

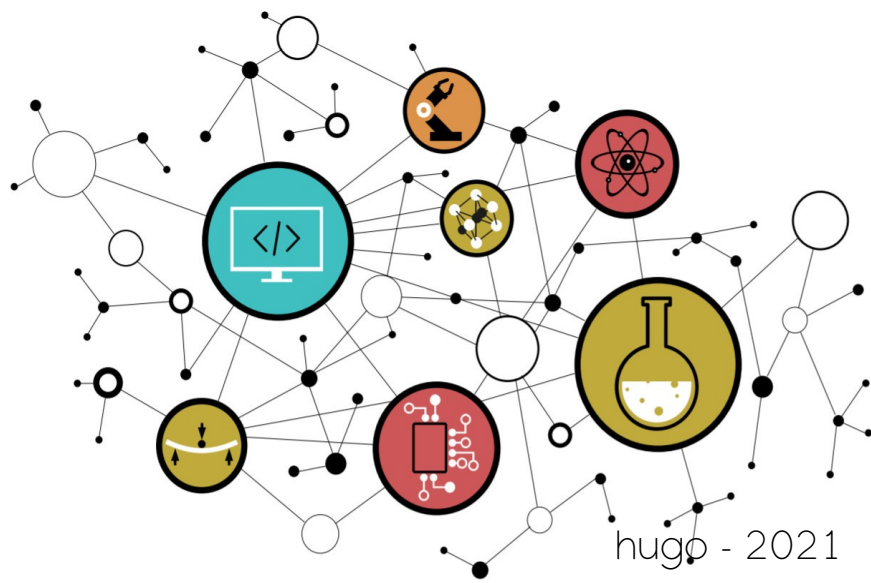
TUTORIEL

INSTALLATION DES OUTILS DE DÉVELOPPEMENT

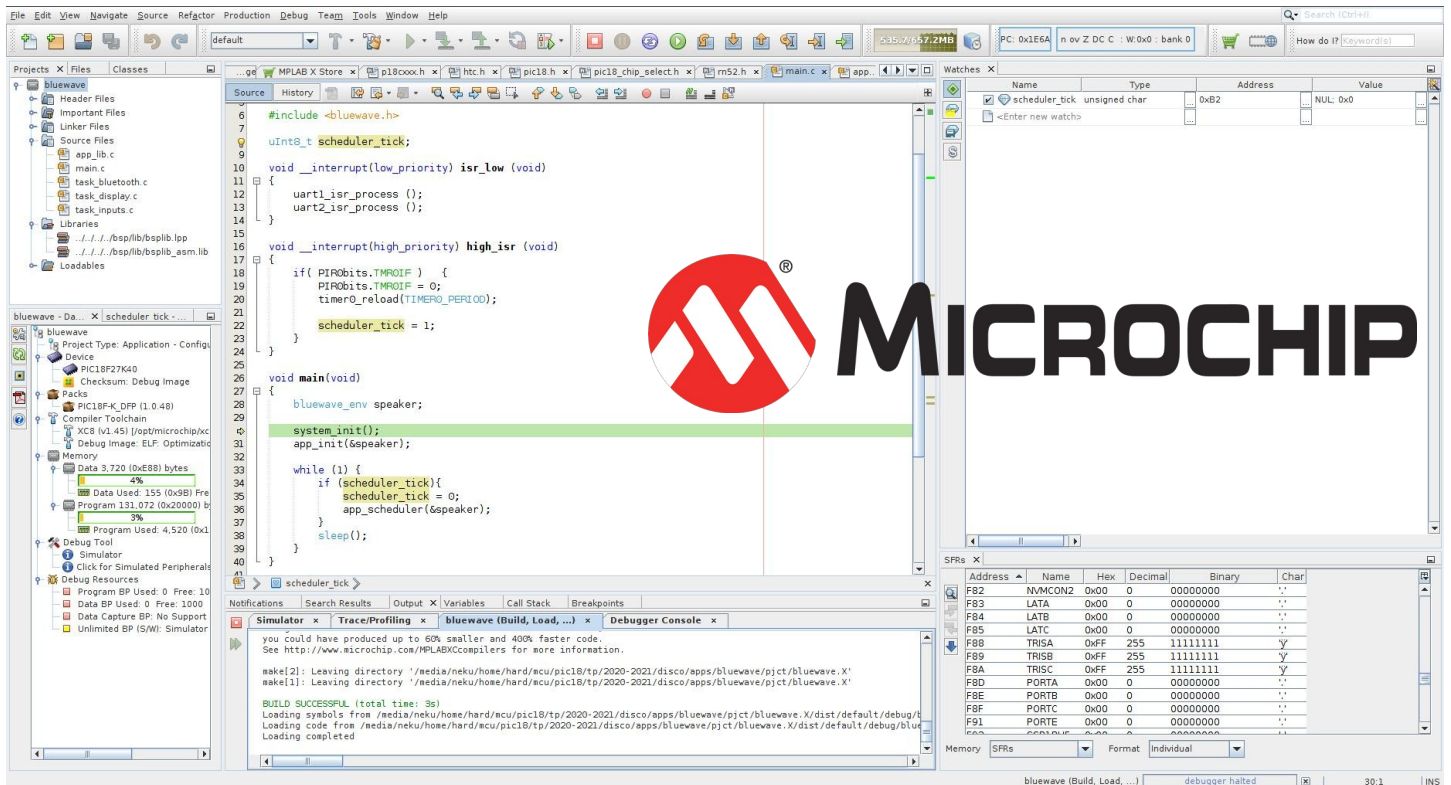
IDE MPLABX ET TOOLCHAIN XC8

TERMINAL SÉRIE TERATERM ET DRIVERS VCP

SIMULATEUR PICSIMLAB ET ÉMULATEUR NULL MODEM COM0COM



OUTILS DE DÉVELOPPEMENT LOGICIEL MICROCHIP



<https://www.microchip.com/>

La totalité de la trame de Travaux Pratiques sera réalisée sur solution logicielle et matérielle Microchip, société Américaine et technologie leader du marché des MCU 8bits avec ses gammes PIC16/PIC18 et AVR depuis le rachat de la société Atmel en 2016 (projet Arduino). Les outils de développement logiciel sont librement téléchargeables et installables depuis le site web officiel de Microchip. Seront à installer l'Environnement de Développement Intégré (ou IDE) MPLABX et la chaîne de compilation C XC8. Ces logiciels sont supportés par plusieurs plateformes Windows, MacOS et GNU/Linux. Installer les deux logiciels suivants. **Bien respecter les versions des outils** :

- IDE (Integrated Development Environment) MPLABX **v5.50**
