

CLEES System

Manual

Edition: 2018-12-29

by

Tompa

Table of Contents

Introduktion.....	3
Om CLEES.....	3
Akronymer.....	3
CLEES Kommunikation.....	4
CLEES använder MQTT.....	4
CLEES MQTT Meddelandestruktur.....	5
CLEES Inställningar.....	7
CLEES Inställningar för ansluten hårdvara.....	7
CLEES Inställningsfiler.....	8
clees_settings.json.....	8
clees_objects.json.....	9
clees_directcontrol.json.....	12

Introduktion

Denna manual beskriver i detalj hur CLEES System fungerar.

Om CLEES

CLEES är en akronym för "Controll your Layout using Ethernet and Easy Scripts" och är ett öppet system för att styra växlar och signaler på din modelljärnväg.

Akronymer

Opi0 Orange Pi Zero

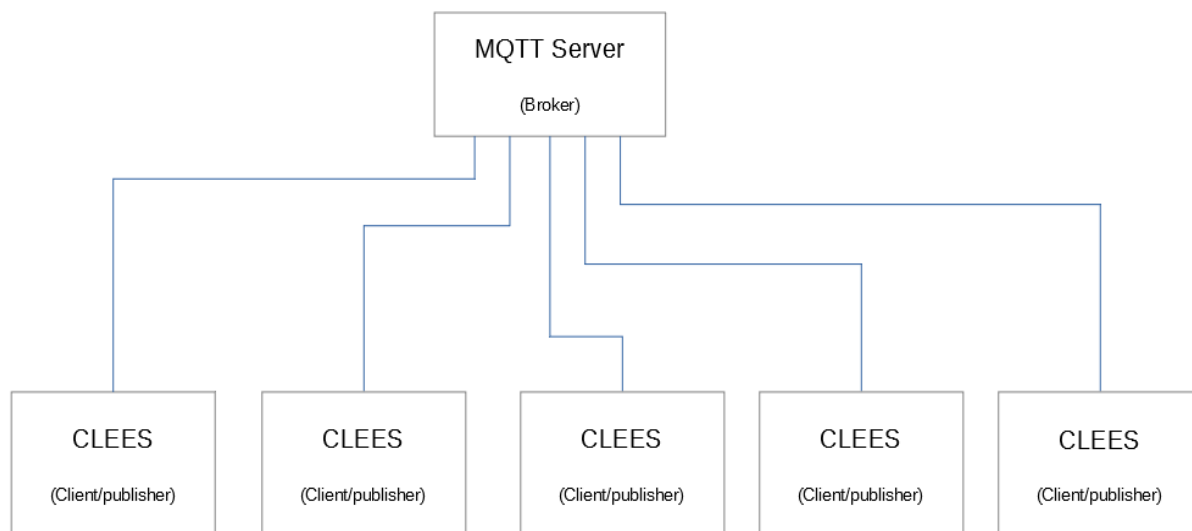
MQTT Message Queuing Telemetry Transport och är ett internetprotokoll

IoT Internet of Things, men vi som håller på med modelljärnväg tänker Internet of Trains

