# MCU to MySQL

## 1. 建立MQTT Broker

```
# login UBUNTU server
$ sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
$ sudo apt-get update
# 安裝 mosquitto port 1883
$ sudo apt-get install mosquitto
$ sudo apt-get install mosquitto-clients
$ sudo apt-get install mc
# mosquitto安裝到 service auto start
$ sudo systemctl start mosquitto.service
$ sudo systemctl enable mosquitto.service
```

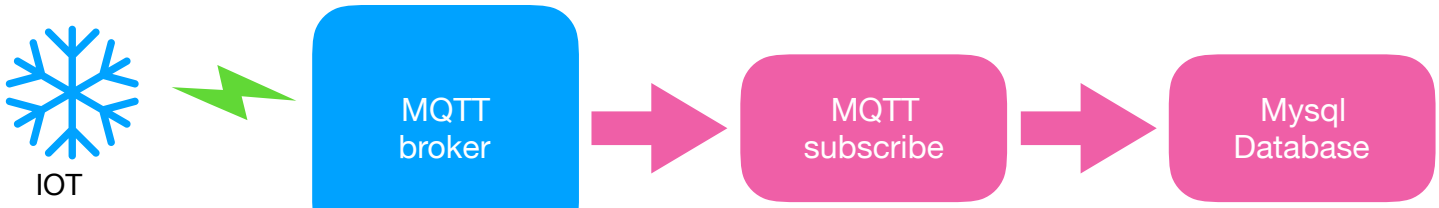
## 2. 安裝裝pypy3,及Python MQTT sdk

```
# 下載pypy3 (pypy3 非必要)
$ wget https://bitbucket.org/pypy/pypy/downloads/pypy3.6-v7.1.1-linux64.tar.bz2
# unzip it to ${PYPY}
$ ln -s ${PYPY}pypy3.6 /usr/local/bin/pypy
# install pip
$ wget https://bootstrap.pypa.io/get-pip.py
$ pypy3 get-pip.py
$ ln -s ${PYPY}pip3 /usr/local/bin/pypip
# 升级 pypy 的 pip wheel
$ pypip install -U pip wheel
# 安裝Python的MQTT sdk(from Eclipse)
$ pypip install paho-mqtt
```

```
# 安裝python需要的module
$ pip install aiohttp
$ pip install mysql-connector-python
$ pip install paho-mqtt
```

## 3. 安裝MySQL

```
# mysql port 3306
$ sudo apt-get install mysql-server
$ sudo apt-get install default-libmysqlclient-dev
# startup mysql
$ sudo systemctl start mysql.service
$ sudo systemctl enable mysql.service
```




topic:
DATA

context:
{type:datatype, value:number}

## * 系统架构 *

## 4. 测试Mosquitto

```
1 启动代理服务
$ systemctl start mosquitto.service

2 订阅主题
mosquitto_sub -v -t sensor
mosquitto_sub -v -t \$SYS\Broker\+

3 发布内容
mosquitto_pub -t sensor -m 12
```

## 5. 任务(7/25)

* 建立 MQTT Service
* 订阅 Topics
* 建立 MySQL Server
* 设计 MySQL DB table 结构
* 将订阅内容，insert到MySQL

未来项目
* 表现资料
* 根据资料发布Topics

## 6. 云,Mosquitto加密碼

server：
139.198.19.224
iothub.proadvancer.com

```
mosquitto 加上用戶密碼
cd /etc/mosquitto
mosquitto_passwd -c passwd pabox
```

