01/12/2017

Guilpain Léo & Legris Thomas

Compte rendu

TP SEM1

Table des matières

[Chapitre 2 : Communication série RS-232 2](#_Toc502959360)

[1. Configuration de la communication série 2](#_Toc502959361)

[2. Envoi d’un caractère 3](#_Toc502959362)

[3. Envoi d’une chaîne de caractères 3](#_Toc502959363)

[4. Réception d’une chaine caractère 3](#_Toc502959364)

[5. Communication bidirectionnelle 4](#_Toc502959365)

[Chapitre n°3 : Télémètre à ultrasons 5](#_Toc502959366)

[Chapitre n° 4 : Joystick 8](#_Toc502959367)

[8](#_Toc502959368)

[Chapitre n°5 : Afficheur LCD 10](#_Toc502959369)

[1. Bus de données 10](#_Toc502959370)

[2. Commande du LCD 11](#_Toc502959371)

[3. Initialisation du LCD 11](#_Toc502959372)

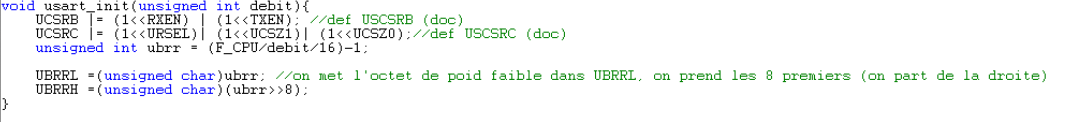
[4. Envoi d’un caractère au LCD 12](#_Toc502959373)

[5. Envoie d’une chaine de caractères au LCD 12](#_Toc502959374)

Chapitre 2 : Communication série RS-232

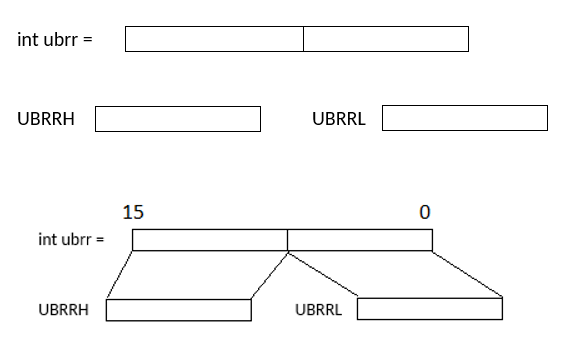
1. Configuration de la communication série

Les caractéristiques de la communication sont gérées dans les paramètres de la communication et non dans le code.

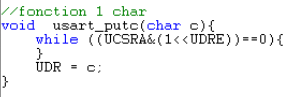


Dans cette fonction, on effectue différentes actions :

* Dans le registre UCSRB :
  + RXEN à 1 : Active la réception
  + TXEN à 1 : Active la transmission
* Dans le registre UCSRC :
  + URSEL à 1 : Sélection de UCSRC
  + UCSZ1 à 1 : 8 bits de données
  + UCSZ0 à 1 : 8 bits de données

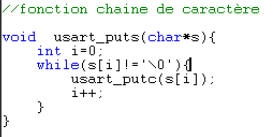


2. Envoi d’un caractère



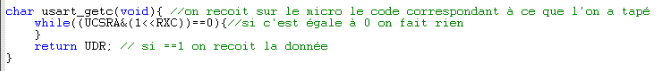
On vérifie dans un premier temps que UDR est prêt à recevoir de nouvelles données. Si UDR est vide cela signifie qu’il est prêt, on peut ainsi stocké la valeur du caractère dans UDR.

3. Envoi d’une chaîne de caractères



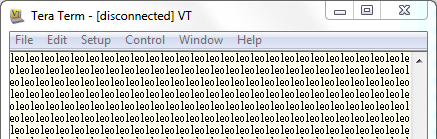
On parcourt la chaine de caractère s (par exemple « toto ») jusqu’à la fin qui est determinée par « \0 ». A chaque itération, on appelle la fonction créées précedement. On stocke ainsi le caractère lu dans UDR à chaque itération.

