



Servo Kontrolü için Haberleşme Kanalı Tasarımı

Bengü BİLGİÇ & Selahaddin HONİ

İÇERİK

Kavramsal Tasarım

Giriş ve Tasarım Kriterleri

Basitleştirilmiş Model

Modülasyon Tekniği

Donanım/Yazılım

Proje Şeması

Teknik Tasarım

Fiziksel Katman

GNU Radio'ya Harici Kaynak Bağlamak

Data Link ve Network

Donanım Senkronizasyonu

Diğer

Demo

KAVRAMSAL TASARIM

Giriş

Uzaktan servo kontrolü kullanım alanları

Arama kurtarma, bomba imha, keşif robotları vb.

Tasarım Kriterleri

Gerçek zamanlı / Minimum Gecikme

Gürbüz(Robust)

Doğruluk

Uzak Mesafelerde çalışmaya uygunluk



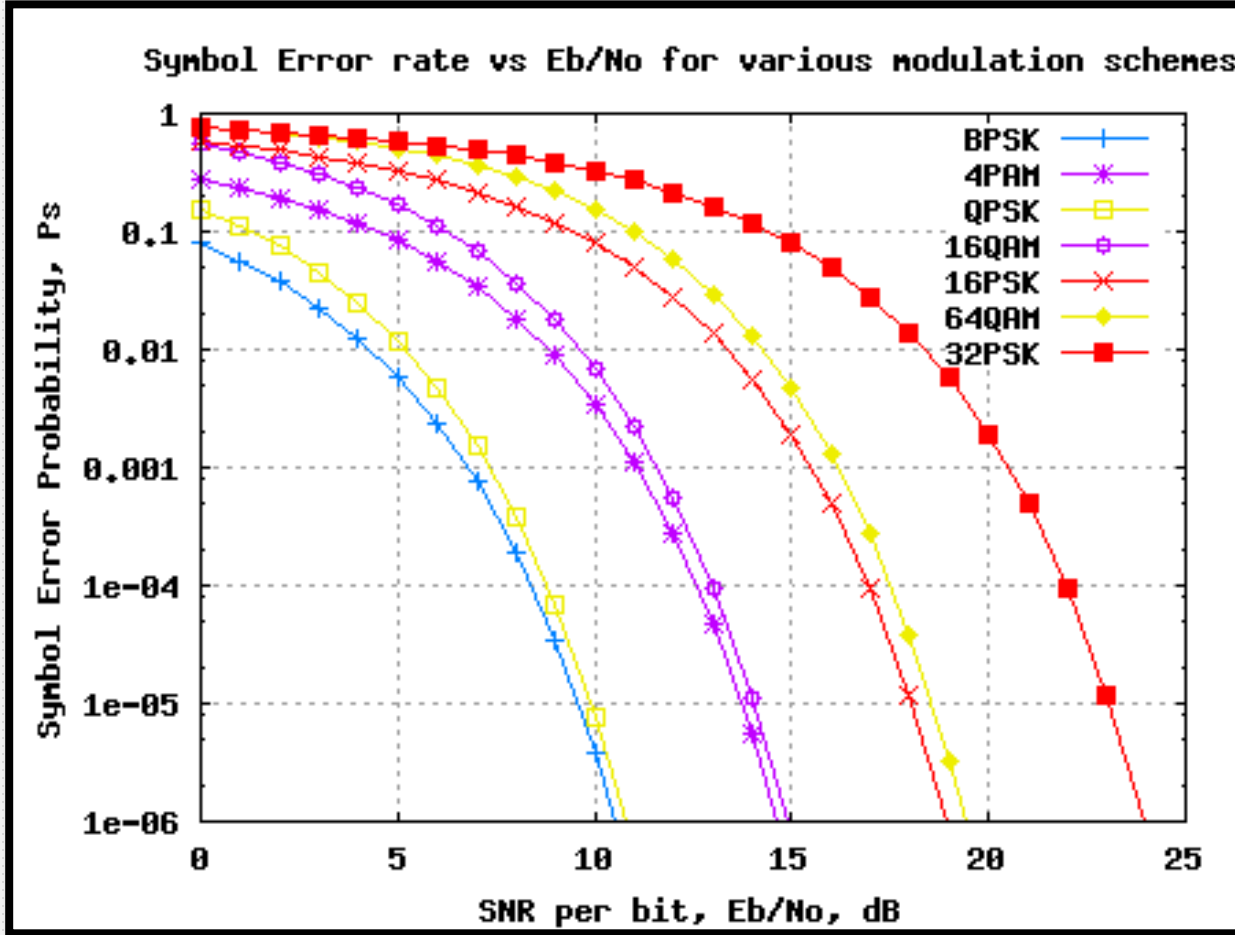
Basitleştirilmiş Model

Servo kontrolü için gerekli iki parametre

- *Pozisyon*
- *Hız*



Modülasyon Tekniği



4PAM – 16QAM

- Uzun mesafelerde zayıflama
- İletilecek veri küçük, gereksiz

BPSK – QPSK

- BER oranı düşük
- Sistem karmaşıklığı
- Dolaylı yoldan gecikme az

BPSK ✓

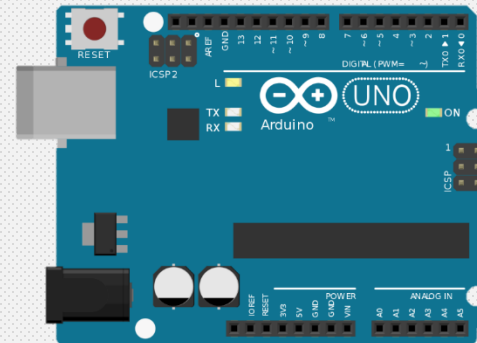
Donanım / Yazılım

	HackRF One	Ettus B200	Ettus B210	BladeRF x40	RTL-SDR	LimeSDR	LimeSDR Mini
Frequency Range	1 MHz - 6 GHz	70 MHz - 6 GHz	70 MHz - 6 GHz	300 MHz - 3.8 GHz	22 MHz - 2.2 GHz	100 kHz - 3.8 GHz	10 MHz - 3.5 GHz
RF Bandwidth	20 MHz	61.44 MHz	61.44 MHz	40 MHz	3.2 MHz	61.44 MHz	30.72 MHz
Sample Depth	8 bit	12 bit	12 bit	12 bit	8 bit	12 bit	12 bit
Sample Rate	20 MSPS	61.44 MSPS	61.44 MSPS	40 MSPS	3.2 MSPS	61.44 MSPS	30.72MSPS
TX Channels	1	1	2	1	0	2	1
RX Channels	1	1	2	1	1	2	1
Duplex	Half	Full	Full	Full	N/A	Full	Full
Interface	USB 2.0	USB 3.0	USB 3.0	USB 3.0	USB 2.0	USB 3.0	USB 3.0
Programmable Logic Gates	64 macrocell CPLD	75k	100k	40k (115k avail)	N/A	40k	16K
Chipset	MAX5864, MAX2837, RFFC5072	AD9364	AD9361	LMS6002M	RTL2832U	LMS7002M	LMS7002M
Open Source	Full	Schematic, Firmware	Schematic, Firmware	Schematic, Firmware	No	Full	Full
Oscillator Precision	+/- 20 ppm	+/- 2 ppm	+/- 2 ppm	+/- 1 ppm	?	+/-1 ppm initial, +/-4 ppm stable	+/- 1 ppm initial, +/- 4 ppm stable
Transmit Power	-10 dBm+ (15 dBm @ 2.4 GHz)	10 dBm+	10 dBm+	6 dBm	N/A	max 10 dBm (depending on freq.)	max 10 dBm (depending on freq.)
Price	\$299	\$686	\$1,119	\$420 (\$650)	~\$10	\$299	\$99

LimeSDR Hakkında
Windows Driver
PothosSDR Environment
GNU Radio 3.7 (Python 2.7)
*Linux : File Sink desteği



