

# **Setup Orange Pi Zero - H2 for CLEES**

Edition: 2017-12-31

by

Tompa

## Table of Contents

Introduktion.....	3
Om CLEES.....	3
Akronymer.....	3
Steg för steg installation.....	3
1. Förbered SDkortet.....	4
1.1 Ladda ned Mainline.....	4
1.2 Packa upp filen.....	4
1.3 Sätt SD-card i datorn.....	4
1.4 Skriv diskavbildningen till SD-card:et.....	4
2. Starta Opi0 för första gången.....	5
2.1 Sätt i SD-card:et i Opi0.....	5
2.2 Anslut nätverk och matningsspänning.....	6
2.3 Kontakta Opi0.....	6
2.4 Logga in på Opi0 första gången.....	7
2.5 Boota om.....	8
2.6 Filhantering via SFTP.....	8
2.7 Stäng av Opi0 kontrollerat med "halt".....	9
3 Grundinställningar i Armbian.....	10
4 Förbereda hårdvaran för CLEES.....	11
4.1 Löd på en stiftlist.....	11
4.2 Aktivera i2c.....	11
4.3 Testa i2c.....	12
4.4 Installera drivrutin för PWM-kortet PCA9685.....	13
4.5 Testa PCA9685 med ett RC-servo.....	14

## Introduktion

Denna guide förklara hur du förbereder och kör igång en Orange Pi Zero enkortsdator i allmänhet och hur du sätter upp den för att fungera som styrenhet och dekoder i ditt CLEES system.

## Om CLEES

CLEES är en akronym för "Controll your Layout using Ethernet and Easy Scripts" och är ett öppet system för att styra växlar och signaler på din modelljärnväg.

## Akronymer

Opi0      Orrange Pi Zero

## Steg för steg installation

Innan du börjar installationen av mjukvara på Opi0 behöver du installera några program på din dator. Om du redan har dessa program eller motsvarande så kan du hoppa vidare till nästa steg

1. Programmet 7-zip, som används för att packa upp filer du laddar ned från internet.  
7-zip hittar du här:  
<http://www.7-zip.org/>
2. Programmet Win32 Disk Imager, som används för att läsa och skiva avbilder av SD-cards. Win32 Disk Imager hittar du här:  
<https://sourceforge.net/projects/win32diskimager/>
3. Programmet SD memory card Formatter, som används för att formatera SD-cards. SD memory card Formatter hittar du här:  
[https://www.sdcard.org/downloads/formatter\\_4/](https://www.sdcard.org/downloads/formatter_4/)
4. Programmet Bitvise, som är ett integrerat program med SSH terminal och SFTP klient för att logga in på och kommunicera med Orange Pi Zero.  
Bitvise hittar du här:  
<https://www.bitvise.com/ssh-client-download>

Alla program här listade är fria att använda (iallafall när detta skrevs).

## 1. Förbered SDkortet

Opi0 ska laddas med Linuxdistributionen Armbian. Armbian kan köras på många olika enkortsdatorer. Du ska välja att ladda ned den som är till "Orange Pi Zero"

<https://www.armbian.com/orange-pi-zero/>

### 1.1 Ladda ned Mainline.<sup>1</sup>

#### 1.2 Packa upp filen

Packa upp den nedladdade filen med med 7-zip till en temporär katalog.

#### 1.3 Sätt SD-card i datorn

Opi0 vill ha ett micro SD-card så du kan behöva en adapter till kortet för att sätta det i datorn.



Armbian  
linux for ARM development boards

Download Getting started Sources Forum Donate

Orange Pi Zero

stbwinner H2+ legacy SPI flash WiFi

Ubuntu server – legacy kernel .torrent (recommended) ?  
Command line interface – server usage scenarios.  
Stable

Ubuntu server – mainline kernel .torrent (recommended) ?  
Command line interface – server usage scenarios.  
Testing