4. Réception d’une chaine caractère

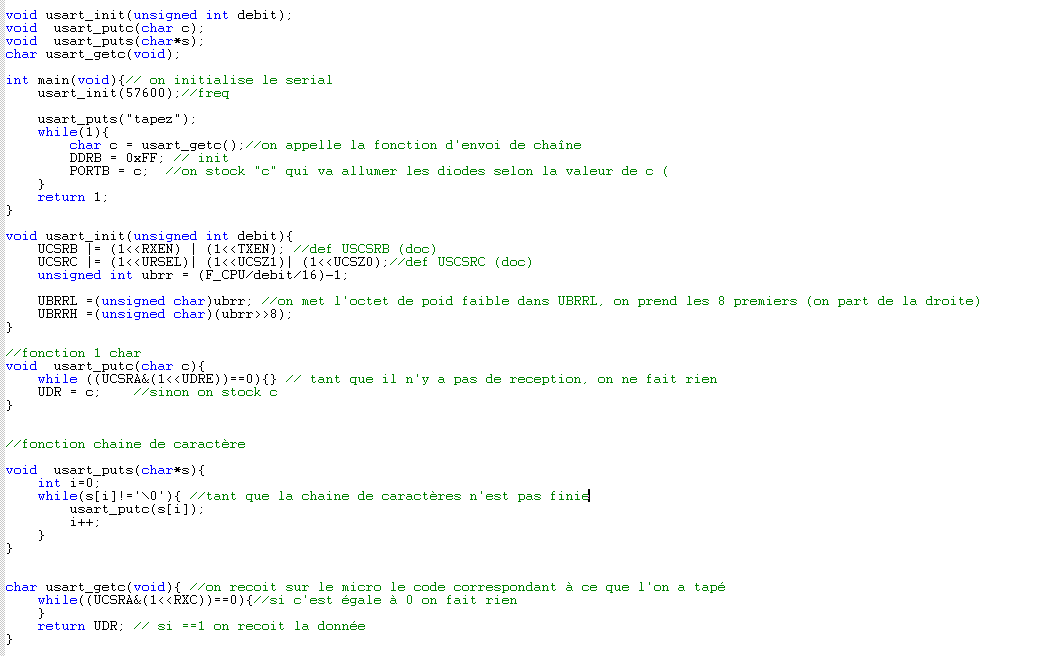


Tant que toutes les données n’ont pas été reçues on ne fait rien, ensuite on renvoie les données qui ont été stockées dans UDR.

Dans l’exemple, on a envoyé la chaine de caractère « leo ».



5. Communication bidirectionnelle

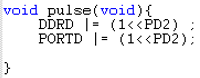
Voici le code complet avec la réception de la chaine de caractère :

Voici le résultat sur les LEDS lorsque l’on appuie sur la lettre « a » du clavier. C’est le code ASCII qui est renvoyé sur les LEDS

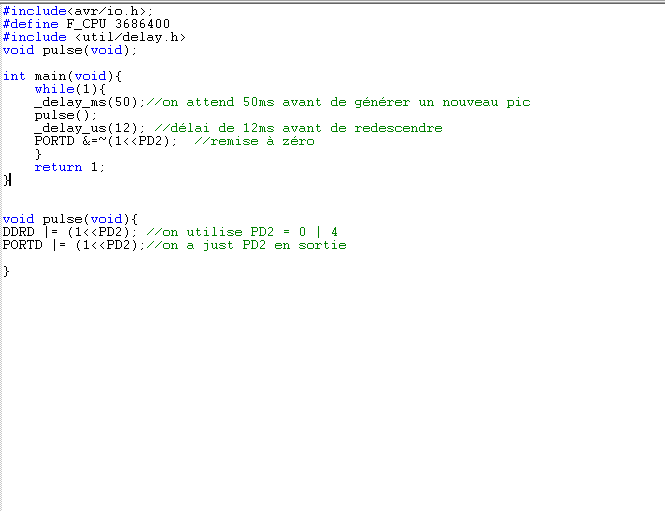
Une image contenant équipement électronique, circuit

Description générée avec un niveau de confiance élevé

Chapitre n°3 : Télémètre à ultrasons

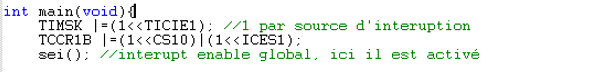


Ici, on a créé une fonction *pulse(void)* permettant de mettre le PORTD n°2 en sortie et à 1. On génère ici un front montant.