Arduino UNO
Yaygın geliştirme kartı
Sayısız doküman desteği



PROJE ŞEMASI

Verici Bilgisayar

Alıcı Bilgisayar

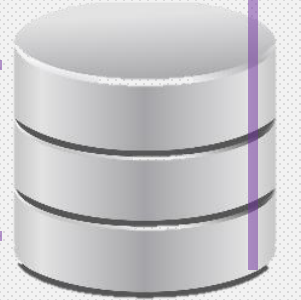
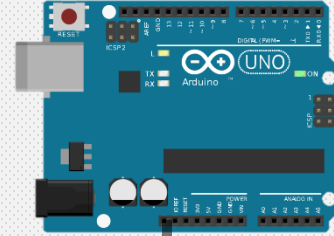
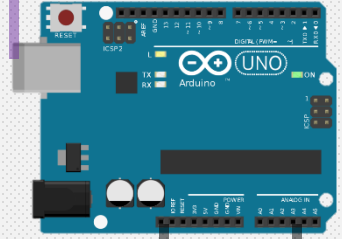
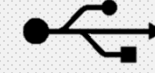
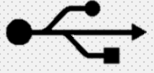
>
UDP

>
UDP

>
UDP

LimeSDR

LimeSDR



Potansiyometreler

Servo

GUI

Veritabanı

TEKNİK TASARIM

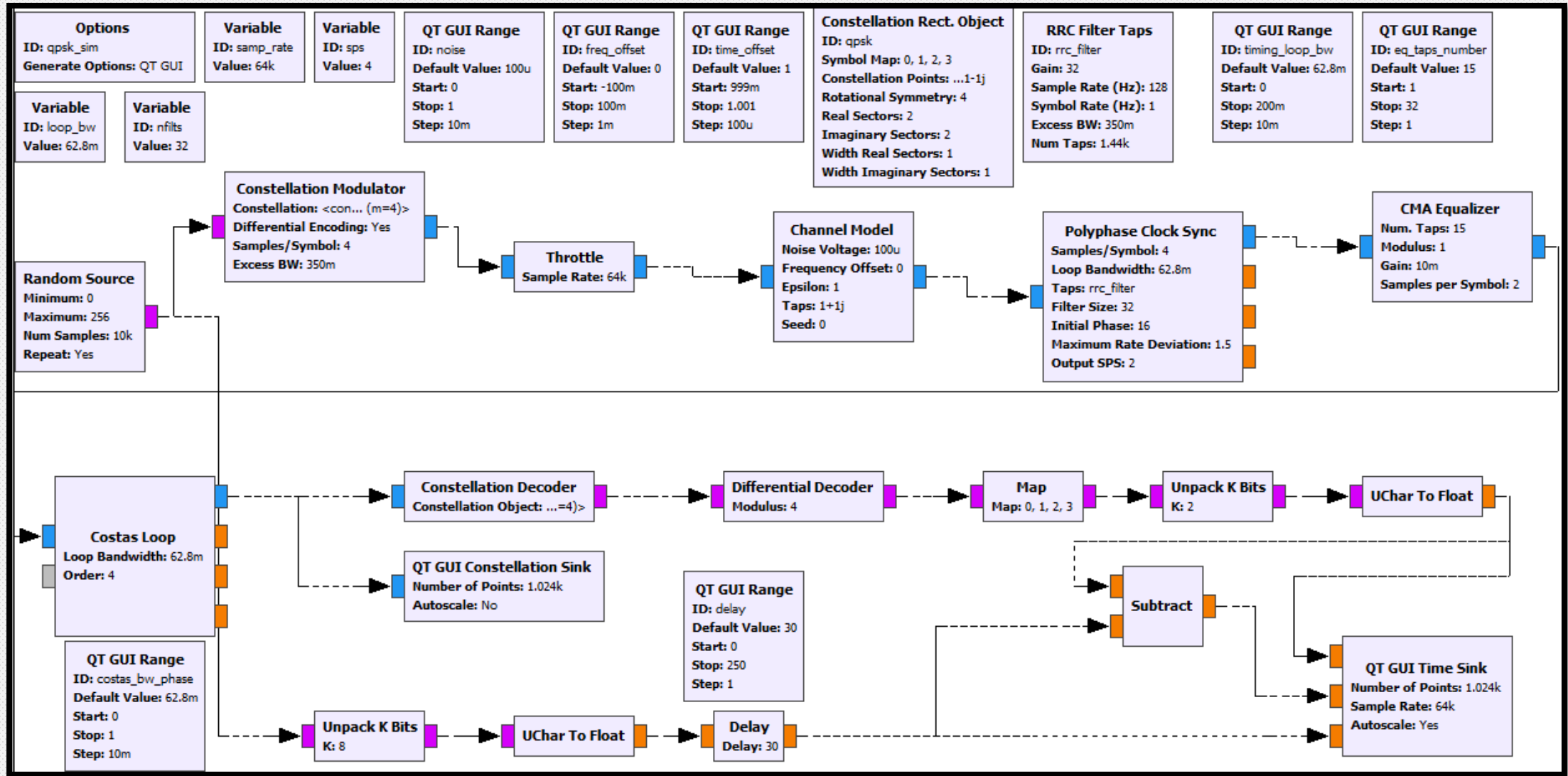
Fiziksel Katman Tasarımı için
GNU Radio Öğrenme Çalışmaları
AM Benzetimi

FM Ses Haberleşmesi
'446MHz Bilgisayarlar ve Telsiz arasında'