- Toolchain C XC8 **v1.45** (Free Mode)
- Lien de téléchargement vers les archives des outils Microchip (IDE MPLABX, toolchain XC8, etc) :

<https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>

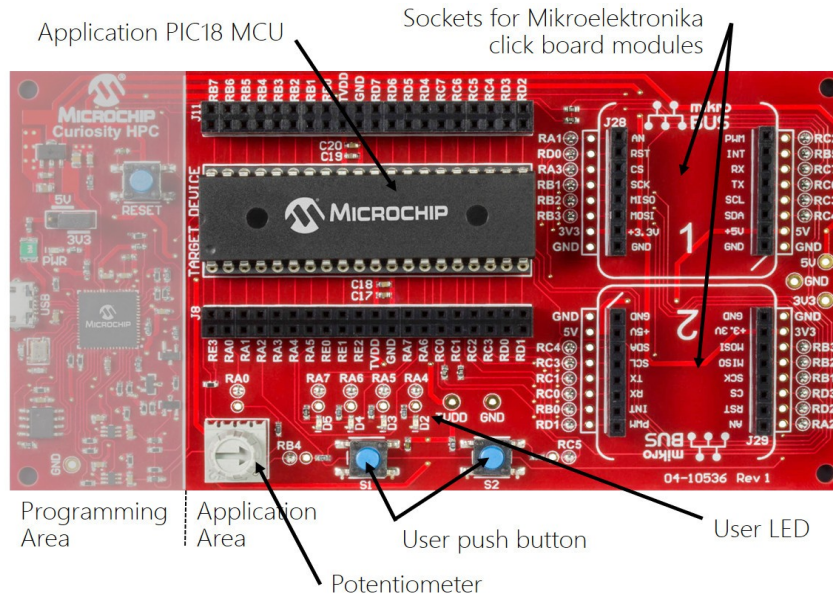
- Conseils :
 - Sous Window, garder les chemins d'installation par défaut sur le lecteur système C:/ . Ne pas installer MPLABX IPE. Vous trouverez ci-dessous l'aide officielle de Microchip :

<https://microchipdeveloper.com/mplabx:installation>

- Sous GNU/Linux, bien s'assurer d'installer les outils dans /opt/microchip/

STARTER KIT CURIOSITY HPC ET MCU PIC18F27K40

Les développements seront réalisés sur MCU 8bits PIC18F27K40 possédant 28 broches et intégrant un CPU 8bits 16 MIPS (Mega Instructions Per Second). Ce MCU offre 128Ko de mémoire programme Flash et ~3Ko de mémoire donnée SRAM. La carte de développement choisie est le starter kit Curiosity HPC de Microchip (cf. ci-dessous). Elle intègre une sonde de programmation JTAG permettant de charger les programmes développés sur ordinateur au sein de la mémoire flash interne du MCU PIC18 applicatif cible.

