

INSTRUCTION MANUAL TO MAKE A FISHBOT SMART WATER METER

- Setting up the Camera
- Adding software to be able to convert images to text
- Code to take regular photos
- Code to convert photos into text
- Code to read out text

Setting up the Camera

You will need a Raspberry Pi and a PiCamera.

A PiCamera Module costs £24

Raspberry Pi Camera Module V2

Delivery from just £2.50	49,000+ customer reviews	No Quibble returns policy	Secure payment system	Next Day Delivery Order by 2PM*
-----------------------------	-----------------------------	------------------------------	--------------------------	------------------------------------



In Stock

£24.00 GBP [Add to Cart](#)

★★★★★ 189 Reviews

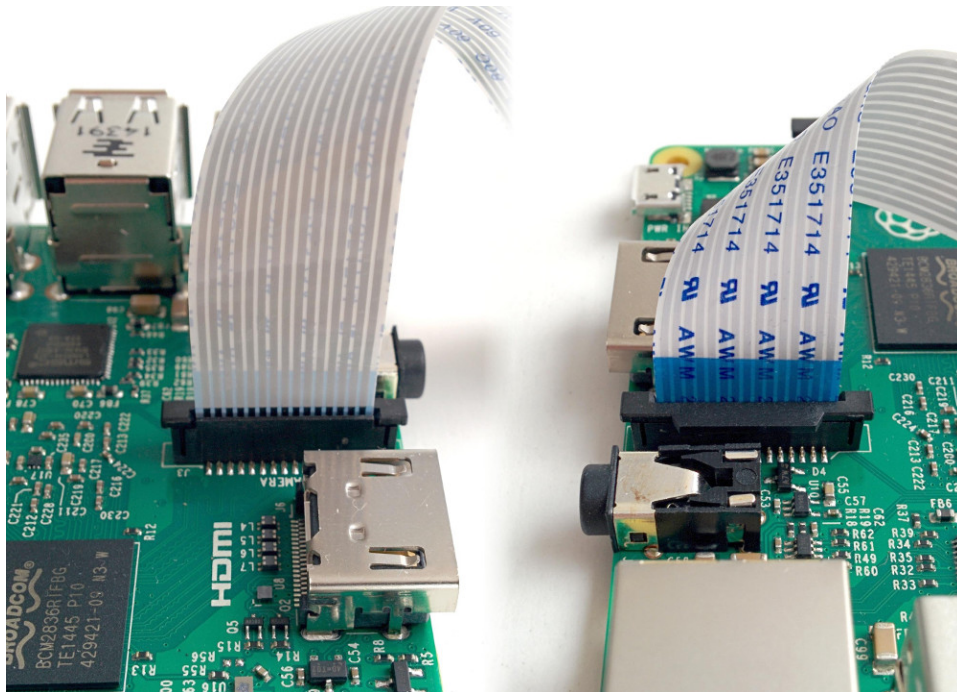
3 Questions \ 4 Answers

Share this Product

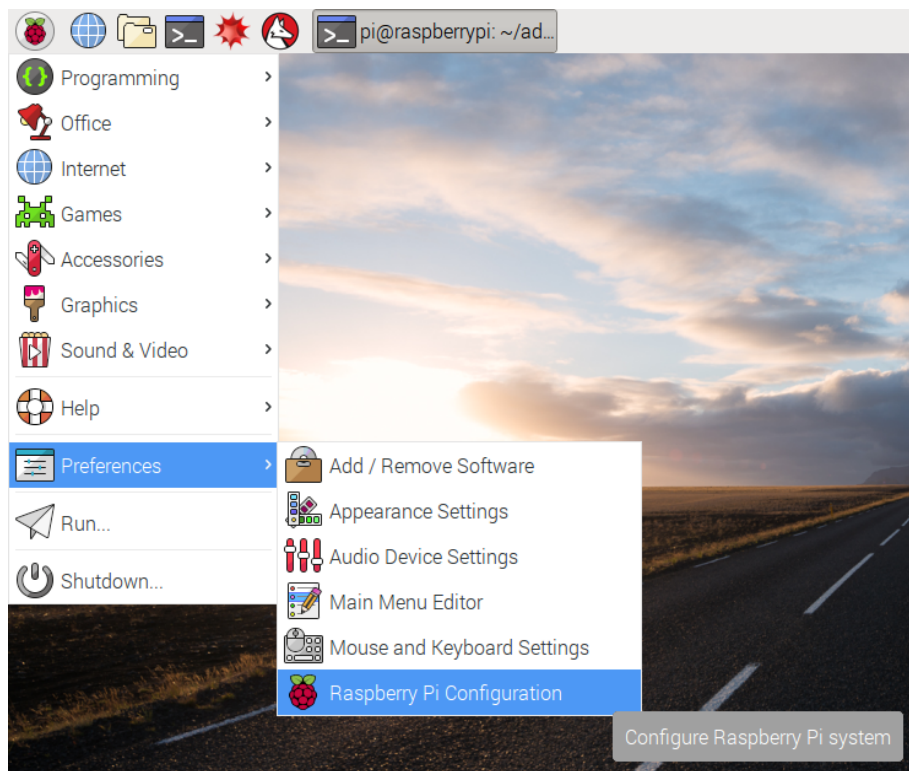