CLEES Kommunikation

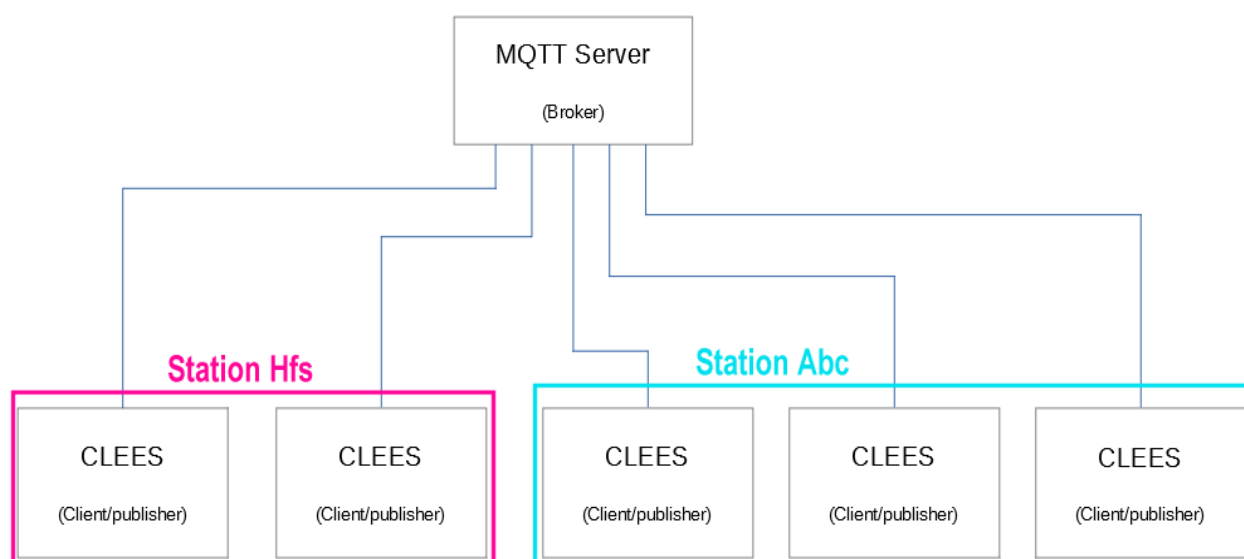
CLEES använder MQTT

MQTT protokollet används för att kommunicera. All kommunikation sker till och från MQTT servern, även kallad "Broker". Ingen kommunikation går mellan enheterna.



En trafikplats består av en eller flera clees enheter. En station med växlar bara i stationsändarna har typiskt två enheten och ett modulkomplex med många sektioner har typiskt en enhet per sektion.

Alla enheter sitt på samma nätverk och använder en gemensam MQTT server.



CLEES MQTT Meddelandestruktur

Grundregler.

Ett CLEES MQTT meddelande innehåller bara gemena tecken.

Ett CLEES MQTT meddelande inleds alltid med "clees/". Det är meddelandets topnivå.

Ett CLEES MQTT meddelande är uppbyggt på följande sätt

clees/<stationsförkortning>/<meddelandetyptyp>/<banobjekttyp>/<banobjektid>

clees/	<stationsförkortning>/	<meddelandetyptyp>/	<banobjekttyp>/	<banobjektid>
clees/	Indentifiera trafikplatsen. Använd den förkortning som trafikplatsen har, oftast treställig.	Talar om vilken typ av meddelande det är. Följande typer finns definierade: cmd/ = kommando rep/ = rapportering Kommande typer på förslag: stat/ = status	Talar om vilken typ av banobjekt som meddelandet gäller för: Följande objekt finns definierade: t/ = växel (turout) s/ = signal o/ = utgång (output) b/ = knapp (button) Kommande typer på förslag: d/ = spårledning (occupant detection)	Stationsunikt id. Kan vara nummer eller alphanumeriskt. Vanligtvis ett tresiffrigt tal

Exempel på hur ett CLEES MQTT meddelande kan se ut:

clees/hfs/cmd/t/203

Exemplelmeddelandet pekar ut växel 203 på trafikplats hfs och är ett kommando

Varje meddelande bär på en datadel. Datadelen innehåller ett värde som beskriver banobjektet. Olika banobjekt har olika typer av värden. Följande värden för respektive banobjekttyp gäller:

<meddelandetyptyp>/<banobjekt>/	Datadel av meddelandet
cmd/t/	Ett kommando till en växel vill att växeln skall göra något. Följande kommando finns definierade: throw = lägg växeln avvikande close = lägg växeln i raktläge Kommande definitioner på förslag: route=123 = ange att växeln tillhör en viss tågväg route=0 = ange att växeln inte längre tillhör tågväg report = återrapportera lock = lås växeln