Ensuite, dans la fonction main, on va créer des impulsions (pics avec front montant et front descendant).

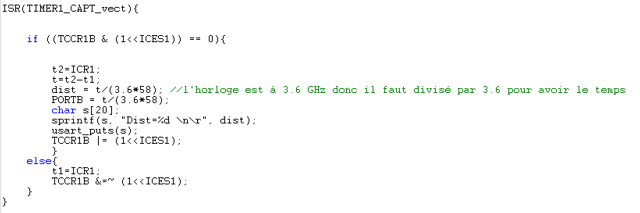
On peut voir qu’on génère un front montant, puis on patiente 12 secondes avant de générer le front descendant (remise à zéro du PORTD n°2).

Ensuite, on gère les interruptions :



Comme écrit ci-dessus, on active dans un premier temps les interruptions (sei).

* Registre TIMSK :
  + TICIE1 à 1 : Activation de l’entrée capture du Compteur/Timer



Lorsqu’il y a une interruption, si on a un front descendant (ICES1 à 0), on stocke la valeur de ICR1 dans la variable t2 puis on fait passer la valeur de ICES1 de 0 à 1. Si on a un front montant (ICES1 à 1), on stocke la valeur de ICR1 dans la variable t1 et on fait passer ICES1 de 1 à 0.

Ensuite, on calcule le temps final t = t2 – t1.

Le SFR04 fournit une echo pulsation proportionnelle à la distance. On a mesuré la largeur du pic en microseconde, c’est pourquoi, pour avoir la distance en centimètre, il faut diviser par 58. Ensuite on divise par la fréquence de l’horloge.

On stocke cette distance sur le PORTB.

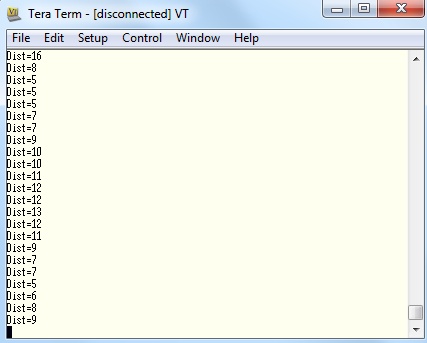
Explication de sprintf :

On réutilise les fonctions du TP précédent pour faire une communication série et transmettre une chaine de caractères.

Voici le code complet (raccord de deux images) :



Lorsque la connection série est activée, on obtient ceci (distance en centimètre) :



Chapitre n° 4 : Joystick

Une image contenant capture d’écran, texte

Description générée avec un niveau de confiance élevé

On réutilise les fonctions des TPs précédents pour la communication série.

* Registre ADMUX :
  + ADLAR à 1 : Ajuster le résultat à gauche. On prend que les bits de poids fort.
* Registre ADCSRA :
  + ADEN à 1 : Active l’ADC
  + ADATE à 1 : Active l’auto ADC Trigger

On ne peut pas gérer les deux axes simultanément, il faut gérer un axe après l’autre.

Sur le 1er axe :

* Registre ADMUX :
  + MUX0 à 0 : Correspond à ADC0
* Registre ADCSRA :
  + ASDC à 1 : Démarre la conversion

Tant que le bit d’interruption n’est pas activé, on ne fait rien (ADIF à 0). En revanche, dès qu’il est activé, on stocke les valeurs de ADCH sur PORTB.

Ensuite, on envoie les valeurs de PINB dans la connexion série et on récupère les valeurs.

Sur le 2ème axe :

* Registre ADMUX :
  + MUX0 à 0 : Correspond à ADC1
* Registre ADCSRA :
  + ASDC à 1 : Démarre la conversion

Tant que le bit d’interruption n’est pas activé, on ne fait rien (ADIF à 0). En revanche, dès qu’il est activé, on stocke les valeurs de ADCH sur PORTB.

