



Lokalne i zdalne programowanie zestawów Raspberry Pi w języku Python

Opracował opiekun przedmiotu dr inż. Krzysztof Chudzik

Wydział Informatyki i Telekomunikacji
Politechnika Wrocławska

Wrocław, 2025.09.30 18:50:03

Spis treści

1 Programowanie z wykorzystaniem narzędzi lokalnych Raspberry Pi	1
1.1 Thonny Python IDE	2
1.2 Geany IDE	2
1.3 Visual Studio Code IDE	3
2 Programowanie zdalne Raspberry Pi	4
2.1 Przygotowanie komputera i VS Code do programowania zdalnego w języku Python	4
2.2 Ustanowienie sesji zdalnej	4
2.3 Więcej o programowaniu zdalnym w VS Code	9

Lista zadań

1 Wykorzystanie Thonny Python IDE do programowania w języku Python	2
2 Wykorzystanie Geany IDE do programowania w języku Python	2
3 Wykorzystanie VS Code do programowania w języku Python	3
4 Uruchamianie programu w sesji zdalnej VS Code	9
5 Utworzenie programu, uruchomienie i debugowanie w sesji zdalnej VS Code	9

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.

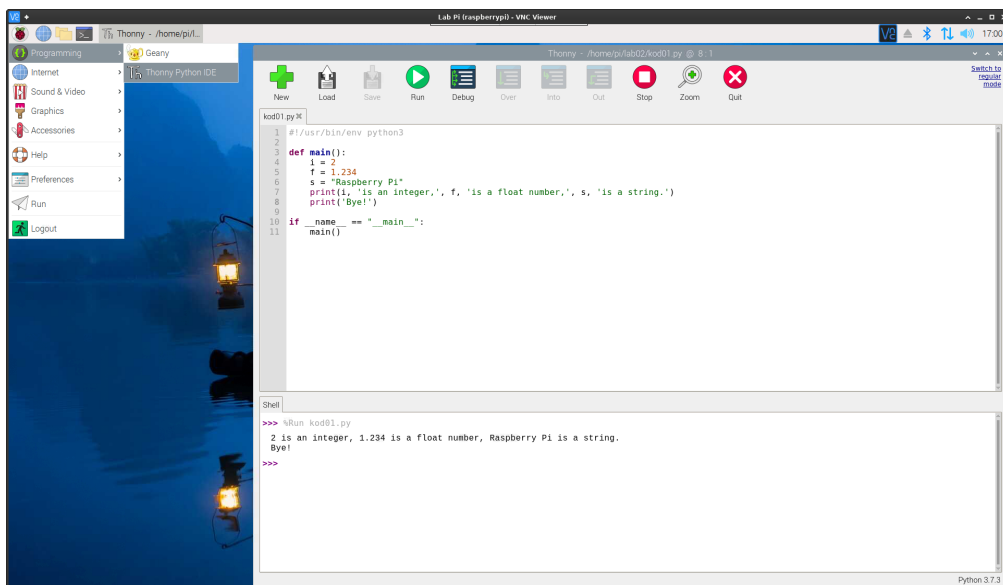
1 Programowanie z wykorzystaniem narzędzi lokalnych Raspberry Pi

Programowanie, uruchamianie i debugowanie programów może odbywać się lokalnie z wykorzystaniem środowisk tekstowych lub graficznych. Nic nie stoi na przeszkodzie, aby do programowania wykorzystywać środowisko tekstowe, niemniej jednak, w czasie zajęć laboratoryjnych, ograniczymy się do środowisk programistycznych z interfejsem graficznym.

Omówimy teraz opcje dostępne lokalnie na płycie Raspberry Pi. Będzie to przegląd wybranych rozwiązań, a nie kompletne wyliczenie wszystkich dostępnych dla płytki rozwiązań.

1.1 Thonny Python IDE

Jest to środowisko pre-instalowane z systemem operacyjnym Raspberry Pi OS. Przeznaczone jest dla początkujących i należy do stosunkowo prostych. Umożliwia zarówno uruchamianie programów jak i ich debugowanie. Uruchamiamy je wybierając *Menu: Programming > Thonny Python IDE*, tak jak pokazano to na ekranie 1.



