

```

#Library Millis
import time

import ttn

app_id = "lorawan_lg01s_gateway"
access_key = "ttn-account-v2.KRlh5dbuKtFtTJV-i07SrpmhCY_oCAjkzi7ekYggQUI"
dev_id = "ttgo_lora_esp32"
payload = { "led_state": "on" }

#Firestore Configuration
import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore

cred = credentials.Certificate("/home/pi/Projek Monitoring Energi/monitoring-energy-2-firebase-adminsdk-dddqf-bda3c4c53b.json")
firebase_admin.initialize_app(cred)

sPzemAvg1_ref = firestore.client().collection('Sensor_PZEM_Avg1')
sPzem1R_ref = firestore.client().collection('Sensor_PZEM1_R')
sPzem1S_ref = firestore.client().collection('Sensor_PZEM1_S')
sPzem1T_ref = firestore.client().collection('Sensor_PZEM1_T')
sPzemAvg2_ref = firestore.client().collection('Sensor_PZEM_Avg2')
sPzem2R_ref = firestore.client().collection('Sensor_PZEM2_R')
sPzem2S_ref = firestore.client().collection('Sensor_PZEM2_S')
sPzem2T_ref = firestore.client().collection('Sensor_PZEM2_T')
sPzem3_ref = firestore.client().collection('Sensor_PZEM3')
sPzem4_ref = firestore.client().collection('Sensor_PZEM4')
sPzemE_ref = firestore.client().collection('Sensor_PZEM_Energy')

#Realtime database configuration
import pyrebase

config = {
    "apiKey": "",
    "authDomain": "monitoring-energy-2.firebaseio.com",
    "databaseURL": "https://monitoring-energy-2-default-rtdb.firebaseio.com/",
    "storageBucket": "monitoring-energy-2.appspot.com"
}
firebase = pyrebase.initialize_app(config)
db = firebase.database()

#Variable

ttnVoltage1R = 0
ttnCurrent1R = 0
ttnPower1R = 0
ttnEnergy1R = 0
ttnPf1R = 0

ttnVoltage1S = 0
ttnCurrent1S = 0
ttnPower1S = 0
ttnEnergy1S = 0
ttnPf1S = 0

ttnVoltage1T = 0
ttnCurrent1T = 0

```

```
ttnPower1T = 0
ttnEnergy1T = 0
ttnPf1T = 0
```

```
ttnVoltage2R = 0
ttnCurrent2R = 0
ttnPower2R = 0
ttnEnergy2R = 0
ttnPf2R = 0
```

```
ttnVoltage2S = 0
ttnCurrent2S = 0
ttnPower2S = 0
ttnEnergy2S = 0
ttnPf2S = 0
```

```
ttnVoltage2T = 0
ttnCurrent2T = 0
ttnPower2T = 0
ttnEnergy2T = 0
ttnPf2T = 0
```

```
ttnVoltage3 = 0
ttnCurrent3 = 0
ttnPower3 = 0
ttnEnergy3 = 0
ttnPf3 = 0
```

```
ttnVoltage4 = 0
ttnCurrent4 = 0
ttnPower4 = 0
ttnEnergy4 = 0
ttnPf4 = 0
```

```
def uplink_callback(msg, client):
    global ttnVoltage3, ttnCurrent3, ttnPower3, ttnEnergy3, ttnPf3
```

```
    print("Received uplink from ", msg.dev_id)
    print(msg.payload_fields)
```

```
    ttnVoltage1R = msg.payload_fields.Voltage3
    ttnCurrent1R = msg.payload_fields.Current3
    ttnPower1R = msg.payload_fields.Power3
    ttnEnergy1R = msg.payload_fields.Energy3
    ttnPf1R = msg.payload_fields.Pf3
```

```
    ttnVoltage1S = msg.payload_fields.Voltage3
    ttnCurrent1S = msg.payload_fields.Current3
    ttnPower1S = msg.payload_fields.Power3
    ttnEnergy1S = msg.payload_fields.Energy3
    ttnPf1S = msg.payload_fields.Pf3
```

```
    ttnVoltage1T = msg.payload_fields.Voltage3
    ttnCurrent1T = msg.payload_fields.Current3
    ttnPower1T = msg.payload_fields.Power3
    ttnEnergy1T = msg.payload_fields.Energy3
    ttnPf1T = msg.payload_fields.Pf3
```

```
    ttnVoltage2R = msg.payload_fields.Voltage3
```

```

ttnCurrent2R = msg.payload_fields.Current3
ttnPower2R = msg.payload_fields.Power3
ttnEnergy2R = msg.payload_fields.Energy3
ttnPf2R = msg.payload_fields.Pf3

ttnVoltage2S = msg.payload_fields.Voltage3
ttnCurrent2S = msg.payload_fields.Current3
ttnPower2S = msg.payload_fields.Power3
ttnEnergy2S = msg.payload_fields.Energy3
ttnPf2S = msg.payload_fields.Pf3

ttnVoltage2T = msg.payload_fields.Voltage3
ttnCurrent2T = msg.payload_fields.Current3
ttnPower2T = msg.payload_fields.Power3
ttnEnergy2T = msg.payload_fields.Energy3
ttnPf2T = msg.payload_fields.Pf3

ttnVoltage3 = msg.payload_fields.Voltage3
ttnCurrent3 = msg.payload_fields.Current3
ttnPower3 = msg.payload_fields.Power3
ttnEnergy3 = msg.payload_fields.Energy3
ttnPf3 = msg.payload_fields.Pf3

ttnVoltage4 = msg.payload_fields.Voltage3
ttnCurrent4 = msg.payload_fields.Current3
ttnPower4 = msg.payload_fields.Power3
ttnEnergy4 = msg.payload_fields.Energy3
ttnPf4 = msg.payload_fields.Pf3

mqtt_client.set_downlink_callback(downlinkCallback)
mqtt_client.send(dev_id, payload, port=1, conf=False, sched="replace")

def downlinkCallback(mid, client):
    print("Received downlink with id ", mid)

handler = ttn.HandlerClient(app_id, access_key)
mqtt_client = handler.data()
mqtt_client.set_uplink_callback(uplink_callback)
mqtt_client.connect()

def millis():
    return int(round(time.time() * 1000))

def millis1(howlong):
    start = millis()

    sPzemAvg1_ref.add({
        "VoltageAvg1":ttnVoltage3,
        "CurrentAvg1":ttnCurrent3,
        "PowerAvg1":ttnPower3,
        "EnergyAvg1":ttnEnergy3,
        "Power_FactorAvg1":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem1R_ref.add({
        "Voltage1R":ttnVoltage3,
        "Current1R":ttnCurrent3,
        "Power1R":ttnPower3,

