

Installation de la carte CAPAX-Xtd®

Plan

- ✿ Présentation de la carte CAPAX-Xtd®
- ✿ Caractéristiques générales de la carte CAPAX-Xtd®
- ✿ Correspondance des capteurs-actionneurs sur la carte ARDUINO MEGA.
- ✿ Installation de l'IDE ARDUINO.
- ✿ Installation de la carte CAPAX-Xtd® avec l'IDE ARDUINO.
- ✿ Installation des bibliothèques additionnelles : <LiquidCrystal>, <Keypad> et <dht>.
- ✿ Utilisation de la bibliothèque <SPI>
- ✿ Utilisation de la bibliothèque <LiquidCrystal>
- ✿ Utilisation de la bibliothèque <Keypad>
- ✿ Utilisation de la bibliothèque <dht>
- ✿ Installation de la carte CAPAX-Xtd® avec Matlab (configuration des fichiers Simulink).

1. Présentation de la carte CAPAX-Xtd®

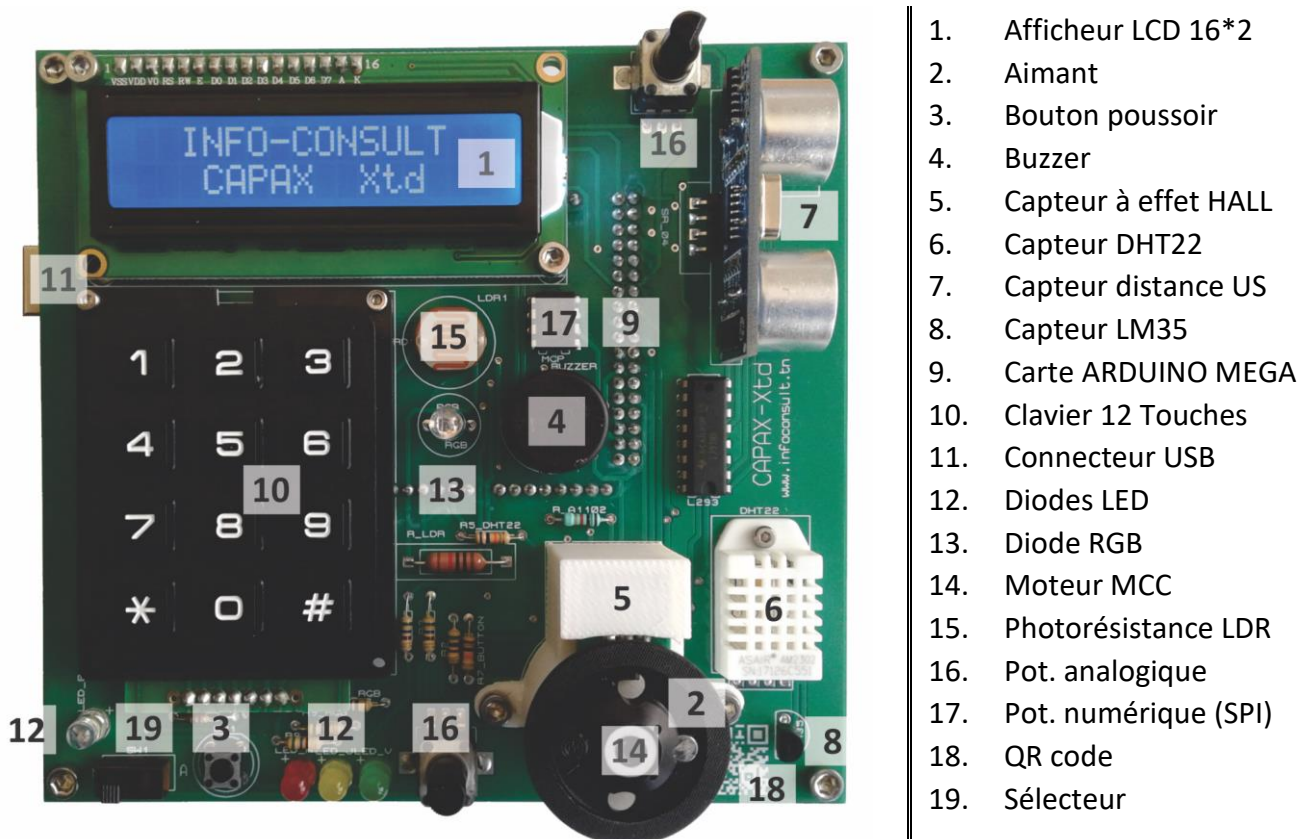


Fig.1 : Nomenclature des composants de la carte CAPAX-Xtd®

2. Caractéristiques générales de la carte CAPAX-Xtd®

☀ **Carte fille s'insérant** sur une carte mère **ARDUINO MEGA**.

☀ Capteurs

- ** **Analogique** : luminosité, température, potentiomètre analogique.
- ** **Numérique** : distance, température, humidité, vitesse de rotation, TOR, clavier.

☀ Actionneurs

- ** **Analogique (PWM)** : Moteur à courant continu MCC, Buzzer, LED simple, LED RGB.
- ** **Numérique** : LED simple, afficheur LCD, potentiomètre numérique.

☀ Environnements de développement

