*Teleoperation of DOBOT Magician using Cyber Glove*

*2022*

*Index*

1. Step 1 : [Set up the ORACLE Virtual Box](#bookmark=id.gjdgxs)
2. Step 2 : [Arduino IDE Installation and ESP 32 Package Installation](#bookmark=id.1fob9te)
3. Step 3 : [DOBOT Magician Python API Interface and Setup](#bookmark=id.2et92p0)
4. Step 4 : [MQTT Protocol Implementation](#bookmark=id.tyjcwt)

*ORACLE Virtual Box*

*Note: If you work on an Ubuntu System, you can skip this step.*

**Installing Oracle Virtual Box (VB) :** Oracle Website -> Windows host -> download latest version

When you want to update Virtual Box to latest version, uninstall and download the latest file from their website. The virtual machines will be auto present when you open the updated virtual box. Don’t upgrade directly, as you might get kernel issues.

**Ubuntu ISO :** Go to Ubuntu website and download the iso file.

*I Setup new Virtual Machine in Oracle VB*

All downloads done within virtual box in ubuntu 20.04 .

Open the VB

1. *New virtual Machine*: Type: Linux

Version: Ubuntu 64bit

Hard Disk: 4000MB

Memory: 25GB

1. *System Settings* : Install Ubuntu Image (Downloaded earlier) from downloads
2. *USB Settings* :

* In settings click the **USB** tab.
* On the right side, click the usb icon with a **green** plus sign.
* Add anything with Silicon Labs USB to Uart Bridge controllers, or anything with Cygnal
* Integrated products CP210x Uart bridge. If you don't see these Options you many need
* to download drivers for your platform [here](https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers).

( <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers> )

* Select English, install Ubuntu
* Select keyboard, English(US)
* Select Normal Installation & Download updates while installing Ubuntu
* Since you are using a virtual machine you can select erase disk and install.
* Username : “Set your username”
* Pswrd : “Set your password”

*//Takes time to install.*

* Connect your online accounts : Skip
* Right ctrl (Host button) + F : Full screen
* But you might get black screen around the box
* So go to devices -> Insert guest additions CD image -> put your password : it will download -> then shutdown ubuntu -> and start again -> press right ctrl + F

For all Ubuntu distributions run these commands after a fresh install.

* **sudo apt install gcc g++ cmake putty unzip curl**
* **sudo apt update**
* **sudo apt upgrade**
* If you are on a virtual machine (Virtual Box), at the top of the window click **Devices**
  + **Shared Clipboard** set to **bidirectional**
  + **Drag and Drop** set to **bidirectional**
  + Finally click **Insert guest additions CD Image**
  + Once done installing in terminal input reboot

These steps are not mandatory. They just make working inside the virtual box easier.

*Arduino IDE Installation and ESP 32 Package Installation*

Use *snap* to download Arduino as given in the following website: <https://linoxide.com/how-to-install-arduino-ide-on-ubuntu-20-04/>

You need to type ./install.sh instead. The ./ indicates that you want to run a file in your current working directiry. Without the leading ./ linux will search in your $PATH for an executable program with that name.

Follow the following steps to add esp32 dev board package to the ide.

1. Go to File -> Preferences -> In additional boards manager URL : <https://dl.espressif.com/dl/package_esp32_index.json>
2. Then click Ok
3. Tools -> Boards -> No ES32 -> so go to Boards Manager -> type esp32 -> Install the package that shows up
4. Go to Tools -> Boards -> ESP 32 dev board

(<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-mac-and-linux-instructions/>)

(<http://arduino-er.blogspot.com/2020/06/install-esp32esp8266-to-arduino-ide-on.html>)

*Go through the above two URLs for further reference.*

# When IDE is first installed, no libraries are present. So install them from library manager and git hub profiles online as zip files and then add them.