Du skall välja ett SD-card med speedclass = 10 eller bättre.<sup>2</sup>

#### 1.4 Skriv diskavbildningen till SD-card:et

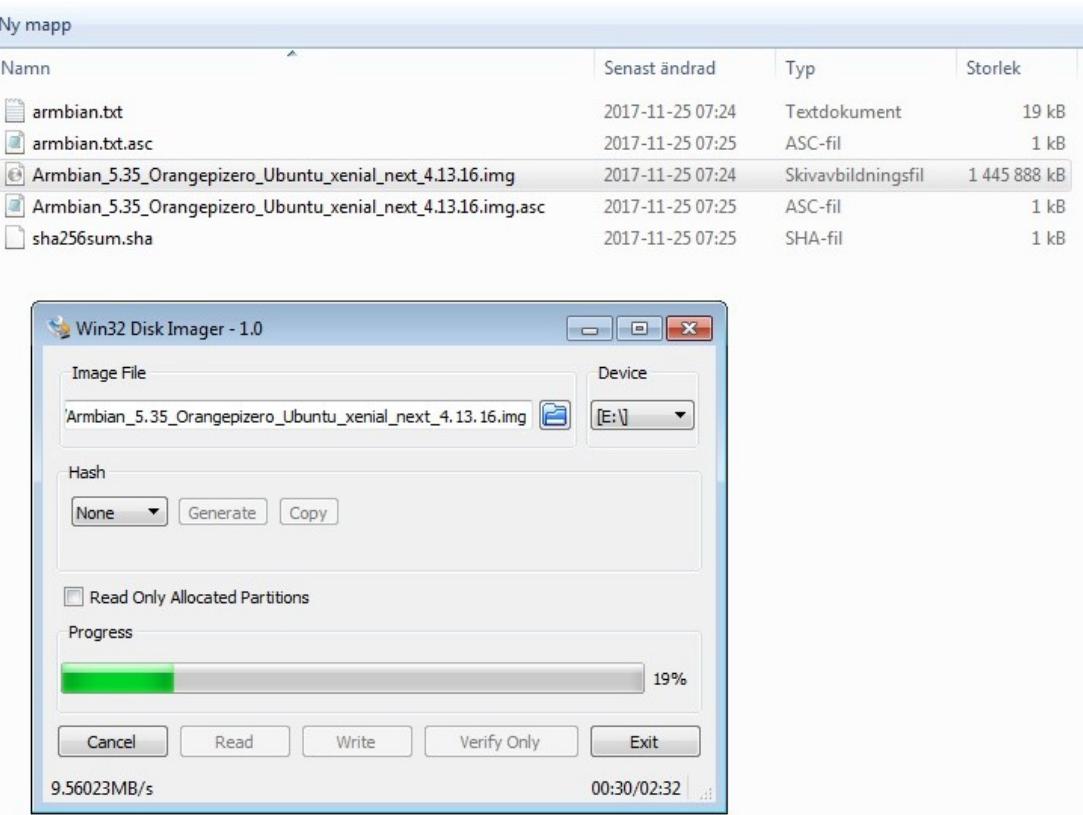
Använd programmet Win32 Disk Imager för att skriva ned Armbianfilen med efternamn .img till SD-card:et. I Win32 Disk Imager programmet väljer du först img fil och sedan klickar du på "Write" knappen så börja programmet skriva Arbianfilen till disken.

Allt som finns på SD-card:et kommer skrivas över. Du behöver normalt inte först formatera SD-card:et om det är ett nytt kort. Men om det skulle behövas så kan du använda programmet SD memory card Formatter.

1 Även om det är vanligt att välja den stabila versionen så är det så i Armbians fall att alla beskrivningar på nätet hänvisar till Mainline och flera inställningar i CLEES beskrivningar går inte att göra i den stabila versionen. Båda versionerna anses stabila men supporten variera.

2 Ett tips är att använde ett SDcard med så lite minne som möjligt. Gärna 8GB om du kan hitta. Det underlättar när man ska göra backupper av hela SD-disken om kortet är litet. Stora kort kan ta timmar att avbilda.

När filen skrivas till disken kan du se skrivhastigheten nedtill i programmet. Den ska vara 10MB/s eller högre. Då vet du att ditt SD-card är class 10 eller bättre



## 2. Starta Opi0 för första gången

### 2.1 Sätt i SD-card:et i Opi0



## 2.2 Anslut nätverk och matningsspänning

Koppla in Opi0 till ditt nätverk. Nätverket måste ha en DHCP server/router som kan dela ut IP-adresser dynamiskt. Alla vanliga nätverk kan detta och har detta aktiverat.

Så snart du kopplar ström till Opi0 kommer den automatiskt att börja installationen som tar ca 35 sekunder. STÄNG INTE AV Opi0 UNDER TIDEN DEN STARTAR UPP!