- ** **Natif** : IDE ARDUINO.
- ** **Autre** : Matlab, LabVIEW, Scratch.

☀ Accompagnement (via www.infoconsult.tn)

- ** **Prise en main** : Introduction à l'utilisation de l'IDE ARDUINO.
- ** **Niveau débutant** : des exercices pour la lecture/commande d'une entrée/sortie analogique ou numérique.
- ** **Niveau expert** : des exercices combinant entrées-sorties et génie logiciel pour être dans le cœur des systèmes embarqués.
- ** **Niveau académique : 6 TP (fascicule étudiant et enseignant)** traitant des thèmes capteur, actionneur, commande et réglage, niveau technicien, licence et ingénieur.

Le tableau suivant résume les différents capteurs et actionneurs utilisés par la carte CAPAX-Xtd® :

Type	Entrées	Sorties	Type
Analogique	Potentiomètre analogique : Tension Capteur LM35 : Température Photorésistance LDR : Luminosité	Moteur : Vitesse de rotation Buzzer : Son Diode LED Blanche : Lumière LED RGB (Rouge) : Lumière LED RGB (Vert) : Lumière LED RGB (Bleu) : Lumière	Analogique (PWM)
Numérique	Capteur DHT22 : Température et humidité Capteur Ultrason : Distance Bouton poussoir : TOR* Clavier rigide 12 T (4*3) : Touches	LED Rouge : Lumière LED Jaune : Lumière LED Verte : Lumière	Numérique
Interruption	Capteur à effet Hall : TOR*	Afficheur LCD (16*2) : Affichage Potentiomètre numérique (SPI) : Résistance	Numérique

