



Podstawy Internetu Rzeczy

Instrukcja do zajęć laboratoryjnych

Laboratorium 7

Wprowadzenie do pracy z zestawami Raspberry Pi

Opracował opiekun przedmiotu dr inż. Krzysztof Chudzik

Wydział Informatyki i Telekomunikacji
Politechnika Wrocławska

Wrocław, 2025.09.30 18:49:59

Spis treści

1	Prezentacja zestawów laboratoryjnych z Raspberry Pi	2
2	Zasady obchodzenia się z zestawami laboratoryjnymi Raspberry Pi	2
2.1	Bezpieczeństwo pracy z zestawem	2
2.2	Rozpoczynanie pracy, zgłaszanie usterek i zdawanie zestawów po zakończeniu zajęć	2
2.3	Program testujący zestaw	2
2.4	Elementarne zasady posługiwania się zestawem	3
2.4.1	Rozpoczynanie pracy z zestawem	3
2.4.2	Zakończenie pracy z zestawem	4
3	Praca z zestawem laboratoryjnym	4
3.1	System Operacyjny Raspberry Pi OS	4
3.2	Praca zdalna - protokół SSH	6
3.3	Terminal VNC	7
3.4	Wyłączanie systemu Raspberry Pi OS	10
4	Programowanie Raspberry Pi	10
4.1	Przygotowanie systemu do programowania układów peryferyjnych zestawu	10
4.2	Konfiguracja środowiska do programowania w języku Python	11
4.2.1	Uruchamianie skryptów (programów) języka Python	11
4.2.2	Moduły języka Python do obsługi układów peryferyjnych zestawu	12
5	Wiedzieć więcej o Raspberry Pi	12

Lista zadań

1	Uruchomienie zestawu laboratoryjnego Raspberry Pi	4
2	Uruchomienie programu testującego zestaw w ramach sesji protokołu SSH	7
3	Uruchomienie programu testującego zestaw w ramach sesji VNC Viewer	10
4	Prawidłowe wyłączanie oraz ponowne uruchamianie zestawu	10
5	Uruchamianie skryptów języka Python z linii komend	12

1 Prezentacja zestawów laboratoryjnych z Raspberry Pi

Druga część laboratorium kursu Podstawy Internetu Rzeczy odbywa się z wykorzystaniem zestawu laboratoryjnego wyposażonego w płytke Raspberry Pi i dodatkowe elementy przyłączone do płytki, które będą wykorzystywane w czasie zajęć laboratoryjnych. W dalszej części laboratorium, ten zestaw laboratoryjny będziemy nazywać w skrócie „zestawem Raspberry Pi”, lub po prostu „zestawem”.

Tę część laboratorium rozpocznie zatem prezentacja zestawu Raspberry Pi. Proszę, aby prezentacji dokonał Wykładowca z wykorzystaniem slajdów: „Zestawy laboratoryjne z Raspberry Pi 4 - Wprowadzenie”. Slajdy te są udostępnione w ramach platformy *ePortal* i są podstawowym dokumentem opisującym konfigurację zestawu.

2 Zasady obchodzenia się z zestawami laboratoryjnymi Raspberry Pi

2.1 Bezpieczeństwo pracy z zestawem

- Zestaw zasilany jest poprzez zasilacz sieciowy napięciem stałym 5V.
- Ponieważ zasilacz podłączony jest jednak do gniazda sieci elektrycznej, należy zachować szczególną ostrożność i liczyć się z możliwością jego przebicia elektrycznego.
- W przypadku stwierdzenia wadliwego funkcjonowania lub uszkodzeń zestawu pod względem elektrycznym czy mechanicznym, należy z zachowaniem największej ostrożności odłączyć zasilacz sieciowy od gniazda sieci elektrycznej jedynie wtedy jeśli ten zasilacz jest sprawny.
- Jeśli jednak zasilacz wydaje się być powodem problemów, to nie wolno go dotykać i należy najpierw wyłączyć napięcie elektryczne sieciowe głównym wyłącznikiem zasilania elektrycznego dla całego laboratorium.
- W wypadku jakichkolwiek podejrzeń czy wątpliwości co do bezpiecznego funkcjonowania urządzenia należy wzywać i informować Wykładowcę.

2.2 Rozpoczynanie pracy, zgłaszanie usterek i zdawanie zestawów po zakończeniu zajęć

Zasady organizacyjne pracy z zestawem są identyczne jak dla zestawów Arduino, które opisane zostały w instrukcji do laboratorium pierwszego w rozdziale 3. Proszę przeczytać je ponownie. Zmianom ulegają jedynie informacje o testowaniu i elementarnych zasadach posługiwania się zestawem laboratoryjnym z Raspberry Pi. Są one podane w kolejnych podrozdziałach.

2.3 Program testujący zestaw

Przed uruchomieniem programu testowego, muszą być zainstalowane moduły języka Python zgodnie z opisem w paragrafie 4.2.2. Dla zestawów laboratoryjnych zostało to już zrobione.

Program testujący zestaw tworzy zbiór skryptów języka Python, które umieszczone są w katalogu `/home/pi/tests/`. Aby ustawić ten katalog jako bieżący wydajemy, na przykład, komendę:

```
cd ~/tests
```

Zakładamy dalej, że kolejne komendy wydawane są, gdy katalog `/home/pi/tests/` jest katalogiem bieżącym.

Poszczególne skrypty testują urządzenia, które znajdują się w nazwie skryptu. Przykładowo, `buzzer.py` testuje sygnalizator dźwiękowy - buzzer. Test danego urządzenia uruchamiamy poprzez uruchomienie danego skryptu. Ponieważ mają ustawione atrybuty wykonywalności i w środku zdefiniowany jest interpreter, wystarczy komenda, na przykład:

```
./buzzer
```

Proszę nie pomijać kropki i znaku „/”.

