

# Create a Python App to stream Modbus protocol in local

## Nasıl çalışır?

Uygulamanın iki parçası var.

1. **SlaveMachine.py** içinde tanımlanan Server. Bu server bizim ana makinemiz olmuş oluyor (Slave). Bu içinde tanımlanan Data block'larında stream edilen data'yı saklıyor. DB gibi düşünülebilir, ancak, bir yandan TCP'den gelen request'leri de kabul edip onlara datayı geri modbus ile yolluyor.
2. **UpdateServer.py** içinde tanımlanan Server'ı update eden coroutine. Bu script'teki kod da sürekli olarak Server'a bağlanıp gerekli coil'deki değeri yazıyor. Bizim implementasyonda "0" coil'ini kullandım.

### SlaveMachine.py



Generic File

Bu kodun çalışması için python2 gerekiyor. Ben 2.7'de denedim.. Bu kod için gerekli dependency'ler pip'den yüklenebilir. Python3'de sorun yaşadım.

```
sudo pip2 install pymodbus
sudo pip2 install cryptography
```

Benim bilgisayarımda 502 port'una bağlanırken access vermedi, sudo ile bağlanmasına izin verdi. 502 dışındaki herşeye bağlanabiliyor olmasına rağmen. Neyse, çalıştırmak için:

```
sudo python2 SlaveMachine.py
```

Bunu bi terminal'de çalışır halde bırakıyoruz.

Şu anda aslında PredixMachine'in buna bağlanabiliyor olması gerekir. Gerekli bilgiler şunlar:

```
VendorName = 'TTC'
ProductCode = 'PredixPark'
ProductName = 'PredixPark'
ModelName = 'PredixPark'
MajorMinorRevision = '1.0'
```

```
DataCoil = 0 # data boyutu 1-bit. # 1=Önünde obje var, 0=Yok
```

### UpdateServer.py



Generic File

Bu kod Betül'ün sensör'den okuma ve server'daki data coil'ini update etme kısmını içeriyor. Bu da infinite-loop şeklinde yazıldı. Sürekli olarak server'daki data güncellenecek, PredixMachine ne zaman okursa artık o zamanki datayı alacak.

Burada RPi diye bir library kullanılıyor. Bu library'nin python2 versiyonunu bulamadım. Neyse ki buradaki pymodbus kodu python3'de arıza çıkarmıyor. Burada pymodbus3 yüklü olmalı.

```
sudo pip3 install pymodbus3
sudo pip3 install cryptography
```

Bunu da bi terminalde çalıştırıp bırakıyoruz.

```
sudo python3 UpdateServer.py
```

**Önemli Not:** Ben bu script'i test edemedim (Raspberry PI kodu içeriyor). Lâkin, çalışıyor olduğunu tahmin ediyorum gözle kontrol ettim. Sistemin çalışıp çalışmadığını kontrol etmek için (bkz: Test)

## Test:

Elimde raspberry olmadığı için Raspberry datası aktarmak yerine, kendim bi routine yazıp saniyede bir 1-0 değerleri yükleyen bir script yazdım:

```
import time
from pymodbus3.client.sync import
ModbusTcpClient
client =
ModbusTcpClient('127.0.0.1')

while True:
    client.write_coil(0, True)
    time.sleep(1.0)
    client.write_coil(0, False)
    time.sleep(1.0)
```



Generic File

Run with:

```
python3 UpdateServerTest.py
```

Ve daha sonra bu datayı da okuyup doğru output alıp almadığımı görmem gerekiyor. Onun için de şunu yazdım.:

```
from pymodbus.client.sync import
ModbusTcpClient

client =
ModbusTcpClient('127.0.0.1')
result = client.read_coils(0,1)
```



Generic File

```
python2 UpdateServerTest.py
```

```
print result.bits[0]
client.close()
```

Bunu diğer iki routine çalışırken ard arda çalıştırdığım zaman

```
Onats-MacBook-Pro:Desktop onatbas$ python2 ReadCoil.py
False
Onats-MacBook-Pro:Desktop onatbas$ python2 ReadCoil.py
True
Onats-MacBook-Pro:Desktop onatbas$ python2 ReadCoil.py
True
Onats-MacBook-Pro:Desktop onatbas$ python2 ReadCoil.py
False
Onats-MacBook-Pro:Desktop onatbas$ python2 ReadCoil.py
False
Onats-MacBook-Pro:Desktop onatbas$ python2 ReadCoil.py
True
```

**\*\*END\*\***

---

---

---

*Onat's previous notes:*

*Betul's code to get binary info from sensors.*



Generic File



Generic File

*Use this to stream this information via modbus over tcp :*

<https://github.com/bashwork/pymodbus>