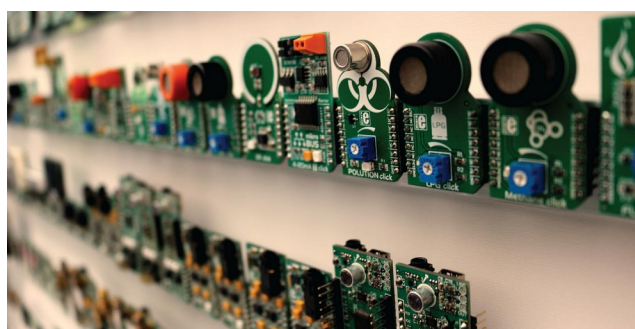


<https://microchipdeveloper.com/boards:curiosityhpc>

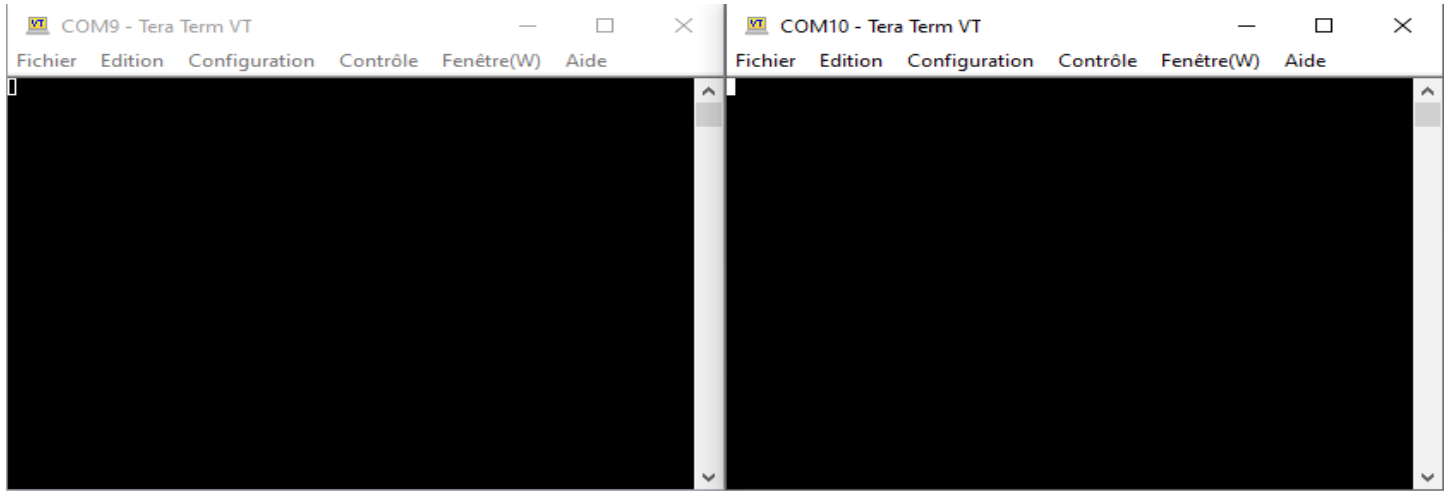
Si vous souhaitez acheter le matériel à vos frais et pour vos propres développements et projets personnels, voici le listing des composants nécessaires pour les TP et des liens vers un fournisseur de composants (autres fournisseurs possibles Farnell, mouser, digikey, etc) :

- Carte curiosity HPC : <https://fr.rs-online.com/web/p/outils-de-developpement-pour-microcontrôleurs/1240947/>
- MCU PIC18F27K40 boîtier SDIP 28 pins : <https://fr.rs-online.com/web/p/microcontrôleurs/1262151/>
- Click Boad USB to UART : <https://fr.rs-online.com/web/p/modules-de-developpement-de-communication-et-sans-fil/8828913/>
- Click Board Audio Bluetooth : <https://fr.rs-online.com/web/p/modules-de-developpement-de-communication-et-sans-fil/1360845/>

La carte Curiosity HPC possède deux connecteurs mikroBUS permettant également d'ajouter des modules externes Click Board proposés par la société Serbe Mikroelektronika (cf. projet Arduino). Des centaines de modules externes sont actuellement disponibles en catalogue (Bluetooth, audio, WIFI, contrôle de moteur, afficheurs LCD, etc) : <https://www.mikroe.com/click>

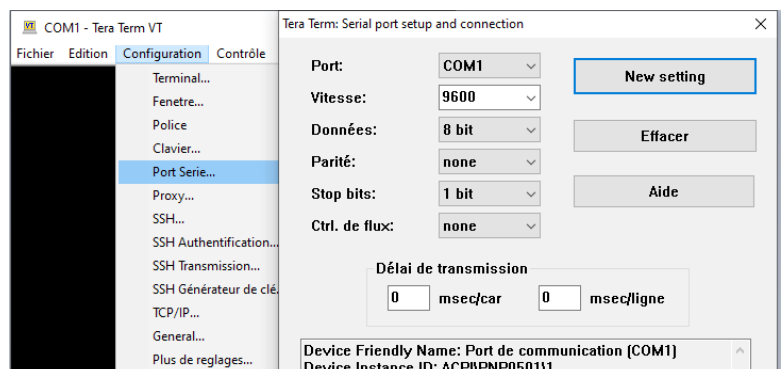
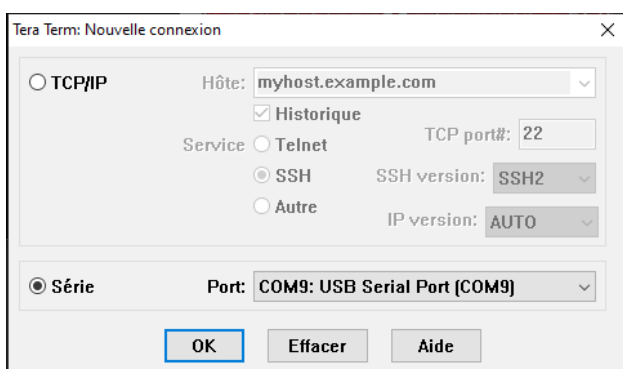