用戶：pabox
密碼：1qaz2wsx3edc

加上用戶密碼的認證，杜絕外部無關的干擾

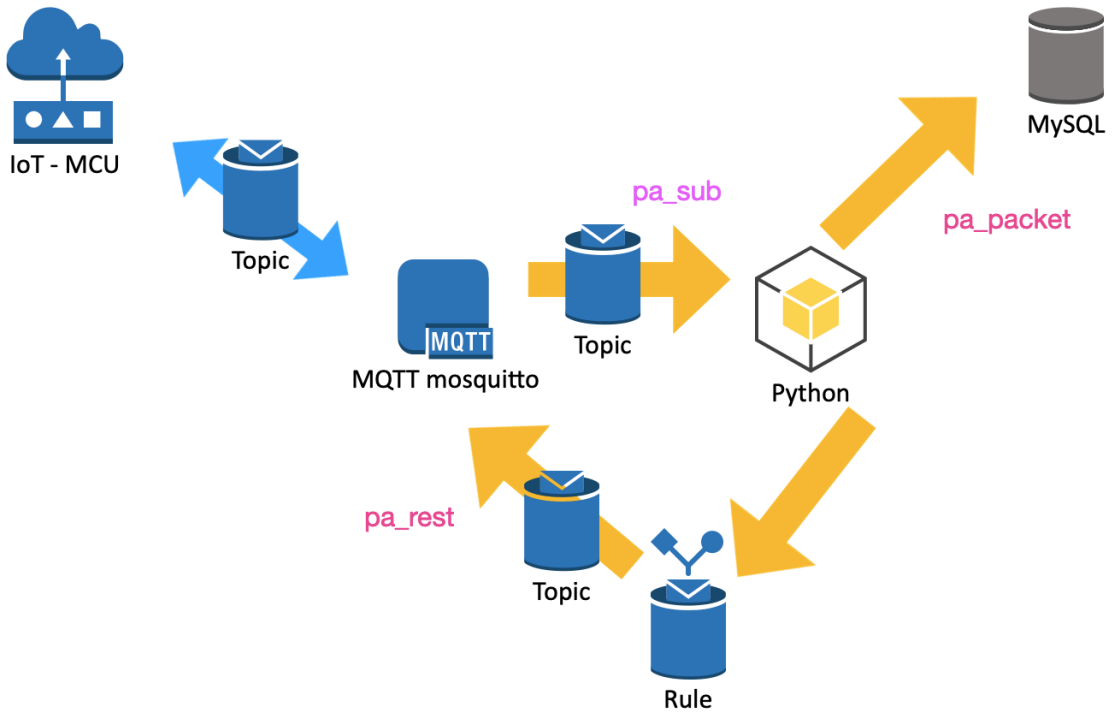
```
add
password_file /etc/mosquitto/passwd
to
/etc/mosquitto/mosquitto.conf
```

## 7. 程序功能

pa_rest.py
提供restful service
1.http://host:9000/pub?topic=___&cmd=___&data=[_,_]
  再发布至MQTT Broker，此处若topic以 _ 為首，
  則將會使用PABOX/当作是prefix，整個topic將會是PABOX/topic[1:]
2.http://host:9000/query/trandata
  查詢最近的trandata
3.http://host:9000/query/item
  查詢item
4.http://host:9000/cmd
  pub可使用的cmd及data樣式

但是目前port9000沒開，可以login server後使用curl 如
```
curl -v -globoff http://host:9000/pub?topic=__\&cmd=__\&data=[_,_]
curl -v -globoff http://host:9000/query/trandata
curl -v -globoff http://host:9000/query/item
curl -v -globoff http://host:9000/query/cmd
```
將對MQTT發布TOPIC及payload packet(cmd,data)

查詢內容：由未來業務內容決定，此處不做，只列舉兩個測試
查詢畫面：牽涉另外業務系統，此處不做



`pa_sub.py:`

訂閱subscribe from MQTT broker
解析message.payload之data packet(`pa_packey.py` parse_packet)
insert into mysql.pabox.trandata
~~設定必須收到訂閱的消息之後，必須回覆的指令(此步驟需釐清業務邏輯，不在我的範圍)~~

`pa_config.py:`

環境參數設定

```
HOST = "139.198.19.224"
PORT = 1883
USER = "pabox"
PASS = "1qaz2wsx3edc"
QoS  = 1
HTTP = 9000
SUB_TOPIC = "PABOX"
MY_HOST = "127.0.0.1"
MY_USER = "iot"
MY_PASS = "iot"
MY_DATABASE ="pabox"

# SUBSCRIBE codes those need to Publish back
MUST_REPLY_CODES = ('02',)

## mysql
GET_SQL={
    "add_trandata":("INSERT INTO `pabox`.`trandata`(`iot_id`,`code`,`code_name`,
                `seq`,`attr01`,`desc`,`data`) VALUES (%s, %s, %s, %s, %s, %s, %s)"),
    "trandata":("SELECT * from `pabox`.`trandata` order by `line_id` desc limit 100"),
    "item":("SELECT * from `pabox`.`item` limit 100"),
    }
```

