

Médian AG43

Automne 2017

Vendredi 10 Novembre 2017

Durée : 1h00

Exercice 1 : Lecture d'une chaîne de caractères complète

Une platine Arduino est un petit ordinateur comportant un petit microcontrôleur, ne pouvant communiquer que par une liaison série avec un ordinateur avec lequel elle est connectée et disposant de connexions logiques et analogiques d'entrées/sorties.

On souhaite traiter des informations (des commandes) provenant par cette liaison série. Les caractères provenant de la liaison série sont placés automatiquement dans une zone mémoire appelée tampon.

On dispose de plusieurs fonctions utiles pour la gestion des communications, dont celles qui serviront dans cet exercice :

- `available()` : fonction qui retourne le nombre de caractères disponibles à la lecture dans le tampon de réception du port Série
- `read()` : fonction qui retourne le premier caractère disponible dans le tampon de lecture, et le retire du tampon. Si aucun caractère est disponible, elle retourne 0.

On considère que la chaîne de caractères constituant une commande est complète, lorsque l'on lit le caractère LF (code ASCII 10). Le tampon de lecture ne sera pas forcément vide, car il est susceptible de contenir plusieurs commandes.

On demande d'écrire une fonction appelée `getCommand(commande)`, qui admet comme argument la chaîne de caractères `commande` (elle est initialement vide), qui place dans cette chaîne la suite de caractères disponibles dans le tampon de réception jusqu'au premier caractère LF rencontré. Elle remplace le caractère LF par le caractère NUL, ce qui permet de délimiter la chaîne de caractères pour être conforme à la norme du langage C. Enfin, elle retourne le nombre de caractères placés dans la chaîne `commande`, sans compter le caractère NUL.

Exercice 2 : Envoi journalier de données enregistrées

On travaille sur une station météo connectée au réseau domestique (wifi). Le logiciel comporte une boucle principale qui commence par lire la date et l'heure et décide s'il faut effectuer une mesure en fonction de l'intervalle souhaité. Cette boucle principale est exécutée plusieurs fois par seconde.

Le travail demandé consiste à écrire un morceau de cette boucle qui est chargé, à chaque changement de jour (donc vers minuit), d'appeler la fonction `sendLog()` qui envoie par email les données enregistrées durant les 24 dernières heures. Vous n'avez pas à écrire la fonction `sendLog()`, on suppose qu'elle existe déjà.

Le morceau de programme est très simple et ne comporte que quelques lignes. Attention à ce qu'il n'envoie pas plusieurs fois de suite les données. La difficulté consiste à détecter le changement de jour.

Vous disposez des variables entières `hours`, `mn`, `sec`, `monthDay`, `month`, `year`, `weekDay` qui sont initialisées en début de boucle. `monthDay` indique le jour du mois (1, ... ,31) et `weekDay` le jour de la semaine (1 pour Lundi, et 7 pour Dimanche). Si vous avez besoin de créer une variable, précisez son type et son utilité.

Exercice 3 : Insertion d'une chaîne dans une autre

Ecrire une procédure appelée `insert(dest, ins, place)` qui insère la chaîne de caractères `ins` dans la chaîne `dest` à partir de la position `place`. Les chaînes se terminent par le caractère NUL. On dispose de la fonction `strlen(chaine)` qui retourne la longueur de la chaîne.

Exemple (pour plus de clarté, les espaces sont représentés par des caractères "_") :

```
dest = "Quelle_journée"  
ins = "_belle"
```

Après appel de la procédure `insert` sous la forme suivante :

```
insert(dest, ins, 6)
```

nous aurons :

```
dest = "Quelle_belle_journee"
```