

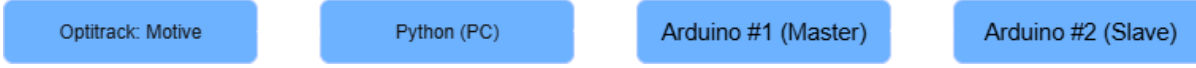
OptiTrack ile Gerçek Zamanlı Konum Paylaşımı Kılavuzu

Sistem Mimarisi

Bu sistem, OptiTrack hareket izleme sisteminden alınan gerçek zamanlı konum verilerinin bir Arduino aracılığıyla Bluetooth modülleri üzerinden başka bir Arduino'ya aktarılmasını sağlar.

Veri Akış Diyagramı:

[OptiTrack] → [Python (PC)] → [Arduino #1 (Master)] → (HC-05) → (HC-05) → [Arduino #2 (Slave)]



Gerekli Donanım ve Yazılım

Donanım:

- 2 adet Arduino UNO / Nano
- 2 adet HC-05 Bluetooth modülü (1 HC05 - 1 HC06 şeklinde de olabilir)

Yazılım:

- Motive (OptiTrack yazılımı - versiyon 1.5.0)
- Python 3.x
- Arduino IDE

Python Tarafı: OptiTrack + Veri Gönderimi

OptiTrack sistemi, gerçek zamanlı pozisyon ve rotasyon verilerini dış uygulamalara aktarabilmek için resmi olarak geliştirdiği **NatNet SDK**'yı sunar. Bu SDK, OptiTrack'ın "Motive" yazılımı üzerinden yayınlanan verileri, TCP/UDP bağlantısı üzerinden almak için kullanılabilir.

NatNet SDK Nedir?

- **NatNet**, OptiTrack tarafından sağlanan ağ tabanlı (network socket) veri iletim protokolüdür.
 - **SDK** (Software Development Kit), Windows için C++ ve C# destekler.
 - Python desteği doğrudan yoktur, fakat community destekli Python wrapper kullanılabilir. Ben de verileri almak için Github'dan bulduğum bir wrapper kullandım.
-

Python Wrapper Nedir?

Python, NatNet SDK'yı doğrudan kullanamaz. Bu nedenle community veya araştırma kurumları tarafından geliştirilmiş bazı **Python wrapper'ları** kullanılabilir. Bu projeler SDK'nın DLL'lerini çağırarak Python tarafında veri alınmasını sağlar.

Biz de bu amaçla GitHub üzerinden bulduğumuz bir Python wrapper projesini kullanarak, Motive'den gelen verileri Python tarafına aktarabiliyoruz.

🔗 **Kullandığımız GitHub Reposu:**

<https://github.com/TimSchneider42/python-natnet-client/tree/master>

Bu Wrapper Nasıl kullanılır?

Bu wrapper ile Motive yazılımından gelen veriler Python tarafından okunup seri port aracılığıyla Arduino'ya aktarılabilir. Bu repodan aldığım gerekli dosyaları motive_arduino_data_streaming (ana dosya) içine yükledim zaten. Ama farklı problemler olursa ve incelemek isterseniz diye referans bıraktım.

Bu wrapperı kullanarak iki dosya yazdım, veri iletimini başlatmak için bu dosyalardan birini çalıştıracamız, ikisinin farkı şöyle:

1. **data_stream:** Bu programda rigid body verilerini real time gönderen bir kod var, içinde farklı amaçlara hizmet eden fonksiyonlar ve açıklamaları var. İleride optitrackten farklı tür veriler de almak isterseniz buna bakabilirsiniz veya diğer natnet_client içindeki dosyaları inceleyebilirsiniz.
2. **data_stream_basic:** Bu program sadece sahnede tanımlı tek bir rigid body'nin konum verisini (x,y,z) tuple formatında alıp 0.05 saniyede bir kez arduinoya serial port üzerinden gönderiyor. Sahnede birden fazla rigid body tanımlı olursa sadece birinin verilerini gönderecektir, bunu isterseniz data_stream dosyasını inceleyerek ihtiyacınıza göre nasıl manipüle edebileceğinizi anlayabilirsiniz.