TERMINAL DE COMMUNICATION UART ET DRIVER

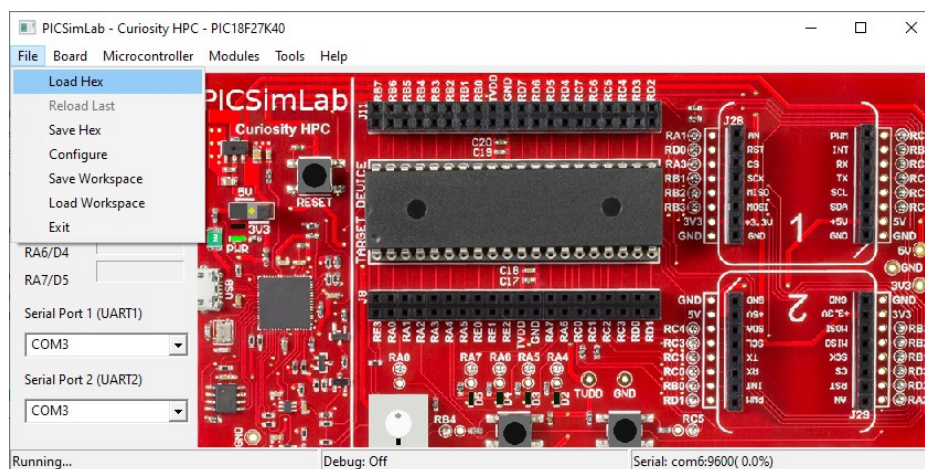


Pour celles et ceux souhaitant réaliser la trame de TP sur leur machine personnelle en présentiel ou travailler sur simulateur PICSimLab, concernant la partie UART seulement, vous aurez également besoin d'outils complémentaires. Il faudra installer les drivers pour modules USB to UART du fabricant FTDI. Ces modules sont portés dans les câbles et composants USB to UART, ou par exemple présent sur le module Click Board Mikro USB to UART. FTDI est l'un des leader du marché sur ces composants de conversion de protocole à protocole. De même, sous Windows, nous conseillons l'utilisation de TeraTerm Pro ou PuTTY comme terminal de communication avec les ports COM virtuels ou réels de l'ordinateur :

- Tera Term (dernière version) : <https://ttssh2.osdn.jp/index.html.en>
- Drivers VCP (Virtual COM Port) FTDI (dernière version) : <http://www.ftdichip.com/Drivers/VCP.htm>
- Exécuter Tera Term : *Double clic sur l'icône C:\Program Files (x86)\teraterm\termpro.exe*
- Configurer Tera Term. Les port COM et débits utilisés seront à adapter à vos besoins :



SIMULATEUR PICSIMLAB



PICSimLab est un simulateur open source développé par Luis Claudio Gambôa Lopes (<https://lcgamboa.github.io/>), un professeur d'université au Brésil. PICSimLab intègre notamment un debugger MPLABX et est apte à exécuter sur carte électronique virtuelle des fichiers binaires exécutables au format .hex générés depuis l'IDE MPLABX de Microchip. Télécharger l'archive picsimlab_win64.zip (pour Windows 64bits) ci-dessous, l'extraire puis lire le tutoriel suivant. Ces versions pré-compilées de PICSimLab n'intègrent que la carte Curiosity HPC de développement et un MCU PIC18F27K40 (solutions de TP). Le simulateur est également disponible dans l'espace moodle lié à l'enseignement (section SIMULATEUR).

- Simulateur PICSimLab pour Windows 64bits avec carte Curiosity HPC et MCU PIC18F27K40 :

