

[s'inscrire via RSS](#)

[ACCUEIL](#)

[BTS SN](#)

[NSI](#)

[FORMATION ISN](#)

[FORMATION SIN](#)

[FORMATION STI2D](#)

[EMBEDDED SYSTEM](#)

Connexion automatique d'une Arduino avec VisualStudio C#

Posted on 7 février 2017

Ce programme permet de piloter la LED connectée sur D13 sur une Arduino (uno, mega, ...). La connexion à la carte Arduino est automatique. Le système identifie le port sur lequel l'Arduino est connecté et tente d'ouvrir le port. L'utilisateur a la possibilité de modifier la connexion, dans le cas où plusieurs cartes Arduino seraient reliées à l'ordinateur par exemple.

Les mots de commandes disponibles sont :

- **ON** : Allumer la LED
- **OFF** : Eteindre la LED
- **STATUS** : Demande le status de la LED



Le code Arduino s'appuie sur l'utilisation de la méthode **readStringUntil()** de l'objet **Serial** :

```
ordre = Serial.readStringUntil('\n');

if(ordre=="ON")
{
    ...
}
if(ordre=="OFF")
{
    ...
}
if(ordre=="STATUS")
{
    ...
}
```

Le code C# utilise la classe **ManagementObjectSearcher** qui permet de rechercher une ressource précise au sein du système. Ici, on cherche les ports séries et leur description. On se connectera ensuite à celui dont la description contient le mot "Arduino".

- Espaces de nom à ajouter :

```
using System.Management;
using System.IO.Ports;
```

- Référence à ajouter au projet (Explorateur de solutions -> Références):

```
System.Management
```

- Au chargement du formulaire,
 - rechercher les ports séries,
 - rechercher le port de connexion de l'Arduino
 - tenter de s'y connecter
 - gérer les erreurs de connexion

```
1 private void Form1_Load(object sender, EventArgs e)
2 {
3     string arduinoLine="";
4     string arduPort="";
5     int arduinoItem = -1;
6     using (var searcher = new ManagementObjectSearcher("SELECT * FROM WIN32_SerialPort"))
7     {
8         string[] portnames = SerialPort.GetPortNames();
9         var ports = searcher.Get().Cast<ManagementBaseObject>().ToList();
10        var tlist = (from n in portnames
11                    join p in ports on n equals p["DeviceID"].ToString()
12                    select n + " - " + p["Caption"]).ToList();
13
14        foreach (string s in tlist)
```

Pages

[About](#)

[BTS SN](#)

[Enseignements](#)

[Formation PT Activity Wizard](#)

[Formation réseau](#)

[Embedded System](#)

[Les cours](#)

[Cours : OS dans l'embarqué \(2015-16\)](#)

[Cours : OS dans l'embarqué \(2016-17\)](#)

[Cours : OS dans l'embarqué \(2017-18\)](#)

[Cours : OS dans l'embarqué \(2018-2019\)](#)

[Cours : OS dans l'embarqué \(2019-2020\)](#)

[Les TP](#)

[TP : objet connecté \(2019\)](#)

[TP : OS linux embarqué \(2015-16\)](#)

[TP : OS linux embarqué \(2016-17\)](#)

[TP : OS linux embarqué \(2017-2018\)](#)

[TP : OS linux embarqué \(2018-19\)](#)

[TP : Prise en main](#)

[TP : Prise en main \(2016\)](#)

[TP : Prise en main \(2019\)](#)

[Formation ISN](#)

[Architecture des machines](#)

[Initiation aux réseaux](#)

[Mise en oeuvre réseau](#)

[Pédagogie de projet](#)

[Pédagogie de projet V2](#)

[Projet terminal d'évaluation](#)

[Robotique](#)

[Simulateur Azolla](#)

[Système d'exploitation](#)

[Formation SIN](#)

[Foxboard G20](#)

[Introduction à Linux](#)

[Programmation web](#)

[Terminal linux et](#)