Förloppet under denna tid kan variera men följande brukar hända:

0 sek – Du ansluter strömmen till Opi0

3 sek – Gul LED i Etherneuttagen blinkar något

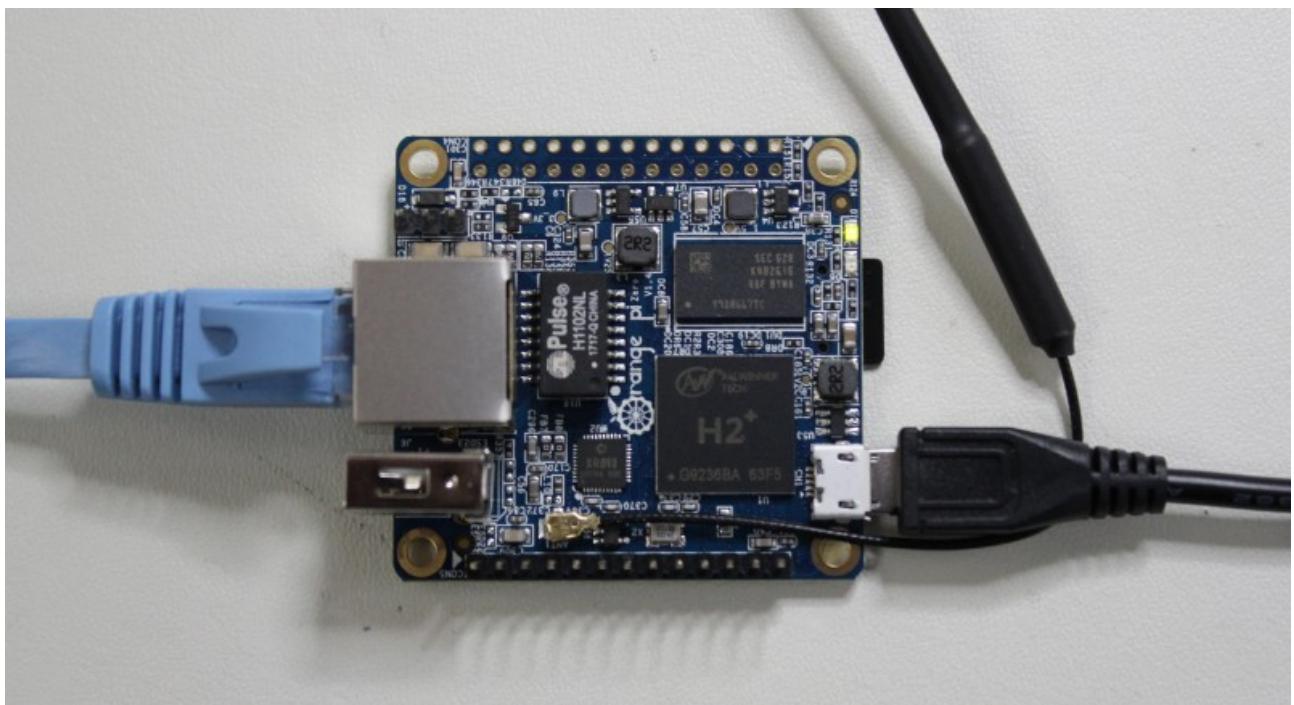
5 sek – Både gul och grön LED i ethernetuttaget blinkar till.

10 sek – Den lilla gröna LEDen på kortet tänds

15-20 sek - Både gul och grön LED i ethernetuttaget blinkar.

30 sek – Opi0 gör en DHCP förfrågan på nätverket.

35 sek – Nu kan du ansluta med SSH till Opi0



## 2.3 Kontakta Opi0

Opi0 har fått en IP-adres från din DHCP server. Du behöver Opi0's IP-adress för att kunna logga in på Opi0. För att få reda på vilken adress Opi0 har kan du logga in på din router och kolla i listan över IP-adresser som är utdelade.<sup>3</sup>

