



TP/Projet d'introduction aux systèmes embarqués : réalisation d'une station météo

Morgan Madec, Lucas Werling, Hassan Omran, Emmanuel Dervieux, Thomas Hingre, Phimant Anousith, Magali Frey, Wilfried Uhring

► To cite this version:

Morgan Madec, Lucas Werling, Hassan Omran, Emmanuel Dervieux, Thomas Hingre, et al.. TP/Projet d'introduction aux systèmes embarqués : réalisation d'une station météo. Journées Pédagogiques du CNFM, 2021, Saint Malo, France. <hal-03435839>

HAL Id: hal-03435839

<https://hal.science/hal-03435839v1>

Submitted on 18 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

TP/Projet d'introduction aux systèmes embarqués : réalisation d'une station météo

Morgan Madec^a, Lucas Werling^a, Hassan Omran^a, Emmanuel Dervieux^a, Thomas Hingre^a, Anousith Phimant^a, Magali Frey, Wilfried Uhring^a

^a Télécom Physique Strasbourg, Université de Strasbourg, France

Contact email : morgan.madec@unistra.fr

Le papier décrit un TP d'introduction aux systèmes embarqués réalisés avec les étudiants de Télécom Physique Strasbourg en deuxième année du cycle de formation ingénieur (BAC +4). Il vise à introduire les concepts fondamentaux des systèmes embarqués sur un exemple concret, en l'occurrence la réalisation d'une station météo, avec une approche de type projet. Les travaux pratiques se déroulent sur 2 jours pendant lesquels les étudiants apprennent à utiliser les interruptions et les timers, les protocoles de communication standards (SPI, I2C et 1-wire) et mettent en place une communication de type UART entre deux microprocesseurs.

I. Contexte

Télécom Physique Strasbourg forme des ingénieurs généralistes avec un spectre large allant de la physique fondamentale à l'informatique en passant par la photonique, l'électronique, l'automatique et le traitement du signal. Les étudiants de cette formation suivent un tronc commun de 3 semestres avant de se diriger dans un premier temps vers l'un des 4 départements de l'école (physique, informatique et réseaux, ingénierie des signaux et des systèmes et santé) lors du 4^e semestre, puis vers l'une des dix spécialités lors de la dernière année. L'une d'entre elles s'intitule « Electronique et Systèmes Embarqués » ou ESE et vise à former les étudiants plus spécifiquement à la conception et la programmation des systèmes embarqués, avec une composante majeure « hardware » [1].

Au cours du 3^e semestre, tous les étudiants suivent le module « Introduction aux systèmes embarqués ». Le cours est précédé, en première année, d'un cours d'informatique en langage C, d'un cours d'électronique numérique et d'un cours d'introduction aux microcontrôleurs incluant notamment deux TP d'initiation à Arduino. Enfin, en préambule du TP/ projet, les étudiants suivent également un cours de 2 heures d'introduction générale aux systèmes embarqués et ont à leur disposition une dizaine de vidéos de 15 minutes détaillant notamment les notions d'interruption, l'utilisation des timers, l'utilisation des interruptions sur Arduino, des généralités sur les protocoles de communication (série/parallèle, half-duplex/full-duplex, synchrone/asynchrone), le protocole SPI, le protocole I2C et l'UART [2].

Le choix de travailler avec Arduino est bien entendu discutable lorsque l'on parle d'un cours sur les systèmes embarqués. Nous l'avons toutefois fait pour faciliter au maximum la prise en main en occultant totalement certains aspects souvent difficiles comme la configuration du microcontrôleur et de son environnement de développement (IDE). Cela permet de répondre au mieux aux deux objectifs du module : i) être suffisamment attractif pour donner aux étudiants les plus intéressés l'envie de poursuivre leur cursus vers la

finalité ESE et ii) être suffisamment général pour donner à tous les étudiants, quelque soit la spécialité qu'ils choisiront, les clefs pour comprendre les problématiques liées aux systèmes embarqués et leur permettre d'aborder la réalisation de systèmes embarqués dans leur domaine de spécialisation propre. Les enseignements de spécialité ESE se font, par contre, sur une plateforme plus en lien avec les outils industriels, en l'occurrence un MSP430, ou ces problématiques de configuration sont abordées. De plus, le TP/projet se limite volontairement à des protocoles de communications filaires. Les protocoles sans fils (Bluetooth, Zigbee, WiFi) ne sont quant à eux abordés qu'en enseignement de spécialité.