## 8. MCU的messages

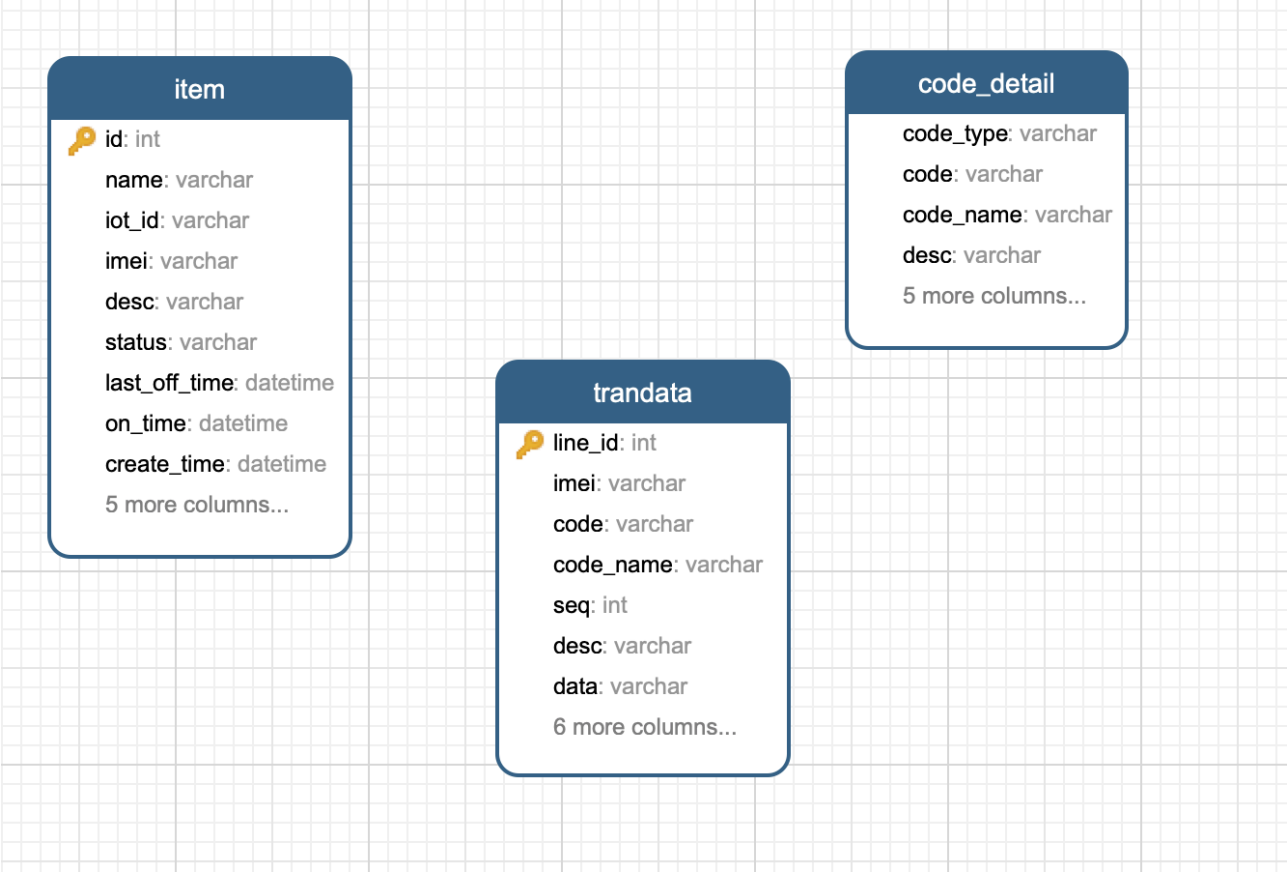
(在pa_packet.py中定义)

```
   "SETM":{"CODE":'00',"TYPE":"1","DESC":"工作模式转换"}, # 1 自控 0, 外控 1
    "PING":{"CODE":'01',"TYPE":"2","DESC":"连接握手(sec)"}, # 2 秒数
    "TSYN":{"CODE":'02',"TYPE":"7","DESC":"时间同步"}, # 7 year2+mon1+mday1+hour1+min1+sec1, 7bytes
    "OFF_":{"CODE":'03',"TYPE":"0","DESC":"断开连接"}, # 0
    "GETT":{"CODE":'04',"TYPE":"1,2","DESC":"获取温度(°C)"}, # 1+2   通道+温度
    "SETT":{"CODE":'05',"TYPE":"1,2,2","DESC":"设置温度上限和下限参数(°C)"}, # 1+2+2 通道+温度上限和温度
    "GETF":{"CODE":'06',"TYPE":"1,2","DESC":"获取风扇状态(mA)"}, # 1+2 风扇电流
    "SETF":{"CODE":'07',"TYPE":"1,1","DESC":"控制风扇动作"}, # 1+1 控制风扇动作
    "GETL":{"CODE":'08',"TYPE":"1,1","DESC":"查询锁的状态"}, # 1+1 通道+锁状态
    "SETL":{"CODE":'09',"TYPE":"1,1","DESC":"控制开锁动作"}, # 1+1 通道+锁动作
    "BIND":{"CODE":'0A',"TYPE":"1,1","DESC":"绑定风扇和温度控制"}, # 1+1 风扇通道+温度通道
    "LOG_":{"CODE":'0B',"TYPE":"0,N","DESC":"终端日志获取"}, # 0
    "GPS_":{"CODE":'0C',"TYPE":"0,N","DESC":"GPS定位经纬度信息"}, # 0 DDDDDddddNDDDDDddddE
    "TCOM":{"CODE":'0D',"TYPE":"2","DESC":"通信时间间隔(sec)"}, # 2
    "TSEN":{"CODE":'0E',"TYPE":"2","DESC":"数据采集唤醒间隔(sec)"}, # 2
    "VBAT":{"CODE":'0F',"TYPE":"2","DESC":"电池电压与电量(mV)"}, # 2
    "VOUT":{"CODE":'10',"TYPE":"2","DESC":"外来电压(mV)"}, # 2
    "SIGN":{"CODE":'11',"TYPE":"1","DESC":"通信信号强度(0~31)"}, # 1
