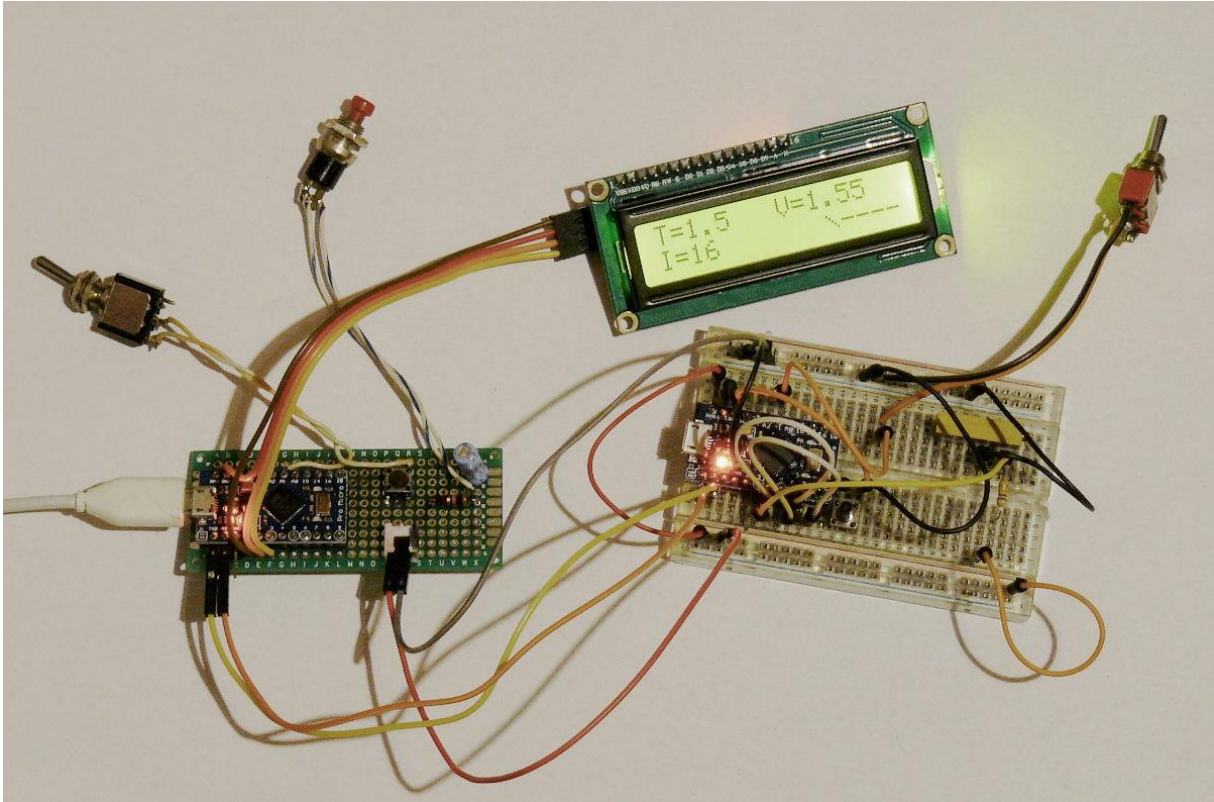


Modules de mesure de température-tension-courant

20/02/2018 – P. Chour

Le montage décrit dans ce document est destiné, d'une part à acquérir des mesures de température, courant et tension (module dit « mobile ») et d'autre part, récupérer ces informations transmises en asynchrone et les afficher sur un afficheur et éventuellement, sur une console branchée en USB (module dit « fixe »).



Mode d'emploi du module « fixe »

Objet

Le module « Fixe » est un équipement destiné à afficher sur un afficheur LCD des mesures de température, tension et intensité transmises sur une voie série asynchrone par un autre module (dit « mobile »).

Il signale les alarmes (valeurs des mesures en dehors d'un intervalle) sur l'afficheur LCD, pas de réception des données du module « mobile ».