```

```

        "Energy1R":ttnEnergy3,
        "Power_Factor1R":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem1S_ref.add({
        "Voltage1S":ttnVoltage3,
        "Current1S":ttnCurrent3,
        "Power1S":ttnPower3,
        "Energy1S":ttnEnergy3,
        "Power_Factor1S":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem1T_ref.add({
        "Voltage1T":ttnVoltage3,
        "Current1T":ttnCurrent3,
        "Power1T":ttnPower3,
        "Energy1T":ttnEnergy3,
        "Power_Factor1T":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzemAvg2_ref.add({
        "VoltageAvg2":ttnVoltage3,
        "CurrentAvg2":ttnCurrent3,
        "PowerAvg2":ttnPower3,
        "EnergyAvg2":ttnEnergy3,
        "Power_FactorAvg2":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem2R_ref.add({
        "Voltage2R":ttnVoltage3,
        "Current2R":ttnCurrent3,
        "Power2R":ttnPower3,
        "Energy2R":ttnEnergy3,
        "Power_Factor2R":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem2S_ref.add({
        "Voltage2S":ttnVoltage3,
        "Current2S":ttnCurrent3,
        "Power2S":ttnPower3,
        "Energy2S":ttnEnergy3,
        "Power_Factor2S":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem2T_ref.add({
        "Voltage2T":ttnVoltage3,
        "Current2T":ttnCurrent3,
        "Power2T":ttnPower3,
        "Energy2T":ttnEnergy3,
        "Power_Factor2T":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })
    sPzem3_ref.add({
        "Voltage3":ttnVoltage3,
        "Current3":ttnCurrent3,
        "Power3":ttnPower3,
        "Energy3":ttnEnergy3,
        "Power_Factor3":ttnPf3,
        "timestamp": firestore.SERVER_TIMESTAMP
    })