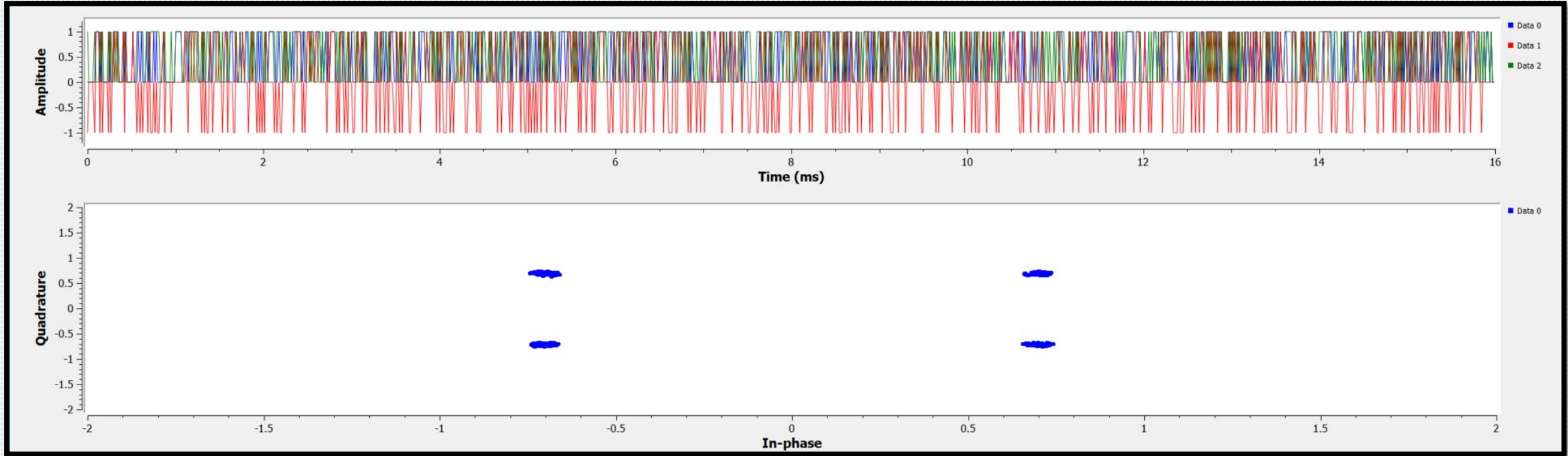
QPSK Benzetimi

BPSK ile metin iletimi

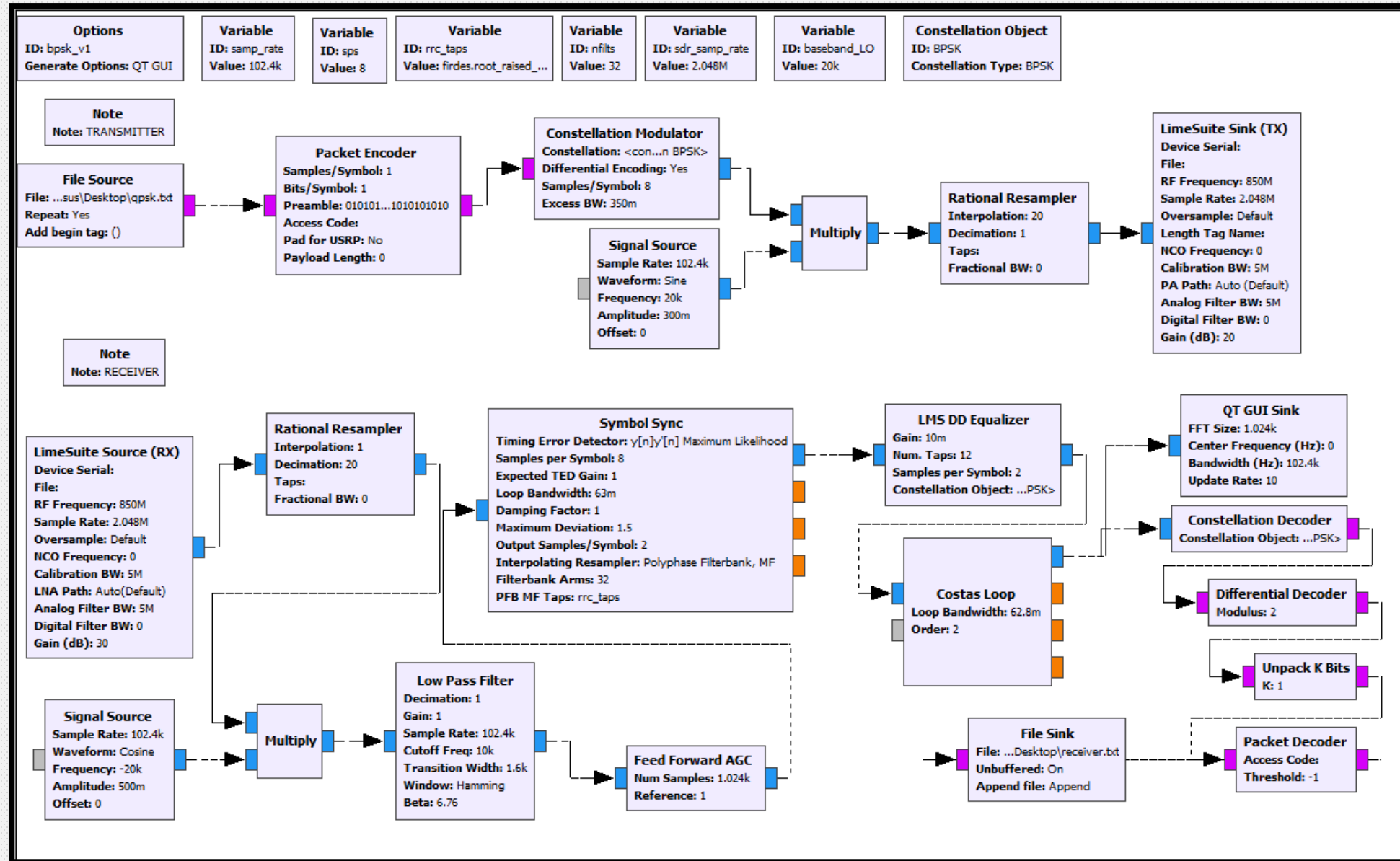
QPSK Benzetimi Blok Tasarımı



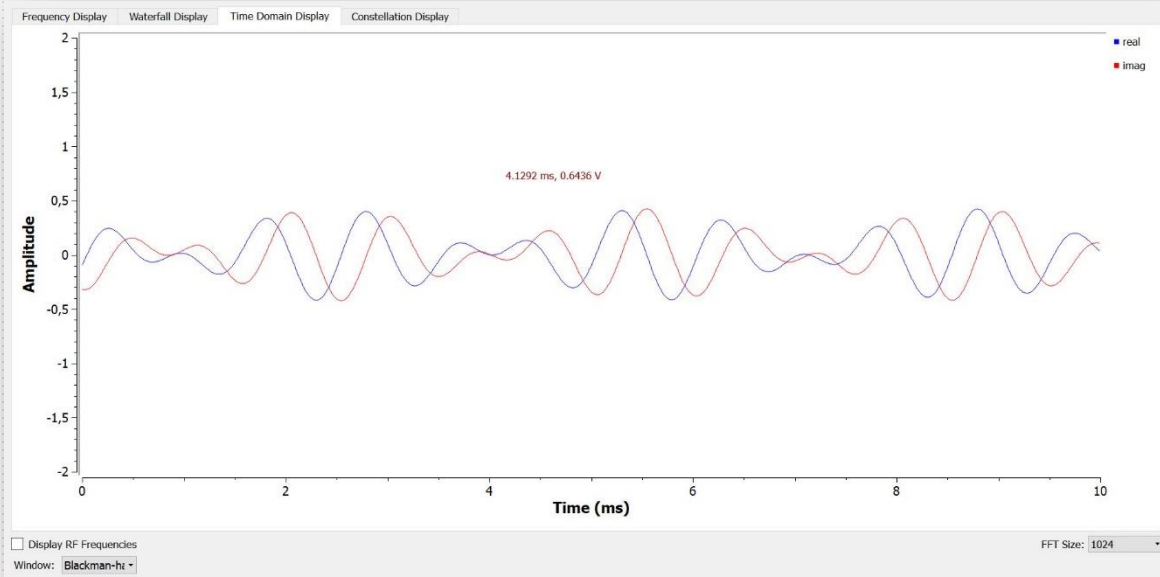
QPSK Benzetimi Sonuçları



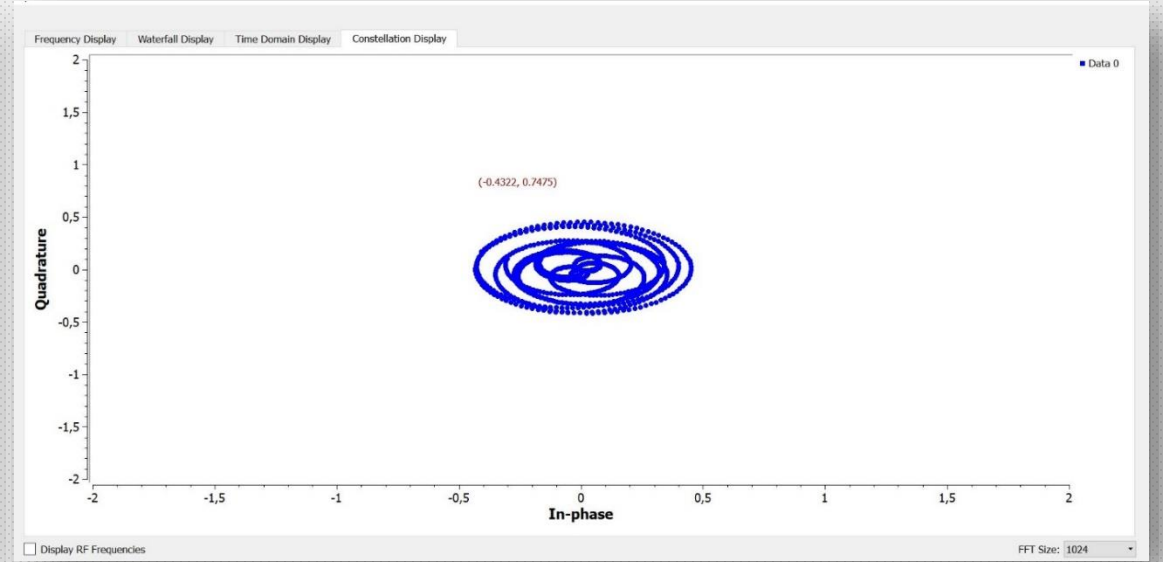
BPSK Metin İletimi Blok Tasarımı



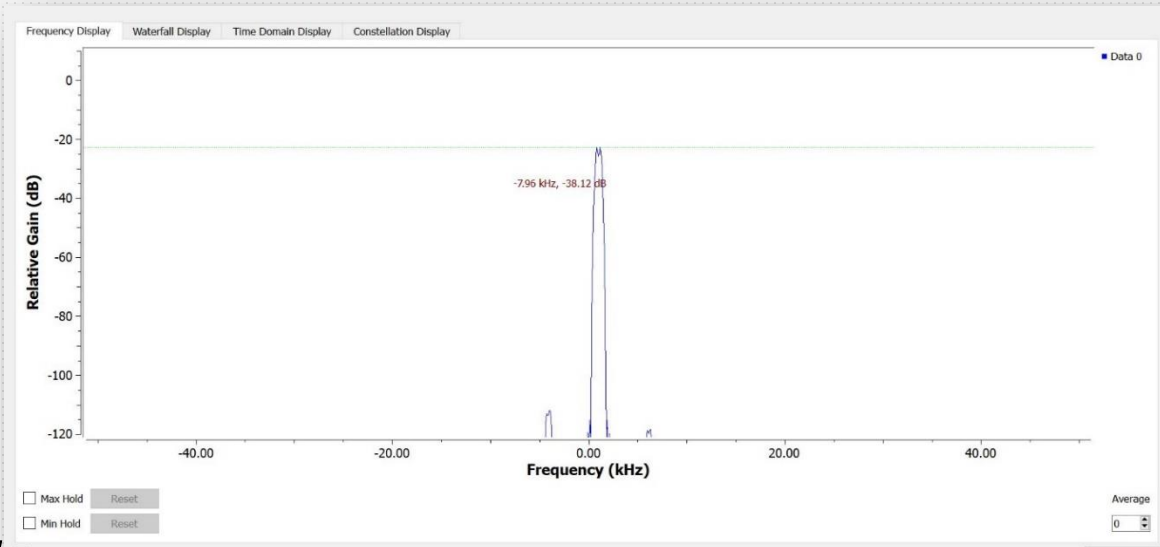