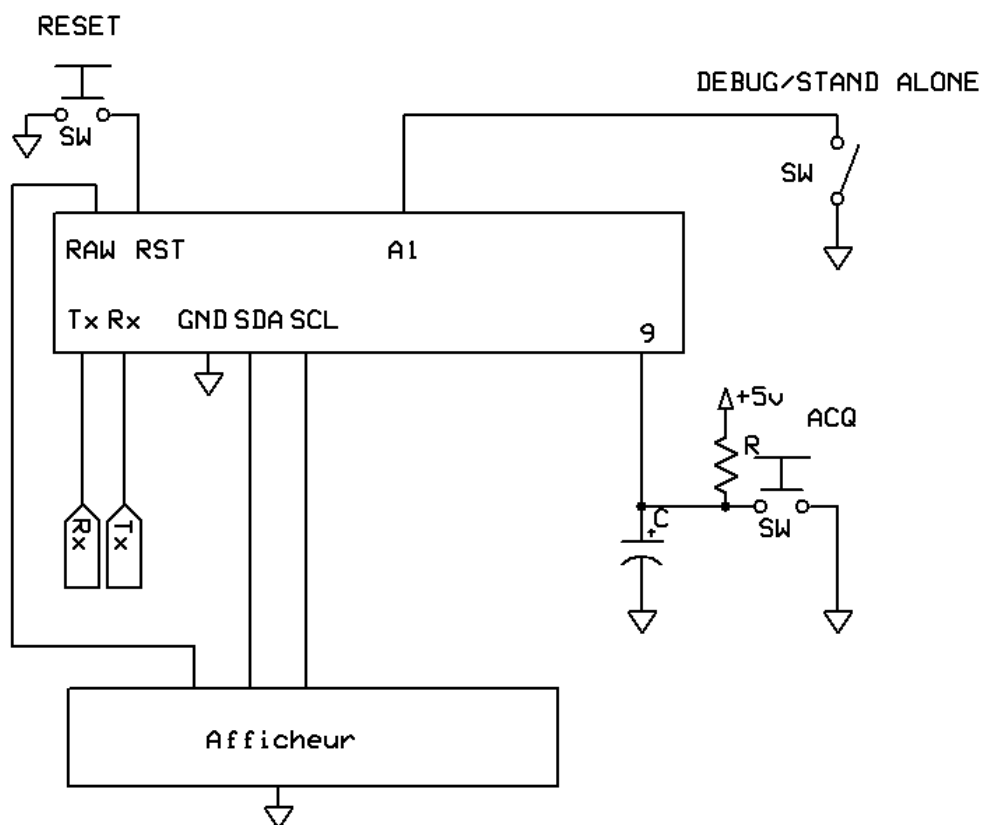
Il dispose d'un mode permettant d'afficher les valeurs des mesures en alarme (mode « Alarme »).