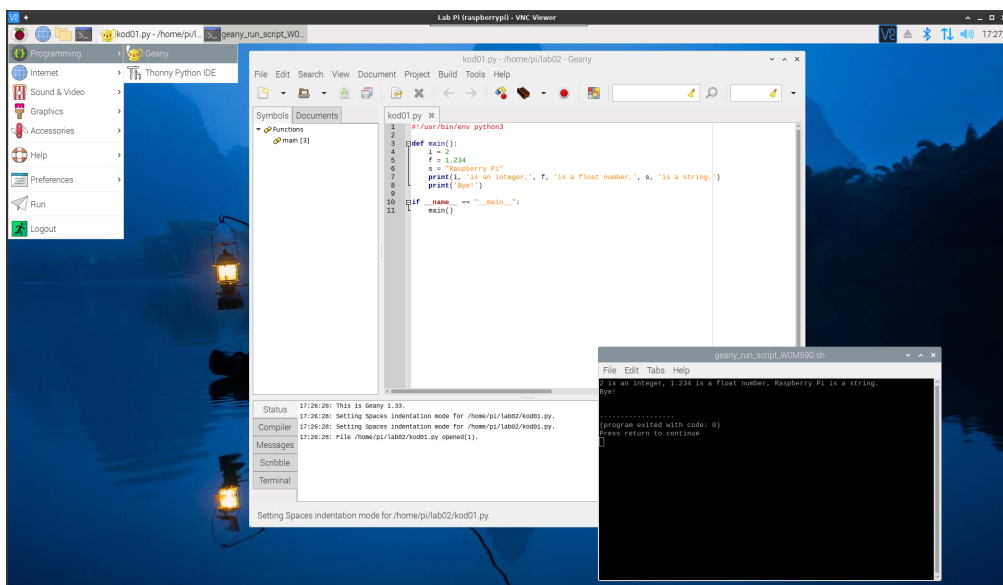
Ekran 1: Thonny Python IDE.

Zadanie 1: Wykorzystanie Thonny Python IDE do programowania w języku Python

Przygotuj, uruchom i przeprowadź sesję debugowania przygotowanego przez Ciebie prostego programu w środowisku Thonny Python IDE.

1.2 Geany IDE

Jest to środowisko pre-instalowane, przeznaczone do programowania w różnych językach, w tym języku Python. Pomimo, iż należy do środowisk lekkich, jest nieco bardziej złożone niż poprzednio prezentowane. Uruchamiamy je wybierając *Menu: Programming > Geany*, tak jak pokazano to na ekranie 2.



Ekran 2: Geany IDE.

Zadanie 2: Wykorzystanie Geany IDE do programowania w języku Python

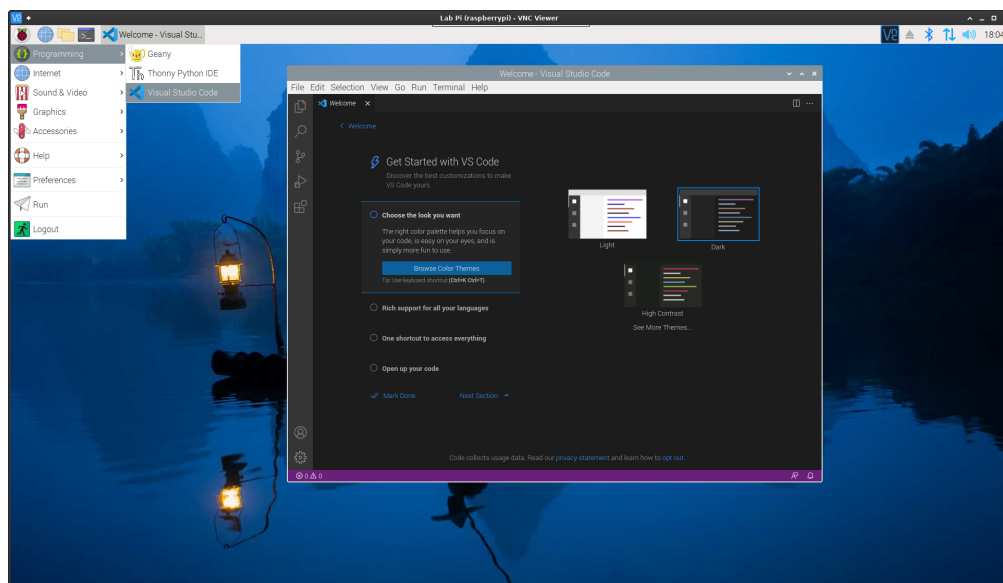
Przygotuj i uruchom prosty program w środowisku Geany IDE.