<https://foad.ensicaen.fr/mod/resource/view.php?id=24874>

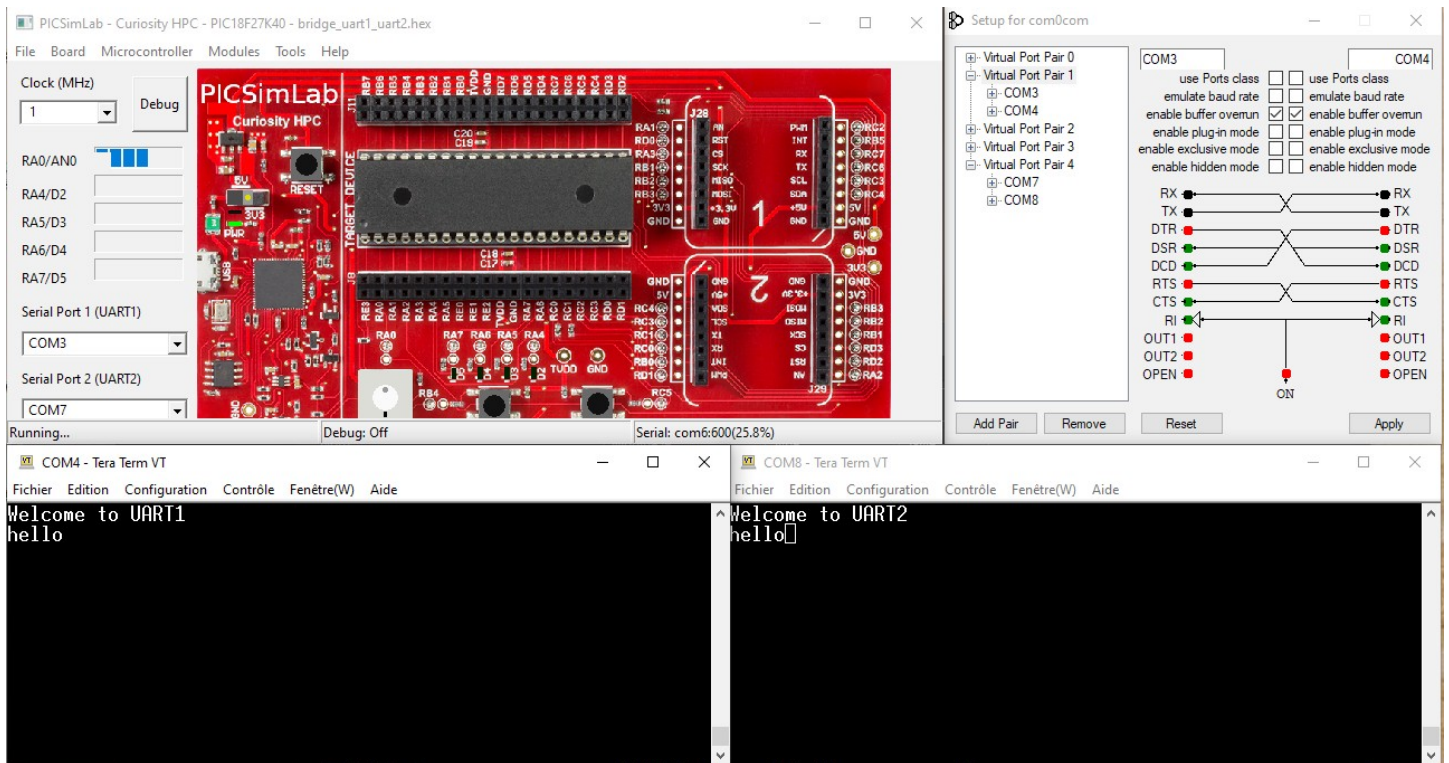
Afin d'aider les étudiants à travailler et approfondir le domaine à distance sans avoir à investir dans du matériel, nous avons lancé depuis 2020 dans le cadre de stages et projets école différents sujets permettant aujourd'hui d'offrir une solution de travail entièrement émulée de la carte de développement (Curiosity HPC et MCU PIC18F27K40) et de certains outils de développement (oscilloscopes et terminaux de communication série). Voici les noms des élèves ayant travaillé sur ce projet (~600-700h de travail) : Mohammed El-Khadraouy (stage 2A SATE) et Lacina Berthe (stage 1A ELEC) en 2020, Hassan Kanso, Valentin Kauffmann et Mounir Abdessamade (projet immersif 3A ELEC APP) en 2021

- Exécuter PICSimLab : *Double clic sur l'icône de l'application picsimlab_win64/picsimlab.exe*
- Charger un programme : File > Load Hex > fichiers de test dans `picsimlab_win64/examples/` ou `disco/apps/hex`

Bugs et limitations connus PICSimLab

- Mode debug MPLABX <-> PICSimLab avec installation de plugin sous MPLABX encore instable (fichier [com-picsim-picsimlab_mplabx5.45.nbm](https://lcgamboa.github.io/picsimlab/MPLABX_Integrated_Debug_pic.html) sous MPLABX v5.45) : https://lcgamboa.github.io/picsimlab/MPLABX_Integrated_Debug_pic.html
- Facteur 2 constaté dans les mesures des timing avec le module oscilloscope PICSimLab durant les TP Timer
- Latences constatées à l'allumage des LEDs sur la carte voire non allumage dans certaines conditions (configurer registres LATA après TRISA)

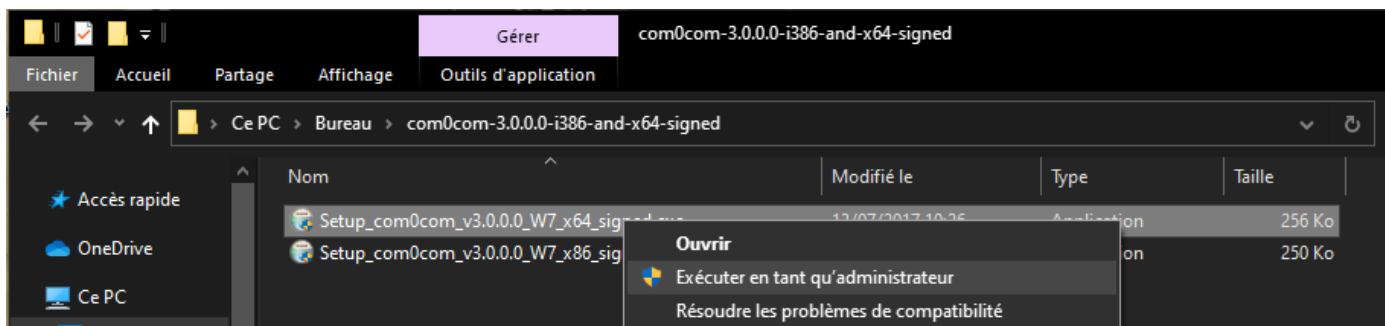
PORT COM PICSIMLAB ET EMULATEUR NULL MODEM