Normalde veri iletimini başlatmak için bu iki dosyadan birini (hangisine ihtiyacınız varsa) çalıştırmanız yeterli, fakat bu programın çalışması için Motive uygulamasında da belirli şeylerin doğru ayarlanmış olması gerekiyor:

Motive Ayarları:

- Motive açık ve live modda çalışıyor olmalı
- **Data Streaming** özelliği açık olmalı, **View → Data Streaming → Broadcast Frame Data** tiklenmiş olmalı
- Yayın tipi olarak **Unicast** etkin olmalı, **Network Options → Type → Unicast** seçilmeli (Yoksa çalışmayacaktır)
- Kalibrasyon dosyası yüklenmelidir (.CAL uzantılı), kameralar sahnede yan yana görünüyorsa kalibrasyon açılmamış demektir. Kalibrasyon dosyası açıldığında kameraların konumu sahnede doğru görünecektir. Pencerenin üst kısmındaki tool bar kısmından direk dosya açma ikonuna tıklayarak var olan bir kalibrasyon dosyası yükleyebilirsiniz. En son benim yaptığım kalibrasyon dosyasını açabilirsiniz mesela, ayse cal 1 dosyası içerisindeki Calibration Very High Quality Exceptional adlı .CAL uzantılı dosyayı kullanabilirsiniz.
- Takip edilecek cismin markerları seçilip sağ tık -> Rigid Body -> Create From Selected Markers yaparak cisme rigid body tanımlanmalıdır. data_stream_basic kodu default olarak tek rigid body konumunu gönderecektir, dolayısıyla sahnede tek rigid body tanımlanmış olmalı. Daha fazla göndermek için data_stream dosyasını inceleyebilirsiniz. Bu dosyada pozisyon harici verilerin de nasıl alınabileceğini görebilirsiniz.
- 2D, 3D ve JT seçenekleri aktif olmalıdır. (JT'yi rigid body tanımladıktan sonra açabiliyorsunuz sadece)
- Data Streaming penceresindeki diğer değerler genelde doğru oluyor zaten fakat herhangi bir hata durumunda yardımcı olabilir diye en son çalışan durumdaki değerleri yazıyorum:
 - Stream markers -> True
 - Stream rigid bodies -> True (Kesinlikle böyle olmalı)
 - Type -> Unicast (kesinlikle öyle olmalı, yoksa çalışmaz)
 - Command port -> 1510
 - Data Port -> 1511
 - Local Interface -> Preferred
 - Multicast Interface -> 239.255.42.99

Python Ortamı:

- `pyserial` ve wrapper bağımlılıkları yüklü olmalı (örn: `NatNetClient`), var olan dosya yollarını değiştirmek sıkıntı yaratır. (Zaten bu dosyalar ana dosya içinde var, `data_stream` veya `data_stream_basic` haricindeki diğer dosyalarla uğraşmanıza gerek yok fakat diğer dosyaların yolu değişmemeli veya silinmemeliler)
- Master bluetooth modülünü takacağımız arduino bilgisayara bağlı olmalı ve `data_stream.py` (veya basic olan) içindeki `serial.Serial("COMX", 9600)` satırında doğru COM portu belirtilmiş olmalı, arduino tak çıkar yapıldığında bazen bu COM numarası değişebiliyor, o zaman hata veriyor. COM numarasının doğru olmasına her zaman dikkat edilmeli. Bununla ilgili hata olursa önce arduinoyu tak çıkar yapmayı deneyebilirsiniz ama olmazsa Aygıt Yöneticisinden arduinonun COM portuna bakıp bu kısmı öyle değiştirebilirsiniz. Örneğin Arduino com portu 7 ise o kod şöyle olmalı, `ser=serial.Serial("COM7", 9600)`

Bunlar tamamlandığında motive tarafında daha bir şey yapmamıza gerek olmuyor. Python kodunu çalıştırdığınızda (`data_stream` veya `data_stream_basic` dosyasını) motiveden gelen konum verileri arduinonun serial portuna gönderilmeye başlayacaktır.

Aktarılan veri türünü ve gönderim zamanlamasını özelleştirmek isterseniz, program dosyası içindeki açıklamaları takip ederek ilgili bölümleri ihtiyaçlarınıza göre düzenleyebilirsiniz.

Herhangi farklı bir hata durumunda, ana dosyada tanımlı bağımlılıkları kontrol ederek eksik veya hatalı kurulumları gözden geçirmeniz önerilir.

