



UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in Informatica

ELABORATO FINALE

TITOLO

Sottotitolo (alcune volte lungo - opzionale)

Supervisore
Fabrizio Granelli

Laureando
Andrea Filippi

Anno accademico 2017/2018

Ringraziamenti

...thanks to...

Indice

Sommario	3
1 Introduzione sulle tecniche di trasmissione	3
1.1 Modulazioni Analogiche	4
1.1.1 AM (amplitude modulation):	4
1.1.2 FM (frequency modulation):	4
1.2 Modulazioni Digitali	4
1.2.1 FSK (frequency shifting key):	4
1.2.2 ASK (Amplitude shift keying):	4
1.2.3 PSK (Phase Shift Keying):	4
1.2.4 DPSK (Differential Phase Shift Keying):	5
1.2.5 QAM (Quadrature amplitude modulation):	5
2 OFDM (orthogonal frequency-division multiplexing):	6
2.1 Principi di funzionamento	6
2.1.1 Ortogonalità delle sottoportanti:	6
2.1.2 Tempi di guardia	7
2.1.3 Equalizzazione	7
2.1.4 Recupero degli errori	7
2.1.5 Sincronizzazione in frequenza	8
2.2 Proprietà e campi di utilizzo	9
3 SDR (Software Defined Radio)	11
3.0.1 USRP (Universal Software Radio Peripheral)	11
3.0.2 RTL-SDR	12
4 GNURADIO	13
5 Crittografia RSA	13
5.1 Caratteristiche	14
5.1.1 Codifica asimmetrica con doppia chiave	14
5.1.2 Algoritmo unico	14
5.1.3 Chiavi intercambiabili	14
5.1.4 Procedimento creazione chiave monodirezionale	14
5.2 Algoritmo	14
5.2.1 Generazione delle chiavi	14
5.2.2 Cifratura	14
5.2.3 Decifratura	14
6 OFDM in Gnu Radio	15
6.1 Trasmettitore	15
6.1.1 Formazione pacchetti dal flusso in ingresso	15
6.1.2 Codifica informazioni in Simboli	16
6.1.3 Allocazione simboli sulle sottoportanti e creazione campioni per l'SDR	16

6.1.4	Passaggio dati all'SDR	16
6.2	Ricevitore	16
6.2.1	Ricezione informazioni, sincronizzazione	17
6.2.2	Equalizzazione e ottenimento simboli	17
6.2.3	Decodifica payload e scrittura file	18
6.3	Testing	18
7	Implementazione Rsa in Gnuradio	18
7.1	Sviluppo dei moduli	19
7.1.1	Definizione proprietà moduli per l'ambiente grafico	19
7.1.2	Algoritmo per cifratura e decifratura	19
7.1.3	Codice di test	20
7.1.4	Importazione in Gruradio	20
8	Conclusioni	21
	Sitografia	22
A	Titolo primo allegato	23
A.1	Titolo	24
A.1.1	Sottotitolo	24
B	Titolo secondo allegato	24
B.1	Titolo	24
B.1.1	Sottotitolo	24

Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Sommario è un breve riassunto del lavoro svolto dove si descrive l'obiettivo, l'oggetto della tesi, le metodologie e le tecniche usate, i dati elaborati e la spiegazione delle conclusioni alle quali siete arrivati.

Il sommario dell'elaborato consiste al massimo di 3 pagine e deve contenere le seguenti informazioni:

- contesto e motivazioni
- breve riassunto del problema affrontato
- tecniche utilizzate e/o sviluppate
- risultati raggiunti, sottolineando il contributo personale del laureando/a

