

Running XDot Examples

Before Installing MBED CLI

We need to install dependencies for running MBED CLI first. The dependencies include.

1. **Python 3** <https://www.python.org/downloads/>
2. **Mercurial** <https://www.mercurial-scm.org/release/windows/>
3. **Git** <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Please make sure to include the installation in the environment path while installing. Follow this video for reference (make sure you take the latest version from the above-provided links)

<https://www.youtube.com/watch?v=XvVyLxHuCpQ>

In this document, the steps are shown for the windows platform. For Mac OS and Linux, follow similar steps of installation.

Installing MBED CLI

For installing MBED CLI, download the installer from here:

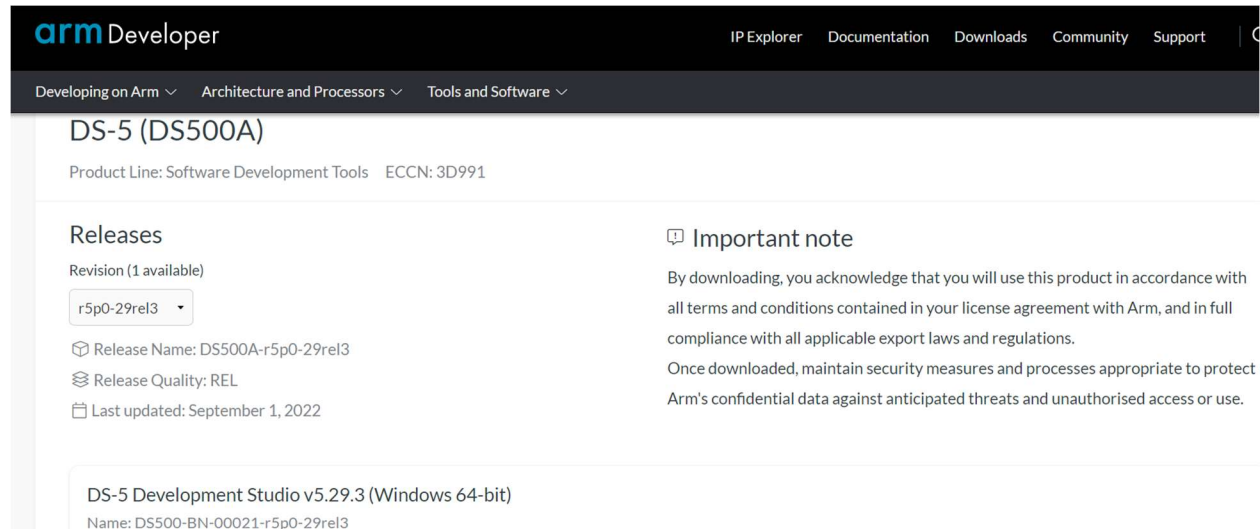
https://github.com/ARMmbed/mbed-cli-windows-installer/releases/download/v0.4.10/Mbed_installer_v0.4.10.exe

Make sure you select the option “include the installation in environment path variables”.

Installing ARM GCC compiler

Go to <https://developer.arm.com/downloads/-/ds-5-downloads>

You might have to register/login to ARM website. Complete the steps. You must see the below page



The screenshot shows the ARM Developer website for the DS-5 (DS500A) product. The navigation bar includes links for IP Explorer, Documentation, Downloads, Community, and Support. The sidebar on the left contains a 'Releases' section with a dropdown menu set to 'r5p0-29rel3', showing release details: Release Name: DS500A-r5p0-29rel3, Release Quality: REL, and Last updated: September 1, 2022. An 'Important note' section is also present, stating that by downloading, the user acknowledges the terms and conditions of the license agreement. The main content area displays the product name 'DS-5 (DS500A)', the product line 'Software Development Tools', and the ECCN '3D991'. At the bottom, there is a link to download the 'DS-5 Development Studio v5.29.3 (Windows 64-bit)' with the file name 'DS500-BN-00021-r5p0-29rel3'.

Select the release and download the zip file. (You might be asked to add extensions for faster downloads)