Pewnym wyjątkiem jest tu plik `ws2812.py` testujący linijkę diod programowalnych, który wymaga podniesienia uprawnień. Uruchamiamy go komendą:

```
sudo ./ws2812.py
```

Skrypt `config.py`, który jest plikiem opisującym konfigurację i jest wykorzystywany przez pozostałe skrypty.

Do uruchamiania wszystkich testów łącznie służy skrypt `main.py`. Uruchamiamy go komendą:

```
sudo ./main.py
```

ze względu na test liniжки diod programowalnych.

Test łączny przebiega następująco:

- Test sygnalizatora - Powinien być słyszalny dźwięk wydawany przez sygnalizator.
- Test liniжки programowalnych diod LED RGB WS2812 - Widoczne powinny być różne sekwencje świecenia.
- Test wyświetlacza OLED - Najpierw powinien pojawić się napis, a potem zdjęcie.
- Test niebieskich diod LED w module sygnalizacyjnym - Zapalą się kolejno diody 1, 2, 3, 4, potem zgasną. Następnie zapalą się łagodnie sterowane sygnałem PWM diody 1 i 2, które zgasną.

- Test enkodera - Należy obrócić go o kilka pozycji w dowolną stronę, aż zapalą się na chwilę niebieskie diody LED o numerach od 1 do 4.
- Test czerwonego przełącznika przyciskanego - Przy zapalonych niebieskich diodach LED nr 3 nacisnąć przełącznik czerwony i jeśli działa poprawnie to zapalą się na chwilę niebieskie diody LED o numerach od 1 do 4.
- Test zielonego przełącznika przyciskanego - Przy zapalonych niebieskich diodach LED nr 4 nacisnąć przełącznik zielony i jeśli działa poprawnie to zapalą się na chwilę niebieskie diody LED o numerach od 1 do 4.
- Test termometrów - W terminalu, w którym uruchomiono skrypt, powinny pojawić się odczyty z termometru DS18B20 i układu BME280.
- Test czytnika kart RFID - Zbliżyć kartę do czytnika (po prawej stronie obudowy układów peryferyjnych). Program powinien dokonać trzech odczytów i zakończyć się.

Odstępstwa od tej procedury lub wygenerowanie przez program testowy wyjątków sygnalizuje możliwą awarię sprzętową.

2.4 Elementarne zasady posługiwania się zestawem

Zasada podstawowa: **Nigdy nie włączamy lub rozłączamy żadnych przewodów, wtyczek, itp., jeśli układ jest załączony do sieci elektrycznej i pracuje.** Kto będzie tak robił, będzie traktowany jak świadomie uszkadzający zestaw z pełnymi tego konsekwencjami.

Zanim przystąpimy do konfigurowania i programowania, pamiętajmy, że z tych samych zestawów korzysta wielu Studentów. Kolejne instrukcje laboratoryjne przygotowane są z założeniem, że zestawy posiadają oryginalną konfigurację laboratoryjną. **Zabrania się wprowadzania trwałych zmian konfiguracyjnych systemu operacyjnego oraz jakiegokolwiek oprogramowania w sposób zmieniający jego działanie, w tym działanie interfejsów graficznych. Zmiany takie mogą uniemożliwić prawidłowe przeprowadzenie kolejnych zajęć.** Proszę umożliwić innym Studentom odbycie zajęć, tak jak zostały one zaplanowane. Kto nie zastosuje się do tego i będzie dezorganizował zajęcia, będzie traktowany jako uszkadzający zestawy i zostanie odsunięty od zajęć.

2.4.1 Rozpoczynanie pracy z zestawem

- Przed rozpoczęciem pracy należy sprawdzić, czy zestaw nie jest nigdzie uszkodzony mechanicznie i nic nie wskazuje na uszkodzenia mechaniczne lub elektryczne. W przypadku wątpliwości wezwać Wykładowcę.
- Zestaw jest przygotowany do pracy z wykorzystaniem protokołów VNC i SSH, które dają pełny dostęp do systemu operacyjnego zestawu z komputera stacjonarnego poprzez wykorzystanie terminala VNC (VNC Viewer) i/lub klienta SSH.
- Podłączenie monitora, klawiatury i ekranu wymagane jest tylko w sytuacjach awaryjnych, jeśli do takich dojdzie, i odbędzie się to na wyraźne polecenie Wykładowcy.
- Jeżeli rozpoczynamy pracę, a zestaw nie jest podłączony, należy:
 - Upewnić się, że zestaw na pewno NIE jest podłączony do sieci elektrycznej, nie jest nigdzie uszkodzony mechanicznie i nie ma znamion uszkodzeń elektrycznych (widoczne zwarcia, uszkodzone przewody, wypalone gniazda elektryczne, itp.).
 - Podłączyć w następującej kolejności, jeśli nie są podłączone:
 - * przewód zasilacza sieciowego do zestawu,
 - * przewód Ethernet,
 - * przewód HDMI, klawiaturę i myszkę, tylko wtedy gdy poprosi o to Wykładowca (odłączyć monitor od gniazda sieci elektrycznej na czas podłączania kabla HDMI do niego),
 - Podłączyć zasilacz sieciowy do gniazda sieci elektrycznej i obserwować niebieskie diody sygnalizacyjne. Powinny zapalić się diody 5 i 6.
 - Jeśli tak się nie dzieje, natychmiast odłączyć zasilacz sieciowy i wezwać Wykładowcę.
 - Jeśli diody zapaliły się, należy obserwować chwilę układ, czy nie ujawniły się jakieś uszkodzenia.
 - Układ uruchamia się ponad jedną minutę, więc należy cierpliwie czekać aż do pojawienia się numeru IP na wyświetlaczu OLED.
 - Korzystając z tego numeru IP wykonać połączenie terminalem VNC lub klientem SSH.
 - Wykonać test zestawu programem testującym.
 - Jeśli wszystko funkcjonuje poprawnie, można przystąpić do wykonywania zadań laboratoryjnych.

