

# Shanghai MXCHIP Information Technology Co., Ltd.

IoT Solution Provider

Version	Author	Date	Note
V0.9	NEO	2019.09.20	Initial version
V0.91	NEO	2019.10.27	Update Tool chain

## Step 1. Install Mxos-cube

### 1. Install Python2.7.13 & Git

--Python Software: <https://www.python.org/downloads/release/python-2713/>

--Git: <https://git-scm.com/>

### 2. Configure python and git to your system path

--Python27: <https://superuser.com/questions/143119/how-do-i-add-python-to-the-windows-path>

Python Pip: same as above, add path(example): C:\Python27\Scripts

Git: <https://stackoverflow.com/questions/26620312/installing-git-in-path-with-github-client-for-windows>

### 3. Install mxos-cube

--Use command: *pip install mxos-cube* in cmd.exe

## Step 2. Global Configure Micoder

1. Copy Micoder file to your defined path

--Software: [http://firmware.mxchip.com/MiCoder\\_v1.3\\_Win32:64.zip](http://firmware.mxchip.com/MiCoder_v1.3_Win32:64.zip)



MiCoder

2. Configure Micoder

--You can set the MiCoder Tools location via the cmd.exe:

```
$ mxos config --global MICODER C:\Users\neofr\Documents\MiCoder
```

```
[mxos] C:\Users\neofr\Documents\MiCoder now set as global MICODER
```

The -G or --global switch tells mxos Cube to set this as a global setting, rather than local for the current program.

Select C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.18362.356]
(c) 2019 Microsoft Corporation. All rights reserved.

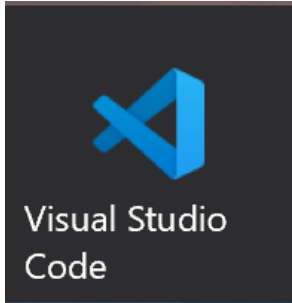
C:\Users\neofr>mxos config --global MICODER C:\Users\neofr\Documents\MiCoder
[mxos] C:\Users\neofr\Documents\MiCoder now set as global MICODER