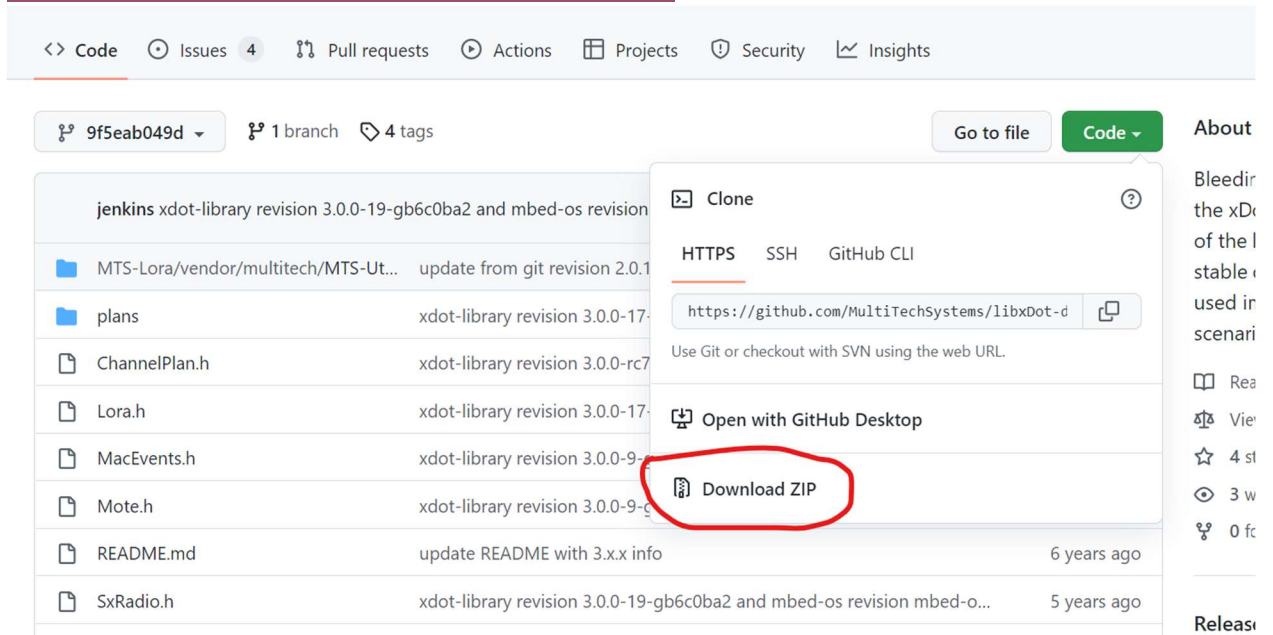
Unzip it. Run the installation file Add it to environment path during installation.

Importing Code and libraries

Create a folder in a path for your Dot examples DOT_PATH\

Open command line

1. Run mbed import <http://os.mbed.com/teams/L-TEK/code/Dot-Examples/>
2. If you get issues related module error, install pip version 22.2.2
3. This should create a director DOT_PATH\Dot-Examples\
4. Run cd Dot-Examples and mkdir libxDot-dev-mbed5
5. Download and unzip contents from <https://github.com/MultiTechSystems/libxDot-dev-mbed5/tree/9f5eab049dc97f9a0ab12219e9871c60f7efb427>



6. Create folder mbed-os folder in DOT_PATH\Dot-Examples
7. Download and unzip contents from <https://github.com/ARMmbed/mbed-os/archive/refs/tags/mbed-os-5.6.2.zip> to this folder

8. Modify example_config.h in DOT_PATH\Dot-Examples\examples\src\ to run the type of example to run

```
1  #ifndef __EXAMPLE_CONFIG_H
2  #define __EXAMPLE_CONFIG_H
3
4  #define OTA_EXAMPLE           1 // see ota_example.cpp
5  #define AUTO_OTA_EXAMPLE     2 // see auto_ota_example.cpp
6  #define MANUAL_EXAMPLE       3 // see manual_example.cpp
7  #define PEER_TO_PEER_EXAMPLE 4 // see peer_to_peer_example.cpp
8  #define CLASS_C_EXAMPLE      5 // see class_c_example.cpp
9
10 // the active example is the one that will be compiled
11 #if !defined(ACTIVE_EXAMPLE)
12 #define ACTIVE_EXAMPLE PEER_TO_PEER_EXAMPLE
13 #endif
14
15 // the active channel plan is the one that will be compiled
16 // options are :
17 //     CP_US915
18 //     CP_AU915
19 //     CP_EU868
20 //     CP_KR920
21 //     CP_AS923
22 //     CP_AS923_JAPAN
23 //     CP_IN865
24 #if !defined(CHANNEL_PLAN)
25 #define CHANNEL_PLAN CP_US915
26 #endif
27
28 #endif
29
```

9. Make sure to Set CP_US915 as channel plan in the config file
10. FOR OTA Example, modify ota_example.cpp deep_sleep=true public_network=true and change network ID and network key and make it use those
11. Add dependency on light sensor of the example by running “mbed add <http://os.mbed.com/teams/Multi-Hackers/code/ISL29011/>”
12. Now run “mbed compile -m XDOT_L151CC -t GCC_ARM” to compile

Refer : <https://www.youtube.com/watch?v=C6LQJnd-E9E> for an example of how to import and compile

13. Now once bin is generated, go to DOT_PATH\Dot-Examples\BUILD\XDOT_L151CC\GCC_ARM and copy the Dot-Examples.bin to the XDOT USB enumeration folder to flash
14. Reset the board to run the example successfully

Getting output from XDOT

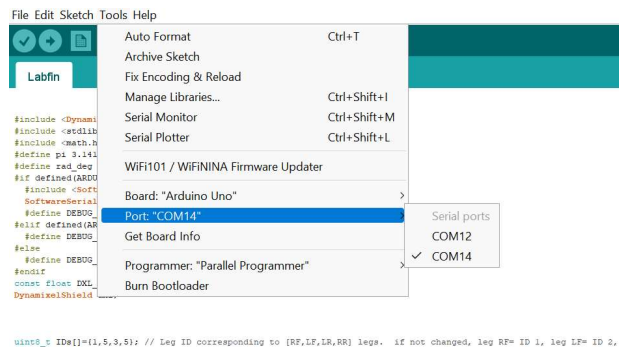
Now that the example is running, we need to get the serial output for status from XDOT.