***TOR** : Tout Ou Rien

3. Correspondance des capteurs-actionneurs sur la carte ARDUINO MEGA

Réf. carte CAPAX-Xtd®	Légende	Broche (Arduino)	E/S	Type
16	Potentiomètre analogique	A4	Entrée	Analogique
14	Photorésistance LDR	A3	Entrée	Analogique
8	Capteur LM35	A15	Entrée	Analogique
6	Capteur DHT 22	A6	Entrée	Numérique
3	Bouton poussoir	D47	Entrée	Numérique
10	Clavier 12 Touches 4*3 (Colonne 1)	D37	Entrée	Numérique
	Clavier 12 Touches 4*3 (Colonne 2)	D35	Entrée	Numérique
	Clavier 12 Touches 4*3 (Colonne 3)	D41	Entrée	Numérique
	Clavier 12 Touches 4*3 (Ligne 1)	D36	Entrée	Numérique
	Clavier 12 Touches 4*3 (Ligne 2)	D39	Entrée	Numérique
	Clavier 12 Touches 4*3 (Ligne 3)	D40	Entrée	Numérique
	Clavier 12 Touches 4*3 (Ligne 4)	D38	Entrée	Numérique
5	Capteur à effet Hall	D18	Entrée	Numérique (INT)
12	Diode RGB (Bleu)	D11	Sortie	Numérique
	Diode RGB (Vert)	D12	Sortie	Numérique
	Diode RGB (Rouge)	D8	Sortie	Numérique
13	Diode LED Verte	D42	Sortie	Numérique
	Diode LED Jaune	D44	Sortie	Numérique
	Diode LED Rouge	D43	Sortie	Numérique
1	Afficheur LCD 16*2 (RS)	D34	Sortie	Numérique
	Afficheur LCD 16*2 (E)	D33	Sortie	Numérique
	Afficheur LCD 16*2 (D1)	D32	Sortie	Numérique
	Afficheur LCD 16*2 (D2)	D31	Sortie	Numérique
	Afficheur LCD 16*2 (D3)	D30	Sortie	Numérique
	Afficheur LCD 16*2 (D4)	D29	Sortie	Numérique
17	Potentiomètre numérique (SPI) (CS)	D53	Sortie	Numérique
	Potentiomètre numérique (SPI) (SI)	D51	Sortie	Numérique
	Potentiomètre numérique (SPI) (SCK)	D52	Sortie	Numérique
12	Diode LED blanche	D9	Sortie	Analogique
4	Buzzer	D10	Sortie	Analogique
7	Capteur de distance US (TRIG)	D23	Sortie	Numérique
	Capteur de distance US (ECHO)	D25	Entrée	Numérique
14	Moteur MCC (En) : vitesse de rotation	D2	Sortie	Analogique
	Moteur MCC (In1) : sens de rotation*	D3	Sortie	Numérique
	Moteur MCC (In2) : sens de rotation*	D5	Sortie	Numérique

* Le sens de rotation du moteur est défini selon le tableau suivant :

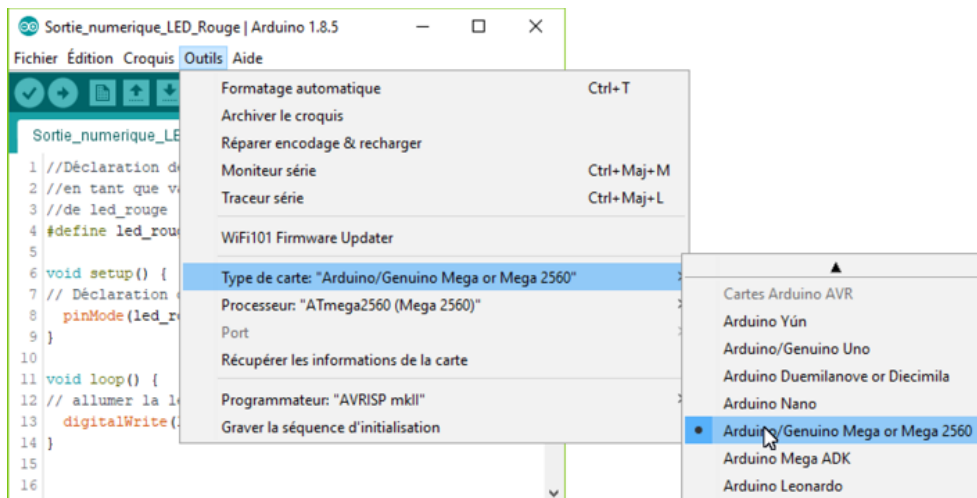
In1	In2	Sens de rotation
HIGH	LOW	Horaire
LOW	HIGH	Trigonométrique

