Use configuration mode

IoT DevKit has the configuration model that you can configure settings like WiFi, IoT Hub connection string and security feature for it.

Before you begin

- Connect IoT DevKit to the computer.
- Download SSH and Telnet client like Putty (https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html) for Windows.

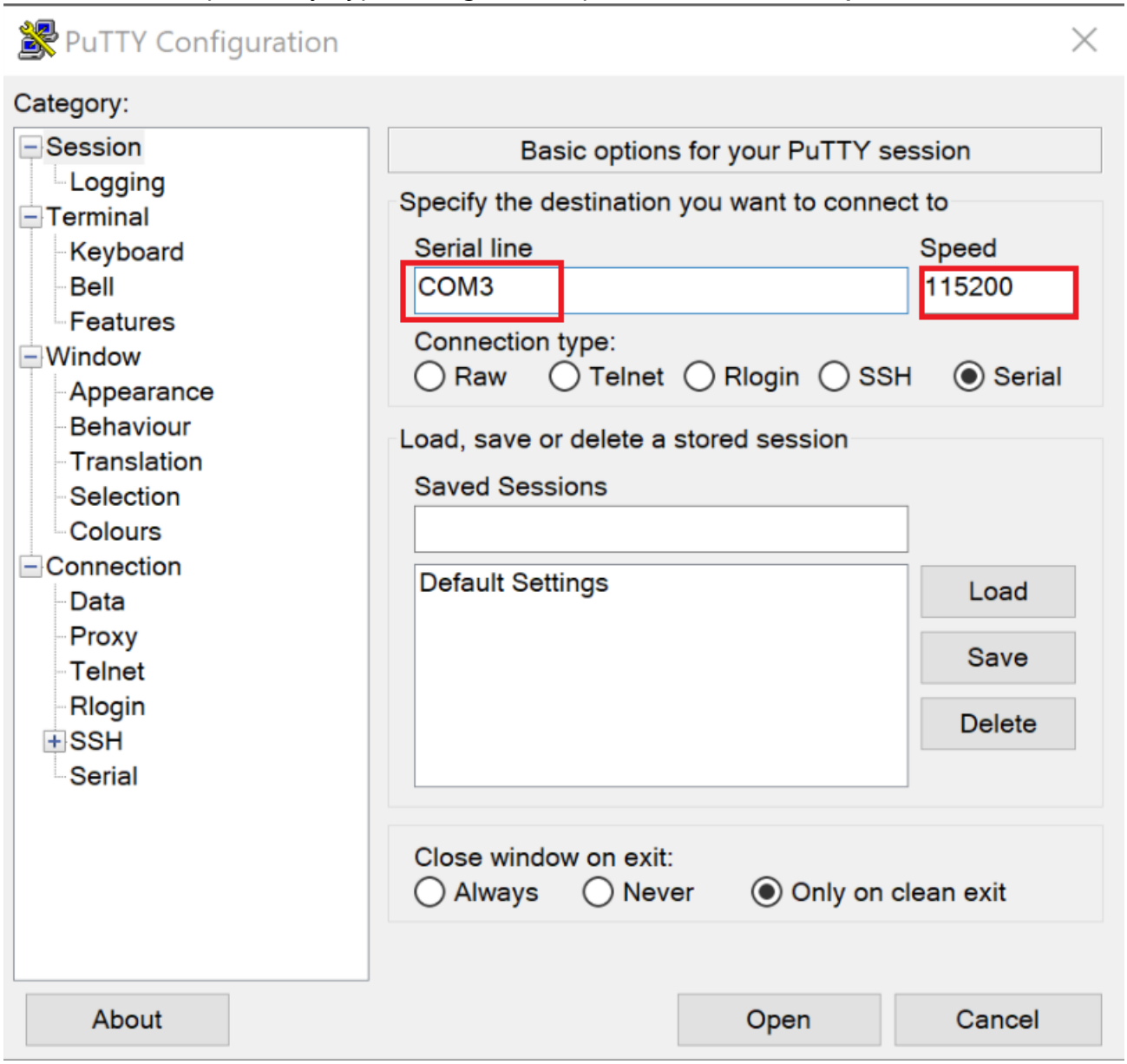
Platform: **Windows** macOS

Windows

Enter Configuration Mode

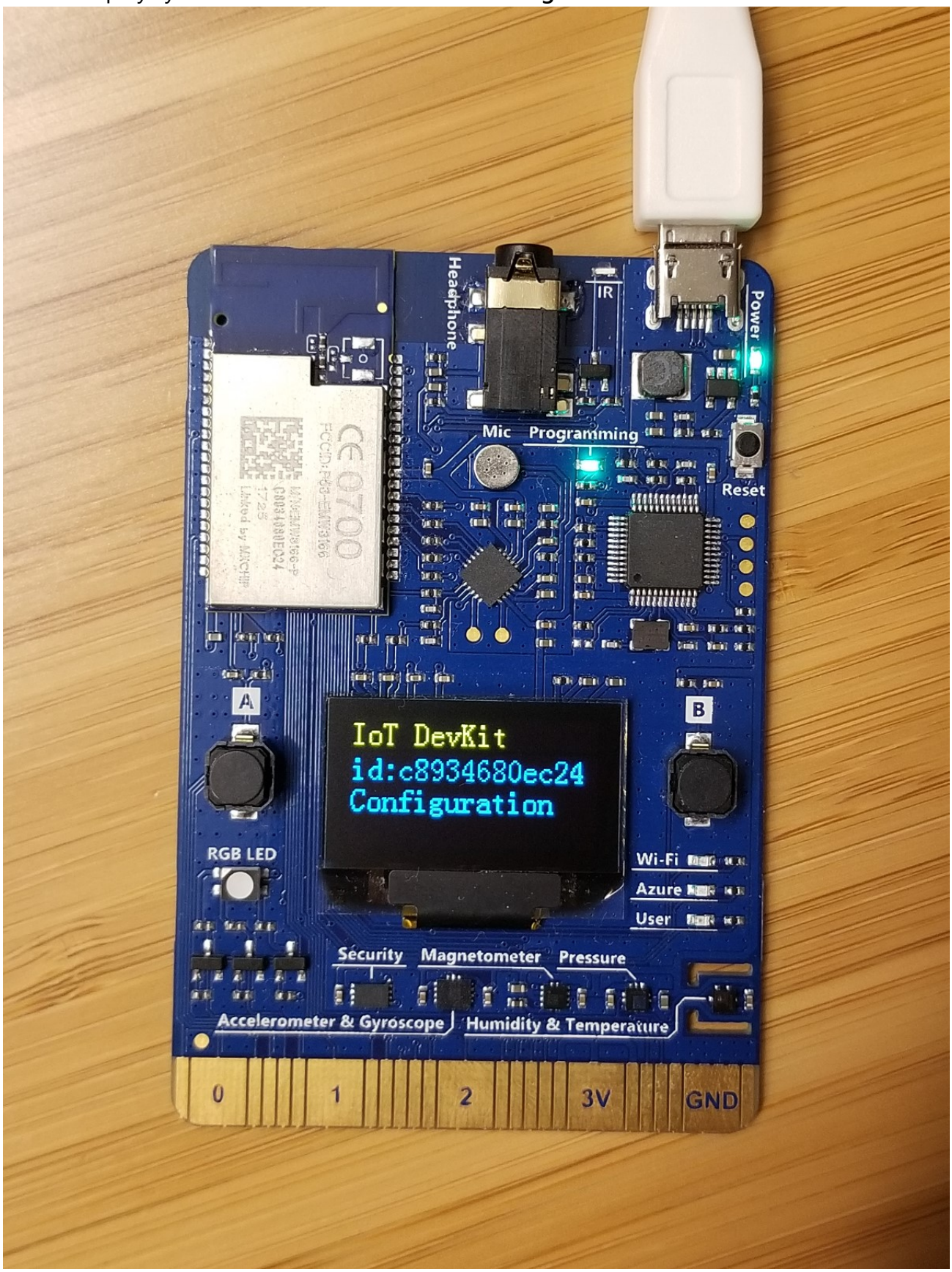
1. Open VS Code and identify the COM port on the status bar.