First of all, with the Pi switched on you'll need to connect the Camera Module to the Raspberry Pi's camera port, then start up the Pi and ensure the software is enabled.

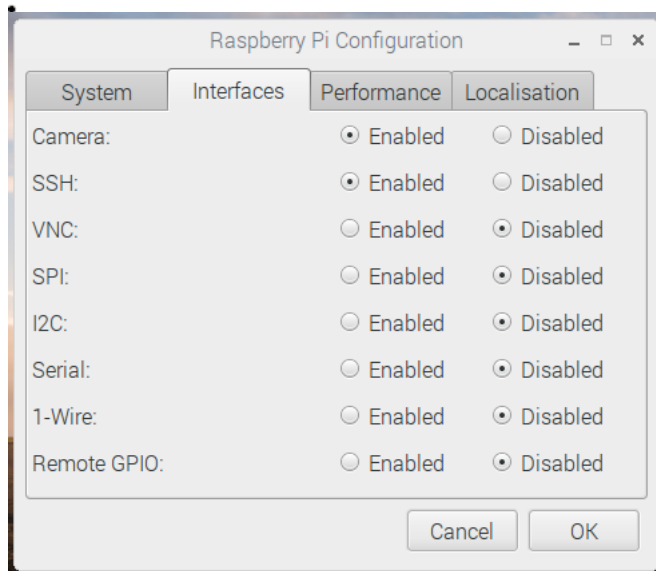
Locate the camera port and connect the camera as shown (it's inbetween the Ethernet and HDMI ports):



Start up the Pi. Open the **Raspberry Pi Configuration Tool** from the main menu:



Ensure the camera software is enabled:



If it's not enabled, enable it and reboot your Pi to begin.

Or if you want to use the Terminal to enable your camera

First make sure your Raspberry Pi is up to date

Type:

```
sudo apt-get update
```

then

```
sudo apt-get upgrade
```

then run

```
sudo-raspi-config
```

open interfacing options, navigate to camera option and enable it. Select **Finish** and reboot the Raspberry Pi.