Make sure in the code we have this line as below if you are using the XDOT Dev board.

```
examples > src > peer_to_peer_example.cpp > pc(USBTX, USBRX)
20 // * these options must match between the two dev
21 // order for communication to be successful
22 ///////////////////////////////////////////////////
23 static uint8_t network_address[] = { 0x01, 0x02, 0x03, 0x04
24 static uint8_t network_session_key[] = { 0x01, 0x02, 0x03,
25 static uint8_t data_session_key[] = { 0x01, 0x02, 0x03, 0x0
26
27 mDot* dot = NULL;
28 lora::ChannelPlan* plan = NULL;
29
30 //Serial pc(USBTX, USBRX);
31 Serial pc(USBTX, USBRX);
32 #ifdef TARGET_FF1705_L151CC
33 AnalogIn lux(A0);
34 #elif defined(TARGET_XDOT_L151CC)
35 I2C i2c(I2C_SDA, I2C_SCL);
36 ISL29011 lux(i2c);
37 #else
38 AnalogIn lux(XBEE_AD0);
39 #endif
40
41 int main() {
```

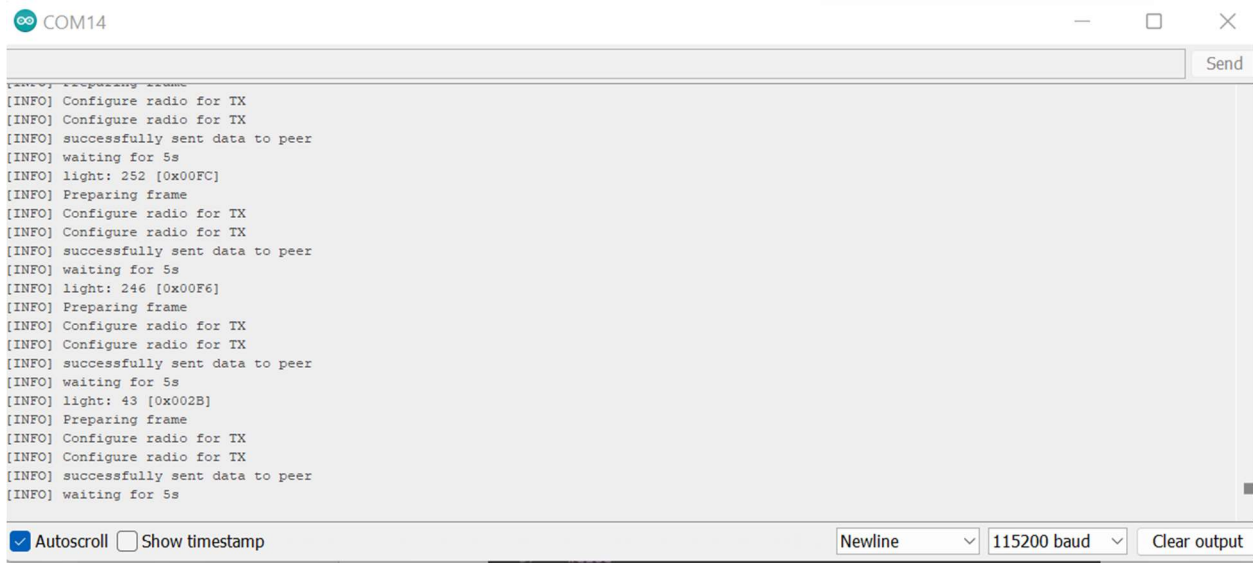
If we are using a standalone XDOT, we might require a USB to UART TTL connector and set pins P9 and P10 instead of USBTX and USBRX

We need to detect the port of UART which is enumerated for communication. So, you can use PuTTY and check the enumerated port from device manager. Or a simpler way is to use a third party application like Arduino which automatically detects the port for you and has a serial monitor as well.



Select the port and open serial monitor ctrl+shift+M

Make sure to set the baud rate to 115200. We should be able to get the below prints



```
[INFO] Preparing frame
[INFO] Configure radio for TX
[INFO] Configure radio for TX
[INFO] successfully sent data to peer
[INFO] waiting for Ss
[INFO] light: 252 [0x00FC]
[INFO] Preparing frame
[INFO] Configure radio for TX
[INFO] Configure radio for TX
[INFO] successfully sent data to peer
[INFO] waiting for Ss
[INFO] light: 246 [0x00F6]
[INFO] Preparing frame
[INFO] Configure radio for TX
[INFO] Configure radio for TX
[INFO] successfully sent data to peer
[INFO] waiting for Ss
[INFO] light: 43 [0x002B]
[INFO] Preparing frame
[INFO] Configure radio for TX
[INFO] Configure radio for TX
[INFO] successfully sent data to peer
[INFO] waiting for Ss
```

☒ Autoscroll ☐ Show timestamp Newline 115200 baud Clear output