<meddelandetyp>/<banobjekt>/	Datadel av meddelandet
cmd/o/	<p>Ett kommando till en digital utgång vill att utgången skall göra något. Följande kommandon finns definierade:</p> <p>activate = utgången aktiveras (hög el låg beronde på om den är inverterad eller ej)</p> <p>deactivate = utgången avaktiveras (hög el låg beronde på om den är inverterad eller ej)</p> <p>Kommande definitioner på förslag:</p> <p>blink = utgången pulsar med en frekvens som angetts för respektive utgång</p>
cmd/s/	Inget definierat
rep/t/	<p>Rapport från en växel. Följande rapporter är definierade:</p> <p>thrown = Växeln ligger i avvikande läge</p> <p>closed = Växeln ligger i rakt läge</p> <p>changing = Växeln har påbörjat omläggning.</p> <p>Kommande definitioner på förslag:</p> <p>locked = Växeln är låst och kan inte läggas om</p> <p>inroute=123 = Växeln ingår just nu i tågväg 123</p>
rep/b/	<p>Rapport från en strömbrytare/knapp/annan manuell kontroll. Följande rapporter är definierade:</p> <p>activated = knappen tryckes ned</p> <p>deactivated = knappen släppes upp</p>
rep/o/	<p>Rapport från en utgång. Följande rapporter är definierade:</p> <p>activated = utgången aktiverades</p> <p>deactivated = utgången avaktiverades</p>

Exempel 1, Växel nr 141 på station vxs lägger om
clees/vxs/rep/t/141 msg=changing

Exempel 2, Knapp v233 på station aab nedtrycktes
clees/aab/rep/b/v233 msg=activated

Exempel 3, Kommando för att rakställa växel infart på station dba
clees/dba/cmd/t/infart msg=close

CLEES Inställningar

CLEES Inställningar för ansluten hårdvara

I filen `clees_io.py` anges vilken typ av hårdvara som är kopplad till Pi:en och som skall användas av CLEES.

Först behöver du ange vilken datortyp som CLEES kör på. I början på `clees_io.py` filen hittar du en sektion som ser ut som nedan.

```
# ----- Hardware Definition -----
# Define what Hardware you are running on by
# Comment out all except the platform you
# are running on
HW = "RPi-3B"          # Raspberry Pi 3 Model B
#HW = "OPi0-H2"        # Orange Pi Zero with H2 Chipset
```

Välj typ av hårdvara genom att ta bort `#` före namnet på den hårdvara du använder. Bara en hårdvara får vara aktiverad. De övriga ska ha `#` först på raden

Ange antal PWM kort som är kopplade på I2C bussen

```
# PCA9685
number_of_pca9685 = 2      # Number of pca9685 modules connected to I2c
start_adr_pca9685 = 0x41   # Address to first pca9685 module.
                           # Additional modules expected to be on consecutive
                           # addresses
freq_pca9685 = [50,60]    # List PWM freq for every pwm. Number of values in
                           # list must match number_of_pca9685
```

PWMfrekvensen för servon skall normalt vara 50Hz men praktiska prov visar att mellan 50 och 80Hz fungerar bra med de flesta servon.

Ange antal I/O kort som är kopplade på I2C bussen

```
# PCF8574
# Comes in two flavors with different address space and will here be defined
# as two separate board types with their lists.
number_of_pcf8574 = 1      # Number of pcf8574 modules connected to I2c
start_adr_pcf8574 = 0x20   # Address to first pcf8574 module.
                           # Additional modules expected to be on consecutive
                           # addresses
number_of_pcf8574A = 1    # Number of pcf8574A modules connected to I2c
start_adr_pcf8574A = 0x38  # Address to first pcf8574A module.
                           # Additional modules are expected to be on
                           # consecutive addresses
```

Ange I/O pinnar som används i Pi:ns egen GPIO

```
# --- GPIO
# List Board Connector pin numbers separated by commas
# Boardpins means the actual pin number in the board connector, not Port or
# Soc pin numbering
# Boardpins that are not GPIO (eg tied to GND, VCC , .. ) can ofcasue not be
# used and will through an error
# Don't use boardpins = 3+5 that is used for I2c since that will screw up
# I2c communication
Boardpins_used_for_input  = [18]
Boardpins_used_for_output = [22]    # 22 used for status LED (Steady=online,
                                     # blink=wait for broker)
```

Flera pinnan anges med komma som separatortecken. Ex:

```
Boardpins_used_for_input  = [18,22,23]
```

CLEES Inställningsfiler

Det finns flera inställningsfiler som styr hur CLEES fungerar. Inställningsfilerna har JSON format

clees_settings.json

Med clees_settings.json ställs grundläggande inställningar, se exempel nedan

```
{
  "stationfullname"      : "Holmfors",
  "stationshortname"     : "hfs",
  "mqtt-broker-ipadress"  : "172.16.1.2",
  "mqtt-broker-port"     : 1883,
  "mqtt-broker-keep-alive" : 30,
  "mqtt-qos"             : 0,
  "systemledgpio"        : 22,
  "systemledoutputobjectid" : "sysled",
  "log-mqtt"              : 1,
  "log-io"                : 1
}
```

SystemLED är Lysdiod som tänds när CLEES har startat upp kör. Lysdioden skall kopplas till en I/O en av Pi:ens GPIO pinnar. Med **"systemledgpio": 22** definieras vilken av pinnarna som skall användas.