II. Cahier des charges

Les étudiants sont en charge de réaliser une station météo composée de deux cartes Arduino communiquant entre elles via une liaison série (UART). La carte-maître gère essentiellement l'interface utilisateur, c'est-à-dire un écran LCD et deux boutons. Elle affiche en permanence l'heure, la température (rafraîchie toutes les secondes) ainsi qu'en alternance deux des 4 autres mesures météo (humidité, pression atmosphérique, qualité de l'air, luminosité). Le système est contrôlé par deux boutons permettant d'alternier entre les modes d'affichage et de régler l'heure. La carte-esclave pilote quant à elle quatre capteurs via des protocoles de communication variés. La communication entre les deux cartes se fait en protocole série (UART). La carte-maître pilote l'ensemble des opérations : elle envoie une commande à la carte-esclave pour que celle-ci interroge ses capteurs uniquement quand cela est nécessaire, c'est-à-dire soit toutes les secondes, soit au moment de rafraîchir l'affichage après avoir appuyé sur les boutons.

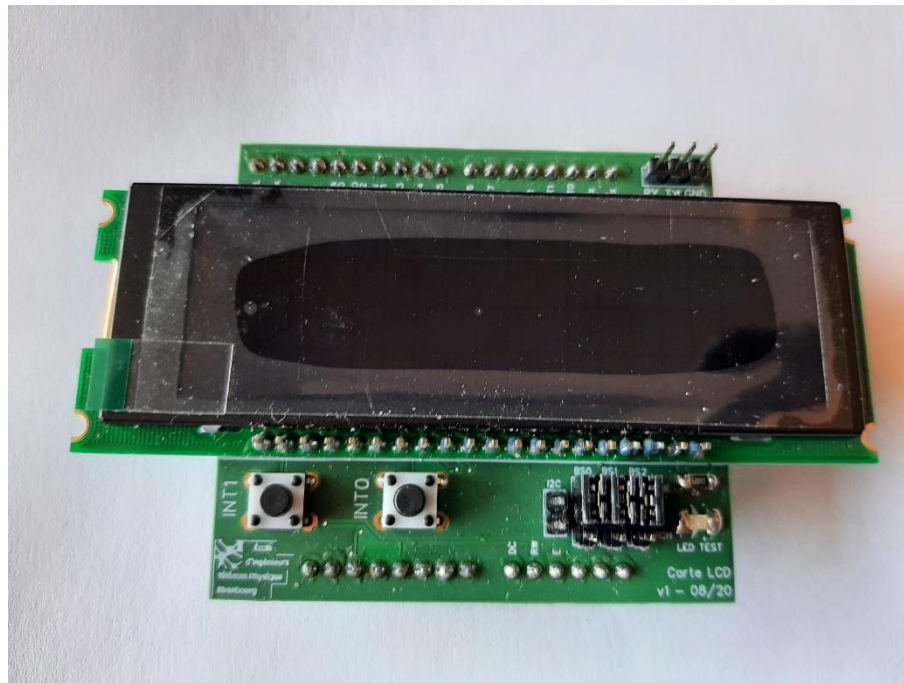


Fig.1. « Shield Arduino » réalisé pour la carte-maître et composé d'un afficheur, de deux boutons de commande branchés sur les entrées d'interruption de l'Arduino et d'un connecteur UART.