Introduzione sulle tecniche di trasmissione

La comunicazione fra gli esseri umani è senzadubbio una delle abilità che hanno permesso all' uomo di evolversi. Sin dall'alba della civiltà l'uomo ha utilizzato la comunicazione per esprimere bisogni o intenzioni ai propri simili. La prima forma di comunicazione è stata senzadubbio quella verbale, metodo molto rapido ed efficace che però non garantisce la durata delle informazioni trasmesse. Altri metodi di comunicazione vennero sviluppati alcuni fra i più interessanti sono le nuvole di fumo che venivano utilizzate dagli indigeni d'america e più recentemente dai cinesi per comunicare lungo la grande muraglia cinese ed i tamburi utilizzati dalle tribù africane. La scrittura apparso circa 7000 anni fa favorendo l'inizio di un progresso che porterà l'uomo verso il ruolo centrale che ha ora sulla terra. Già nell'epoca della nascita di cristo l'uomo aveva instaurato una rete di comunicazione in forma scritta che interessava tutto il vecchio continente e lo collegava anche al mondo indiano ed orientale. Altri metodi un po' particolari vennero sfruttati prima dell' invenzione dell' elettricità come ad esempio i piccioni viaggiatori che dimostrano avere un packet loss del 17% oppure l'utilizzo di una lingua formata da fischi fra le lunghe valli nelle isole dell'arcipelago delle Canarie. Con la scoperta della corrente elettrica si è aperto per noi un nuovo mondo di possibilità fra le quali quella di trasferire immense quantità di informazioni velocemente e su lunghe distanze. Dapprima il telegrafo e poi il telefono fino ad arrivare alle trasmissioni analogiche seguite dall' avvento dell' era digitale. Nelle telecomunicazioni moderne per trasmettere si utilizza una portante (segnale elettrico oppure onda elettromagnetica) alla quale vengono aggiunte le informazioni secondo diverse tecniche dette anche modulazioni. Alcune principali modulazioni sono elencate e discusse in seguito.

Modulazioni Analogiche

- **1.1.1 AM (amplitude modulation):**

la modulazione in ampiezza è stata una delle prime modulazioni utilizzate grazie alla sua facilità di implementazione in hardware. Il segnale viene direttamente sommato alla portante in modo analogico, la sua semplicità è ormai l'unico vantaggio in quanto una soluzione di questo tipo è soggetta ad interferenze di qualsiasi origine, è sufficiente infatti una semplice attenuazione del segnale per influire direttamente sui dati ricevuti. Questa modulazione viene ancora utilizzata per la trasmissione radio che grazie a frequenze molto basse (khz) e potenze elevate (kw) permette di comunicare su distanze mondiali

- **1.1.2 FM (frequency modulation):**

la modulazione viene effettuata variando la frequenza del segnale portante alzandola o abbassandola in relazione alle informazioni da trasmettere, è più efficiente della modulazione in ampiezza in quanto non necessita di variare la potenza. Richiede però dei circuiti più complessi che siano in grado di svolgere il compito di codifica/decodifica. La modulazione in frequenza FM è tuttora utilizzata per la trasmissione della radio anche se sta venendo pian piano sostituita dalla radio digitale "DAB"

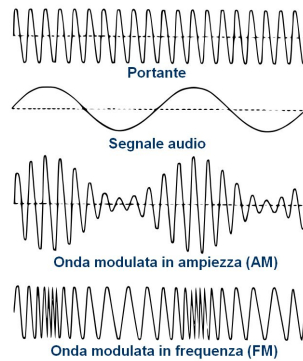


Figura 1.1: rappresentazione segnale modulato am e fm nel tempo

Modulazioni Digitali

- **1.2.1 FSK (frequency shifting key):**

questa tecnica di modulazione codifica l'informazione variando la frequenza della portante in valori predefiniti, ad esempio per ottenere una codifica binaria alterna due frequenze diverse. Questa tecnica ha il vantaggio di essere facile nell'implementazione e poco soggetta ad interferenze tuttavia necessita di una maggiore larghezza di banda rispetto ad altre modulazioni digitali quali psk o ask. [13]