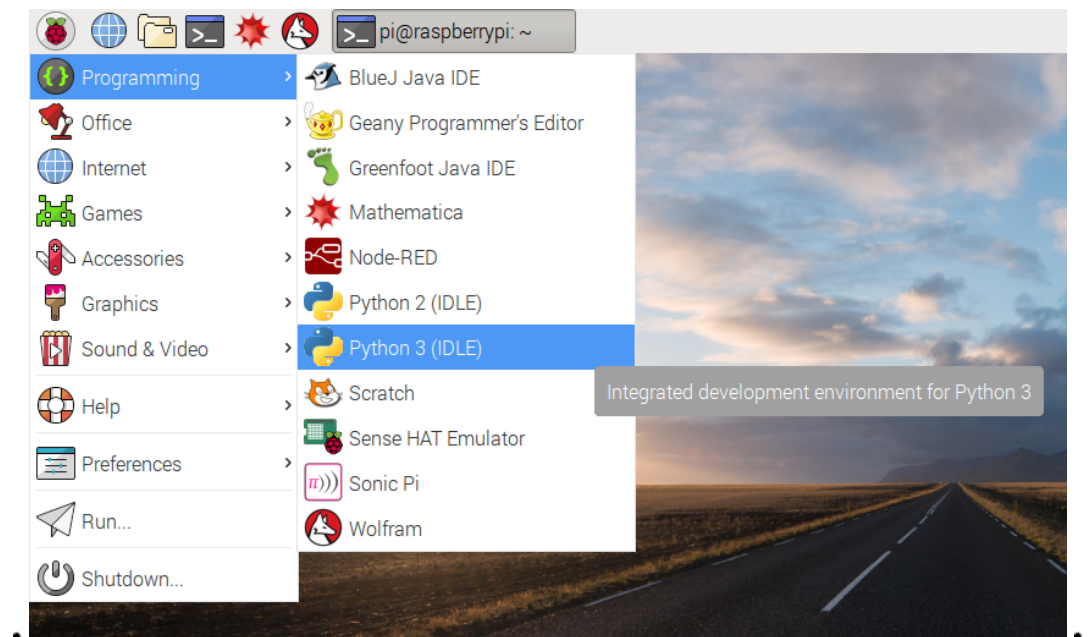
If you want to test the Camera works

Now your camera is connected and the software is enabled, you can get started by trying out the camera preview.

Open **Python 3** from the main menu:

Open a new file and save it as `camera.py`. It's important that you **do not** save it as `picamera.py`. Enter the following code:

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview()
sleep(10)
camera.stop_preview()
```



Save with **Ctrl + S** and run with **F5**. The camera preview should be shown for 10 seconds, and then close. Move the camera around to preview what the camera sees.

Adding Software to be able to convert text in images to text and read out

In terminal type the following code:

```
sudo apt-get install tesseract-ocr
```

```
sudo pip3 install pillow pytesseract
```

```
sudo apt-get install espeak
```

```
sudo apt-get install espeak python-espeak
```

Smart Meter code

1) First add code to see what the camera is doing and take a photo:

```
import picamera
from time import sleep

#Take still photos
camera = picamera.PiCamera()
camera.start_preview()
camera.capture('image.jpg')
sleep(50)
```

2) This part opens the image and then converts the text in the photo to text

```
from PIL import Image
import pytesseract

#open image to read text in image and convert
to text
im = Image.open("image.jpg")
#convert image to string
text =
pytesseract.image_to_string(im, lang='eng').st
rip()
#display number read
print(text)
```

2) This part speaks the text

```
import os
os.system('espeak "Save Water Reading
says{}"'.format(text))
sleep(2)
```

This is all the Python code put together:

```
import picamera
from time import sleep
from PIL import Image
import pytesseract
import os

#start video to see what photo is being taken
camera= picamera.PiCamera()
camera.resolution = (1024, 768)
camera.start_preview(fullscreen=False,
window=(120,120,256,192))
sleep(2)
#Take photo
while True:
    camera.capture('image.jpg')
    print ("Image taken")
    sleep(2)
    #open image to read text in image and convert
    to text
    im = Image.open("image.jpg")
    #convert image to string
    text =
    pytesseract.image_to_string(im,lang='eng').st
    rip()
    #display number read
    print(text)
    #speak number read
    os.system('espeak "Save Water Reading
    says{}"'.format(text))
    sleep(2)
#stop video if there is a problem
else:
    camera.stop_preview()
```