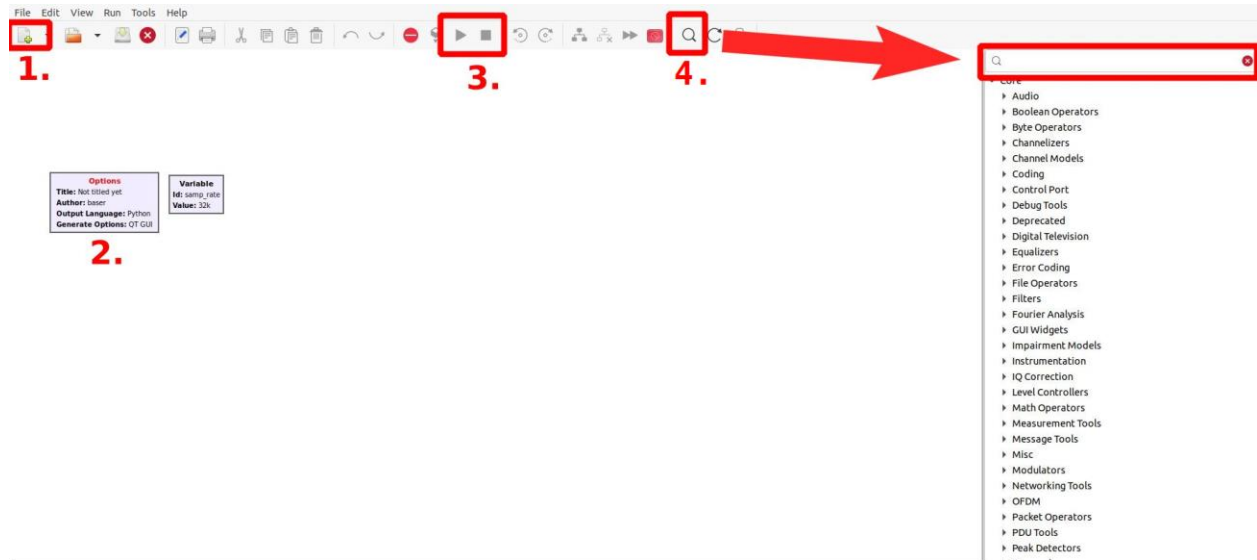


GNU Radyo Dijital Modülasyon Deneyi

1) GNU Radyo Arayüzü

Aşağıda GNU Rayo penceresi ilk açıldığı zaman görüntülenen arayüz verilmiştir. Pencere içerisi bu şekilde bu şekilde boş değilse “1” ile belirtilen yeni sayfa açma tuşuna basabilirsiniz. Pencere içerisindeki beyaz kısım akış alanı olarak isimlendirilmektedir. Deney içerisinde de bu şekilde tanımlanacaktır. “2” ile verilen kutuya çift tıklayarak oluşturacağımız akış için ayar penceresini görüntülenebilir. Deneye devam etmeden önce bu ayar penceresinde “Generate Options” olarak “QT GUI” seçili olduğuna emin olunmalıdır.

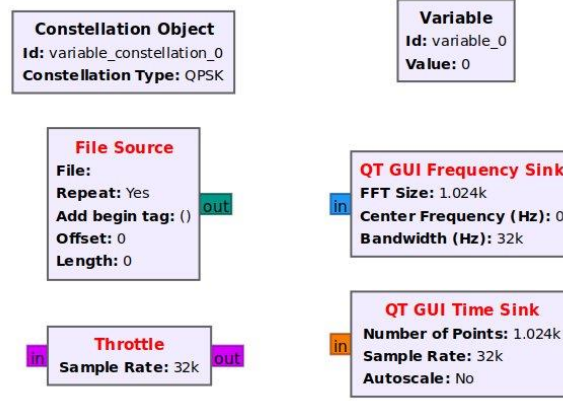
“3” ile verilen butonlar oluşturulan akış diyagramının çalıştırılmasını ve durdurulmasını sağlarlar. “4” ile verilen buton sağda görünen arama penceresini açar. Burayı kullanarak akış alanına yeni araçlar eklenebilir.



2) Sinyal Gönderme

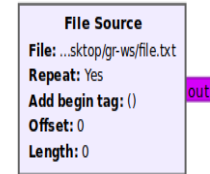
QPSK sinyal gönderim işlemi için ilk olarak modüle edilecek birler oluşturulmalı ve bu bitler QPSK modülatörüne verilmelidir. Bu işlemler için GNU Radyo’ da “**Random Source**”, “**Throttle**” ve “**Constellation Modulator**” blokları kullanılacaktır. Bu bloklar sağ üstteki arama penceresinden eklenmelidir. Ayrıca elde edilen çıkışların görüntülenebilmesi için “**Qt GUI Frequency Sink**” ve “**QT GUI Time Sink**” kullanılacaktır. “**Constellation Modulator**” bloğu çalışabilmek için modülasyon parametrelerine ihtiyaç duyar. Bu parametreler “**Constellation Rect. Object**” isimli blok