- **1.2.2 ASK (Amplitude shift keying):**

la modulazione viene effettuata variando l'ampiezza del segnale portante alzandola o abbassandola in relazione alle informazioni da trasmettere, richiede un canale più affidabile in grado di ricevere anche i livelli di ampiezza più bassi. Trova ancora utilizzo nelle fibre ottiche e viene tuttora utilizzata la sua versione a codifica binaria (solo due livelli di potenza della portante presente/non presente) che prende il nome di OOF(on/off keying) in passato utilizzata anche per trasmettere messaggi in codice morse. [2]

- **1.2.3 PSK (Phase Shift Keying):**

questo tipo di modulazione codifica le informazioni in ingresso variando la fase della portante, ne esistono varie versioni che differiscono per il numero di valori diversi. La versione più semplice è

la binaryPSK che varia di metà periodo mentre versioni come la 4PSK di un quarto e così via per la variante 8PSK e 16PSK. Tali possibili sfasature della portante vengono dette costellazione e vengono di norma rappresentate come coordinate complesse su un grafico. Sulle ascisse si trova la portante mentre sulle ordinate si trova in quadratura ovvero sfasata di 90° . La lunghezza del vettore fra l'origine e uno dei punti della costellazione rappresenta l' ampiezza del segnale modulato mentre l'angolo rappresenta la sfasatura rispetto alla portante. Esiste una variante di 4PSK detta QPSK che differisce per una disposizione della costellazione ruotata di 45° .

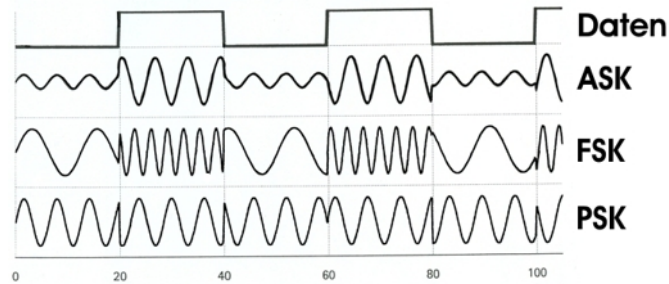


Figura 1.2: rappresentazione segnale modulato nel tempo ASK FSK e PSK [14]

• 1.2.4 DPSK (Differential Phase Shift Keying):

questa tecnica differisce da PSK solo per la particolarità di codificare il simbolo non utilizzando una costellazione fissa, le informazioni sono espresse come cambio di fase rispetto al simbolo precedente. Tale caratteristica rende questa modulazione robusta sia contro variazioni di ampiezza come psk sia contro distorsioni della fase del segnale ricevuto.

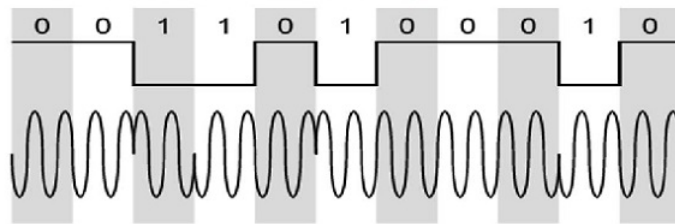


Figura 1.3: rappresentazione segnale modulato DPSK [8]

• 1.2.5 QAM (Quadrature amplitude modulation):

è una tecnica di modulazione simile a psk ma introduce la modulazione anche in ampiezza. La costellazione risulta avere punti non più equidistanti dall' origine. QAM come PSK presenta varianti che differiscono per il numero di punti sulla costellazione, in sistemi moderni si utilizzano anche 256 punti. Una particolarità comune a psk consiste nel fatto che due punti della costellazione adiacente differiscono per un solo bit, ciò incrementa l' efficacia di un eventuale error recovery. QAM viene anche utilizzato per trasferire più flussi analogici contemporaneamente, questo particolare utilizzo fa sì che QAM venga considerato anche come una tecnica di modulazione analogica. [19]

