



BS
Electronics

Manuel d'utilisation Ulogic16

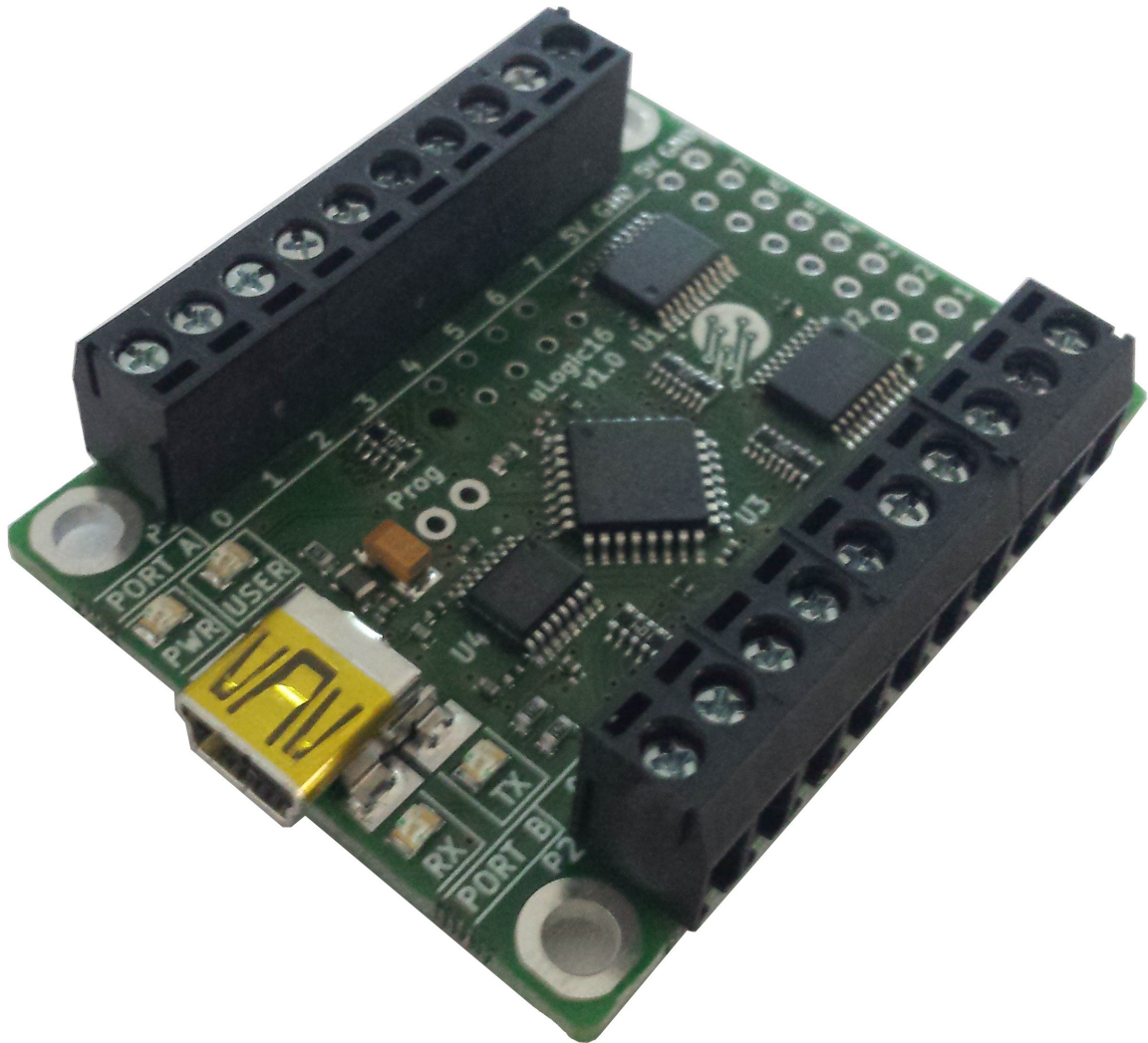




Table des matières

Manuel d'utilisation.....	1
1 Présentation.....	3
2 Mode de fonctionnement.....	4
2.1 Mode Entrée-sortie.....	4
2.1.1 Entrée.....	4
2.1.2 Sortie.....	5
2.2 Mode Logique.....	6
2.2.1 Description.....	6
2.2.2 Commandes utiles.....	6
2.2.3 Fonction logique.....	7
2.2.4 Fonctionnement interne.....	7
3 Communication.....	8
3.1 Généralités.....	8
3.2 Protocole.....	9
3.3 Listes des commandes.....	9
3.4 Prise en main.....	11
4 Caractéristiques matérielles.....	11
4.1 Généralités.....	11
4.2 Entrées/Sorties.....	12
4.3 Alimentation.....	12
5 Contacts.....	12

1 Présentation

Le système μ Logic16 est une carte d'interface permettant le déploiement rapide de fonctions logiques.

Grâce à son logiciel dédié, il est possible d'y intégrer vos fonctions logiques en quelques clics et rendre vos systèmes autonomes en se passant du développement de microcontrôleurs ou FPGA qui peuvent être longs et coûteux.

Fonctionnalités :

- 2 ports configurables en entrée ou sortie indépendamment, accessible sur un bornier.
- 8 entrées ou sorties par port.
- 2 modes de fonctionnement : exécution de schémas logiques ou entrée-sortie pilotable par PC.
- Protocole de communication simple par commande ascci compatible avec n'importe quel hyperterminal ou logiciel spécifique.
- Alimentation en 5V par le port USB ou par bornier.

2 Mode de fonctionnement

Le système comporte deux modes de fonctionnement sélectionnable par une commande sur la liaison série.

Au démarrage, le système choisi le dernier mode utilisé.

