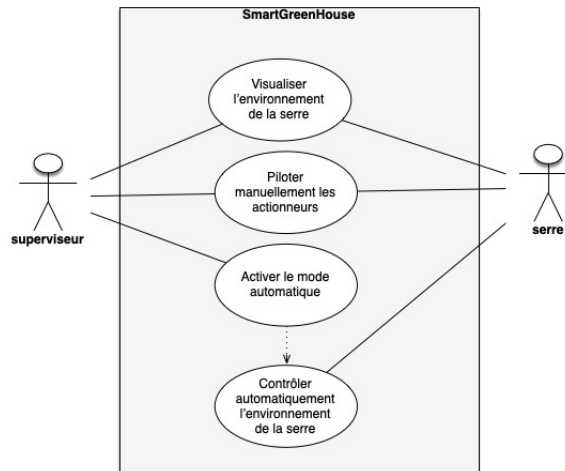


FIG. 2 – Cas d'utilisation du système

Le but de cette activité pédagogique n'étant pas de devenir un spécialiste de la culture sous serre, nous ne nous attarderons pas sur la partie contrôle. Pour la réaliser, il faudra concevoir quelques règles (sous forme de conditions avec des seuils) avec les données des capteurs et déclenchant l'activation des actionneurs.

4 Détails des unités de commande électronique

La documentation technique est fournie dans une archive déposée sur hippocampus.

4.1 Calculateurs utilisés

4.1.1 STM32F303 K8

Les ECU stm32-1, stm32-2 et stm32-3 sont basés sur les cartes de TP violettes comportant un STM32F303K8 dont les broches de sorties sont décrites sur la figure 3. La carte comprend un transceiver CAN. Le schéma électrique de la carte est fourni dans les documents de l'archive.

Attention les interrupteurs du bornier doivent être mis à off pour être certain qu'il n'y aura pas de problème de connectique.

Les connexions avec les capteurs et actionneurs sont spécifiques pour chaque sous-système. Vous utiliserez des breadboard pour faire les montages.

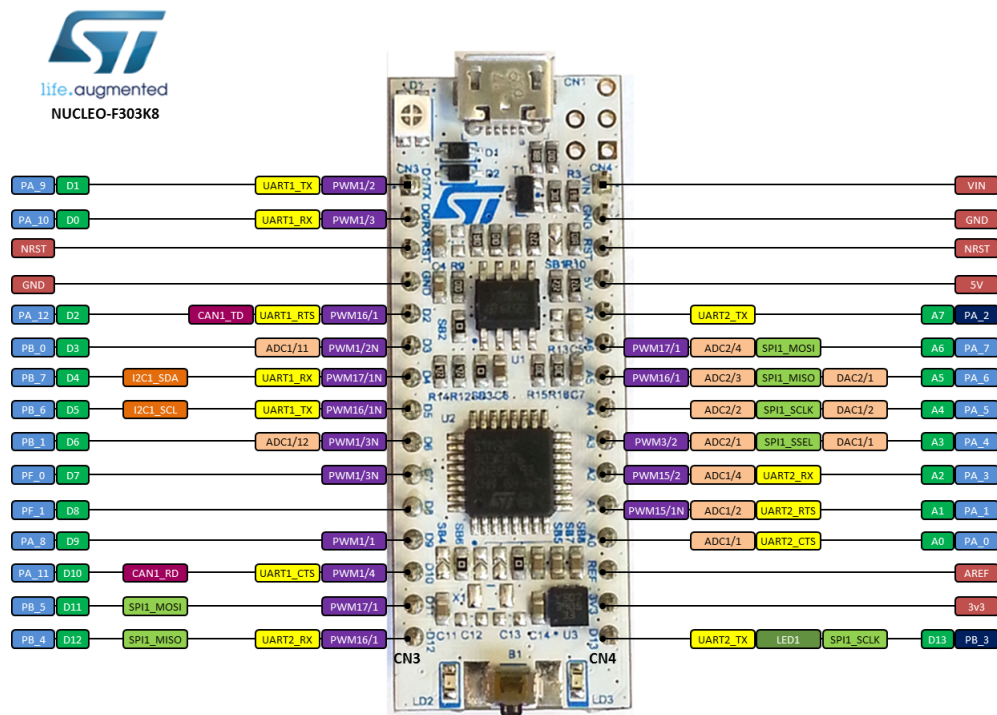


FIG. 3 – Broches du STM32F303K8

Vous pouvez utiliser tous les outils que vous voulez pour programmer sur les cartes STM32. Par défaut nous recommandons l'utilisation de mbed (version [Keil online](#)) qui offre un ensemble de bibliothèques assez proche de ce que l'on peut trouver sur Arduino.

