

Programmation de l'ObcP

Programmation via le DFU

Cette partie est dédiée à la programmation de votre objet connecté personnalisé. La programmation passe par le DFU :

https://www.st.com/resource/en/application_note/cd00264379-usb-dfu-protocol-used-in-the-stm32-bootloader-stmicroelectronics.pdf

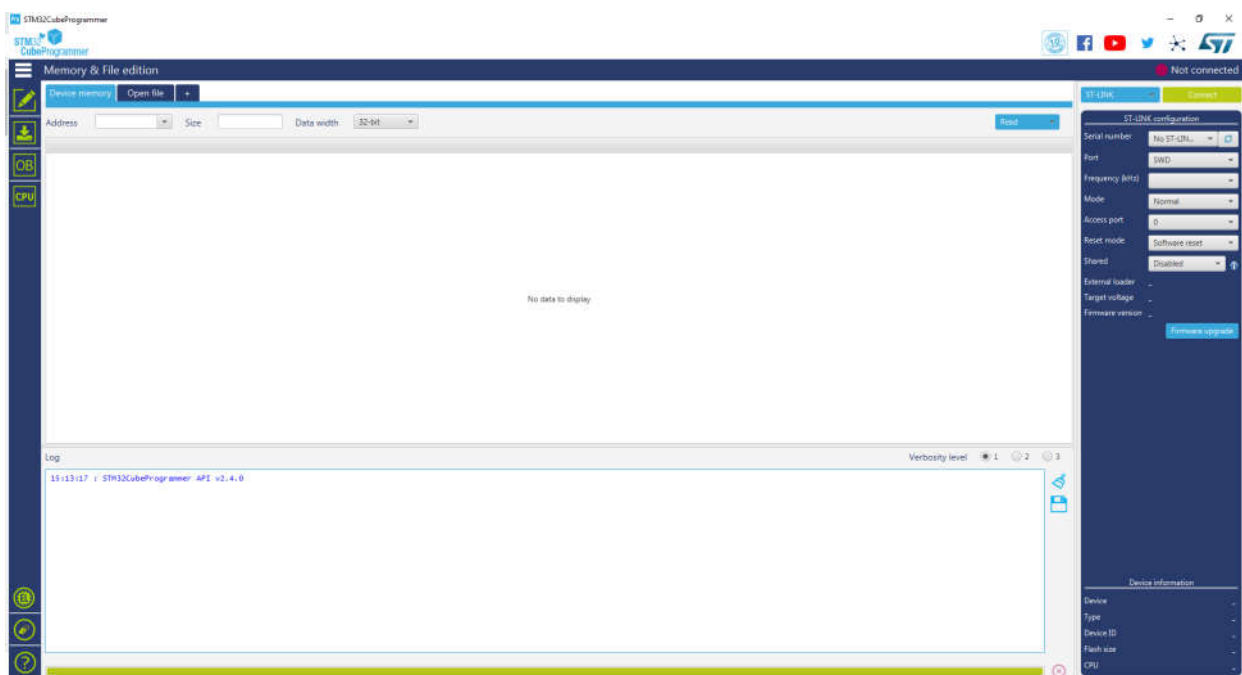
Un logiciel fourni par ST Microelectronics est disponible pour la programmation des microcontrôleurs de la famille des STM. Ce programmeur peut effectuer le téléversement du programme aussi bien par debugger que par DFU. Il est disponible dans le répertoire électronique de l'image ENSMM, dans la partie microcontrôleurs de moodle ou encore téléchargeable à cette adresse :

<https://www.st.com/en/development-tools/stm32cubeprog.html>

Protocole de programmation de l'ObcP

Étape 1 : Depuis CUBEIDE, lancer la compilation et récupérer le fichier .bin (par exemple mon_programme.bin)

Étape 2 : Lancer le programme STM32CubeProgrammer :



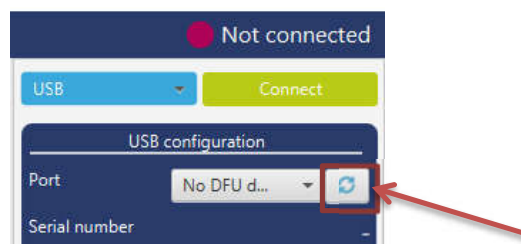
Dans la partie en haut à droite choisir USB :



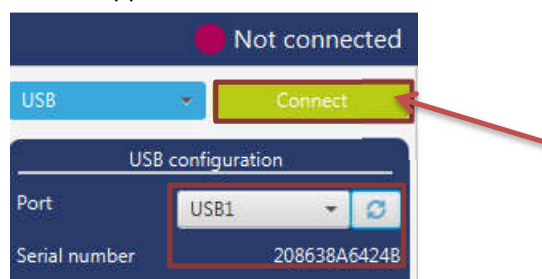
Etape 3 : Connecter l'ObcP via son câble USB. Mettre l'ObcP en mode **Prog** (interrupteur SW4) mettre la carte sur **On** (interrupteur SW2) et appuyer sur le bouton poussoir **SW3 reset**.