Verici Zaman Domeni



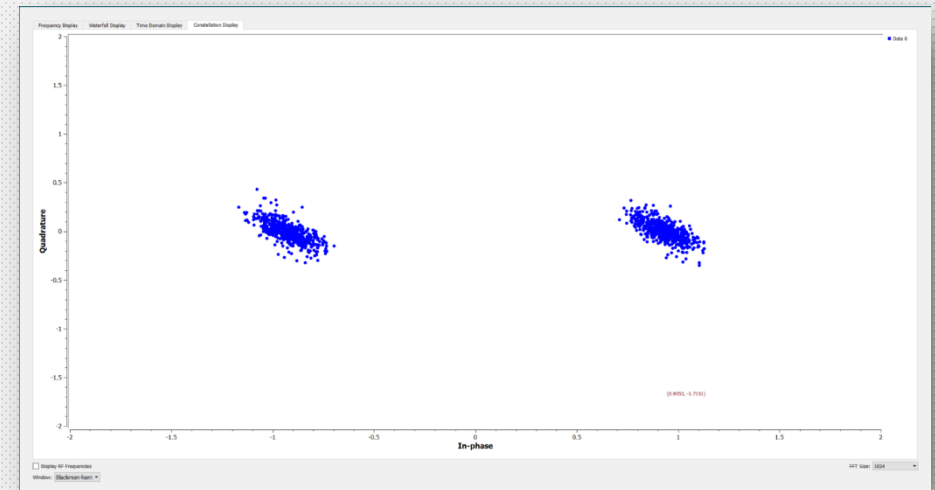
Verici I-Q Dizilimi



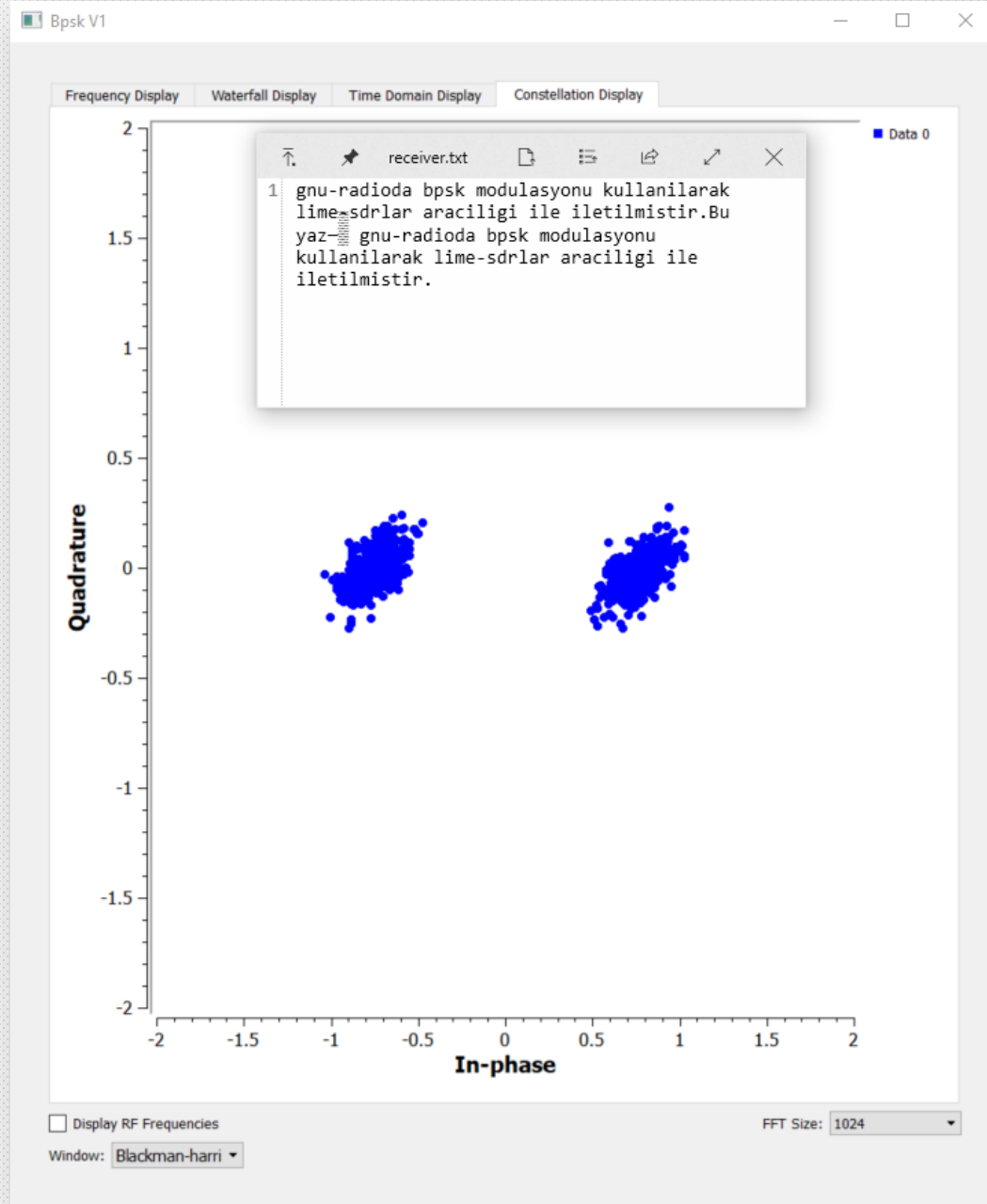
Verici Frekans Domeni



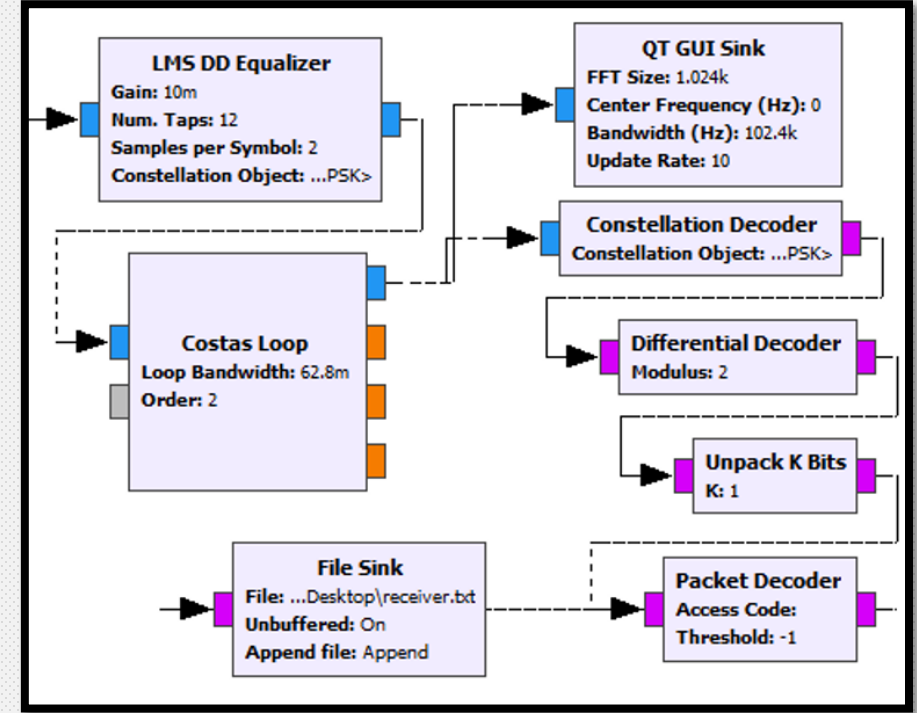
Alıcı I-Q Dizilimi



Alıcı Tarafa Ulaşan Metin

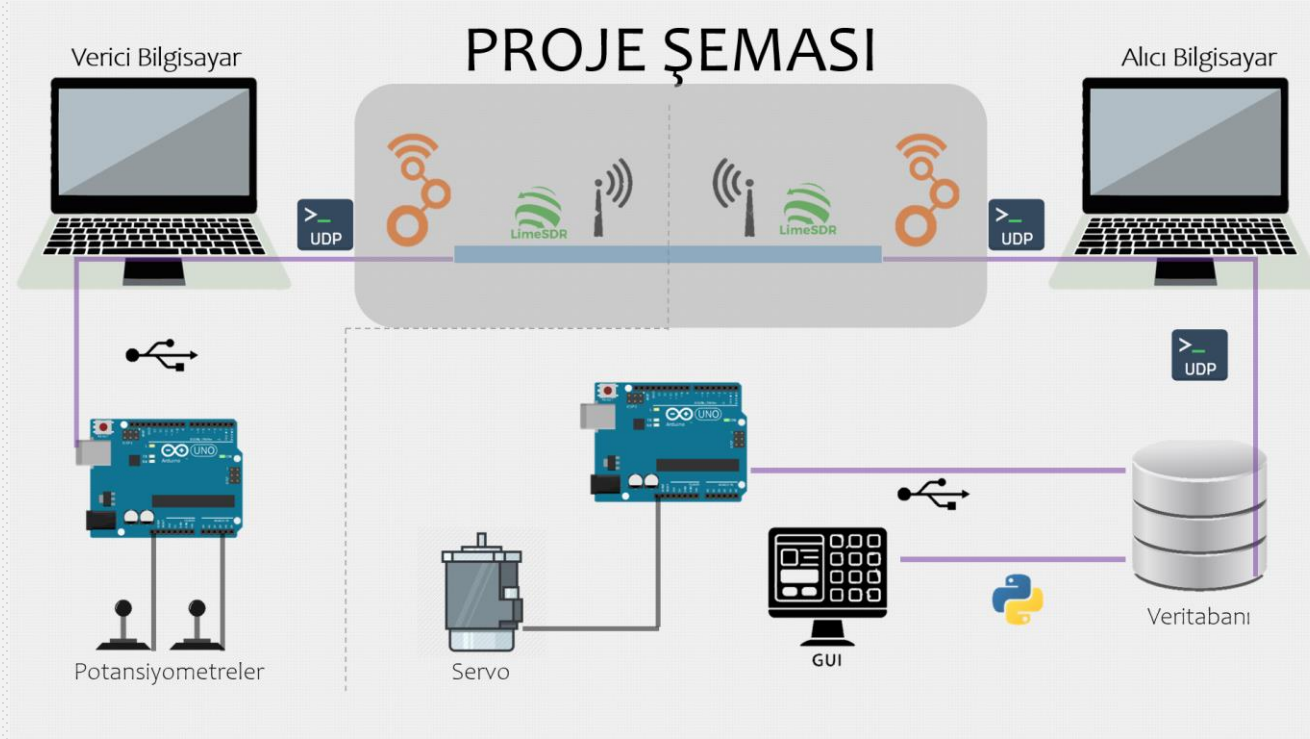


Hatırlatma

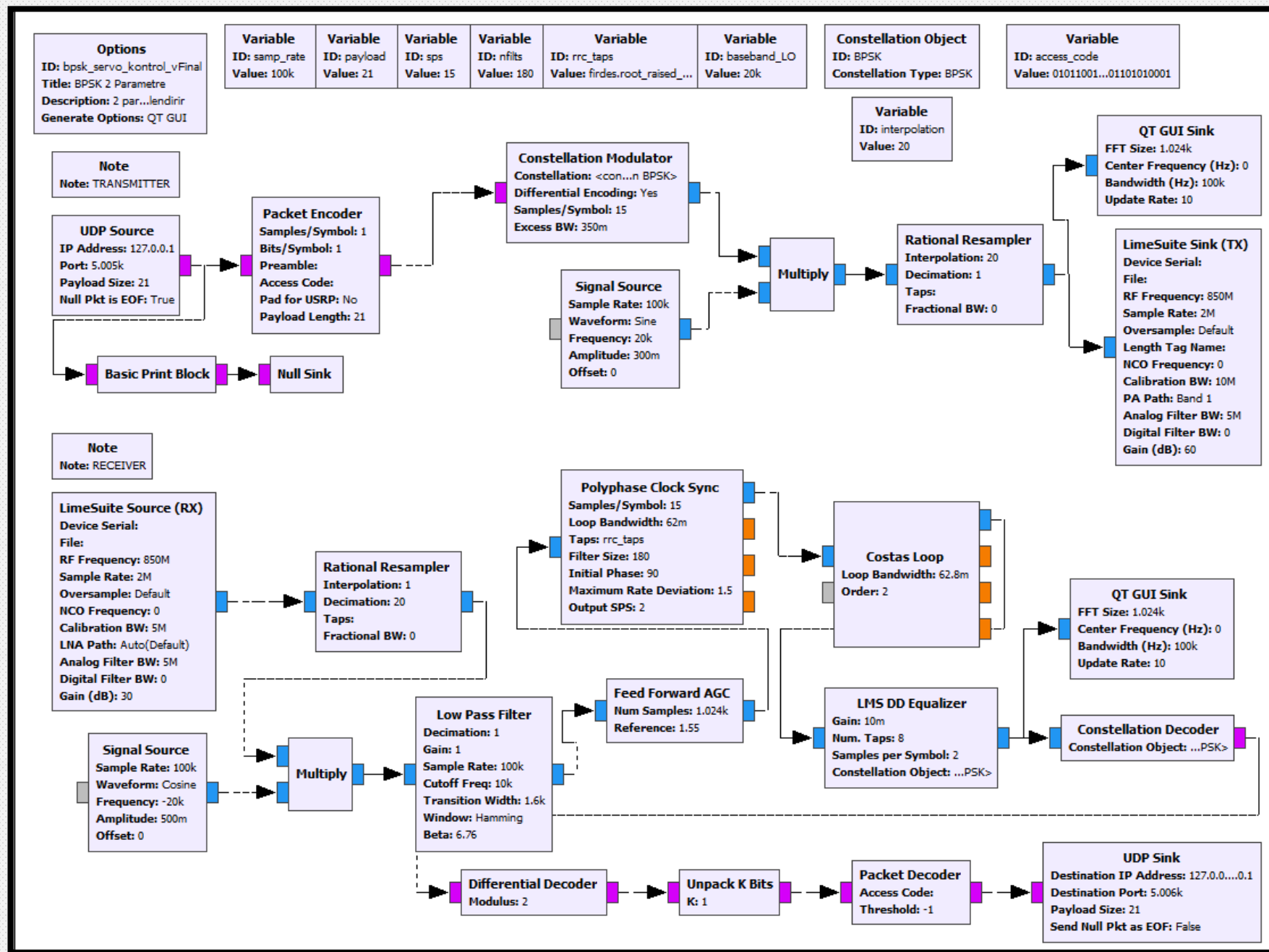


Alıcı Bloğu Decoder Bölgesi

TEKNİK TASARIM



BPSK Servo Kontrolü için
Fiziksel Katman Tasarımı

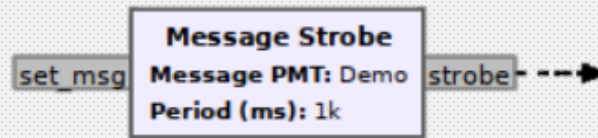


TEKNİK TASARIM

GNU Radio'ya Harici Kaynak Bağlamak

1- Message Strobe

*Mesajlar PMT objesi olarak taşınır
Stream yapmak için uygun değil*

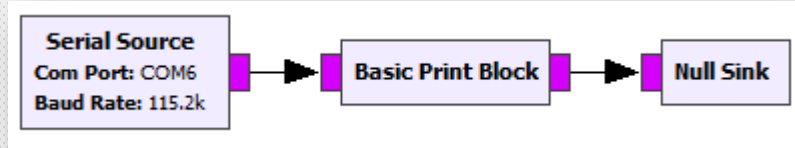


PMT (Polymorphic Type)