*DOBOT Magician Python API Interface and Setup*

The DOBOT Magician has a DOBOT API interfaced with a python API. This API can be accessed using the PyDobot library available at : <https://github.com/luismesas/pydobot>

The setup for the same is as follows,

**Terminal:**

* Install: Python3 (You will need Python 2.7 or Python 3.x. Download “ <https://www.python.org/> “ and install.)
* Pip is a python library manager. Pip drivers and Pip module allow Python functions to run in terminal. ( <https://linuxize.com/post/how-to-install-pip-on-ubuntu-18.04/> )
* Install Pydobot (Install Pydobot using pip (command : **pip install pydobot**))

You can also download its zip file from: <https://github.com/luismesas/pydobot>

* You will also need pyserial module. **pip install pyserial** (If pyserial is not installed exceptions will be thrown when executing examples)
* Download the CP210x USB to UART Bridge VCP Drivers from <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>
* Use the commands ‘**groups**’ to check what groups are you a part of. Use ‘**sudo usermod -a ubuntu -G dialout**’ or ‘**newgrp dialout**’ to add yourself (username here: ubuntu) to *dialout* group and similarly to *ttl* group as well.
* The *dialout* group helps you control the usb ports without which you will be thrown the error ‘port access denied’.
* And every time you add yourself to a new group make sure you reboot.

**To Run the python scripts for Dobot control:**

* Open Ubuntu VB
* Open Terminal
* Check if *pydobot* library is installed : **pip3 show pydobot**
* Connect DOBOT Magician to laptop using USB port and power it on. The Dobot takes a few seconds to initialize.
* Check **Devices** -> **USB** -> **Silicon Labs** is detected
* Use **dmesg** command and **lusb** commands to identify what port is selected and make the corresponding changes in the script **(ttyUSB0/1/2 \*)**

( **USB tab** under **devices** may not show the **silicon labs port** if it isn’t added in the virtual box , so make sure to add it in **USB settings** in the **VB** first using the **green plus usb icon**.)

*Note*: *If the module you are trying to import i.e. the library , and any of your python files are named the same , then you will get a circular error. So make sure they are named differently.*

*Generally, the Python Circular Import problem occurs when you accidentally name your working file the same as the module name and those modules depend on each other. This way the python opens the same file again and again which causes a circular loop and eventually throws an error.*

* In Terminal : execute the file

**cd Documents/PyDobot** (location where the file is stored)

**python3 Py.py** (here Py.py is the file you are trying to run)

*Note:* To compile python files : **python -m py\_serial Py.py** or **python3 -m py\_serial Py.py**

*Note: Information on how to operate and troubleshoot technical issues with the Dobot Magician are provided in the manual – Dobot Magician User Guide, download-able from their website.*

*The Dobot Studio is their official application which can be downloaded from their website and installed on your system.*

***PyDobot*** *is an unofficial python library that utilizes the python API interfaced with the Dobot API for control of DOBOT Magician.*

Note**:** Fix port access on Linux : <https://support.arduino.cc/hc/en-us/articles/360016495679-avrdude-ser-open-can-t-open-device-Permission-denied-Linux->

It will give you the importance of ‘dialout’ group.

*Note : When you add yourself to a new group, for the effect to take place the system needs to be rebooted.*

*The USB connection is not handled by the ESP32, as an ESP32 only has serial lines (rx and tx), no USB. A dev board typically includes some sort of USB to serial converter chip (sometimes refered to as FTDI). This chip communicates with your computer, resulting in the creation of a virtual COM port. If you have a driver for that chip installed, otherwise it won't work. Without a driver the chip probably turns up as unsupported or unrecognized device in your hardware list. Although the ESP32 module may be the same, you might have a dev board with a different USB to serial converter than other devices you might have tested on your computer. So if one dev board works and another one doesn't, that does not necessarily mean the board is defective. You might just need to install a driver for the USB to serial chip on your dev board.*

*MQTT Protocol Implementation*

Refer: [Mosquitto MQTT Broker: Tutorial for Installation on Ubuntu 20.04 | ArubaCloud.com](https://www.arubacloud.com/tutorial/how-to-install-and-secure-mosquitto-on-ubuntu-20-04.aspx)