III. Matériel

La carte-maître

La carte-maître prend la forme d'un « shield Arduino » constituée d'un écran LCD 4x20 caractère Newhaven NHD-0420CW-AY3 [3] et de deux boutons connectés respectivement sur les entrées d'interruption INT0 et INT1 de la Arduino Uno (Figure 1). L'écran dispose de 4 modes de communication avec le microcontrôleur : les interfaces parallèles 4 et 8 fils ainsi que les interfaces série SPI et I2C. La carte est prévue pour permettre d'utiliser n'importe laquelle de ses interfaces en positionnant trois cavaliers. Pour le TP, nous privilégierons l'interface SPI. Les étudiants pourront également tenter de communiquer avec l'écran en utilisant les autres interfaces s'ils en ont le temps. La carte dispose également de trois broches pour la communication UART (signaux RX, TX et GND) avec la carte-esclave.

La carte-esclave

La carte-esclave est constituée quant à elle de quatre capteurs (Figure 2):

- Un capteur de température et d'humidité DHT22 [4] monté sur un module Grove. Il communique ses données via une liaison 1-wire. À chaque requête, initié par le microcontrôleur en plaçant la ligne de communication au niveau bas, le capteur délivre une trame de 40 bits à un baudrate fixe. Ces 40 bits sont constitués de 16 bits indiquant la température, 16 bits indiquant l'humidité et 8 bits de checksum ;
- Un capteur de pression atmosphérique HP206C [5] également monté sur un module Grove. Celui-ci communique une donnée de pression atmosphérique codée sur 20 bits et transmise en I2C. Il effectue également une mesure de température qui peut donc être corrélée à celle du DHT22 ;
- Un capteur de luminosité VEML7700 [6] retournant une mesure de l'éclairage ambiant également codée sur 16 bits. La sensibilité et le temps d'intégration du capteur peuvent également être contrôlés par le microcontrôleur via l'interface I2C.
- Un capteur de qualité de l'air Winsen MP503 [7] disposant d'une sortie analogique dont la tension est relative à la concentration de divers polluants dans l'air tels que les émissions d'alcool, le méthane ou encore la fumée.

La carte dispose également du même connecteur permettant la communication série entre les deux cartes, d'un interrupteur permettant d'interrompre facilement la communication série sans retirer les filles (nécessaire à la programmation individuelle de chacune des deux cartes puisque celle-ci utilise également l'UART) ainsi que 4 BNC permettant de monitorer les signaux I2C, le signal 1-wire issu du capteur de température/humidité et la tension analogique issue du capteur de pollution.

