

	HoMIDoM	V1.0
	Driver – Arduino++ USB	

Description

Ce driver permet de communiquer avec l'interface des Arduinos via le port USB pour permettre d'envoyer des ordres et recevoir des commandes et/ou données. L'utilisation de module RF 433MHz connectés sur les Arduinos permet d'étendre la communication avec des Arduinos déportés, les commandes étant relayées par le module Arduino connecté en USB au serveur HoMIDoM.

Le driver permet aussi de s'interfacer avec des Shields connectés aux Arduinos tels que : DHT11/21/22 (Mesure de température et d'humidité), BMP085/180 (Baromètre – Thermomètre), DS1307 (RTC Horloge temps réel), Sondes One-Wire, Afficheurs LCD (16x2 20x4), Détecteurs Infrarouge, Keypad, Etc...

Interfaces physiques / virtuelles compatibles

Le driver permet de gérer autant d'Arduino qu'on le souhaite grâce à l'utilisation de module RF, de ce fait, la configuration de l'adresse des Arduinos est à configurer directement dans les composants.

Liste du matériel compatible avec ce Driver

Pour permettre de fonctionner avec ce driver, les Arduinos devront être programmés avec un Sketch spécifique mais paramétrable (Consulter le forum pour la mise à jour). Deux versions de Sketch sont disponibles : un pour l'interface Arduino maître connectée en USB au serveur, l'autre pour toutes les interfaces Arduinos périphériques n'utilisant que le module RF pour communiquer. Tous les modèles Arduinos sont compatibles (Les tests ont été validés avec les modèles Nano, Uno, Mega)

Shields validés : DHT22, BMP180, RTC(DS1307), OneWire(DS18S20) RF433(Tx : MX-05V - Rx : MX-FS-03V)

Configuration du driver

Voici la liste des caractéristiques du driver :

NOM	EDITABLE	DESCRIPTION	VALEURS POSSIBLES
Icone	Oui	Image représentant le Driver	
Nom	Non	Libellé de driver (qui sert aussi à l'affichage)	
Enable	Oui	Activation du Driver	Coché / Décoché
Description	Non	Description qui peut être le modèle du driver ou autre chose	

	HoMIDoM	V1.0
	Driver – Arduino++ USB	

StartAuto	Oui	Coché si le driver doit être activé dès le démarrage du serveur ou Décoché s'il doit être activé manuellement	Coché / Décoché
Protocole	Non	le protocole de communication utilisé par le driver	
Adresse IP	Oui	Adresse IP (facultatif) en TCP	non utilisé
Port IP	Oui	Port IP (facultatif) en TCP	non utilisé
Adresse UDP	Oui	Adresse IP (facultatif) en UDP	non utilisé
Port UDP	Oui	Port IP (facultatif) en UDP	non utilisé
Port COM	Oui	Port Com (facultatif)	
Refresh	Oui	Paramètre de rafraichissement ou de pooling (facultatif) en ms	
Modele	Oui	Modèle du driver (CM11, CM15...)	non utilisé
Version	Non	Version du driver	1.2

	HoMIDoM	V1.0
	Driver – Arduino++ USB	

Voici la liste des paramètres avancés du driver :

NOM	DESCRIPTION	VALEURS POSSIBLES
DEBUG	Activer le Debug complet, par défaut : False Active aussi un mode Debug sur les Sketchs des Arduinos (renvoi des informations via les liaisons RF et USB à HoMIDoM).	True / False

Liste des types de devices compatibles

Voici la liste des types de devices compatibles avec ce driver :

- APPAREIL
- BAROMETRE
- BATTERIE
- COMPTEUR
- DETECTEUR
- DIRECTIONVENT
- ENERGIEINSTANTANEE
- ENERGIE TOTALE
- GENERIQUEBOOLEEN
- GENERIQUESTRING
- GENERIQUEVALUE
- HUMIDITE
- LAMPE
- PLUIECOURANT
- PLUIE TOTALE
- SWITCH
- TEMPERATURE
- TEMPERATURECONSIGNE
- UV
- VITESSEVENT
- VOLET