UWAGA: Za wyświetlanie adresu IP na wyświetlaczu OLED odpowiada usługa (service) `ip-oled`. Gdy zachodzi potrzeba sprawdzenia statusu, wyłączenia usługi (szczególnie wtedy, gdy uruchamiamy inny program korzystający z wyświetlacza OLED) lub włączenia tej usługi, posługujemy się następującymi poleceniami z konsolowymi:

```
systemctl status ip-oled.service
sudo systemctl stop ip-oled.service
sudo systemctl start ip-oled.service
```

2.4.2 Zakończenie pracy z zestawem

- **Nie wolno rozpoczynać od odłączenia zasilacza z gniazda!!!** Kto będzie tak robił, będzie traktowany jak świadomie uszkadzający zestaw.
- Najpierw wyłączyć (zamknąć) system operacyjny, tak aby wszelkie operacje zostały prawidłowo zakończone.
- Zamknięcie systemu wykonujemy w środowisku graficznym lub z linii poleceń tak, jak to opisano w podrozdziale 3.4.
- Odczekujemy około 15 sekund od wydania polecenia o zamknięciu systemu operacyjnego.
- Po upływie czasu 15 sekund, można odłączyć zasilacz sieciowy od gniazda sieci elektrycznej. Po pewnym czasie powinny zgasnąć niebieskie diody sygnalizacyjne numer 5 i 6.
- Po zgaśnięciu diod 5 i 6, **jeśli takie polecenie wyda Wykładowca**, należy w następującej kolejności odłączyć:
 - przewód Ethernet,
 - przewód zasilacza sieciowego od strony zestawu.
 - przewody HDMI, klawiatury i myszki, jeśli były wykorzystywane, (odłączyć monitor od gniazda sieci elektrycznej na czas odłączania kabla HDMI od niego),
- Należy jeszcze raz dokonać oceny stanu zestawu, wtyczek, przewodów, elementów peryferyjnych. Jeśli są jakieś uwagi to należy przekazać je Wykładowcy.
- Jeśli nie ma uwag do zestawu i jego funkcjonowania, to należy, zgodnie z instrukcją Wykładowcy, pozostawić zestaw na stole wyłączony lub zwrócić zestaw w sposób i miejsce wskazane przez Wykładowcę.

UWAGA: Zestaw został tak skonfigurowany, aby nie zapisywał trwale zmian w systemie plikowym. Pliki utworzone przed wyłączeniem lub restartem systemu zostaną utracone. Dlatego własną pracę (skrypty języka Python) należy skopiować na pendrive.

Zadanie 1: Uruchomienie zestawu laboratoryjnego Raspberry Pi

Podłącz i uruchom zestaw zgodnie z procedurą opisaną w podrozdziale 2.4.1.

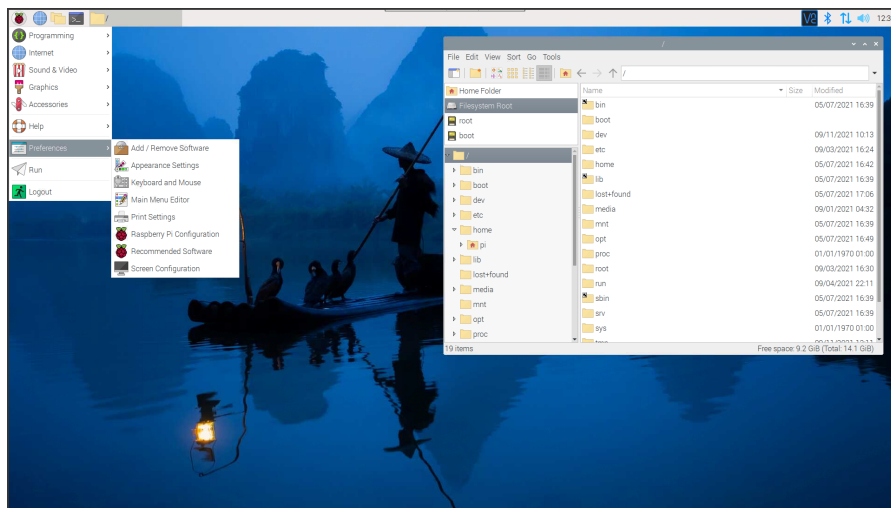
3 Praca z zestawem laboratoryjnym

3.1 System Operacyjny Raspberry Pi OS

Płytką Raspberry Pi wymaga do swojej pracy zainstalowania systemu operacyjnego. Zalecanym systemem jest [Raspberry Pi OS \(link\)](#). Jest on wspierany przez producenta płytki i taki został zainstalowany na zestawach laboratoryjnych.

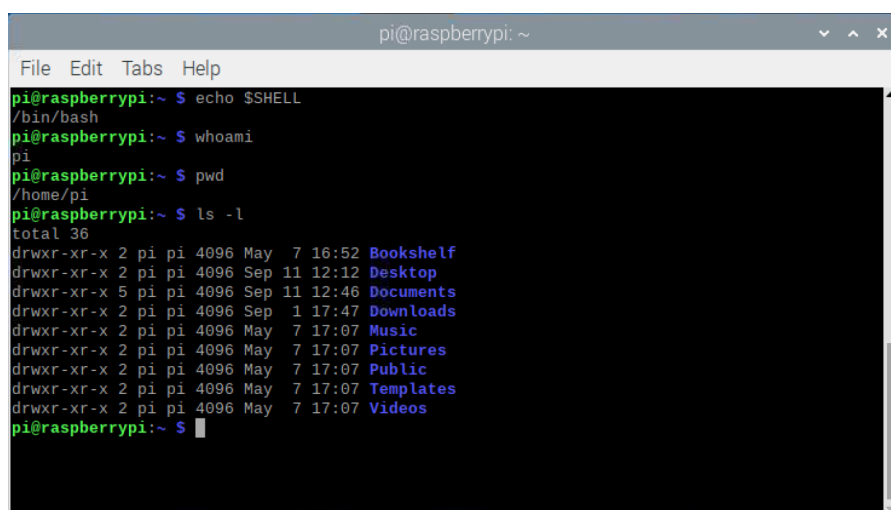
Warto wiedzieć, że system ten, przed zmianą nazwy, nosił nazwę Raspbian. Taka występuje w części dokumentów w Internecie. Jednak, napotkanie nazwy Raspbian w dokumentacji, sugeruje, że dokumentacja może być nieaktualna. Proszę na to zwracać uwagę.

