



Digi XBee PyCharm IDE Plugin

User Guide

Revision history—90002445

Revision	Date	Description
A	December 2020	Initial release
B	August 2021	2.1.0 release.

Trademarks and copyright

Digi, Digi International, and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

© 2021 Digi International Inc. All rights reserved.

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

Customer support

Gather support information: Before contacting Digi technical support for help, gather the following information:

- Product name and model
- Product serial number (s)
- Firmware version
- Operating system/browser (if applicable)
- Logs (from time of reported issue)
- Trace (if possible)
- Description of issue
- Steps to reproduce

Contact Digi technical support: Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Feedback

To provide feedback on this document, email your comments to

techcomm@digicom.com

Include the document title and part number (Digi XBee PyCharm IDE Plugin User Guide, 90002445 B) in the subject line of your email.

Contents

Digi XBee PyCharm IDE Plugin User Guide

Install the Digi XBee PyCharm IDE Plugin	6
Requirements	6
Supported devices	6
Install the plugin	6
Discover the new UI elements	7
Create a Digi project from scratch	9
Import a Digi sample	9
Write your code	11
Import Python libraries and packages	13
XBee MicroPython libraries	13
Python packages	14
Build and run the project	14
Requirements	14
Device selection and run	14
Use the console to communicate with your device	17
XBee REPL console	17
Digi SSH console	18
How-Tos	18
Change the selected device	18
Do not automatically run MicroPython applications at module startup	19
Show the quick documentation popup	19
Enable/disable Python/MicroPython sources compilation	19
Change the associated platform of the project	20
Exclude files or folders from the compilation	20
Configure your Digi Remote Manager account	20

Digi XBee PyCharm IDE Plugin User Guide

The Digi XBee PyCharm IDE Plugin allows you to write, build and run Python or MicroPython applications for Digi devices in a quick and easy way. This is what you can do with it:

- Create Python or MicroPython projects from scratch or import one of the available examples.
- Get help while you write your code thanks to the syntax highlight, quick documentation, and code completion features.
- Build and upload Python or MicroPython applications to your Digi device with just one click.
- Add libraries that facilitate the usage of external peripherals or non-standard APIs.
- Communicate with your Digi device through the integrated MicroPython REPL or SSH console to see the application output or execute quick tests.
- Run MicroPython applications and communicate with any XBee Cellular module through Digi Remote Manager, regardless of its location—minimum firmware version: x18.

Install the Digi XBee PyCharm IDE Plugin	6
Discover the new UI elements	7
Create a Digi project from scratch	9
Import a Digi sample	9
Write your code	11
Import Python libraries and packages	13
Build and run the project	14
Use the console to communicate with your device	17
How-Tos	18

Install the Digi XBee PyCharm IDE Plugin

To download and install the Digi XBee MicroPython PyCharm Plugin, see the following requirements and installation steps.

Requirements

- A computer with the following characteristics:
 - 64-bit operating system:
 - Microsoft Windows 10, 8, 7 (SP1)
 - macOS 10.11 or later
 - Linux
 - 4 GB RAM minimum, 8 GB RAM recommended.
 - 1.5 GB hard disk space + at least 1 GB for caches.
 - 1024x768 minimum screen resolution.
- PyCharm 2019.3.1 or higher.
- Python 3.6 or higher.

Supported devices

XBee modules

- XBee 3 RF (Zigbee, DigiMesh, 802.15.4)
- XBee 3 Cellular (LTE Cat 1, LTE-M/NB-IoT)
- XBee Cellular (LTE Cat 1, 3G)

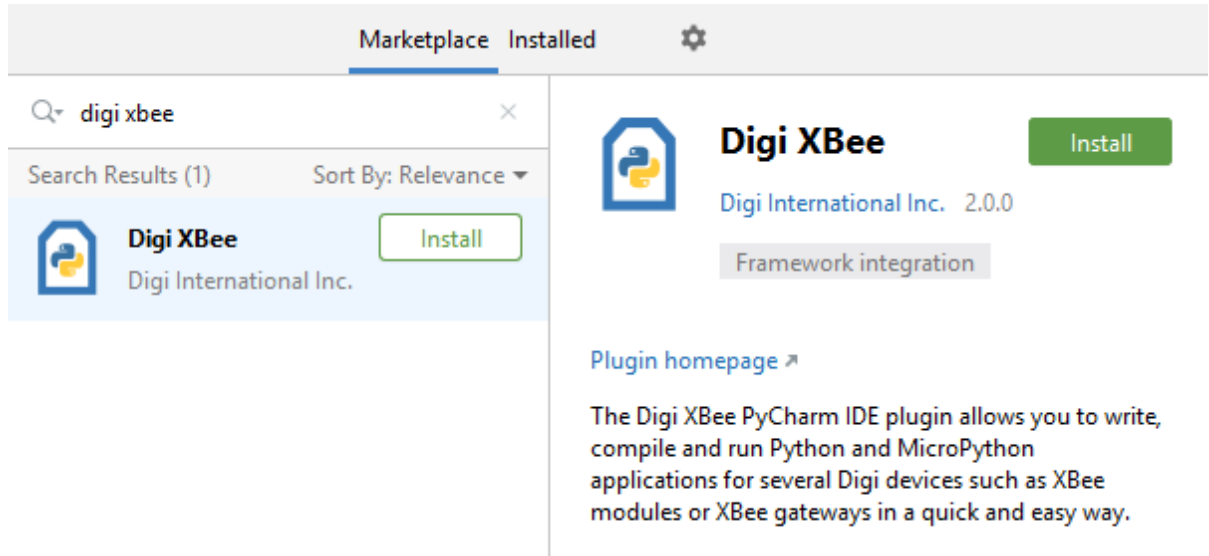
XBee gateways

- IX 15

Install the plugin

Install the Digi XBee PyCharm IDE Plugin from the PyCharm's plugins marketplace:

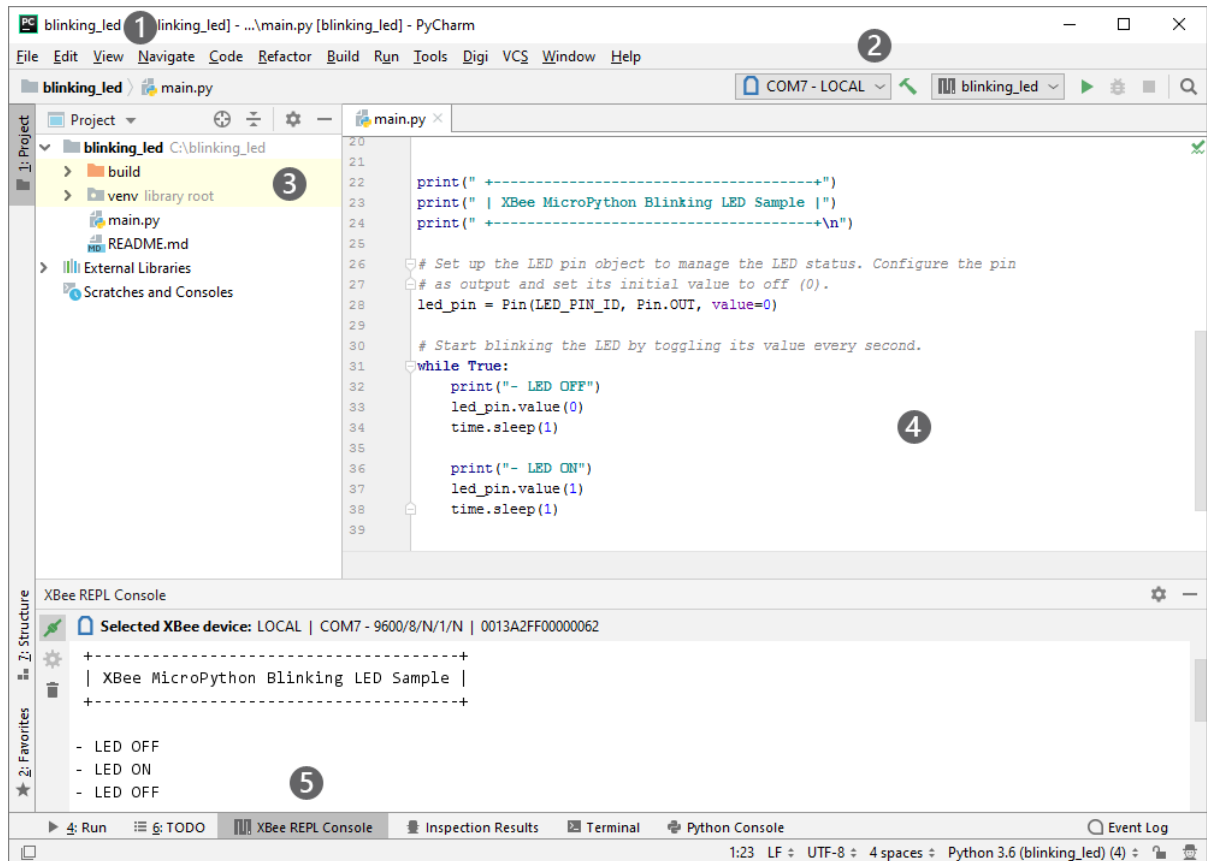
1. Open PyCharm.
2. Go to the **Plugins** window by doing one of the following:
 - a. Select **Configure > Plugins** if you are on the **Welcome** screen, or
 - b. Select **File > Settings > Plugins** if you have a project open.
3. Type **Digi XBee** in the **Marketplace** search box.



4. Click **Install**.
5. When finished, click **Restart IDE** to complete the plugin installation.

Discover the new UI elements

The Digi XBee PyCharm IDE Plugin adds some new elements to the PyCharm UI in order to facilitate the development of Python or MicroPython applications.



1. Main menu.

- The **File** menu includes two new options to [create a Digi project](#) and to [import a Digi sample](#).
 - The new **Build** menu allows you to build the project, generating the corresponding compiled files for each Python/MicroPython source, and clean the project.
 - The new **Digi** menu offers actions to open the REPL or SSH consoles, import a library, reset the selected XBee device or format the XBee file system. Some of these options only appear depending on the selected platform.
2. **Navigation bar.** It allows you to select or change the Digi device where the application will be flashed into, [build the project and run it](#).
 3. **Project view.** The **build** and **libs** folders are automatically created. The first one contains the Python/MicroPython compiled files, and the second one the libraries you have imported—if any.
 4. **Editor.** The plugin provides some features that help you to [write your code](#).
 5. **Tool Window bar.** The **XBee REPL Console** and **Digi SSH Console** tools allow you to quickly [communicate with your Digi device](#).

Note For more information about the IDE, see the PyCharm [Quick Start Guide](#).

Create a Digi project from scratch

To create a Digi project, do one of the following:

1. On the **Welcome** screen, click **New Digi Project**. Or:
2. On the main menu, choose **File > New Digi Project**.

The **Create Digi Project** wizard opens.

1. Select the category your target device belongs to.
2. Click **Next**.
3. Select the platform to run the Python/MicroPython application.
4. Click **Next**.
5. If you are creating a MicroPython project for an XBee platform and want to import any of the existing libraries in your project, check them and click **Next**.

Note This step is optional. You can always add any library after creating the project. See [Import Python libraries and packages](#) for more information.

6. Specify the project location and Python interpreter if you do not want to use the default configuration.
7. Click **Create**.

Import a Digi sample

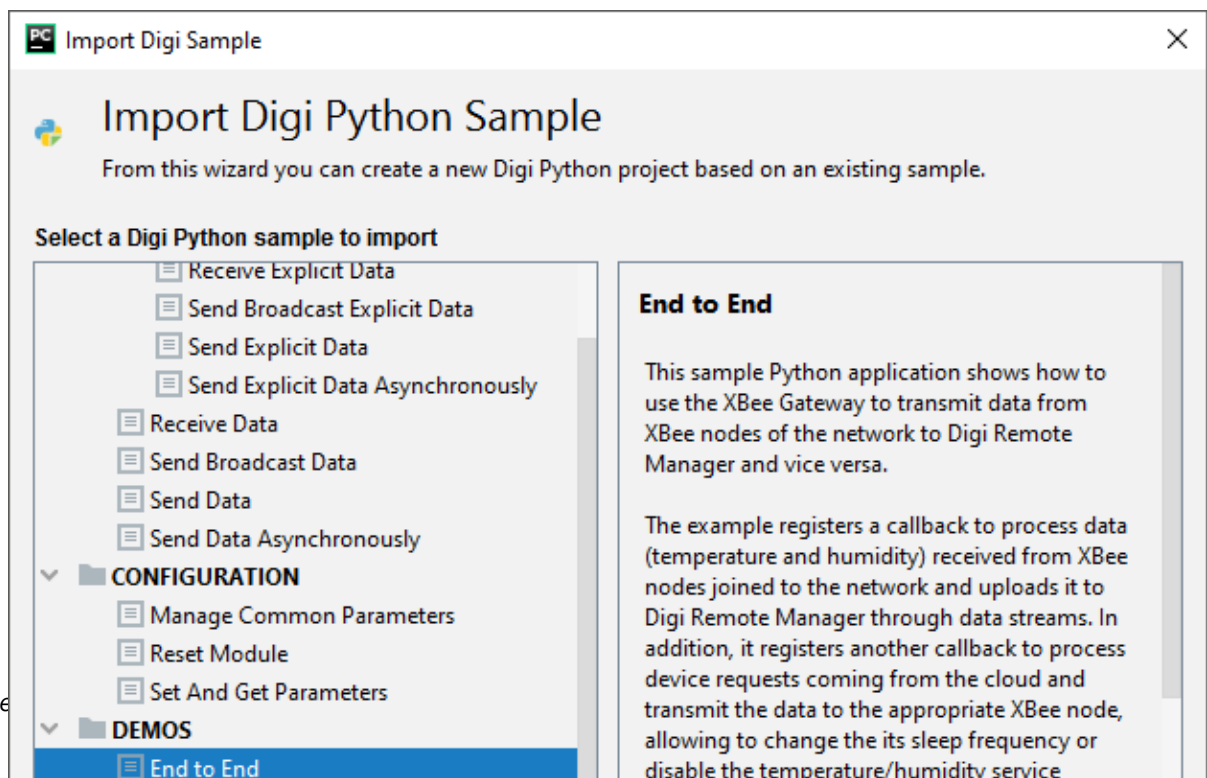
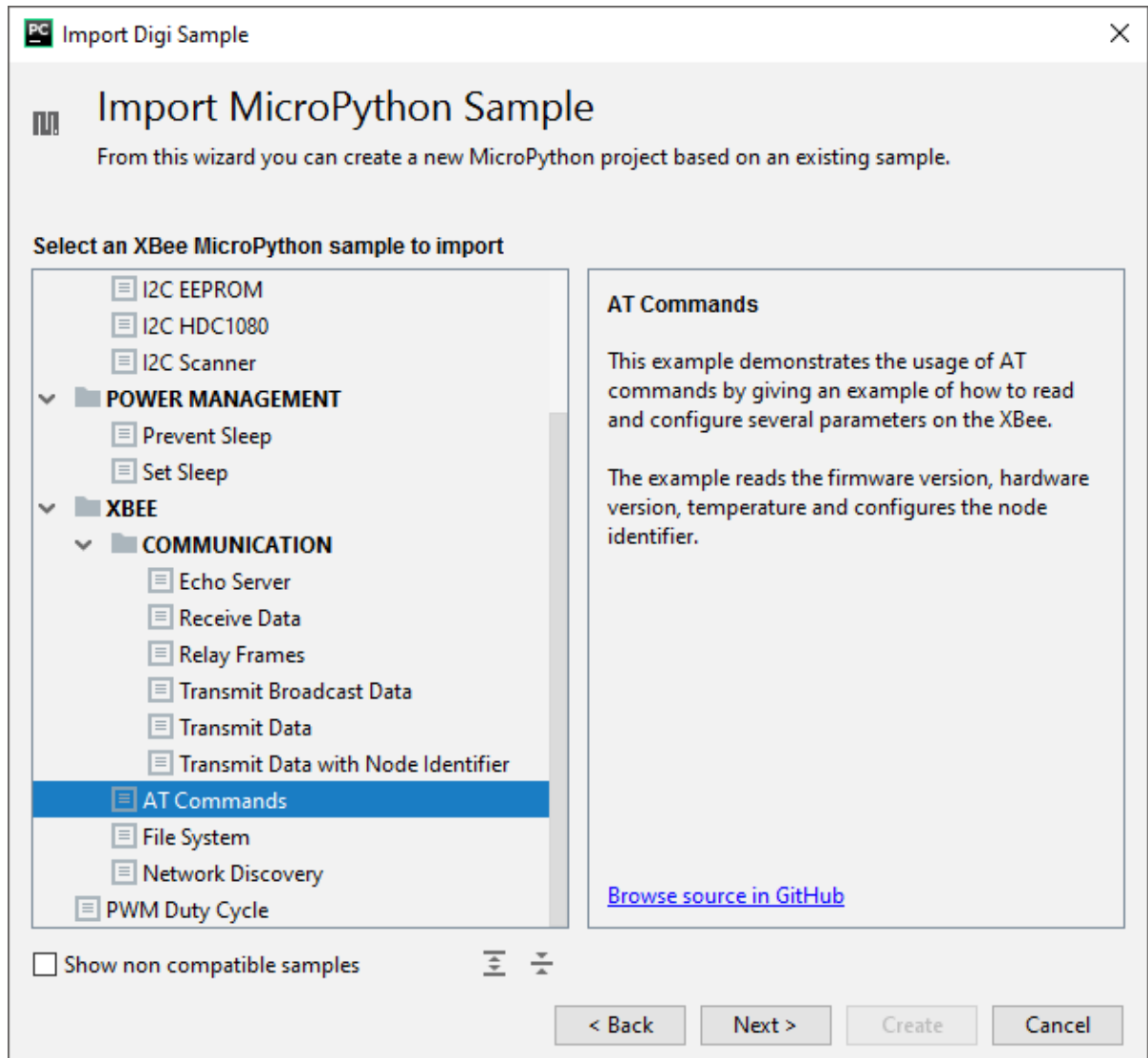
To import a Digi sample project, do one of the following:

1. On the **Welcome** screen, click **New Digi Project**. Or:
2. On the main menu, choose **File > Import Digi Sample Project**.

The **Import Digi Sample** wizard opens.

1. Select the category your target device belongs to.
2. Click **Next**.
3. Select the platform to run the Python/MicroPython application.
4. Click **Next**.
5. Select the sample to import and click **Next**.

Note You can see detailed information about each sample by clicking it. If you want to see the sample source, click **Browse source in GitHub**.



- Specify the project location and Python interpreter if you do not want to use the default configuration. Click **Create**.

Write your code

When you have a project, either created from scratch or imported from a sample, you can add as many Python/MicroPython or other files as you want. All the files and directories created in the project are uploaded to the Digi device.

Note By default, when building:

- MicroPython files are compiled.
- Python files are not compiled.

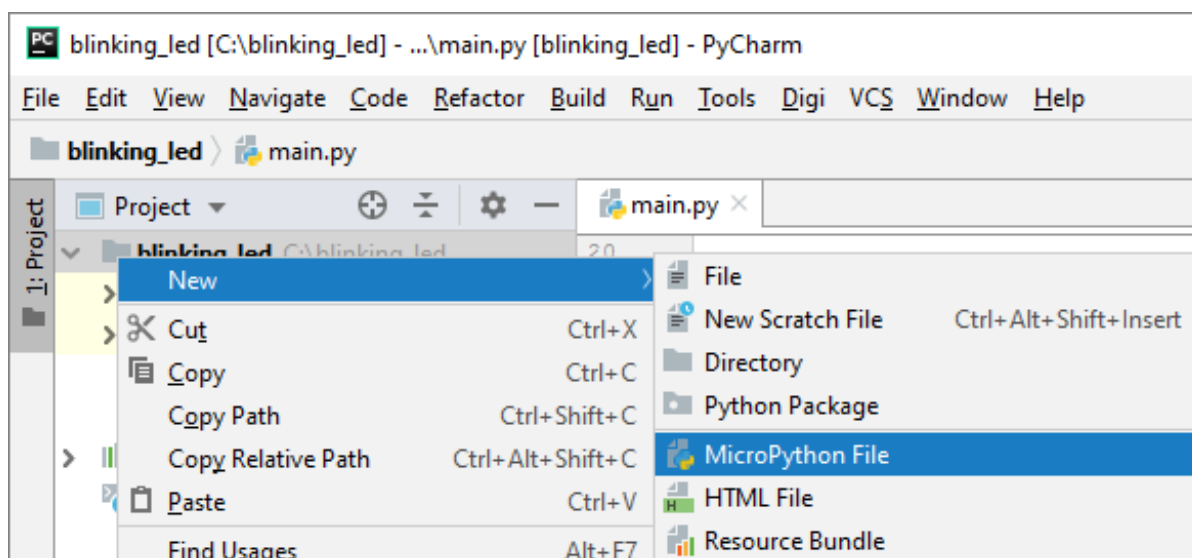
The rest of files and directories are transferred as they are.

Compiled files are smaller, and allows to optimize space in the device file system.

You can enable/disable the source compilation from the project preferences.

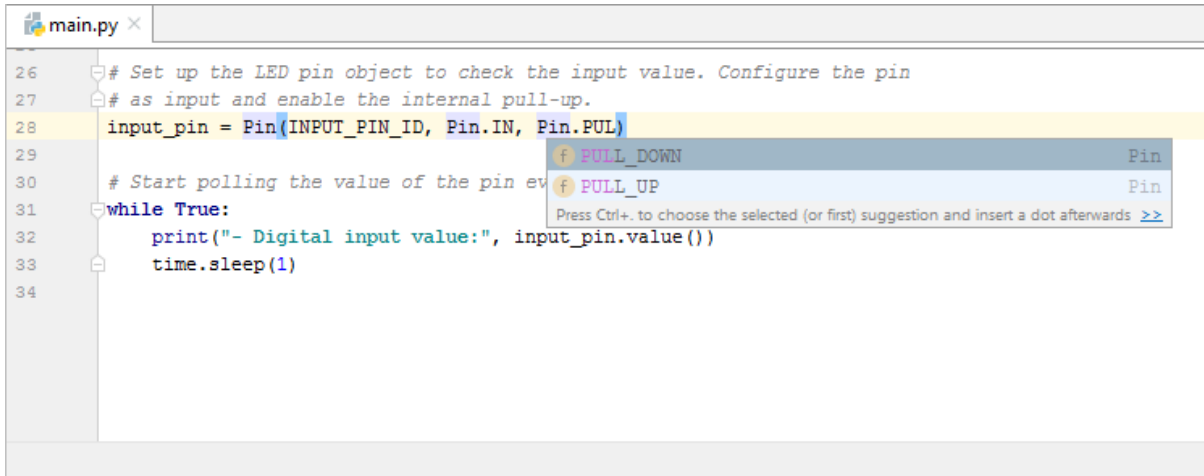
See [Enable/disable Python/MicroPython sources compilation](#).

To create new files or directories, right-click the project view and select **New > MicroPython File** or **New > Directory**.



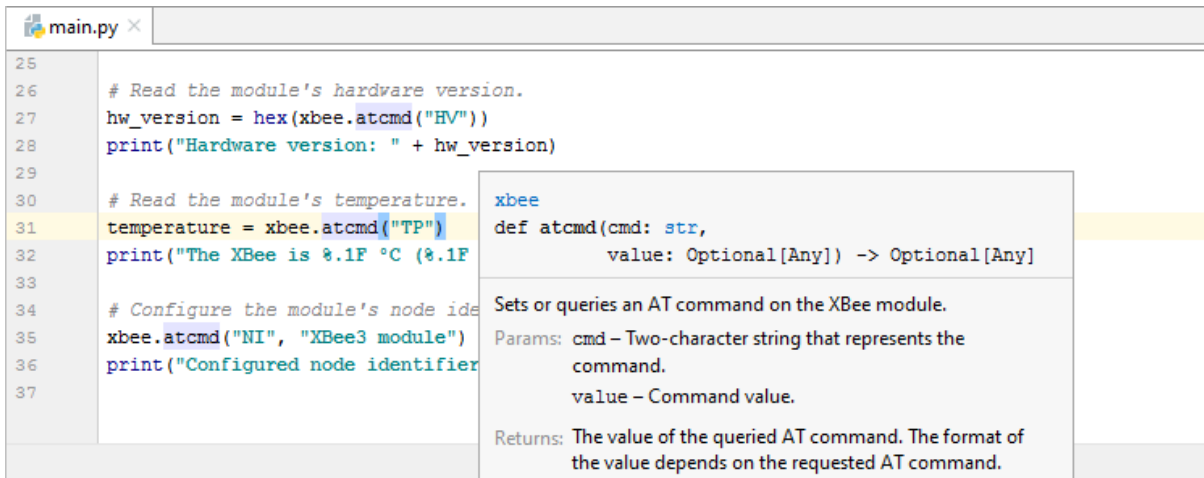
In addition, the Digi XBee PyCharm IDE Plugin offers you some features and help while you are writing your code in the PyCharm editor:

- Code completion.** As you write the code, the plugin auto-completes and suggests method, constant, and class names available in the Digi Python/MicroPython ecosystem.



- **Quick documentation.** The plugin displays a quick documentation popup when you use any method or class, describing its signature, parameters, and return values.

Note The quick documentation popup is disabled by default in the IDE. To enable see [Show the quick documentation popup](#).



- **Syntax and error highlight.** As with other languages, Python/MicroPython keywords are highlighted as well as any error in the code, explaining the error and how to fix it.

```

58     print("Displaying log...\n")
59
60     # Open again the log file to read its contents.
61     with uio.open(LOG_FILE) as log:
62         while True:
63             line = log.readline()
64             if not line:
65                 break
66             print(line)
67
68     print("\nLog file read")
69

```

with uio.open(LOG_FILE) as log > while True

Import Python libraries and packages

The plugin allows the import of specific libraries and packages in your project to extend the functionality of the Digi Python/MicroPython API within the device. Depending on the target platform you choose, you can import XBee MicroPython libraries for XBee modules, or Python packages for other Digi devices such as gateways, routers or ConnectCore modules.

XBee MicroPython libraries

The XBee MicroPython libraries are modules that Digi writes or modifies to address differences between XBee and other MicroPython platforms. There are two possible ways to import an XBee MicroPython library:

- Import libraries at project creation time.

When creating a new MicroPython project using the **New Digi project** wizard, its third step allows the selection of the libraries for the project. Once the project is created, libraries are copied to the **lib** project folder and the library imports are automatically added to the **main.py** file.

- Import libraries at any time.

Once a project is created, you can import one or more XBee MicroPython libraries. Use these steps:

1. Select the **Digi > Import Libraries** menu option to open the **Import Libraries** dialog.
2. Select the libraries you want to import and click **OK**.

Note Libraries are filtered to list only those compatible with the XBee platform selected when creating the XBee MicroPython project. If you want to see all the available libraries select **Show non compatible libraries** option.

3. Libraries are copied to the **lib** folder of the project and the library imports are automatically added to file **main.py**.

Python packages

When working with a Digi Python project, you can add and use any standard Python package from the Python Package Index (PyPI) which is compatible with your target device. To do so:

1. Create a file called **requirements.txt** in the root folder of your project—if not already created.
2. Add the desired Python packages to the file, one per line.
3. If they are not already installed on your local Python environment, PyCharm asks you to install them. Click **Install requirements**.

Note Packages are installed on the target device when you launch the application, so the device must have access to the Internet at that moment.

Build and run the project

Once you have written your Python/MicroPython application or imported an existing sample, you can run and test it in a Digi device.

Requirements


Before running your application, make sure your computer can communicate with the target device:

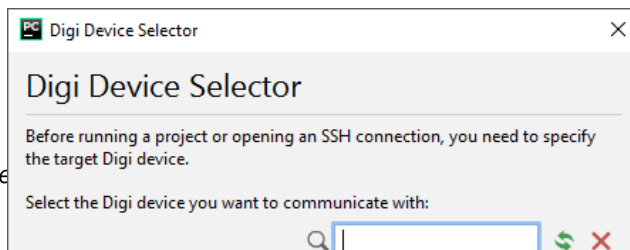
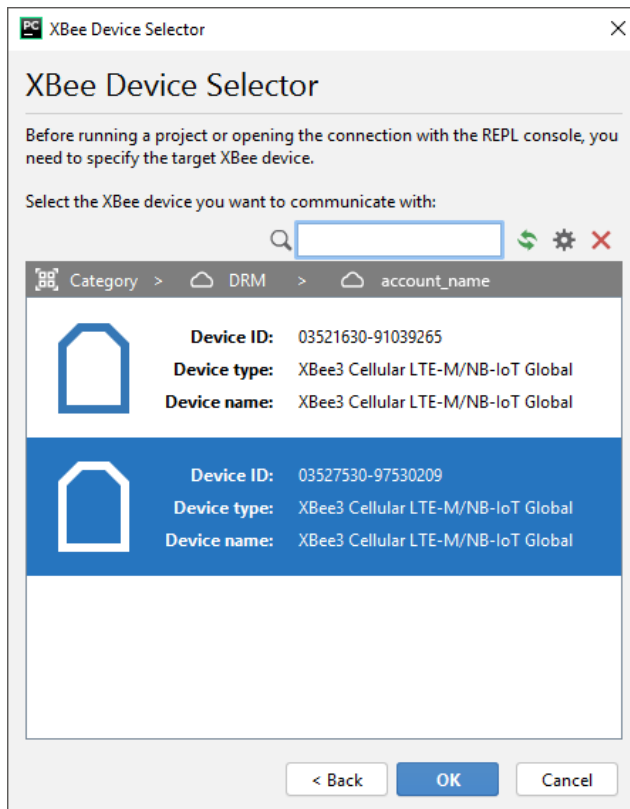
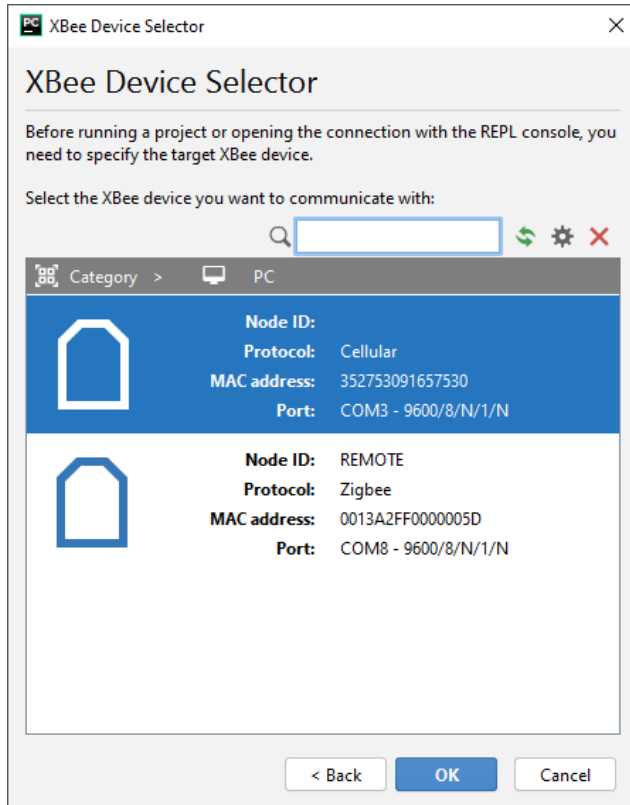
- If your device is an XBee 3 RF Module—Zigbee, DigiMesh or 802.15.4—it must be physically attached to your computer.
- If your device is an XBee 3 Cellular Modem, it can be either physically attached to your computer or added to your Digi Remote Manager account.

Note If you want to use a device connected to Digi Remote Manager, you need to add your Digi Remote Manager account in **Digi > Digi Remote Manager Accounts** if not previously done.


- If your device is not an XBee, it must have the SSH service enabled and be on the same network as the computer.

Device selection and run

1. Click **Run**  on the Navigation bar, or right-click the project folder and choose **Run '<project_name>'**. The Python/MicroPython source files build automatically.
2. If you have not selected your device before, choose first one of the available categories for your project type:
 - a. **Local PC**. It includes all XBee devices that are directly attached to your computer. This option only appears when working with MicroPython projects.
 - b. **Digi Remote Manager**. It includes all XBee 3 Cellular devices that are added to any of the configured Digi Remote Manager accounts. This option only appears when working with MicroPython projects for XBee Cellular devices and if you have configured at least one Digi Remote Manager account.
 - c. **LAN XBee Gateways**. It includes all XBee Gateways that are on the same local network as your computer. This option only appears when working with Python projects.
3. Click **OK**. The plugin starts the search of Digi devices.



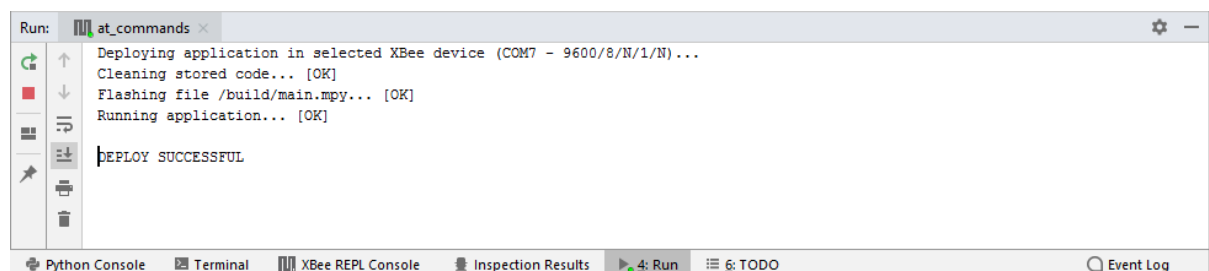
Note If your device does not appear on the list:

- If your XBee device has a special serial port configuration, click **Discovery settings**  and specify its serial configuration.
- For other Digi devices:
 - Ensure that your device has the mDNS service enabled and is on the same network as the computer.
 - Or click the link **Click here to add it manually** to specify its IP address, port, username, and password.

4. Depending on the type of your project, do the following:
 - a. MicroPython: If the selected XBee device is not working in **MicroPython mode**, a dialog asks you to enable it in order to improve the user experience. We highly recommend doing this in test phase and when your application prints output in the console.
 - b. Python: If you did not enter the device credentials, a dialog asks you to enter them. Type the device user and password and click **OK**.

Note If your Digi device requires credentials for the connection and you want PyCharm to remember them, select the **Remember user and password** option and make sure PyCharm has enabled the passwords storage in **Preferences > Appearance & Behavior > System Settings > Passwords**.

5. The Python/MicroPython application is uploaded to the selected device and executed automatically.



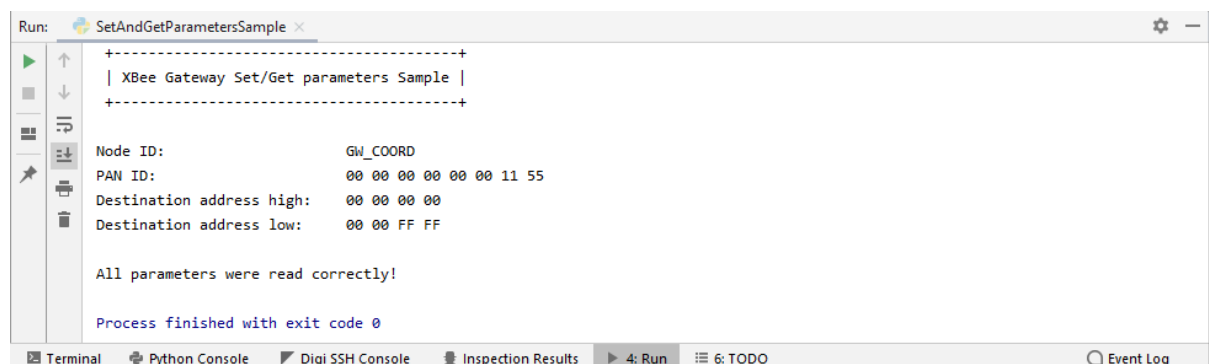
Run: at_commands

```

Deploying application in selected XBee device (COM7 - 9600/8/N/1/N)...
Cleaning stored code... [OK]
Flashing file /build/main.mpy... [OK]
Running application... [OK]

DEPLOY SUCCESSFUL
  