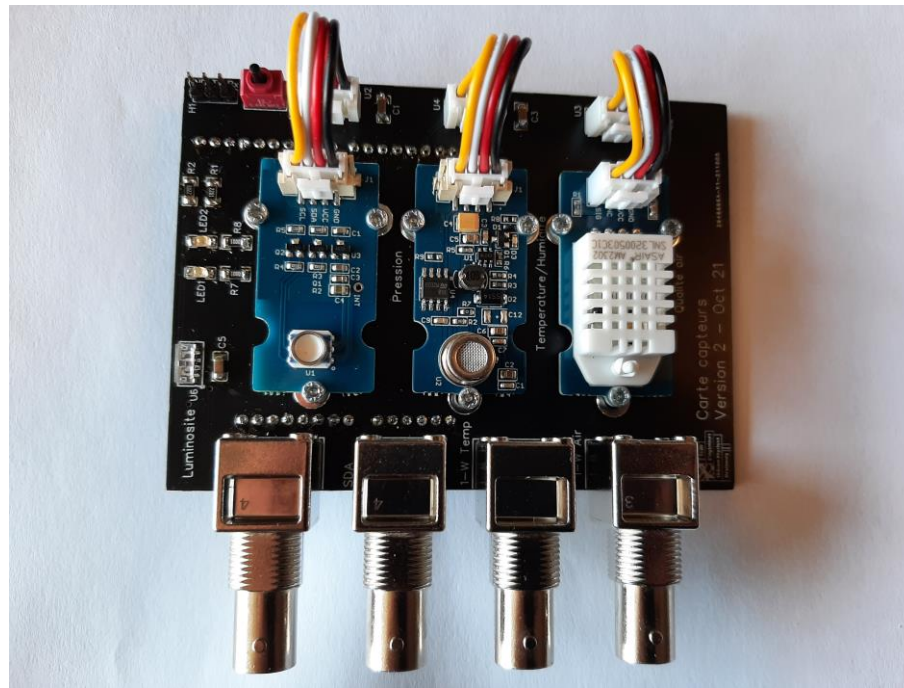


Fig.2. « Shield Arduino » réalisé pour la carte esclave composée de quatre capteurs, quatre BNC pour monitorer les signaux issus des capteurs et un connecteur UART.

IV. Déroulement du TP

Le TP se déroule en mode projet, c'est-à-dire que l'on fournit aux étudiants le cahier des charges, les documentations techniques des 4 capteurs et les schémas des deux shields. Toutefois, pour les guider, nous leur proposons un certain nombre de points de validation listés ci-dessous :

- 1) Pilotage de l'afficheur LCD
 - a. Communiquer avec l'écran LCD en SPI ;
 - b. Afficher un message statique à un endroit déterminé de l'afficheur ;
 - c. Afficher un message dynamique (par exemple un nombre de secondes à un endroit déterminé de l'afficheur).
- 2) Gestion des interruptions et des timers :
 - a. Utiliser les interruptions INT0 et INT1 pour piloter une LED ;
 - b. Utiliser le timer 2 et son interruption pour faire clignoter une LED à une fréquence de 1 Hz ;
 - c. Utiliser un même timer pour générer deux signaux synchrones à des fréquences différentes, par exemple 1 Hz et 4 Hz.
- 3) Communication avec les capteurs :
 - a. Communiquer avec le capteur de température/pression seul en 1-wire et afficher les valeurs lues sur le moniteur série ;
 - b. Communiquer avec le capteur d'humidité seul en I2C et afficher les valeurs lues sur le moniteur série ;
 - c. Communiquer avec le capteur de qualité de l'air seul en utilisant les convertisseurs analogiques numériques et afficher les valeurs lues sur le moniteur série ;

- d. Communiquer avec le capteur de luminosité seul en I2C et afficher les valeurs lues sur le moniteur série. Faire en sorte d'adapter automatiquement la sensibilité et/ou le temps d'exposition de sorte que les valeurs ne soient jamais hors limite ;
 - e. Rassembler en un seul programme les codes des 4 capteurs et commander la lecture d'un des capteurs à l'aide d'un caractère envoyé depuis le moniteur série.
- 4) Communication entre processeurs
- a. Allumer une LED sur la carte-esclave lorsque l'on appuie sur un bouton de la carte maître ;
 - b. Allumer une LED sur la carte-maître lorsque l'on appuie sur un bouton de la carte-maître après avoir initié une communication va-et-vient entre les deux cartes (le maître envoie une commande à l'esclave qui y répond, ce qui déclenche l'allumage de la LED).
 - c. Echanger une chaîne de caractère entre les deux Arduino.

V. Résultats attendus

Les résultats attendus sont décrits dans une vidéo disponible sur YouTube [8] et sur les Figures 3 à 5. Sur la Figure 3, on peut voir le montage complet, avec les deux cartes Arduino portant la carte d'affichage pour l'une et la carte de capteur pour l'autre. Les deux Arduino sont reliés au PC via deux ports USB distincts. Les deux cartes sont reliées via leur liaison série (fils bleu et orange). Sur l'afficheur, la première ligne correspond à l'heure, la seconde à la température, puis les deux suivantes à la pression atmosphérique, l'humidité, la qualité de l'air et la luminosité alternativement. La Figure 4 est un montage montrant les 4 affichages possibles.