HC-05 Modüllerinin Yapılandırması

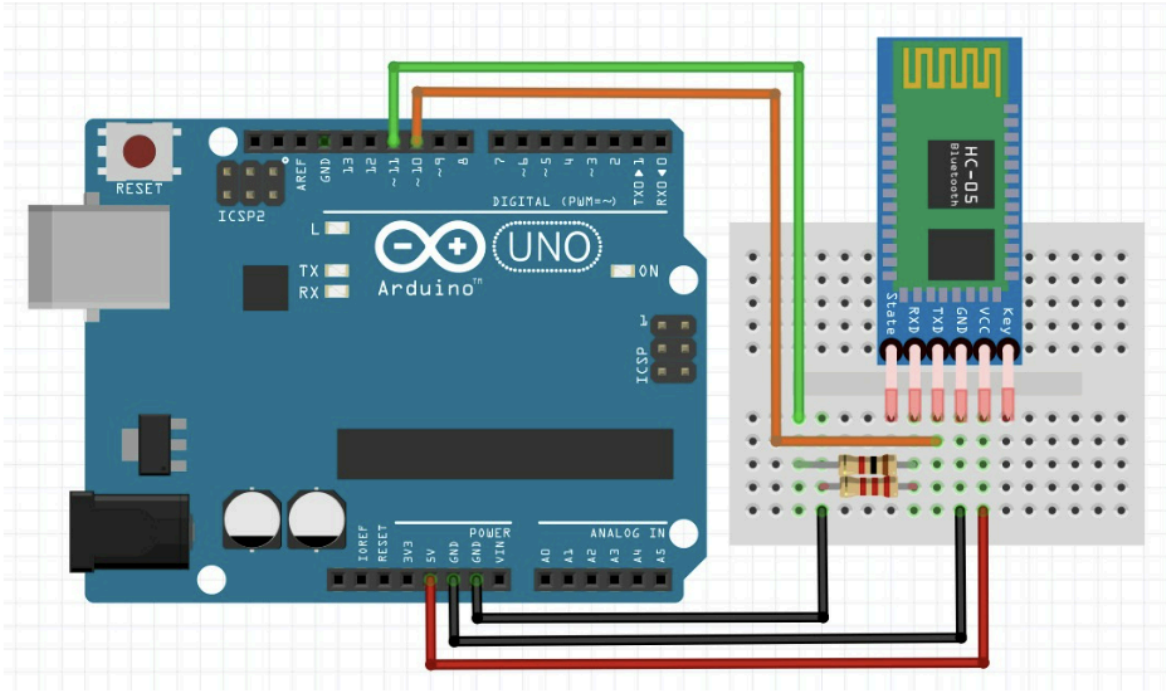
Bluetooth ile veri aktarımı için bir modül **master**, diğeri **slave** olarak ayarlanmalıdır. Burada iki modülü AT moduna alıp bir kez eşleyeceğiz ve daha sonra modüller her açıldığında birbirlerine bağlanacaklar. Yani aşağıdakileri bir kez yapacağız sadece. Benim en son kullandığım modüller zaten eşlenmiş oldukları için bunları onlara tekrar yapmanıza gerek yok, onlar direk birbirlerine bağlanacaktır. Ancak ben başka modüller kullanmak isterseniz diye bu kısmı hazırladım.

Şimdi arduinoya aldığımız verileri okuyup master bluetooth modülü üzerinden slave modüle göndereceğiz ve slave modülün olduğu arduinodan bu verileri okuyacağız.

İlk olarak iki modülü de devreye aynı şekilde bağlayabiliriz.

Devre bağlanma şeması:

DEVRE BAĞLANTI ŞEMASI:



Burada kodumuzu kullandığımız pin numarasına göre yazmalıyız. Ben modülü 2 ve 3 numaralı pinlere bağladığım için kodumu ona göre yazmışım, buna dikkat edilmeli.