```

Python Console | Terminal | XBee REPL Console | Inspection Results | Run | TODO | Event Log



Run: SetAndGetParametersSample

```

+-----+
| XBee Gateway Set/Get parameters Sample |
+-----+

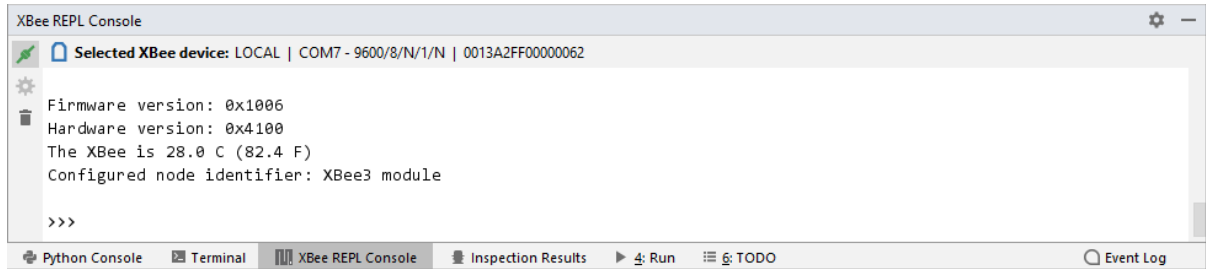
Node ID:           GW_COORD
PAN ID:            00 00 00 00 00 11 55
Destination address high: 00 00 00 00
Destination address low:  00 00 FF FF

All parameters were read correctly!

Process finished with exit code 0
  
```

Terminal | Python Console | Digi SSH Console | Inspection Results | Run | TODO | Event Log

For MicroPython projects, if the XBee device is in MicroPython mode and the application prints any message in the console, you can see it in the **XBee REPL Console**.



Note For more information about the REPL Console, see [Use the console to communicate with your device](#).

Use the console to communicate with your device


Another extension of the plugin is the console, which you can use it to communicate and interact with your Digi device. Depending on your target platform, you can use:


- XBee REPL console for XBee modules
- SSH console for other Digi devices such as gateways, routers or ConnectCore modules.

XBee REPL console

Use this console to communicate with XBee devices connected to your computer—if they are in MicroPython mode—see the output of a running application, perform quick tests and upload code to flash.

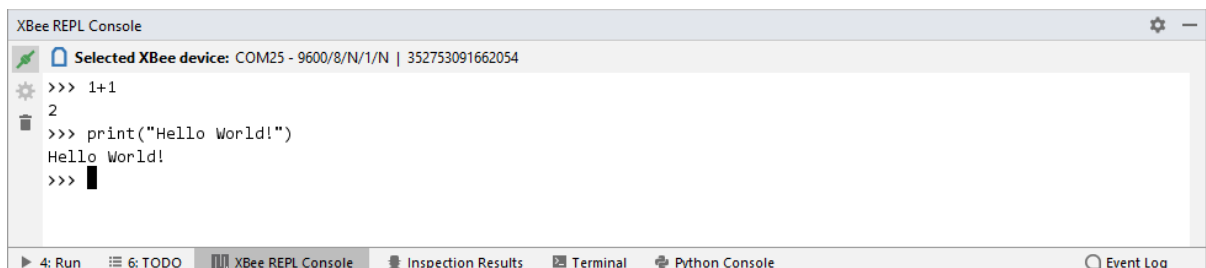
To work with the REPL console:

1. Click **XBee REPL Console** in the **Tool Window** bar or go to **Digi > Open REPL Console**.
2. Click the **Connect** button .
3. If you did not select your XBee device before, choose one from the list of discovered devices and click **OK**.




Note If your XBee device has a special serial port configuration and does not appear on the list, click the **Discovery settings** button  and specify its serial configuration.

4. If the selected device is not working in MicroPython mode, a dialog asks you to enable it. Click **Yes**.

Once connected, you can send any command to the device. It evaluates the input and returns a result.



Note For more information about the available commands on the REPL console, see [REPL \(Read-Evaluate-Print Loop\) examples](#).


Click  to disconnect the console,  to discover new XBee devices, or  to clear the contents of the console.

Digi SSH console

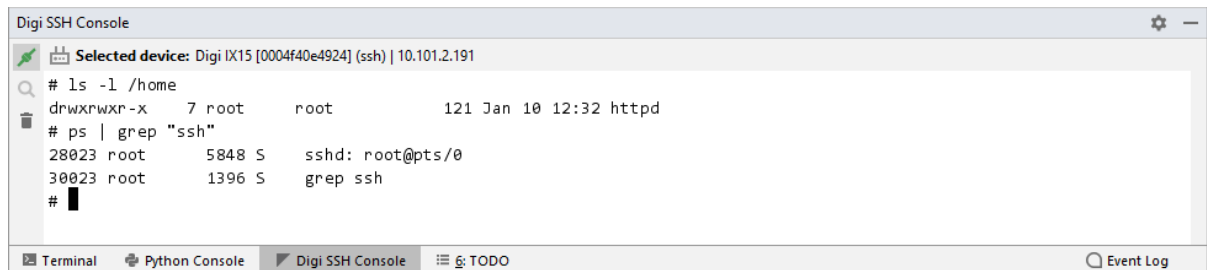
Use this console to communicate with Digi devices over SSH and execute commands remotely.




Note Your device must have the SSH service enabled and be on the same network as the computer.

To work with the SSH console:

1. Click **Digi SSH Console** in the **Tool Window** bar or go to **Digi > Open Digi SSH Console**.
2. Click the **Connect** button .
3. If you did not select your Digi device before, choose one from the list of discovered devices and click **OK**.
4. If you did not enter your device credentials, a dialog asks you to enter them. Type the device user and password and click **OK**.

Once connected, you can execute any command in the device.