[programmation](#)

[Formation STI2D](#)

[ET22A](#)

[ET22B](#)

[ET22C](#)

[ET22D](#)

[Fox board g20](#)

[Connexions](#)

[Branchements](#)

[Connexion avec MiniCom](#)

[Connexion avec](#)

[HyperTerminal](#)

[Trouver l'IP avec ipscan](#)

[Connexion ssh sous linux](#)

[Connexion ssh sous](#)

[Windows](#)

[Connexion au site web](#)

[embarqué](#)

[Transfert de fichiers](#)

[Connecter une clé USB](#)

[Le serveur web embarqué](#)

[Pages html](#)

[Pages PHP](#)

[Gestion des bases de](#)

[données](#)

[Couple php/sqlite](#)

[Projet : Contrôle Domotique](#)

[Réglage de la date et de](#)

[l'heure](#)

[Utilisez les ports GPIO](#)

[Programmation](#)

[C/C++](#)

```
15 | {
16 |     comboBox1.Items.Add(s);
17 |     if (s.Contains("Arduino"))
18 |     {
19 |         arduinoLine = s;
```

Les fichiers ici : <https://github.com/msilanus/ArduinoToVisualStudio>

Remplis sous: [Arduino](#), [Divers](#), [Electronique](#), [OS](#), [Visual Studio C#](#)

[Commentaires](#)

Commentaires (0)

Trackbacks (0)

(Souscrire aux commentaires de cet article)

Désolé, le formulaire de commentaire est fermé pour le moment

[Kinect Microsoft dans un projet Windows Form C#](#) »

« [Reconnaissance faciale par Eigenfaces](#)

[Python](#)

[Les ports séries asynchrones](#)

[En ligne de commande \(shell\)](#)

[En C](#)

[En Python](#)

[En php](#)

[NSI](#)

[Algorithmique](#)

[Architectures matérielles et](#)

[systèmes d'exploitation](#)

[Le Web](#)

[Les bases de données](#)

[Les bases de l'informatique](#)

[Représentation des données](#)

[Réseau](#)

[Packet Tracer](#)

[Formation enseignants](#)

[Virtualisation](#)

[VMware Workstation.](#)

Catégories

[Arduino](#)

[Azolla](#)

[C/C++](#)

[Cayenne](#)

[Cisco](#)

[CSS](#)

[Divers](#)

[Electronique](#)

[HTML](#)

[IOT](#)

[JavaScript](#)

[Linux](#)

[LoRa](#)

[Lua](#)

[mqtt](#)

[MySQL](#)

[Node red](#)

[NSI](#)

[OS](#)

[Packet Tracer](#)

[PHP](#)

[Python](#)

[Qt](#)

[Raspberry Pi](#)

[Réseau](#)

[Robotique](#)

[The Things Network](#)

[Visual Studio C#](#)

Liste de Liens

[ACMESystems](#)

[Doc Fox Board G20](#)

[Lextronic](#)

Archive

[novembre 2022](#)

[octobre 2021](#)

[mai 2020](#)

[février 2020](#)

[novembre 2019](#)

[juillet 2018](#)

[juin 2018](#)

- [avril 2018](#)
- [mars 2018](#)
- [février 2018](#)
- [juin 2017](#)
- [mars 2017](#)
- [février 2017](#)
- [janvier 2017](#)
- [avril 2016](#)
- [mars 2016](#)
- [décembre 2015](#)
- [octobre 2014](#)
- [janvier 2014](#)
- [novembre 2013](#)
- [mai 2013](#)
- [avril 2013](#)
- [mars 2013](#)
- [février 2013](#)
- [janvier 2013](#)
- [décembre 2012](#)
- [novembre 2012](#)
- [juin 2012](#)
- [mars 2012](#)
- [octobre 2011](#)
- [février 2011](#)
- [décembre 2010](#)

Meta

- [Connexion](#)
- [RSS](#)
- [Commentaires RSS](#)