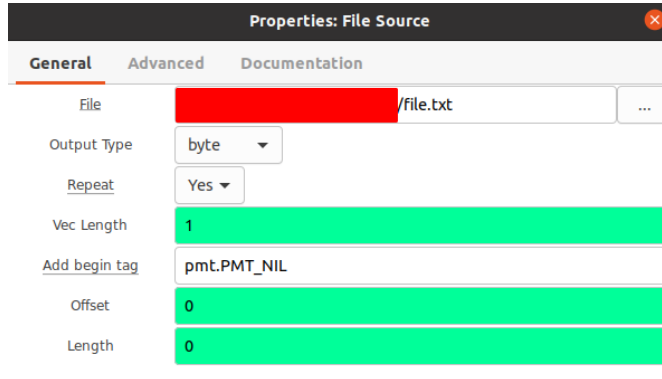
ile sağlanabilmektedir. Onu da arama kısmı kullanılarak akış içerisine eklemeliyiz. Kullanılacak blokların bazıları aşağıda verilmiştir.



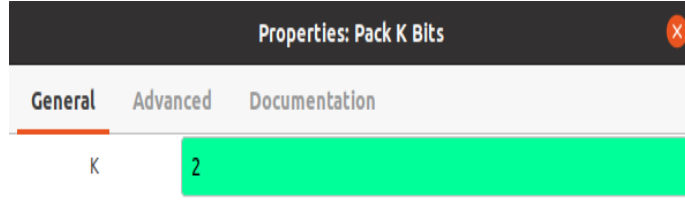
Verilen blokların bağlantı noktaları kenarlarında bulunmaktadır ve farklı renkte olabilirler. Bu renkler onlara verilecek girişin yada alınan çıkışın türünü ifade etmektedir. İki bloğun birbirine bağlanabilmesi için bağlantı noktalarının renkleri aynı olmalıdır. Bloklar arasında bağlantı yapmak için ilk bloğun çıkış bağlantı noktasına sol tıklanmalı daha sonra diğer bloğun giriş bağlantı noktasına sol tıklanmalıdır. Yeşil renk **integer** tipini, mor renk **byte** tipini, turuncu renk **float** tipini ve mavi renk ise **complex** veri tipini ifade eder. Bazı bloklar giriş ya da çıkış için bağlantı noktası bulunmaz. Bu tip bloklar akış içerisinde değişken ya da ayar işlemleri için kullanılırlar. **Ayrıca ismi kırmızı görünen bloklar içerisinde bir hata mevcuttur. O hata düzeltilmeden hazırlanan akış çalıştırılmaz.**

File Source: Bu blok parametre olarak gösterilen dosyayı akış içerisinde kaynak olarak kullanır. Dosya içerisindeki parametre ile bloğun çıkış tipi aynı olmasına dikkat edilmelidir. File olarak verilecek dosya masaüstünde “file.txt” olarak bulunmaktadır. Ayarları aşağıdaki gibi yapılmalıdır.





Pack K Bits: Bu blok girişindeki k sayıdaki biti birleştirerek çıkışında byte oluşturur. İlk bit MSB olacak şekilde birleştirme yapılır. Örnek olarak K değeri 4 girilmişse ve girişinde 01001000 gelmişse çıkışında 4 ve 8 üretir. Ayarı aşağıdaki resimdeki gibi yapılmalıdır.



Constellation Object: Sayısal haberleşme için gerekli olan modülasyon noktalarını belirlenmesini sağlayan bloktur. Ancak bu sadece ayar için kullanılır. Herhangi bir giriş ya da çıkış içermez. Kullanılacak blok içerisinde bu bloğun ID' si yazılarak ayarlanan modülasyonun gerçekleştirilmesi sağlanabilir.



Properties: Constellation Object

General Advanced Documentation

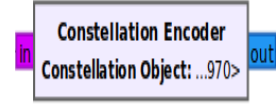
Id qpsk

Constellation Type QPSK

Soft Decisions Precision 8

Soft Decisions LUT None

Constellation Encoder: Bu blok parametre olarak aldığı Constellation Objesine göre girişindeki byte' ları modüle eder. Modülasyon ayarları Constellation Objesinde gerçekleştirileceği için bu blok fazla ayar gerektirmez.