Dans le cas du mode logique, si la mémoire qui contient les informations de la fonction logique est corrompu, le système passera automatiquement sur le mode entrée-sortie.

2.1 Mode Entrée-sortie

Le mode entrée-sortie permet un accès simple a l'état des ports.

Deux ports sont disponibles : PortA, PortB.

Chacun peut être configuré indépendamment : entrée ou sortie.

Pour sélectionner ce mode, utilisez la commande "**MODE**<sp>**IO**<cr>".

2.1.1 Entrée

Configuration d'un port en entrée :

Utiliser la commande "**PORTX**<sp>**IN**<cr>".

Remplacer **X** par **A** OU **B**.

(Ex : "**PORTA**<sp>**IN**<cr>" cette commande configurera le portA en entrée).

Il existe deux types de lecture possible :

Dans les deux cas, utiliser la commande "**READ**<sp>**XXX**<cr>".

- Lecture du port entier.

Remplacer **XXX** par **PORTA** OU **PORTB**.

(Ex : "**READ**<sp>**PORTA**<cr>" cette commande renvoie une valeur entre 0 et 255, cette valeur correspond a l'état des broches du portA).

- Lecture d'un bit.

Remplacer **XXX** par **PA0** , **PA1** ... **PA7** OU **PB0** , **PB1** ... **PB7**.

(Ex : "**READ**<sp>**PA2**<cr>" cette commande renvoie 0 ou 1 qui correspond a l'état du bit PA2).

2.1.2 Sortie

Configuration d'un port en Sortie :

Utiliser la commande "**PORTX**<sp>**OUT**<cr>".

Remplacer **X** par **A** OU **B**.

(Ex : "**PORTB**<sp>**OUT**<cr>" cette commande configurera le portB en sortie).

Il existe plusieurs types d'écriture possible :

- Écriture du port entier

Utiliser la commande "**WRITE**<sp>**PORTX**<sp>**VAL**<cr>".

Remplacer **X** par **A** OU **B**.

Remplacer **VAL** par la valeur à écrire sur le port, **VAL** peut être écrit en décimal (Ex :**63**) ou en hexadécimal (ex :**0X3F**).