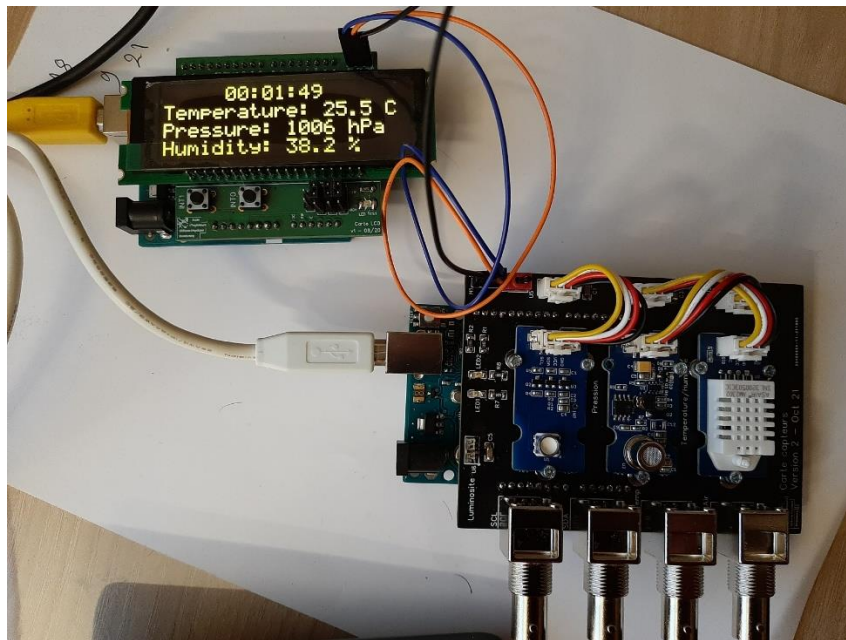


Fig.3. Montage complet composé des deux cartes Arduino surmontées des deux shields, connectés au PC via un port USB pour l'alimentation et la programmation et connectés entre elles via l'UART (RX/TX).



Fig.4. Montage montrant les quatre configurations d’affichage possibles. L’heure et la température sont affichées en permanence, la qualité de l’air, la luminosité, la pression atmosphérique et l’humidité en alternance, contrôlée par un bouton.

D’un point de vue de la programmation, deux IDE Arduino sont nécessaires, l’une pour piloter la carte maître et l’autre pour piloter la carte esclave (Figure 5). Pour chaque IDE, le moniteur série associé permet de suivre la communication qui s’établit entre les deux Arduino. On voit par exemple en bas à gauche de la Figure 5 que le maître communique avec l’esclave au moyen de caractère (T pour demander une mesure de température, P pour une mesure de pression, et H pour une mesure d’humidité). De la même manière, on peut lire la réponse de l’esclave en bas à droite de la Figure 5. Il s’agit d’une trame de 12 caractères. Les 4 premiers correspondent à la température (ici 24.4°C), les quatre suivants à la pression atmosphérique (1006 hPa) et les quatre derniers à l’humidité (38.4 %).