Raspberry Pi OS jest systemem wywodzącym się z dystrybucji systemu Linux [Debian \(link\)](#) i dlatego duża część funkcjonalności Raspberry Pi OS jest zgodna z systemem Debian. Środowisko graficzne niekoniecznie przypomina środowiska graficzne współczesnych systemów Linux. Jego funkcjonalność i wygląd zostały dostosowane do możliwości płytki. Środowisko przedstawia ekran 1.



Ekran 1: Środowisko graficzne Raspberry Pi OS.

W Raspberry Pi OS dostępne jest środowisko tekstowe (linia komend, konsola) systemu Linux (ekran 2).

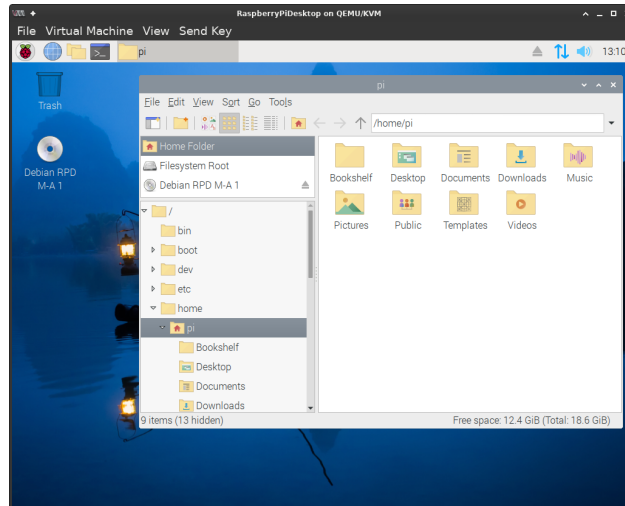


Ekran 2: Środowisko tekstowe w emulatorze terminala systemu Raspberry Pi OS.

Osoby znające środowisko tekstowe Linux, nie będą miały żadnych problemów z pracą w środowisku tekstowym Raspberry Pi OS. Osoby, które nie mają doświadczenia z pracą w środowisku tekstowym Linux, lub mają je mniejsze, proszę o zapoznanie się z materiałami dostępnymi na witrynie linuxcommand.org ([link](#)), w szczególności, z materiałami ze strony [Learning the Shell](#) ([link](#)).

Ponadto, istnieje możliwość uruchomienia jako maszyny wirtualnej systemu [Raspberry Pi Desktop](#) ([link](#)) na komputerze klasy PC lub Mac. Jest to środowisko najbardziej zbliżone do tego, z którym mamy do czynienia na płytkach Raspberry Pi. Nie jest ono jednak identyczne, ponieważ w środowisku wirtualizacyjnym brakuje układów specyficznych dla płytki Raspberry Pi. Stąd też, ograniczone są jego możliwości konfiguracyjne i użytkowe. Nie jest też ono aktualizowane, niestety.

Ekran 3 przedstawia system Raspberry Pi Desktop działający jako maszyna wirtualna.



Ekran 3: Raspberry Pi Desktop.

3.2 Praca zdalna - protokół SSH

Protokół SSH pozwala przyłączyć się do środowiska tekstowego (konsoli, terminala). Zapewnia on bezpieczne uwierzytelnianie i komunikację. Oprogramowanie klienckie protokołu jest wbudowane w wiele systemów operacyjnych, lub łatwo je pozyskać i zainstalować.

Przykładowo, system Windows posiada oprogramowanie klienckie protokołu SSH jako element wbudowany, o czym nie wszyscy wiedzą. Ekran 4 prezentuje krótką sesję wykonaną klientem protokołu SSH z systemu Windows do Raspberry Pi OS.

```

Microsoft Windows [Version 10.0.19043.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\stud>ssh
usage: ssh [-46AcFgKkWNqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]

C:\Users\stud>ssh pi@192.168.0.116
The authenticity of host '192.168.0.116 (192.168.0.116)' can't be established.
ECDSA key fingerprint is SHA256:
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.116' (ECDSA) to the list of known hosts.
pi@192.168.0.116's password:
Linux raspberrypi 5.10.17-v7l+ #1414 SMP Fri Apr 30 13:20:47 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep  4 22:11:35 2021

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@raspberrypi:~$ whoami
pi
pi@raspberrypi:~$ pwd
/home/pi
pi@raspberrypi:~$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether dc:a6:32:46:6d:7a brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.116/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
        valid_lft 5506sec preferred_lft 4006sec
    inet6 fe80::dab0:080b:e411:7c94/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether dc:a6:32:46:6d:7c brd ff:ff:ff:ff:ff:ff
pi@raspberrypi:~$ logout
Connection to 192.168.0.116 closed.

C:\Users\stud>

```

Ekran 4: Sesja SSH zrealizowana klientem systemu operacyjnego Windows.

Polecenie, które należy wydać, aby ustanowić sesję SSH to dla zestawu o numerze 192.168.0.116 to:

```
ssh pi@192.168.0.116
```

Protokół SSH pozwala nie tylko na pracę zdalną w sesji terminalowej tekstowej, ale bezpieczne przesyłanie plików pomiędzy systemami operacyjnymi. Protokół SSH wykorzystamy również do zdalnych prac programistycznych na płycie Raspberry Pi ze zintegrowanego środowiska programistycznego Visual Studio Code uruchomionego na komputerze klasy PC.

