

# 1. Control RGB light

---

The underlying firmware of the robot arm expansion board is developed separately. We provide an interface to allow users to call it. This underlying firmware is responsible for controlling the bus servos, PWM servos, and RGB lights.

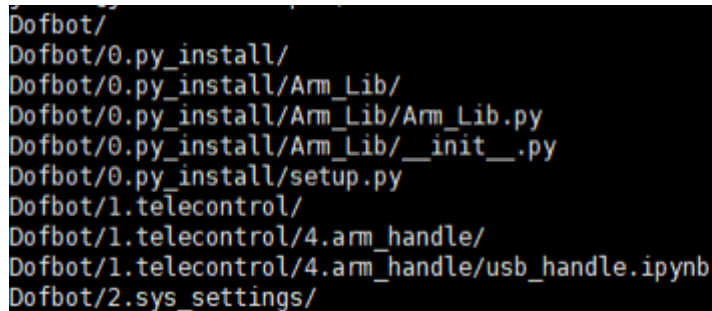
The underlying driver source code has been packaged into a python library, and the underlying driver firmware has been installed in the system image provided by Yahboom.

If you want to transplant it to your own system, you can find the **Dofbot.tar.gz** compressed package in the program source code summary folder, and then remotely transfer it to the Raspberry Pi system through winscp software.

Enter the following command to decompress the firmware package.

```
tar -vxzf Dofbot.tar.gz
```

After successful decompression, the following interface will appear.

A terminal window with a black background and yellow text showing the directory structure of the Dofbot package. The paths listed are: Dofbot/, Dofbot/0.py\_install/, Dofbot/0.py\_install/Arm\_Lib/, Dofbot/0.py\_install/Arm\_Lib/Arm\_Lib.py, Dofbot/0.py\_install/Arm\_Lib/\_\_init\_\_.py, Dofbot/0.py\_install/setup.py, Dofbot/1.telecontrol/, Dofbot/1.telecontrol/4.arm\_handle/, Dofbot/1.telecontrol/4.arm\_handle/usb\_handle.ipynb, and Dofbot/2.sys\_settings/.

```
Dofbot/  
Dofbot/0.py_install/  
Dofbot/0.py_install/Arm_Lib/  
Dofbot/0.py_install/Arm_Lib/Arm_Lib.py  
Dofbot/0.py_install/Arm_Lib/__init__.py  
Dofbot/0.py_install/setup.py  
Dofbot/1.telecontrol/  
Dofbot/1.telecontrol/4.arm_handle/  
Dofbot/1.telecontrol/4.arm_handle/usb_handle.ipynb  
Dofbot/2.sys_settings/
```

Then install it into the system through the following command:

```
cd Dofbot/0.py_install && sudo python3 setup.py install
```

Enter the user password and press Enter to confirm. If you see the installation prompt Arm\_Lib=x.x.x version number, the installation is successful.

```
yahboom@yahboom-virtual-machine: ~/Dofbot/0.py_install
copying build/lib/Arm_Lib/__init__.py -> build/bdist.linux-x86_64/egg/Arm_Lib
byte-compiling build/bdist.linux-x86_64/egg/Arm_Lib/Arm_Lib.py to Arm_Lib.cpython-38.pyc
byte-compiling build/bdist.linux-x86_64/egg/Arm_Lib/__init__.py to __init__.cpython-38.pyc
creating build/bdist.linux-x86_64/egg/EGG-INFO
copying Arm_Lib.egg-info/PKG-INFO -> build/bdist.linux-x86_64/egg/EGG-INFO
copying Arm_Lib.egg-info/SOURCES.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying Arm_Lib.egg-info/dependency_links.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying Arm_Lib.egg-info/top_level.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
zip_safe flag not set; analyzing archive contents...
creating 'dist/Arm_Lib-0.0.5-py3.8.egg' and adding 'build/bdist.linux-x86_64/egg'
to it
removing 'build/bdist.linux-x86_64/egg' (and everything under it)
Processing Arm_Lib-0.0.5-py3.8.egg
Removing /usr/local/lib/python3.8/dist-packages/Arm_Lib-0.0.5-py3.8.egg
Copying Arm_Lib-0.0.5-py3.8.egg to /usr/local/lib/python3.8/dist-packages
Arm-Lib 0.0.5 is already the active version in easy-install.pth

Installed /usr/local/lib/python3.8/dist-packages/Arm_Lib-0.0.5-py3.8.egg
Processing dependencies for Arm-Lib==0.0.5
Finished processing dependencies for Arm-Lib==0.0.5
```

## 1.API Introduction

The corresponding API for RGB lights is: `Arm_RGB_set(R, G, B)`

Function: Set the color of RGB light.

Parameter explanation:

R: Controls the brightness of the red RGB light, ranging from 0-255. The larger the value, the brighter the brightness.

G: Controls the brightness of the green RGB light, ranging from 0-255. The larger the value, the brighter the brightness.

B: Control the brightness of the blue RGB light, ranging from 0-255. The larger the value, the brighter the brightness.

Return value: None.

## 2.Code content

Code path.: `/home/jetson/Dofbot/3.ctrl_Arm/1.rgb.ipynb`

```
Cycle through the RGB lights on the robot arm expansion board to illuminate red,
green, and blue.
#!/usr/bin/env python3
#coding=utf-8
import time
from Arm_Lib import Arm_Device
# Get the object of the robotic arm
Arm = Arm_Device()
time.sleep(.1)
def main():
    while True:
```

```

Arm.Arm_RGB_set(50, 0, 0) #RGB Red light on
time.sleep(.5)
Arm.Arm_RGB_set(0, 50, 0) #RGB Green light on
time.sleep(.5)
Arm.Arm_RGB_set(0, 0, 50) #RGB blue light on
time.sleep(.5)
print(" END OF LINE! ")
try :
    main()
except KeyboardInterrupt:
    # Release the Arm object
    del Arm
    print(" Program closed! ")
    pass

```

Open the 1.rgb.ipynb file from jupyter lab, and click the "Run" entire notebook button on the jupyter lab toolbar. You can see that the RGB light on the robot arm expansion board lights up red, green, and blue lights in a cycle every 0.5 seconds.



If you want to quit, click the Stop button on the toolbar.

