

PROTOCOLE DE COMMUNICATION DE LA CARTE ACQUISITION STM32

Présentation

Ce document fait partie du projet Ordinateur de Bord pour Voiture de Collection. Il Décrit le protocole de communication de la carte Acquisition STM32.

Version document : 1.0

Version programme : 2.0

Contact

Pour me contacter :

- J ZEHNNE : [jihzed@gmail.com](mailto:jhzed@gmail.com)
- [Page de la carte](#)
- [FaceBook](#)

Table des matières

Protocole de communication de la carte Acquisition STM32	1
Présentation.....	1
Contact.....	1
Table des matières	1
Généralités	2
Configuration :	2
interprétation de l'état des LEDs:	2
COMMUNIQUATION via l'UART	2
Généralités :.....	2
Configuration par défaut de l'UART :.....	2
Composition des trames UART	2
Liste des commandes UART	2
Action 0 : [0][voies][Sous Action][param]:	2
Action 10 : [10][voies][Sous Action][param]	2
Action 20 : [20][voies][Sous Action][param optionel]	3
Action 30 : [30][voies][Sous Action][param]	3
Action 100 : [100][voies][Sous Action][param]	3
Action 110 : [110][voies][Sous Action][param]	4
Action 120 : [120][voies][Sous Action][param]	4
Action 200 : [200][répète][Action][voies][Sous Action][param].....	4
COMMUNIQUATION via LE BUS CAN.....	5
Généralités :.....	5
Configuration du BUS CAN :.....	5
Liste des commandes (1^{er} type)	5
lecture des 8 entrées tout ou rien	5
lecture des 4 entrées compteurs (1 par adresse).....	5
configuration du temps d'inhibition des compteurs en ms.....	5
lecture des 6 entrées analogique (1 par adresse).....	6
commande des 4 sorties tout ou rien.....	6
commande des 4 sorties PWM (1 par adresse)	6
Liste des commandes UART (2nd type)	6

GENERALITES

Configuration :

Logiciel de la carte : version 2.x

Version matériel : Carte acquisition CAN A.0, A.1,...

ce document version 1.0

interprétation de l'état des LEDs:

clignotant vert 1Hz : démarrage normal

clignotant rouge 1Hz : reçue trame UART ou CAN mauvaise ou non comprise

clignotant rouge 5Hz : erreur interne au programme

scintillement vert : communication en cours

scintillement rouge : entrées compteurs en train de compter

COMMUNIQUATION VIA L'UART

Généralités :

Configuration par défaut de l'UART :

- TTL3V3, 9600 bauds, 8N1

- La vitesse peut être changée jusqu'à 115200 bauds

Composition des trames UART

Les trames UARTS sont au format ascii (chaîne de caractères) et se composent de 4 nombres séparés par un espace.

Elles se finissent par un caractère 13 (EOF) :

[Action] [voies] [Sous Action] [params] + [EOF]

Action : identifiant de la commande

voies : liste des entrée / sorties (bit 0 : entrée 0, ...)

Sous Action : indique le paramètre que l'on va écrire ou (parfois) lire.

params : liste des paramètres numériques séparés par des espaces.

-> Les paramètres à 0 sont facultatifs s'ils sont placés à la fin :

"50 01 00" = "50 01"

Les réponses sont émises sur le même format (nombres à la suite),

à l'exception de EOF qui est chr(13) suivi de chr(10).

Liste des commandes UART

Action 0 : [0][voies][Sous Action][param]:

renvoie des chaîne d'identification de la carte.

[voies] : ignore

[Sous Action] :

0 : identification de la carte

1 : contact de l'auteur (moi !)

2 : version du logiciel de la carte

[param] : ignore

note : [voies] est ignorée, par convention on le laisse à 0.

exemple :

vous envoyez "0 0 1\r" (le retour chariot ne sera plus explicité par la suite)

la carte retourne "Carte Acquisition STM32\r\n"

Action 10 : [10][voies][Sous Action][param]

Gestion des entrées tout ou rien

voies :

bit 0 : voie 0
 ...
 bit 7 : voie 7
 Sous Action : ignore
 param : ignore

Action 20 : [20][voies][Sous Action][param optionel]

lecture des entrées compteurs

voies :

bit 0 : voie 0
 bit 1 : voie 1
 bit 2 : voie 2
 bit 3 : voie 3

Sous Action :

bit 0 : demande de fréquence, réponse en 1/10Hz
 bit 1 : demande de nombre d'impulsion depuis la dernière demande (provoque la mise à 0)
 bit 2 uniquement : Fixe le temps d'inhibition des compteurs [voies]

exemples :

commande : 20 2 2 (demande fréquence compteur 2)