Zadanie 2: Uruchomienie programu testującego zestaw w ramach sesji protokołu SSH

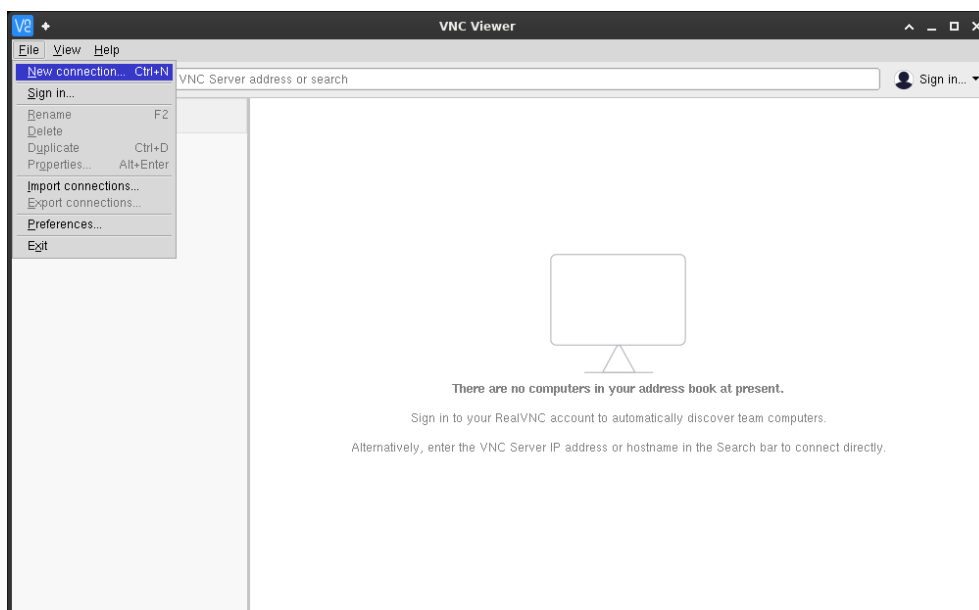
Ustanów sesję protokołu SSH z komputera stacjonarnego do zestawu. Uruchom program testujący zestaw laboratoryjny Raspberry Pi i zweryfikuj poprawność jego funkcjonowania.

3.3 Terminal VNC

Alternatywą dla pracy w ramach sesji terminalowej protokołu SSH jest możliwość pracy w ramach zdalnej sesji ze środowiskiem graficznym Raspberry Pi. Służy do tego protokół VNC. Należy wykorzystać oprogramowanie klienckie, które można pobrać ze strony [Download VNC Viewer \(link\)](#). Informacje licencyjne i przydatne w wypadku problemów znajdują się na stronach [VNC Connect and Raspberry Pi \(link\)](#) oraz [Download VNC Server for Raspberry Pi \(link\)](#).

Terminal VNC (VNC Viewer) pobieramy w odpowiedniej wersji i instalujemy stosownie do systemu operacyjnego, na którym chcemy go używać. Na komputerach w laboratorium powinien być już zainstalowany.

Po uruchomieniu tworzymy skrót do nowego połączenia, co ułatwi nam pracę. Wybieramy *Menu: File > New Connection* tak, jak zaprezentowane na ekranie 5.



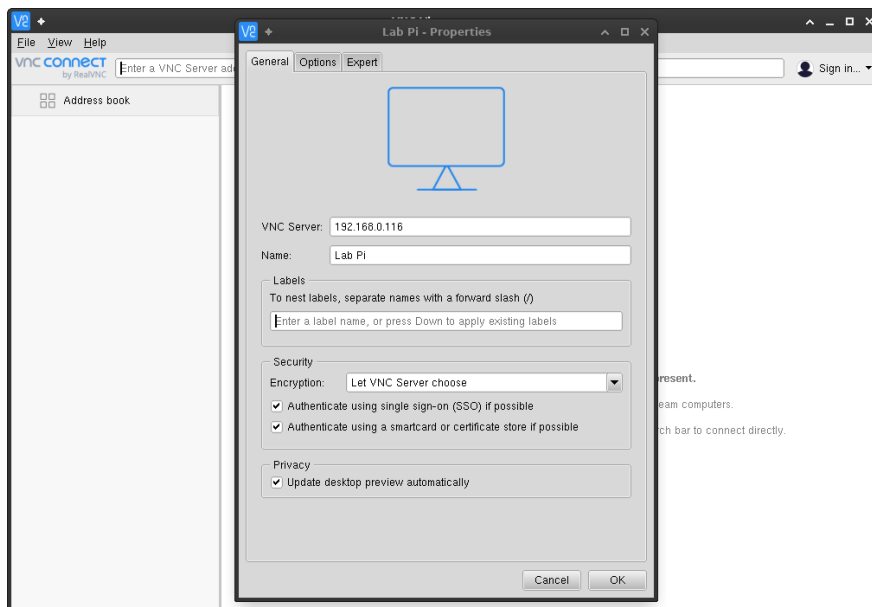
Ekran 5: Okno programu VNC Viewer z wybraną opcją utworzenia nowego połączenia.

W oknie dialogowym, przedstawionym na ekranie 6, uzupełniamy pola: *VNC Server* poprzez wpisanie numeru IP zestawu Raspberry Pi, oraz *Name*, które pozwoli nam łatwo identyfikować to połączenie, szczególnie, jeśli łączymy się z różnymi zestawami Raspberry Pi.

