

Programming a Robot Arm

Introduction

This idea was provided by [Bodge N Hackitt](#) at [The MagPi magazine](#) and leans heavily on that work.

Amongst other retailers, Maplin sell a yellow robot arm. At the time of writing it costs £35 (inc. VAT) and that includes the USB interface rather than the manual remote control.

The arm comes with its own software, but I threw that out in favour of programming the robot arm using Python.

The arm has been successfully tested using:

- Windows 7
- Mac OS X 10.8 (Mountain Lion)
- Raspian Wheezy

In other words, it works on Windows, Mac and Raspberry Pis.

The software setup is a bit different for each, and requires a bit of confidence as well as access to the command line. Once the machine is set up you can use the arm WITHOUT the command.



Windows Instructions

You need 2 programs in order to get the arm working

Install libusb-win32

This is the library with instructions to let your computer talk to the robot arm via USB.

Go to the libusb-win32 website (<http://sourceforge.net/projects/libusb-win32/>) and download the latest release (press the big green button). Once you have downloaded the file, right click on it and choose **extract all**.

In the **bin** folder is a program called **inf-wizard.exe**. Make sure the arm is plugged in and switched on. Run the wizard and you will see an **Unknown Device** with vendor ID 0x1267 and product ID 0x0000.

Open **Device Manager** and you should see an unknown USB device with a warning triangle. This is your robot arm. Unplug and reconnect the arm, then right click and follow the options to update the drivers.

Choose a manual search and point the search at the folder where you saved your INF file. The driver should then install and if you look at the driver details you will see libusb-win32 mentioned!

Installing pyusb

This is the library that lets you write USB commands from Python.

Go to the pyusb website (<http://sourceforge.net/projects/pyusb/>) and download the zip file (big green button). Double click this file and it should unpack into a folder.

Open the command prompt (CMD) and navigate to the folder where you downloaded pyusb. Mine is in C:\arm\pyusb-1.0.0a1

```
cd arm\pyusb-1.0.0a1
python setup.py install
```

If that last line doesn't work, find out where Python is installed and try this:

```
c:\Python32\python setup.py install
```



Programming the arm

Phew! Well done for making it this far!

Go to The MagPi article (<http://www.themagpi.com/issue/issue-3/article/skutter-write-a-program-for-usb-device/>) and copy the longish-Python program.

Paste this into IDLE or in any text editor.

Either run the program as normal (Save and press F5) or at the command line type in:

```
python arm.py
```

You should see the robot are moving around.

You only need to edit the code at the bottom of the program, the lines that look like this:

```
MoveArm(1, [0, 1, 0])
```

The first number is the duration of the move, in seconds.

The codes for moving the arm work as follows:

Code	Result
[1, 0, 0] or [2, 0, 0]	Open or close the jaws
[4, 0, 0] or [8, 0, 0]	Raise or lower from the wrist
[16, 0, 0] or [32, 0, 0]	Raise or lower from the elbow
[64, 0, 0] or [128, 0, 0]	Raise or lower from the shoulder
[0, 1, 0] or [0, 2, 0]	Rotate clockwise or anti-clockwise
[0, 0, 1]	Turn the light on

You can combine these, so

[1, 1, 1] will open the jaws, rotate clockwise and turn the light on at the same time.

[5, 0, 0] will open the jaws and raise from the wrist at the same time.

You may find that some of the codes are the wrong way round - this usually means you have wired up or fitted the motors back to front!



Linux / Raspberry Pi Instructions

You need 2 programs in order to get the arm working

Install libusb

If you are using a Raspberry Pi then libusb is probably already installed.

If you think you need to install libusb then try the following:

```
sudo apt-cache search libusb
```

This will help you find the correct libusb package to install. It will be something like libusb-1.0.18.

```
sudo apt-get install libusb-1.0.18
```

This will install libusb.

Installing pyusb

At the moment there isn't a pyusb package you can install using apt.

In a webbrowser, go to the PyUSB website (<http://sourceforge.net/projects/pyusb/>) and download the tar file (big green button). Double click this file and it should unpack into a folder.