(Ex : "**WRITE**<sp>**PORTA**<sp>**0X3F**<cr>" cette commande écrit la valeur 63 sur le portA).

(Ex : "**WRITE**<sp>**PORTA**<sp>**63**<cr>" cette commande écrit la valeur 63 sur le portA).

- Mise à 0 d'un bit

Utiliser la commande "**CLR**<sp>**XXX**<cr>".

Remplacer **XXX** par **PA0** , **PA1** ... **PA7** OU **PB0** , **PB1** ... **PB7**.

(Ex : "**CLR**<sp>**PB3**<cr>" cette commande force le bit PB3 à 0).

- Mise à 1 d'un bit

Utiliser la commande "**SET**<sp>**XXX**<cr>".

Remplacer **XXX** par **PA0** , **PA1** ... **PA7** OU **PB0** , **PB1** ... **PB7**.

(Ex : "**CLR**<sp>**PB3**<cr>" cette commande force le bit PB3 à 0).

Il est toujours possible d'utiliser les commandes de lecture pour vérifier l'état des sorties.

2.2 Mode Logique

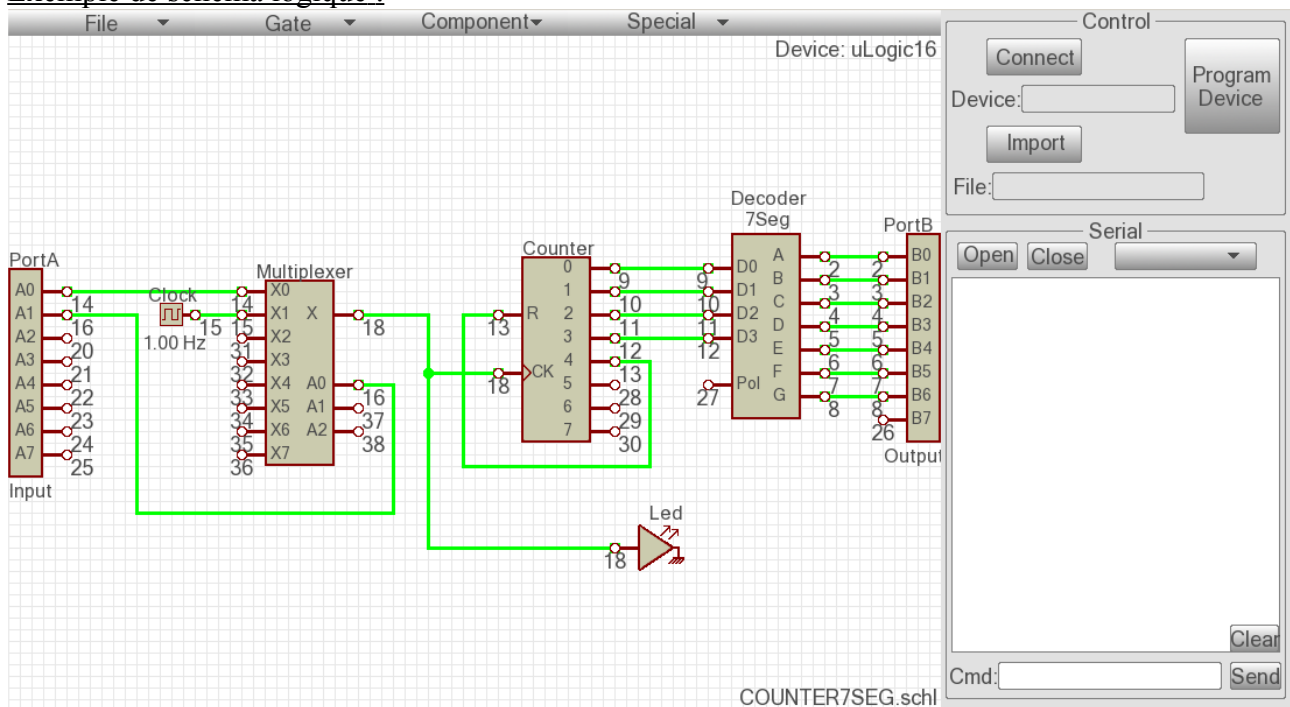
2.2.1 Description

Le mode logique permet à la carte de devenir autonome.