Jeśli z jakiegoś powodu wyświetlacz OLED nie pokazuje aktualnego adresu IP, to najprostszą metodą odczytania adresu IP zestawu Raspberry Pi jest wydanie w jego terminalu polecenia

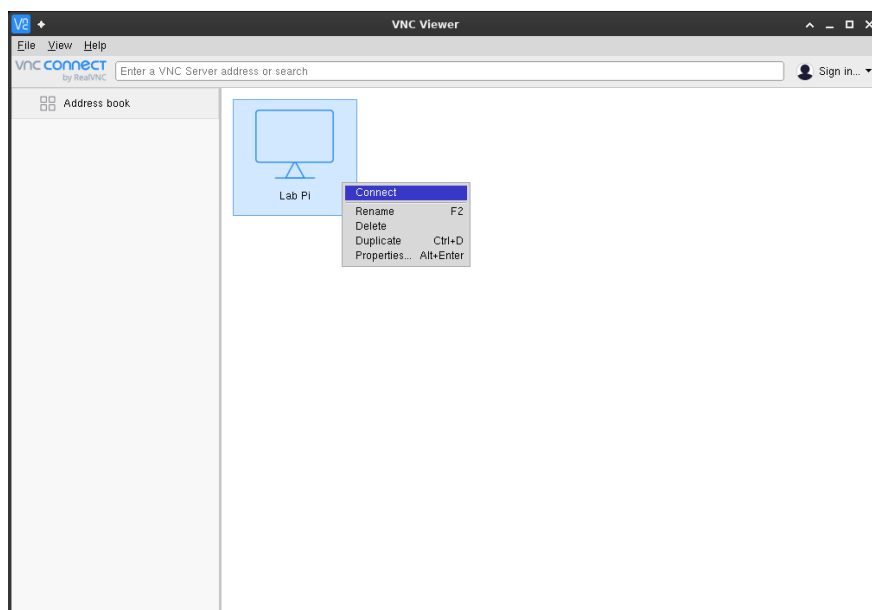
```
hostname -I
```

Proszę zwrócić uwagę, że opcja „I” musi być pisana wielką literą.



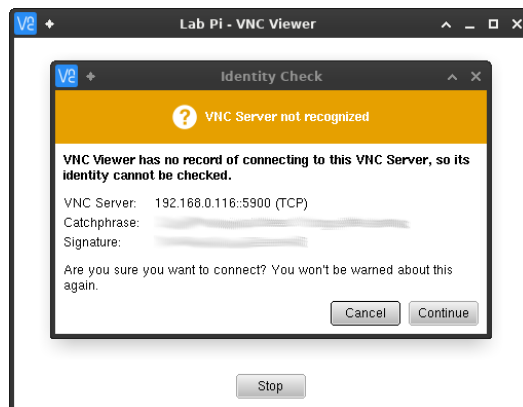
Ekran 6: Okno dialogowe programu VNC Viewer do konfiguracji połączenia VNC.

Gdy zatwierdzimy informacje klawiszem *OK*, na pulpicie programu utworzona zostanie ikona połączenia. Klikając na nią możemy rozpocząć połączenie, tak jak pokazano na ekranie 7.



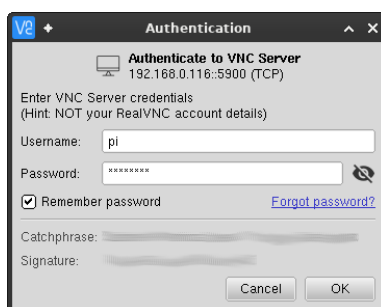
Ekran 7: Rozpoczynanie połączenia w programie VNC Viewer.

Jeśli połączenie ustanawiane jest pierwszy raz, jesteśmy proszeni o potwierdzenie, na podstawie przedstawionych informacji, że łączymy się z prawidłowym zestawem Raspberry Pi, jak to przedstawiono na ekranie 8.



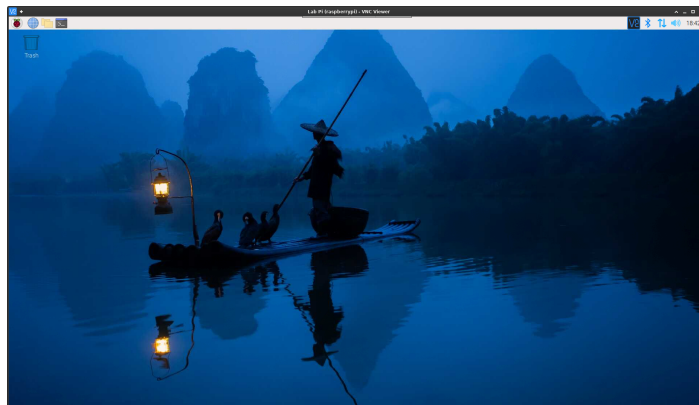
Ekran 8: Weryfikacja połączenia ustanawianego pierwszy raz w programie VNC Viewer.

Zanim połączenie zostanie ustanowione, musimy uwierzytelnić się podając nazwę użytkownika i hasło z zestawu Raspberry Pi, jak na ekranie 9.



Ekran 9: Uwierzytelnienie połączenia VNC w programie VNC Viewer.

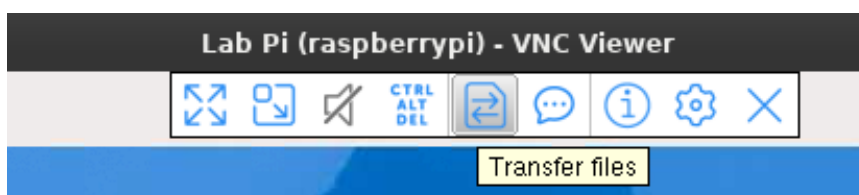
Jeśli wszystko przebiegnie prawidłowo, ustanowiona zostanie sesja VNC, jak to ilustruje ekran 10.



Ekran 10: Ustanowiona sesja VNC w programie VNC Viewer.

Można teraz rozpocząć pracę. Monitora, klawiatura i myszka podłączone bezpośrednio do zestawu Raspberry Pi są zbędne.

U góry, na środku ekranu programu VNC Viewer znajduje się rozwijalne menu. W rozwiniętej formie przedstawia je ekran 11. Sa tam dodatkowe, użyteczne funkcjonalności programu w postaci ikonek. Informacje o funkcji wybranej ikonki uzyskamy zatrzymując nad wybraną ikonką myszkę.