1.3 Visual Studio Code IDE

Instalacja środowiska w Raspberry Pi OS odbywa się sekwencją komend:

- `sudo apt update`
- `sudo apt install code`

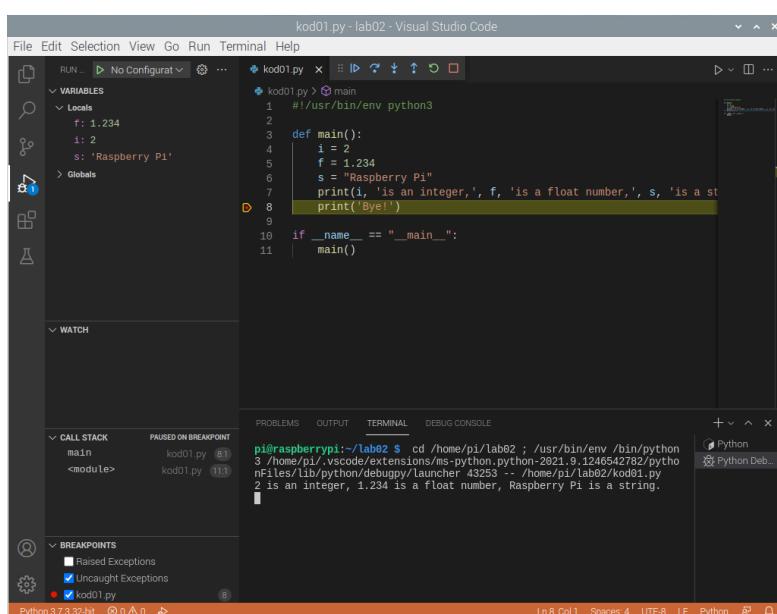
VS Code został zainstalowany na zestawach laboratoryjnych. Po zainstalowaniu program dostępny jest w Raspberry Pi OS w *Menu: Programming > Visual Studio Code*, jak ilustruje to ekran 3.



Ekran 3: Visual Studio Code w Raspberry Pi OS.

Przed rozpoczęciem programowania w Pythonie, należy zainstalować rozszerzenie (ang. extension) *Python* dostarczane przez firmę Microsoft (extension Id: ms-python.python). Rozszerzenie Python zostało zainstalowane na zestawach laboratoryjnych. Dokumentacja rozszerzenia dostępna jest, na przykład, na stronie [VS Marketplace](#) (link).

Dalsza praca odbywa się tak samo jak na komputerze czy laptopie, co ilustruje ekran 4 z prezentacją sesji debugowania bardzo prostego programu.



Ekran 4: Sesja debugowania w Visual Studio Code w Raspberry Pi OS.

Zadanie 3: Wykorzystanie VS Code do programowania w języku Python

Przygotuj, uruchom i przeprowadź sesję debugowania przygotowanego przez Ciebie prostego programu w środowisku VS Code.

2 Programowanie zdalne Raspberry Pi

Programowanie lokalne, opisane w poprzednim rozdziale, wykorzystuje lokalne zasoby Raspberry Pi do uruchamiania środowisk programistycznych i uruchamiania programów. Raspberry Pi jest urządzeniem w swojej klasie stosunkowo wydajnym, jednak uruchamianie bardziej złożonych środowisk programistycznych może powodować duże obciążenie układu. Responsywność środowiska programistycznego niekoniecznie będzie taka, do jakiej przyzwyczajony jest użytkownik komputera stacjonarnego, czy laptopa. Uruchamianie oprogramowania nie można przenieść poza płytkę Raspberry Pi ze względu na specyfikę sprzętu i podłączone urządzenia peryferyjne. Można jednak doprowadzić do sytuacji takiej, że programista będzie uruchamiał programy na płytce Raspberry Pi, ale środowisko programistyczne uruchomione będzie na komputerze i łączyć się będzie z płytką Raspberry Pi w celu zdalnego uruchomienia i debugowania programów. Zaprezentowane zostanie rozwiązanie oparte o środowisko programistyczne VS Code i komunikację z wykorzystaniem protokołu SSH.