Elimizde iki hc05 modülü var, bunları birbirine eşlemek için iki modülü de AT moduna alıp ayarlar yapmamız gerek. İkisi de aynı şekilde AT moduna alınıyor fakat AT moduna geçince vereceğimiz komutlar farklı olacak. İlk önce AT moduna almayı görelim:

1. AT moduna alma:

- HC-05 modülün butonuna basılı tutarken arduinoya güç veriyoruz ve bırakıyoruz, modülün ışığı 2 saniyede bir yanmaya başladıysa AT moduna geçtik demektir.
2. AT komutlarını çalıştırabilmek için Arduino'ya aşağıdaki gibi bir AT komut terminali yüklemiş olmalısın:

Örnek Kod (AT komutları için terminal):

```
#include <SoftwareSerial.h>

SoftwareSerial BT(3, 2); // RX, TX,

void setup() {

  Serial.begin(9600);      // Arduino serial monitor için

  BT.begin(38400);        // HC-05 AT modu varsayılan baudrate

  Serial.println("AT komutu bekleniyor...");
```

```

}

void loop() {

    if (Serial.available()) {

        BT.write(Serial.read());

    }

    if (BT.available()) {

        Serial.write(BT.read());

    }

}

```

Bu kodu yükledikten sonra AT modunu test etmek için seriale AT yazabiliriz, OK diye cevap dönerse sıkıntı yok demektir. Şimdi AT komutları ile iki modülü ayrı ayrı yapılandıralım. Burada bilgisayara bağlı olan (yani veri gönderen taraf) master olarak ayarlanmalı, diğeri (veri alacak olan) slave olarak ayarlanmalı. Slaveden başlamalıyız:

Slave Modül İçin AT Komutları:

- **AT** → Cevap: **OK**
- **AT+ROLE=0** → Slave olarak ayarlamak için
- **AT+NAME=SLAVE** → Modül ismini değiştir
- **AT+UART=38400, 0, 0** → UART ayarı, bu UART değeri diğer master modülü ile aynı olmak zorunda, ben ikisini de 38400 yaptım
- **AT+ADDR?** → Slave Bluetooth adresini gösterir (bunu kaydedin), daha sonra master modülü ayarlarken kullanacağız
- **AT+PSWD=1234** → İki modülün şifresi aynı olmalı, ikisini de 1234 yapıyoruz.

Master İçin AT Komutları:

3. **AT+ROLE=1** → Master olarak ayarla
4. **AT+NAME=MASTER** → Modül adını master yapalım
5. **AT+UART=38400, 0, 0** → İki modülün de UART değeri aynı olmalı

6. `AT+CMODE=0` → Sadece belirli bir cihaza bağlansın, yani sadece slave modüle bağlanması için bunu yapıyoruz
7. `AT+BIND=98D3,31,FB3421` → Slave adresi girilir, slave modülü mastera tanıtmak için
8. `AT+LINK=98D3,31,FB3421` → Slave ile bağlanmak için
9. `AT+PSWD=1234` → İki modülün şifresi aynı olmalı, ikisini de 1234 yapıyoruz.

Her komuttan sonra terminalden OK cevabı almalıyız. Almıyorsa bir hata var demektir. Bu durumda şunları kontrol edebilirsiniz:

- Modül ışığı 2 saniyede bir yanıp sönmeli.
- Arduinoya AT komutları için olan kod yüklenmiş olmalı.
- Serial monitörde Both NL & CR seçili olmalı ve baud rate de `serial.begin(x)`'deki x ile aynı olmalı.

Bu komutları yazıp OK yanıtını aldıktan sonra iki arduinoya da tekrar güç vererek modülleri normal moda geçirin. Biraz bekledikten sonra iki modül de 5 saniyede iki kez yanıp sönüyorsa bu bağlandıkları anlamına gelir. Bu modülleri her seferinde tekrar yapılandırmamıza gerek yok, tekrar takıp çıkarsak da birbirlerine bağlanacaklardır.

Şimdi arduinoya bluetooth üzerinden veri alışverişi yapması için gereken kodu yazabiliriz.