SystemLED kan fås signalera via MQTT. Ange då **"systemledoutputobjectid": "namn"** ett namn och det namnet kommer användas i MQTT meddelandet.

CLEES har en loggfil. Till den loggas normalt bara start, stop och eventuella fel som uppstår i mjukvaran eller dess inställningar. Det går att utöka loggen med fler meddelanden vilket underlättar felsökning och idriftsättning. Med **"logg-mqtt": 1** kommer alla mqtt meddelanden att loggas. Med **"log-io": 1** kommer alla in och utgångs-operationer att loggas.

Sätt till 0 för att hindra loggning av respektive händelse.

Om du väljer ha dessa extra loggningar aktiverade, tänk då på att loggfilen kommer med tiden växa sig väldigt stor.

clees_objects.json

Med clees_objects.json bygger man upp listor över vilka banobjekt som anslutna och till vilken hårdvara dom är anslutna till. Varje typ av banobjekt har sin lista med dess egenskaper och inställningar. I början på filen finns också några mer generella inställningar.

```
{
  "speedsteps" : 3,
  "homepositiondelayonstart" : 0.8,
  "turnouts" : [
    {
      "id" : "301",
      "devicetype" : "pca9685",
      "devicenr" : 0,
      "pwmnr" : 0,
      "pwmwidthclosed" : 200,
      "pwmwidththrown" : 450,
      "homeposition" : "C",
      "inroute" : 0,
      "slaveoutputid" : "451"
    }
  ],
  "buttons" : [
    {
      "id" : "120",
      "devicetype" : "pcf8574A",
      "devicenr" : 0,
      "pinnr" : 0,
      "inverted" : 1
    }
  ],
  "outputs" : [
    {
      "id" : "451",
      "devicetype" : "pcf8574A",
      "devicenr" : 0,
      "pinnr" : 6,
      "inverted" : 1,
      "stateonstart" : "deactivated"
    }
  ]
}
```

Generella inställningar för växlar:

"speedsteps": 3 anger hur snabbt växelservot kommer att flytta sig när växeln läggs om. Kan ställas från 1 och högre. 1 betyder långsamt och högre tal snabbare.

"homepositiondelayonstart" : 0.8 anger med hur långt mellanrum växlarna kommer ställas till sin hemposition när CLEES startas. Växlarna läggs om en efter en för att inte dra för mycket ström vid uppstart. Siffran som anges är i sekunder och delar av sekunder kan också anges, tex 0.8

"turnouts" : [...] håller en lista på sektionens växlar.

För varje växel behöver man ange till vilken hårdvara den är ansluten, ge växeln ett id och lite annan information.

"id": "301" ger växeln en identifiering. Detta id används i MQTT meddelandet och behöver alltså vara unikt för varje växel på stationen. Lämpligt är att ge växeln ett id som växlar i verkligheten har. Både siffror och bokstäver är tillåtet men använd inte mellanslag eller andra specialtecken.

"homeposition": "C" anger i vilket läge växeln skall ställas när CLEES startar. C betyder Closed och T betyder Thrown

"inroute": 0 associerar växeln med en speciell tågväg. Skall alltid anges 0. Ännu ej implementerad.

"devicetype": "pca9685" anger till vilken typ av hårdvara som styr växeln. pca9685 betyder det PWMkort som kan driva 16st servon och är bestäckt med just kretsen pca9685

"devicenr": 0 pekar ut vilket kort (om flera pca9685 används) som styr växeln. Noll (0) betyder det kort som har adressen som anges i clees_io.py som startadress. Ett (1) betyder det kort som har adressen +1 som anges i clees_io.py som startadress osv...

"pwmnr": 0 pekar ut vilken av servoutgångarna (0-15) på pca9685 kortet som styr växeln.