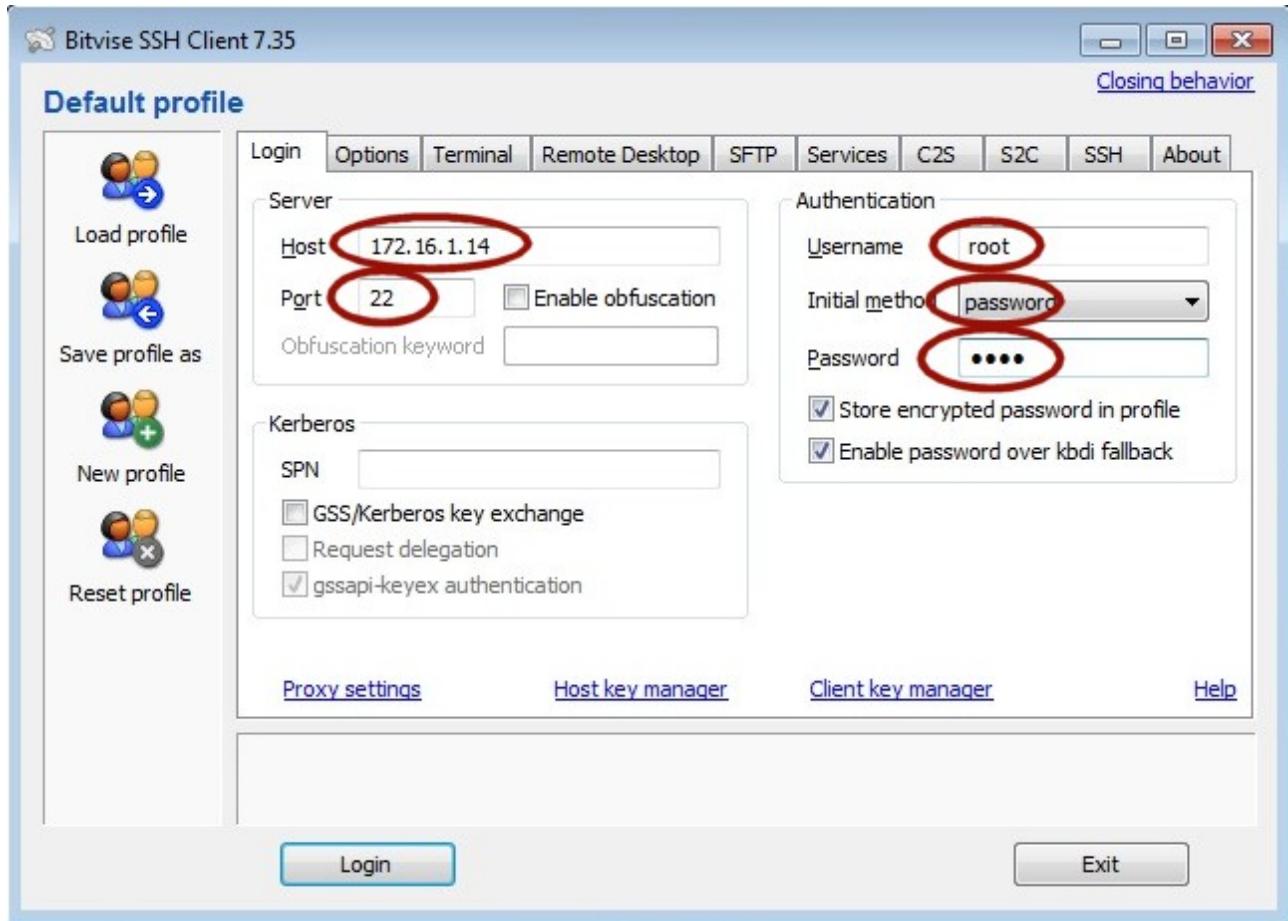
För att logga in på Opi0 används bäst programmet Bitvise. Vill du inte använda Bitvise så fungerar andra också bra, tex SSH klient typ Putty och SFTP via WinSCP.

Fortsättningsvis kommer anslutning till Opi0 beskrivas med programmet Bitvise.

<sup>3</sup> Du kan också använda ett program som LANeye som övervakar nätverket och upptäcker när någon ansluter till nätverket. Då syns direkt att Opi0 har anslutit och vilken IP-adress den har.

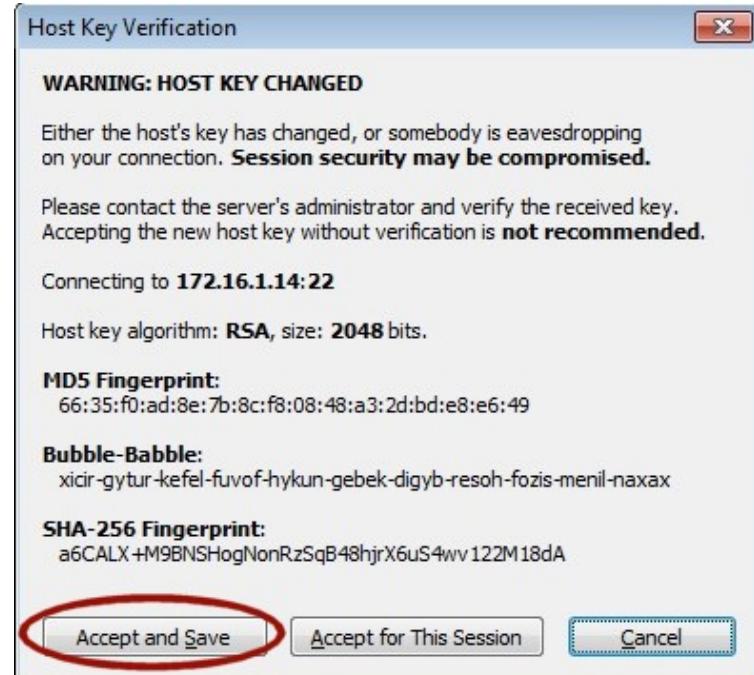
## 2.4 Logga in på Opi0 första gången

Starta Bitvise och ange IP-adress, Port=22, username = root och Password = 1234, allt enligt bilden nedan