    "CIMI":{"CODE":'12',"TYPE":"0,N","DESC":"SIM卡CIMI码"}, # 0
    "FXMA":{"CODE":'13',"TYPE":"2","DESC":"风扇最大的工作电流(mA)"}, # 2
    "DOWA":{"CODE":'14',"TYPE":"2","DESC":"舱门开启告警时间设定(sec)"}, # 2
    "TWAR":{"CODE":'15',"TYPE":"1,2","DESC":"温度通道告警设置(°C)"}, # 1 + 2 通道+温度
    "VLOW":{"CODE":'80',"TYPE":"2","DESC":"电池过低(mV)"}, # 2 mV
    "FLOW":{"CODE":'81',"TYPE":"2","DESC":"风扇堵转或者故障(mA)"}, # 2 mA
    "DOAL":{"CODE":'82',"TYPE":"2","DESC":"舱门开启时间过长告警(sec)"}, # s
    "DOTI":{"CODE":'83',"TYPE":"2","DESC":"舱门开启动作(sec)"}, # 2
    "VOVA":{"CODE":'84',"TYPE":"2","DESC":"外供电开始(mV)"}, # 2 mV
    "T_AL":{"CODE":'85',"TYPE":"1,2","DESC":"箱体温度告警(°C)"}, # 1+2 温度*10
```

```
int.from_bytes(pac[6:8], byteorder='little', signed=True) => low byte first
```

```
PACKET:
PACKET_HEAD=b'\x55\x8A'
DATA_LENGTH=LENGTH(PACKET)-4
PACKET_SEQ=AutoIncrement INT(0~255)
PACKET_CODE=code
PACKET_DATA=bytes
PACKET_CHK=checksum
PACKET_TAIL=b'\xFE\xAA'
```

# 9. 建立tables


item
id: int
name: varchar
iot_id: varchar
imei: varchar
desc: varchar
status: varchar
last_off_time: datetime
on_time: datetime
create_time: datetime
5 more columns...
code_detail
code_type: varchar
code: varchar
code_name: varchar
desc: varchar
5 more columns...
trandata
line_id: int
imei: varchar
code: varchar
code_name: varchar
seq: int
desc: varchar
data: varchar
6 more columns...