4. Installation de l'IDE ARDUINO

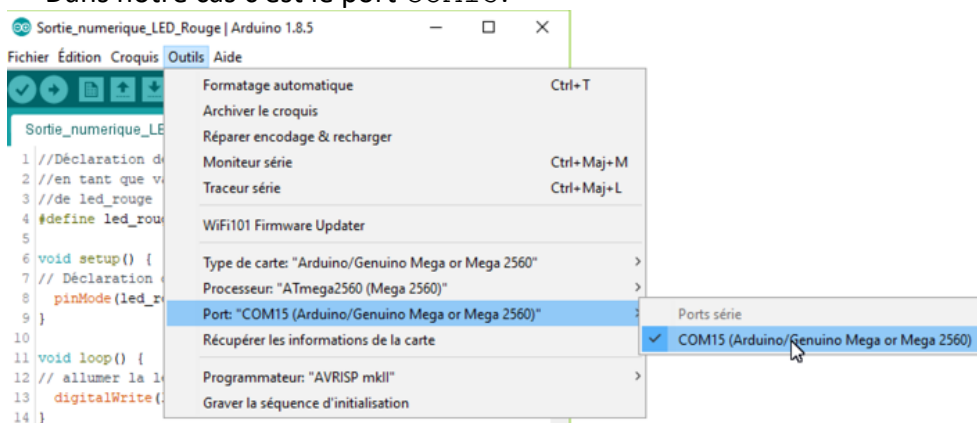
- ☀ Aller vers <https://www.arduino.cc/en/Main/Donate> et cliquer ensuite sur JUST DOWNLOAD ou CONTRIBUTE & DOWNLOAD.
- ☀ A la fin du téléchargement, décompresser le fichier arduino-1.8.5-windows.zip
- ☀ Cliquer sur le répertoire arduino-1.8.5 et créer un raccourci du programme arduino.exe sur le bureau.

5. Installation de la carte CAPAX-Xtd® avec l'IDE ARDUINO

- ☀ Par l'intermédiaire du câble USB, Brancher la carte CAPAX-Xtd® à une entrée USB du PC
- ☀ Lancer le programme ARDUINO
- ☀ Dans le menu Outils, choisir Type de carte puis Arduino/Genuino Mega or Mega 2560.



- ☀ Dans le menu Outils, choisir Port puis sélectionner le port sur lequel la carte y est branchée. Dans notre cas c'est le port COM15.



La communication entre la carte CAPAX-Xtd® et l'IDE ARDUINO est maintenant établie.

6. Installation des bibliothèques additionnelles

- ☀ A partir du site : <http://www.infoconsult.tn/fr/telechargements.html>, télécharger le fichier capax_xtd_librairies.zip. (4^{ème} ligne, cliquer sur énoncé).
- ☀ Décompresser le fichier téléchargé et ensuite copier les répertoires <DHT>, <Keypad> et <LiquidCrystal> sous le répertoire <arduino-1.8.5\librairies>.

7. Utilisation de la bibliothèque <SPI . h>, circuit MCP41010.

```
#include <SPI.h> // Ajout de la bibliothèque
...
# define CS 53
...
}
void setup() {
...
pinMode(CS,OUTPUT);
...
SPI.begin(); // Initialisation de la bibliothèque
MCP41010Write(128); // initialisation du Pot. Numérique par une valeur par défaut.
...
}

void MCP41010Write(int vMCP41010) {
digitalWrite(CS,LOW); // sélectionner le circuit MCP41010
SPI.transfer(B00010001); // Pour la configuration du mot de commande, p. 18 datasheet du CI MCP41010
SPI.transfer(vMCP41010); // Ecriture de la valeur correspondante à la résistance de sortie.
digitalWrite(CS,HIGH); // de-sélectionner le circuit MCP41010
}
```

Zone déclaration des variables.

Remarque : la bibliothèque <SPI> est fournie en standard avec l'IDE Arduino.

8. Utilisation de la bibliothèque <LiquidCrystal . h>, écran LCD

```
#include <LiquidCrystal.h> // Ajout de la bibliothèque
...
LiquidCrystal lcd(34,33,32,31,30,29);
...
}
void setup() {
...
lcd.begin(16, 2); // initialisation de l'écran LCD 16 colonnes et 2 lignes
...
}

void loop() {
...
lcd.setCursor(N°col, N° lig); // pour positionner le curseur
lcd.print("String"); // écriture d'un chaîne de caractère
lcd.print(var); // écriture d'une variable
...
}
```

Zone déclaration des variables.

9. Utilisation de la bibliothèque <Keypad . h>, clavier

```
#include <Keypad.h> // Ajout de la bibliothèque
...
const byte LIGNES = 4;
const byte COLONNES = 3;
char touches[LIGNES][COLONNES] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte BrochesLignes[LIGNES] = {36, 39, 40, 38 };
byte BrochesColonnes[COLONNES] = {37, 35, 41};
Keypad clavier=Keypad(makeKeymap(touches),BrochesLignes,BrochesColonnes,LIGNES,COLONNES);
char InputKey; // variable pour la lecture d'un caractère du clavier
...

void loop() {
...
InputKey = clavier.getKey(); // lecture d'un caractère du clavier
...
}
```

Zone déclaration
des variables.