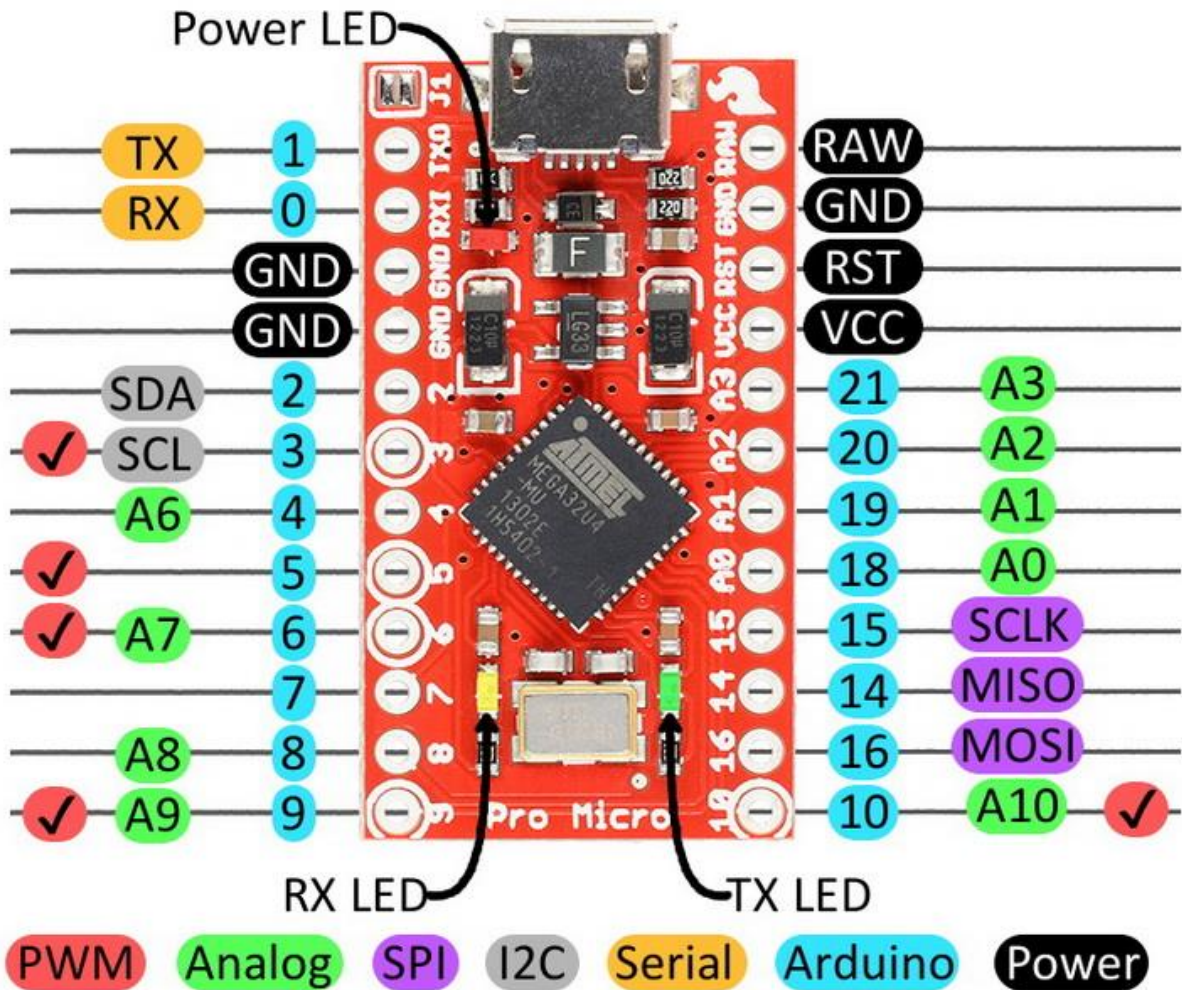
Ce module peut être connecté à une console sur PC permettant de lui envoyer des commandes et de recevoir des valeurs de mesures et autres messages.

Aspects matériels

Le module fixe est composé :

- D'une carte Arduino Pro Micro,
- D'un afficheur 2 lignes de 16 caractères chacune accessible en I²C et connecté au port SDA (pin 2), SCL (pin 3), Raw (alimentation non régulée 5V), GND (masse).
- D'un bouton poussoir nommé « ACQ » qui permet de basculer l'affichage en mode « Alarme » ou en mode « Normal ». Le fait de passer du mode « Alarme » au mode « Normal » acquitte et réinitialise les alarmes. Il est branché sur la pin 9.
- D'un interrupteur nommé « DEBUG » qui permet de passer au mode « DEBUG » (communications avec une console sur PC via l'USB) au mode « STAND ALONE » (sans communication avec une console sur PC via l'USB). Il est branché sur la pin 19.
- D'une ligne d'entrées/sorties connectée à Tx (pin 1) et Rx (pin 0) par laquelle on reçoit les données émises par le module « Mobile » et on émet des données vers le module « Mobile ».
- D'un bouton « Reset ».