安装 mysql 后

```
$ sudo mysql
create user 'iot'@localhost identified by 'iot';
grant all on *.* to iot@localhost;
exit;
```

```
$ mysql -u iot -piot
create database pabox;
use pabox;
```

| | ITEM | | |
|---|---|---|---|
| ID | VARCHAR(30) | internal Unique ID | |
| NAME | VARCHAR(200) | IOT名字 | |
| IMEI | VARCHAR(200) | | |
| ITEM_DESC | VARCHAR(200) | IOT描述 | |
| STATUS | VARCHAR(10) | valid ? | |
| CREATE_TIME | DATETIME | 建立时间 | |

| | CODE_DETAIL | | |
|---|---|---|---|
| CODE_TYPE | VARCHAR(10) | | |
| CODE | VARCHAR(200) | | |
| DETAIL | VARCHAR(200) | | |
| DESC | VARCHAR(200) | | |

| | TRANDATA | | |
|---|---|---|---|
| ITEM_ID | INTEGER | | |
| LOCATION | VARCHAR(200) | GPS | |
| TRAN_CODE_TYPE | VARCHAR(200) | CODE_DETAIL.CODE_TYPE | |
| TRAN_CODE | VARCHAR(200) | CODE_DETAIL.CODE | |
| VAL | NUMBER | message value | |
| CREATE_TIME | DATETIME | | |

以上归纳出建立tables的script：

```
CREATE DATABASE `pabox`;
USE `pabox`;

CREATE TABLE `code_detail` (
  `code_type` varchar(32) DEFAULT NULL,
  `code` varchar(45) DEFAULT NULL,
  `code_name` varchar(45) DEFAULT NULL,
  `desc` varchar(45) DEFAULT NULL,
  `attr01` varchar(45) DEFAULT NULL,
  `attr02` varchar(45) DEFAULT NULL,
  `attr03` varchar(45) DEFAULT NULL,
  `attr04` varchar(45) DEFAULT NULL,
  `attr05` varchar(45) DEFAULT NULL,
  UNIQUE KEY `code_UNIQUE` (`code_type`,`code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


CREATE TABLE `item` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(64) DEFAULT NULL,
  `iot_id` varchar(45) DEFAULT NULL,
  `desc` varchar(45) DEFAULT NULL,
  `status` varchar(45) DEFAULT NULL,
  `last_off_time` datetime DEFAULT NULL,
  `on_time` datetime DEFAULT NULL,
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP,
  `imei` varchar(45) DEFAULT NULL,
  `attr01` varchar(45) DEFAULT NULL,
  `attr02` varchar(45) DEFAULT NULL,
  `attr03` varchar(45) DEFAULT NULL,
  `attr04` varchar(45) DEFAULT NULL,
  `attr05` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `item_iot_id` (`iot_id`),
  KEY `item_imei` (`imei`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


CREATE TABLE `trandata` (
  `line_id` int(11) NOT NULL AUTO_INCREMENT,
  `imei` varchar(32) DEFAULT NULL,
  `code` varchar(45) DEFAULT NULL,
  `code_name` varchar(45) DEFAULT NULL,
  `seq` int(11) DEFAULT NULL,
  `desc` varchar(45) DEFAULT NULL,
  `data` varchar(200) DEFAULT NULL,
  `attr01` varchar(45) DEFAULT NULL,
  `attr02` varchar(45) DEFAULT NULL,
  `attr03` varchar(45) DEFAULT NULL,
  `attr04` varchar(45) DEFAULT NULL,
  `attr05` varchar(45) DEFAULT NULL,
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`line_id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

## 10. 安裝執行

安裝

```
$ mkdir mqtt
$ cp pa*.py mqtt/*
```

執行

```
$ ~/mqtt/pa_sub.py
```