Les entrées et sorties sont pilotées suivant le schémas logiques definis par l'utilisateur.

Le mode logique est selectionné par default lors d'une programmation par l'utilisateur.

Exemple de schéma logique :



Sur l'exemple ci dessus, le portB est destiné à être connecté à un afficheur 7 segment.

Toujours sur cet exemple, on voit un compteur qui pilote un décodeur 7 segment.

Ce compteur est cadancé par une horloge interne ou externe que l'on peut choisir par l'intermédiaire du bit PA0 et du multiplexeur.

La led USER présente sur la carte est utilisée pour visualiser l'horloge.

2.2.2 Commandes utiles

Dans le mode de fonctionnement logique, les commandes d'écriture et de lecture sur les ports restent fonctionelles.

Cependant, écrire sur une pin qui est déjà piloté par le schéma logique n'a pas de sens et ne fonctionnera pas correctement.

Il est aussi possible d'accéder à tous les bit internes du schémas logique.

Utiliser la fonction "**READM**<sp>**XXX**<cr>".

Remplacer **XXX** par le numero écrit en dessous de la broche à analyser.

(Ex : "**READM**<sp>**18**<cr>" cette commande récupère l'état du bit d'horloge qui cadence le compteur).

En suivant la même méthode, il est possible d'écrire sur n'importe quel bit interne.

En revanche, la problématique est la même que pour les ports, écrire sur un bit qui n'est pas en l'air ne marchera pas ou génèrera un fonctionnement inattendu.

(Ex : "**CLRM**<sp>**27**<cr>" cette commande met a 0 le l'entrée Pol du décodeur).

(Ex : "**SETM**<sp>**27**<cr>" cette commande met a 1 le l'entrée Pol du décodeur).

2.2.3 Fonction logique

Liste des fonctions logiques :

- AND, OR, XOR, NAND, NOR, XNOR, NOT.
- Bascule D(Set,Reset), Bascule RS(NOR)
- Multiplexeur 8bit, Démultiplexeur 8bit
- Registre à décalage 8bit
- Mémoire tampon 8bit
- Compteur 8bit
- Décodeur Hexa → 7segment
- Monostable, Debounceur, Horloge
- Etat logique

2.2.4 Fonctionnement interne

Il est important de comprendre que la carte uLogic simule toutes les fonctions logiques.

Les fonctions logique ne sont pas présentes physiquement en comparaison avec un fpga ou la programmation permet le cablage des portes qui existe réellement en interne.

Le déroulement du traitement fonctionne en 5 étapes :

- Récupération et mise en mémoire de l'état des entrées.
- Exécution de toutes les fonctions logiques internes.
- Lecture de la mémoire pour agir sur les sorties et la led.
- Gestion de la liaison série (si besoin).
- Calcul du temps de traitement.



BS
Electronics

L'ensemble de ces tâches nécessite un temps de traitement qui peut devenir problématique pour des signaux rapides.

Le programme le plus simple (16 entrées sorties, 1 led) prendra environ 4 μ S.

Chaque fonction logique prendra entre 0,5 μ S à 3 μ S en moyenne. Ce temps dépend de la complexité de la fonction logique utilisée.

Ex : un schéma qui requiert 10 μ S de traitement ne fonctionnera pas correctement si les signaux entrants changent d'états toutes les 9 μ S.

L'utilisation de la liaison série est un processus qui prend du temps, il vaut mieux éviter de l'utiliser trop souvent pour le traitement de signaux rapides.

Il est possible de récupérer le temps de traitement avec la commande : "*READ*<sp>*TIME*<cr>".