Afin d'utiliser les deux port COM proposés par le simulateur PICSimLab et de les rediriger vers deux terminaux asynchrone, par exemple deux terminaux TeraTermPro (cf. ci-dessus), il faudra installer un émulateur Null Modem. Plusieurs technologies cohabitent sur le marché, nous installerons com0com. Ce logiciel est chargé de rediriger l'utilisation d'un port COM série utilisé par une application vers une autre application. L'émulateur com0com sert donc de passerelle d'application à application. Après installation et configuration de com0com, charger le programme `picsimlab_win64/examples/bridge_uart1_uart2.hex` ou `disco/apps/hex/bridge_uart1_uart2.hex` (cf. ci-dessus). Observons la configuration de l'exemple ci-dessus :

- port COM3 (UART1 PICSimLab) redirigé vers port COM4 (TeraTerm fenêtre de gauche)
- port COM7 (UART2 PICSimLab) redirigé vers port COM8 (TeraTerm fenêtre de droite)
- Emulateur Null Modem com0com :

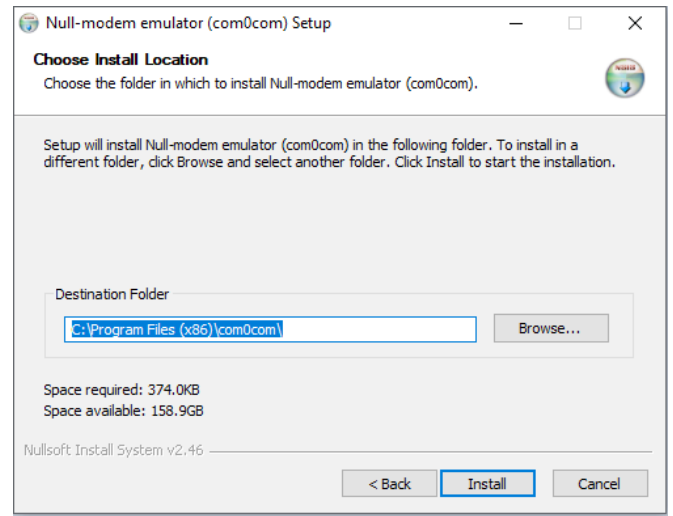
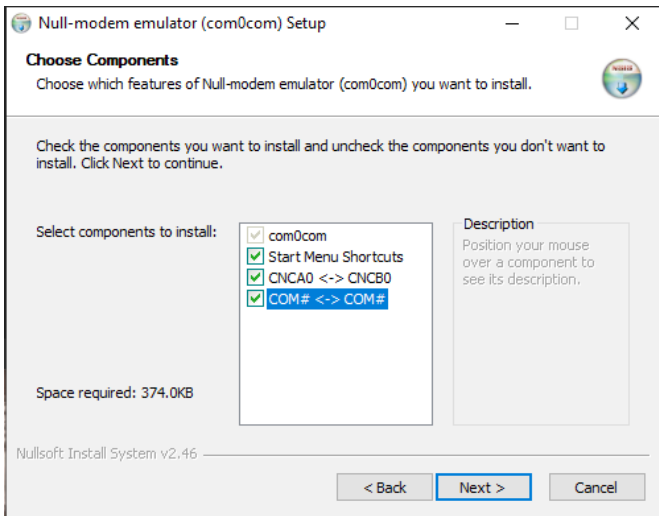
<https://sourceforge.net/projects/com0com/files/com0com/3.0.0.0/com0com-3.0.0.0-i386-and-x64-signed.zip/download>



