github.com/sevenreek

rev. 19-02-2020

Mikrodokumentacja techniczna i instrukcja instalacji oprogramowania

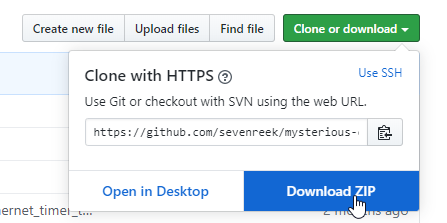
# Struktura projektu

Podfoldery dzielą projekt na 3 części. *DashboardServer* zawiera skrypty w języku Python służące do obsługi serwera administracyjnego. *ERAdmin* zawiera wszystkie elementy wystroju i funkcjonowania strony internetowej uruchamianej przez skrypty z *DashboardServer.* Na folder *TimerServer* składają się wszystkie elementy niezbędne do prawidłowego funkcjonowania serwerów odmierzających czas i zarządzających pokojami (Raspberry Pi). Oprócz skryptów Python utrzymujących serwer na każdym z urządzeń znajdują się tu również pliki konfiguracyjne.

Cały kod źródłowy dostępny jest w jednym repozytorium na platformie github.com:

* <https://github.com/sevenreek/mysterious-complete>

Może on zostać ściągnięty poprzez przeglądarkę w archiwum:



Lub na komputerze z zainstalowanym gitem (<https://git-scm.com/>) poprzez polecenie:   
*git clone https://github.com/sevenreek/mysterious-complete.*

# Konfiguracja i uruchomienie maszyny administrującej

## Niezbędny sprzęt

Oprogramowanie powinno działać na dowolnej maszynie wyposażonej w kartę sieciową z Ethernetem i będącą w stanie uruchomić Pythona 3 oraz współczesną przeglądarkę internetową, zatem każdy przeciętny komputer z systemem Windows 8 lub wyższym lub nadal wspieraną wersję Linuxa.

## Niezbędne oprogramowanie

Do uruchomienia skryptów znajdujących się w folderze DashboardServer a w konsekwencji wystartowania strony zarządzającej pokojami niezbędne jest następujące oprogramowanie:

1. Python 3: <https://www.python.org/downloads/>

Oprogramowanie było testowane na wersjach 3.7.0 dla Windows oraz 3.6.8 dla Linux Ubuntu. Nowsze wersje Pythona 3 również powinny działać bezproblemowo. Przy instalacji warto zaznaczyć opcję Add Python 3.X to PATH aby móc uruchamiać Pythona z poziomu wiersza polecenia w każdym miejscu. Koniecznie upewnić się, że opcja dotycząca instalacji pip jest zaznaczona. W Linuxie instalacja poprzez polecenie: *sudo apt-get install python3.*

1. Python Bottle: <https://bottlepy.org/docs/dev/>

Biblioteka była testowana w wersji 0.12. Nowsze wersje również powinny działać bezproblemowo. Instalacja poprzez pip: otworzyć wiersz polecenia i zainstalować Bottle na komputerze z Internetem za pomocą polecenia *pip install bottle;* dla Linuxa *pip3 install bottle.* Biblioteka może zostać także ściągnięta ze strony i umieszczona w folderze DashboardServer.

1. Python requests: <https://requests.readthedocs.io/en/master/user/install/#install>

Biblioteka była testowana w wersji 2.7.0. Nowsze wersje również powinny działać bezproblemowo. Instalacja poprzez pip: otworzyć wiersz polecenia i zainstalować requests na komputerze z Internetem za pomocą polecenia *pip install requests;* dla Linuxa: *pip3 install requests.* Biblioteka może zostać także ściągnięta ze strony i umieszczona w folderze DashboardServer.

## Opcjonalne oprogramowanie

Wyżej wymienione oprogramowanie jest niezbędne do prawidłowego funkcjonowania jednak dodatkowo warte instalacji są:

1. VNC Viewer: <https://www.realvnc.com/en/connect/download/viewer/>

Zdalny pulpit VNC pozwala na wyświetlanie aktualnego ekranu z Raspberry Pi poprzez Ethernet i ewentualną naprawę problemów czy monitorowanie ich zachowania. Każda poprawnie skonfigurowana Raspberry Pi będzie obsługiwać tę funkcję.