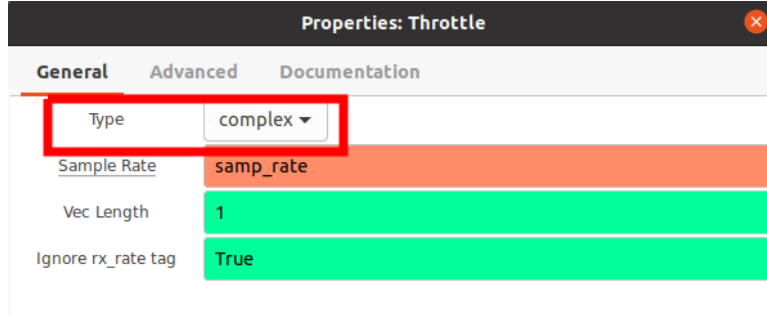
Properties: Constellation Encoder

General Advanced Documentation

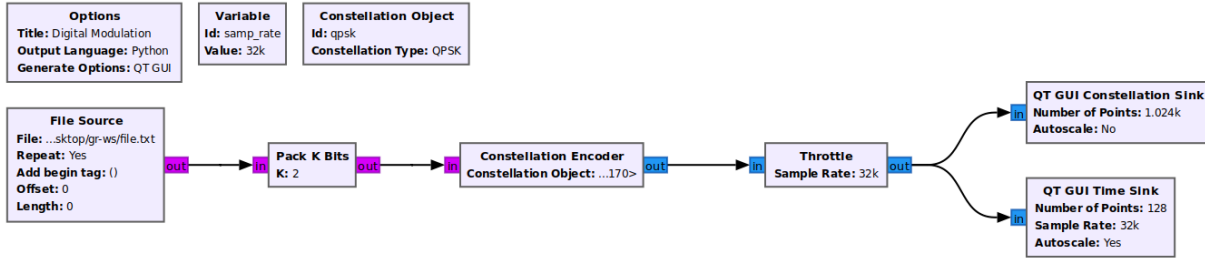
Constellation Object qpsk

Throttle: Bu blok akış içerisinde eklendiği koldaki veri hızını ayarlar. Veri hızı ayarlama işlemi “**Sample Rate**” parametresi değiştirilerek yapılır. Eklendiği yoldaki veri tipine göre Type parametresi seçilmelidir. Eğer GNU Radyo bir donanım ile kullanılıyorsa bu bloğun kullanılmasına gerek yoktur. Çünkü donanımların GNU Radyo’ daki bloğunda Sample Rate ayarlamak için yer bulunur.



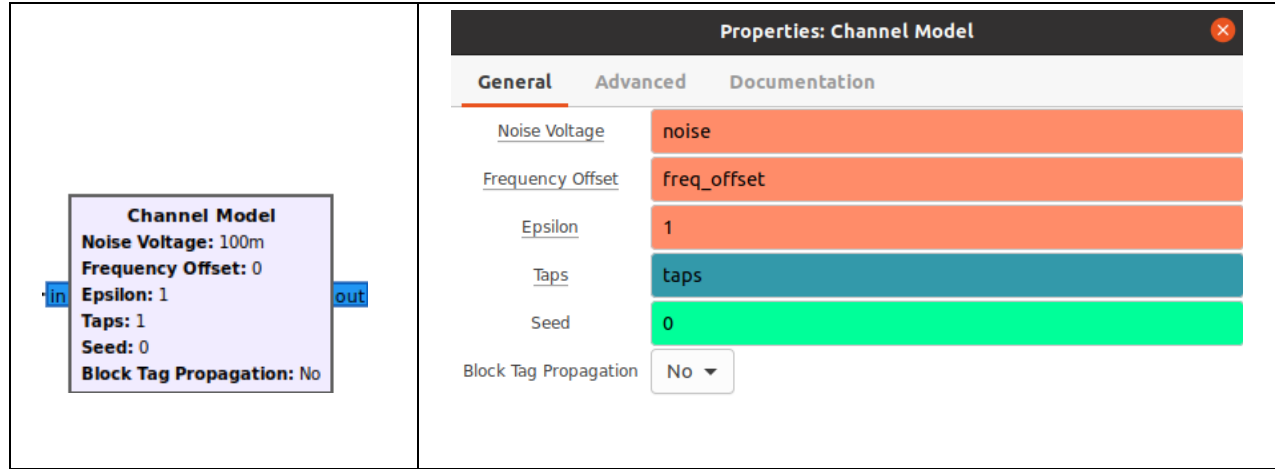


Sonuç olarak oluşturulacak ilk akış diyagramı aşağıda verilen resimdeki gibi olmalıdır. Bu kısımda sadece veri üretim kısmı oluşturulmuştur. Daha önceden ayarlanan bitlerden oluşan veri File Source bloğu ile akış içerisine alınır. QPSK modülasyonu yapılacağı için 2 bit ile bir sembol oluşturulur. Oluşturulan semboller QPSK modüle edilir ve akış içerisinde çizdirilir. Akışın sonuna eklenen “**QT GUI Constellation Sink**” gelen verileri X-Y grafiği şeklinde çizdirir. Böylece QPSK modüle edilen semboller reel – imajiner olarak görüntülenebilirler. “**QT GUI Time Sink**” ise gelen verileri zamana göre çizdirir. Akış oluşturulduktan sonra çalıştırılır ve grafikler incelenir.