C:\Users\neofr>mxos config --list
[mxos] Global config:
MICODER=C:\Users\neofr\Documents\MiCoder
```

## Step 3. Install Visual Studio Code

Download and Install Visual Studio Code

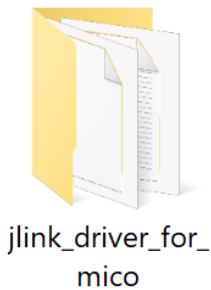
--Software: <https://code.visualstudio.com/>



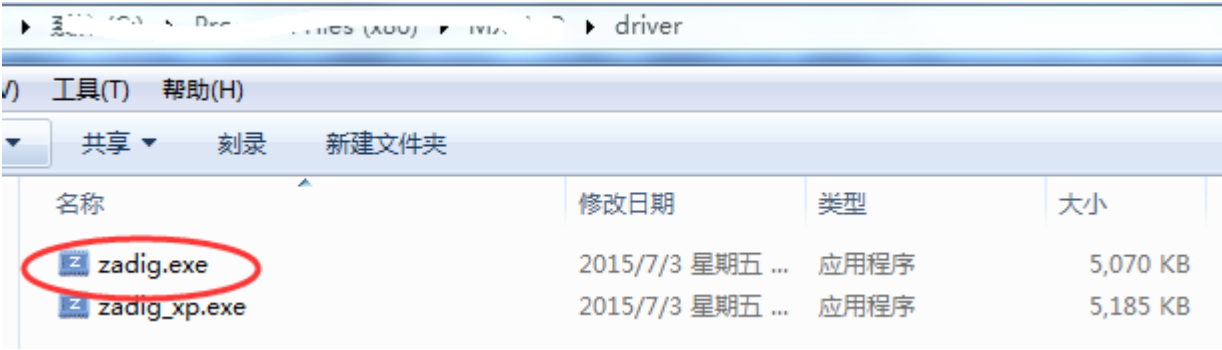
# Step 4. Install Driver

## 1. Install Driver

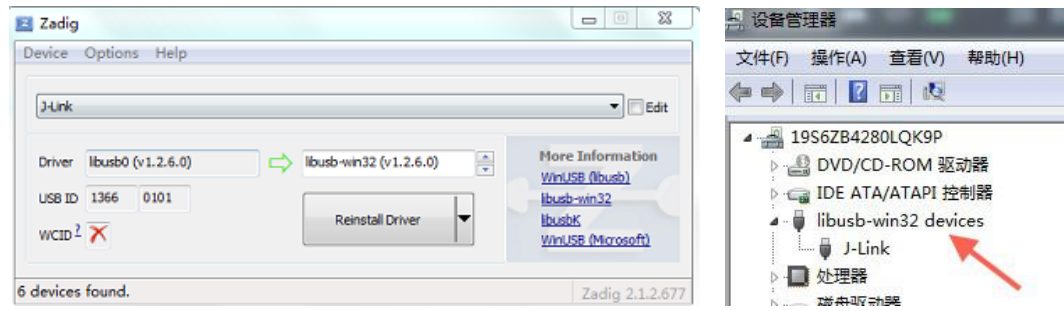
--Software: \MXCHIP Install for Windows\Driver Install



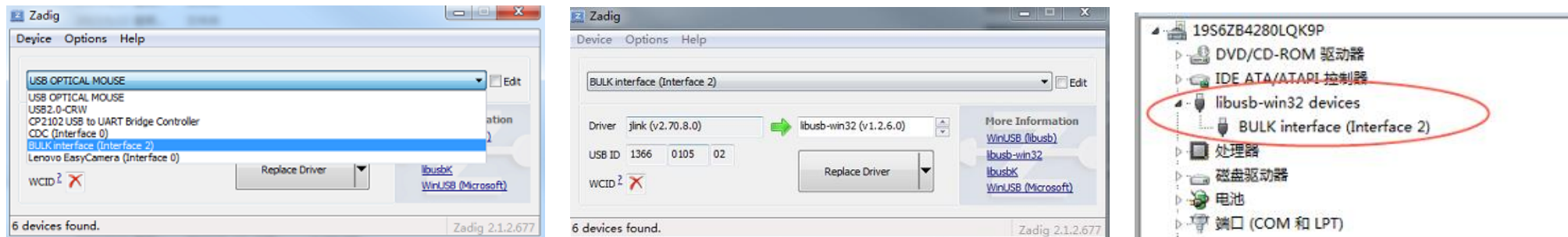
2. “CP210x\_Universal\_Windows\_Driver” is for UART driver, “Setup\_JLink\_V600i.exe” is for Jlink driver, “jlink\_driver\_for\_mico” is for MX1290 series when download by Jlink  
Run: zadig.exe or zadig\_xp.exe (for Windows XP) in “jlink\_driver\_for\_mico”



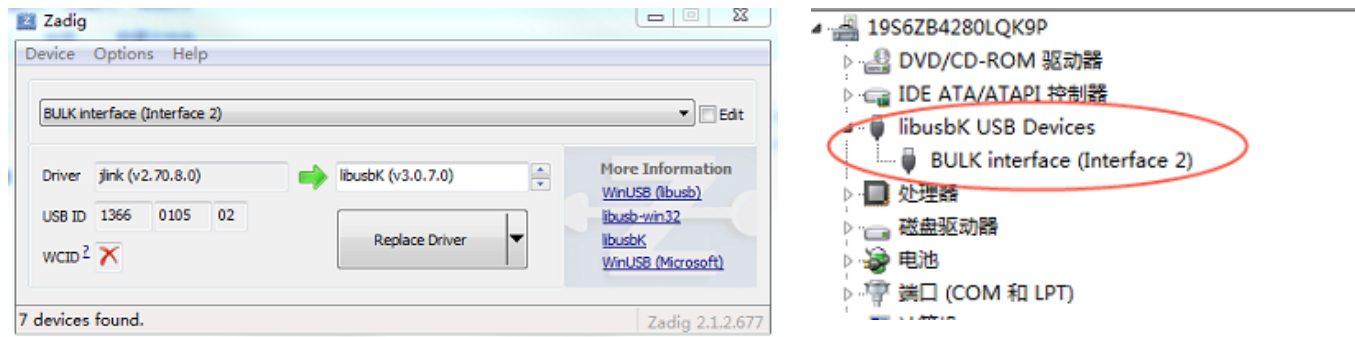