At the terminal, go to the folder where you unpacked the pyusb files.

```
cd ~/pyusb-1.0.0b1
```

Configure and install pyusb

```
python setup.py install
```



Programming the arm

Phew! Well done for making it this far!

Go to The MagPi article (<http://www.themagpi.com/issue/issue-3/article/skutter-write-a-program-for-usb-device/>) and copy the longish-Python program.

Paste this into IDLE or in any text editor. I quite like to use **nano** at the Terminal, but I'm a self-confessed nerd.

Either run the program as normal (Save and press F5) or at the Terminal type in:

```
python arm.py
```

You should see the robot are moving around.

You only need to edit the code at the bottom of the program, the lines that look like this:

```
MoveArm(1, [0, 1, 0])
```

The first number is the duration of the move, in seconds.

The codes for moving the arm work as follows:

Code	Result
[1, 0, 0] or [2, 0, 0]	Open or close the jaws
[4, 0, 0] or [8, 0, 0]	Raise or lower from the wrist
[16, 0, 0] or [32, 0, 0]	Raise or lower from the elbow
[64, 0, 0] or [128, 0, 0]	Raise or lower from the shoulder
[0, 1, 0] or [0, 2, 0]	Rotate clockwise or anti-clockwise
[0, 0, 1]	Turn the light on

You can combine these, so

[1, 1, 1] will open the jaws, rotate clockwise and turn the light on at the same time.

[5, 0, 0] will open the jaws and raise from the wrist at the same time.

You may find that some of the codes are the wrong way round - this usually means you have wired up or fitted the motors back to front!



Mac Instructions

You need 3 programs in order to get the arm working.

Installing libusb

This is the library with instructions to let your computer talk to the robot arm via USB.

Go to the libusb website (<http://www.libusb.org/>) and download the **tarball** for version 1.0. This is a file that ends in .tar.bz2. Double click this file and it should unpack into a folder.

Open the **terminal** and type in the following commands:

Move yourself into the folder where you unpacked the libusb files.

```
cd ~/Downloads/libusb-1.0.18
```

Configure and install libusb

```
./configure  
make  
sudo make install
```

Installing pyusb

This is the library that lets you write USB commands from Python.

Go to the pyusb website (<http://sourceforge.net/projects/pyusb/>) and download the zip file (big green button). Double click this file and it should unpack into a folder.

Open the **terminal** and type in the following commands:

Move yourself into the folder where you unpacked the pyusb files.

```
cd ~/Downloads/pyusb-1.0.0b1
```

Configure and install pyusb

```
python setup.py install
```

Downloading usb_arm.py

Go to the usb_arm website (https://github.com/dannystaple/robot_arm), then right click and download the usb_arm.py file. This is pre-programmed with all the complicated bits for talking directly to the arm. You can install this, or simply give each student a copy. As long as it sits in the same folder as their own Python program they'll be fine.



Programming the arm

Phew! Well done for making it this far!

Now, write the following program out. This can be done in IDLE or in any text editor. I quite like to use **nano** at the Terminal, but I'm a self-confessed nerd.

```
import usb_arm

arm = usb_arm.Arm()
arm.move([0,0,1])
```

Either run the program as normal (Save and press F5) or in the Terminal type in:

```
python arm.py
```

You should see the light on the end of the arm flash on for a second and then off.

The codes for moving the arm work as follows:

Code	Result
[1,0,0] or [2,0,0]	Open or close the jaws
[4,0,0] or [8,0,0]	Raise or lower from the wrist
[16,0,0] or [32,0,0]	Raise or lower from the elbow
[64,0,0] or [128,0,0]	Raise or lower from the shoulder
[0,1,0] or [0,2,0]	Rotate clockwise or anti-clockwise
[0,0,1]	Turn the light on

You can combine these, so

[1,1,1] will open the jaws, rotate clockwise and turn the light on at the same time.
[5,0,0] will open the jaws and raise from the wrist at the same time.

You may find that some of the codes are the wrong way round - this usually means you have wired up or fitted the motors back to front!