1. Git: <https://git-scm.com/>

System kontroli wersji dla deweloperów. Znacząco ułatwi synchronizację w przypadku zdalnej pomocy.

1. Klient SSH np. PuTTY: <https://www.putty.org/>

Funkcjonalnie zbliżona do VNC Viewer jednak ogranicza użytkownika do wiersza polecenia pomijając graficzny aspekt interfejsu. Raczej narzędzie do debugowania lub pierwszej konfiguracji nowych Raspberry.

1. Przeglądarka oparta na chromium: <https://www.google.com/intl/pl_pl/chrome/>

Layout strony nie był testowany na innych przeglądarkach niż Google Chrome. Żadna aktualna przeglądarka nie powinna mieć problemu z wyświetleniem elementów strony jednak Chrome lub Firefox mają najwięcej narzędzi deweloperskich.

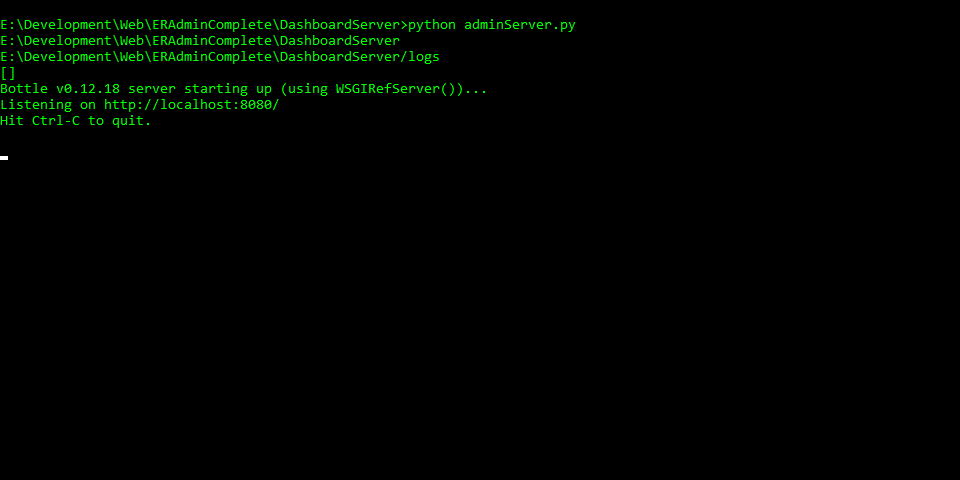
1. Ubuntu Subsystems for Windows: [https://www.microsoft.com/pl-pl/p/ubuntu-1804-lts/9n9tngvndl3q](https://www.microsoft.com/pl-pl/p/ubuntu-1804-lts/9n9tngvndl3q?rtc=1&activetab=pivot:overviewtab)

Minimalny wiersz polecenia emulujący Linuxa w wersji Ubuntu18 z poziomu Windowsa. Funkcjonalność mocno opcjonalna i absolutnie niewymagana. W testach ten wiersz polecenia okazał się troszkę mniej problematyczny przy zamykaniu i konfiguracji aplikacji, jednak może być mniej intuicyjny dla użytkowników niedoświadczonych w obsłudze interfejsów tekstowych.