3) Kanal Bozulma Etkileri

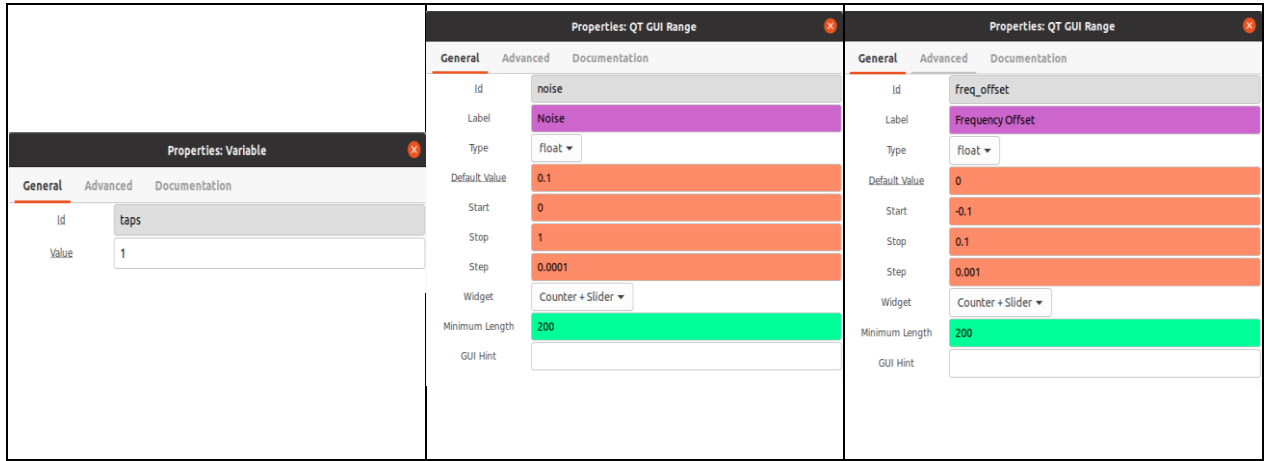
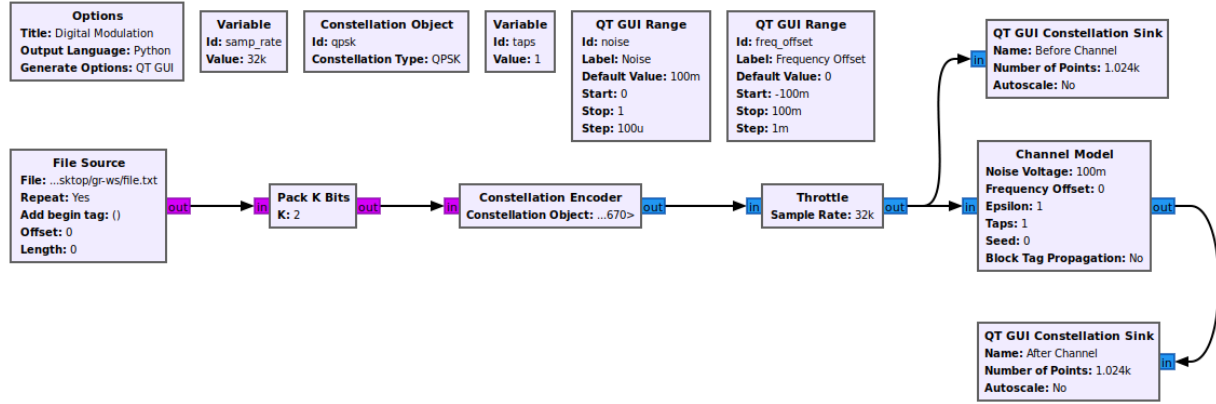
2. kısımda basit bir QPSK iletimi için gerekli verici yapısı gösterildi. Bu kısımda ise vericiden iletilen sinyal alıcıya ulaşılırken bozulma etkisi gösterilecektir. Bunun için öncelikle akışa “**Channel Model**” bloğu eklenmelidir. Bu blok haberleşme kanalındaki birkaç parametrenin simule edilmesini sağlar. Bu parametrelerden ilki gürültüdür. Eklenen blok içerisindeki gürültü değeri değiştirilerek haberleşme kanalına etki edecek gürültü genliği değiştirilebilir.



Haberleşme kanalına etki eden ikinci parametrede alıcı ve verici cihazlar arasındaki saat farkıdır. Bu saat farkı elektronik elemanlardaki üretim toleransından meydana gelir. Verici sinyali f_c frekans ile göndermeye ayarlandığında aslında $f_c + f_{\text{delta1}}$ ile gönderir. Alıcı da ise farklı bir saat çalışmaktadır. Bu saat de $f_c + f_{\text{delta2}}$ ile çalışır. Sonuç olarak alıcıda alınan işaret f_{delta1} ile f_{delta2} 'ye bağlı olarak değişiklik gösterecektir. Simülasyon ortamında gönderilmek istenen işarete bu etki Channel Model bloğundaki Frequency Offset parametresi kullanılarak eklenebilir.

Haberleşme kanalındaki bir değer bozucu etki de çok yolluluk etkisidir. Çok yolluluk etkisi, vericiden gönderilen sinyaller yansımalar vb. nedenler ile alıcıya birden fazla sayıda ulaşmasına denir. Alıcı ile verici arasındaki her haberleşme yolu farklı kanaldır. Bu kanallar Channel Model bloğundaki “Tap” parametresi değiştirilerek akışa eklenebilir. Tap sayısı ve tap' ların genlikleri arttıkça kanal bozulma etkisi artacaktır. Bu kısım sonraki bölümünlerde kullanılacaktır.

Bazı parametrelerin simülasyonun çalışması sırasında değerlerin güncellenebilmesi için kanal parametreleri bloklara değişken olarak verilecektir. Bunun için “QT GUI Range” bloğu kullanılacaktır. QT GUI Range bloğuna verilen ID, akışta değişken ismi olarak kabul edilir ve bu değişkenin kullanıldığı parametrenin değeri akış çalışırken değiştirilebilir. Yukarıdaki resimde de görüldüğü üzere Channel Model bloğunda noise, freq_offset ve taps isimli 3 değişken kullanılmıştır. Bu değişkenlerden noise ve freq_offset QT GUI Range olarak eklenmelidir. Diğerisi ise “variable” olarak eklenebilir. Chanel Model eklendikten sonra oluşan akış diyagramı aşağıdaki gibidir.

