jonathanschemoul [10:04 AM]

dmesg on the console to know the tty

[10:04]

on linux it will be ttyUSB0

[10:04]

pip install esptool adafruit-ampy

[10:05]

download http://micropython.org/resources/firmware/esp8266-20170612-v1.9.1.bin

[10:05]

esptool.py --baud 460800 write\_flash --flash\_size=detect --flash\_mode dio 0 esp8266-20170612-v1.9.1.bin

.. erase ..

python C:\Users\Ran\_the\_User\Anaconda2\Lib\site-packages\esptool.py --baud 460800 --port COM8 write\_flash --flash\_size=detect  --flash\_mode dio 0 esp8266-20170612-v1.9.1.bin

later connect with 115200

jonathanschemoul [10:39 AM]

>>>import network

>>> wlan = network.WLAN(network.STA\_IF)

>>> wlan.active(True)

#5 ets\_task(4020ed88, 28, 3fff9ea8, 10)

>>> wlan.con

connect config

>>> wlan.connect('Macs-Labs', 'macs2015')

>>> wlan.ifconfig()

('0.0.0.0', '0.0.0.0', '0.0.0.0', '208.67.222.222')

>>> wlan.ifconfig()

('172.19.3.238', '255.255.255.0', '172.19.3.254', '132.70.60.124')

[10:42]

import webrepl\_setup

[10:43]

https://github.com/micropython/webrepl/archive/master.zip

elazarl [10:45 AM]

joined #general

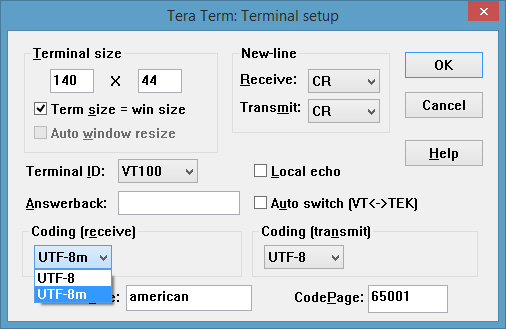
jonathanschemoul [11:10 AM]

https://iotbytes.wordpress.com/nodemcu-pinout/

IoT Bytes Pradeep Singh

NodeMCU Pinout

This page provides details for NodeMCU ESP12 Developement Kit V1.0 Pins



<http://docs.micropython.org/en/latest/pyboard/reference/speed_python.html?highlight=map>

<https://github.com/micropython/micropython/tree/master/esp8266>

<https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>

<https://docs.micropython.org/en/latest/esp8266/esp8266/quickref.html>

<https://www.adafruit.com/product/3316>

pc : <https://pypi.python.org/pypi/paho-mqtt/1.1>

<https://pypi.python.org/pypi/paho-mqtt/1.2.3>

pc: >pip install paho-mqtt

<http://mqtt.org/>

from umqtt import MQTTClient

from umqtt.simple import MQTTClient

alonisser [9:52 AM]

Docs:

http://docs.micropython.org/en/latest/esp8266/esp8266/quickref.html

yotam [9:53 AM]

Firmware: http://micropython.org/resources/firmware/esp8266-20170612-v1.9.1.bin

idanaroz [9:53 AM]

joined #general

yotam [9:53 AM]

Installation instructions (flushing the drive etc.): https://docs.micropython.org/en/latest/esp8266/esp8266/tutorial/intro.html#intro

jonathanschemoul [9:58 AM]

hi

randv [9:59 AM]

joined #general

jonathanschemoul [10:04 AM]

dmesg on the console to know the tty

[10:04]

on linux it will be ttyUSB0

[10:04]

pip install esptool adafruit-ampy

[10:05]

download http://micropython.org/resources/firmware/esp8266-20170612-v1.9.1.bin

[10:05]

esptool.py --baud 460800 write\_flash --flash\_size=detect --flash\_mode dio 0 esp8266-20170612-v1.9.1.bin

eylonmalin [10:09 AM]