Du kommer få en varning som denna men det är inte så allvarligt Det beror på att Bitvise och Opi0:an inte har träffats förut.

Acceptera och spara



Första gången du ansluter till Opi0:an så tvingas du byta lösenord på root kontot.

```
root@172.16.1.14:22 - Bitvise xterm
Changing password for root.
(current) UNIX password: [REDACTED]
```

Skriv upp lösenordet så du inte glömmer bort det. Tappar du bort det så kan du behöva installera om alltihop från början igen.

Du kommer uppmanas göra ett användarkonto. Gör det genom att följa instruktionerna på skärmen.

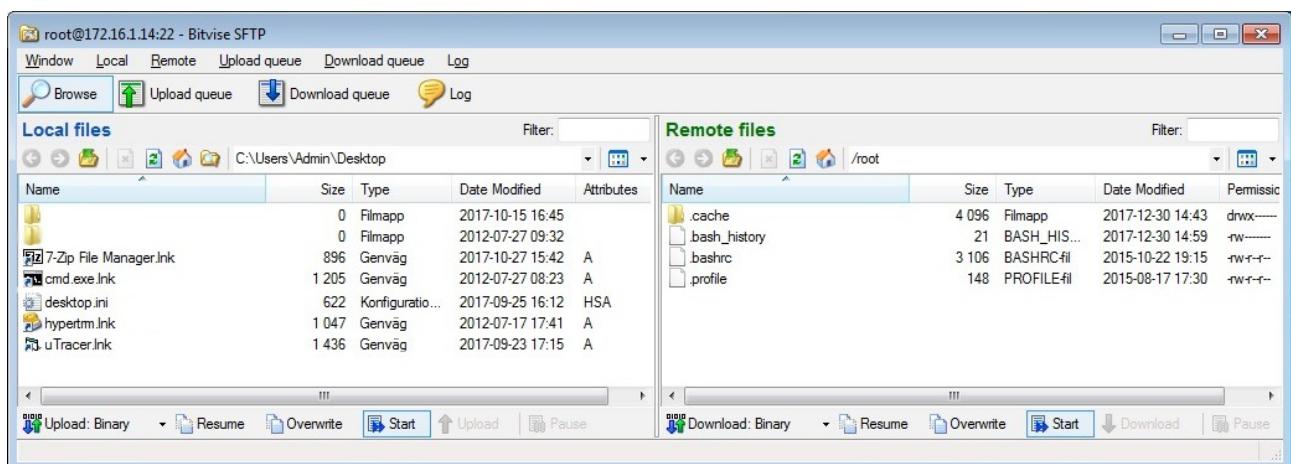
## 2.5 Boota om

Avsluta din första session med att skriva ordet "reboot"

Nu kopplar Opi0:an ned förbindelsen, blinkar med sina LEDar och återkommer efter ca 15-20 sekunder. Nu kommer Bitvise fråga efter lösenordet igen för det har ju ändrats. Skriv in ditt nya rootlösenord och du kommer in.<sup>4</sup>

## 2.6 Filhantering via SFTP

Bivise öppnar ett FTP-fönster. Där kan du kopiera och flytta/skapa filer på Opi0:ans filsystem.



4 Du kan spara dina kontonställningar i Bitvise genom att spara en profil. Gör en profil för root och en annan profil för ditt konto du skapade nyss.