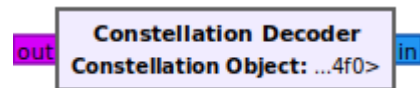


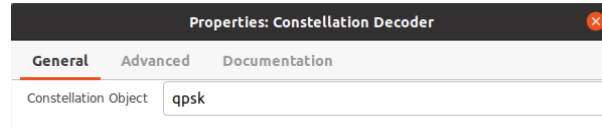
Akışa eklenmesi gereken değişkenler yukarıda verilmiştir. ID' ler doğru şekilde yazılmalıdır. Yanlış yazılırsa akış çalıştırılırken hata verecektir. Bunlara ek olarak QT GUI Constellation Sink bloklarındaki **"Name"** parametrelerinden üsttekini Before Channel, alttakini After Channel olarak değiştirerek akış çalıştırıldığında gösterilen arayüzde gösterilen grafikleri ayırmak kolaylaştırılabilir. Daha sonra akış çalıştırılıp gürültü ve ofset frekansı slider' ları hareket ettirilerek grafiklerdeki değişimler gözlemlenir.

4) Dekodlama

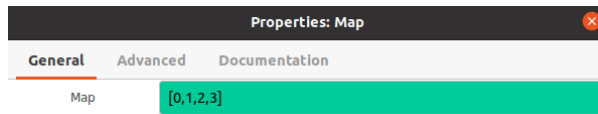
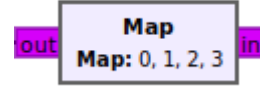
Dekodlama vericide alınan modüle edilmiş sembollerden bitleri elde etme işlemine verilen isimdir. Bu kısım da aslında ilk kısımda yapılan işlemlerin tam tersi gerçekleştirilecektir. Alınan veri demodülatöre verilecek, daha sonra semboller bitlere ayrılacak ve görselleştirilmek üzere QT Time Sink ile çizidilerek dekodlama işlemi tamamlanacak. Dekodlama işlemi için kullanılan bloklar aşağıdaki gibidir.

Constellation Decoder: Bu alıcıdaki encoder' in tersine gelen dijital modüleli verileri demodüle eder. Çalışmak için vericide kullandığımız Constellation Objesi parametre olarak verilmelidir.

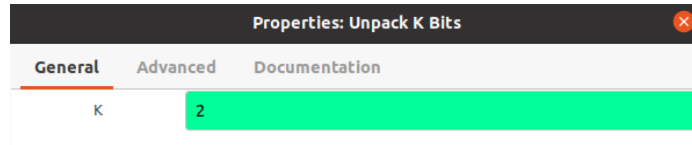




Map: Girişindeki veriyi parametre olarak aldığı değerlere göre oranlar. Bu blok kullanımı sonrasındaki bitlere ayırma işleminin daha doğru gerçekleşmesini sağlar. Kullanımı zorunlu değildir.