Python ve C++ arası ortak veri tipi objesi

Bir çeşit kodlama gibi

GNU Radio'ya Harici Kaynak Bağlamak



```
↑ serial_source.py
1 import numpy as np
2 from gnuradio import gr
3 import serial
4
5 class blk(gr.sync_block):
6
7     def __init__(self, COM_PORT="COM6", BAUD_RATE=9600):
8         gr.sync_block.__init__(
9             self,
10             name='Serial Source',
11             in_sig=None,
12             out_sig=[np.int8]
13         )
14         try:
15             self.ser = serial.Serial(COM_PORT, baudrate=BAUD_RATE, timeout=1)
16         except:
17             print "(!) Baglanti kurulamadi"
18
19     def work(self, input_items, output_items):
20         incomingLine = self.ser.readline()
21         output_items[0] = np.array([ord(char) for char in incomingLine],
22                                     dtype="int8")
23
24         return len(output_items[0])
```

2- Gömülü Python Bloğu

- Varsayılan stream blok çıkışları veri tipi ile aynı çıkış verilmesine rağmen başarısız
- Muhtemel çözüm BLOB (Binary Large Object) kullanılması
- Dokümantasyon yetersiz
- Zaman sınırsız değil

GNU Radio'ya Harici Kaynak Bağlamak

3- Socket Bağlantısı

Pozitif

GNU Radio varsayılan bloklar

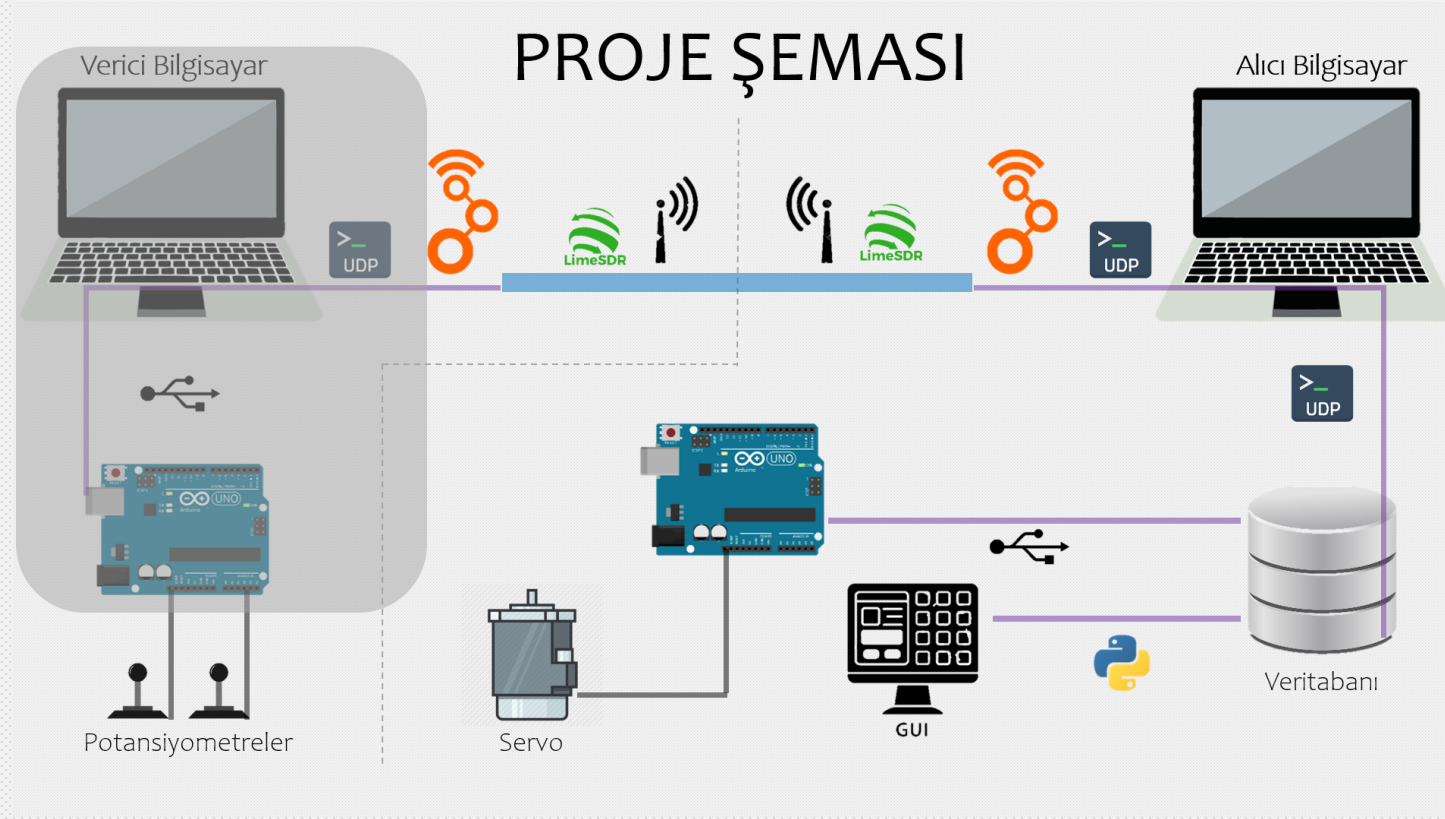
Negatif

Varsayılan blokların 'buffer' kontrolü yapılamıyor

Arduino direkt UDP bağlantısı yapamıyor

UDP için aynı anda yalnızca bir bağlantı

TEKNİK TASARIM



BPSK Servo Kontrolü Vericisi için
Data Link ve Network Katman Tasarımı

Serial->UDP Köprüsü

Payload Uzunluğu Seçimi

Büyük olduğunda Arduino Serial Buffer doluyor
Ne kadar küçük olmalı?

```
ser2udp.py
1 import socket, serial
2 from os import system
3 system("title "+"Serial-to-UDP")
4
5 UDP_IP = "127.0.0.1"
6 UDP_PORT = 5005
7 COM_PORT = 'COM10'
8
9 print(f"Transfer {COM_PORT} > {UDP_IP} {UDP_PORT}")
10
11 sock = socket.socket(socket.AF_INET, # Internet
12                      socket.SOCK_DGRAM) # UDP
13
14 ser = serial.Serial(COM_PORT, baudrate=2400, timeout=None)
15
16 while True:
17     MESSAGE = ser.read(21)
18     sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
19     print(MESSAGE)
```

<param0,param1>

param = [0,1023]

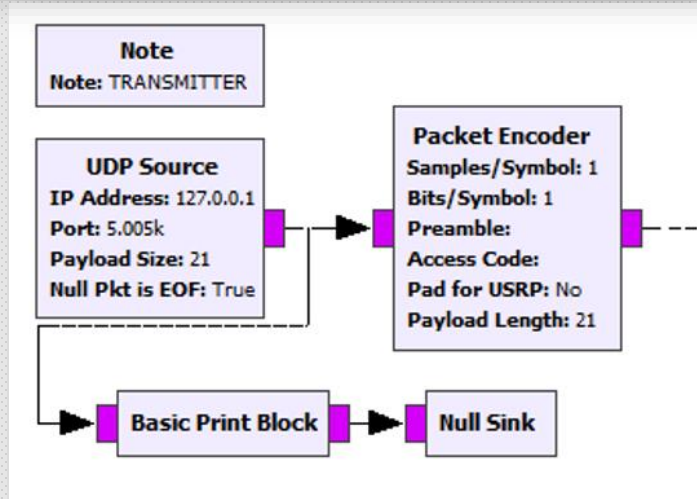
Veri sözcüğü

min: 5 ve max: 11

“1023,1023><1023,1023>”

21 byte

Bu sayede, alıcıda ayıklama sırasında kod karmaşıklığı azalıyor. (Arayüz)