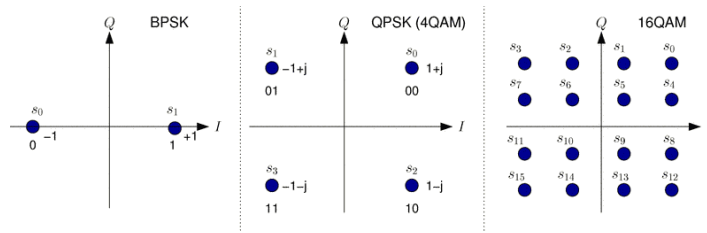


Figura 1.4: Costellazioni BPSK, QPSK(4QAM), 16PSK [3]

OFDM (orthogonal frequency-division multiplexing):

OFDM è una tecnica di codifica digitale su portanti multiple, su ogni sotto portante vengono modulate le informazioni utilizzando una delle varie tecniche digitali disponibili (solitamente si utilizza una versione di PSK oppure di QAM). OFDM trova svariati utilizzi nei sistemi di comunicazione moderni come ad esempio adsl, fibra ottica, 4G, wi-fi(802.11a/g/n/ac), radio/televisione digitale e WiMAX. [15] [16]

Principi di funzionamento

• 2.1.1 Ortogonalità delle sottoportanti:

la divisione della larghezza di banda disponibile in sezioni più piccole non è una caratteristica unica di OFDM, ciò che lo distingue è la soluzione al problema di interferenza fra i sotto canali. La soluzione classica è quella di lasciare delle piccole bande di frequenza vuote dove non si trasmette fra un canale e quello adiacente, da notare che in questo tipo di approccio si ha una allocazione inefficiente della larghezza di banda disponibile. OFDM utilizza la proprietà di ortogonalità per riuscire addirittura a sovrapporre parzialmente un canale con il successivo evitando sprechi di banda e aumentando così l'efficienza spettrale (indica la bontà del sistema nello sfruttare in maniera più o meno efficiente la banda disponibile [10]). La spiegazione matematica di questa tecnica è complessa e fa uso della trasformata di Fourier, è tuttavia possibile intuirne il funzionamento dall'immagine sottostante. L'immagine rappresenta la disposizione delle sotto portanti, sulle ascisse troviamo le frequenze mentre sulle ordinate l'ampiezza. Ogni picco rappresenta la presenza di un simbolo modulato sulla rispettiva portante. E' possibile notare che in questa particolare disposizione il rumore che genererebbe ogni canale all'esterno della propria banda si annulla esattamente in corrispondenza delle frequenze di trasmissione dei simboli adiacenti non creandogli disturbo. Perché l'ortogonalità garantisca che non ci siano interferenze fra le diverse sottoportanti è necessario che il tempo di trasmissione dei simboli sia uguale in tutto il sistema, in ambienti caratterizzati da una variazione dell'attenuazione sul mezzo trasmissivo ad esempio sul doppino adsl un eventuale algoritmo finalizzato ad aumentare il throughput non potrà agire sulla velocità di trasmissione dei simboli ma sulla grandezza della costellazione utilizzata per modulare nella sottoportante (passando ad esempio da un bpsk che trasferisce un bit per simbolo a 16psk che ne trasferisce 4)). Altro requisito fondamentale che verrà ripreso nella sezione dedicata all'implementazione sul software gnuradio è la necessità di avere un'ottima sincronizzazione sulla frequenza fra trasmettitore e ricevitore, le cause di una cattiva sincronizzazione possono essere varie un esempio è l'effetto Doppler causato dal movimento di un apparato rispetto all'altro durante la comunicazione, il cosiddetto multi-path per le reti wireless oppure semplici imperfezioni sul clock (generatore di frequenza) fra gli apparati.

