

Module	Features	Update/Get Domoticz data via	Send/Get data to/from module	Send/Get module data via
mqtt_Cluster.js	High Availability Domoticz cluster management based on two servers (active/passive nodes) synchronized via MQTT. Under normal condition, this script running in the backup server monitors the main server and synchronizes its own Domoticz database. If the main server fails, this script manages the backup server in order to continue to deliver the service.	Update --> MQTT[B_domoticz/in] Get <-- MQTT[M_domoticz/in, M_domoticz/out] Get(Heartbeat) <-- JSON_API Update(Failure Alert) --> JSON_API		
iot_ESP8266_GM43.ino	Lighting management. Lighting can be switched on/off from legacy wall pushbuttons or Domoticz. If lighting is switched on/off using the wall pushbuttons, an MQTT feedback message is sent to Domoticz.	Update --> MQTT[domoticz/in] Get <-- MQTT[domoticz/out]	NO	
iot_ESP8266_DHT22.ino	Temperature sensor using DHT22. Answer to HTTP JSON requests.	NO	NO	
iot_ESP8266.js	Polls the temperature sensors and log the values within Domoticz database. Compute and log degrees-days. Monitor the temperature sensors and Raise failure flag if sensors don't answer to requests. Compute total heating energy usage from all heaters energy usage data (ACS712sensors). Computes Thermal loss and Heating/Cooling Ratios.	Update --> JSON_API	iot_ESP8266_DHT22.ino	Get <-- JSON_API
iot_ESP8266_ACS712.ino	Heater management. Read ACS712 sensors. Send via MQTT its heater energy usage. Start/stop its heater according to commands received from MQTT. Self Learning of its heater Nominal Power.	Update --> MQTT[domoticz/in] Get <-- MQTT[domoticz/out]	iot_Orchestrator.js	Send --> MQTT[heating/in] Get <-- MQTT[heating/out]
iot_Orchestrator.js	Manage Heaters and Heating Zones (Scheduled TOP Start/Stop sent by this program to heating/out). Compute and log heaters characteristics (listen heaters consumption log messages at domoticz/in and heating/in). Monitor ESP8266-ACS712/Heaters, ESP8266/Lighting, Dahua VTH Secpanel and Raspberry/Alarm server and raise failure flag if one of them die (listen MQTT Will messages at domoticz/in). Bridge for Domoticz SecurityPanel to allow to arm/disarm the wired alarm by sending secured MD5 signed MQTT message. Update Domoticz SecurityPanel state when alarm arming/disarming was done by another Alarm client like VTH SecurityPanel.	Get <-- JSON_API Update --> JSON_API Get <-- MQTT[domoticz/out]	iot_ALARM-SVR.js, Dahua-VTH-SecPanel.py and iot_ESP8266_ACS712.ino	Get <-- MQTT[heating/in] Get <-- MQTT[domoticz/in] Get <-- MQTT[domoticz/out] Send --> MQTT[heating/out] Send --> MQTT[domoticz/out]
Dahua-VTH-SecPanel.py	Bridge between Dahua DHIP/DVRIP protocol and MQTT. Allow to use Dahua VTH as a SecurityPanel to arm/disarm the wired alarm by sending secured MD5 signed MQTT message. Update VTH alarm state when alarm arming/disarming was done by another Alarm client like Domoticz SecurityPanel.	Get <-- MQTT[domoticz/out]	iot_ALARM-SVR.js and iot_Orchestrator.js	Get <-- MQTT[domoticz/out] Send --> MQTT[domoticz/out]
iot_ALARM-SVR.js	Alarm server. Manage the CVQ6081 wired alarm Appliance : arm and disarm the Alarm using the Raspberry GPIO/Relay as a keyswitch and get alarm Alert state at CVQ6081 backpanel	Update --> MQTT[domoticz/in] Get <-- MQTT[domoticz/out]	iot_Orchestrator.js and Dahua-VTH-SecPanel.py	Get <-- MQTT[domoticz/out]
DomoticzLogRotate.sh	Filter various home automation log files to give a 360° view on only the major events with one logfile dashboard . Logrotate this 360° view logfile to keep daily images. Backup Domoticz databases of main and backup servers.			