Ensuite, on envoie les valeurs de PINB dans la connexion série et on récupère les valeurs.

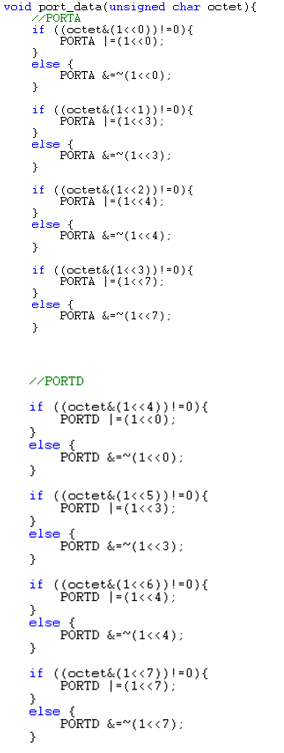
Après la connexion en série, on obtient ceci :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

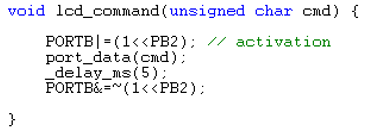
Chapitre n°5 : Afficheur LCD

1. Bus de données



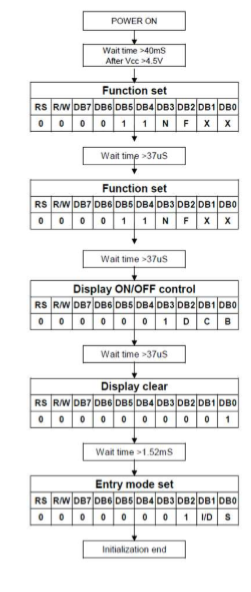
Cette fonction permet d’envoyer l’octet « octet » au bus de données.

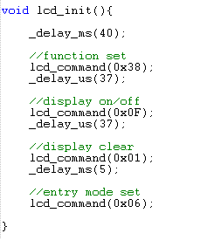
2. Commande du LCD



On active le bit E en mettant PB2 à 1, puis on envoie la commande « cmd » sur le bus de données puis après avoir attendu 5 millisecondes, on désactive le bit E.

3. Initialisation du LCD



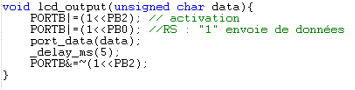


Pour initialiser le LCD, il faut suivre la procédure de gauche.

Les différentes commandes sont :

* Function Set (0x38)
* Display ON/OFF (0x0F)
* Display clear (0x01)
* Entry mode set (0x06)

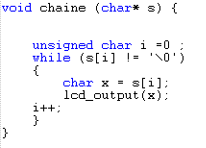
4. Envoi d’un caractère au LCD



Le but est d’envoyer la donnée au LCD (ici « data »).

Pour cela, on fait passer le bit E (PB2) à 1 et on fait passer le bit RS (PB0) à 1, cela correspond à l’envoie de données. Ensuite on envoie la donnée « data » sur le bus, puis après 5 millisecondes, on fait passer le bit E à 0.

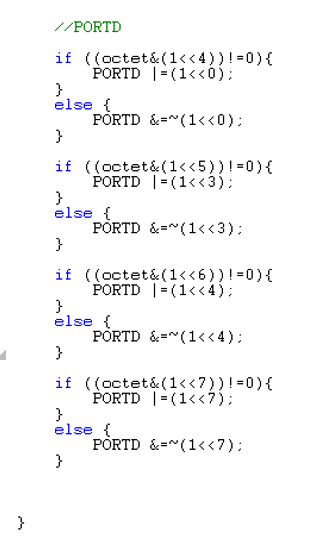
5. Envoie d’une chaine de caractères au LCD



On envoie un caractère après l’autre sur le bus de données. Voici ce que l’on obtient après avoir envoyé « toto ».

Une image contenant équipement électronique

Description générée avec un niveau de confiance élevé

Une image contenant capture d’écran

Description générée avec un niveau de confiance élevéVoici le code complet :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé