Intel® Edison – Linux Yocto 3.5

Przygotowanie i dostosowanie do rozwijania aplikacji w języku C/C++

|  |  |
| --- | --- |
| Autor | Bartłomiej Krasoń |
| Wersja | 1 |
| Data modyfikacji | 14-11-2018 |

Spis treści

[Wprowadzenie 3](#_Toc529998823)

[Dane techniczne stanowiska 3](#_Toc529998824)

[Przygotowanie stanowiska 4](#_Toc529998825)

[Podłączenie płytki 4](#_Toc529998826)

[Instalowanie sterowników 4](#_Toc529998827)

[Konfiguracja Edisona 4](#_Toc529998828)

[Flashowanie 4](#_Toc529998829)

[Połączenie szeregowe UART 5](#_Toc529998830)

[Połączenie Wi-Fi\* 5](#_Toc529998831)

[Instalacja oprogramowania 6](#_Toc529998832)

[Przygotowanie Yocto Toolchaina 6](#_Toc529998833)

[Zakładanie projektu C++ 7](#_Toc529998834)

[Wrzucanie aplikacji na Edisona 11](#_Toc529998835)

[Uruchamianie aplikacji na Edisonie 13](#_Toc529998836)

# Wprowadzenie

Niniejszy dokument prezentuje w jaki sposób przygotować się do rozwijania aplikacji na platformie Intel® Edison, z zainstalowanym systemem operacyjnym Linux Yocto w wersji 3.5. Celem dokumentu jest przedstawienie krok po kroku wykonanie wszystkich niezbędnych czynności począwszy od podłączenia płytki do komputera, aż do uruchomienia na niej pierwszego napisanego przez nas programu *Hello World*.

# Dane techniczne stanowiska

Rozdział ten stanowi zbiór wszystkich używanych programów, systemów itp. oraz ich wersji jakie stanowią stanowisko pracy nad zadanych tematem. Udostępniane są również odnośniki umożliwiające pozyskanie programów lub wgląd do dokumentacji technicznych, dodatkowe poradniki.

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa / Wersja | Typ | Opis | Linki |
| Intel® Edison | moduł | moduł obliczeniowy, zawierający w sobie mikroprocesor Intel® Atom™ (dual-core 500MHz) oraz moduły do obsługi Wi-Fi, Bluetooth, USB itp. | [Hardware Guide](https://www.intel.com/content/dam/support/us/en/documents/edison/sb/edison-module-hardware-guide.pdf)  [Product Brief](https://www.intel.com/content/dam/support/us/en/documents/edison/sb/edison_pb_331179002.pdf) |
| Intel® Edison Breakout Board | płytka deweloperska | płytka do której wpina się moduł Edisona, posiada potrzebne złącza, umożliwiające mu komunikację z systemem hostem w celu jego zaprogramowania | [Hardware Guide](https://www.intel.com/content/dam/support/us/en/documents/edison/sb/edisonbreakout_hg_331190006.pdf) |
| Windows / 10 | host system | system operacyjny na którym będziemy pracować, na nim piszemy oprogramowanie, które później umieszczamy na systemie targetowym (docelowym) |  |
| Linux Yocto / 3.5 | target system | system operacyjny, który zainstalowany jest w module Edisona, na nim docelowo będzie działać napisana przez nas aplikacja | [Latest Yocto Image](https://downloadcenter.intel.com/download/27074/Intel-Edison-Yocto-Poky-image) |
| Intel® System Studio / 2018 | środowisko | środowisko tworzenie oprogramowania, w którym będziemy rozwijać nasz program w języku C/C++ | [Pobierz](https://software.intel.com/sites/landingpage/intel-system-studio-configurator/?entitlement=evaluation&edition=ultimate#/) |
| PuTTY / 0.70 | program | program służący do komunikacji między systemem hostowym z systemem targetowym zainstalowanym na Edisonie | [Pobierz](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html) |
| WinSCP / 5.13.4 | program | program służący do przenoszenia plików miedzy systemami hostowym - targetowym | [Pobierz](https://winscp.net/eng/download.php) |
| Intel® Edison Toolchain | SDK | jest to komponent wymagany przez środowisko ISS 2018, umożliwiający cross-kompilację - budowanie programów na systemie innym, niż docelowy system działania aplikacji | [Pobierz](https://downloadcenter.intel.com/download/26997/Intel-Edison-Cross-Compiler-Toolchain?product=83267) |
| Intel® Edison Configuration Tool | sterowniki | program konfiguracyjny instalujący sterowniki na komputerze-hoscie, wymagane do prawidłowego podłączenia płytki deweloperskiej | [Pobierz](https://downloadcenter.intel.com/download/26993/Intel-Edison-Configuration-Tool) |

# Znalezione obrazy dla zapytania Intel edison mini breakout boardPrzygotowanie stanowiska

Intel® Edison

Intel® Edison Mini Breakout-Board

USB/UART (terminal)

USB/OTG (zasilanie + transfer danych)

## Podłączenie płytki

Aby umożliwić pracę z modułem, należy podłączyć jego płytkę deweloperską do komputera na którym znajduje się system hostowy. Na początku moduł należy podłączyć dwoma kablami micro USB typu B – zapewniające komunikację szeregową przez terminal oraz zasilanie i transfer danych.

Szczegółowy opis jak zamontować moduł do płytki deweloperskiej oraz jak podłączyć płytkę do komputera znajduje się pod tym linkiem: <https://software.intel.com/en-us/node/628223>.

## Instalowanie sterowników

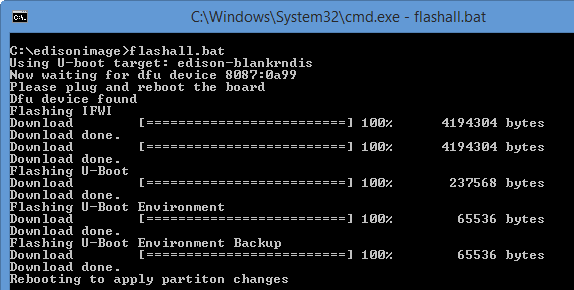
Aby umożliwić prawidłową komunikację między modułem a komputerem-hostem (a dokładniej mówiąc miedzy target systemem a host systemem), należy zainstalować na host systemie sterowniki, obsługujące złącza USB płytki deweloperskiej. Serowniki do pobrania spod [linku](https://downloadcenter.intel.com/download/26993/Intel-Edison-Configuration-Tool).

## Konfiguracja Edisona

Aby zapewnić prawidłowe działanie modułu, należy przed pierwszym przystąpieniem do pracy na nim prawidłowo go skonfigurować.

### Flashowanie

Pierwszym etapem konfiguracji jest ”wyflashowanie” – wgranie najnowszej, świeżej wersji target systemu do modułu Edisona.

* Pobierz najnowszy obraz systemu Linux Yocto spod linku: <https://downloadcenter.intel.com/download/27074/Intel-Edison-Yocto-Poky-image>.
* Rozpakuj zawartość.
* Pobierz ostatnią wersję **dfu-util.exe** i **libusb-1.0.dll** ze strony: <http://dfu-util.sourceforge.net/releases/> (bądź bezpośrednio dla Windows 64-bit: [pobierz](http://dfu-util.sourceforge.net/releases/dfu-util-0.9-win64.zip)).
* Rozpakuj zawartość w tym samym miejscu do obraz systemu.
* Otwórz **Wiersz Poleceń** w lokalizacji gdzie rozpakowałeś pliki .zip.
* Uruchom skrypt **flashall.bat**.
* Gdy zostaniesz poproszony ”*Pleas plug and reboot the borad*”, podłącz płytkę do komputera (złącze USB/OTG).

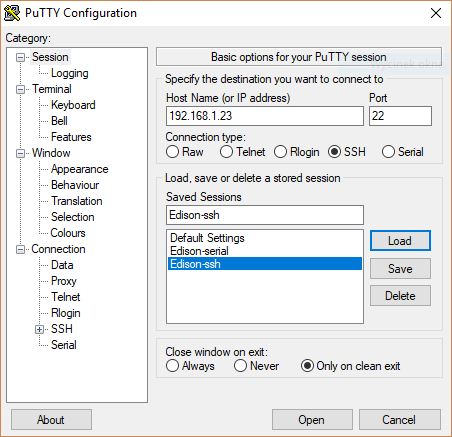
\*Żródło: [https://software.intel.com/en-us/flashing-firmware-on-your-intel-edison-board-windows](https://software.intel.com/en-us/flashing-firmware-on-your-intel-edison-board-windows%20)

### Połączenie szeregowe UART

Aby zapewnić połączenie szeregowe, wymagane jest podłączenie modułu złącza USB/UART do komputera-hosta oraz zasilenie modułu. Do obsługi połączenia posłuży nam program PuTTY. Opis jak wykonać pierwsze połączenia szeregowe za pomocą PuTTY opisane jest pod tym linkiem: <https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows>.

### Połączenie Wi-Fi\*

\*UWAGA – aby zapewnić połączenie z modułem przy użyciu Wi-Fi, płytka uprzedni musi być skonfigurowana poprzez połączenie szeregowe, które opisane jest powyżej.

Opis przeprowadzenia konfiguracji przy użyciu połączenia szeregowego, umożliwiające połączenie Wi-Fi z modułem znajduje się pod linkiem: [https://software.intel.com/en-us/connecting-your-intel-edison-board-using-wifi](https://software.intel.com/en-us/connecting-your-intel-edison-board-using-wifi%20). Po przeprowadzeniu takiej konfiguracji, możliwe jest utworzenie bezprzewodowej sesji **ssh** programem **PuTTY**:

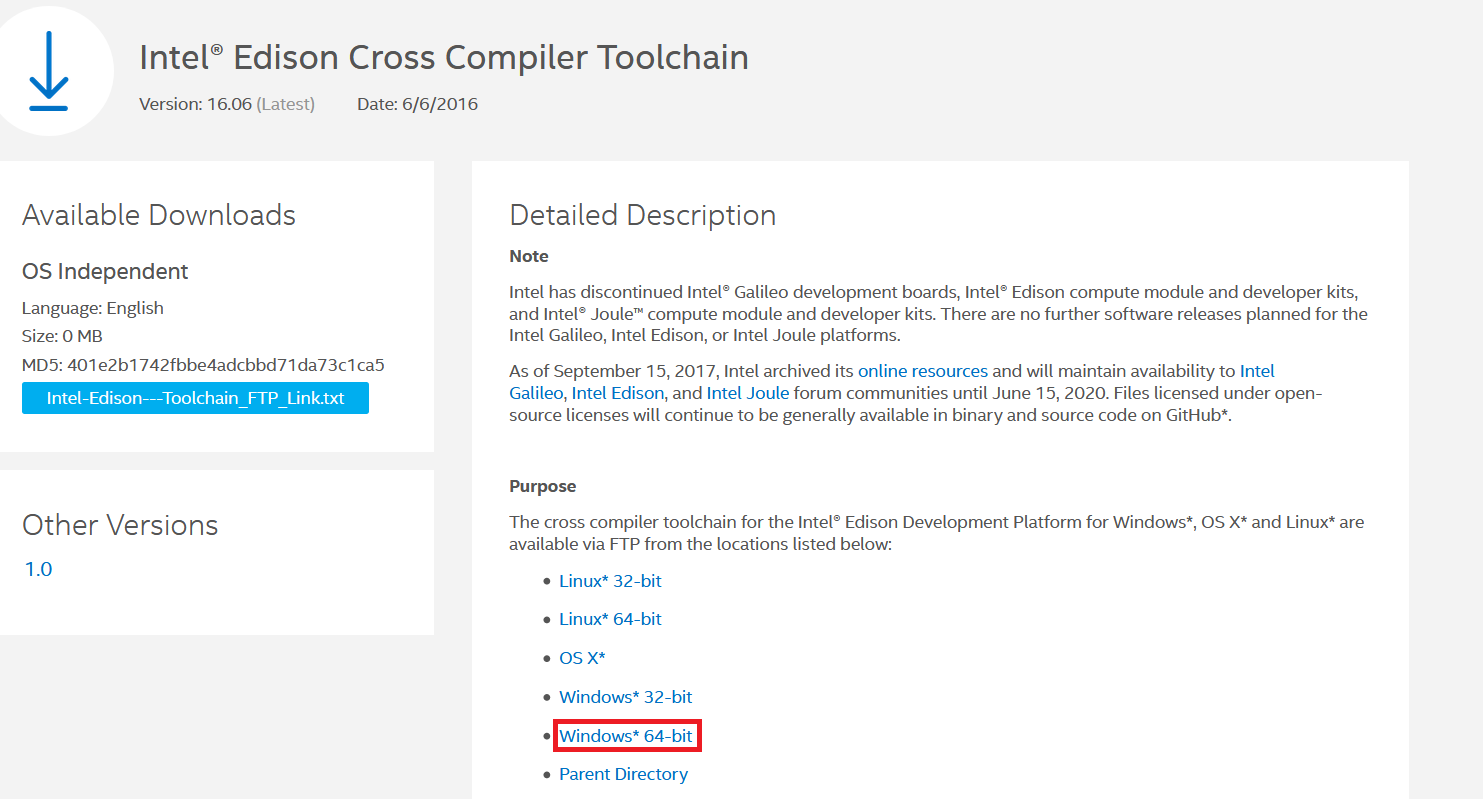
Gdzie **adres IP** ustawiamy na ustalony w naszej lokalnej sieci adres modułu Edisona. Można go uprzednio sprawdzić poleceniem **ifconfig** w terminalu połączenia szeregowego. **UWAGA** – adres IP modułu może się zmieniać za każdy ponownym podłączeniem go do lokalnej sieci Wi-Fi.

## Instalacja oprogramowania

Wymagane oprogramowanie w celu napisania i przeniesienia aplikacji na target system jest następujące:

* Intel® System Studio 2018 - [Pobierz](https://software.intel.com/sites/landingpage/intel-system-studio-configurator/?entitlement=evaluation&edition=ultimate#/) – środowisko do pisania i budowania aplikacji w języku C/C++ uruchamianych docelowo na target systemie.
* WinSCP - [Pobierz](https://winscp.net/eng/download.php) – umożliwi przeniesienie nam pliku wynikowego aplikacji na docelowy target system
* PuTTY - [Pobierz](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html) – umożliwia nam m. In. uruchomienie przeniesionej na niego aplikacji

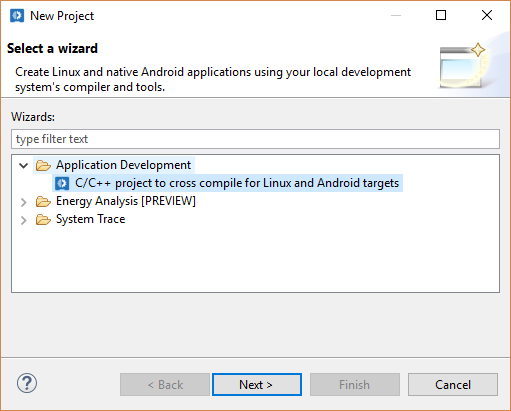
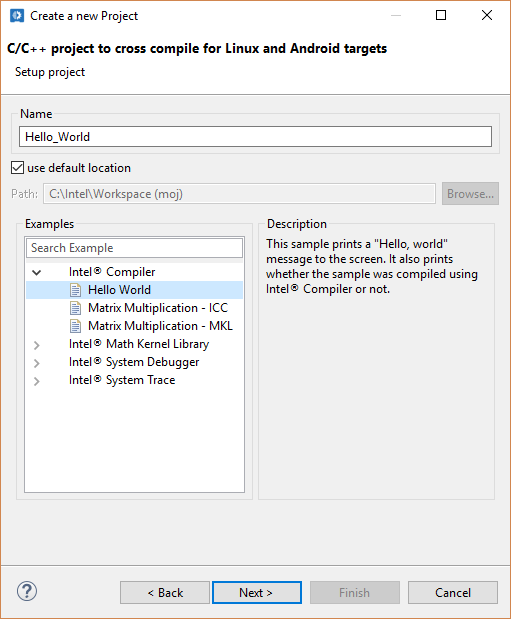
## Przygotowanie Yocto Toolchaina

Rozwój naszej aplikacji obywa się na zasadzie cross-kompilacji. Wymaga ona komponentu nazywanego ”toolchainem”, który umożliwia budowanie aplikacji uruchamianych na innym systemie niż host system komputera, na którym pisana jest aplikacja. Dedykowany cross-compile toolchain dla Edisona pobierzemy spod linku: [https://software.intel.com/sites/landingpage/intel-system-studio-configurator/?entitlement=evaluation&edition=ultimate#/](https://software.intel.com/sites/landingpage/intel-system-studio-configurator/?entitlement=evaluation&edition=ultimate%23/) klikając odpowiadający nam odnośnik:

Następnie pobrany plik, należy otworzyć np. programem WinRar **jako administrator**. Wypakować.

## Zakładanie projektu C++

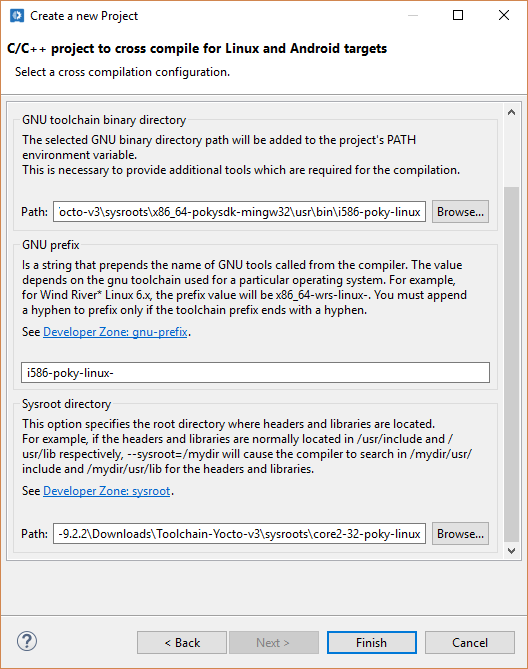
W programie Intel® System Studio 2018:

* Wchodzimy w *File* -> *New* -> *Project.*
* Z folderu *Application Developement* wybieramy *Project to cross compile for Linux and Android targets*.
* Wprowadzamy nazwę naszego pierwszego projektu i wybieramy z pola *Examples* aplikację *Helo World*.
* W następnym oknie wypełniamy ścieżki do pobranego i wypakowanego wcześniej toolchaina:
  + *GNU toolchan binary direktory* -> *Path* ustawiamy na:

**[*lokalizacja wypakowania*]\sysroots\x86\_64-pokysdk-mingw32\usr\bin\i586-poky-linux**

* + *GNU prefix* ustawiamy na: **i586-poky-linux-**
  + *Sysroot direktory* -> *Path* ustawiamy na:

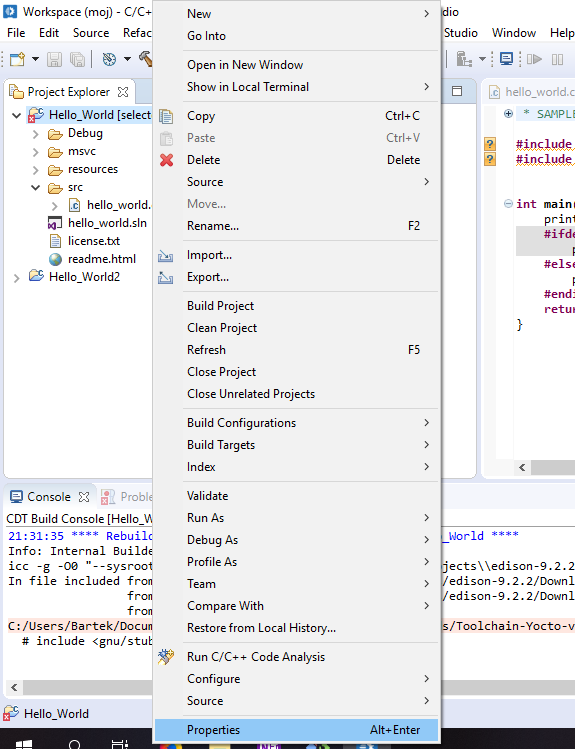
**[*lokalizacja wypakowania*]\sysroots\core2-32-poky-linux**



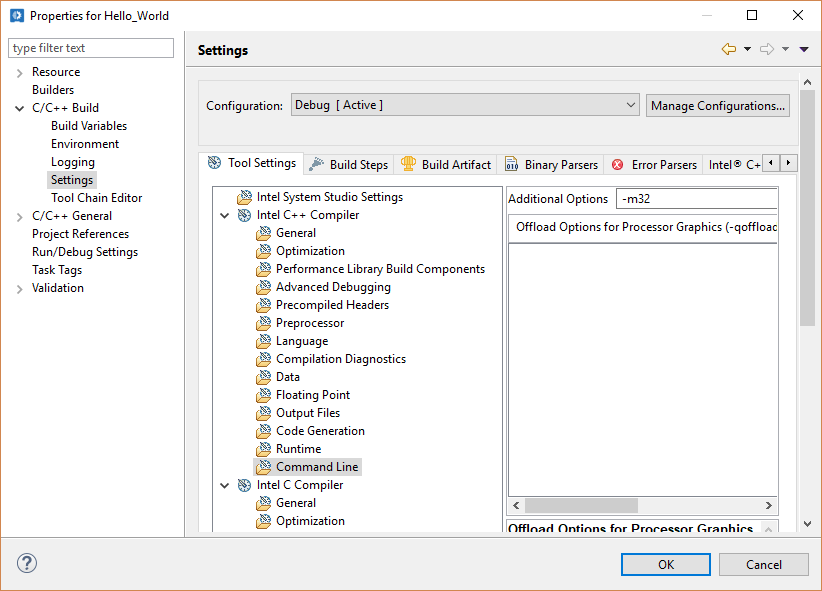
* Klikamy *Finish* i projekt powinien się założyć.

Następnie musimy ustawić opcje buildera, aby budował on aplikację przeznaczoną na architekturę procesora IA-32:

* Na założonym projekcie klikamy PPM i wybieramy *Properties*.



* Wchodzimy w *C/C++ Build* -> *Settings*.
* W zakładce *Tool Settings* wybieramy *Intel C++ Compiler* -> *Command Line*.
  + W *Additional Options* wpisujemy: **-m32**
  + **WAŻNE!** Zjeżdżamy niżej i klikamy *Apply*.
* Tę samą czynność powtarzamy dla *Intel C Compiler* -> *Command Line*.
* Tę samą czynność powtarzamy dla *Intel C++ Linker* -> *Miscellaneous*.



* Po ustawieniu wszystkich opcji klikamy *Ok*.

Po tych czynnościach nasz program powinien się prawidłowo buildować.

## Wrzucanie aplikacji na Edisona

Do tej operacji posłuży nam program **WinSCP** do pobrania spod linku: <https://winscp.net/eng/download.php>. Po uruchomieniu, program należy skonfigurować podobnie jak , w formularzu wypełniając:

*Protokół pliku*: **SCP**

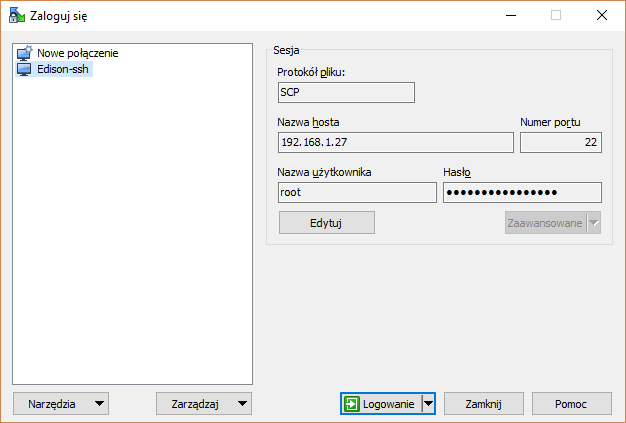
*Nazwa hosta*: **[*adres ip Edisona*]**

*Numer portu*: **22**

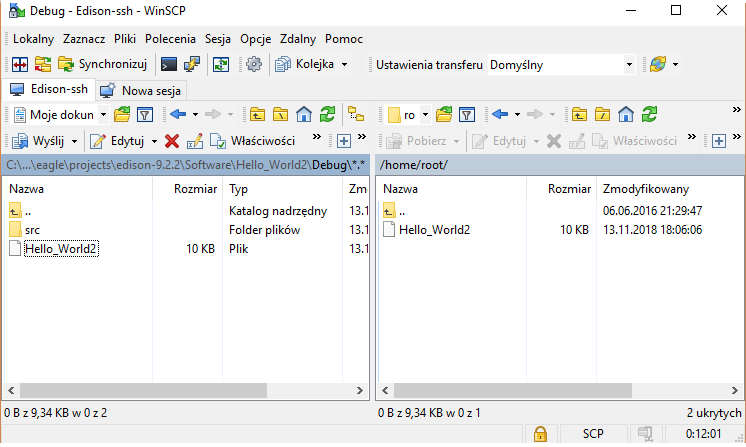
*Nazwa użytkownika:* **Root**

*Hasło*: **[*takie jak ustawione w trakcie konfiguracji Wi-Fi*]**

Następnie klikamy logowanie:



W programie po lewej stronie wyświetlane są pliki znajdujące się na komputerze-hoscie, natomiast po prawej pliki systemu zainstalowanego na Edisonie. Pliki można w łatwy sposób przenosić miedzy systemami, po prosu je przeciągając pod wskazane lokalizacje. Należy przenieść plik wynikowy skompilowanego przez nas programu *Hello World* do katalogu */home/root* Edisona.



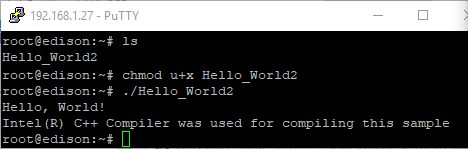
## Uruchamianie aplikacji na Edisonie

Aby uruchomić aplikację potrzebna nam jest sesja terminala na Edisonie. Przykłady tworzenia sesji (odsyłacze):

Po uruchomieniu terminala powinniśmy być pod lokalizacją: **/home/root**.

Aby sprawdzić czy plik został przesłany używamy komendy: **ls**

Następnie, aby można było uruchomić przesłany plik wynikowy, musimy nadać sobie uprawnienia komendą: **chmod u+x [*nazwa pliku*]**

Uruchamiamy program wpisując: **./[*nazwa pliku*]**