

Programming Picoh in Python



Python is a programming language with an emphasis on easy to read code. It is simple enough for beginners to get started with but powerful enough to enable complex projects. It runs on different types of computers and is widely supported on different operating systems including Windows, macOS and Raspbian.

The language is enhanced by a wide range of modules written by different programmers that help complete a range of tasks. There are modules for analysing, manipulating and generating graphics, sound and text, interacting with machines, the internet, and artificial intelligence and many other digital processes.

When Python is installed it is packaged with a code editor called IDLE which appears as an application on your computer. IDLE can be used to write, edit and run Python programs.

The image shows two windows from a Python environment. The left window is titled "Python 3.6.4 Shell" and displays the Python startup banner: "Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)] on win32 Type "copyright", "credits" or "license()" for more information. >>>". The right window is titled "Untitled*" and shows a code editor with the following Python code: "from picoh import picoh" and "picoh.reset()".

IDLE has two windows, on the left is the shell, this is where programs run and display any print information or requests for input.

On the right is the script area, this is where Python programs are written.

If you don't see the right-hand script select New File from the File menu.

Python programs are written as text and need to be saved with the .py file extension. Programs run from the first line, line by line to the end and then stop although it is possible to loop part or all of a program.

The Picoh library provides access to the full range of Picoh features in a simple to use package.

To install the Picoh package please follow the setup guide for your platform:

[Mac](#)

[Windows](#)

[Pi](#)

Once installed, to make use of the Picoh library in a program it needs to be imported using an import statement at the beginning of your code.

Import Statement

<pre>from picoh import picoh</pre>

The import statement comes at the start of a program.

To reset Picoh use the `picoh.reset()` command. Putting the import and reset commands together at the start of a program will connect to picoh and reset it to the rest position leaving it ready to be programmed.

Open IDLE, click File -> New File and type or paste in the following:

```
from picoh import picoh
picoh.reset()
```

Next click Run -> Run Module to run the script, you will be prompted to save your file and can save it wherever you like on your computer but please give

it a name other than picoh.py as that would cause an error. It is a good idea to create a folder called something like PicohPrograms to keep things organised.

Once saved the program should run, unless the motors have been moved from their home position Picoh won't move quite yet, if they have been moved Picoh's motors should all move back to their home position. As you go through the guide, click run again each time you change your program to see the results on Picoh.

Move Picoh

To make Picoh move use the following command:

```
picoh.move(picoh.EYETURN, 10)
```

The picoh.move function has two parameters; these are the bits of information the function needs to perform its job. Enter values for each parameter inside brackets after the function. First the function needs the name of the motor to move and second the position to move it to. The two parameters are separated by a comma.

The following movements can be achieved:

Motor No	Movement Name	Command (Replace 10 with any number 0-10)
0	Head Nod	picoh.move(picoh.HEADNOD, 10)
1	Head Turn	picoh.move(picoh.HEADTURN, 10)
2	Eye Turn	picoh.move(picoh.EYETURN, 10)
3	Lid Blink	picoh.move(picoh.LIDBLINK, 10)
5	Bottom Lip	picoh.move(picoh.BOTTOMLIP, 10)

6	Eye Tilt	<code>picoh.move(picoh.EYETILT, 10)</code>
---	----------	--

Add the following code to your program with your import and reset code:

```
picoh.move(picoh.HEADTURN,0)

picoh.wait(2)

picoh.move(picoh.HEADTURN,10)

picoh.wait(0.5)

picoh.move(picoh.HEADTURN,5)
```

This program also includes the `picoh.wait()` command. This command makes the program wait for the specified number of seconds before moving to the next line. In this example the wait gives the motor time to move before the next move command is sent.

Try expanding on the example above and including a longer sequence of movements and waits.

Remember to separate move commands for the same motor with wait commands to give them time to move.

Can you make Picoh Smile?
Can you make Picoh Frown?
Can you make Picoh Blink?