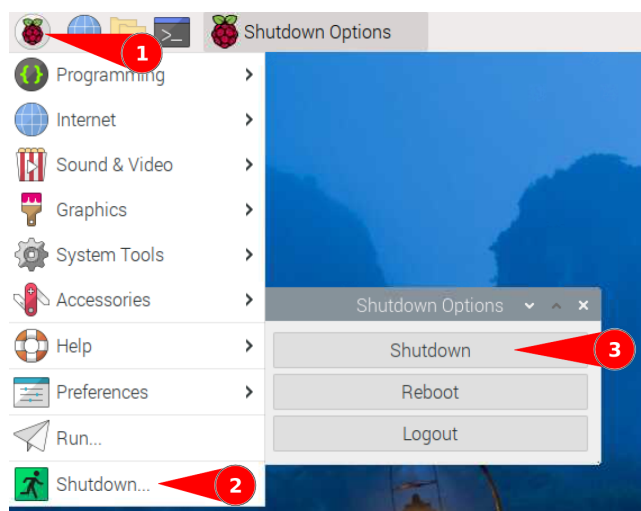
Ekran 11: Rozwijalne menu w programie VNC Viewer.

Zadanie 3: Uruchomienie programu testującego zestaw w ramach sesji VNC Viewer

Ustanów sesję VNC z komputera stacjonarnego do zestawu. Uruchom program testujący zestaw laboratoryjny Raspberry Pi i zweryfikuj poprawność jego funkcjonowania.

3.4 Wyłączanie systemu Raspberry Pi OS

Wyłączenie systemu Raspberry Pi OS możemy wykonać na dwa sposoby. Pierwszy to wyłączenie w środowisku graficznym. Wybieramy *Menu:Shutdown* i następnie w oknie dialogowym *Shutdown options* wybieramy polecenie *Shutdown*, tak jak pokazano na ekranie 12. W tym i kolejnych laboratoriach, przy opisywaniu nawigacji w środowisku graficznym Raspberry Pi OS, przez *Menu* będziemy rozumieć ikonkę z malinką w lewym górnym rogu środowiska graficznego, po kliknięciu której rozwija się lista menu niższych poziomów i poleceń.



Ekran 12: Wyłączenie Raspberry Pi OS w środowisku graficznym.

Jeśli chcemy jedynie uruchomić system ponownie, to w oknie *Shutdown options* wybieramy polecenie *Reboot*

Aby wyłączyć Raspberry Pi OS w środowisku tekstowym (konsola, emulator terminala, sesja protokołu SSH) wydajemy polecenie `sudo poweroff` (polecenie `poweroff` pisane jest jako jedno słowo). Aby uruchomić system ponownie, wydajemy polecenie `sudo reboot`.

Należy pamiętać, że w przypadku wyłączenia, po wydaniu polecenia zamknięcia systemu, należy odczekać około 30 sekund, aby system pozamykał wszystkie otwarte pliki i zapisał wszystko co było do zapisania na medium, z którego został uruchomiony, a następnie odłączył zasilanie wewnętrznych układów płytki Raspberry Pi.

Ponowne uruchomienie, po wydaniu polecenia zamknięcia systemu (*Shutdown* lub `poweroff`), jest możliwe poprzez odłączenie zasilania, odczekanie około 5 sekund i ponowne załączenie zasilania.

Przy manipulowaniu kablami i pendrive, proszę zachowywać zasady opisane w rozdziale 2.

Zadanie 4: Prawidłowe wyłączanie oraz ponowne uruchamianie zestawu

Zachowując zasady opisane w rozdziale 2 oraz podrozdziale 3.4 wykonać dwie procedury:

- wyłączenie systemu, a następnie jego włączenie,
- ponowne uruchomienie systemu (reboot).

4 Programowanie Raspberry Pi

4.1 Przygotowanie systemu do programowania układów peryferyjnych zestawu

Aby wykorzystać w pełni wszystkie układy peryferyjne zestawu, funkcjonalność interfejsu SPI powinna być skonfigurowana poprzez dodanie w pliku `/boot/config.txt` linii o treści `dtoverlay=spi1-3cs` tak, jak pokazano to na ekranie 13. Jeśli chcemy zobaczyć zawartość tego pliku, czynimy to komendą

```
cat /boot/config.txt
```

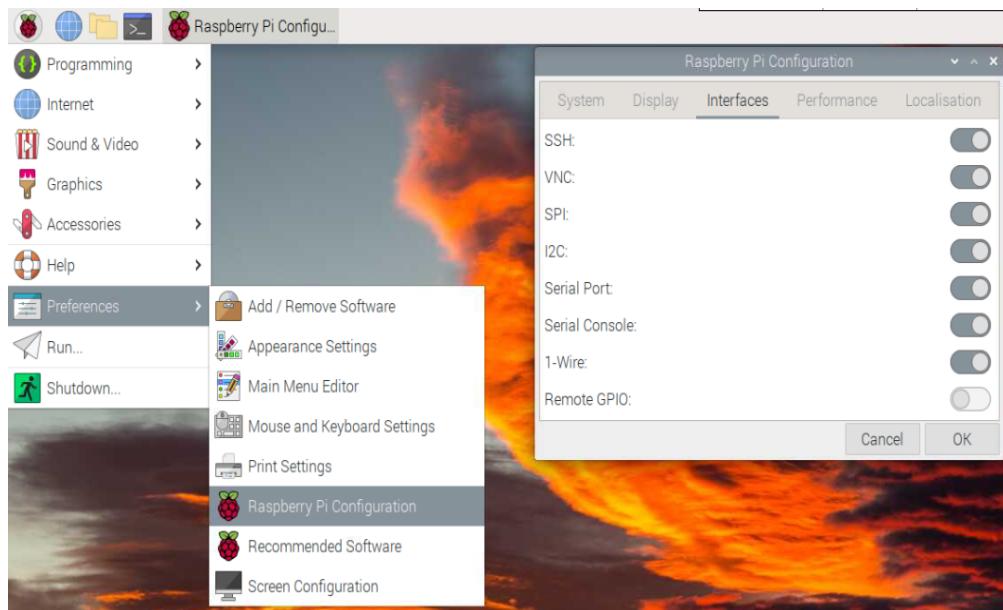
```
[pi4]
# Run as fast as firmware / board allows
arm_boost=1

[all]

dtoverlay=spi1-3cs
dtoverlay=w1-gpio
enable_uart=1
```

Ekran 13: Konfiguracja dodatkowa magistrali SPI.

