```
ESP32 sim800
=============================================
```

1. Message(AMSG)

```
       buffer max 20 records
       sms(recv)          --|           |-->  sms(send)
       mail(POP3)         --|--> AMSG --|-->  mail(SMTP)
       mqtt(subscribe)    --|           |-->  mqtt(public)
                                        |-->  logfile
```

```
struct amsg {
  bool enabled;
  bool alloc;
  bool subjForMe;
  bool okTelef;
  uint8_t typTO;
  uint8_t typFROM;
  char telef[XTELEF_SIZE];
  char from[XFROM_SIZE];
  char to[XTO_SIZE];
  char subject[XSUBJECT_SIZE];
  char msg[AMSG_MSG_SIZE];
  char smsTime[XSMSTIME_SIZE];
};
```

1.1 Message (input)

- sms(receive)

```
       parameters seting(WEB page SMS):
              typto: mail,mqtt,file
       sms -> AMSG
```

- mail(POP3)

```
       parameters seting(WEB page Email-POP3):
              typto: sms,mqtt,file
              subject
       if subject == POP3subject
              mailForMe=true
       if subject == "telef<+phone number>"
              mailForSMS=true
       if line in body(msg) == "telef<+phone number>\r\n"
              mailForSMS=true
       if mailForMe or mailForSMS then
              mail -> AMSG
              mail deleted
       else
              mail skip
```

- mqtt(subscribe)

```
       !! no to logfile
       1. to AMSG sms
              topic: cmdn/mqtt_DEVname/sentSMS
              msg:   JSON(Telef,Msg)
              -->    AMSG.telef, AMSG.msg
       2. to AMSG mail
              topic: cmdn/mqtt_DEVname/sentMAIL
              msg:   JSON(Recipient,Subject,Msg)
              -->    AMSG.to, AMSG.subject, AMSG.msg
       3. led1
              topic: cmdn/mqtt_DEVname/led1
              msg:  "on" / "off"
       4. delaySTATUS
              topic: cmdn/mqtt_DEVname/delaySTATUS
              msg:  value //unit seconds
       5. delaySENSOR
              topic: cmdn/mqtt_DEVname/delaySENSOR
              msg:  value //unit seconds
```

```
6. another
      topic: cmdn/mqtt_GLOBname/STATUS
      topic: cmdn/sonoffs/STATUS
      topic: cmdn/tasmotass/STATUS
```

1.2. Message action(output)

- sms(send)

```
if AMSG.telef is OK //== "+xxxxxxxxx"
      send sms ( AMSG.telef, AMSG.msg)
```

- mail(SMTP)

```
var p1,p2
if AMSG.to is OK //== "xxx@xxx.xxx"
      p1 = AMSG.to
else
      p1 = SMTP.recipient
if AMSG.subject != ""
      p2 = AMSG.subject
else
      p2 = SMTP.subject
send mail (p1, p2, AMSG.msg)
```

- mqtt(public)

```
topic: tele/mqtt_DEVname/amsg
msg:   JSON(typfrom,typto,from,to,subject,telef,smstime,msg)
```

- logfile(append)

```
write JSON(typfrom,typto,from,to,subject,telef,smstime,msg)
```

===

2. MQTT public STATUS, SENSOR

```
topic: stat/mqtt_DEVname/STATUS5
msg:   JSON(StatusNET{Hostname,IPAddress,Gateway,Subnetmask})

topic: tele/mqtt_DEVname/SENSOR
msg:   JSON(Time,Power,Charger,Led1,SHT{Temperature,Humidity},TempUnit)
```

```
=============================================
```

3. Event

```
      two destination DST[] parameters(WEB page Event-Dest[n]):
            telef, recipient
      four event EVENT[] parameters seting(WEB page Event-Event[n]):
            to: sms,mail,mqtt,file,dst1,dst2
            function
            subject
            msg
```

3.1. Event action(output)

```
outmsg = "EVT" + indx(1,2,3,4)
      + ";" + typ
      + ";" + "ON"/"OFF"
      + ";" + EVENT[].msg
```

- sms(send)

```
      if DST[].telef is OK //== "+xxxxxxxxx"
            send sms ( DST[].telef, outmsg)
```

- mail(SMTP)

```
      if DST[].recipient is OK //== "xxx@xxx.xxx"
            send mail (DST[].recipient, EVENT[].subject, outmsg)
```

- mqtt(public)

```
      topic: tele/mqtt_DEVname/event
      msg:   outmsg
      msg:   JSON(Time,Event,Func,State,Subject,Msg)
```

- logfile(append)

```
      write JSON(Time,Event,Func,State,Subject,Msg)

=============================================
```

4. FTP

```
      user: "esp32"
      pasw: "root"
=============================================
```

5. Setting

```
      file aval.ino
  {"NET01",VPTYP_STR,XNAME_SIZE,1,0,(void *)&WIFIssid,"...." },
  {"NET02",VPTYP_STR,XPASS_SIZE,1,0,(void *)&WIFIpasw,"...." },
  {"NET03",VPTYP_STR,XNAME_SIZE,1,0,(void *)&WIFImdns,"esp32gsm" },
  {"NET04",VPTYP_STR,XIP_SIZE,1,0,(void *)&WIFIip,"10.0.0.5" },
  {"NET05",VPTYP_STR,XIP_SIZE,1,0,(void *)&WIFImask,"255.255.255.0" },
  {"NET06",VPTYP_STR,XIP_SIZE,1,0,(void *)&WIFIgate,"10.0.0.1" },
  {"NET07",VPTYP_STR,XIP_SIZE,1,0,(void *)&WIFIdns,"10.0.0.1" },
  {"NET08",VPTYP_STR,XNAME_SIZE,1,0,(void *)&SNTPserver,"pool.ntp.org" },
  {"NET09",VPTYP_BOOL,1,1,0,(void *)&WIFIstatic,NULL },
  {"NET10",VPTYP_UI8,1,0,0,(void *)&wifi_log,NULL },

  {"MQT00",VPTYP_BOOL,1,1,0,(void *)&mqtt_enabled,NULL },
  {"MQT01",VPTYP_STR,XIP_SIZE,1,0,(void *)&mqtt_server,NULL },
  {"MQT02",VPTYP_STR,XNAME_SIZE,1,0,(void *)&mqtt_DEVname,"ESP32gsm"},
  {"MQT03",VPTYP_STR,XNAME_SIZE,1,0,(void *)&mqtt_GLOBname,"Tasmota" },
  {"MQT04",VPTYP_STR,XUSER_SIZE,1,0,(void *)&mqtt_user,"admin" },
  {"MQT05",VPTYP_STR,XPASS_SIZE,1,0,(void *)&mqtt_pasw,"1234" },
  {"MQT06",VPTYP_UI32,4,1,0,(void *)&mqtt_refrSTATUS,"60" },
  {"MQT07",VPTYP_UI32,4,1,0,(void *)&mqtt_refrSENSOR,"30" },
  {"MQT10",VPTYP_UI8,1,0,0,(void *)&mqtt_log,NULL },

  {"WEB01",VPTYP_STR,XUSER_SIZE,1,0,(void *)&WEBuser1,"admin" },
  {"WEB02",VPTYP_STR,XPASS_SIZE,1,0,(void *)&WEBpass1,"admin" },
  {"WEB03",VPTYP_STR,XUSER_SIZE,1,0,(void *)&WEBuser2,"root" },
  {"WEB04",VPTYP_STR,XPASS_SIZE,1,0,(void *)&WEBpass2,"root" },

=============================================
```