/mymqtt/info.txt

---

while with the project
/myblog/
blog:
http://kll.engineering-news.org/kllfusion01/articles.php?article_id=125
http://kll.engineering-news.org/kllfusion01/articles.php?article_id=149
download:
http://kll.engineering-news.org/kllfusion01/downloads.php?
cat_id=5&download_id=47

there is already a
python webserver with mqtt database collection to SQLite database file
and web page to show a featured graph from it
and even a (admin login protected) mqtt operation
available,

---

here i wanted to pack a snappy tool set for the
* Raspberry Pi
* Python
* MQTT
beginner/ but without the webserver and all the overhead,
still things like some code and settings and path structure are compatible
between the 2 projects
download:
http://kll.engineering-news.org/kllfusion01/articles.php?article_id=149
(at the end)

```
pi@raspberrypi:~/Projects/mymqtt $ tree
.
├── info.txt   < this file >
├── install.sh
├── mqtt_from_database.py
├── mqtt_remote.py
├── mqtt_settings.py
├── mqtt_sim.py
├── mqtt_to_database.py
├── mymqttstart.desktop
├── start_mqttdb_service
├── static
│   └── data
│       └── myMQTT.db
└── tools
    ├── mqttstart.desktop.template
    ├── sketch_book
    │   ├── Arduino_ESP8266_OLED
    │   │   └── ESP8266_MQTT
    │   │       ├── DHT22_disabled.ino
    │   │       ├── ESP8266_MQTT.ino
    │   │       ├── images.h
    │   │       ├── OLED.ino
    │   │       └── Signal.ino
    │   ├── arduinoinst.sh
    │   └── Processing353
    │       ├── DataBase_reader
    │       │   └── DataBase_reader.pde
    │       └── MQTT_monitor
    │           └── MQTT_monitor.pde
    └── start_mqttdb_service.template

9 directories, 20 files
pi@raspberrypi:~/Projects/mymqtt $
```

---

let's assume you have a RPI running RASPBIAN
+ + here use :
DEBIAN stretch
Linux version 4.14.98-v7+
Raspberry Pi 3 Model B Rev 1.2

updated but mostly original
and you want begin with

* python and
* MQTT and
* SQLite database and
* Processing IDE / JAVA mode

this toolset might give a easy start.

-a-
after download unzip this, let,s say to
/home/pi/Projects/mymqtt

-b-
environmental workbench
ever heard of .bash_aliases
i love to tweak my terminal LINUX..
under tool find a example you can copy up to your home like
/home/pi/.bash_aliases
when you now open a terminal, this is executed, shows you add info and
informs you what add commands are created: (here we concentrate on the MQTT
related only)

```
# MQTT
alias MQTTsys='mosquitto_sub -v -t \$SYS/#'
alias MQTTshow='mosquitto_sub -h 127.0.0.1 -t "#" -v'
alias MQTT_LED_ON='mosquitto_pub -h 127.0.0.1 -t "pcu213/in" -m "1"'
alias MQTT_LED_OFF='mosquitto_pub -h 127.0.0.1 -t "pcu213/in" -m "0"'
alias MQTT_SIM='mosquitto_pub -h 127.0.0.1 -t "pcu213/sensor/out" -m
"{\"val0\":10.10,\"val1\":11.11,\"val2\":12.12}"'
#
```

now you can with a short
MQTT_SIM
send a test record to MQTT (if installed) and
that will be stored in database (if the python service is started)

or a
MQTTshow
prints on terminal all mqtt traffic

and many more goodies like
screen shot
* screen   snapf
* active window snap (with timer)
* region   snaps (mouse select region
all this save to
/home/pi/Pictures/

-c-
./install.sh

on install you get first a installation of just a ICON on desktop
file: mymqttstart.desktop
name: MQTTdb servicestart
execute: /home/pi/Projects/mymqtt/start_mqttdb_service

next a series of questions
-1- pyformata   (here not needed)
-2- MQTT server (the broker if needed on RPI running)
-3- paho-mqtt   (for python tools)
-4- Arduino IDE (if want program some MQTT clients like ESP or arduino
ethernet)
-5- Processing IDE (no auto install!)
sorry, could only print the curl command line (pipes to sudo),
you need to
Copy Paste [Enter]
manually for install.

(possibly better do a reboot)

-d-

when you start the
* mymqtt serverstart
from icon on desktop
a terminal window opens,
first a empty database file is created
/mymqtt/static/data/myMQTT.db
and you keep the window open // or shut service down by [crtl][c]
here see also (very detailed as now DiagP = True)
mqtt records seen and saved

(yes, on long term that part should be a service
you need to start at boot right after the MQTT server)

now can test from terminal like with
MQTT_SIM
or

-e-
./mqtt_sim.py
now that is a example for python to do same,
but send timed 20 records if sinus waves

-f-
./mqtt_remote.py 0   # or 1 ...
from terminal (or like the myblog app does from python called)
can start a command to a mqtt client device what listens to this topic
(that was originally used for a ESP 8266 wifi board to switch its LED remotely)

-g-
./mqtt_from_database.py
a readout of the database records to terminal

-h- all above use the defaults you can adjust in
mqtt_settings.py :

```
"""filename: mqtt_settings.py
"""
mqtt_broker="127.0.0.1"
mqtt_out="pcu213/sensor/out"
mqtt_in="pcu213/in"

# for DB see /static/data/
MQDATABASE = 'myMQTT.db'
DB_exists = False
TABLE_NAME = mqtt_out.replace("/","_")          # SQL not take the mqtt topic
structure
TB_exists = False
```

PROCESSING
if you install Processing IDE (now 3.5.3)
and setup up in preferences a sketch_book path / you might use (or copy them)
/home/pi/Arduino/Processing353/
you have 2 short examples

/DataBase_reader
use: BezierSQLib 0.3.1
import de.bezier.data.sql.*;

/MQTT_monitor
use: MQTT 1.7.1 Joel Gaehwiler
import mqtt.*;