Make Picoh speak

To enable Picoh to speak the library includes a function, `picoh.say()`. For simple use the `say` function only needs one parameter, the text you want Picoh to say, enclosed in `""`.

```
picoh.say("Hello my name is Picoh")
```

The function takes the text and converts it into an audio file using a text to speech synthesizer. It then generates a file containing a series of times and lip positions in order to be able to move the lips in time with the speech. The command will also prevent the program from moving to the next line until the speech has concluded. This all happens in the background, a single `picoh.say()` function is enough to move the lips, generate and play the speech audio and wait long enough for the speech to finish.

```
from picoh import picoh

picoh.reset()

picoh.wait(1)

picoh.say("Hello my name is Picoh, good to meet you")

picoh.wait(1)

picoh.say("Goodbye")
```

```
picoh.move(picoh.HEADNOD,3)

picoh.move(picoh.LIDBLINK,1)

picoh.wait(1)

picoh.close()
```

This program also includes the `picoh.close()` function which simply powers down Picoh's motors.

The `picoh.say()` command has two more optional parameters.

The `say` function has a parameter called `untilDone`, which controls whether the program waits while the speech is spoken. Enter `False` for the second parameter in the `say` command to prevent the program from waiting while the speech is being spoken:

```
from picoh import picoh

picoh.reset()
picoh.wait(1)
picoh.say("Don't finish speaking before I move my eyes",False)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,5)

picoh.wait(3)

picoh.say("Wait until i've finished speaking before moving my eyes",True)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
```

```
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,5)
```

The say command has one more parameter you can use. The third parameter turns the lip sync on or off. These say commands have all three parameters one after another, separated by commas:

```
from picoh import picoh

picoh.reset()
picoh.say("Speak without moving my lips, then move my eyes", True, False)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,5)
picoh.wait(1)

picoh.say("Speak with my lips moving, then move my eyes.", True, True)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,5)
picoh.wait(1)

picoh.say("Speak without my lips moving, and move my eyes at the same time", False,
False)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,5)
picoh.wait(4)

picoh.say("Speak with my lips moving, and move my eyes at the same time", False,
True)
picoh.move(picoh.EYETURN,10)
picoh.wait(0.6)
picoh.move(picoh.EYETURN,0)
picoh.wait(0.6)
```

```
picoh.move(picoh.EYETURN,5)
picoh.wait(4)

picoh.close()
```

Loops

To make a program repeat indefinitely a simple option is a while loop.

```
from picoh import picoh
picoh.reset()
picoh.wait(1)

while True:
    picoh.move(picoh.HEADNOD,10)
    picoh.wait(2)
    picoh.move(picoh.HEADNOD,0)
    picoh.wait(2)
```

Notice that the lines inside the loop need to be indented, four spaces in from the left. Many programming languages use brackets to denote blocks within lines of code, Python uses indentations instead.

To loop something a certain number of times you can use a for loop.

The `for x in range()` code will run the program once for each value in the range so `range(0,5)` will run 5 times. The first time `x` will equal 0, then 1 and so on up to 4, the last whole number within the specified range.

Again the section inside the loop is indented one tab from the left.

```
from picoh import picoh
picoh.reset()
picoh.wait(1)

for x in range(0,5):
    picoh.say("I will say this 5 times")

for x in range(0,2):
    picoh.say("I will say this 2 times")
```



```
picoh.close()
```

The variable `x` can also be used inside the loop, for example to move through all of a motor's positions:

```
from picoh import picoh
picoh.reset()
picoh.wait(1)

for x in range(0,10):
    picoh.move(picoh.HEADTURN,x)
    picoh.wait(1)
```

Try changing the values in the `range(,)` command and see the effect this has on the movements.

Please note, only numbers 0-10 will work as this is the full range of the motor.



Now you know the basics of making Picoh move, wait and speak, check out the [Examples](#) folder and try running and modifying some of the programs.

If you have any questions or queries about controlling PicoH from Python, want to share a project with us or want to suggest improvements to the library please feel free to get in contact on info@ohbot.co.uk.