3 Communication

3.1 Généralités

La communication s'établit via un port USB, grâce au driver USB → UART intégré sur la carte (FT230) la communication est simple, elle se résume à une simple liaison série.

La communication est possible via n'importe quel terminal ou logiciel spécifique.

Il suffit simplement d'installer le driver approprié.

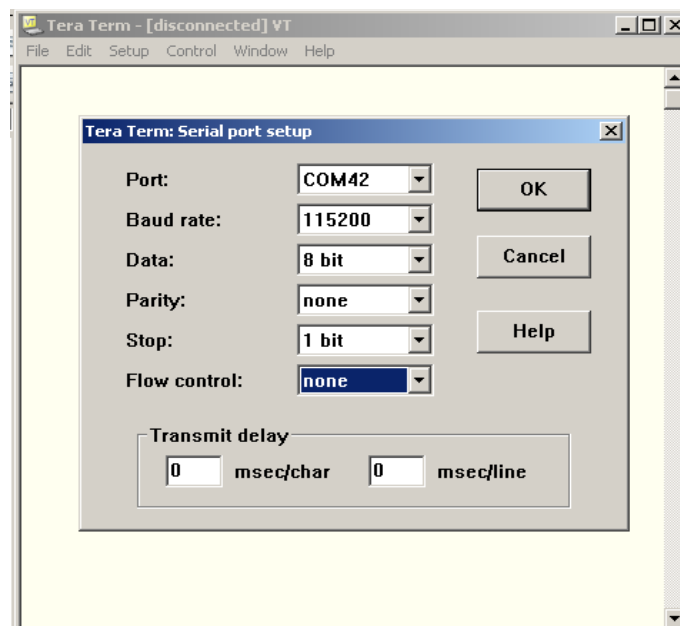
Sous windows, le driver devrait être trouvé automatiquement.

Dans le cas contraire, vous pouvez le télécharger sur le site de FTDI :
<http://www.ftdichip.com/Drivers/VCP.htm>

Configuration de la liaison série :

- 115200 Baud.
- 8 Bits de données.
- 1 Bit de stop.
- Pas de parité.
- Pas de contrôle de flux.

Exemple de configuration sous TeraTerm.



3.2 Protocole

Toutes les commandes ont la même construction et doivent être écrites en ASCII et en majuscules impérativement.

La construction est la suivante. Tout ce qui est en rouge doit être envoyé via la liaison série.

"**COMMANDE**<sp>**PARAMETRE**(optionnel)<cr>"

<sp> Correspond au caractère espace (code ASCII : 0x20).

<cr> Correspond au caractère carriage return (code ASCII : 0x0D), indique la fin de la commande. Il est souvent généré automatiquement sur le terminal lors de l'appui sur Entrée.

Lors de la réception d'une commande, la carte renvoie OK ou une valeur selon la commande.

Si la commande n'est pas correcte un message d'erreur est retourné.

3.3 Listes des commandes

Commandes de lecture :

Commande	Paramètres	Description	Réponse	Mode de fonctionnement
READ	PORTA, PORTB	Renvoie la valeur décimale du port.	Valeur entre 0 et 255. Ex : "128<cr>"	Mode logic. Mode ES.
READ	PA0...PA7, PB0...PB7	Renvoie l'état du bit choisi.	Valeur entre 0 et 1. Ex : "0<cr>"	Mode logic. Mode ES.
READ	MODE	Renvoie le mode de fonctionnement	"Mode Logic<cr>" "Mode IO<cr>"	Mode logic. Mode ES.
READ	TIME	Renvoie le temps de traitement du schéma logique.	Ex : "10,2<sp>uS<cr>"	Mode logic.
READM	XXX (valeur décimale)	Renvoie l'état du bit interne (XXX). Ex : "READM<sp>18<cr>"	Valeur entre 0 et 1. EX : "1<cr>"	Mode logic.