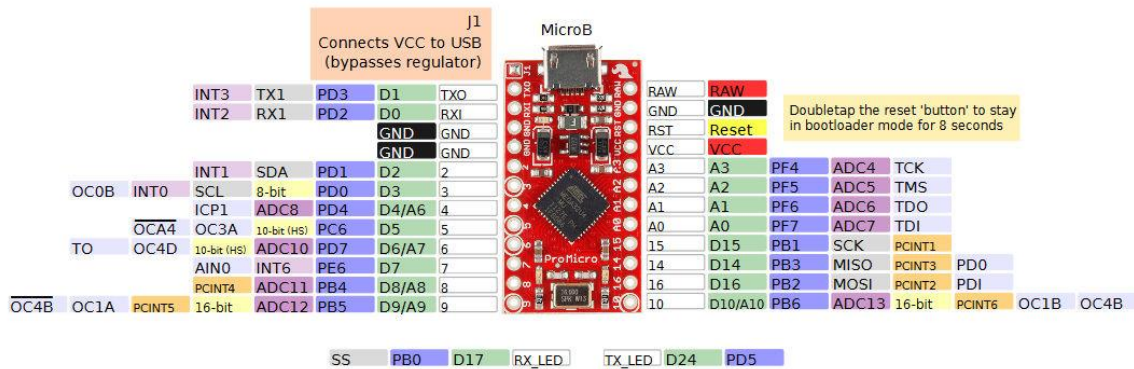


Name	ADC
Power	PWM
GND	Serial
Control	Ext Interrupt
Arduino	PC Interrupt
Port	Misc

The Arduino IDE renders all PWM pins as 8-bit

Pro Micro (Dev-12640)

16MHz/5V



Afficheur

La taille maximum affichable pour un nombre est fixée à 6 digits (point décimal compris). Si la taille de la partie entière à afficher est supérieure à 6 digits, le programme affiche « **** » quel que soit le mode (« Normal » ou « Alarme », voir ci-après).

Mode « Normal »

Dans l'exemple ci-après, les valeurs des mesures sont toutes à « 0.00 » pour la température (T), la tension (V), l'intensité (I).

Le cadran en bas à droite affiche les alarmes et un indicateur d'activité qui change toutes les secondes. Lorsqu'il n'y a pas d'alarme, l'afficheur affiche « --- ».

T=0.0	V=0.00
I=0	/----

Lorsqu'il y a une alarme, une lettre affiche la mesure concernée. Exemple s'il y a une alarme sur la température (T) et un problème de réception (R) :

T=0.0	V=0.00
I=0	/RT--

Exemple s'il y a une alarme sur la température et l'intensité :

T=0.0	V=0.00
I=0	/-T-I

Mode « Alarme »

En mode « Alarme », l'afficheur affiche les valeurs des mesures en dehors d'un intervalle ou --- pour les mesures qui ne sont pas en alarme. Dans l'exemple ci-après, la température et l'intensité sont en alarme. Il n'y a pas d'alarme sur la tension.

Le texte ALARM permet à l'utilisateur de savoir que l'affichage affiche en mode « Alarme ».

T=60.1	V=---
I=100	ALARM

Bouton « ACQ »

Un appui sur le bouton ACQ fait basculer l'affichage en mode « Alarme » s'il était en mode « Normal » ou en mode « Normal » s'il était en mode « Alarme ».

Le fait de passer du mode « Alarme » au mode « Normal » réinitialise les indicateurs d'alarme à l'état « pas d'alarme ».

Interrupteur « DEBUG »

L'interrupteur permet d'entrer ou sortir du mode « DEBUG » du programme. En mode « DEBUG », le programme accepte les commandes de la console d'un PC auquel il est connecté en USB. Il transmet les alarmes en cours et d'autres informations (voir plus loin).

Bouton « RESET »

Le bouton RESET effectue un reset du processeur. Le délai entre l'appui sur le bouton et le redémarrage du programme utilisateur présent dans la mémoire flash du microcontrôleur est assez long. En effet, l'Arduino Pro Micro comporte un chargeur de programme (Boot Loader) qui est appelé suite à un reset. C'est seulement s'il ne reçoit rien qu'au bout d'un « certain temps », il lance le programme utilisateur éventuellement présent dans la flash du microcontrôleur.

Commandes console

La carte Arduino Pro Micro peut être connectée à un PC sous Windows via sa prise USB et peut communiquer par une voie série asynchrone (émulée) avec un programme de type « console ».

Le voie série utilisée dépend de l'environnement du PC. Elle se nomme COMx où x est le numéro du port de communication attribué par le PC.

Pour que cette communication fonctionne, il est nécessaire d'avoir préalablement installé un driver de communication adapté. Sous Windows 10, ce driver est normalement pré-installé. Sinon, il faut aller le chercher sur le net (voir le site Sparkfun pour plus de détails).

