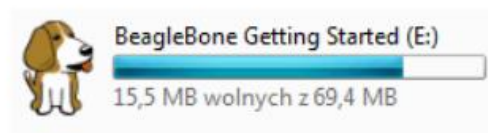


Za pomocą dołączonego kabla łączymy płytkę z komputerem PC. Po podłączeniu powinna zaświecić się dioda PWR (znajdująca się pomiędzy portem ethernetowym, a złączem DC 5V), a diody przy złączu microUSB powinny migać.

Jeżeli wszystko działa poprawnie, na PC, w "Moim Komputerze" powinna być widoczna pamięć masowa "BeagleBone Getting Started".



Gdy ją otworzymy, powinniśmy ujrzeć następujące pliki:

Nazwa	Data modyfikacji	Typ	Rozmiar
uEnv	2013-03-18 12:48	Dokument tekstowy	1 KB
u-boot	2013-03-18 12:48	Plik obrazu dysku	371 KB
START	2013-03-18 12:53	Chrome HTML Do...	17 KB
README.md	2013-03-18 12:53	Plik MD	1 KB
MLO	2013-03-18 12:48	Plik	98 KB
LICENSE	2013-03-18 12:53	Dokument tekstowy	42 KB
ID	2013-03-18 12:53	Dokument tekstowy	1 KB
autorun	2013-03-18 12:53	Informacje instala...	1 KB
Drivers	2013-03-18 12:53	Folder plików	
Docs	2013-03-18 12:53	Folder plików	
App	2013-03-18 12:53	Folder plików	

Zawartość dysku

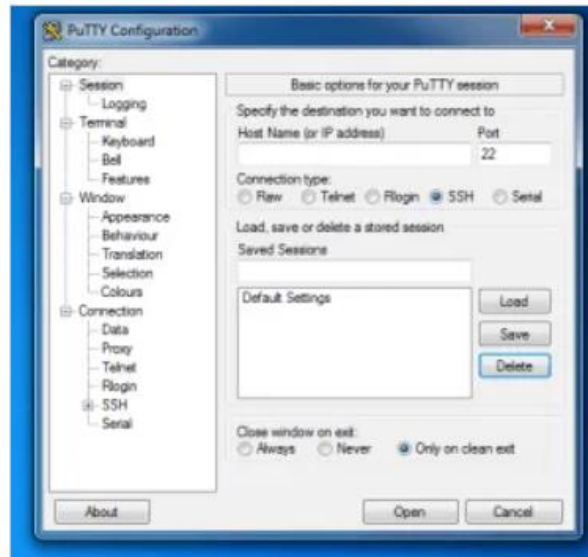
Widzimy, że **Step 1** i **Step 2** podświetliły się na zielono. Oznacza to, że sterowniki zostały zainstalowane poprawnie i nawiązaliśmy połączenie z BBB. Teraz w pasku przeglądarki wpisujemy <http://192.168.7.2> (uwaga nie działa na Internet Explorer) i przechodzimy na stronę minikomputera.



Informacje o BBB

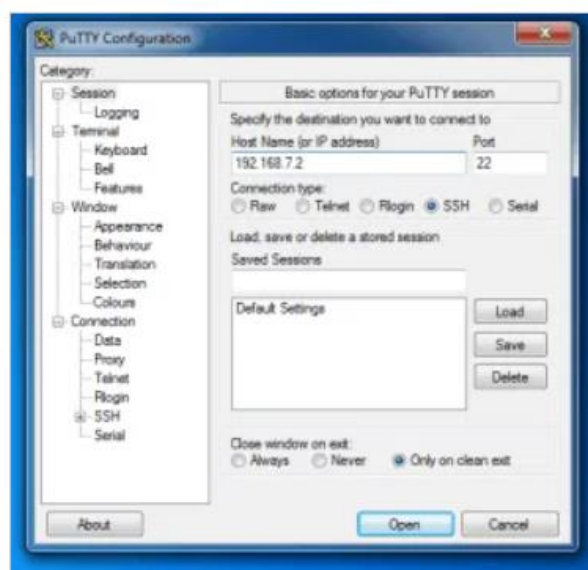
Aby połączyć się z BeagleBone Black poprzez SSH musimy pobrać terminal, który takie połączenie obsługuje. Osobiście używam Putty (możemy go pobrać ze strony <http://www.putty.org/>). Program nie wymaga instalacji.

Po uruchomieniu Putty ujrzymy następujący widok:



Program PuTTY

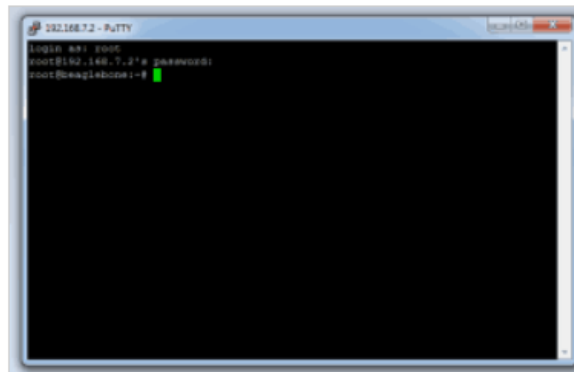
W polu Host Name (or IP address) wpisujemy adres Beaglebone Black (192.168.7.2). W odróżnieniu od innych minikomputerów, np. Raspberry Pi, Beaglebone Black nie wymaga łączenia się za pośrednictwem routera. Zawdzięczamy to temu, że **płytką poprzez USB tworzy wirtualny port ethernetowy**. W polu port wpisujemy 22, natomiast w polu Connection type wybieramy SSH.



Ustawienia

Jeżeli ustawienia się zgadzają, klikamy przycisk "Open". Po chwili pojawi się okno terminala. Musimy podać login. Logujemy się na koncie domyślnym **Root**. W polu wpisujemy **login as : root**.

Dalej zostaniemy poproszeni o podanie hasła do roota. **Domyślnie konto root nie ma hasła**, dlatego klikamy enter.



Połączenie przez SSH

Teraz jesteśmy już zalogowani na Beglebone Black. Możemy tworzyć foldery, pisać programy, kompilować itd. **Gdy chcemy zakończyć** prace z Beglebone Black wpisujemy w konsoli polecenie:

shutdown -h now


Powoduje ono zamknięcie systemu. Innym sposobem na wyłączenie płytki jest dłuższe przytrzymanie przycisku POWER znajdującego się na płycie tuż koło portu ethernetowego. Gdy system zostanie zamknięty wszystkie diody zgasną. Teraz możemy bezpiecznie odłączyć Beglebone Black od komputera.

W następnej części kursu postaramy się przybliżyć i omówić obsługę UART na płycie BBB.

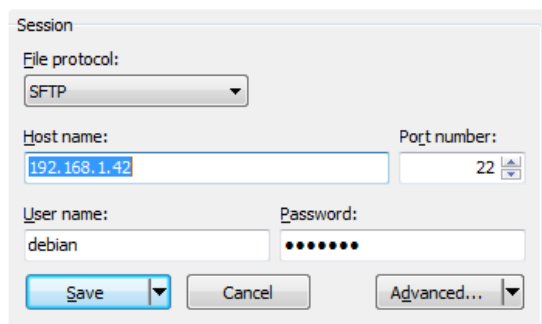
W sali laboratoryjnej 2.7.14 BT zrealizowano przewodową i bezprzewodową sieć (WiFi 2.4 GHz). Aby podłączyć się do sieci przewodowej należy podpiąć komputer za pośrednictwem **zielonej wtyczki RJ-45** (należy pamiętać, aby po skończeniu ćwiczeń komputer ponownie był podłączony do sieci za pośrednictwem **niebieskiej wtyczki RJ-45**). Poniżej przedstawiono dane niezbędne do podłączenia się z bezprzewodową siecią Wi-Fi (**dane mogły ulec zmianie, więc zawsze poproś prowadzącego o aktualne dane**): SSID: lab_2.7.14, hasło: mobilne2018.

Pamiętaj, że w przypadku platformy sprzętowej Raspberry Pi – platforma musi również być podłączona do sieci za pośrednictwem **zielonej wtyczki RJ-45**. Poniżej przedstawiono adresy IP i MAC dostępnych w zestawach platform sprzętowych:

BeagleBone Black (login: debian | password: temppwd):

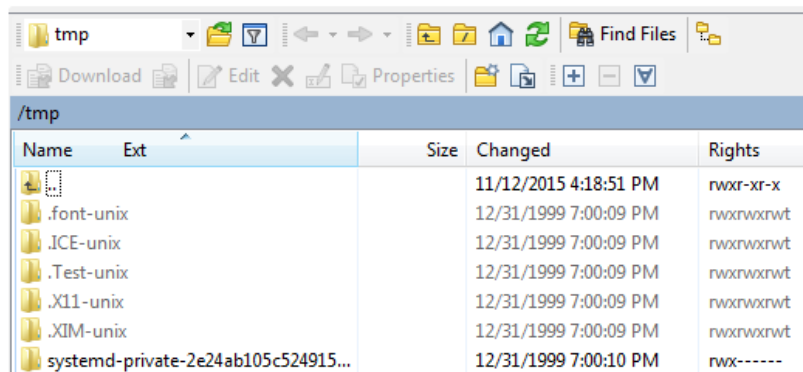
- BBB_1: 192.168.0.111 (MAC: 38-D2-69-4C-9C-16);
- BBB_2: 192.168.0.112 (MAC: 60-64-05-4F-B8-28);
- BBB_3: 192.168.0.113 (MAC: 38-D2-69-53-A7-35);
-  BBB_4: 192.168.0.114 (MAC: 38-d2-69-97-32-0D);
- BBB_5: 192.168.0.115 (MAC: 98-84-E3-B9-D0-24);
- BBB_6: 192.168.0.116 (MAC: 98-84-E3-B7-27-63);
- BBB_7: 192.168.0.117 (MAC: 38-D2-69-3E-3F-87);
- BBB_8: 192.168.0.118 (MAC: 60-64-05-4F-A3-48).

Otwórz WinSCP i edytuj ustawienia sesji w sposób podobny do tego (zamień adres IP na adres IP swojego BBB):



Kliknij przycisk Zaloguj (prawdopodobnie będziesz musiał ponownie potwierdzić pytanie „Kontynuuj połączenie z nieznanym serwerem i dodaj jego klucz hosta do pamięci podręcznej”).

Teraz powinieneś móc używać WinSCP do przesyłania plików pomiędzy komputerem z systemem Windows a BBB!



1. W terminalu SSH połączonym z platformą wpisz:

```
bash
```

[Skopiuj kod](#)

```
mkdir /home/12345_67890
```

Zamień `12345_67890` na swój numer indeksu.

b) Konfiguracja SFTP w PyCharm

1. Otwórz PyCharm i przejdź do **File > Settings > Build, Execution, Deployment > Deployment**.
2. Kliknij **+** i wybierz **SFTP** jako typ połączenia.
3. Uzupełnij pola:
 - **Name:** Nazwij połączenie (np. "Raspberry Pi").
 - **SFTP host:** Adres IP Twojej platformy.
 - **Root path:** `/home/username` (gdzie `username` to nazwa użytkownika na platformie).
 - **Username i Password:** Login i hasło użytkownika na platformie.
4. Połącz się i sprawdź, czy widzisz katalogi swojej platformy.

BeagleBone Black (login: `debian` | password: `temppwd`):

 BBB_4: 192.168.0.114 (MAC: 38-d2-69-97-32-0D);

3. Utwórz plik `z4.py` w nowym folderze

1. Przejdź do katalogu `/home/12345_67890` na platformie.
2. W terminalu SSH wpisz:

```
bash
```

[Skopiuj kod](#)

```
touch /home/12345_67890/z4.py
```

To polecenie stworzy pusty plik o nazwie `z4.py`.

LUB PYCHARM !!!

$$\frac{3,3 - 1,9 [V]}{0,02 [A]} = 70 [\Omega]$$

2. Podłączenie diody LED do BeagleBone Black

1. Podłącz diodę LED i rezystor do płyty BBB:

- Podłącz katodę (krótszą nogę) diody LED do masy (GND) na BBB.
- Anodę (dłuższą nogę) diody LED połącz przez rezystor 68 Ω do wybranego pinu GPIO, np. pinu P8_10.

2. Schemat podłączenia:

- Pin P8_10 (GPIO) \rightarrow Rezystor 68 Ω \rightarrow Anoda LED
- Katoda LED \rightarrow GND (np. pin P8_1 lub P8_2)

Pamiętaj: Sprawdź dokumentację BBB, aby potwierdzić, że wybrany pin obsługuje GPIO i nie jest zarezerwowany do innych funkcji.

a) Instalacja biblioteki `Adafruit_BBIO` (jeśli nie jest zainstalowana)

Na BeagleBone Black najłatwiej jest sterować pinami GPIO przy pomocy biblioteki **Adafruit_BBIO**.

Aby zainstalować bibliotekę, połącz się z BBB przez SSH i wykonaj następujące polecenie:

```
bash
```

 Skopiuj kod

```
sudo apt-get update
sudo apt-get install python3-pip
pip3 install Adafruit_BBIO
```

```

import Adafruit_BBIO.GPIO as GPIO
import time

LED_PIN = "P8_10"
GPIO.setup(LED_PIN, GPIO.OUT)

def short_blink():
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(0.25)
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(0.25)

def long_blink():
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(0.75)
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(0.25)

try:
    while True:
        # Trzy krótkie mignięcia
        for _ in range(3):
            short_blink()

        # Przerwa
        time.sleep(0.5)

        # Trzy długie mignięcia
        for _ in range(3):
            long_blink()

        # Przerwa
        time.sleep(0.5)

        # Trzy krótkie mignięcia
        for _ in range(3):
            short_blink()

        # Dłuższa przerwa po zakończeniu sygnału SOS
        time.sleep(3)

except KeyboardInterrupt:
    # Czyszczenie ustawień GPIO po zakończeniu programu
    GPIO.cleanup()

```

```
import Adafruit_BBIO.GPIO as GPIO
import time

LED_PIN = "P8_10"
GPIO.setup(LED_PIN, GPIO.OUT)

def short_blink():
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(0.25)
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(0.25)

def long_blink():
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(0.75)
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(0.25)

try:
    while True:
        # Trzy krótkie mignięcia
        for _ in range(3):
            short_blink()

        # Przerwa
        time.sleep(0.5)

        # Trzy długie mignięcia
        for _ in range(3):
            long_blink()

        # Przerwa
        time.sleep(0.5)

        # Trzy krótkie mignięcia
        for _ in range(3):
            short_blink()

        # Dłuższa przerwa po zakończeniu sygnału SOS
        time.sleep(3)

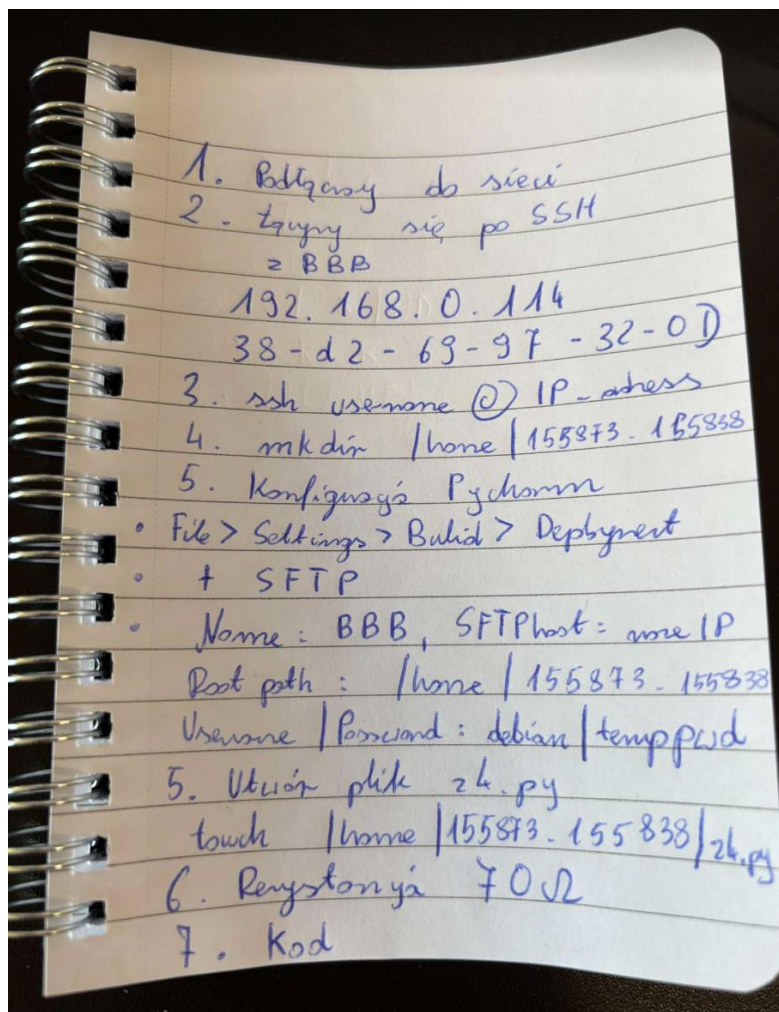
except KeyboardInterrupt:
    # Czyszczenie ustawień GPIO po zakończeniu programu
    GPIO.cleanup()
```



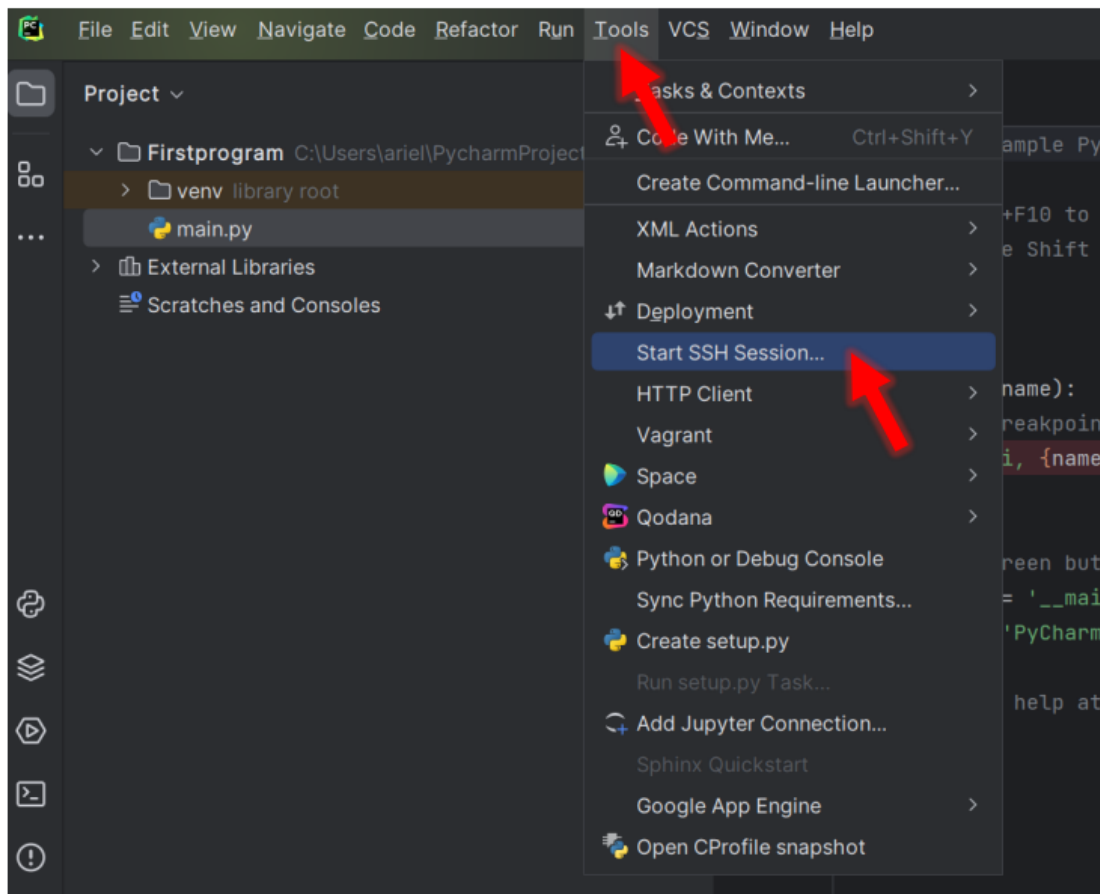
```

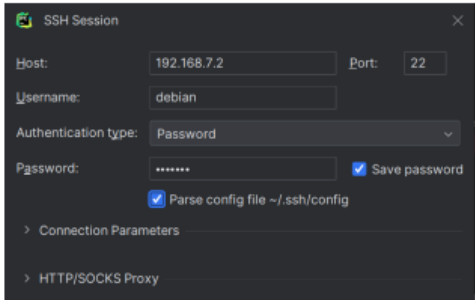
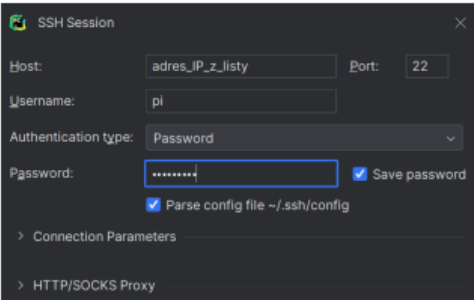
import Adafruit_BBIO.GPIO as GPIO
import time
GPIO.setup("GPIO0_20", GPIO.OUT)
while True:
    for _ in range(3):
        GPIO.output("GPIO0_20", GPIO.HIGH)
        time.sleep(0.25)
        GPIO.output("GPIO0_20", GPIO.LOW)
        time.sleep(0.25)
    time.sleep(0.5)
    for _ in range(3):
        GPIO.output("GPIO0_20", GPIO.HIGH)
        time.sleep(0.75)
        GPIO.output("GPIO0_20", GPIO.LOW)
        time.sleep(0.25)
    time.sleep(0.5)
    for _ in range(3):
        GPIO.output("GPIO0_20", GPIO.HIGH)
        time.sleep(0.25)
        GPIO.output("GPIO0_20", GPIO.LOW)
        time.sleep(0.25)
    time.sleep(3)

```



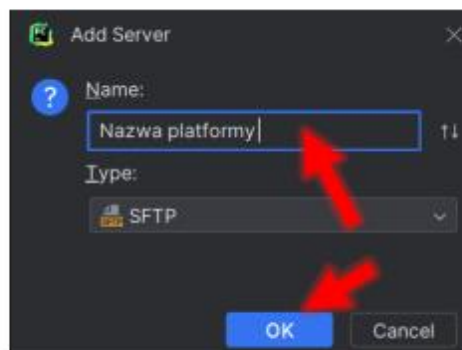
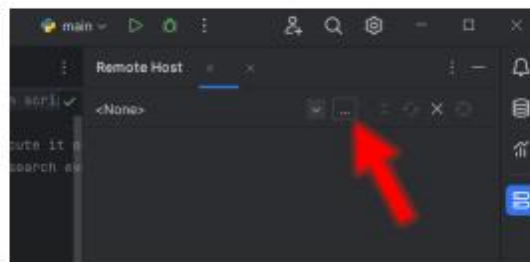
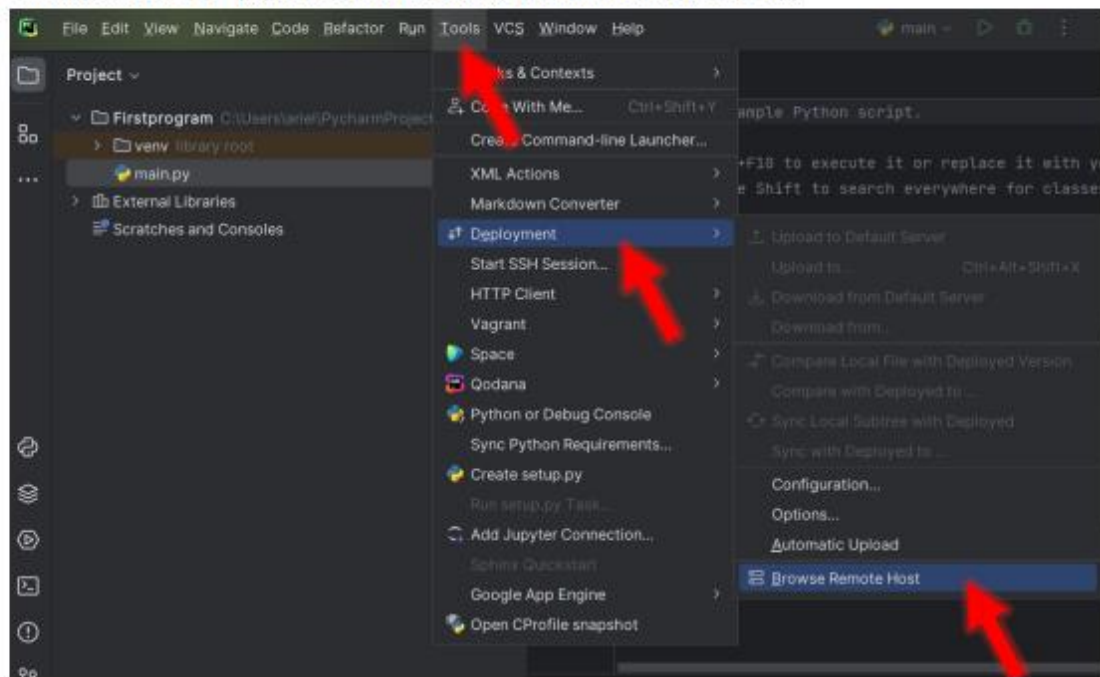
Sposób 2:

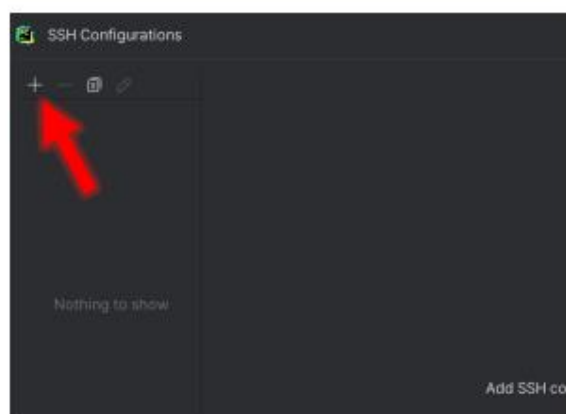
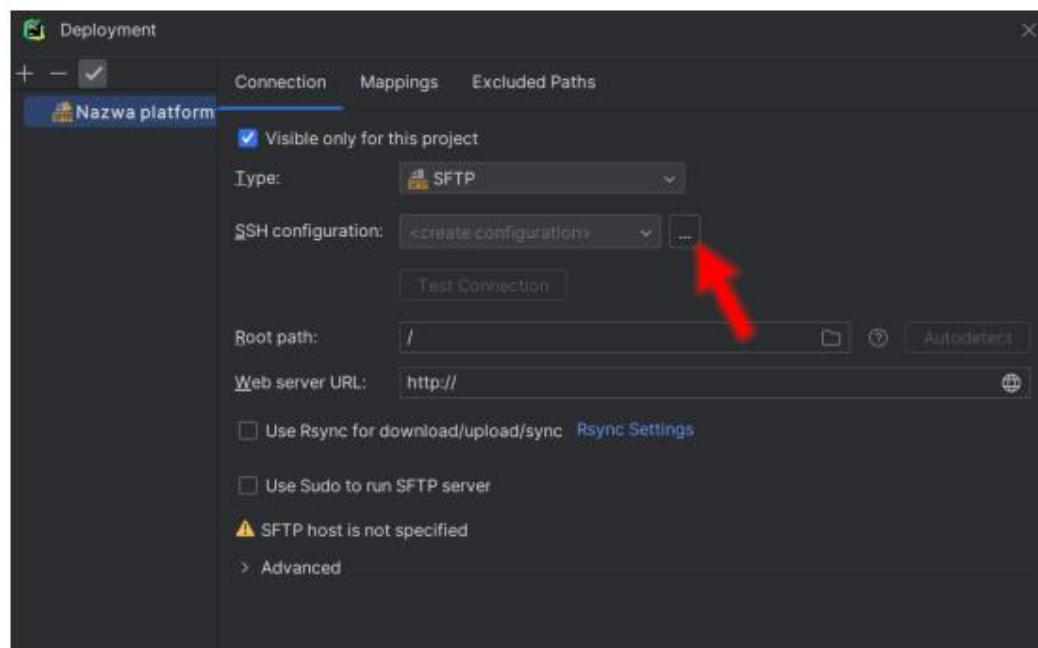


BeagleBone Black (via USB)	Raspberry Pi ¹
Password: temppwd	Password: raspberry
	

Uwaga! Login i hasło mogą się zmienić. Poproś prowadzącego zajęcia o aktualne dane do logowania.

4. Konfiguracja SFTP (wymiana plików między komputerem a platformą)





Uzupełnij okienko analogicznie jak w przypadku konfiguracji SSH.