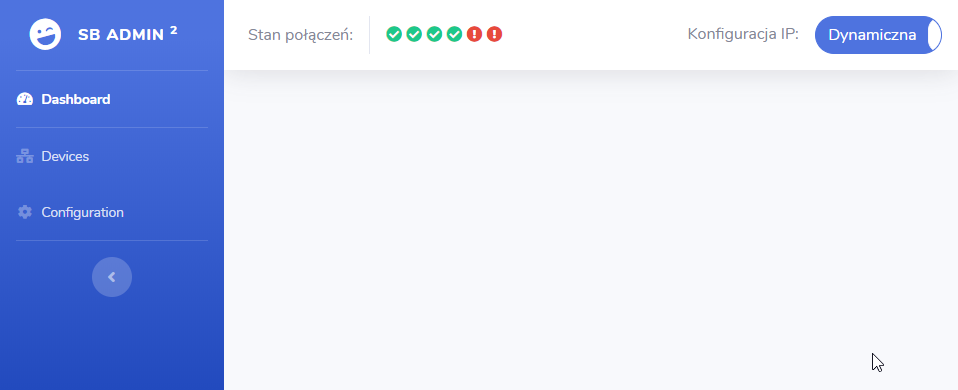
## Uruchomienie aplikacji

Uruchomienie serwera odbywa się poprzez egzekucję skryptu *adminServer.py* w folderze *DashboardServer*. Odpowiednia konfiguracja Pythona powinna pozwolić na zrobienie tego poprzez podwójne kliknięcie na plik w systemie Windows. W przeciwnym wypadku należy to zrobić z pozycji wiersza polecenia: trzymając klawisz prawy shift, nacisnąć prawym przyciskiem myszy w folderze *DashboardServer* i z menu kontekstowego wybrać opcję *Otwórz wiersz polecenia*, *Otwórz okno PowerShell,* lub podobną. W otwartym oknie należy wtedy wpisać polecenie *python adminServer.py.*

Rezultat powinien wyglądać następująco:



Wbrew pojawiającej się informacji o kombinacji Ctrl+C, na niektórych terminalach w tym domyślnych Windowsowych kombinacja ta nie jest w stanie wyłączyć aplikacji i należy to zrobić poprzez zamknięcie okna, lub kombinację Alt+F4. Jeśli nie pojawiły się żadne błędy strona administracyjna powinna być dostępna na urządzeniu pod adresem <http://localhost:8080/index.html> oraz <http://localhost:8080/admin> i wyglądać mniej więcej tak:



Aplikacja komunikuje się z Raspberry Pi za pomocą dwóch portów:

* 8080 dla połączeń w protokole HTTP: cała komunikacja od pobrania aktualnego stanu po polecenia do konkretnych urządzeń.
* 4000 dla broadcastów UDP wysyłanych przez Raspberry Pi do wszystkich urządzeń w sieci: broadcasty te służą do identyfikacji Raspberry Pi i są odbierane przez serwer na komputerze, pozwalając na dynamiczną konfigurację adresów IP.

To rozwiązanie sprawia, że konfiguracja sieci powinna ograniczyć się do odblokowania w routerze wyżej wymienionych portów oraz odpowiedniego ustawienia zapory sieciowej Windowsa (lub jej wyłączenie dla sieci z urządzeniami). Porty te w razie potrzeby można zmienić, jednak wymaga to ingerencji w kod źródłowy w DashboardServer/adminServer.py, TimerServer/CONFIGURATION (na każdej z Raspberry Pi), ERAdmin/dashboard.js, ERAdmin/devices.js.

# Konfiguracja i uruchomienie serwera pokoju

## Niezbędny sprzęt

Program testowany był na Raspberry Pi w wersjach:

* B+ V1.2 2014
* 3B V1.2 2015
* 3B+ 2017

Każda z powyższych nie wykazała problemów w ciągłej operacji programu, jednak naturalnie nowsze wersje są wielokrotnie szybsze od poprzednich, zatem zarówno proces każdorazowego uruchomienia jak i konfiguracji jest wielokrotnie szybszy i przyjemniejszy. Wszystkie inne wersje RPi3 jak i Raspberry Pi 4 również powinny działać.

Dodatkowo niezbędny będzie adapter microSD-USB, lub potencjalnie wbudowany w laptopa slot microSD, aby móc umieścić system Raspbian na karcie SD dla każdej z Raspberry Pi. Zalecana minimalna pojemność kart microSD to 16GB (prawdopodobnie wystarczy 8GB). Zalecane jest także użycie jednej pojemności i serii kart dla wszystkich urządzeń.

## Niezbędne oprogramowanie

1. Raspbian Buster with desktop: <https://www.raspberrypi.org/downloads/raspbian/>

Obraz systemu dla Raspberry Pi.

1. balenaEtcher: <https://www.balena.io/etcher/>

Program do wgrywania obrazu systemu na karty SD.