Modüller eşlendikten sonra master modülün takılı olduğu arduinoya bu kodu yükleyelim.

```
#include <SoftwareSerial.h>
```

```
const byte rxpin = 3;  
const byte txpin = 2;  
SoftwareSerial BTSerial(rxpin,txpin); // rx/tx
```

```
void setup(){  
  pinMode(rxpin, INPUT);  
  pinMode(txpin, OUTPUT);  
  Serial.begin(9600);  
  BTSerial.begin(38400); // Buradaki değer ayarladığımız AT+UART=... değeri ile aynı olmalı  
}
```

```
String line="";
```

```
void loop(){
```

```
  if(BTSerial.available())  
  {  
    Serial.write(BTSerial.read());  
  }
```

```
  if(Serial.available())
```

```
{
```

```
String data=Serial.readStringUntil('\n');
BTSerial.println(data);
}
}
```

Bluetooth ile slave modüle gelen veriyi serial porta yazdırmak için Slave modülüne de şu kodu yükleyebiliriz:

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(3, 2); // RX, TX

void setup() {
  Serial.begin(9600);
  BTSerial.begin(38400); // HC-05 genelde 9600 baud
  Serial.println("Hazır");
}

void loop() {
  if (BTSerial.available()) {
    String gelen = BTSerial.readStringUntil('\n');
    Serial.print("Slave aldı: ");
    Serial.println(gelen);
  }
}
```

Bunları yaptıktan sonra data_stream_basic dosyamızı çalıştırdığımızda sahnedeki rigid body tanımlı nesnenin pozisyon verisini her 0.05 saniyede bir serial monitor üzerinden görebileceğiz. Burada biz sadece veriyi string olarak okuyup basıyoruz, veri pozisyon verisi içeren bir tuple şeklinde olduğu için siz o veriyi manipüle edip kullanabilirsiniz. Veri formatı: (x,y,z) , 6 precision, 0.05 saniyede bir (daha hızlı olunca veri gönderimi bozulmaya başlıyor, ama belki o sorun çözülürse daha hızlı veri aktarılabilir.)

Test ve Sorun Giderme

- Python'da veri alıp almadığınızı print fonksiyonu ile test edebilirsiniz, eğer veri python'da alınamıyorsa ve dosya çalışınca hata da göstermiyorsa muhtemelen motivedeki ayarlarda sorun vardır, onları kontrol edebilirsiniz.
- Eğer python dosyasını çalıştırınca hata veriyorsa COM portu kontrol edebilirsiniz, başka bir hata varsa python IDE'nin gösterdiği hata mesajına bakarak sorunu çözebilirsiniz.
- Eğer python'da veri alırken sıkıntı olmuyorsa, ama veri slave takılan arduinoda görünmüyorsa o zaman modüllerin eşlenip eşlenmediğini kontrol edebilirsiniz. Modüller eşlenmediyse kaynak bıraktığım kısımdan yardım alabilirsiniz.

- Modüller eşlendiği halde veri görünmüyorsa muhtemelen arduino kodunda bir sıkıntı vardır. Koddaki baud rate değerlerinin modüllerle uyumlu olmasına ve rx, tx pinlerinin doğru kullanılmasına dikkat edin.

9. Geliştirme Önerileri

- JSON parser eklemek (ArduinoJson kütüphanesi)
- CRC/hata kontrolü entegrasyonu
- Gelen veriye göre cihaz kontrolü (servo, motor, vb.)
- Dosyalardaki bazı print fonksiyonları deneme ve test amaçlı kullanılmaktadır, bunları sistem oturduktan sonra yorum satırına alabilirsiniz.

NOT: Bu sistemi bir HC05 (master) ve bir HC06 (slave) ile de yapabilirsiniz. HC06'da AT komutlarını çalıştıramazsınız ama yine de modülün bluetooth adresini modüle telefondan bağlanıp öğrenebilirsiniz. Burada modüllerin UART değerinin de aynı olması gerekiyor, HC06'nın UART değerini direk göremeseniz de deneyerek onu bulabilirsiniz. (9600,0,0 ya da 38400,0,0 olacaktır muhtemelen)

Herhangi bir sorun durumunda bakılabilecek kaynaklar:

<https://projedenemeleri.blogspot.com/2019/01/2-arduinoyu-bluetooth-ile-haberlestirmek.html>

<https://www.youtube.com/watch?v=hyME1osgr7s&t=7s>

<https://akademi.robolinkmarket.com/hc-05-bluetooth-modulu-at-komutlari/>

<https://github.com/TimSchneider42/python-natnet-client/tree/master>

Hazırlayan: Ayşe Atik

Tarih: 29.07.2025

Proje Adı: OptiTrack ile Gerçek Zamanlı Konum Paylaşımı