--> réponse : 200 (compteur 2 à 20 Hz)

commande : 20 3 2 (demande fréquence compteur 1&2)

--> réponse : 1000 200 (compteur 1 à 100Hz & compteur 2 à 20Hz)

commande : 20 2 3 (demande fréquence et nombre de ticks compteur 2)

--> réponse : 200 1000 (compteur 2 à 20Hz, 1000 impulsions depuis dernière demande)

commande : 20 15 3 (demande fréquence et ticks tous compteurs)

--> réponse 1000 2000 200 400 300 600 55 110

(compteur 1 : 100 Hz et 2000 impulsions,
 compteur 2 : 20 Hz et 400 impulsions,
 compteur 3 : 30 Hz et 600 impulsions,
 compteur 4 : 5,5 Hz et 110 impulsions)

commande : 20 15 1 10 (fixe le temps d'inhibition des 4 compteurs à 10 ms)

--> réponse : (pas de réponse à cette commande)

Action 30 : [30][voies][Sous Action][param]

Gestion des entrées ADC

voies :

bit 0 : voie 0
 ...
 bit 5 : voie 5

Sous Action : ignoré

param : ignoré

exemple :

commande 30 48 (lecture des voies 5 et 6)

--> réponse : 2000 1000 (voie 5 à 2000 LSB et voie 6 à 1000 LSB)

Action 100 : [100][voies][Sous Action][param]

configuration des la carte

Sous Action :

0 : RAZ les erreurs
 (pas de param)
 1 : active ou désactive l'echo de l'UART.
 param = 0 --> désactive ; param = 1 --> active
 2 : change la vitesse de l'UART peut être par la vitesse spécifiée /100
 param peut être : [96, 192, 384, 576, 1152] (soit 1/100 de la vitesse voulue)

3 : lit les gouttes d'adresse CAN. Le change ne prend effet qu'après un RESET
(pas de param)

param : voir ci-dessus.

exemples :

commande 100 0 1 1 : active l'écho sur l'UART

commande 100 0 3 : lit les gouttes d'adresse CAN

--> réponse 0 (si toutes les gouttes sont ouvertes)

Action 110 : [110][voies][Sous Action][param]

Gestion des sorties tout ou rien 1A

voies :

bit 0 : voie 0

bit 1 : voie 1

bit 2 : voie 2

bit 3 : voie 3

Sous Action :

0 : off

autre valeur : on

Action 120 : [120][voies][Sous Action][param]

Gestion des sorties PWM

voies :

bit 0 : voie 0

bit 1 : voie 1

bit 2 : voie 2

bit 3 : voie 3

Sous Action :

bit 0 : OFF (param = 0) ou ON (param <> 0)

bit 1 : PWM (param = pwm)

bit 2 uniquement : réglage fréquence (param = fréquence)

notes :

il est possible de démarrer et fixer la PWM en même temps (voir exemple)

la consigne PWM doit être comprise entre 1 et 10000

la consigne de fréquence est exprimée en Hz, entre 1 et 5000. La précision de

la fréquence est inférieur à 1% avant 1kHz, et mauvaise au delà

exemples :

commande : 120 15 3 5000 : démarre les 4 voies PWM et fixe à 50% leur rapport cyclique.

commande : 120 2 1 0 : arrêt de la voie 2

commande : 120 4 2 2000 : fixe le rapport cyclique de la voie 3 à 20%

commande : 120 8 1 1 : démarre la voie 4 sans changer sa valeur

commande : 120 0 4 500 : fixe la fréquence à 500 Hz

Action 200 : [200][répète][Action][voies][Sous Action][param]

auto-émission d'une trame en permanence

repete : 1 : active l'auto-émission

0 : arrête la répétition

[Action][voies][Sous Action][param] : trame telle que décrite ci-dessus

note : une seule trame peut être répétée.

La fréquence d'émission est 2 fois celle de la LED (1/2 secondes donc)

exemple :

200 1 20 15 3 : active l'affichage de tous les compteurs en auto-émission

200 0 : stop l'auto-émission

COMMUNIQUATION VIA LE BUS CAN

Généralités :

Configuration du BUS CAN :

Can 2.0b, 1Mb/s

- adresse de base configurable de 0x000 à 0x7f0 par tranche de 0x010
- la carte occupe 16 adresses à partir de l'adresse de base (+0 à +15)

Ci dessous dans la description des trames :

o : octets

w : word (16 bits) transmit sous la forme [o1 o2]

dw : double word (32 bits) transmis sous la forme [o1 o2 o3 o4]