<http://mosquitto.org/>

*Programs included* : There are two folders : ***Dobot*** and ***ESP32***

**Dobot** contains the programs used for operating the Dobot Magician.

**ESP32** contains the programs running on both the esp32 s.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| THINGSPEAK | CloudWriteAvg | Glove | Writes the potentiometer data to ThingSpeak Cloud Server using write API key over HTTP protocol. |
| CloudReadAvg | ESP32 | Reads the potentiometer data from ThingSpeak Cloud Server using read API key over HTTP protocol. |
| FinalWspc.py | Terminal | Operates the DOBOTMagician in the positive workspace. |
| MOSQUITTO  (No Fidelity) | Mosquitto\_Write | Glove | Publishes potentiometer data to the MOSQUITTO Broker running on the computer using MQTT v3.1.1 . |
| Mosquitto\_Read | ESP32 | Subscribes the potentiometer date from the MOSQUITTO Broker running on the computer using MQTT v3.1.1 . |
| FinalWspc.py | Terminal | Operating the DOBOTMagician in the positive workspace. |
| MOSQUITTO (With Fidelity) | MosquittoWrite2 | Glove | Publishes potentiometer data to the MOSQUITTO Broker running on the computer using MQTT v3.1.1 . |
| PahoFinal.py | Terminal | Subscribes to the Mosquitto broker using Paho client. |
| Test\_Array | Glove | Publishes potentiometer data. Takes 20 values, each at an interval of 250ms. Publishes it at an interval of 7s. |
| PahoFinal2.py | Terminal | Subscribes and executes the 20 values. |
| PahoFinal2copy.py | Terminal | Subscribes and executes the values but has a buffer to continuously take values from the glove. |
| Serial | Serial | Glove | Sends Potentiometer Data over serial communication |
| FinalWspc.py | Terminal | Operates the DOBOTMagician in the positive workspace. |

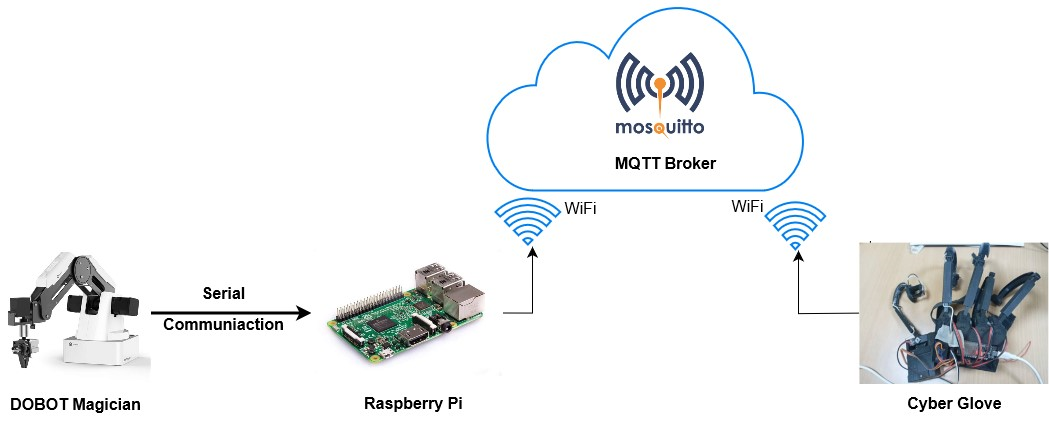
Extra Programs:

PyDobot2.py : Vanilla program used to operate pydobot. Simple demonstration of the PyDobot library.

Position.py: Prints the cartesian co-ordinates of the end-effector and angular joint positions

Note : When flashing the esp32, press the Boot button continuously until the flashing is executed. Then press the

The EN button is connected to the EN pin on the ESP32, so pressing it will reset the ESP32.  
  
The BOOT button is connected to GPIO0 (which is also a [bootstrapping pin to set the boot mode](https://github.com/espressif/esptool/wiki/ESP32-Boot-Mode-Selection)), so pressing it will pull GPIO0 low



Protocol : MQTT

Broker : Mosquitto

Publishing Client : Raspberry Pi

Subscribing Client: Cyber Glove