you may need to :

```sudo chmod 777 /dev/ttyUSB0```

[10:09]

before running the esptool.py

jonathanschemoul [10:11 AM]

screen /dev/ttyUSB0 115200

yaron [10:35 AM]

joined #general

jonathanschemoul [10:39 AM]

>>>import network

>>> wlan = network.WLAN(network.STA\_IF)

>>> wlan.active(True)

#5 ets\_task(4020ed88, 28, 3fff9ea8, 10)

>>> wlan.con

connect config

>>> wlan.connect('Macs-Labs', 'macs2015')

>>> wlan.ifconfig()

('0.0.0.0', '0.0.0.0', '0.0.0.0', '208.67.222.222')

>>> wlan.ifconfig()

('172.19.3.238', '255.255.255.0', '172.19.3.254', '132.70.60.124')

[10:42]

import webrepl\_setup

[10:43]

https://github.com/micropython/webrepl/archive/master.zip

elazarl [10:45 AM]

joined #general

jonathanschemoul [11:10 AM]

https://iotbytes.wordpress.com/nodemcu-pinout/

IoT Bytes Pradeep Singh

NodeMCU Pinout

This page provides details for NodeMCU ESP12 Developement Kit V1.0 Pins

elazarl [11:22 AM]

Mac users, looks like you need to install this driver from here:

http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Driver: http://www.silabs.com/documents/public/software/Mac\_OSX\_VCP\_Driver.zip

silabs.com

USB to UART Bridge VCP Drivers | Silicon Labs

The CP210x USB to UART Bridge Virtual COM Port (VCP) drivers are required for device operation as a Virtual COM Port to facilitate host communication with CP210x products. These devices can also interface to a host using the direct access driver.

[11:22]

Then /dev/tty.SLAB\_USBtoUART would appear

yotam [11:23 AM]

@elazarl did that work for you? Were you able to screen to the terminal?

jonathanschemoul [11:31 AM]

pip install adafruit-ampy

[11:31]

ampy -p /dev/ttyUSB0 put boot.py (edited)

yotam [11:35 AM]

https://learn.adafruit.com/micropython-basics-load-files-and-run-code/boot-scripts

learn.adafruit.com

MicroPython Basics: Load Files & Run Code

Learn a simple workflow for developing code on a MicroPython board.

elazarl [11:42 AM]

uploaded this image: @yotam Yes:

Add Comment

elazarl [11:42 AM]

command line ~ ❯❯❯ screen /dev/tty.SLAB\_USBtoUART 115200

jonathanschemoul [12:46 PM]

https://www.hackster.io/bucknalla/mqtt-micropython-044e77

Hackster.io

MQTT & MicroPython

Get started subscribing and publishing messages, in MicroPython, using MQTT! By Alex Bucknall. (20kB)

randv [1:39 PM]

for PC: https://pypi.python.org/pypi/paho-mqtt/1.1

pypi.python.org

paho-mqtt 1.1 : Python Package Index

MQTT version 3.1/3.1.1 client class

[1:39]

sorry: newer one : https://pypi.python.org/pypi/paho-mqtt/1.2.3

pypi.python.org

paho-mqtt 1.2.3 : Python Package Index

MQTT version 3.1.1 client class

alonisser [1:50 PM]

Analog to Digital pins http://wiki.jackslab.org/ESP32\_ADC

randv [3:48 PM]

added and commented on this Python snippet: PC\_mqttExample\_Client

import paho.mqtt.client as mqtt

print dir (mqtt)

​

import time

from datetime import datetime

from math import radians

from pytz import timezone

​

newReceivedMQTTdata=-1

​

# The callback for when the client receives a CONNACK response from the server.

def on\_connect(client, userdata, flags, rc):

print("Connected with result code "+str(rc))

​

# Subscribing in on\_connect() means that if we lose the connection and

# reconnect then subscriptions will be renewed.

# client.subscribe("$SYS/#")

client.subscribe("myNewRan")

​

# The callback for when a PUBLISH message is received from the server.

def on\_message(client, userdata, msg):