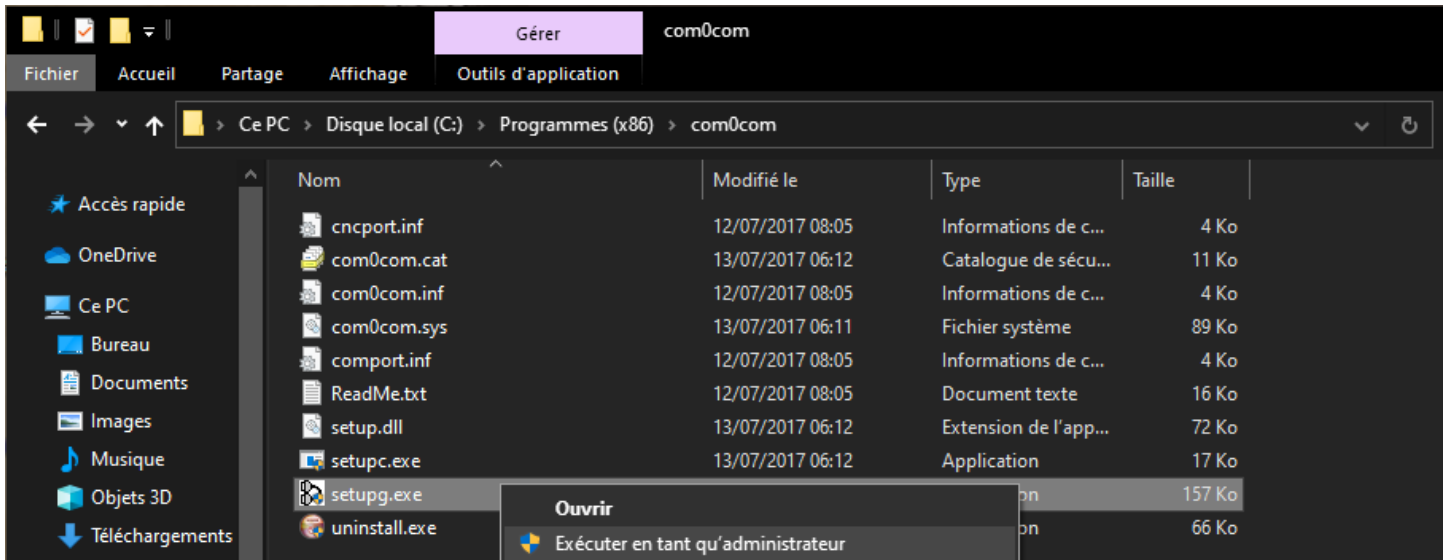
- Tutoriel officiel PICSimLab pour installer et configurer com0com :

https://lcgamboa.github.io/picsimlab/Com0com_Installation_Config.html

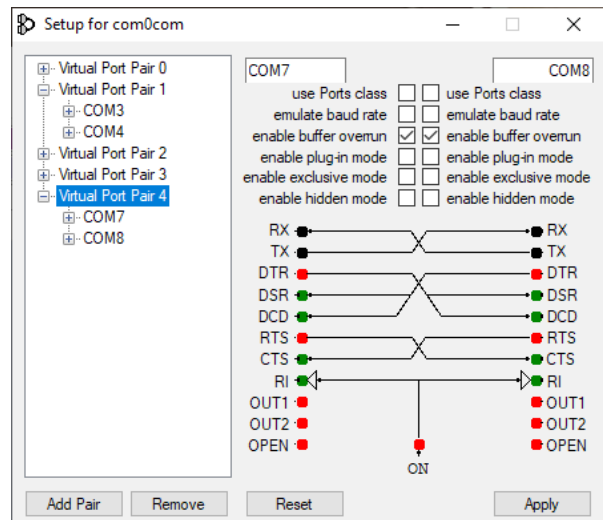
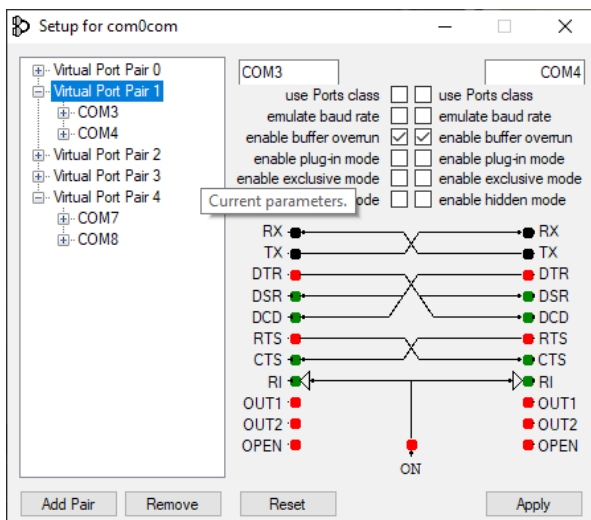
- Installer com0com :



- Exécuter com0com : C:/Programmes(x86)/com0com/setupg.exe



- Configurer com0com. Les ports COM illustrés ci-dessous (COM3, COM4, COM7 et COM8) sont donnés à titre indicatif. Adapter les port COM utilisés à vos besoins. Si un port COM est utilisé par une autre application, en sélectionné un autre :



SYNTHÈSE DES OUTILS A INSTALLER – moins de 1h

IDE MPLABX ET TOOLCHAIN XC8

- IDE (Integrated Development Environment) MPLABX **v5.50**
- Toolchain C XC8 **v1.45** (Free Mode)
- Lien de téléchargement vers les archives des outils Microchip (IDE MPLABX, toolchain XC8, etc) :
<https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>

