

# Installatie van VM voor gebruik Unitree GO1

---

Installatie wordt gedaan op een nieuwe linux, hier het geval op een VM in Virtual Box. Na alles gevolgd te hebben kan de robot aangestuurd worden door herkenning van personen.

1. [Initiële settings VM](#)
2. [Script person detection](#)
  1. [Basis installaties](#)
  2. [Github clonen](#)
  3. [Installaties van nodige programma's](#)
3. [ROS2](#)
  1. [ROS2 installeren](#)
  2. [MQTT subscriber](#)
4. [Verbinden met de robot](#)
5. [CameraSDK voor de robot](#)
  1. [UnitreeCameraSDK op lokale linux](#)
  2. [UnitreeCameraSDK op robot](#)
  3. [Testen van installaties](#)
    1. [Streamer opzetten op de robot](#)
    2. [Streamer ontvangen op de lokale linux](#)
6. [Alles opzetten](#)
  1. [Robot](#)
    1. [Streamer opzetten](#)
    2. [MQTT subscriber aanzetten](#)
    3. [Twister aanzetten](#)
  2. [Lokale linux machine](#)

## 1. Initiële settings VM

- 3D acceleratie aan

## 2. Script person detection

Om het script voor het detecteren van personen en hun afstand te berekenen, zijn enkele installaties en stappen nodig.

### 2.1 Basis installaties

Het linux account aan sudo toevoegen.

```
CD
vi /etc/sudoers
```

Basis installaties voor een nieuwe VM

```
apt-get install git
sudo apt-get update
sudo apt-get install python3
sudo apt install python3-pip
```

## 2.2 Github clonen

```
git clone https://github.com/Bo-vcu/ucll-person-detection
```

Het script is te vinden in PersonDetection -> detection\_classes.py

Het model in PersoDetection -> models

## 2.3 Installaties van nodige programma's

```
pip3 install opencv-python
```

We raden u aan om Tensorflow te gebruiken boven ultralytics, omdat dit de gstreamer automatisch kan uitzetten.

Als u wel graag Ultralytics wil gebruiken, weet dat u andere dingen moet aanpassen of eerst een snapshot kan maken van de VM om daarna te kunnen testen met de installatie van Ultralytics.

Installatie van mqtt:

```
pip install paho-mqtt
```

## 3 ROS2

### 3.1 ROS2 installeren

Uitgebreide informatie is te vinden op deze website: <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>

Deze commando's moeten uitgevoerd worden op de lokale linux machine:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo apt update && sudo apt install curl -y
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-  
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release &&  
echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list >  
/dev/null  
sudo apt update  
sudo apt upgrade  
sudo apt install ros-humble-ros-base  
sudo apt install ros-dev-tools
```

Als je commando uitvoert wat iets te maken heeft met ROS2 altijd dit commando eerst uitvoeren:

```
source install/setup.bash
```

## 3.2 MQTT subscriber

MQTT installeren door.

```
sudo apt install -y mosquitto
```

## 3.3 Twister

Deze installatie moeten gedownload worden.

```
sudo apt install ros-humble-xacro
```

# 4 Verbinden met de robot

Vooralleerst moet de lokale machine verbonden zijn met hetzelfde netwerk als de robot. Dit wil zeggen dat op de linux machine het ip adres moet aangepast worden naar 192.168.123.162/24. Voor de VM moet het netwerk type gewijzigd worden van NAT naar Bridged adapter, Naam naar de Ethernet adapter en Promiscuous-modus 'Alle toestaan'. Hierdoor verdwijnt de internettoegang op de VM, dus voor internettoegang terug veranderen naar NAT en vice versa.

```
sudo ip a add 192.168.123.162/24 dev enp46s0
```

Daarna met de robot zelf verbinden.

```
ssh unitree@192.168.123.13
```

password: 123

## 5 CameraSDK voor de robot

### 5.1 UnitreeCameraSDK op lokale linux

Haal eerst de SDK op via git.

```
git clone git@github.com:unitreerobotics/UnitreecameraSDK.git
```

Enkele installaties zijn nodig.

```
sudo apt install libopencv-dev  
sudo apt-get install libglew-dev libglfw3-dev libglm-dev
```

Enkele aanpassingen aan sommige bestanden is nodig.

Voor de **trans\_rect\_config.yaml**:

```
diff --git a/trans_rect_config.yaml b/trans_rect_config.yaml  
index db182b8..722b80f 100644  
--- a/trans_rect_config.yaml  
+++ b/trans_rect_config.yaml  
@@ -23,13 +23,13 @@ IpLastSegment: !!opencv-matrix  
    rows: 1  
    cols: 1  
    dt: d  
-   data: [ 15. ]  
+   data: [ 162. ]  
#DeviceNode  
DeviceNode: !!opencv-matrix  
    rows: 1  
    cols: 1  
    dt: d  
-   data: [ 0. ]  
+   data: [ 1. ]  
#fov (perspective 60~140)  
hFov: !!opencv-matrix  
    rows: 1  
@@ -59,7 +59,7 @@ Transmode: !!opencv-matrix  
    rows: 1  
    cols: 1  
    dt: d  
-   data: [ 0. ]  
+   data: [ 3. ]  
#Transmission rate(FPS) in the UDP transmitting process. <= FrameRate  
Transrate: !!opencv-matrix  
    rows: 1
```

En in **example\_getimagestrans**:

```

diff --git a/examples/example_getimagetrans.cc b/examples/example_getimagetrans.cc
index 9d21905..24011d1 100644
--- a/examples/example_getimagetrans.cc
+++ b/examples/example_getimagetrans.cc
@@ -35,14 +35,16 @@ gateway 192.168.123.1
#include <iostream>
int main(int argc, char** argv)
{
-   std::string IpLastSegment = "15";
+   // std::string IpLastSegment = "15";
+   std::string IpLastSegment = "162";
   int cam = 1;
   if (argc>=2)
       cam = std::atoi(argv[1]);
   std::string udpstrPrevData = "udpsrc address=192.168.123."+ IpLastSegment + "
port=";
   // 端口：前方·下巴·左·右·腹部
   std::array<int, 5> udpPORT = std::array<int, 5>{9201, 9202, 9203, 9204,
9205};
-   std::string udpstrBehindData = " ! application/x-rtp,media=video,encoding-
name=H264 ! rtph264depay ! h264parse ! omxh264dec ! videoconvert ! appsink";
+   //std::string udpstrBehindData = " ! application/x-rtp,media=video,encoding-
name=H264 ! rtph264depay ! h264parse ! omxh264dec ! videoconvert ! appsink";
+   std::string udpstrBehindData = " ! application/x-rtp,media=video,encoding-
name=H264 ! rtph264depay ! h264parse ! avdec_h264 ! videoconvert ! appsink";
   std::string udpSendIntegratedPipe = udpstrPrevData +
std::to_string(udpPORT[cam-1]) + udpstrBehindData;
   std::cout<<"udpSendIntegratedPipe:"<<udpSendIntegratedPipe<<std::endl;
   cv::VideoCapture cap(udpSendIntegratedPipe);

```

Daarna moet de SDK gebuild worden.

```

cd UnitreecameraSDK
mkdir build && cd build
cmake ..
make

```

Na aanpassingen aan de c++ scripts moet er opnieuw gebuild worden.

```

rm -rf build/
mkdir build && cd build
cmake ..
make

```

## 5.2 UnitreeCameraSDK op robot

Als de UnitreecameraSDK nog niet is geïnstalleerd is op de robot, kopieer je de UnitreecameraSDK omdat de Robot geen internet heeft. Voer dit script uit in de folder waar de UnitreecameraSDK lokaal staat.

```
scp -r UnitreecameraSDK unitree@192.168.123.13:/home/unitree/
```

Na het verbinden met de robot, [hier](#) beschreven, moet de SDK gebuild worden.

```
cd
cd /home/unitree/UnitreecameraSDK
mkdir build && cd build

cmake ..
make
```

## 5.3 Testen van installaties

### 5.3.1 Streamer opzetten op de robot

Voer op de robot het script **example\_putImagestrans** van de SDK uit.

```
./bins/example_putImagestrans
```

Hierna staat de streamer op en kan dit beeld ontvangen worden op de lokale linux machine.

### 5.3.2 Streamer ontvangen op de lokale linux

Nu kan de streamer ontvangen worden door **example\_getimagestrans** van de SDK uit te voeren.

```
./bins/example_getimagestrans
```

Oftewel ontvangen door de streamer zelf:

```
gst-launch-1.0 udpsrc port=9201 ! application/x-rtp, media=video, encoding-
name=H264 ! rtph264depay ! h264parse ! avdec_h264 ! videoconvert ! autovideosink
```

## 6 Alles opzetten

### 6.1 Robot

#### 6.1.1 Streamer opzetten

Eerst verbinden met de robot, [hier](#) beschreven.

Enkele processen moeten gekilled worden opdat de UnitreecameraSDK werkt.

```
ps -aux | grep point_cloud_node | awk '{print $2}' | xargs kill -9
ps -aux | grep mqttControlNode | awk '{print $2}' | xargs kill -9
ps -aux | grep live_human_pose | awk '{print $2}' | xargs kill -9
```

Hierna kan de streamer aangezet worden door example\_putImagestrans in de bins folder uit te voeren.

```
./bins/example_putImagestrans
```

### 6.1.2 MQTT subscriber aanzetten

Zet de MQTT subscriber aan in de betreffende map src.

```
source install/setup.bash
cd mqtt
ros2 run mqtt_ros mqtt_subscriber
```

### 6.1.3 Twister aanzetten

```
ros2 run twister twister
```

## 6.2 Lokale linux machine

Het script uitvoeren. Het script is te vinden in PersonDetection -> detection\_classes.py

```
python3 detection_classes.py
```