Dans la suite du chapitre, {RC} représente le caractère « Retour Chariot » ou 0x0D en hexadécimal. La console utilisée doit être configurée pour transmettre un [RC] lorsque l'on appuie sur la touche entrée.

Typiquement, on utilisera la console de l'environnement de développement Arduino (« Moniteur »).

Commandes

Les commandes sont insensibles à la casse.

L'espace est un séparateur. La virgule peut aussi servir comme séparateur entre deux groupes de données. Au moins un séparateur doit être inséré dès qu'une ambiguïté est susceptible d'apparaître dans une commande. Par exemple :

- ATmin=3 : dans cette commande, l'analyseur lexical considèrera que ATmin est un identifiant (qui n'existe pas dans le cas présent).
- A Tmin= 3 : dans cette commande, l'analyseur lexical considèrera que A est un identifiant (qui existe comme commande dans le cas présent) puis Tmin est un autre identifiant (qui existe aussi). Par contre, on peut ou pas mettre un ou plusieurs séparateurs entre Tmin et = car = n'est pas un caractère accepté pour un identifiant.

Pour les valeurs réelles, le point est utilisé pour séparer la partie entière de la partie décimale (exemple : 10.30)

Les signes sont acceptés pour les valeurs réelles (+ ou -).

Les débordements pour les valeurs réelles ou entières ne sont pas gérés.

En cas d'erreur, l'analyseur le signal en affichant un marqueur à l'endroit où il a arrêté son analyse.

Exemple :

ATmin = 3

+-----erreur de syntaxe

Affichage des intervalles de mesures normales

A{RC}

Affiche les intervalles de mesure en dehors desquels une alarme est déclenchée. Exemple :

A T=[10.00, 20.00], V=[3.30, 5.00], I=[10.00, 20.00]

Les intervalles de mesures normales sont ici :

- Température : entre 10°C et 20°C
- Tension : entre 3.30V et 5.00V
- Intensité : entre 10mA et 20mA

Initialisation des intervalles de mesures normales

À [Température min] [,] Température max] [,] Tension min] [,] Tension max] [,] Intensité min] [,] Intensité max]{RC}

- Température min : TMin = <valeur>
- Température max : TMax = <valeur>
- Tension min : VMin = <valeur>
- Tension max : VMax = <valeur>
- Intensité min : IMin = <valeur>
- Intensité max : IMax = <valeur>

Exemple :

A TMin = 10, VMax = 12{RC}

Assigne 10 à Tmin et 12 à VMax, les autres valeurs sont inchangées.

Affichage des facteurs de conversion

F{RC}

Affiche la valeur des facteurs qui multipliés aux mesures correspondantes lues dans les convertisseurs analogiques numériques du module « Mobile » donnent les valeurs réelles des mesures. Exemple :

F T=0.123, V=2.0, I=0.1

Initialisation des facteurs de conversion

F [Facteur Température] [,] Facteur Tension] [,] Facteur Intensité] {RC}

- Facteur Température : T = <valeur>
- Facteur Tension : V = <valeur>
- Facteur Intensité : I = <valeur>

Exemple :

F T = 0.2, I = 1.1{RC}

Assigne 0.2 au facteur de conversion pour la température et 1.1 au facteur de conversion pour le courant.

Calcul d'un facteur de conversion F :

- La tension de référence pour le convertisseur analogique numérique (ADC) est de 2.56 volts.
- La résolution du convertisseur est 0x3FF soit 1023 en décimal.
- La formule générale pour la conversion est :

$$\text{Résolution du convertisseur} / \text{Tension de référence} = \text{Valeur lue} / \text{Tension mesurée}$$

Donc

$$\text{Tension mesurée} = (\text{Tension de référence} \times \text{valeur lue}) / \text{Résolution du convertisseur}$$

Supposons que la tension maximum que l'on souhaite lire soit 10V. Un diviseur de tension devra amener cette tension à 2.56V. Le facteur de conversion F est l'inverse de la valeur du diviseur. Ici 3.90625.

Finalement, la valeur affichée sera donc la tension mesurée multipliée par le facteur de conversion.