TERMINAL SÉRIE TERATERM ET DRIVER VCP

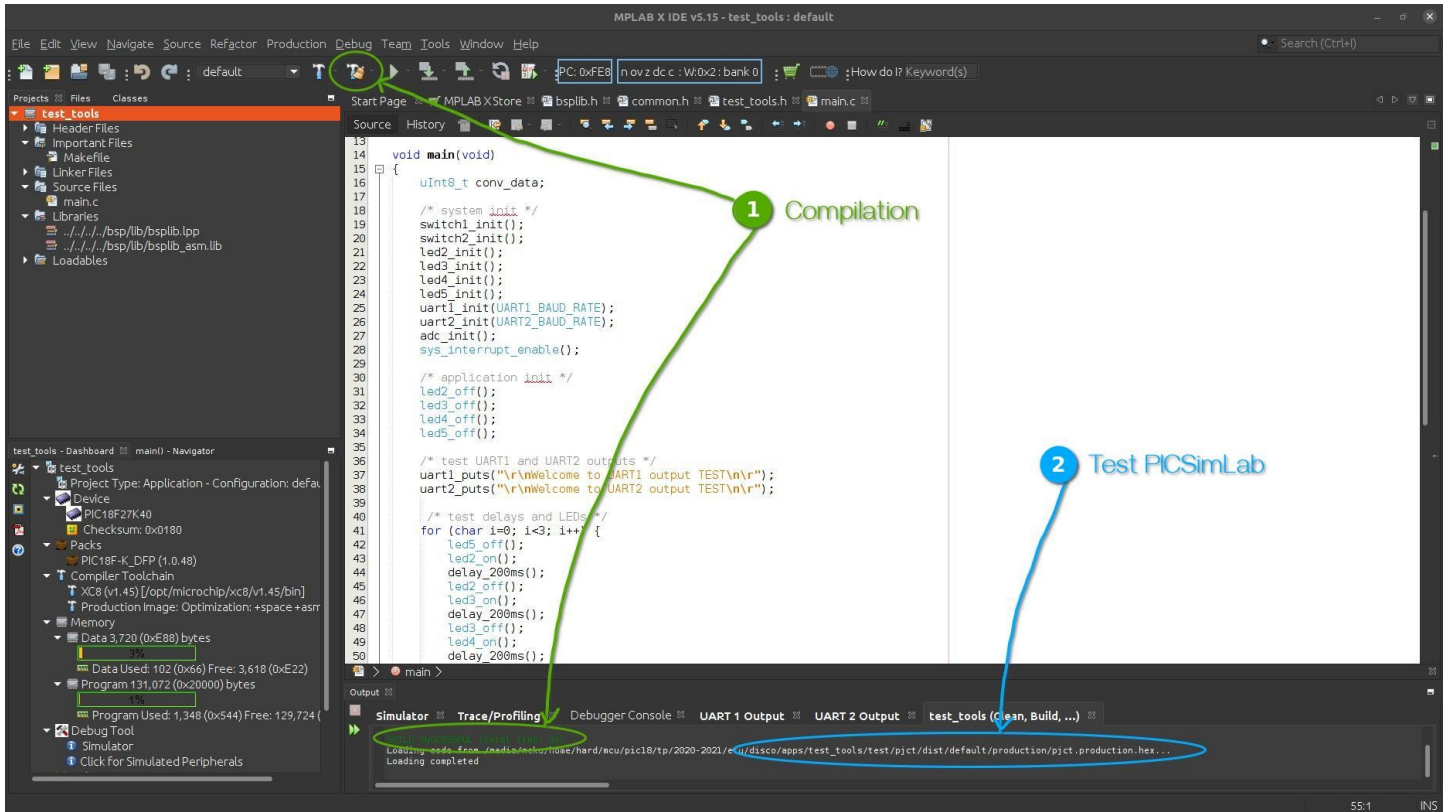
- Tera Term (dernière version) : <https://ttssh2.osdn.jp/index.html.en>
- ScanaStudio pour les sondes logiques SQ50 et SQ100 de IKALOGIC (dernière version) :
<https://ikalogic.com/scanastudio/>
- Drivers VCP (Virtual COM Port) pour chip USB to UART de FTDI (dernière version) :
<http://www.ftdichip.com/Drivers/VCP.htm>

SIMULATEUR PICSIMLAB ET EMULATEUR NULL MODEM

- Simulateur PICSimLab pour Windows 64bits avec carte Curiosity HPC et MCU PIC18F27K40 :
<https://foad.ensicaen.fr/mod/resource/view.php?id=24874>
- Emulateur Null Modem com0com :
<https://sourceforge.net/projects/com0com/files/com0com/3.0.0.0/com0com-3.0.0.0-i386-and-x64-signed.zip/download>

TEST DES OUTILS DE DÉVELOPPEMENT – moins de 5mn

- Ouvrir l'environnement de développement *MPLABX IDE* et le simulateur de carte *PICSimLab*
- Ouvrir le projet de test : *File > Open Project... > ../disco/apps/test_tools/test/pjct*
- Compiler le projet et vérifier la sortie (cf. ci-dessous) : *Lire et analyser la fonction principale*



- Sous MPLABX IDE, observer après compilation le dossier où a été sauvé le fichier .hex de sortie :

BUILD SUCCESSFUL

Loading code from **/<your_computer_path>/dist/default/production/pjct.production.hex**

- Sous PICSimLab, charger le fichier .hex précédent dans le simulateur : *File > Load Hex > ...file.hex*