4.1.2 Feather M0 WiFi

Aucune nouveauté pour cette carte par rapport à ce que vous avez en TP. À vous de faire la passerelle entre le CAN et le superviseur en utilisant MQTT.

4.1.3 Basys 3

Aucune nouveauté pour cette carte. Référez-vous au sujet de TP sur le FPGA pour l'utilisation de la bibliothèque CAN. Vous utiliserez les broches de PMOD pour les entrées et sorties (voir section suivante pour les connexions). Le travail sur le Basys 3 est clairement celui qui demandera le plus de temps, si vous avez l'intention de le réaliser, prévoyez une sous-équipe (monôme ou binôme) sur le sujet dès le début.

4.2 Description des sous-systèmes

4.2.1 stm32-1 : capteur thermique

Le sous-système stm32-1 est constitué d'un stm32 et d'un capteur thermique (voir figure 1). L'interconnexion entre les deux composants est réalisée via un bus I2C.

Le capteur thermique est un capteur D6T-44L-06H de OMRON. Il revoit une matrice 4x4 dont chaque point représente la température qu'un objet irradie. La documentation complète du composant est fournie dans **D6T01_ThermalIRSensorWhitepaper.pdf**.

La connexion électrique ainsi que la structure de la trame I2C sont fournies dans le document de référence. Attention il va falloir mettre en place un pull-up sur les lignes SDA et SCL (voir schéma 4).

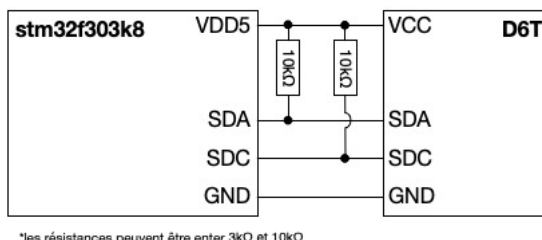


FIG. 4 – Schéma de connexion entre le stm32 et le D6T

4.2.2 stm32-2 : CO2 et THP

Le sous-système stm32-2 (voir figure 1) est constitué d'un stm32, d'un capteur de CO2 et d'un capteur de température/humidité/pression (THP). L'interconnexion entre le stm32 et le capteur de CO2 se fera en liaison série, et celle avec le capteur THP en SPI.

Le capteur de CO2 est un XENSIVTM PAS CO2 de Infineon. Vous trouverez une documentation détaillée dans **Infineon-PAS_CO2_General_Design-In_Guideline.docx.-ApplicationNotes-v01_01-EN.pdf**.

La figure 5 montre le montage à réaliser pour connecter le capteur de CO2. Il nécessite une alimentation de 12V. Un convertisseur boost (Step-Up) vous est fourni afin de monter la tension de 3.3V à 12V. La carte est une **Pololu 12V Step-Up Voltage Regulator U3V12F12**. Attention pour éviter les pics de tension, il faut ajouter une capacité de 33μF entre le VIN et le GND.

Idem au niveau de la broche VDD12 du capteur avec l'ajout de capacités de 47μF, 100nF et 10nF (voir fig 6 de Infineon-PAS_CO2_General_Design-In_Guideline.docx.-ApplicationNotes-v01_01-EN.pdf).

Vous utiliserez l'UART pour communiquer avec ce composant. Vous trouverez une description détaillée de la communication dans **Infineon-programming_guide_PAS_CO2_evaluationkit-ApplicationNotes-v02_00-EN.pdf**.

Le capteur de température, humidité et pression (THP) est une carte Grove 101020513 avec lequel vous communiquerez en SPI (attention le pont permettant de communiquer en I2C a été dessoudé). Ce capteur est basé sur un BME680 de Bosch dont vous trouverez la documentation dans l'archive annexe.

Le composant étant une carte Grove la connexion avec un micro-contrôleur ne nécessite aucun montage supplémentaire. La communication en SPI est décrite dans la datasheet du BME680.