```

```

})
sPzem4_ref.add({
    "Voltage4":ttnVoltage3,
    "Current4":ttnCurrent3,
    "Power4":ttnPower3,
    "Energy4":ttnEnergy3,
    "Power_Factor4":ttnPf3,
    "timestamp": firestore.SERVER_TIMESTAMP
})
sPzemE_ref.add({
    "Energy_Total":ttnEnergy3,
    "timestamp": firestore.SERVER_TIMESTAMP
})

pzemAvg1 = {"VoltageAvg1":ttnVoltage3,
            "CurrentAvg1":ttnCurrent3,
            "PowerAvg1":ttnPower3,
            "EnergyAvg1":ttnEnergy3,
            "Power_FactorAvg1":ttnPf3
            }
pzem1R = {"Voltage1R":ttnVoltage3,
          "Current1R":ttnCurrent3,
          "Power1R":ttnPower3,
          "Energy1R":ttnEnergy3,
          "Power_Factor1R":ttnPf3
          }
pzem1S = {"Voltage1S":ttnVoltage3,
          "Current1S":ttnCurrent3,
          "Power1S":ttnPower3,
          "Energy1S":ttnEnergy3,
          "Power_Factor1S":ttnPf3
          }
pzem1T = {"Voltage1T":ttnVoltage3,
          "Current1T":ttnCurrent3,
          "Power1T":ttnPower3,
          "Energy1T":ttnEnergy3,
          "Power_Factor1T":ttnPf3
          }
pzemAvg2 = {"VoltageAvg2":ttnVoltage3,
            "CurrentAvg2":ttnCurrent3,
            "PowerAvg2":ttnPower3,
            "EnergyAvg2":ttnEnergy3,
            "Power_FactorAvg2":ttnPf3
            }
pzem2R = {"Voltage2R":ttnVoltage3,
          "Current2R":ttnCurrent3,
          "Power2R":ttnPower3,
          "Energy2R":ttnEnergy3,
          "Power_Factor2R":ttnPf3
          }
pzem2S = {"Voltage2S":ttnVoltage3,
          "Current2S":ttnCurrent3,
          "Power2S":ttnPower3,
          "Energy2S":ttnEnergy3,
          "Power_Factor2S":ttnPf3
          }
pzem2T = {"Voltage2T":ttnVoltage3,
          "Current2T":ttnCurrent3,
          "Power2T":ttnPower3,

```

```

        "Energy2T":ttnEnergy3,
        "Power_Factor2T":ttnPf3
    }
    pzem3 = {"Voltage3":ttnVoltage3,
            "Current3":ttnCurrent3,
            "Power3":ttnPower3,
            "Energy3":ttnEnergy3,
            "Power_Factor3":ttnPf3
    }
    pzem4 = {"Voltage4":ttnVoltage3,
            "Current4":ttnCurrent3,
            "Power4":ttnPower3,
            "Energy4":ttnEnergy3,
            "Power_Factor4":ttnPf3
    }
    pzemEnergy = {
        "EnergyTotal":ttnEnergy3
    }

    db.child("Sensor_PZEM_Avg1").set(pzemAvg1)
    db.child("Sensor_PZEM1_R").set(pzem1R)
    db.child("Sensor_PZEM1_S").set(pzem1S)
    db.child("Sensor_PZEM1_T").set(pzem1T)
    db.child("Sensor_PZEM_Avg2").set(pzemAvg2)
    db.child("Sensor_PZEM2_R").set(pzem2R)
    db.child("Sensor_PZEM2_S").set(pzem2S)
    db.child("Sensor_PZEM2_T").set(pzem2T)
    db.child("Sensor_PZEM3").set(pzem3)
    db.child("Sensor_PZEM4").set(pzem4)
    db.child("Energy_Total").set(pzemEnergy)

    valRelay1 = db.child("Control_Relay1").get().val()
    valRelay2 = db.child("Control_Relay2").get().val()
    valRelay3 = db.child("Control_Relay3").get().val()
    valRelay4 = db.child("Control_Relay4").get().val()

    print("Send to firebase...")
    print("Read Value Relay 1: "+ str(valRelay1.get("valRelay1")))
    print("Read Value Relay 2: "+ str(valRelay2.get("valRelay2")))
    print("Read Value Relay 3: "+ str(valRelay3.get("valRelay3")))
    print("Read Value Relay 4: "+ str(valRelay4.get("valRelay4")))
    print("-----")
    print("")

    while ( (start+howlong) > millis() ):
        pass

def millis2(howlong):
    start = millis()

    sPzem3_ref.add({
        "Voltage3":0,
        "Current3":0,
        "Power3":0,
        "Energy3":0,
        "Power_Factor3":0,
        "timestamp": firestore.SERVER_TIMESTAMP
    })

```

```
pzem3 = {"Voltage3":0,  
         "Current3":0,  
         "Power3":0,  
         "Energy3":0,  
         "Power_Factor3":0  
        }
```

```
db.child("Sensor_PZEM3").set(pzem3)
```

```
while ( (start+howlong) > millis() ):  
    pass
```

```
while True:  
    millis1(2000)  
    #millis2(2000)
```