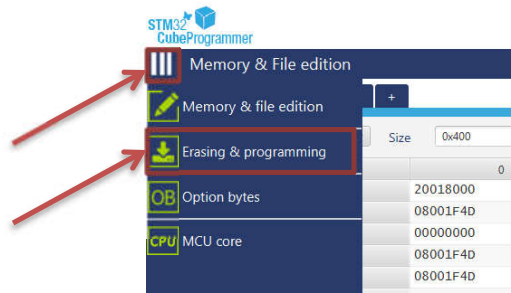
Cliquer sur le bouton refresh :



Un fois la carte detectée en mode DFU, elle apparait dans **Port et Serial Number**, cliquer alors sur **Connect** :



Aller ensuite dans le menu **erasing & programming** :



Aller ensuite dans **Browse** pour choisir le fichier **Bin** à programmer, puis cliquer sur **Start Programming** pour lancer le televersement du programme en mémoire flash :



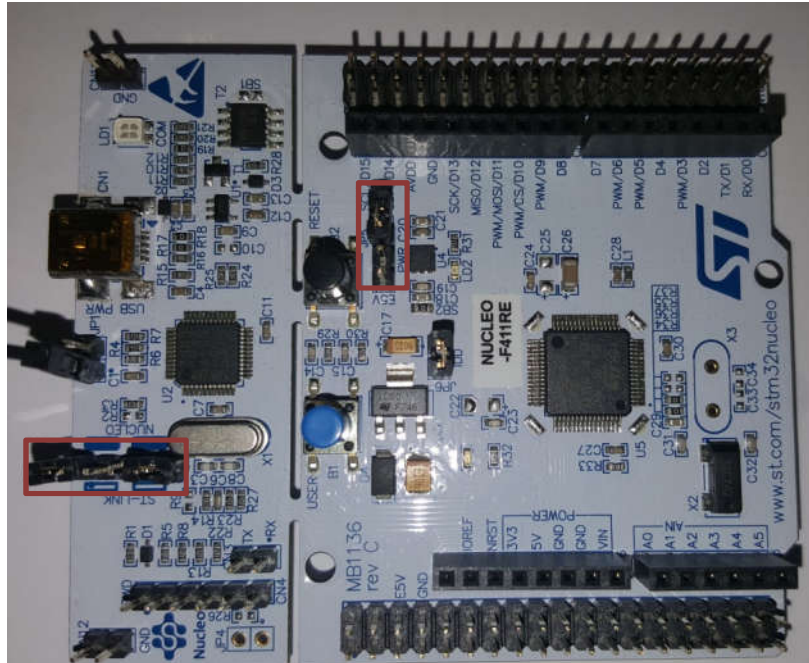
Etape 4 : Mettre l'ObcP en mode **Run** (SW4) et appuyer sur le bouton poussoir **SW3 reset**. Le programme doit se lancer... le port USB peut être utilisé pour une communication (**Virtual Com port** par exemple).