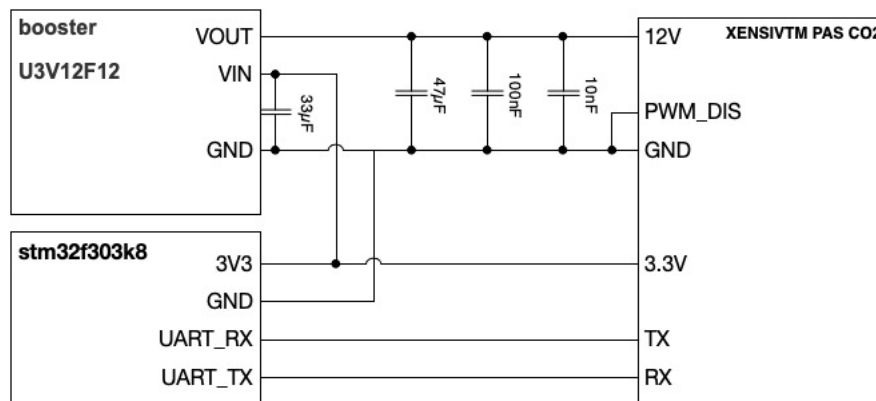


FIG. 5 – Schéma de connexion entre le stm32 et le XENSIVTM PAS CO2

4.2.3 stm32-3 : Humidificateur

Le sous-système stm32-3 (voir figure 1) est constitué d'un stm32 et d'un humidificateur. L'humidificateur est une carte Grove Water Atomization V1.0¹ et se pilote en tout ou rien.

Attention à ne pas mettre ses doigts n'importe où, il peut y avoir des appels de courant important...

La carte étant préconditionnée aucun montage supplémentaire n'est nécessaire. Pour les tests vous pourrez utiliser un gobelet avec de l'eau, mais attention à ne pas en mettre partout !!!

4.2.4 FeatherM0-3 : WiFi

Le sous-système featherM0 est constitué d'une carte FeatherM0 WiFi et d'un ordinateur distant (voir figure 1). L'ordinateur distant peut être un PC de la salle de TP ou bien une de vos machines. La raspberry avec le serveur MQTT sera à disposition pour mettre en place un réseau sans fil.

Le développement de la partie sur l'ordinateur de supervision est optionnelle. Vous pouvez utiliser un client générique comme MQTTX pour afficher les données reçues. Si vous avez envie de développer une interface plus sympathique² vous pouvez mais ne le faite pas au détriment du reste.

4.2.5 Basys3 : Leds

Le sous-système basys3 est composé d'une carte Basys 3, d'un transceiver MCP2562 et d'un afficheur de leds. Ce système simule le contrôle de l'éclairage de la serre. L'afficheur de leds est un NeoPixel Jewel constitué de 7 LED RGB adressables via le protocole WS2812. La documentation concernant le protocole WS2812 est fournie dans **WS2812.pdf** (en archive). La documentation concernant le MCP2562 est aussi disponible dans l'archive.

1. https://wiki.seeedstudio.com/Grove-Water_Atomization/

2. "Ce que l'on fait en programmation Android est tellement super que j'ai envie de faire une app sur mon smartphone qui affiche les mesures reçues..."

Afin de connecter la carte Basys 3 au CAN, il vous faut un transceiver MCP2562. La communication entre le CAN et le transceiver se fera sur une broche d'un PMOD de la Basys 3 (à vous de la définir). Le composant MCP2562 ayant besoin d'une alimentation en 5V (et la sortie sur les PMOD étant à 3.3V) vous pouvez repiquer sur la breadboard l'USB qui sort de la carte en utilisant l'adaptateur micro USB qui vous est fourni (voir figure 6).

Pour l'interconnexion avec les leds, il faut une adaptation de niveau du signal en sortie du PMOD vers un signal TTL. Pour cela nous vous fournissons un buffer SN74AHCT125N (voir la figure 6 ou <https://learn.adafruit.com/neopixels-on-raspberry-pi/raspberry-pi-wiring> pour un exemple de montage).

Le contrôle de la led représente l'éclairage. Afin de simuler cela, prévoyez dans le contrôle plusieurs niveaux d'éclairage qui se traduiront par un nombre plus ou moins important de leds allumées (vous pouvez aussi jouer sur les couleurs mais ce n'est pas nécessaire et risque de vous prendre du temps inutilement).