Powinny być też włączone wszystkie interfejsy, które będą wykorzystywane podczas laboratorium, tak jak przedstawiono to na ekranie 14.



Ekran 14: Konfiguracja interfejsów wykorzystywanych podczas laboratorium.

4.2 Konfiguracja środowiska do programowania w języku Python

4.2.1 Uruchamianie skryptów (programów) języka Python

Podczas laboratorium, do programowania zestawów Raspberry Pi, wykorzystywać będziemy głównie język programowania Python, ze względu na łatwość programowania Raspberry Pi w tym języku. Duże znaczenie ma też dostępność sterowników urządzeń peryferyjnych zestawu laboratoryjnego przygotowanych dla programów w tym języku. W nowszych systemach Raspberry Pi OS instalowana jest jedynie wersja Python 3. W starszych może być jeszcze wersja Python 2 i może być ona domyślnym interpreterem.

Aby to zweryfikować wystarczy wydać serię komend:

```
python -V
python3 -V
```

Skrypty możemy uruchomić komendą:

```
python3 skrypt.py
```

Można też wpisać pierwszą linię skryptu w następującej formie:

```
#!/usr/bin/env python3
```

i nadać plikowi skryptu atrybut wykonywalności w systemie plikowym. Skrypt uruchamiamy bez podawania jawnie interpretera (jest zdefiniowany w skrypcie), jak na przykładzie poniżej:

```
chmod u+x skrypt.py
./skrypt.py
```

Proszę nie pomijać kropek przed znakiem „/”, tam gdzie jest wymagana. Wymagana będzie wtedy, gdy nie podamy pełnej ścieżki do skryptu, a sekwencja „./” wskazujemy bieżący katalog. Obowiązują zatem zasady uruchamiania skryptów, nie tylko języka Python, takie jak w większości dystrybucji systemu Linux.

4.2.2 Moduły języka Python do obsługi układów peryferyjnych zestawu

W systemie Raspberry Pi OS domyślnie zainstalowane są biblioteki języka Python, które pozwalają obsługiwać część urządzeń peryferyjnych. Są to układy, które sterowane są poziomami logicznymi lub współczynnikiem wypełnienia przebiegu prostokątnego z portów GPIO lub same wysyłają sygnały logiczne odczytywane przez te porty. W zestawie laboratoryjnym Raspberry Pi instalacji dodatkowych sterowników nie wymagają:

- przełączniki przyciskane zielony i czerwony,
- enkoder,
- moduł niebieskich diod sygnalizacyjnych,
- sygnalizator dźwiękowy - buzzer.

Obsłudze programowej nie podlegają i nie wymagają żadnego oprogramowania:

- stabilizator napięcia 3,3V dla wyświetlacza OLED,
- konwerter poziomów logicznych dla linijki diod LED RGB oraz sygnalizatora dźwiękowego.

Dla pozostałych urządzeń w zestawach laboratoryjnych zainstalowano dodatkowe biblioteki w postaci modułów języka Python:

- Linijka programowalnych diod LED RGB WS2812

```
sudo pip3 install board  
sudo pip3 install adafruit-circuitpython-neopixel
```
- Czujnik temperatury DS18B20

```
sudo pip3 install w1thermsensor
```
- Moduł z układem BME280 - czujnik temperatury, wilgotności i ciśnienia

```
sudo pip3 install adafruit-circuitpython-bme280
```
- Moduł czytnika kart zbliżeniowych RFID

```
sudo pip3 install mfrc522
```

- Wyświetlacz OLED

Dokumentacja i biblioteka do obsługi wyświetlacza OLED dostarczane są przez producenta modułu wyświetlacza i udostępnione na [stronie modułu OLED \(link\)](#), w dziale *Resources*, pod odnośnikiem *Code (link)* znajdują się biblioteki sterowników. Proszę wersję biblioteki z witryny producenta potraktować jako materiał źródłowy. Nie będzie ona działać prawidłowo. W zestawie laboratoryjnym wykorzystywana jest jej wersja dostosowana do połączeń zestawu laboratoryjnego i można ją znaleźć w kodzie programu testowego zestawu.

Wszystkich sterowników powyżej wymaga program testujący poprawność funkcjonowania zestawu laboratoryjnego.

Zadanie 5: Uruchamianie skryptów języka Python z linii komend

Wykorzystując dowolny edytor tekstu, przygotuj prosty skrypt języka Python o nazwie `myscript.py` podejmujący interakcję z użytkownikiem poprzez konsolę. Skonfiguruj pierwszą linię wskazującą interpreter `python3` i nadaj plikowi atrybuty wykonywalności. Uruchom skrypt poleceniem

```
./myscript.py
```

5 Wiedzieć więcej o Raspberry Pi

- [Strona Raspberry Pi Foundation \(link\)](#) jest podstawowym źródłem informacji.
- Wiele informacji, podanych w systematyczny i przystępny sposób, można uzyskać na stronie [Raspberry Pi Documentation \(link\)](#).
- Więcej bezpłatnych materiałów dostępnych jest na stronie [The MagPi Magazine \(link\)](#), gdzie dostępne są, między innymi, działy z wydaniem magazynu, z książkami, instrukcje (tutoriale) i artykuły tematyczne. Jest tu naprawdę ogromny zasób wiedzy na temat Raspberry Pi i jego wykorzystania.