Liste des commandes possibles

Voici la liste des types des commandes possibles avec leur description et paramètre ainsi que les devices compatibles :

- ON/OFF : Allumer/éteindre ou monter/descendre le module
 - Paramètres : aucun

	HoMIDoM	V1.0
	Driver – Arduino++ USB	

- Devices : Lampe, Appareil, Vollet, Switch, Générique Booléen...
- DIM : Variation
 - Paramètre : Pourcentage
 - Devices : Lampe
- CONFIG_TYPE_PIN : Configurer le PIN sur l'arduino suivant le modèle du composant
- PWM : Permet d'envoyer la valeur passée en paramètre (0-255) au PIN en sortie.
- SETLCD : Ecris une chaine de caractère à partir d'une position sur l'afficheur.
- SETVAR : modifier la valeur de la variable X (numéro de PIN) sur l'Arduino
- READ : Lire une entrée ou un Shield sur un Arduino
- READX : Lire toutes les valeurs en entrée, Shields y compris, sur l'Arduino et mettre à jour tous les composants associé à ce dernier (même adresse que le composant sur lequel le READX est lancé : manuellement ou via une macro)

Configuration des devices associés à ce driver

Voici la liste des caractéristiques du device à configurer pour fonctionner correctement avec ce driver :

NOM	DESCRIPTION	VALEURS POSSIBLES
Solo	pas utilisé avec ce driver	Coché / Décoché
Adresse1	Configurer l'adresse de l'interface.	0 (Arduino maitre) 1 à xxx (Arduino périphérique)
Adresse2	Numéro de PIN ou de Shield	1 / 2 ... 26 ... DHT1 / BMP1 ...
Modèle	Type de PIN ou de Shield sur l'Arduino	ANALOG_IN : Entrée analogique (1/1023) DIGITAL_IN : Entrée digitale (ON/OFF) DIGITAL_OUT : Sortie digitale (0-255) 1WIRE LCD : Action sur la LED de rétroéclairage (ON/OFF) BMPxx DHTxx CUSTOM : Envoi une commande spéciale interprétable par un Arduino
Refresh	Pour aller interroger l'arduino toutes les x secondes	

	HoMIDoM	V1.0
	Driver – Arduino++ USB	

Matériel compatible testé (Liste non exhaustive)

- Arduino (Version originale ou non) ou dérivés :

- Nano (R3)



- Uno (R3)



- Mega (R3)



- ATTiny85 (nécessite une librairie spécifique)



- Transmetteur RF 433MHz

- Emetteur (TX) : MX-05V *



- Récepteur (RX) : MX-FS-03V *



* Ces 2 modules sont en général vendus par paire pour moins d'un Euro (Provenance Chine port compris)

Nota : D'autres transmetteurs, moins répandus, peuvent être utilisés mais étant plus gourmands en terme de ressource mémoire des Arduino (librairie moins « light » que pour le modèle testé donc Sketch plus gros).

	HoMIDoM	V1.0
	Driver – Arduino++ USB	

- Shields et autres périphériques spécifiques :
 - DHT22 (Capteur de température/humidité)



- BMP180 (Capteur de pression atmosphérique/température)



- Ecran LCD 16x2 ou 20x4 (Contrôleur à base de HD44780)



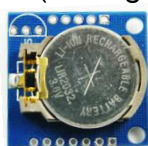
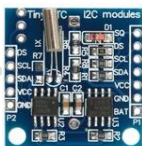
- Clavier 4x4



- DS18B20 (Sonde One-Wire)



- RTC DS1307 (Horloge temps réel)



- PIR (Détecteur Infrarouge)



	HoMIDoM	V1.0
	Driver – Arduino++ USB	

- Etc...
(Détecteur de flamme, de proximité (et mesure de distance à ultrason), de fumée, de gaz, de lumière, de son, d'obstacle infrarouge, de niveau d'eau, accéléromètre, etc...) [Voir lien.](#)

Sketchs Arduino

A Venir ;-)

Messages affichés dans les logs

Voici une liste non exhaustive de logs possibles pour ce driver

- Xx
- Xx