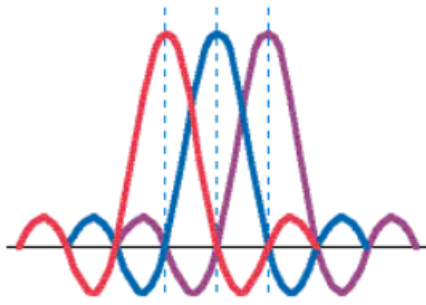


Figura 2.1: ortogonalità sottoportanti OFDM [25]

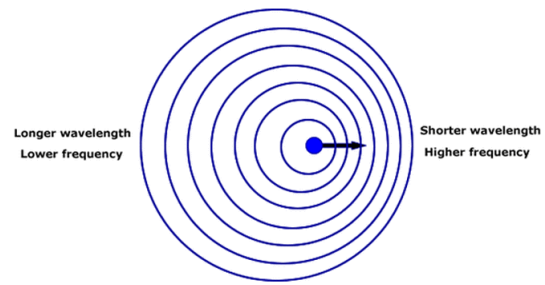


Figura 2.2: effetto Doppler [9]

• 2.1.2 Tempi di guardia

OFDM può soffrire di ISI (intersymbol interference) per uno dei problemi appena esposti nella sezione precedente. Questo problema avviene quando un simbolo trasmesso arriva al ricevitore assieme al precedente facendo fallire la decodifica, per risolvere questo problema viene aggiunto un tempo detto di guardia lungo solitamente attorno ad $1/10$ del simbolo. Inizialmente durante questo breve intervallo non veniva trasmesso nulla, successivamente è risultato più efficiente trasmettere l'ultimo pezzo del simbolo dopo (cyclic prefix) favorendo la corretta decodifica ricevitore.

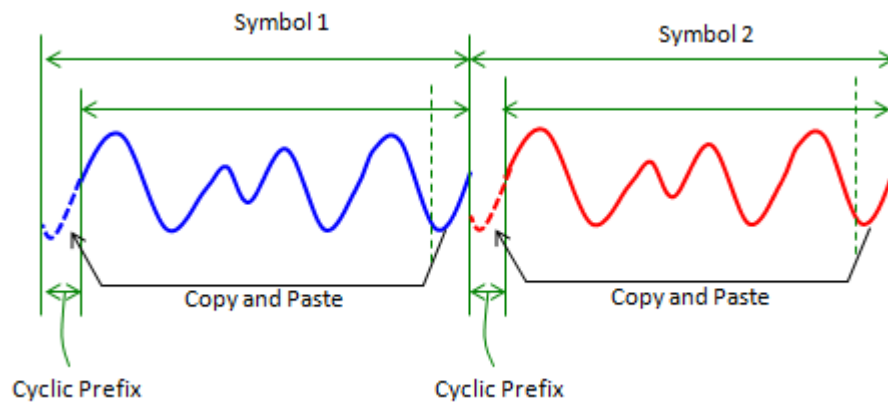


Figura 2.3: cyclic guard [6]

• 2.1.3 Equalizzazione

L'equalizzazione del segnale ricevuto è una procedura fondamentale per la corretta decodifica delle informazioni. L'obiettivo di questa procedura è quello di modificare il segnale ricevuto cercando di agire esattamente nel modo opposto di come è stato distorto dal canale in modo da annullarne la distorsione, per ottenere tale risultato esistono numerosi algoritmi di equalizzazione che differiscono per gli approcci diversi in relazione alle informazioni disponibili sul mezzo trasmissivo oppure alle informazioni ottenute durante la trasmissione stessa. Esistono due tipi di equalizzazione possibile uno nel dominio del tempo quindi analizzando lo scorrere dei simboli e uno nel dominio delle frequenze che analizza il comportamento del canale nelle varie sottoportanti. All'inizio di una trasmissione OFDM vengono inviati dei simboli pilota ben noti al ricevitore che li utilizza per stimare la distorsione del canale di trasmissione, il ricevitore aggiusta le proprie previsioni anche mentre sta ricevendo grazie al preambolo contenuto nei pacchetti. Da notare che l'equalizzatore assieme al segnale aumenta anche il rumore.

• 2.1.4 Recupero degli errori

Individuare e correggere gli errori avvenuti nella trasmissione è un compito complesso. Un sistema OFDM necessita la correzione di errore per essere robusto.