- Lorsqu'un nombre fait plus de 1 octet (w ou dw), il est transmis de la manière :

o1 o2 o3 o4 avec o1 les bits de poids faibles, o4 les bits de poids forts.

une valeur est calculée comme ceci pour un dw :

valeur sur 4 octets = (o1) + (o2 << 8) + (o3 << 16) + (o4 << 24)

(<< décalage de x bits)

Liste des commandes (1^{er} type)

Les trames CAN peuvent être de 2 types :

La carte occupe 16 adresses mémoire à partir de l'adresse de base configurée par les gouttes d'adresse. Une requête ou une écriture à une adresse sera considérée comme un ordre précis :

- lecture des voies d'entrées de la carte (adresses 0 à 10) accessibles via des requêtes bus CAN (bit RTR activé).
- Les commandes de requête n'ont pas de syntaxe puisque pas de données transmises
- L'écriture des voies de sortie ou configuration des voie compteur est accessible via des envoie de données sur les adresses correspondantes.
- pour les exemples, la commande Linux est donnée en assumant que l'adresse de base est \$400.

lecture des 8 entrées tout ou rien

adresse base + 0 (trame de requête) :

trame de retour : [o1 o2 o3 o4 o5 o6 o7 o8]

Les octets reçus sont soit à 1, soit à 0

Les 4 premiers correspondent aux 4 entrées TOR

Les 4 derniers correspondent à l'état logique des 4 entrées compteurs (il ne faut pas en tenir compte si on les utilise en compteur !)

exemple : `cansend can0 400#r`

réponse : `400#00 00 00 00 00 00 00 00` (toutes les entrées sont à 0)

lecture des 4 entrées compteurs (1 par adresse)

adresse base + 1 à 4 (trame de requête) :

trame de retour sur 8 octets : [dw1][dw2]

dw1 : frequence en 1/10 de Hz

dw2 : nombre de ticks depuis le dernier appel

exemple : `cansend can0 401#r`

réponse : `401#20 01 00 00 50 01 00 00` (compteurs 1 : 288 Hz, 336 ticks depuis dernier appel)

configuration du temps d'inhibition des compteurs en ms

adresse base + 1 à 4 (trame de commande) :

data de la forme : [dw1]

pas de trame de retour

exemple : `cansend can0 401#e8 03 00 00` (fixe le temps d'inhibition du compteur 1 à 1000 ms)

lecture des 6 entrées analogique (1 par adresse)

adresse base + 5 à 10 (trame de requête) :

trame de retour sur 2 octets : [w1]

note : l'ADC est sur 12 bits.

commande des 4 sorties tout ou rien

adresse base + 11 (trame de commande) :

data de la forme : [o1 o2 o3 o4]

o1 : état de la sortie 1, o2 état de la sortie 2, ...

états possibles : ff ne pas changer ; 1 : activer ; 0 désactivé

pas de réponse

exemple : `cansend can0 40c#ff 01 ff 00` (active la sortie 2 et désactive la sortie 4)

commande des 4 sorties PWM (1 par adresse)

adresse base + 12 à 15 (trame de commande) :

data de la forme : [o1 dw1]

avec :

o1 : consigne à changer :

bit 0 : OFF (param = 0) ou ON (param > 0)

bit 1 : change la valeur de la PWM

dw1 = valeur de la PWM de 0 à 10000

bit 2 uniquement : réglage fréquence en Hz

dw1 = fréquence de 1 à 5000

dw1 : valeur à appliquer

pas de réponse

exemple : `cansend can0 40d#03 88 13 00 00` (active et met la sortie PWM 2 à 50%)

Liste des commandes UART (2nd type)

Cette communication permet de passer par le protocole normalement destiné à l'UART.

Une écriture à l'adresse de base d'une trame au format UART :

[adresse]#[Action][voies][Sous Action][params]

les trames sont envoyées au format binaire.

- adresse, Action, voies sont codés sur 1 octets chacun

- params est codé sur 4 octets (32 bits).

--> La trame totale fait donc 7 octets.

Voir la section destinée à l'UART pour les détails sur les trames, la réponse se fera sous forme de mots de 32 bits par lot de 8 (8 octets étant le max sur une trame CAN) (pour comprendre, envoyer les trames sur l'UART et le CAN en même temps)

notes : il n'est pas possible d'appeler la fonction 200 (autoémission)

exemples :

`400#78 0f 3 f4 01` (commande 120 15 3 500, démarre les 4 voies PWM et fixe à 50% leur rapport cyclique)

`400#20 30` (commande 30 48, lecture des voies ana 5 et 6)

réponse : `d0 07 00 00 e8 30 00 00` soit 2000 (voie 5) 1000 (voie 6)