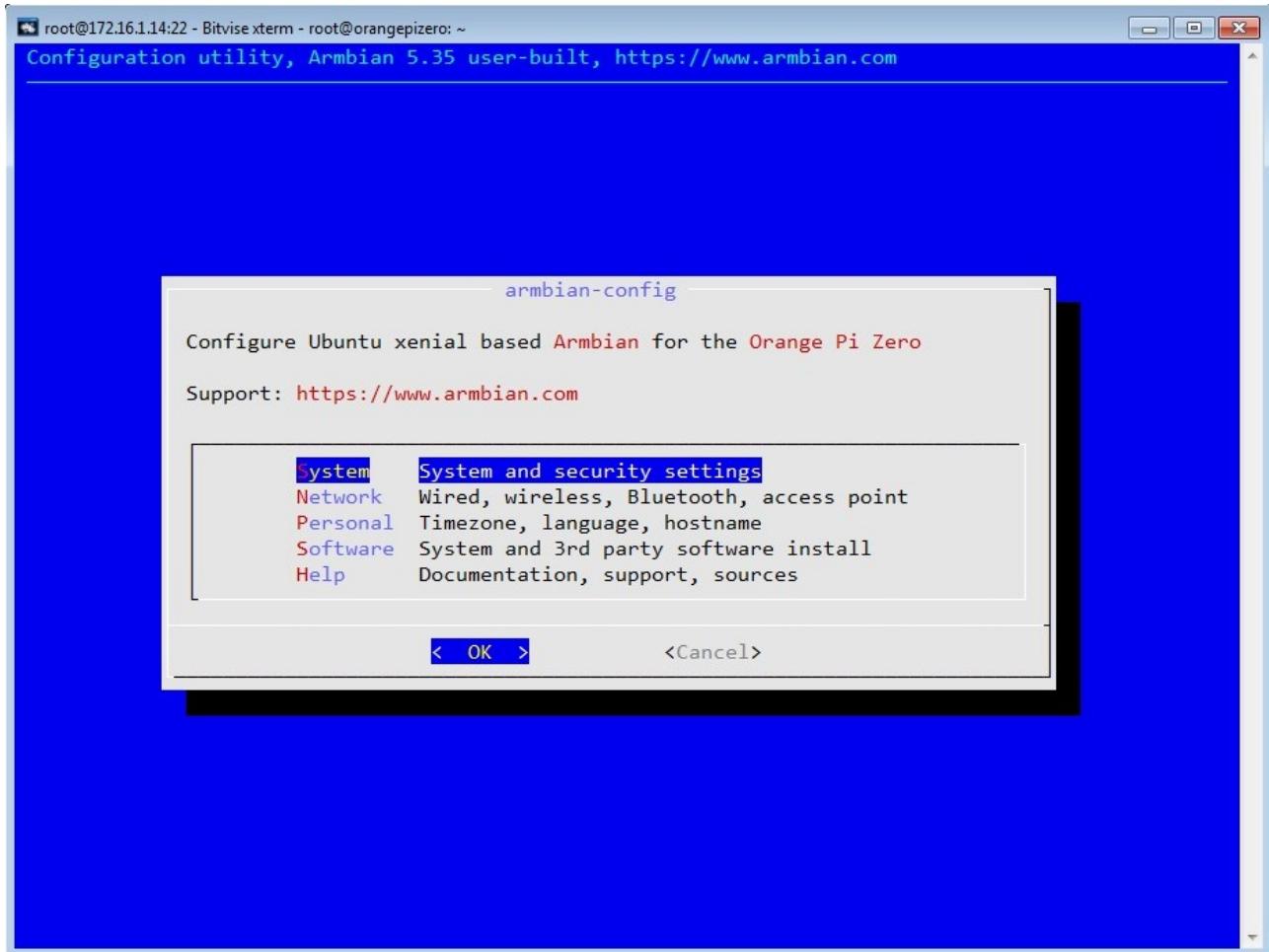
## 2.7 Stäng av Opi0 kontrollerat med "halt"

En linuxdator är alltid lite känslig för abrupt avstängning. Du ska alltid innan du tar bort strömmen först ge kommandot "halt". Om du inte är inloggad som root kan du behöva skriva "sudo halt".

Om du glömmer ge kommandot "halt" finns risk för att filsystemet på SD-card:et blir skadat.

### 3 Grundinställningar i Armbian

Följande grundinställningar i Armbian bör man göra. För det använder du det inbyggda config-stödet. Logga in som root och skriv "armbian-config" och configverktygen startar.



Gör dig familjär med detta verktyg för du kommer behöva göra flera inställningar längre fram.

Nu ska du ändra

Tidzon till Europa och Stockholm (eller annan världsdel/stad där du befinner dig).