2.1 Przygotowanie komputera i VS Code do programowania zdalnego w języku Python

Instalacja interpretera języka Python:

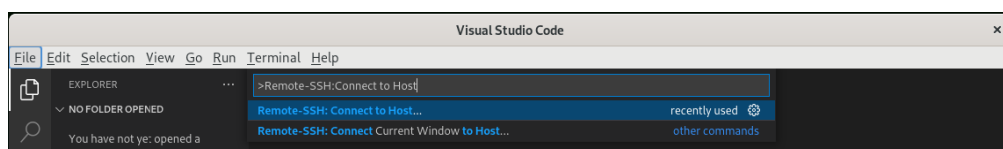
- Zainstaluj na komputerze język Python 3, jeśli nie jest zainstalowany. Stosowne instrukcje znajdują się, na przykład, na stronie [Python Setup and Usage \(link\)](#).
- Zweryfikuj poprawność działania interpretera języka Python komendą `python3 --version`, a następnie napisz i uruchom krótki program typu „Hello World!”, aby sprawdzić, czy interpreter działa poprawnie.

Instalacja i konfiguracja VS Code do pracy zdalnej:

- Zainstaluj VS Code zgodnie z instrukcjami na stronie [Setting up Visual Studio Code \(link\)](#).
- Zainstaluj rozszerzenie (ang. extension) *Python* dostarczane przez firmę Microsoft (extension Id: ms-python.python). Dokumentacja rozszerzenia dostępna jest, na przykład, na stronie VS Marketplace poświęconej [rozszerzeniu Python \(link\)](#).
- Zainstaluj rozszerzenie *Remote - SSH* dostarczane przez firmę Microsoft (extension Id: ms-vscode-remote.remote-ssh). Dokumentacja rozszerzenia dostępna jest, na przykład, na stronie VS Marketplace poświęconej [rozszerzeniu Remote - SSH \(link\)](#).

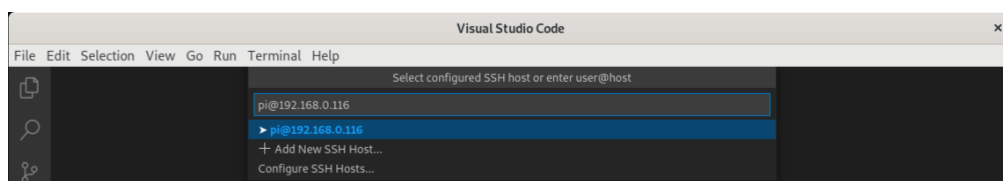
2.2 Ustanowienie sesji zdalnej

Po uruchomieniu VS Code naciskamy Ctrl+Shift+P lub klawisz F1 i w palecie komend wpisujemy lub wybieramy komendę **Remote SSH: Connect to Host** jak na ekranie 5.



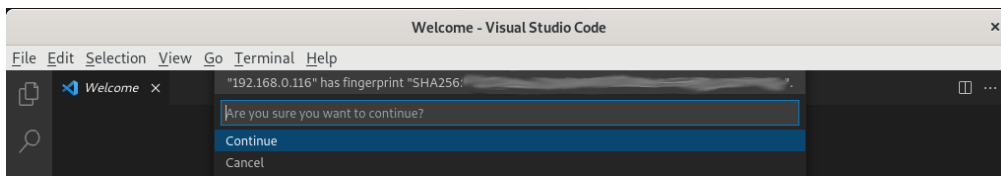
Ekran 5: Wybór połączenia sesji zdalnej z wykorzystaniem protokołu SSH.

Następnie podajemy zdalnego użytkownika i numer IP (lub nazwę hosta) zdalnego urządzenia jak na ekranie 6. Numer IP urządzenia Raspberry Pi uzyskamy, na przykład, wpisując w emulatorze terminala komendę `hostname -I`.



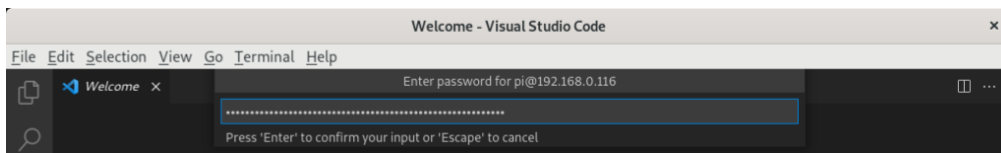
Ekran 6: Podanie zdalnego użytkownika i adresu.

Jeśli połączenie z wybranym urządzeniem jest dokonywane po raz pierwszy z danego komputera, należy zaakceptować ustanawianie połączeń do tego urządzenia wybierając polecenie **Continue** jak na ekranie 7.