Commandes d'écriture :

Commande	Paramètres	Description	Réponse	Mode de fonctionnement
SETM	XXX(valeur decimale)	Met à 1 le bit interne (XXX). Ex : "SETM<sp>11<cr>"	"OK<cr>"	Mode logic.
CLRM	XXX(valeur decimale)	Met à 0 le bit interne (XXX). Ex : "CLRM<sp>14<cr>"	"OK<cr>"	Mode logic.
SET	PA0...PA7, PB0...PB7	Met à 1 le bit en paramètre. Ex : "SET<sp>PA0<cr>"	"OK<cr>"	Mode logic. Mode ES.
CLR	PA0...PA7, PB0...PB7	Met à 0 le bit en paramètre. Ex : "CLR<sp>PB2<cr>"	"OK<cr>"	Mode logic. Mode ES.
WRITE	P1 :PORTA, PORTB. P2 :VAL.	Ecrit sur le port en paramètre(P1) la valeur en paramètre(P2). Ex : "WRITE<sp>PORTA<sp>44<cr>" Ex : "WRITE<sp>PORTA<sp>0X2C<cr>" P2 peut-être en décimal ou hexadecimal	"OK<cr>"	Mode logic. Mode ES.
LEDON	N/A	Allume la led user Ex:"LEDON<cr>"	"OK<cr>"	Mode logic. Mode ES.
LEDOFF	N/A	Eteint la led user Ex:"LEDOFF<cr>"	"OK<cr>"	Mode logic. Mode ES.

Autre commandes :

Commande	Paramètres	Description	Réponse	Mode de fonctionnement
PORTA	IN, OUT	Configure la direction du portA. Ex:"PORTA<sp>IN<cr>"	"OK<cr>"	Mode ES.
PORTB	IN, OUT	Configure la direction du portB. Ex:"PORTB<sp>OUT<cr>"	"OK<cr>"	Mode ES.
MODE	LOGIC, IO	Configure la carte dans le monde souhaité. Ex"MODE<sp>LOGIC<cr>"	"OK<cr>"	Mode logic. Mode ES.
DEVICE	N/A	Retourne le type de cartes	Ex : "uLogic16<cr>"	Mode logic. Mode ES.
VERSION	N/A	Retourne la version du logiciel interne	Ex : "v1.0<cr>"	Mode logic. Mode ES.
ERASE FLASH	N/A	Efface la memoire utile au mode logique. Ex : "ERASE<sp>FLASH<cr>"	"OK<cr>"	Mode logic. Mode ES.

3.4 Prise en main

Connecter le PC à la carte uLogic.

Windows doit détecter le périphérique, il cherchera tout seul le driver de l'interface USB.

Dans le cas contraire ou si vous utilisez un autre système d'exploitation, vous pouvez le télécharger à l'adresse suivante : <http://www.ftdichip.com/Drivers/VCP.htm>

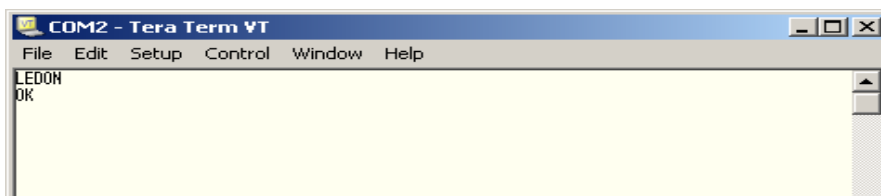
Une fois le driver correctement installé il y a deux possibilités :

- Utiliser un terminal (ex:TeraTerm).

Configurer le terminal comme expliqué dans le chapitre communication.

Une fois le port correctement configuré, entrer la commande <LEDON> puis faire entrer, les led TX et RX clignotent et la led USER s'allume.

Ci-dessous un exemple de commande.