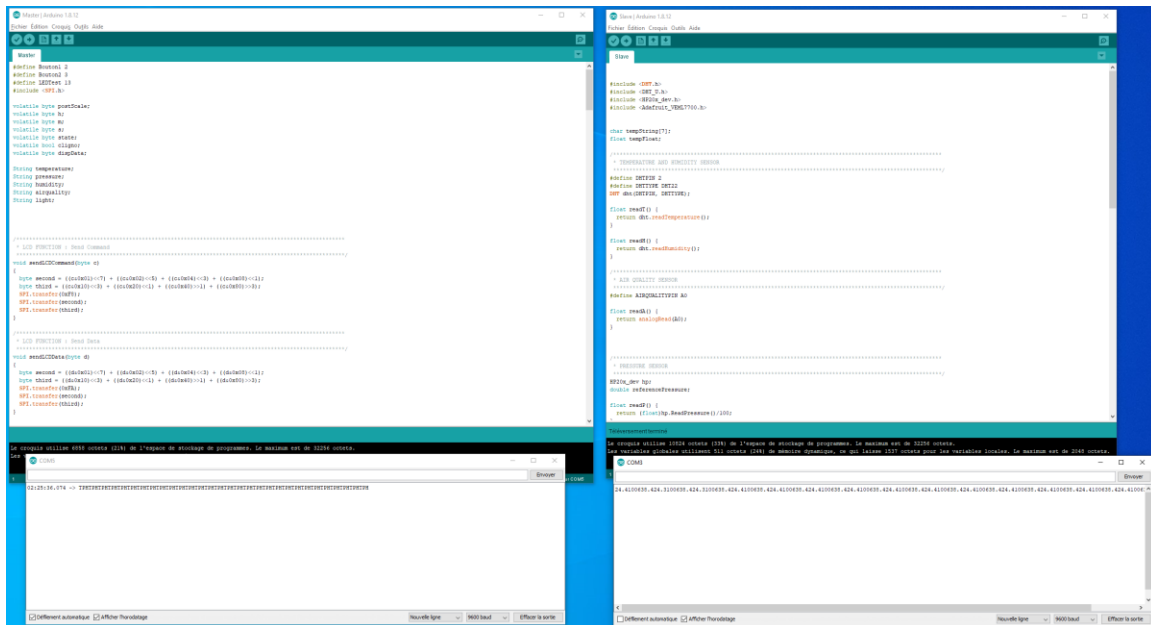


Fig.5. Capture d’écran montrant les deux IDE Arduino, celle du maître à gauche et celle de l’esclave à droite. Les fenêtres du bas correspondent aux moniteurs séries respectifs de chacun des IDE.

VI. Compétences acquises et conclusion

Ce TP permet aux étudiants d'acquérir les compétences suivantes :

- Programmer sur Arduino une application complète et conséquente ;
- Élaborer une démarche permettant d'arriver à ce but en décomposant le problème et en testant individuellement chaque partie ;
- Intégrer en un seul projet plusieurs fragments de code validés individuellement ;
- Utiliser des bibliothèques Arduino externes développées pour chaque capteur ;
- S'appuyer sur une documentation technique, sur un GitHub ou sur des exemples de codes pour comprendre le fonctionnement à la fois du matériel et du logiciel associés aux protocoles de communication.

Références

1. Télécom Physique Strasbourg – École d'ingénieur de l'Université de Strasbourg. *Site web*: <https://www.telecom-physique.fr/>
2. Chaîne YouTube contenant les vidéos de cours associés aux travaux pratiques décrits dans ce papier : <https://www.youtube.com/playlist?list=PLSuadvVXceA2N5o7QT1mYIIVV7NpYWnKd3>
3. Documentation technique de l'écran LCD Newhaven NHD-0420CW-AY3 disponible sur le site web de Newhaven. *Site web*: <https://www.newhavendisplay.com/specs/NHD-0420CW-AY3.pdf>
4. Documentation technique capteur de température et d'humidité DHT22 disponible sur le site web de Lextronic. *Site web*: <https://www.lextronic.fr/capteur-humidite-temperature-dht-22-14983.html>
5. Documentation technique capteur de pression atmosphérique HP206C disponible sur le site web de Lextronic. *Site web*: <https://www.lextronic.fr/module-grove-barometre-101020068-20868.html>
6. Documentation technique capteur de luminosité VEML7700 disponible sur le site web de Digikey. *Site web*: <https://www.digikey.fr/product-detail/fr/vishay-semiconductor-opto-division/VEML7700-TR/VEML7700CT-ND/5820244>
7. Documentation technique capteur de qualité de l'air MP503 disponible sur le site web de Winsen. *Site web*: <https://www.winsen-sensor.com/sensors/voc-sensor/mp503.html>
8. Vidéo de démonstration : <https://youtu.be/Fn9i2dyTzvA>