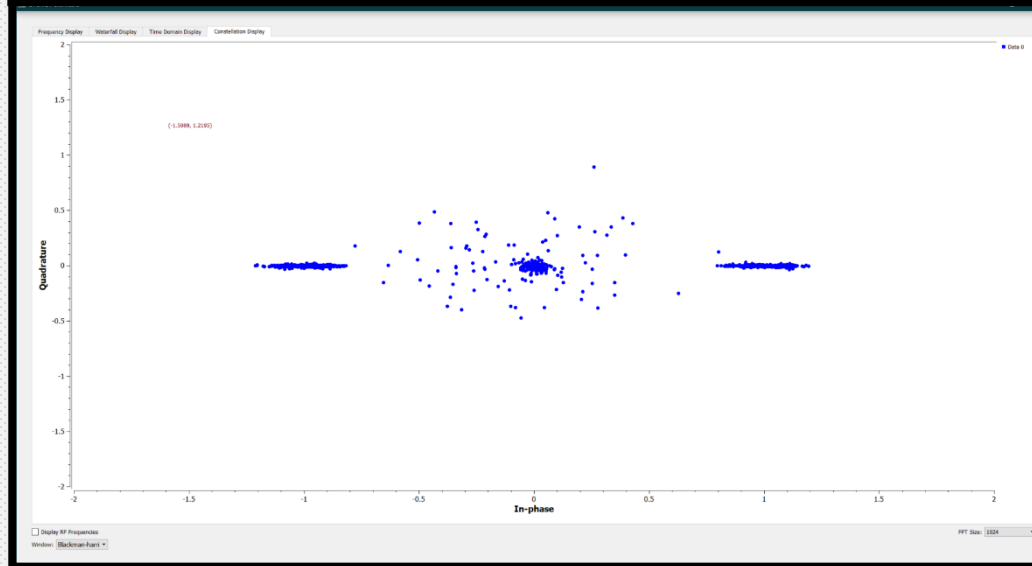
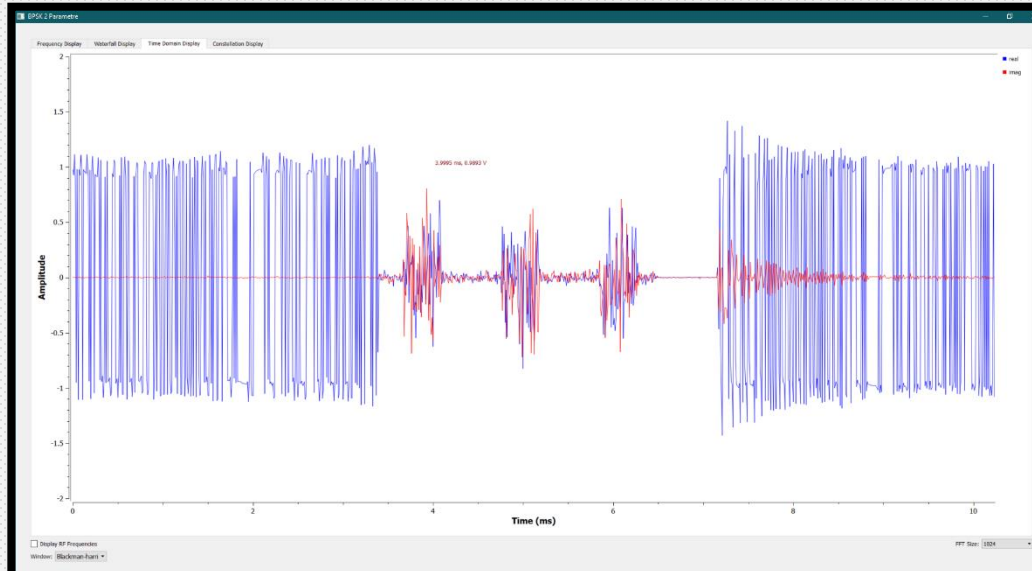
TEKNİK TASARIM

Donanım Senkronizasyonu *Verici tarafı Baudrate Seçimi*

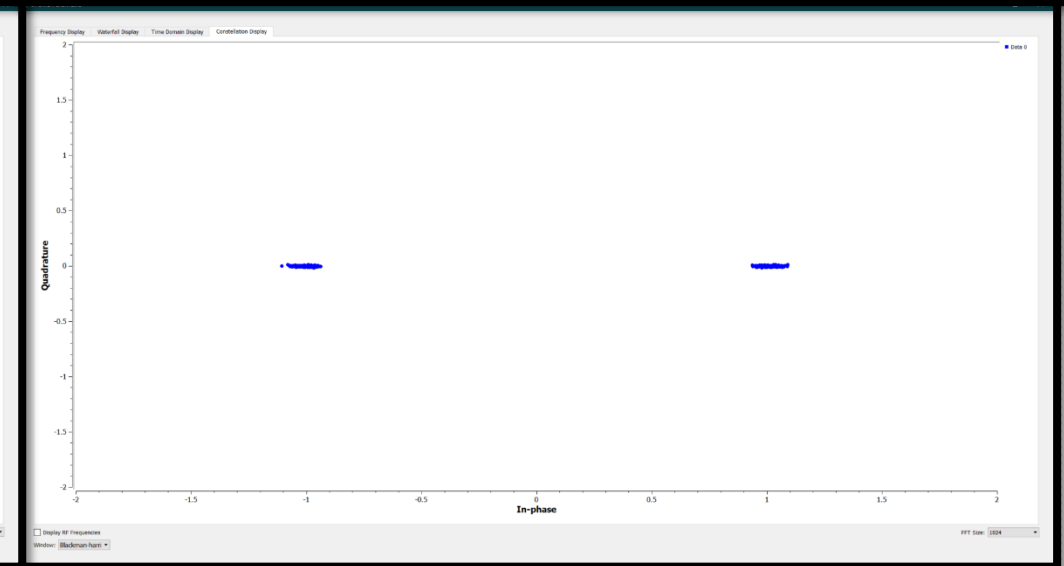
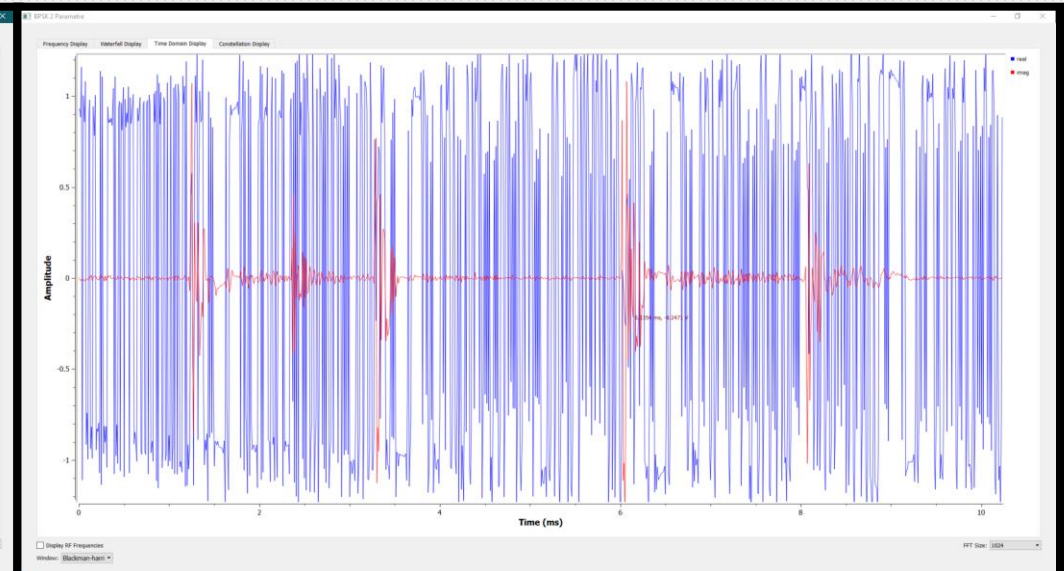
Hızı artırmak I-Q diziliminde gürültüyü azaltırken paketler arası boşluğun azalması Arduino'nun Serial Buffer'ının dolmasına ve ciddi gecikmelere neden olmaktadır.

Sistemdeki en zayıf halkayı Arduino cihazı oluşturmaktadır.