2. Download and open Putty, type the right COM port and **115200** for **Speed**.

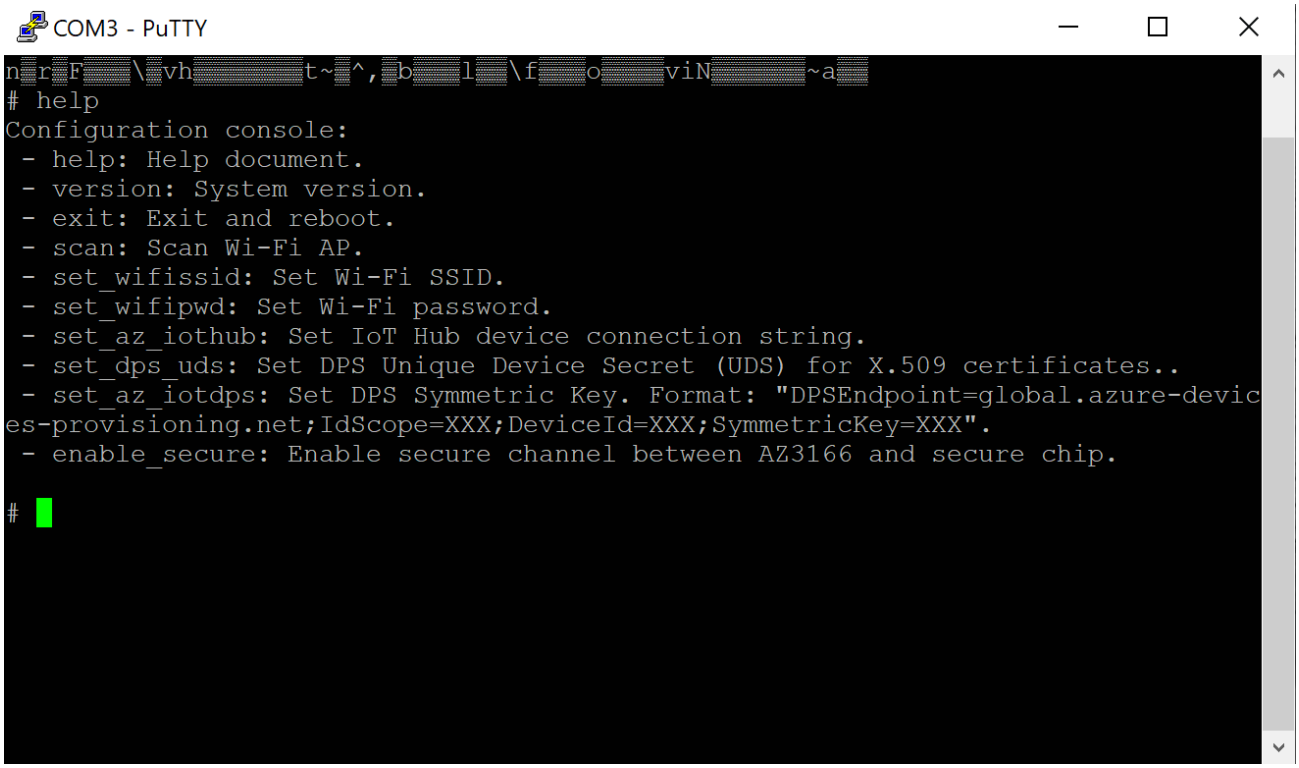


3. Click **Open**. A serial monitor window will open.

4. On the device, hold down button A, then push and release the reset button. Observe the screen displays your device MAC address and '**Configuration**'.



5. Serial monitor window displays command list available.



Commands

Commands for the IoT DevKit Configuration Mode.

help

```
help
```

Display the commands list

version

```
version
```

Display **DevKit SDK**, **Mico**, **mbed-os**, **mbed TLS lib** and **WiFi lib** version details.

exit

```
exit
```

Exit Configuration Mode and reboot into normal mode.

scan

```
scan
```

Scan and list all available WiFi SSID.

set_wifissid

```
set_wifissid [SSID]
```

Connect specific WiFi SSID.

set_wifipwd

```
set_wifipwd [password]
```

Set password for WiFi SSID you connected with command **set_wifissid**.

set_az_iothub

```
set_az_iothub [device connection string]
```

Set device connection string for Azur IoT Hub. View Understand Different Connection Strings in Azure IoT Hub (https://blogs.msdn.microsoft.com/iotdev/2017/05/09/understand-different-connection-strings-in-azure-iot-hub/) to learn more.

set_dps_uds

```
set_dps_uds [unique device secret]
```

Set Unique Device Secret that is used as device unique key to be used for calcuation by Device Identifier Composition Engine (DICE) (https://trustedcomputinggroup.org/work-groups/dice-architectures/) and be used to register on IoT Hub Device Provisioning Service (https://docs.microsoft.com/en-us/azure/iot-dps/about-iot-dps).

set_az_iotdps

```
set_az_iotdps [DPS Symmetric Key]
```

Set Symmetric Key (https://docs.microsoft.com/azure/iot-dps/concepts-device) based on the shared access signature (SAS) token that is supported by DPS. The Symmetric Key format is `"DPSEndpoint=global.azure-devices-provisioning.net;IdScope=XXX;DeviceId=XXX;SymmetricKey=XXX"`.

enable_secure

```
enable_secure
```

Enable secure channel on STSAFE secure chip.

Here are what will happen when enabling the security feature:

- A symmetric key initiated by the chip will be set and stored on the secure chip.
- All existing data stored in EEPROM (e.g. WiFi password) will be automatically encrypted.
- Once enabled, all data reading and writing on the device will be encrypted from that moment.

See Understand security chip (https://microsoft.github.io/azure-iot-developer-kit/docs/understand-security-chip/) for more details.

Problems and feedback

If you encounter problems, you can find FAQs (https://microsoft.github.io/azure-iot-developer-kit/docs/faq/) if you encounter problems or reach out to us from our Gitter channel (https://gitter.im/Microsoft/azure-iot-developer-kit).

Was this documentation helpful?

Yes No

Updated: January 08, 2018