

5TC-BED, TP1 ez430-RF2500

Port série UART et ADC10

October 1, 2019

But du TP

Prise en main du port série UART beaucoup utilisé en embarqué pour communiquer avec entre un ordinateur. Les chapitres correspondant dans la documentation sont les chapitres 15 et 16: USCI pour Universal Serial Communication Interface. Nous étudierons aussi le convertisseur analogique numérique ADC10 (chapitre 20) essentiellement utilisé pour récupérer la température de la carte ez430.

1 Mode UART

L'USCI est un périphérique qui permet d'implémenter plusieurs protocoles série synchrones (SPI, I2C) ou asynchrone (UART, IrDA). Il est divisé en deux modules (USCI A et USCI B) qui peuvent être utilisés en parallèle. Le module USCI A peut être configuré pour transmettre des données suivant un des protocoles suivants: UART, IrDA ou SPI.

L'UART est utilisé pour transmettre des informations entre la clé EZ430 et l'ordinateur hôte. Lorsque l'on branche la clé sur le port USB, un port (COM4 sous windows, ttyACM0 sur linux) apparaît sur la machine hôte. Il faut alors configurer la connexion série de la même manière sur la machine hôte et sur le MSP qui pourra alors envoyer des informations en utilisant, par exemple, la fonction `printf`.

La mise au point du module UART est relativement complexe car elle fait intervenir des notions de modulation et de communication analogique. Nous allons survoler le fonctionnement pour être capable de comprendre les mécanismes importants.

En général la communication depuis un PC vers par le port série se fait par un logiciel dédié (`minicom` pour linux, expliqué en annexe, ou `hyperterminal` pour Windows). Malheureusement le ports série de l'ez430 ne fonctionne pas très bien avec ces outils à cause d'un bug du driver d'encapsulation de l'uart de l'ez430 dans l'USB (puisque au final, c'est bien une communication usb qui transporte l'information). Nous vous proposons un utilitaire appelé `ezconsole` qui permet de faire l'interface avec l'ez430.

- Parcourez le chapitre 15 de la documentation (15-20 minutes) de manière à avoir une idée globale de la structure de l'USCI A0. Regardez en particulier: La figure 15-1 (p15-4), la section 15.3.11 (sélection de la vitesse en bauds), La table 15-4 (configuration "standard" des registres) et les registres associés à USCI A0. Les registres `UCA0CTL0` et `UCA0CTL1` (USCI_A0 control register) sont utilisés pour contrôler les options de transmission, les registres `UCA0BR0` et `UCA0BR1` (USCI_A0 Baud rate control register) permettent de régler le baud rate en fonction de l'horloge. les registres `UCA0RXBUF` et `UCA0TXBUF` (USCI_A0 receive & transmit buffer) contiennent le caractère transmis ou reçu.
- Analyser le driver `$SRC/ez430-drivers/src/uart.c`, comprenez chacune des commandes de la fonction `uart_init()`, même chose pour les fonctions `putchar()` et `getchar()`.
- Déplacez vous dans le répertoire `$SRC/ez430-drivers/example/serial`

Q1— Programmez la clé EZ430 avec ce programme et vérifiez que le port série fonctionne en lançant `ezconsole` (binaire disponible sur Moodle).

Q2— Modifier l'écho (par exemple en faisant un double écho), bien comprendre que la communication dans les deux sens n'a aucune raison d'être symétrique

2 Convertisseur Analogique Numérique

Le MSP430 est équipé d'un convertisseur analogique-numérique capable d'échantillonner une valeur analogique (tension sur l'un des 10 fils d'entrée) sur 10 bits. L'information lue par le MSP430 correspond à une tension, la conversion est réglée par les paramètres V_{R+} et V_{R-} qui sont fixé par l'utilisateur: V_{R-} correspondra à l'information digitale 0 et V_{R+} à l'information digitale maximale: 0x3FF.

Comme pour les autres périphériques, il faut d'abord sélectionner l'horloge ADC10CLK qui va servir à l'échantillonnage et à la conversion. Une fois cette horloge réglée, l'échantillonnage est déclenché par le signal SHI (plusieurs déclencheurs possibles) il prend une certaine durée puis la conversion en numérique de la valeur capturée prend 13 cycles d'horloge. Étudiez l'architecture de l'ADC en figure 20-1. Lisez les section 20.2.1, 20.2.3, 20.2.5.

L'ADC a plusieurs modes de fonctionnement: il peut échantillonner une seule valeur, faire une série d'échantillonnage, échantillonner en continu, etc. Ces différents modes complexifient la maîtrise du composant, notamment par la possibilité de transférer des blocs d'échantillons plutôt que de les lire un par un. Nous étudierons uniquement l'échantillonnage d'une valeur déclenchée par une interruption du timer (figure 20.5). Étudiez les registres de configuration de l'ADC: ADC210CTL0 et ADC10CTL1

- Même travail que pour le driver `uart`, allez voir le driver `adc10` `$SRC/ez430-drivers/src/adc10.c` et comprenez les commandes de la fonction `adc10_sample_temp`
- Déplacez vous dans le répertoire `$SRC/ez430-drivers/example/adc-temp`
- Ouvrez le fichier `main.c`, que fait ce programme?

Q3— programmer le capteur, vérifier que vous recevez bien la température, la mesure est-elle correcte?

- Modifier ce programme pour afficher les information d'une autre manière (plus jolie).
- Essayer de proposer une calibration de la température.

A Annexe 1: configuration du port série sous linux avec minicom

Minicom est un programme de contrôle de modem et d'émulation de terminal pour les OS Unix-like, écrit par Miquel van Smoorenburg d'après le populaire Telex pour MS-DOS. Minicom apporte une émulation totale ANSI et VT100, un langage de script externe, et d'autres choses encore. Nous allons l'utiliser pour communiquer avec le port série créé par la connection du cable USB-série sur le device `/tty/USB0`

Il faut configure le débit du port série (9600 bauds) et le device linux correspondant au port série (`/dev/ttyUSB0`).

On peut soit lancer minicom avec les bon paramètres: `minicom -b 9600 -D /dev/ttyUSB0`

Soit lancer minicom en mode setup pour le configure manuellement (souvent nécessaire): `minicom -s`

Selectionner "configuration du port série", choisir la lettre (A et E) pour configurer le device et le débit (9600 8N1). Une fois en mode de fonctionnement normal, vous voyez dans le terminal ce qui arrive sur le port série et ce que vous tapez au clavier est envoyé sur le port série (caractères ascii). Pour rentrer en mode configuration taper `Ctrl-a z` puis `0` pour configurer minicom. Enfin `Ctrl-a q` pour sortir