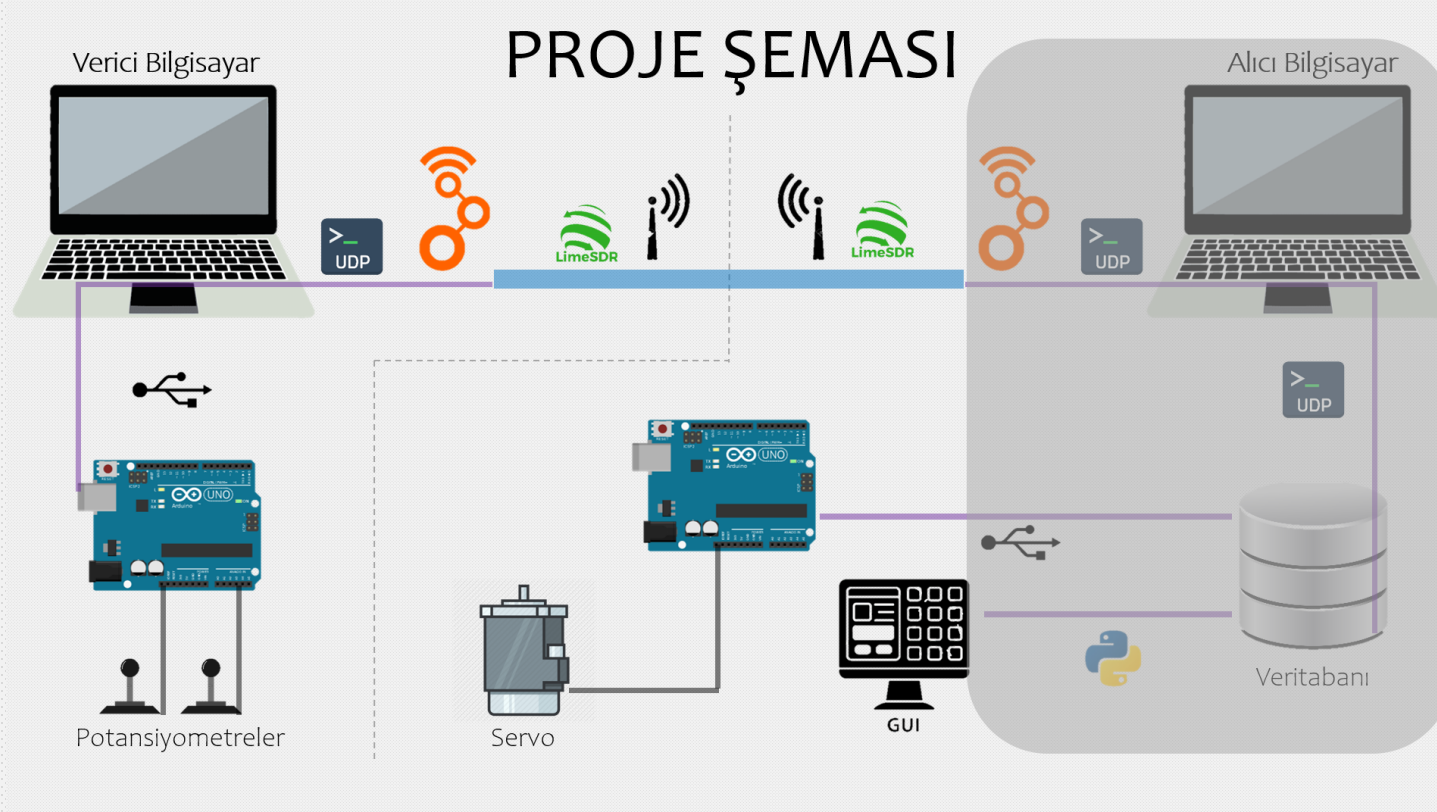
Baudrate: 3200 bps



Baudrate: 4800 bps



TEKNİK TASARIM



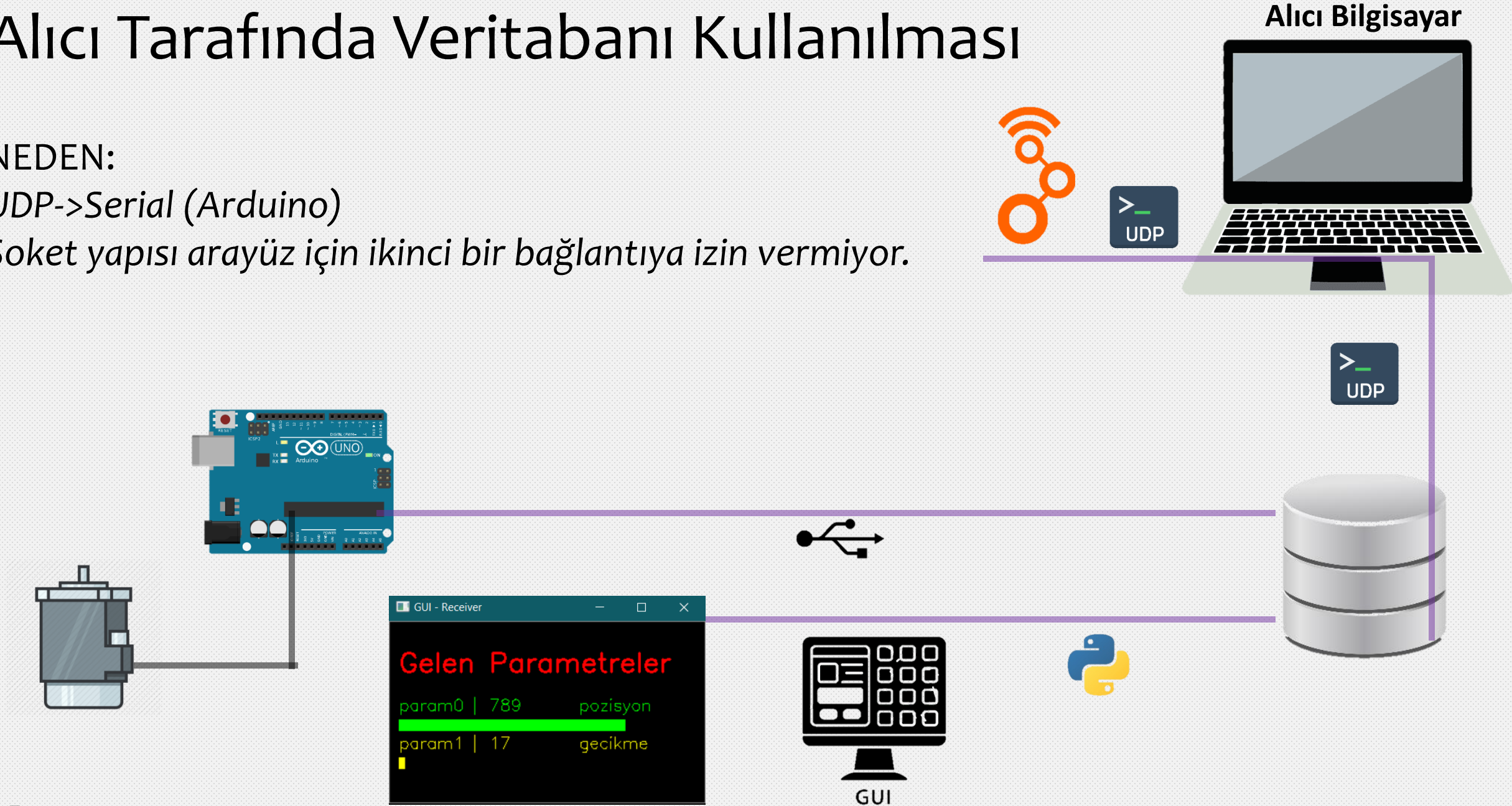
BPSK Servo Kontrolü Alıcısı için
Data Link ve Network Katman Tasarımı

Alıcı Tarafında Veritabanı Kullanılması

NEDEN:

UDP->Serial (Arduino)

Soket yapısı arayüz için ikinci bir bağlantıya izin vermiyor.



Servo Kontrolcüsü Arduino Kodu

```
void loop() {  
  if (Serial.available() > 0) {  
    int inChar = Serial.read();  
    if (inChar == '<') {  
      delay(80);  
      int param0 = Serial.readStringUntil(',').toInt() / 100;  
      int param1 = Serial.readStringUntil('>').toInt() / 100;  
  
      pos = map(param0, 0, 10, 0, 180);  
      dly = map(param1, 0, 10, 5, 40);  
  
      Serial.print("P:");  
      Serial.print(pos);  
      Serial.print("D:");  
      Serial.println(dly);  
      //      servo.write(pos);  
      //      delay(dly);  
    }  
    else {  
      delay(1);  
    }  
  }  
}
```

- *Veri ayıklaması yapılmadan önce bir miktar gecikme eklenerek Seri port buffer'ının dolması beklenmelidir.*
- *Potansiyometreler stabil çalışmadığından pozisyon bilgisinin kuantalanması verimi artırır.*

Demo