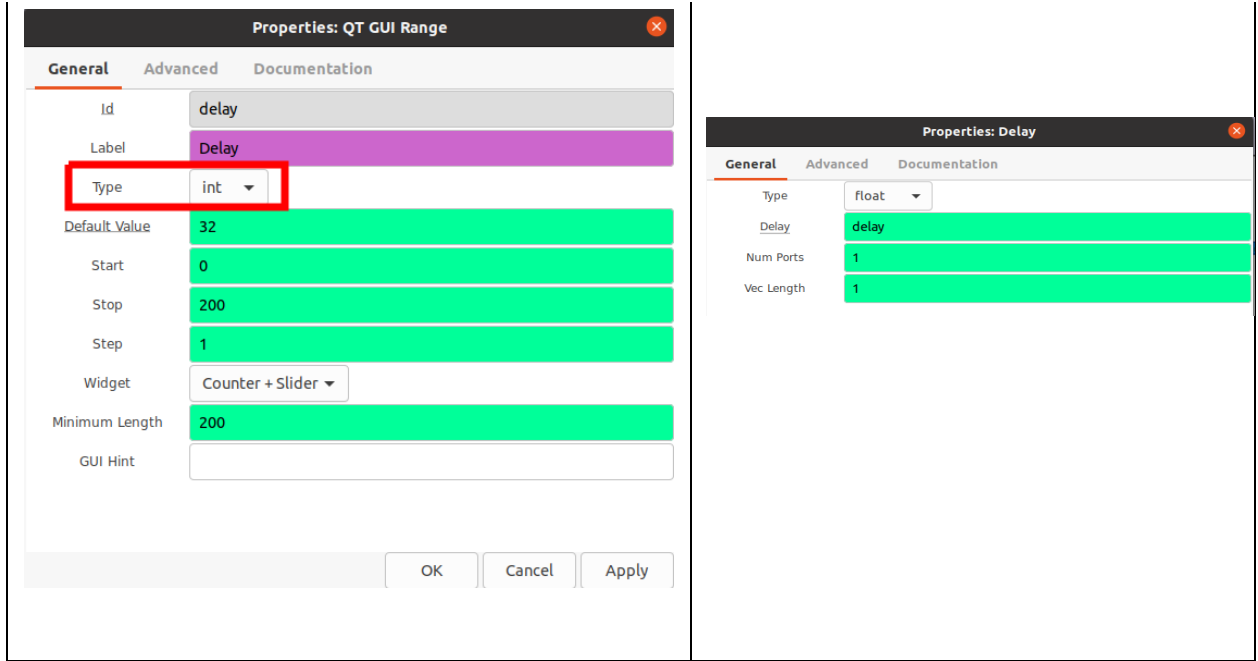


Unpack K Bits: Bu blok girişde kullanılan Pack K Bits bloğunun tersi olarak çalışır. Gelen sembolleri parametre değeri kadar bite ayırarak çıkışa aktarır.

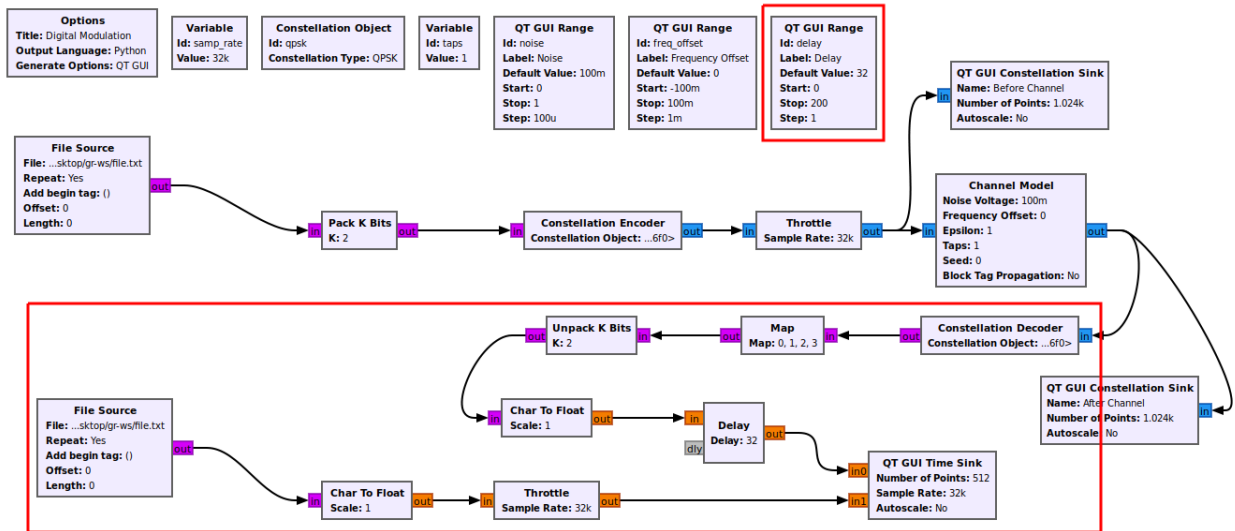


Delay: Girişindeki veriyi parametre olarak aldığı değerde sembol sayısı kadar geciktirerek çıkışa aktarır. Bu bloğa alıcı çıkışındaki bitler ile girişteki bitleri karşılaştırmak için ihtiyaç duyulmaktadır. İki ayrı koldan gelen verilerin aynı anda son bloğa ulaşmaları için gerekli gecikmeyi bu blok ile sağlayacağız. Akışta verinin hızlı ilerlediği kola eklenmelidir. Parametre olarak aldığı değer simülasyon sırasında değiştirilebilmesi için akışa QT GUI Range olarak eklenmiştir.





Sonuç olarak oluşturulacak akış diyagramı aşağıdaki şekildedir. Kırmızı ile işaretli olan kısımlar yeni eklenmiştir. Kullanılan **“Char To Float”** isimli bloklar isminden de anlaşıldığı gibi akış üzerindeki veri formatını değiştirmek için kullanılmaktadır. Farklı çeşitleri mevcuttur. Blokları bağlamak için gerekli yerlerde farklı çeşitleri kullanılabilir. Hazırlanan akış çalıştırıldığında zaman akış diyagramının altındaki çıktıyı verecektir. Akış içerisinde delay parametresini değiştirerek giriş ve çıkış değerlerinin üst üste oturması sağlanabilmektedir. Gönderilen veri paketinin başında çok sayıda sıfır bulunmaktadır. Referans olarak sıfırlar kullanılarak delay slider’ ı değiştirilebilir. Tam üst üste oturtulması mümkün olmasa bile yaklaştırılıp alınan ve verilerin aynı olduğunun kontrol edilmesi yeterlidir. Daha sonrasında gürültü artırılarak gürültünün demodülasyona etkisi incelenebilir.