1. SD Card Formatter: <https://www.instalki.pl/programy/download/Windows/narzedzia_dyskowe/SDFormatter.html>

Program do skutecznego formatowania kart SD.

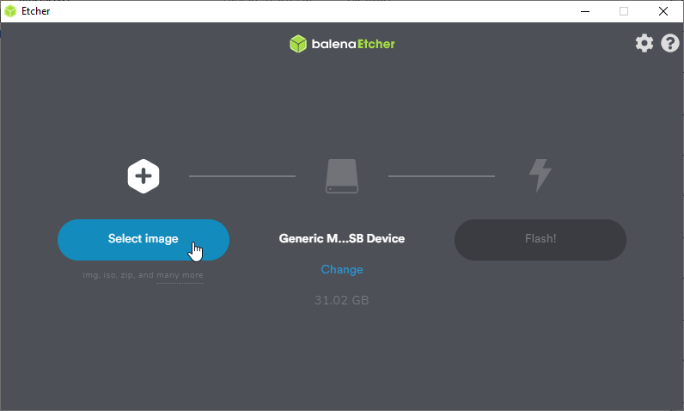
## Instalacja systemu

### Formatowanie karty

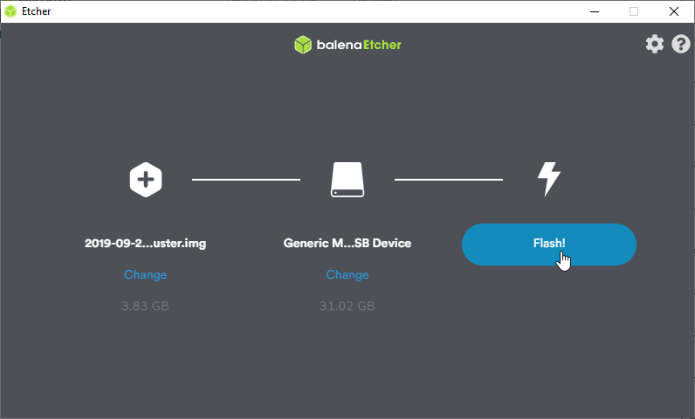
Kartę należy umieścić w odpowiednim adapterze, podłączyć do komputera i uruchomić SD Card Formatter.

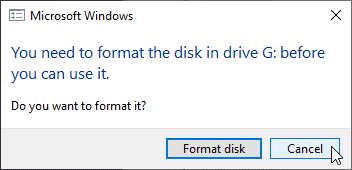
W okienku upewniamy się, że wybieramy dobrą literkę dysku w opcji *Select card* i klikamy Format aby przypadkiem nie usunąć danych z innego podłączonego przenośnego nośnika jak pendrive, nie przejmując się nazwą woluminu. Proces może zająć kilka minut w zależności od pojemności karty.

### Wypalanie systemu

Ściągnięty Raspbian Buster możemy zostawić spakowany w archiwum .rar:

Z kompletnym systemem uruchamiamy balenaEtcher i poprzez przycisk *Select image* wskazujemy lokalizację obrazu w archiwum.

Ostatnim punktem jest wciśnięcie przycisku *Flash!,* który uruchamia proces wypalania systemu na karcie. Zajmie to kilka minut (około 3-5). W trakcie można normalnie korzystać z komputera jednak należy uważać by nie rozłączyć adaptera, gdyż wymagane będzie ponowne formatowanie karty. Na koniec program zweryfikuje poprawność nagranych danych (3-4 minuty).

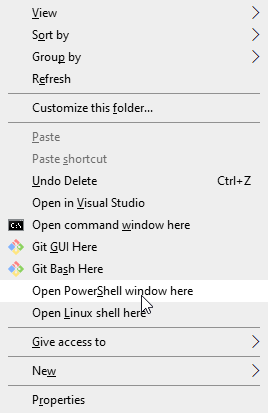


Po zakończeniu Windows prawdopodobnie spróbuje odczytać dane z karty i wyświetli komunikat o konieczności sformatowania karty. Należy zignorować ten komunikat i pod żadnym pozorem nie formatować ponownie karty narzędziami Windowsa.