global newReceivedMQTTdata

if \_\_name\_\_ == '\_\_main\_\_':

print(msg.topic+" "+str(msg.payload))

newReceivedMQTTdata=(datetime.now(tz=timezone('Europe/Moscow')).minute+datetime.now(tz=timezone('Europe/Moscow')).second/60.+datetime.now(tz=timezone('Europe/Moscow')).microsecond/1000000./60.,int(msg.payload))

# if \_\_name\_\_ == '\_\_main\_\_':

# print newReceivedMQTTdata

​

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

​

client.connect("iot.eclipse.org", 1883, 60)

​

# Blocking call that processes network traffic, dispatches callbacks and

# handles reconnecting.

# Other loop\*() functions are available that give a threaded interface and a

# manual interface.

​

# client.loop\_forever()

print("start listening to ..myNewRan..")

client.loop\_start()

​

if \_\_name\_\_=='\_\_main\_\_':

print("will wait for 10.5 sec ")

time.sleep(10.5)

client.loop\_stop()

print("finished listening")

1 Comment Click to expand inline 48 lines

example for using MQTT to recieve simulated sensor data. and to plot them as stream in HTML page

randv [3:48 PM]

added this Python snippet: stream\_from\_broker.py

#import libraries

from bokeh.io import curdoc

from bokeh.models import ColumnDataSource

from bokeh.plotting import figure

# from random import randrange

import PC\_mqttExample\_ClientRecieving as myMQ

​

rollNum = 15

# axLimit = 11

​

#create figure

# f=figure(x\_range=(0,axLimit+1),y\_range=(0,axLimit+1))

f=figure()

​

#create columndatasource

source=ColumnDataSource(data=dict(x=[],y=[],size=[]))

​

#create glyphs

f.circle(x='x',y='y',size='size',fill\_color='olive',line\_color='yellow',source=source)

f.line(x='x',y='y',line\_color='green',source=source)

​

#create periodic function

def update():

# print source.data['size'] # returns a list type

# new\_data=dict(x=[randrange(1,axLimit)],y=[randrange(1,axLimit)],size=[rollNum+1] )

newBrokerData = myMQ.newReceivedMQTTdata # recieve data and new index

new\_data = dict(x=[newBrokerData[0]], y=[newBrokerData[1]], size=[rollNum + 1])

source.stream(new\_data,rollover=rollNum)

source.data['size'] = [ x-1 for x in source.data['size'] ]

# print "streamimng :"

# print newBrokerData

#print(source.data)

​

#add figure to curdoc and configure callback

curdoc().add\_root(f)

curdoc().add\_periodic\_callback(update,100) # freq of mSec

Add Comment Click to expand inline 37 lines

randv [3:48 PM]

added this Python snippet: PC\_mqttExample\_ClientBroadcasting.py

import paho.mqtt.client as mqtt

print dir (mqtt)

​

import time

​

####################################

mqttc = mqtt.Client()

mqttc.connect("iot.eclipse.org")

mqttc.loop\_start()

i=0

​

while True:

i=i+1

temperature = i#sensor.blocking\_read()

# res=mqttc.publish("paho/temperature", temperature)

res=mqttc.publish("myNewRan", temperature)

if res[0]!=0:

print"possible publish error"

# print i, res

time.sleep(0.5)

​

##########################################

from umqtt.simple import MQTTClient

Add Comment Click to expand inline 24 lines

randv [3:48 PM]

added this Plain Text snippet: luanchStreamPlotting.bat

bokeh serve stream\_from\_broker.py -

Add Comment

bityon [3:51 PM]

Our code is here: https://github.com/dan-gittik/micropython

GitHub

dan-gittik/micropython

Contribute to micropython development by creating an account on GitHub.

<http://docs.micropython.org/en/latest/esp8266/esp8266/tutorial/intro.html#intro>

pip install paho-mqtt