3. Select the menu: Options->List All Devices, select J-Link from the drop-down list, select "libusb-win32(v1.2.6.0)" for the driver, click "Replace Driver", and return to "successful" to close. "libusb-win32 devices" will appear in the device manager. If you are using JLinkV8, the update process is as follows:



If you are using JLinkV9, the update process is as follows:



Due to different JLink firmware versions from different manufacturers, some JLink drivers may not work properly after replacing them with libusb-win32 (v1.2.6.0). In this case, try using "libusbK (v3.0.7.0)".

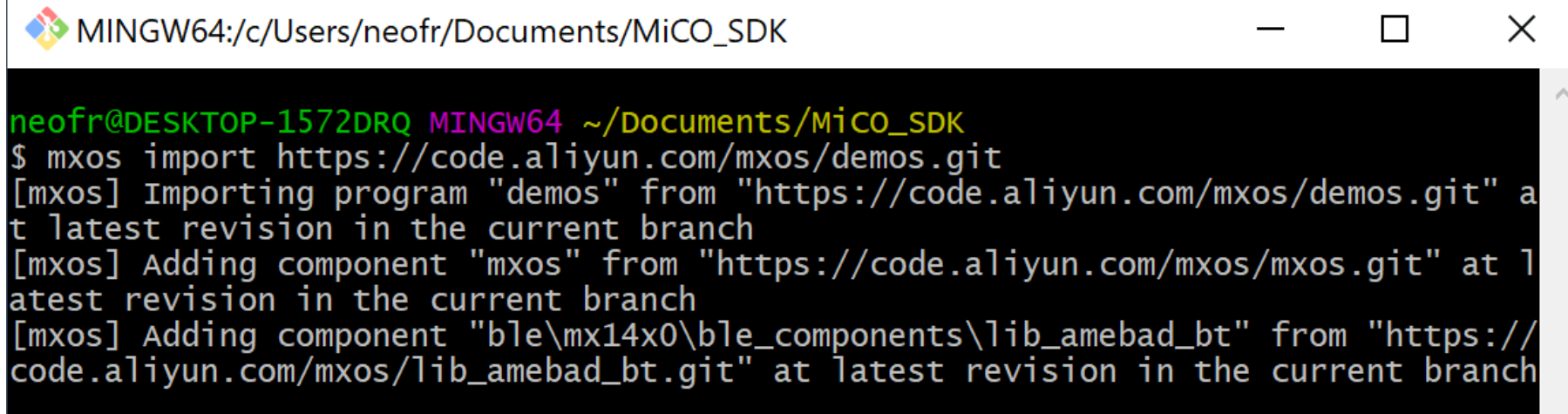


## Step 5. Download SDK

### 1. SDK

--Https: <https://code.aliyun.com/mxos/demos.git>

### 2. Use Git Bash to download SDK with command: *mxos import https://code.aliyun.com/mxos/demos.git*

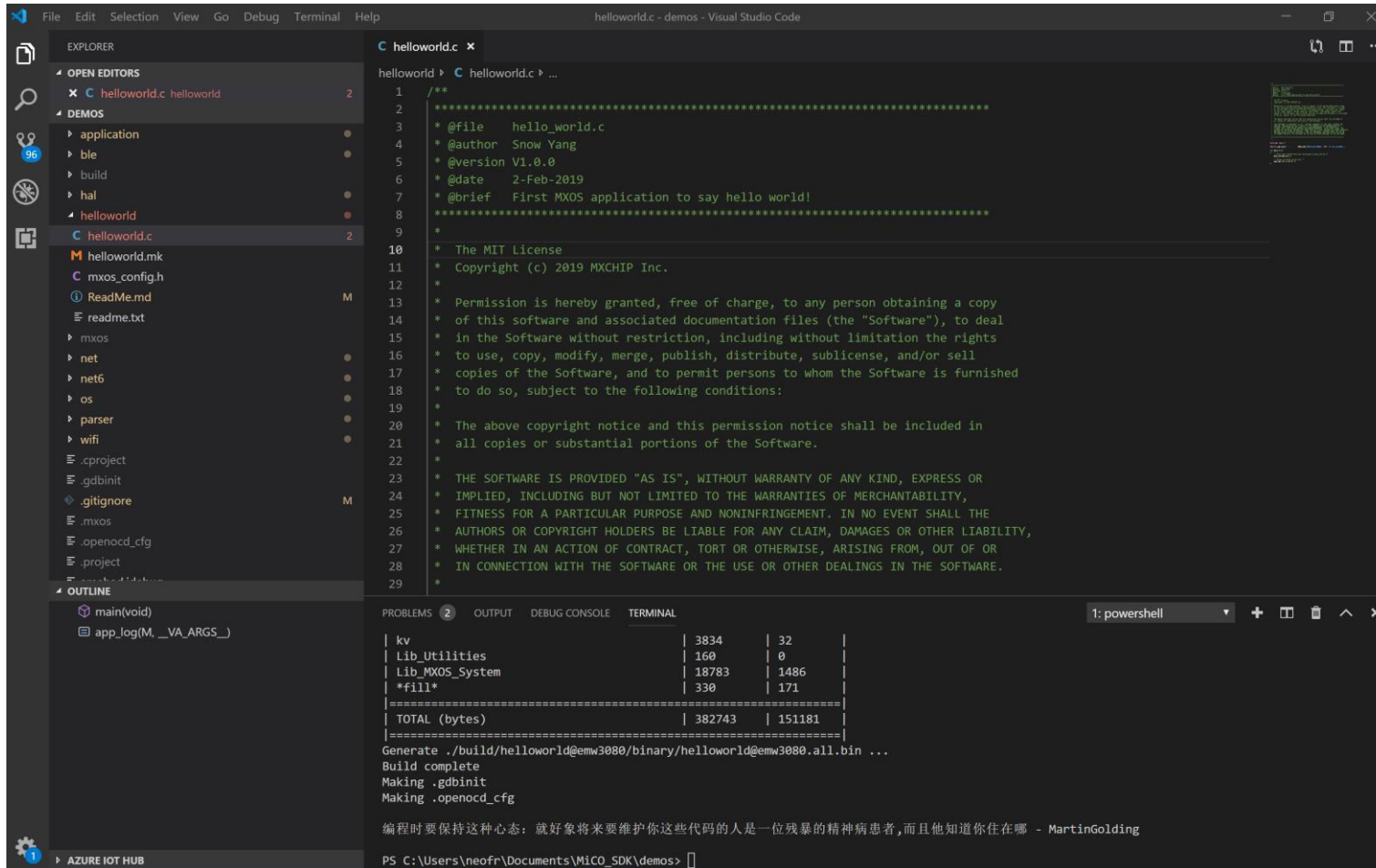