Dans cette hypothèse, si on lit 1023 sur l'ADC, la valeur affichée sera :

$$V = (F \times \text{tension de référence} \times \text{valeur lue}) / \text{résolution du convertisseur}$$

$$V = 3.90625 \times 2.56 \times 1023 / 1023 = 10 \text{ (Ouf on a eu chaud).}$$

Si on lit 511 sur l'ADC, la valeur affichée sera :

$$V = 3.90625 \times 2.56 \times 511 / 1023 = 4.99$$

Acquittement alarme

ACQ{RC}

Commute l'afficheur en mode « alarme ». Équivalent à un premier appui sur le bouton « ACQ ».

Reset alarme

RESET{RC}

Commute l'afficheur en mode « normal » et reset les alarmes. Équivalent à un second appui sur le bouton « ACQ ».

Mode Interrogation

Le module mobile peut, soit envoyer spontanément ses données toutes les secondes (mode dit « BasseConsommation », voir la description du module mobile pour plus de détails), soit les envoyer lorsqu'il reçoit une demande de la part du module fixe (commande « ENQ »).

I OFF{RC}

Sort du mode interrogation. Le mobile doit envoyer ses données toutes les secondes.

I ON [<période en seconde >]{RC}

Entre dans le mode interrogation. Par défaut, la période est de 1 seconde. On peut préciser une autre période qui doit être supérieur à 0.1 et inférieure à 10.

On notera que la valeur réelle de la période peut être plus grande que celle indiquée : elle dépend de la charge des processeurs des modules fixe et mobile.

Exemple :

I ON 0.2

Entre dans le mode interrogation avec une période de 0.2 secondes.

I {RC}

Affiche le mode en cours et la période. Exemple :

I ON 2

Signifie que le module fixe est en mode interrogation avec une période de 0.2s.

I OFF

Signifie que le module fixe attend les émissions spontanées du module mobile.

Messages

En mode « DEBUG », la carte peut émettre spontanément des informations vers la console à laquelle elle est raccordée.

DEBUG on

Indique que l'interrupteur « DEBUG » vient d'être commuté sur « DEBUG ».

DEBUG off

Indique que l'interrupteur « DEBUG » vient d'être commuté sur « STAND ALONE ».

Alarme <mesure> = <valeur>

Signale que la mesure <mesure> (T, V ou I pour Température, Tension ou Intensité) est en alarme et affiche la valeur de la mesure.

Sauvegarde des paramètres

Le programme sauvegarde en EEPROM la valeur des derniers paramètres entrés. Les valeurs par défaut écrites lors de la première exécution du programme sont :

FT		VMin		Periode	1.0
FV		VMax			
FI		IMin			
TMin		Imax			
TMax		Interrogation	false		

Au démarrage, le programme charge ces valeurs en RAM et démarre dans le dernier mode utilisé (Mode Interrogation = true ou false).

Pour vérifier si des valeurs ont été écrites en EEPROM, le programme teste la présence d'une « valeur magique ». Si cette valeur est présente, l'EEPROM est supposées avoir été initialisée. Sinon, les valeurs par défaut sont écrites en EEPROM.

Il pourrait se produire que la mémoire EEPROM contienne cette valeur à la fabrication (ou que la carte a déjà été utilisée pour une autre application qui aurait stocké cette valeur). Si c'est le cas, les paramètres lus seront probablement sans signification. L'utilisateur devra entrer manuellement les bonnes valeurs de paramètres via les commandes décrites précédemment.

Protocole entre modules « fixe » et « mobile »

Le module mobile effectue des mesures de température, tension et intensité en vue de les transmettre vers le module fixe.

Une trame de mesure entre le module mobile et le module fixe a toujours la même structure :

<STX>[T<valeur de T>][V{<valeur de V>}][I<valeur de I>]<ETX>

- STX représente le caractère ASCII STX = 0x02
- ETX représente le caractère ASCII ETX = 0x03
- T, V, I sont codés en ASCII
- <valeur de T (ou V ou I)> : valeur brute entière sur 10 bits du convertisseur codées en caractères ASCII.

De son côté, le module fixe filtre tous les caractères jusqu'à rencontrer le caractère STX.