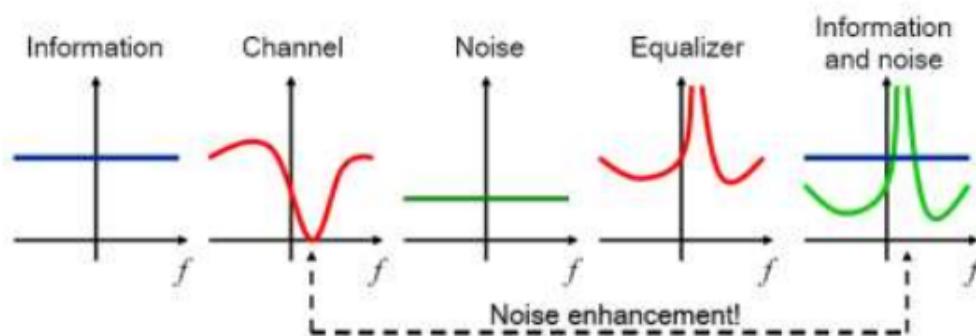


Figura 2.4: principio funzionamento equalizzatore [11]

CRC (Cyclic Redundancy Check)

CRC è una tecnica di controllo dell' errore, il funzionamento si basa sull' aggiunta alle informazioni da controllare di un breve codice di controllo. CRC non è in grado di correggere un eventuale errore e non è adatto per verificare la correttezza delle informazioni da un eventuale manomissione. Le informazioni di controllo aggiunte da CRC al messaggio iniziale rappresentano una divisione fra il cosiddetto polinomio generatore (specifico per ogni variante CRC) e il polinomio generato dal messaggio a cui viene sottratto il resto. Anche se di complessa spiegazione questa operazione può essere interamente eseguita in hardware mediante shift register e xor risultando molto efficiente. Il ricevitore eseguirà la divisione fra il polinomio generatore e il messaggio verificando che il resto sia nullo, se così non fosse il ricevitore è certo che sia presente almeno un errore nelle informazioni ricevute. [5]

Convolutional Coding

OFDM può utilizzare CC (Convolutional coding) per la determinazione e (a differenza di CRC) la correzione dell' errore, il principio di funzionamento di questo algoritmo si basa sulla creazione di un diagramma a stati che permette di codificare non solo la sequenza di bit in ingresso ma anche la loro transizione di stato. Il rapporto fra quantità di bit in ingresso e quello in uscita è detto code rate e può variare a seconda delle circostanze, un code rate di 1/2 ad esempio aggiunge 1 bit ogni bit in ingresso mentre con un rapporto 4/5 viene aggiunto un bit ogni 4. Meno bit aggiunti si traduce in meno bit da inviare ma minore efficacia nella correzione d'errore. [4] Spesso in OFDM la tecnica di Convolutional coding viene utilizzata assieme ad altre tecniche di recupero errore più complesse come ad esempio Reed-Solomon in grado di recuperare ulteriormente informazioni danneggiate. E' bene puntualizzare che esiste un limite matematicamente dimostrato insuperabile alla quantità di informazioni trasferibili su un canale affetto da rumore, questo limite è detto di Shannon. [17]

• 2.1.5 Sincronizzazione in frequenza

Lo sfasamento in frequenza fra trasmettitore e ricevitore è un problema comune a tutte le telecomunicazioni. OFDM ne è particolarmente sensibile data la necessità di mantenere l' ortogonalità fra le sottoportanti, altrimenti cominciano fenomeni non desiderati come l'ICI (Inter Carrier Interference) oppure sfasature del segnale modulato.

- **tipologie di sottoportanti** Le sottoportanti in OFDM sono classificabili in tre tipologie, La maggior parte sono adibite al trasporto delle informazioni, Le restanti sono utilizzate per l'invio di simboli pilota. Esiste un'ultima sottoportante particolare che non trasmette nulla (nemmeno la frequenza portante) situata a metà fra tutte le altre che serve al ricevitore per sapere l'esatto centro dell' intera banda.