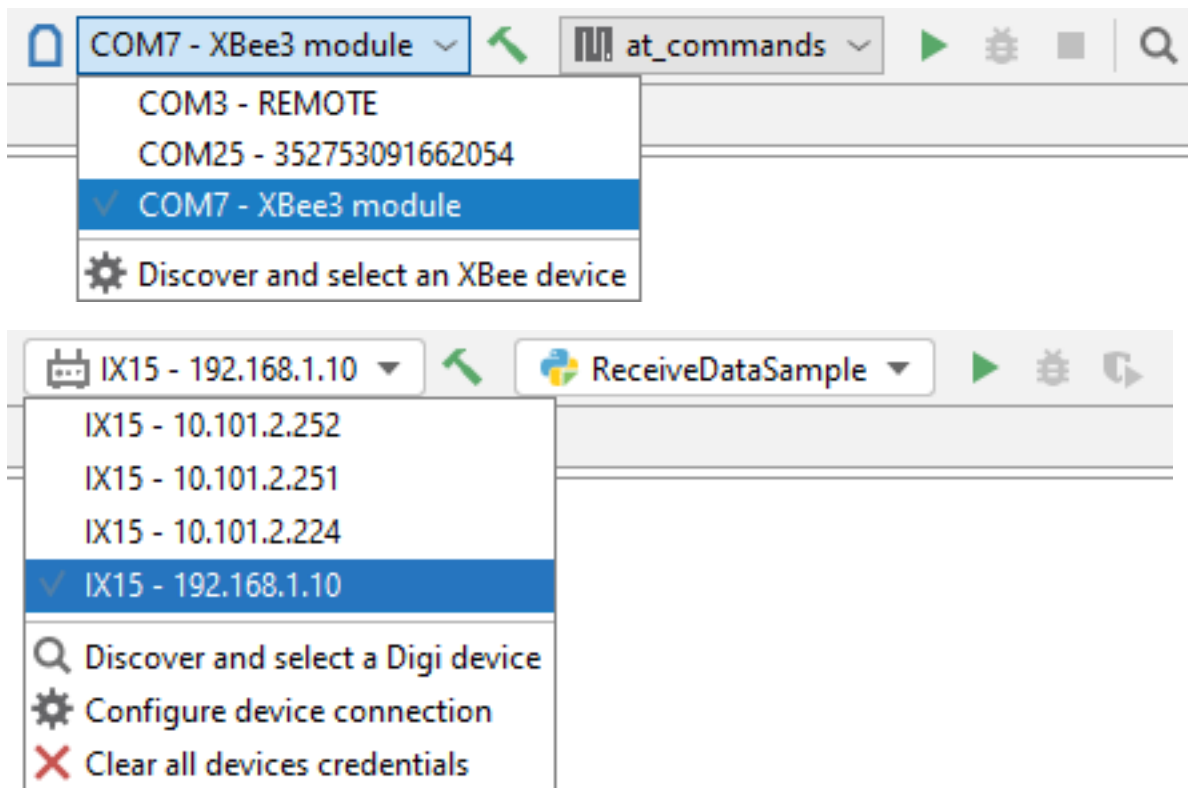


Click  to disconnect the console,  to discover new devices, or  to clear the contents of the console.

How-Tos

Change the selected device

When you run a Python/MicroPython application or open the console, you have to select a device to communicate with. You can change this device or discover new ones at any time from the Navigation bar.



Do not automatically run MicroPython applications at module startup

By default, the plugin configures the XBee device to automatically run the flashed MicroPython application every time it boots. This behavior can be changed in the project's run configuration (**Run > Edit configurations...**). Once there, uncheck the **Automatically run MicroPython application at module startup** option.

Show the quick documentation popup

One of the most interesting features of the plugin is that it helps you while you code with complete documentation about modules, methods, and classes. This option is disabled by default in PyCharm.

To enable it:

1. Go to **File > Settings > Editor > General > Code Completion**.
2. Check the **Show the documentation popup** option.

You can also enable the quick documentation on mouse move in **Settings > Editor > General > Show quick documentation on mouse move**.

Enable/disable Python/MicroPython sources compilation

Although we recommend compiling your source files before transferring them to the target device, you can enable or disable this feature from the project settings:

- If disabled, Python/MicroPython source files (*.py) are transferred to the device when launching your project.
- If enabled, compiled files are transferred to the device when launching your project.

To enable/disable the compile option:

1. Go to **File > Settings**.
2. Select the **Project: <project_name>** option in the left menu.
3. Select **XBee MicroPython** or **Digi Python** option in the **Project settings** panel.
4. Check or uncheck **Compile source files with 'mpy-cross'** or **Compile Python sources** option.
5. Click **OK**.

Change the associated platform of the project

When you create a new project or import a sample, you have to specify the target platform to launch the application. This configuration helps the Digi XBee PyCharm IDE Plugin to filter the samples and libraries and list only those compatible with the selected platform.

To change the target platform of your project once it has been created:

1. Go to **File > Settings...**
2. Select the **Project: <project_name>** option in the left menu.
3. Select the **XBee MicroPython** or **Digi Python** option in the **Project settings** panel.
4. Choose the new **Target platform**.
5. Click **OK**.

Exclude files or folders from the compilation

PyCharm allows you to exclude any file or folder from the compilation so that they are not transferred to the XBee device. To do so:

1. Go to **File > Settings**.
2. Select the **Project: <project_name>** option in the left menu.
3. Select the **Project Structure** in the Project settings panel.
4. In the **Exclude files** field, type the masks that define the names of files and folders to be excluded, for example, *.pyc. Use ; to separate name patterns, * for any number of symbols, and ? for one.
If the name of a file matches any of these patterns, the file is treated as excluded. If the name of a folder matches a pattern, all the files in this folder and in its subfolders are marked as excluded.

Configure your Digi Remote Manager account

If you want to run a MicroPython application on a remote XBee Cellular added to Digi Remote Manager or communicate with its REPL, you need to enter the credentials of your account. To do so:

1. Go to **Digi > Digi Remote Manager Accounts**.
2. Click **Add DRM Account**.
3. Enter the username and password of your Digi Remote Manager account and click **OK**.

If the credentials are correct, your account will be added to the list and you will be able to work with your remote devices. You can add as many Digi Remote Manager accounts as you want so you can use devices from different accounts.

Note Your credentials are securely stored by PyCharm. In order to use this functionality, make sure PyCharm has enabled the passwords storage in **File > Settings > Appearance & Behavior > System Settings > Passwords**.