10. Utilisation de la bibliothèque <dht.h>, capteur DHT22

```
#include <dht.h> // Ajout de la bibliothèque dht
...
# define DHT22_pin A6
float chkDHT22; // initialisation de la bibliothèque <dht>
float temp_DHT22; // lecture de la temperature [°C]
float hum_DHT22 // lecture de l'humidité [%]
...

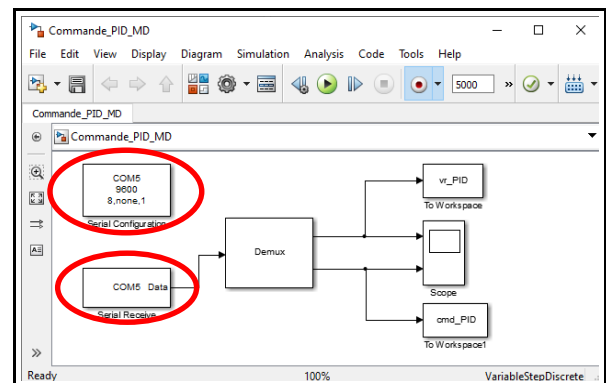
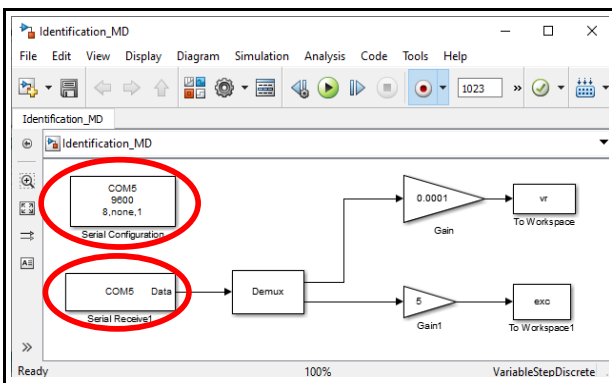
void loop {
...
  chkDHT22 = DHT.read22(DHT22_pin); // initialisation de la bibliothèque <dht>
  temp_DHT22 = DHT.temperature; // Température [°C]
  hum_DHT22 = DHT.humidity; // Humidité [%]
...
}
```

Zone déclaration des variables.

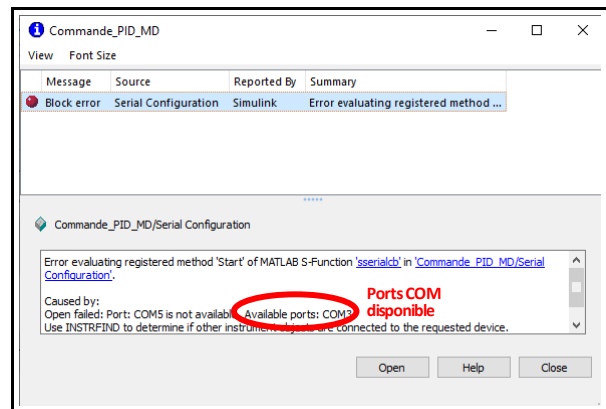
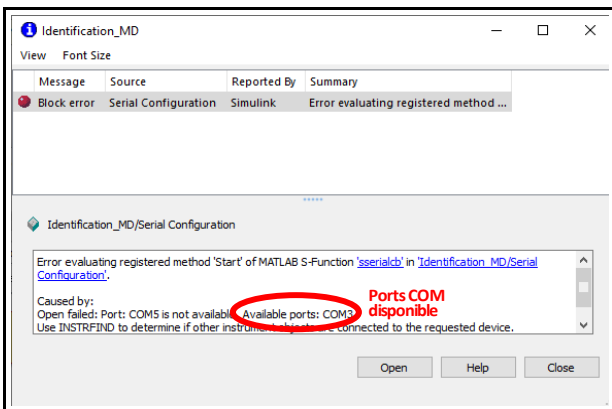
11. Installation de la carte CAPAX-Xtd® avec Matlab (configuration des fichiers Simulink).