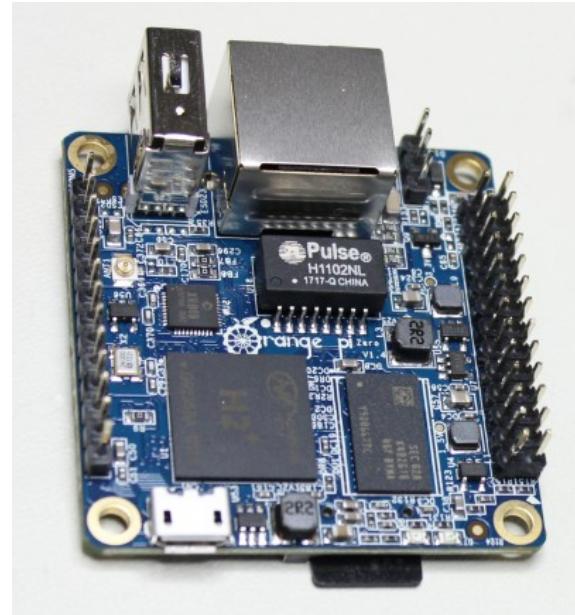
## 4 Förbereda hårdvaran för CLEES

CLEES använder i/o-pinnar för att kommunicera med annan hårdvara.

### 4.1 Löd på en stiftlist

Opi0 kommer med en 26polig anslutning för att kommunicera med annan hårdvara via i2c eller GPIO. Men Opi0 kortet saknar stiftlist så en sådan behöver du löda dit.

När du lött dit den ska det se ut som på bilden intill.



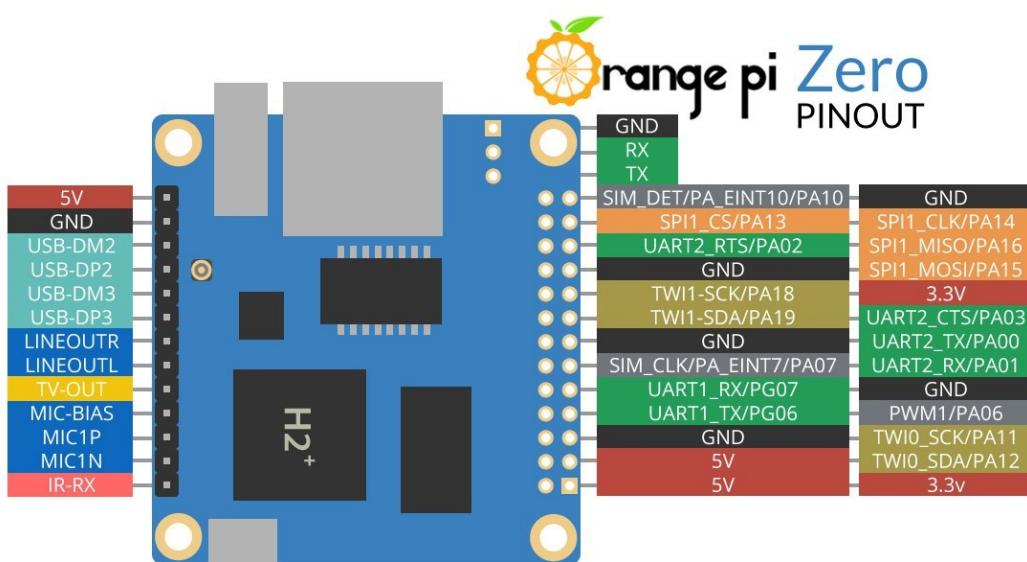
### 4.2 Aktivera i2c

Vi behöver aktivera en i2c-kanal som skall användas för att kommunicera med andra kort. Aktiveringens görs i armbian-config

Starta armbian-config. Gå in under "System" -> "Hardware" och aktivera i2c0 och spara.

Reboot för att inställningen ska träda i kraft

Nu har vi en I2c-buss på pin 3+5 i den 26poliga kontakten.