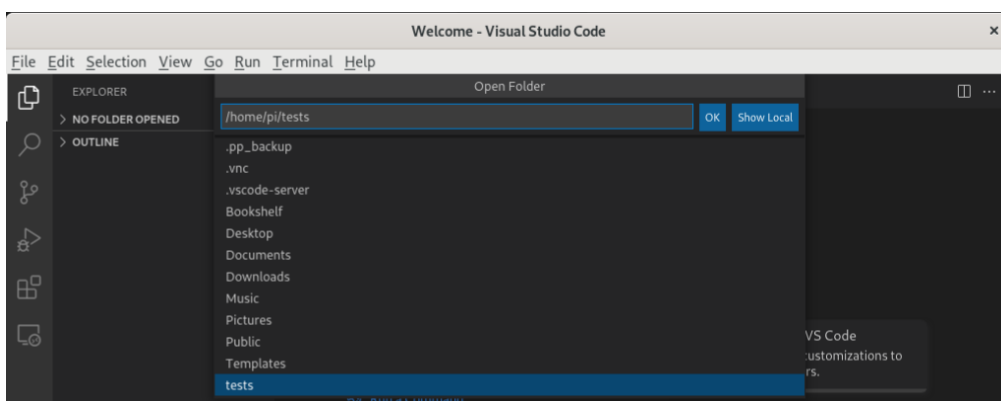
Ekran 7: Akceptacja urządzenia zdalnego.

W następnym kroku podajemy hasło użytkownika zdalnego (ekran 8).



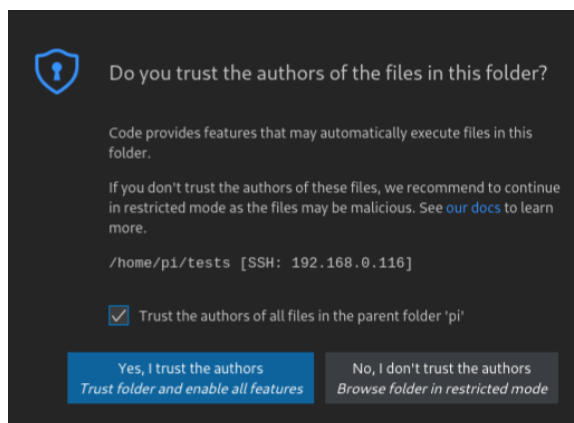
Ekran 8: Wprowadzenie hasła.

Po zalogowaniu, wybieramy katalog roboczy (ekran 9). Do dalszej demonstracji wybierzemy katalog zawierający programy (skrypty) do testowania urządzeń peryferyjnych zestawu laboratoryjnego.



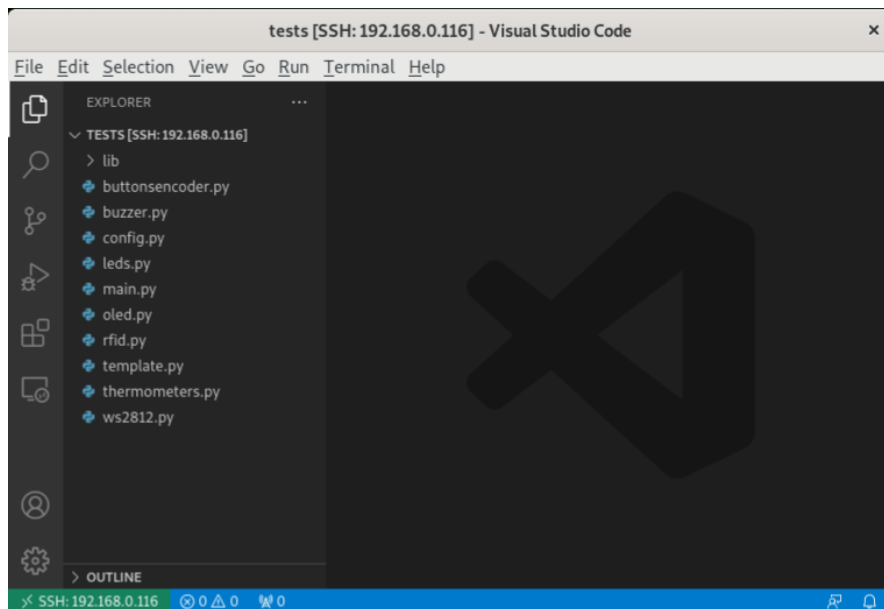
Ekran 9: Wybór katalogu (folderu) roboczego na urządzeniu zdalnym.