Properties: QT GUI Constellation Sink

General Advanced Trigger Config Documentation

Type: Complex

Name: After Channel

Number of Points: 1024

Grid: No

Autoscale: No

Y min: -2

Y max: 2

X min: -2

X max: 2

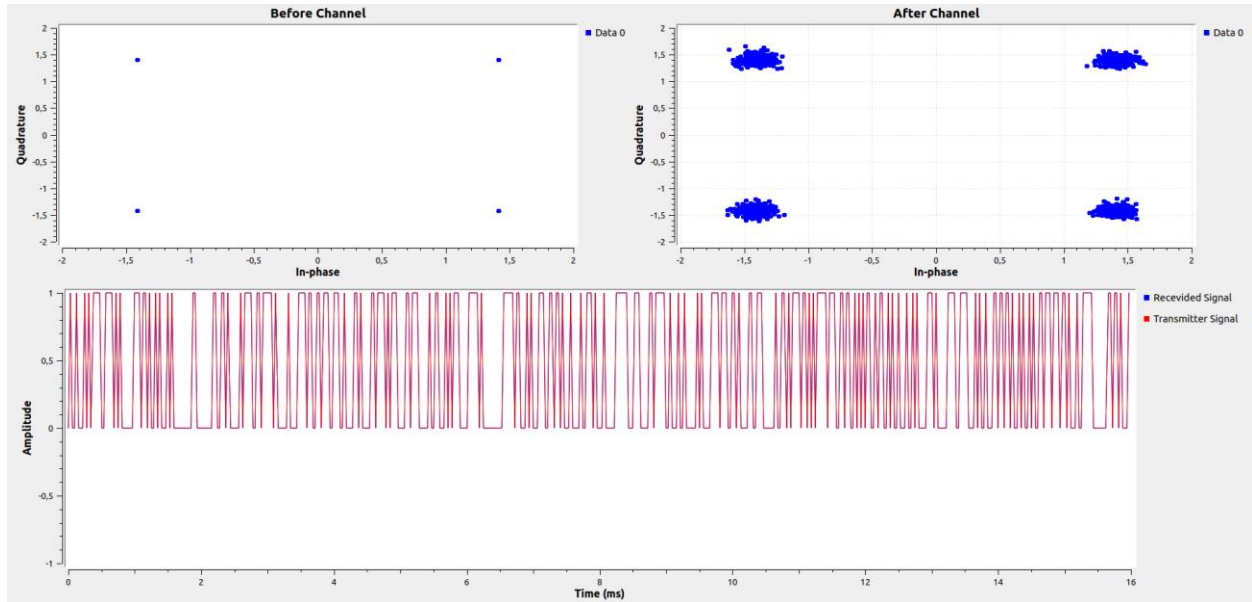
Number of Inputs: 1

Update Period: 0.10

GUI Hint:

OK Cancel Apply

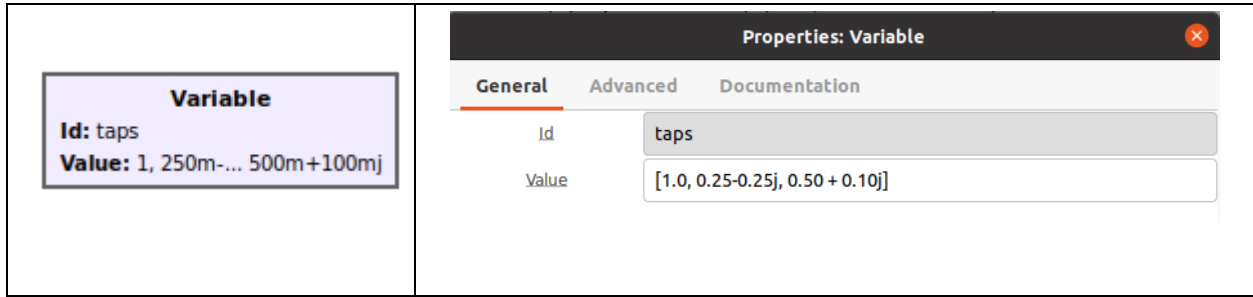
Verilen akışta olduğu gibi 2 girişli QT GUI Time Sink elde edebilmek için blok ayarları arasında “**Number of Inputs**” parametresi artırılmalıdır. Sonuç olarak elde edilecek çıktı doğru delay seçilmesi halinde aşağıdaki gibi olacaktır.



5) Çok Yolluluk Etkileri

Önceki bölümdeki haberleşme kanalını oluşturduk. Ancak haberleşme kanalında tap parametresi 1 olduğu için herhangi bir çok yolluluk etkisi yoktu. Şimdi bu parametre değiştirilerek kanal çok yollu bir hale getirilecektir. GNU Radyo ortamında çok taplı kanal oluşturmak için Channel Model bloğuna dizi şeklinde kanal tap' ları verilmelidir. Tap sayısı ve tap' ların genlikleri arttıkça kanal zayıflatması artacaktır.