Après avoir détecté le STX, le module fixe récupère les valeurs codées de la trame qu'il stocke dans un tampon de réception. Lorsqu'il rencontre le caractère ETX, le module fixe sait qu'il a reçu toute la trame. Il convertit alors les valeurs reçues en valeurs numériques, les affiche et se remet en attente d'un nouveau caractère STX, et ainsi de suite.

Ce protocole ne dispose pas de contrôle de perte d'intégrité des données : on suppose que la communication suffisamment est peu bruitée pour que ce type d'erreur ne se produise pas.

Le débit entre la partie mobile et fixe est par défaut fixée à 76800bps.

Mode d'emploi du module « mobile »

Objet

Le module « mobile » effectue des mesures de température, tension et courant et les envoie vers le module « fixe » en asynchrone. Le circuit de donnée est fourni par l'environnement (transmission sans fil).

Aspects matériels

Le module « mobile » est composé d'une carte Arduino Pro Micro à laquelle sont raccordés des capteurs de température, tension et courant.

- Le capteur de température doit être connecté sur la pin 4,
- Le capteur de tension doit être connecté sur la pin 6,
- Le capteur de courant doit être connecté sur la pin 8,

La tension max fournie par ces capteurs ne doit pas dépasser 2560mV.

Le module mobile doit être alimenté en 5V.

La vitesse de transmission est figée et fixée par une constante. Les tests ont été faits à un débit de 76800bps (valeur par défaut).

Un bouton « RESET » est branché sur la pin RST de la carte Arduino (voir module mobile pour plus de détails).

Un bouton « Basse Consommation » est branché sur la pin 19 et permet de commuter entre un mode « Basse Consommation » et un mode « Interrogation » :

- En mode « Basse Consommation », le module mobile transmet spontanément ses mesures puis se met en veille pendant une seconde, puis transmet spontanément ses mesures, etc. Dans ce mode de fonctionnement, une liaison unidirectionnelle est suffisante.
- En mode « Interrogation », le module mobile ne se met pas en veille. Il attend un caractère ENQ (0x05) puis transmet ses mesures. Le rythme de la transmission est donc déterminé par le module fixe. Dans ce mode de fonctionnement, une liaison bidirectionnelle à l'alternat est nécessaire. Il peut se produire des collisions.

L'horloge du processeur est supposée être à 16MHz. Par défaut, le programme fonctionne à 8MHz. En changeant le define « Prescaler » du programme, il est possible de fonctionner à 16MHz ou 4MHz. Les autres valeurs ne sont pas supportées.

Mise en œuvre

- Connectez les sondes à la carte Arduino.
- Connectez Tx et Rx au circuit de données.
- Choisissez le mode de fonctionnement (« Basse Consommation » ou « Interrogation »).

- Alimenter l'ensemble sous 5V.

Considération sur le mode basse consommation

Les cartes Arduino ne sont pas vraiment prévues pour faire de la basse consommation alors même que le processeur dispose de nombreuses possibilités dans ce domaine.

Dans le cas présent, nous ne considérons que le module « Mobile » qui est le plus critique pour l'application (le module « Fixe » est supposé alimenté par le secteur, le module « Mobile » est dans un environnement où l'alimentation est une ressource critique). Sans modifications, il consomme environ 30mA lorsqu'il est alimenté sous 5V à 16MHz (par analogie avec la carte Arduino Pro Mini).

Suppression des LED

La première chose à faire est de supprimer les 3 LEDs qui sont sur la carte. Le plus propre est de les dessouder. Le gain estimé est de l'ordre de 3mA et plus lorsque la carte émet à cause des LEDs Rx et Tx.

Court-circuitage du régulateur

Le régulateur de tension peut être court-circuité (strap J1). Là encore, le gain est de l'ordre d'un peu plus de 3mA. Dans le cas d'usage de cette réalisation, la source d'alimentation n'étant pas très stable, le régulateur n'est pas court-circuité.

Fonctionnement en LowPower

En mode basse consommation, le programme se met en mode « Idle » après avoir désactivé la plupart des périphériques. L'horloge de la CPU est réduite à 125kHz. Le processeur se réveille au bout d'une seconde. LED présentes, la consommation est de l'ordre de 14mA. En mode « interrogation », la consommation est de l'ordre de 23mA.