"pwmwidthclosed": 200 Anger vilken pulslängd som PWMen skall ställas till för att servot skall skicka växeln till Closed (= rakspår). Normalt ett tal mellan 100 och 500 som behöver justeras för varje servo när dessa är monterade i anläggningen eller modulen.

"pwmwidththrown": 450 Anger vilken pulslängd som PWMen skall ställas till för att servot skall skicka växeln till Thrown (= avvikandespår). Normalt ett tal mellan 100 och 500 som behöver justeras för varje servo när dessa är monterade i anläggningen eller modulen.

"slaveoutputid": "451" Ange id:et på den output som skall aktiveras när växeln ställs i Thrown. Används för 2R att driva relät som polariserar växeln.

"buttons" : [...] håller en lista på sektionens knappar.

"id": "120" ger knappen en identifiering. Detta id används i MQTT meddelandet och behöver alltså vara unikt för varje knapp på stationen. Både siffror och bokstäver är tillåtet men använd inte mellanslag eller andra specialtecken.

"devicetype": "pcf8574A" anger till vilken typ av hårdvara som knappen är kopplad till. pcf8574 eller pcf8574A betyder det IOkort som har 8st I/O pinnar och är bestyckade med just kretsen pcf8574 eller pcf8574A

"devicenr": 0 pekar ut vilket kort (om flera pcf8574A används) som knappen är kopplad till. Noll (0) betyder det kort som har adressen som anges i clees_io.py som startadress. Ett (1) betyder det kort som har adressen +1 som anges i clees_io.py som startadress osv...

"pinnr": 0 pekar ut vilken av de 8st I/O pinnarna som knappen är kopplat till. Ange 0..7

"inverted": 1 Anger om pinnen skall tolkas inverterat eller ej. Knappar som kopplar ingången till signaljord (GND) när knappen aktiveras anges som "inverted": 1 för att CLEES skall tolka knappen på rätt sätt. Ange 0..1

"outputs" : [...] håller en lista på sektionens digitala utgångar.

"id": "451" ger utgången en identifiering. Detta id används i MQTT meddelandet och behöver alltså vara unikt för varje knapp på stationen. Både siffror och bokstäver är tillåtet men använd inte mellanslag eller andra specialtecken.

"devicetype": "pcf8574A" anger till vilken typ av hårdvara som utgången är kopplad till. pcf8574 eller pcf8574A betyder det IOkort som har 8st I/O pinnar och är bestyckade med just kretsen pcf8574 eller pcf8574A

"devicenr": 0 pekar ut vilket kort (om flera pcf8574A används) som utgången är kopplad till. Noll (0) betyder det kort som har adressen som anges i clees_io.py som startadress. Ett (1) betyder det kort som har adressen +1 som anges i clees_io.py som startadress osv...

"pinnr": 6 pekar ut vilken av de 8st I/O pinnarna som utgången är kopplat till. Ange 0..7

"inverted": 1 Anger om pinnen skall styras inverterat eller ej när den aktiveras. Ange 0..1

"stateonstart": "deactivated" Anger om utgången skall vara aktiverad eller avaktiverad när CLEES startas. Ange "activated" eller "deactivated".

clees_directcontrol.json

Filen `clees_directcontrol.json` används för att bygga upp enkla samband vad man vill ska hända när ett viss meddelande skickas via MQTT servern. `clees_directcontrol.json` filen ska du använda om du vill att en speciell knapp ska styra en speciell växel.

I `clees_directcontrol.json` bygger man upp en lista av händelser, där vi väntar på ett speciellt meddelande och skickar då ett annat meddelande. Nedan syns ett exempel där knapp 120 och knapp 121 styr växel 301

```
"reportmessages" : [  
  {  
    "listenfor": "rep/b/120",  
    "whenmsg"   : "activated",  
    "sendto"    : "cmd/t/301",  
    "withmsg"   : "close"  
  },  
  {  
    "listenfor": "rep/b/121",  
    "whenmsg"   : "activated",  
    "sendto"    : "cmd/t/301",  
    "withmsg"   : "throw"  
  }  
]
```

"listenfor": "rep/b/120" betyder att vi väntar på att "button" med id="120"

"whenmsg": "activated" ska rapportera "activated"

"sendto": "cmd/t/301" och då skickar vi "Command" till "Turnout" med id="301"

"withmsg": "close" att växeln skall gå till "close"