Tak, jak w pracy lokalnej, VS Code prosi o akceptację istniejących artefaktów (ekran 10).



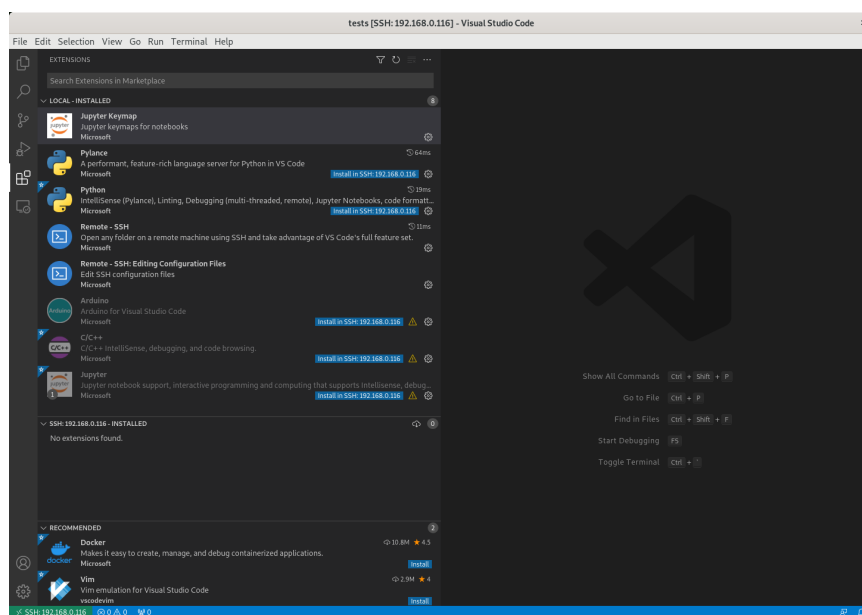
Ekran 10: Akceptacja istniejących artefaktów.

Jeżeli wszystkie operacje przebiegły prawidłowo, rezultat powinien być podobny do tego na ekranie 11. Proszę zwrócić uwagę, że w oknie *Explorer* (lewy panel) wyraźnie jest wskazane w jego górnej części, że pliki są prezentowane w ramach sesji SSH.



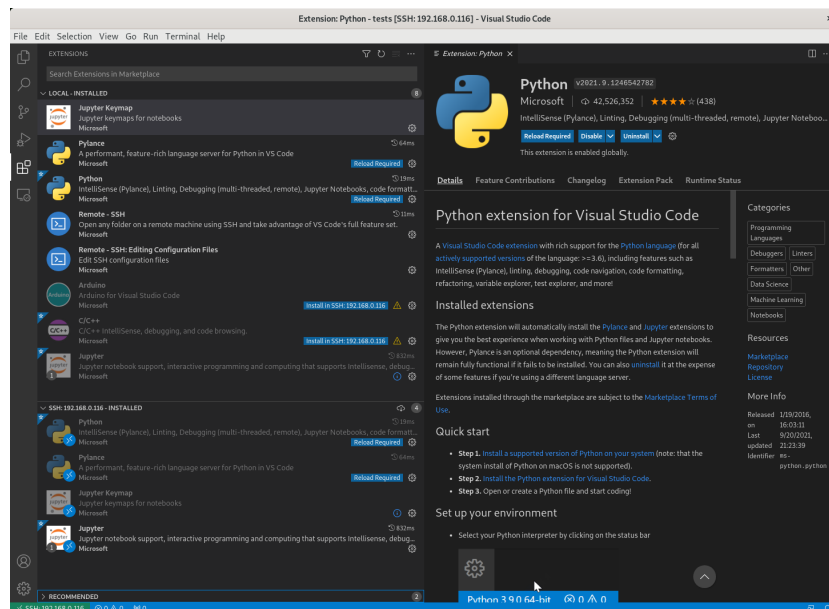
Ekran 11: Sesja zdalna ustanowiona.

W opisaną dotychczas procedurze, w czasie ustanawiania sesji, po stronie urządzenia, do którego przyłączamy się uruchomione zostało specjalne oprogramowanie serwerowe, dla którego środowisko programistyczne VS Code jest klientem. Jeśli włączymy widok pokazujący zainstalowane rozszerzenia tak, jak na ekranie 12, widzimy, jeśli robimy to pierwszy raz i nikt przed nami nie instalował rozszerzeń zdalnie, że na urządzeniu zdalnym nie są dostępne żadne rozszerzenia, natomiast rozszerzenia w obszarze lokalnym mają niebieskie przyciski *Install in SSH*.