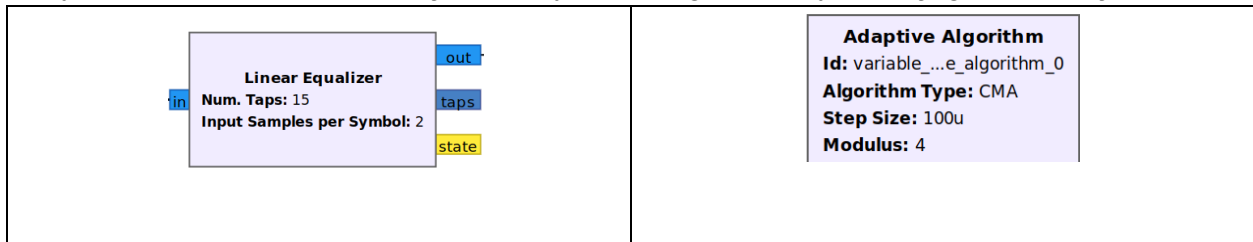
Önceki kısımda hazırladığımız akış diyagramında senkronizasyon sağlandıktan sonra frekans ofseti değiştirilerek çıktılardaki değişim incelenir. Daha sonra akış durudurulup kanal tap' ları $[1.0, 0.25-0.25j, 0.50 + 0.10j]$ şeklinde 3 taplı bir kanal ile değiştirilir ve çıkışlardaki değişim incelenir. Gürültüyü artırma ya da azaltma yaparak bit hataları giderilmeye çalışılır.

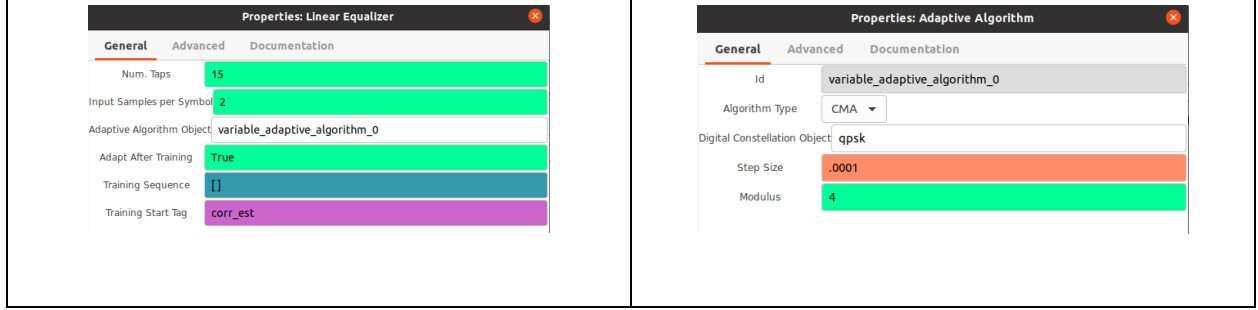


6) Denkleştiriciler

Özellikle çok taplı kanallarda meydana gelen bozulmaları azaltmak için kanal denkleştiriciler kullanılır. Kanal denkleştirici aslında alınan veriyi kanalın tersi ile çarpar. Teorik olarak bu işlemin alınan işareti tamamen düzeltmesi gerekir. Ancak pratikte kanal katsayıları bilinmediği için ancak düzeltme uygulanabilir. Hazırladığımız akışda da gerçekçi olması için kanal katsayılarını bilmeyen bir denkleştirici kullanılacaktır.

Akışa denkleştirici eklemek için “**Linear Equalizer**” bloğunu kullanacağız. Blok arama kısmından aranarak bu blok akışa eklenebilir. Ancak bu bloğun çalışması için yine bir ayar bloğuna ihtiyaç vardır. Eklenecek ayar bloğu denkleştiricinin kullanacağı algoritmayı ve algoritma parametrelerini belirlemede kullanılacaktır. Denkleştiriciye ayar yapmak için kullanılacak blok “**Adaptive Algorithm**” isimli bloktur. Bu bloğu arama ekranından akışa ekleyebilirsiniz. Eklenen bloklar içerisinde yapılması gereken ayarlar aşağıda verilmiştir.





Linear Equalization bloğundaki Adaptive Algorithm Object kısmına eklenen Adaptive Algorithm bloğunun ID' sinin yazıldığına dikkat edilmelidir. Hatalı yazılırsa akış çalışmayacaktır. Denkleştirici eklendikten sonra akış tekrar çalıştırılır. Çok tap' lı kanal etkilerinin giderilip giderilmediği constellation ekranına bakılarak kontrol edilir. Eğer kanalda meydana gelen bozulmalar azaldı (gönderilen semboller 4 noktada toplandı) ise kanalın frekans ofsetini değiştirerek meydana gelen etkiler incelenir. Alınana ve gönderilen semboller tam üst üste gelmeyebilir. Görsel algılanmasının kolay olması için sembol paketinin başına eklenen sıfırlar denkleştiricinin tam doğru çalışmamasına sebep olmaktadır. Bu sebeple alınan ile gönderilen veriler yakın olacak şekilde gecikme ayarı yapmanız yeterlidir.

7) Faz ve Frekans Düzeltme

Lineer denkleştirici, gelen veriyi kanalın tersi ile çarparak bozulmamış sembolü elde etmeye çalıştığını önceki bölümde söyledik. Ancak kanalda meydana gelen frekans kaymalarını denkleştirici önleyemez. Kanalda meydana gelecek frekans kaymalarını düzeltmek için GNU Radyo' da "**Costas Loop**" isimli bir blok bulunmaktadır. Bu BPSK, QPSK ve 8PSK için frekans düzeltmesi sağlayabilmektedir. Bloğu akış ekranındaki arama menüsünden akışa ekleyebiliriz. Eklendikten sonra yapılması gereken parametreleri ayarları aşağıda verilmiştir.