```
MINGW64:/c/Users/neofr/Documents/MiCO_SDK
neofr@DESKTOP-1572DRQ MINGW64 ~/Documents/MiCO_SDK
$ mxos import https://code.aliyun.com/mxos/demos.git
[mxos] Importing program "demos" from "https://code.aliyun.com/mxos/demos.git" at latest revision in the current branch
[mxos] Adding component "mxos" from "https://code.aliyun.com/mxos/mxos.git" at latest revision in the current branch
[mxos] Adding component "ble\mx14x0\ble_components\lib_amebad_bt" from "https://code.aliyun.com/mxos/lib_amebad_bt.git" at latest revision in the current branch
```



## Step 6. Use SDK with VS Code

### 1. Open demos folder in VS code



### 2. Compile with command in TERMINAL: *mxos make helloworld@emw3080*

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\neofr\Documents\MiCO_SDK\demos> mxos make helloworld@emw3080
Making config file for first time
processing components: helloworld emw3080 MXOS
Making build/helloworld@emw3080/libraries/EMW3080.a
Making build/helloworld@emw3080/libraries/Lib_MXOS_Kernel.a
Making build/helloworld@emw3080/libraries/Lib_MXOS_Security_SRP_6a.a
Making build/helloworld@emw3080/libraries/Lib_MXOS_Security_Sodium.a
Making build/helloworld@emw3080/libraries/aws.a
Making build/helloworld@emw3080/libraries/Lib_wolfSSL.a
Compiling App_Helloworld
Compiling MXOS
Compiling Lib_MXOS_Security
Compiling Lib_MXOS_System
Compiling Lib_Utillities
Compiling platform_mx1290
```