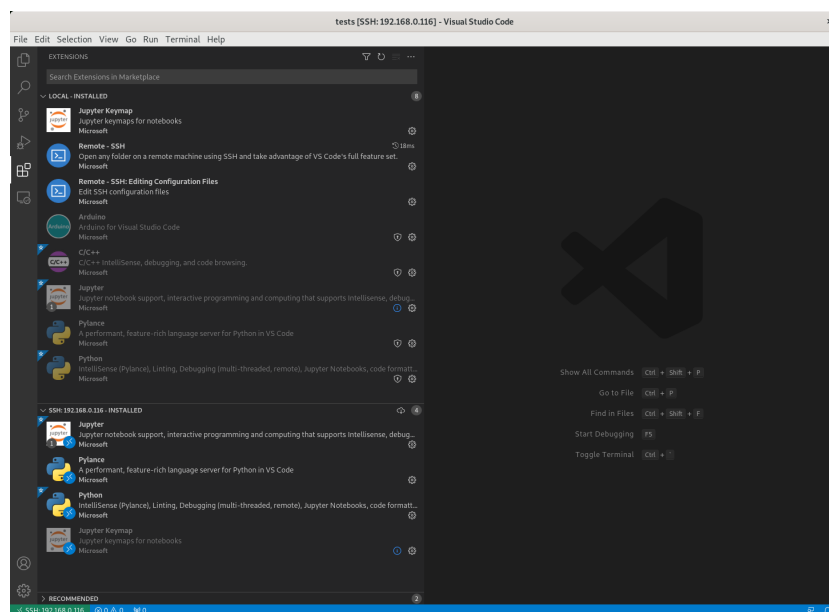
Ekran 12: Rozszerzenia zainstalowane wyłącznie lokalnie.

Jeśli nikt tego nie zrobił wcześniej, wybieramy *Install in SSH* dla rozszerzenia *Python*. Instalacja zajmuje chwilę, ale po jej zakończeniu widzimy, że doinstalowane zostały na urządzeniu zdalnym rozszerzenia do programowania w języku Python. Pokazuje to ekran 13.



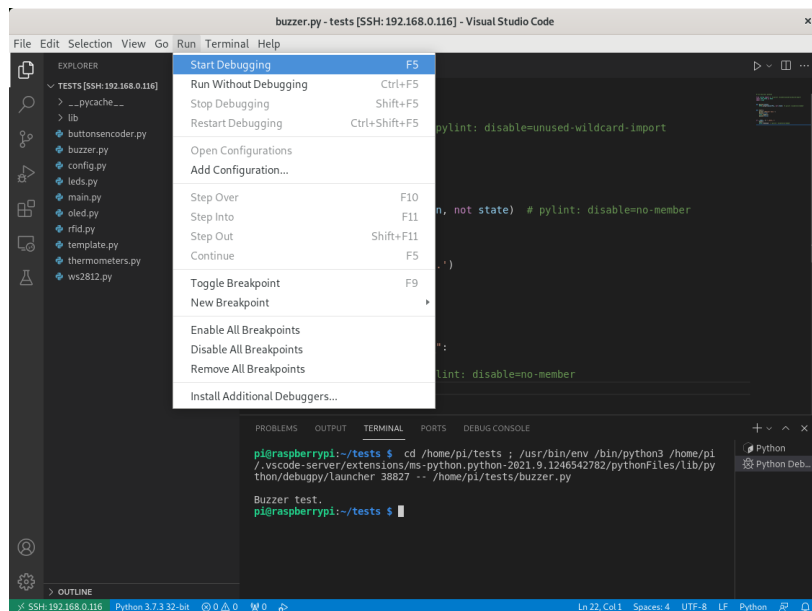
Ekran 13: Rozszerzenia zainstalowane zdalnie - Wymagane przeładowanie środowiska programistycznego.

Naciskamy niebieski przycisk *Reload Required*, aby dopełnić procedurę instalacyjną. W jej rezultacie, rozszerzenia zostają zainstalowane na urządzeniu zdalnym, jak to ilustruje ekran 14.