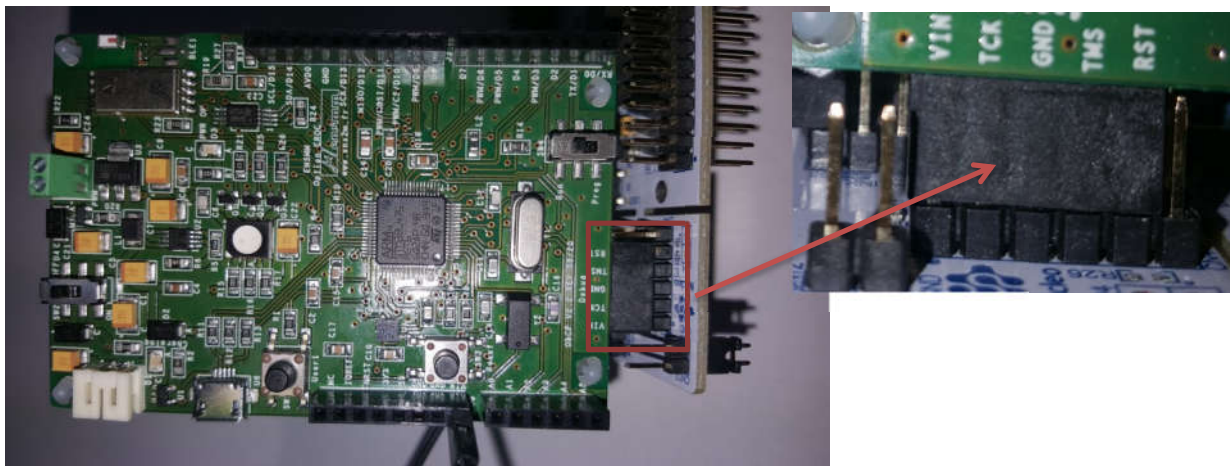
Programmation via un debugger

Il est tout à fait possible d'utiliser le debugger d'une carte Nucleo pour programmer l'ObcP, la programmation se fait alors par un simple « glisser-déplacer ».

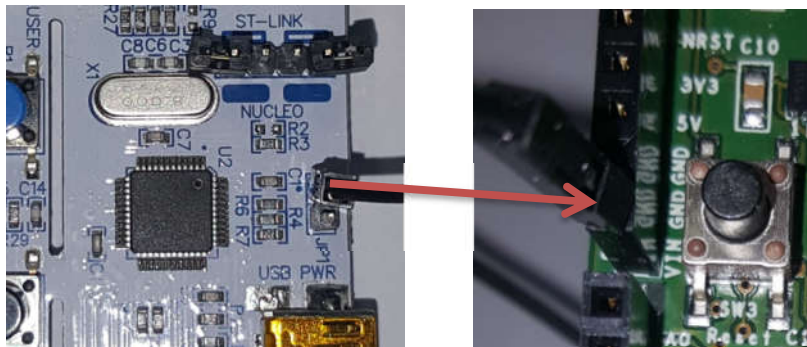
Etape 1 : Configurer la carte Nucleo ; les cavaliers ST-LINK et JP5 doivent être désactivés :



Etape 2 : Connecter l'ObcP sur le connecteur SWD de la carte Nucleo via son port de debug :



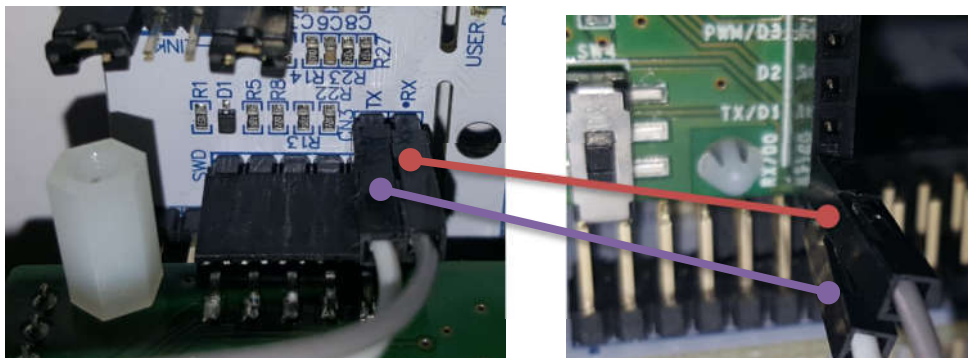
Etape 3 : Connecter l'alimentation de la carte Nucleo (Pin 1 de JP1) sur l'entrée Vin de l'ObCP :



Etape 4 : Mettre à jour le firmware du debugger avec STM32 Cube programmer par exemple :



La carte ObCP apparait maintenant comme un lecteur USB et peut désormais être programmer par « glisser-déplacer ». Petit plus, il est possible de connecter le TX RX pour se servir du terminal comme sur la carte Nucleo en connectant RX de l'ObCP sur TX de la carte Nucleo et TX de l'ObCP sur RX de la carte Nucleo :



Brochage et définition de l'ObCP

Les ports d'entrées/sorties sont définis comme sur la carte Nucléo en ce qui concerne le port compatible Arduino. Les autres ports sont définis comme indiqué dans le tableau suivant :

Fonctions	Brochage
Entrées / Sorties	
LED verte	PC_8
LED rouge	PC_6
LED bleue	PC_9
Bornier sortie PWM	PB_1
Bouton poussoir	PC_13
Accéléromètre	
MOSI	PC_12
MISO	PC_11
CS	PC_5
SCLK	PC_10
Bluetooth low energy	
MOSI	PB_15
MISO	PB_14
nCS	PB_2
RESET	PC_2
IRQ	PB_12
SCK	PB_13