## 4.3 Testa i2c

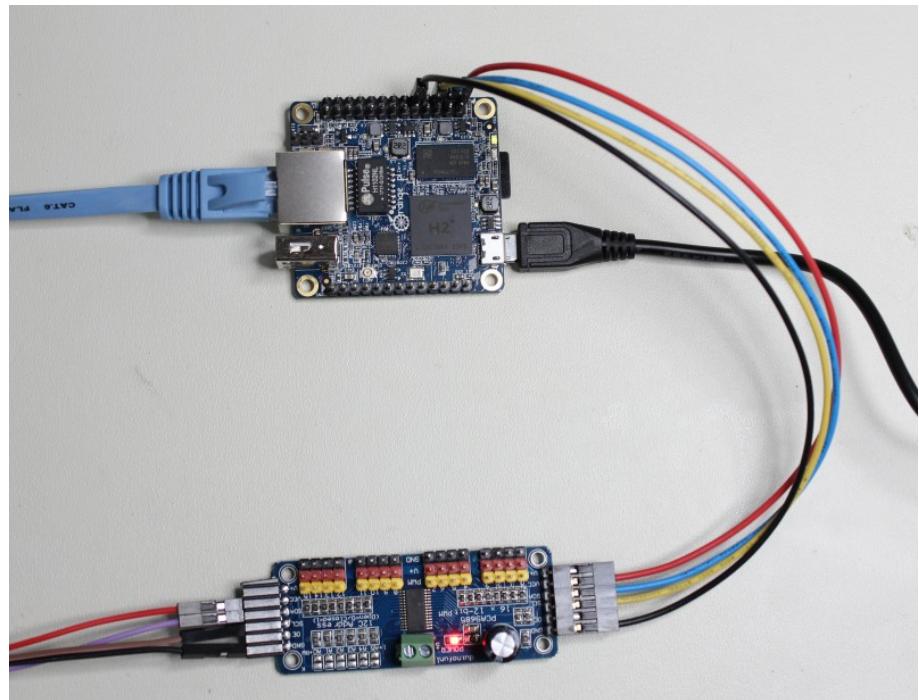
Vi ska installera ett litet bra verktyg för i2c som kan detektera vilka adresser på i2c-bussen som det sitter abonnenter på.

Kör kommandot "apt-get install i2c-tools"

När det är klart, skriv följande kommando "i2cdetect -y 0".  
och terminalen kommer visa en tabell över alla i2c adresser på buss 0

```
root@orangepizero:~# i2cdetect -y 0
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: --
10: --
20: --
30: --
40: --
50: --
60: --
70: --
```

Eftersom inget ännu är kopplat till bussen hittas ingen abonnent. Men om vi kopplar in ett av våra CLEES kort, tex ett PWM kort



Så ser det ut såhär

```
root@orangepizero:~# i2cdetect -y 0
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: --
10: --
20: --
30: --
40: 40 --
50: --
60: --
70: 70 --
```

Nu har vi en fungerande i2c buss.

## 4.4 Installera drivrutin för PWM-kortet PCA9685

Det finns en drivrution för Adafruits PWM-kort som fungerar fint även till andra kort så länge dom är bestyckade med PCA9685 kretsen.

Du hittar den på Github, här

[https://github.com/adafruit/Adafruit\\_Python\\_PCA9685](https://github.com/adafruit/Adafruit_Python_PCA9685)

Du kan installera direkt från github. Ge följande kommandon:

```
sudo apt-get install git build-essential python-dev
```

```
cd ~
```

```
git clone https://github.com/adafruit/Adafruit_Python_PCA9685.git
```

```
cd Adafruit_Python_PCA9685
```

```
sudo python setup.py install
```

## 4.5 Testa PCA9685 med ett RC-servo

Koppla in ett servo till första stiftlisten, se bilden. Anslut också 5VDC till skruvplinten

Test görs bäst med ett pythonskript.

- Skapa ett testbibliotek, lämpligtvis under root. Kalla biblioteket "test". (Använd SFTP klienten i Bitvise)
- Skapa en fil i testbiblioteket och klistra in servotest.py koden nedan"
- Spara filen (servotest.py)

Via terminalen kör den nya filen

```
cd test
python servotest.py
```

Servot ska nu röra sig fram och tillbaka.

```
--- servotest.py ---
from __future__ import division
import time

# Import the PCA9685 module.
import Adafruit_PCA9685

# Initialise the PCA9685 specify a address and/or bus:
pwm = Adafruit_PCA9685.PCA9685(address=0x40, busnum=0) #bus0 = i2c0

# Configure min and max servo pulse lengths
servo_min = 250 # Min pulse length out of 4096
servo_max = 400 # Max pulse length out of 4096

# Helper function to make setting a servo pulse width simpler.
def set_servo_pulse(channel, pulse):
    pulse_length = 1000000 # 1,000,000 us per second
    pulse_length /= 60 # 60 Hz
    print('{0}us per period'.format(pulse_length))
    pulse_length /= 4096 # 12 bits of resolution
    print('{0}us per bit'.format(pulse_length))
    pulse *= 1000
    pulse /= pulse_length
    pwm.set_pwm(channel, 0, pulse)

# Set frequency to 60hz, good for servos.
pwm.set_pwm_freq(60)

print('Moving servo on channel 0, press Ctrl-C to quit...')
while True:
    # Move servo on channel 0 between extremes.
    pwm.set_pwm(0, 0, servo_min)
    time.sleep(1)
    pwm.set_pwm(0, 0, servo_max)
    time.sleep(1)
#-----
```