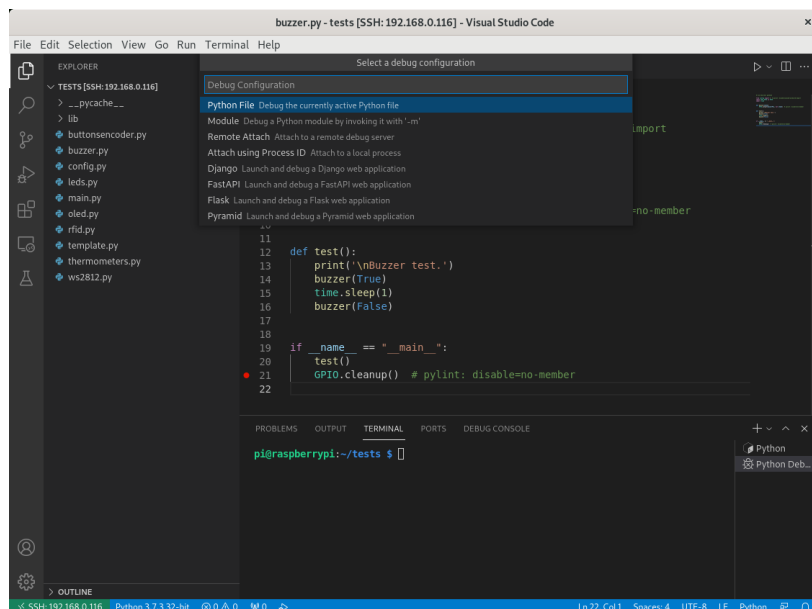


Ekran 14: Rozszerzenia zainstalowane zdalnie - Środowisko programistyczne przeładowane.

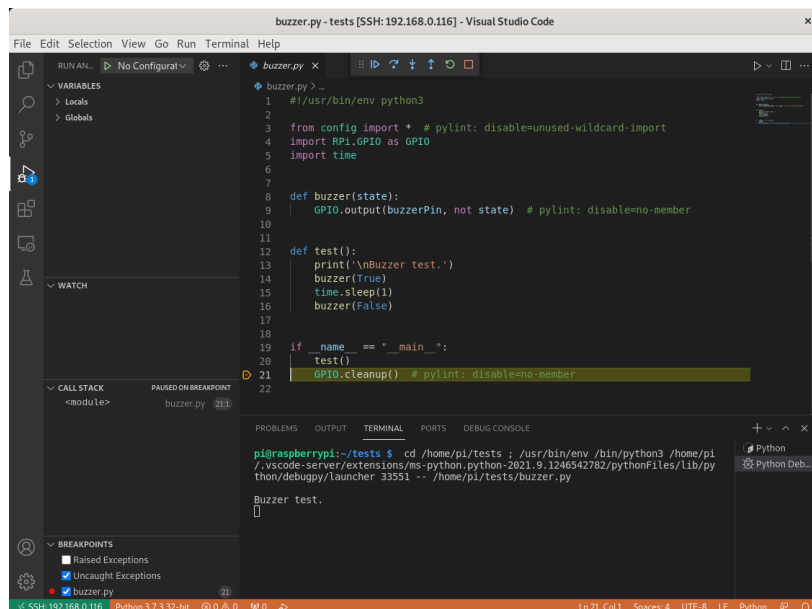
Możemy teraz zdalnie uruchamiać i debugować programy, jak pokazano to na ekranach 15, 16 i 17.



Ekran 15: Uruchomienie zdalnej sesji debugowania.



Ekran 16: Wybór opcji debugowania.



Ekran 17: Sesja zdalna debugowania.

2.3 Więcej o programowaniu zdalnym w VS Code

- [VS Code Remote Development \(link\)](#)
- [Remote development over SSH \(link\)](#)

Zadanie 4: Uruchamianie programu w sesji zdalnej VS Code

Uruchom program testujący zestaw laboratoryjny Raspberry Pi w sesji zdalnej z VS Code.

Zadanie 5: Utworzenie programu, uruchomienie i debugowanie w sesji zdalnej VS Code

Przygotuj własny prosty program uruchomiony na zestawie laboratoryjnym Raspberry Pi. Uruchom go i przeprowadź debugowanie w sesji zdalnej z VS Code. Program nie musi wykorzystywać żadnych urządzeń peryferyjnych. Będzie to przedmiotem kolejnych zajęć laboratoryjnych.