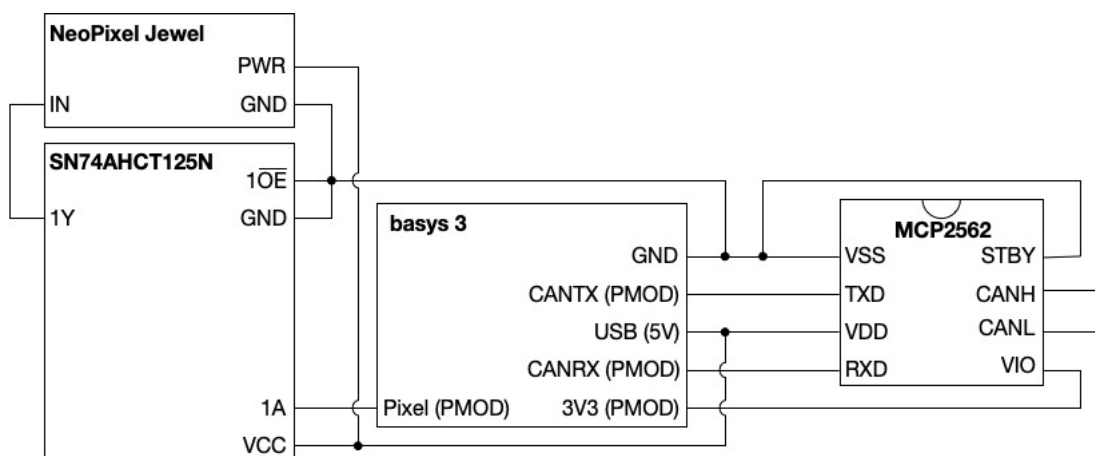


FIG. 6 – Schéma de connexion entre le basys 3, le MCP2562 et le NeoPixel Jewel

5 Etapes de réalisation des sous-systèmes

Le tableau 2 indique les étapes à réaliser pour implémenter chaque sous-système. **Assurez-vous de valider³ chaque étape avant de passer à la suivante.** Si vous n'arrivez pas au bout du développement vous aurez au moins des éléments fonctionnels à montrer lors de la démonstration. Vous pouvez utiliser ce tableau pour faire le point à chaque début et fin de séance, cela devrait vous aider à organiser votre travail. Les étapes ne représentent pas toute la même quantité de travail, à vous d'y faire attention.

3. Qui dit valider, dit prévoir des tests et s'assurer qu'ils passent...

TAB. 2 – Etapes de réalisation pour chaque sous-système

Etapes	Check
stm32-1 : thermique	
Mettre en place l'envoi périodique d'un message sur le CAN avec une donnée structurée du même type que le capteur thermique	
Echanger une trame avec le capteur thermique	
Récupérer et structurer les données du capteur thermique	
Envoyer les données du capteur sur le CAN	
stm32-2 : CO2	
Mettre en place l'envoi périodique d'un message sur le CAN avec une donnée de type CO2	
Réaliser et valider le montage pour le capteur de CO2	
Faire une bibliothèque pour les transactions d'écriture et de lecture UART	
Récupérer une donnée du capteur et la transmettre sur le CAN	
Intégrer le code sur la carte smt32-2 avec le code du capteur THP	
stm32-2 : THP	
Mettre en place l'envoi périodique d'un message sur le CAN avec une donnée structurée représentant les données du capteur THP	
Valider la communication SPI (en observant si nécessaire à l'analyseur logique)	
Faire une bibliothèque pour les transactions d'écriture et de lecture SPI	
Récupérer les données du capteur et les transmettre sur le CAN	
Intégrer le code sur la carte smt32-2 avec le code du capteur CO2	
stm32-3 : humidificateur	
Mettre en place la réception d'un message CAN pour contrôler l'humidificateur	
Faire la bibliothèque de contrôle de l'humidificateur	
Activer l'humidificateur suite à l'arrivée du message CAN	
FeatherM0 : MQTT	
Identifier les besoins en terme de messages, de QoS et de service MQTT à mettre en place (Keep alive, Last Will, etc.)	
Définir les topics MQTT et valider la réception sur un PC distant	
Implémenter un pont générique entre CAN et MQTT	
Réaliser les parsers des messages CAN vers les messages MQTT	
Eventuellement développer l'interface de supervision	
Basys 3 : CAN	
Réaliser le montage avec le transceiver	
Valider la bibliothèque CAN avec la réception de différents niveaux d'éclairage	
Intégrer avec le contrôleur de led	
Basys 3 : leds	
Réaliser le montage avec l'adaptateur de tension	
Concevoir l'automate d'envoi d'une trame de contrôle d'une led	
Concevoir l'automate d'envoi d'une trame de contrôle de plusieurs leds	
Intégrer avec la réception d'un message CAN	