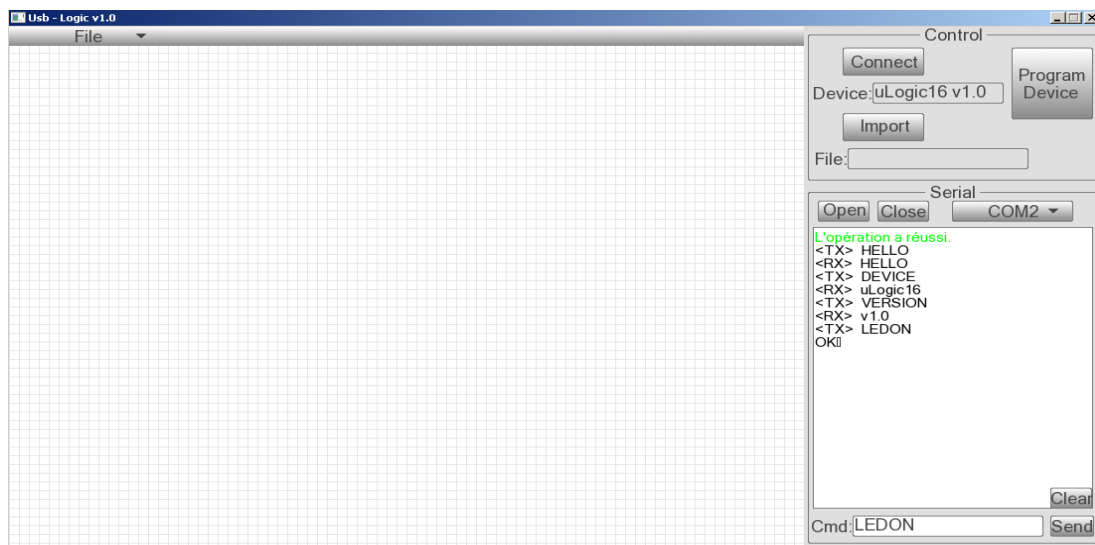


- Utiliser le logiciel LogiControl disponible à l'adresse suivante.

Cliquer sur connect, le logiciel devrait détecter automatiquement la carte uLogic, inscrire son nom et sa version de logiciel.

Entrer les commandes dans la barre en bas à droite puis cliquer sur send.

Ci-dessous, un exemple de commande.



Pour commencer l'utilisation du mode logic, télécharger la documentation du logiciel LogiControl à l'adresse suivante : <http://www.bselectronics.fr/Files/Other/MANUEL LOGICONTROL.pdf>

4 Caractéristiques matérielles

4.1 Généralités

La carte uLogic comprend :

- Un connecteur mini USB pour permettre la communication avec le PC.
- Une led indiquant l'état de l'alimentation (PWR).
- Deux led indiquant les transferts de données entre la carte et le PC (TX et RX).
- Une led utilisable directement par l'utilisateur (USER).
- Deux rangées de bornier pour accéder aux ports.
- Un microcontrôleur STM32F33.

Les plans électriques sont disponibles à l'adresse

suivante : <http://www.bselectronics.fr/Files/Other/Schematic uLogic16 v1-0.pdf>

<http://www.bselectronics.fr/Files/Other/Implantation uLogic16 v1-0.pdf>

4.2 Entrées/Sorties

La carte comporte un buffer (74ACT245MTCX) par port avec les caractéristiques suivantes.

Courant maximal sink/source	50mA
Tension d'entrée maximale	0V to VCC (5V)
Tension max niveau BAS	1,5V
Tension min niveau HAUT	2V

Pour plus d'informations, se référer à la documentation constructeur à l'adresse suivante :

4.3 Alimentation.

L'alimentation peut venir du PC grâce aux 5V du port USB.

Il est possible d'alimenter la carte grâce aux borniers.

Tension maximale d'alimentation	5,5V
---------------------------------	------



Tension minimale d'alimentation	4,5V
Consommation (hors entrées sorties)	~40mA

5 Contacts

N'hésitez pas à nous faire part de vos impressions, de vos problèmes et de vos idées d'amélioration.

Mail : bastien@sclafer.com

Site web : <http://www.bselectronics.fr/>