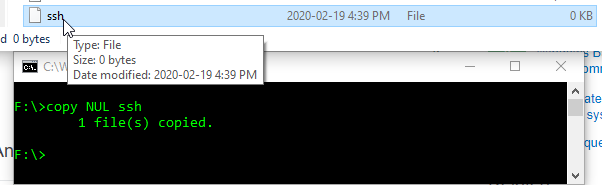
### Konfiguracja systemu

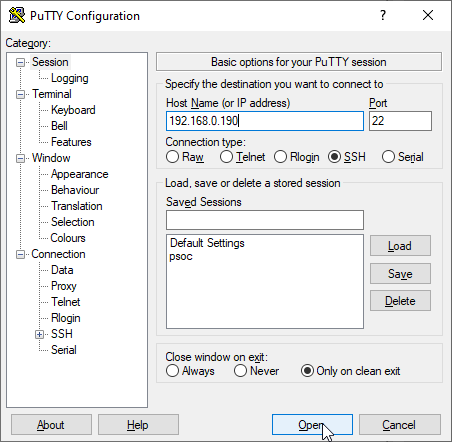
Wypalony Raspbian nadal nie ma zainstalowanego programu ani wymaganych bibliotek. Zależnie od dostępnego sprzętu Raspberry Pi można skonfigurować używając kilku metod. Dla przeciętnego użytkownika najprostsze może być podłączenie do RPi klawiatury, myszki (opcjonalnie) i ekranu przez HDMI. Dodatkowo konieczne będzie połączenie z Internetem poprzez Wi-Fi (RPi 3B+), lub kabel Ethernet. Ta metoda jest opisana jako ostatnia.

#### SSH

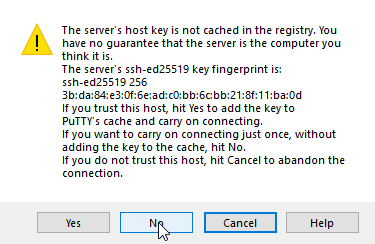
Ponowne umieszczenie adaptera w komputerze powinno wyświetlić dwa dyski w eksploratorze Windowsa:

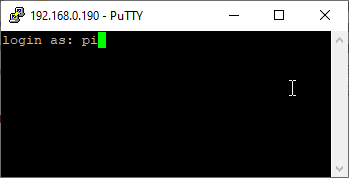
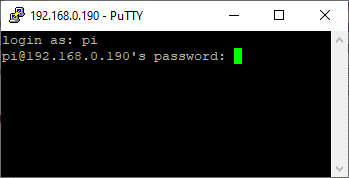
Otwieramy dysk z nazwą *boot.* W jego środku należy stworzyć plik bez żadnej zawartości, o rozmiarze 0B i nieposiadający żadnego rozszerzenia. Jedynym „niezerowym” elementem tego pliku powinna być nazwa „***ssh***”*.* W Windowsie taki plik można stworzyć lepszym edytorem tekstu jak Notepad++ lub poprzez wiersz polecenia:

Trzymając lewy shift klikamy prawym przyciskiem myszy w folderze boot i otwieramy okno PowerShell lub wiersz polecenia. W nowootwartym oknie używamy polecenia *copy NUL ssh.* Wielkość liter ma znaczenie i poprawne wykonanie polecenia powinno utworzyć plik ssh:

Tak przygotowaną kartę można już umieścić w Raspberry Pi, i podłączyć je do zasilania i tej samej sieci za pomocą kabla Ethernet. W tej i następnej metodzie konieczne jest znalezienie adresu IP Raspberry. Najłatwiej uzyskać je z pomocą strony internetowej routera – najczęściej <http://192.168.0.1/> lub <http://192.168.1.1/>. W jednej z rubryk powinny być widoczne adresy IP oraz MAC podłączonej do sieci Raspberry Pi (oraz wszystkich innych urządzeń w danej sieci):

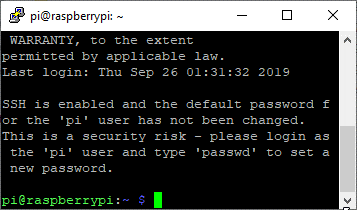
Kopiujemy adres IP i otwieramy ściągniętego wcześniej klienta SSH np. PuTTY. W programie wybieramy opcję SSH. Adres należy umieścić w odpowiedniej rubryczce, zostawiając domyślny port 22. Następnie otwieramy połączenie poprzez przycisk *Open.*

Prawdopodobnieuzyskamy ostrzeżenie postaci widocznej obok, które ignorujemy klikając na **Yes** lub **No**.