Avertissement : Brancher la carte CAPAX-Xtd® avant de démarrer Matlab, sinon la carte ne sera pas reconnue.

Avant de démarrer une opération d'identification (identification.slx) ou de commande PID (Commande_PID.slx), il faut s'assurer que le port COM est correctement configuré pour les blocs SerialConfiguration et SerialReceive.



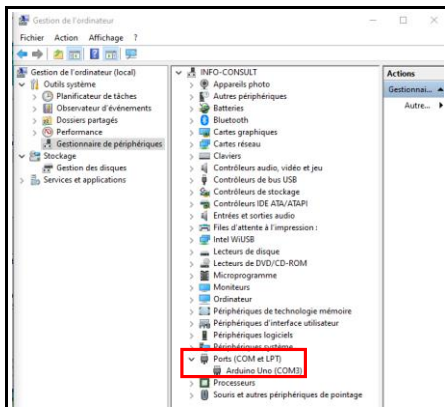
Si le port COM n'est pas correctement configuré, en cliquant sur le bouton vert , l'écran suivant s'affiche :



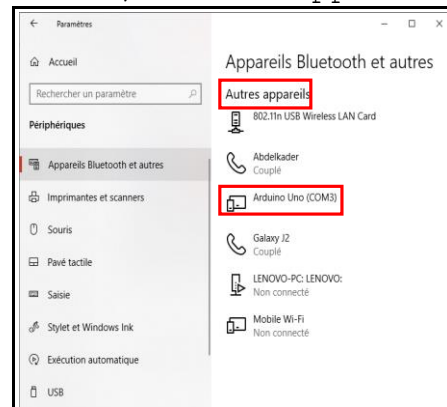
Pour obtenir l'adresse du port COM, il faut consulter soit l'IDE ARDUINO, menu outil puis port (voir au-dessus §5), soit

- ✿ le gestionnaire de périphérique
- ✿ menu Windows, Démarrer, Paramètres, Périphériques, appareils Bluetooth et autres, Autres appareils

Le gestionnaire de périphérique

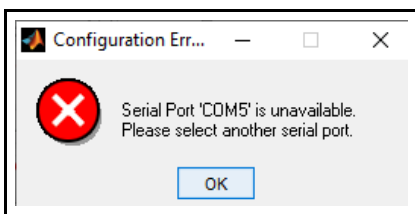


Menu Windows, Démarrer, Paramètres, Périphériques, appareils Bluetooth et autres, Autres appareils

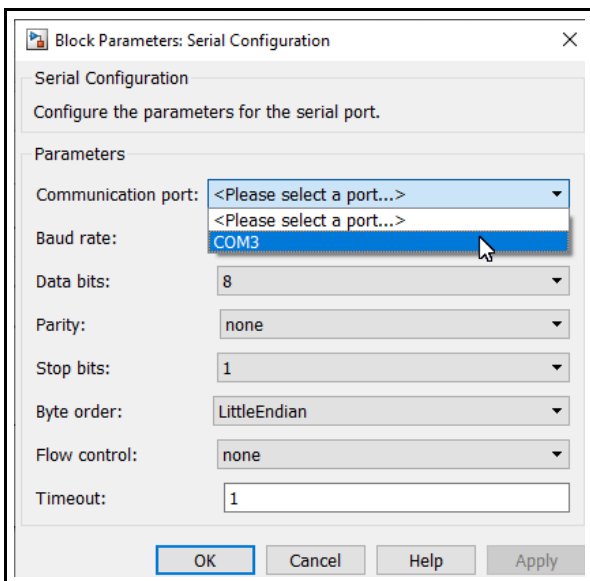


Configuration du port COM sur les fichiers Identification.slx et Commande_PID.slx

Double cliquer sur le bloc SerialConfiguration, l'écran suivant s'affiche :



Cliquer ensuite sur Ok et choisir le port COM adéquat.



Répéter la même opération avec le bloc SerialReceive des fichiers Identification.slx et Commande_PID.slx.