6 Détails sur les activités pédagogiques

6.1 Objectifs du TD

Si vous en êtes là dans la lecture du document c'est que vous avez déjà bien avancé dans le TD... L'objectif de ce TD est avant tout de préparer la suite du travail et de passer par une première étape de conception.

Il faut qu'à la fin de la séance vous ayez au moins :

- identifié les composants du système,
- identifié le travail à réaliser sur chaque sous-système,
- défini l'ensemble des messages CAN qui seront utilisés.

Pour définir les messages CAN, créer un document partagé (pour pouvoir l'éditer au fur et à mesure) dans lequel vous indiquerez les messages qui seront échangés sur le CAN. **Pour chaque message, spécifier qui est l'émetteur, qui sont les récepteurs, quel est son identifiant et quel est le format des données.**

Une fois ce travail réalisé :

- répartissez vous en monôme ou binôme et attribuez vous un sous-système sur lequel commencer (vous vous attribuerez les sous-systèmes restant plus tard),
- parcourez les documents propres à votre sous-système pour bien identifier le travail qu'il faudra réaliser,
- identifiez pour votre sous-système les composants et les éléments logiciels qui seront nécessaires de produire,
- commencer par définir la structuration de ces éléments logiciels et des données qui seront utilisées.

Si vous avez suffisamment avancé, vous pouvez commencer à coder ou préparer les projets pour produire le code : inclusion des bibliothèques, fichier d'entête, etc.

6.2 Déroulement des TP

Vous allez avoir 2,5 séances de TP pour réaliser l'application, autant dire que ça va être serré. Il faut absolument que vous vous répartissiez le travail le plus efficacement possible. Moins vous travaillerez en binôme pour les phases d'implémentation, plus ce sera efficace (regarder une personne coder n'a jamais été très productif). Il est possible de scinder le projet en le découpant suivant les sous-systèmes. Le travail collectif se fait principalement sur la définition des données échangées.

Attention l'intégration est toujours une phase délicate. Normalement pour ce projet, si vous avez bien spécifié les données échangées sur le CAN ça ne devrait pas poser trop de problème.

6.3 Déroulement de la démonstration

La démonstration durera 10 min et se fera devant tout le monde. Le créneau prévu sera la dernière heure du dernier créneau de TP, vous n'aurez pas le droit d'avancer pendant cette heure même si vous n'êtes pas les premiers à présenter (comme dans Top Chef tout le monde s'arrête en même temps). Si vous ne faites pas l'intégration au fur et à mesure, prévoyez au moins 1h pour la faire (il y a de forte chance pour que ça ne suffise pas).

Le but de la démonstration est de montrer ce que vous avez réalisé. Il faudra donc vraiment le montrer en fonctionnement... Pensez à montrer chaque élément de manière indépendante pour valider votre implémentation. Vous pouvez utiliser le tableau 2 pour vous guider pendant votre démonstration et montrer ce qui fonctionne.

L'idéal serait que la personne qui a développé une partie en fasse la démonstration.

Bonne chance...

7 Liste matériel par groupe

Désignation	Référence	Quantité
Transceiver	MCP2562	1
Adaptateur micro USB	Adafruit Carte USB Micro B	1
Adaptateur TTL	SN74AHCT125N	1
Boost (Step-Up)	Pololu U3V12F12	1
Capteur THP	Grove 101020513	1
Capteur de CO2	XENSIV™ PAS CO2	1
Capteur thermique	D6T-44L-06H	1
Humidificateur	Grove Water Atomization V1.0	1
Leds	NeoPixel Jewel	1
Carte stm32CAN	STM32F303K8	3
Carte Feather	Feather M0 WiFi	1
Cart Basys 3	Basys 3	1
Breadboard	-	4
Résistances 10kΩ	-	2
Capacité 33 μF	-	1
Capacité 47 μF	-	1
Capacité 10 nF	-	1
Capacité 100 nF	-	1