Powinno pojawić się okno logowania do systemu. Używamy domyślnego loginu i hasła dla systemu Raspbian i potwierdzamy Enterem:

***pi***

***raspberry***

 Znaki hasła nie będą widoczne. Powinien przywitać nas zielony kolor i ostrzeżenie sugerujące zmianę hasła użytkownika z domyślnego.

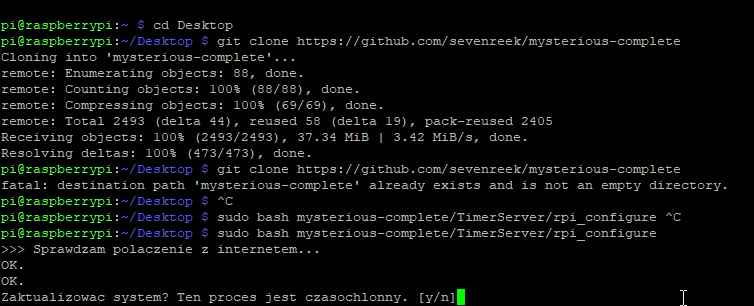
Do ukończenia konfiguracji należy teraz tylko ściągnąć repozytorium z githuba i uruchomić plik *rpi\_configure* z uprawnieniami administratora. Zrobimy to następującym ciągiem poleceń:

cd Desktop

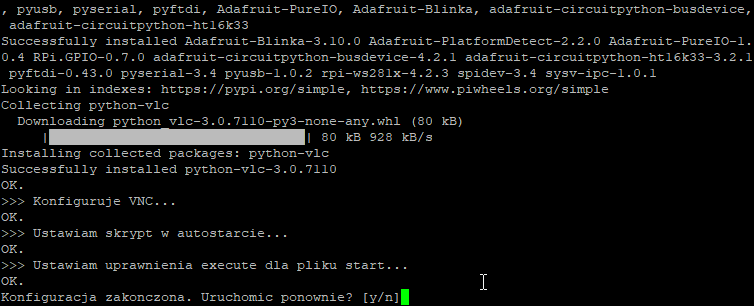
git clone https://github.com/sevenreek/mysterious-complete

sudo bash mysterious-complete/TimerServer/rpi\_configure

Przywita nas następujące pytanie. Aktualizacja systemu nie jest konieczna do prawidłowego funkcjonowania, a faktycznie jest bardzo czasochłonnym procesem, zwłaszcza na starszych wersjach Raspberry Pi. Możemy ją bezpiecznie pominąć wciskając klawisz ***n***.



Plik powinien bez problemu skonfigurować całą niezbędną funkcjonalność w ciągu kilku minut. Egzekucja skończy się pytaniem o restart płytki. Po ponownym uruchomieniu program powinien wystartować wraz z uruchomieniem się pulpitu i stać się widoczny w aplikacji administracyjnej. Wciskamy ***y*** aby uruchomić płytkę ponownie. Zamknie to naszą sesję SSH z błędem o utracie połączenia.



Po restarcie możemy ponownie łączyć się z Raspberry Pi w celach rekonfiguracji wykorzystując tę samą metodę, pomijając oczywiście polecenia do ściągnięcia projektu (git clone) i pierwszej konfiguracji (rpi\_configure).

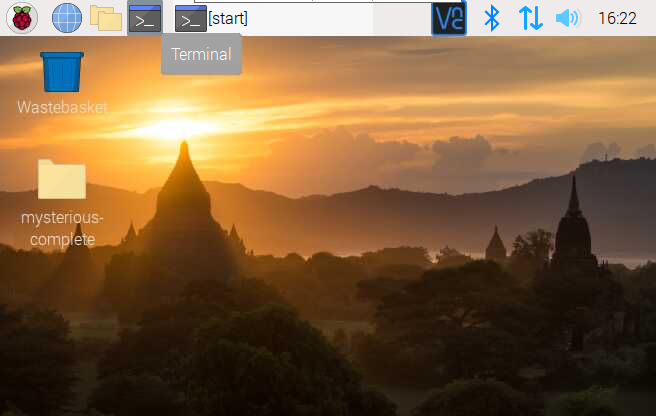
#### Klawiatura, myszka.

Pierwsze uruchomienie Raspberry Pi powinno pokazać ekran powitalny proszący o konfigurację. Możemy ją wykonać lub zignorować. Należy otworzyć terminal i wprowadzić do niego dokładnie ten sam ciąg poleceń co w sekcji SSH:

cd Desktop

git clone https://github.com/sevenreek/mysterious-complete

sudo bash mysterious-complete/TimerServer/rpi\_configure

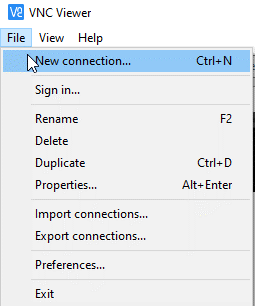
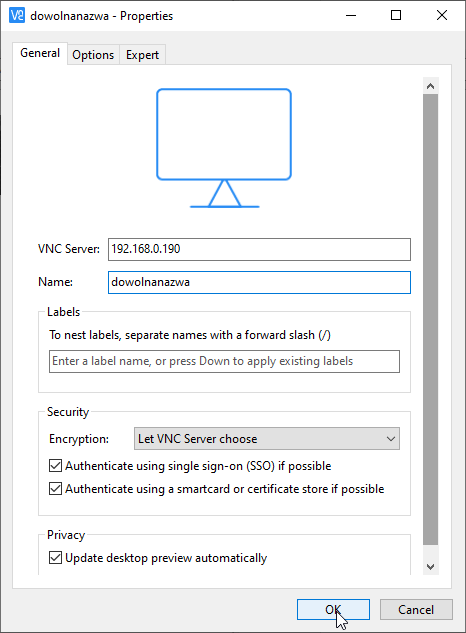


# Ustawienia pokoju

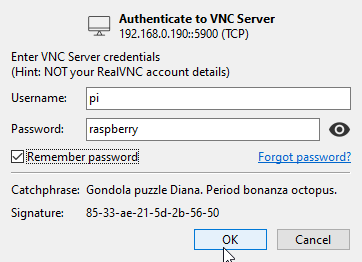
Oprócz wcześniej wspomnianych ustawień portów do komunikacji folder TimerServer zawiera pliki konfiguracyjne wielu innych aspektów.

# Funkcje dodatkowe

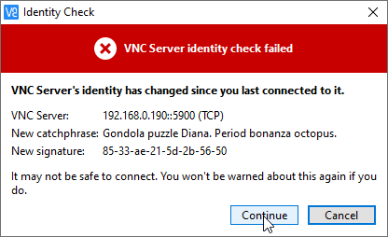
## Zdalny pulpit VNC

Konfiguracja Raspberry Pi za pomocą *rpi\_configure* instaluje na każdej z nich serwer zdalnego pulpitu VNC. Aby się z nim połączyć otwieramy VNC Viewer i tworzymy nowe połączenie.

Konieczna jest znajomość adresu IP danego Raspberry Pi, który to wpisujemy w pole VNC Server. W pole Name wpisujemy przyjazną użytkownikowi nazwę, pamiętając, że jeśli router nie został skonfigurowany by nadawać statyczne adresy każdemu z urządzeń przy następnym uruchomieniu adres ten może być przypisany innemu lub żadnemu urządzeniu. Metody znajdywania adresu opisane są w sekcji SSH**.**

Podwójne kliknięcie na powstały widok otworzy ostrzeżenie, które ignorujemy i logujemy się do urządzenia. Domyślnie:

***pi***

***raspberry***

Zminimalizowanie terminala z pracującym programem oraz zignorowanie ostrzeżenia o domyślnym haśle systemu wyświetli standardowy pulpit Raspbiana