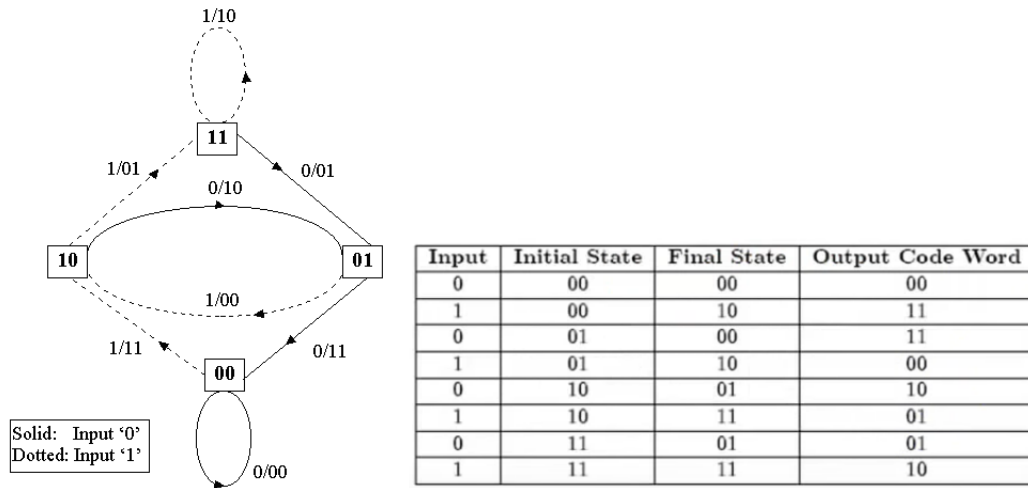


Figura 2.5: diagramma a stati e tabella utilizzati per un implementazione di un codificatore CC con output di lunghezza doppia rispetto all'input.



Figura 2.6: Tipologie sottopottanti OFDM [26]

- **trasmissione Preambolo** L'invio di informazioni in OFDM è preceduto da informazioni che vengono utilizzate per la sincronizzazione dal ricevitore. Queste informazioni includono un preambolo e in alcune applicazioni un simbolo contenente alcuni parametri come il numero di simboli OFDM in arrivo.

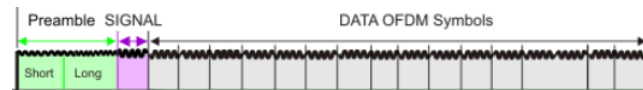


Figura 2.7: Sequenza invio OFDM [26]

Le sfasature in frequenza si dividono principalmente in due tipi, la prima è detta CFO (Carrier Frequency Offset) e rappresenta la sfasatura rispetto alle sottoportanti tra trasmettitore e ricevitore mentre la seconda è nota come SFO (Sampling Frequency Offset) e indica l'errore nella frequenza di campionamento. Il problema del CFO si manifesta in una sfasatura del campionamento dei simboli riceviti e viene corretto sincronizzando il ricevitore utilizzando il preambolo (la sua lunghezza determina la precisione). L'SFO si manifesta in una sfasatura dei punti sulla costellazione e corretto utilizzando i simboli pilota sempre presenti in alcune specifiche sottoportanti. [12]

Proprietà e campi di utilizzo

OFDM grazie all'ortogonalità delle portanti permette di avere un'ottima efficienza sull'utilizzo della banda, inoltre la scelta statica o dinamica del tipo di modulazione da utilizzare in ogni sottoportante lo rende adatto sia in situazioni dove è presente un canale molto distorto che in casi in cui è possibile raggiungere un elevato throughput. Grazie ai tempi di guardia fra i simboli trasmessi e al prefisso ciclico (Cyclic prefix) OFDM risulta robusto contro il problema della sovrapposizione sul segnale in ricezione di componenti provenienti da segnali riflessi (multipath propagation). OFDM richiede un'ottima sincronizzazione e risulta quindi sensibile all'effetto

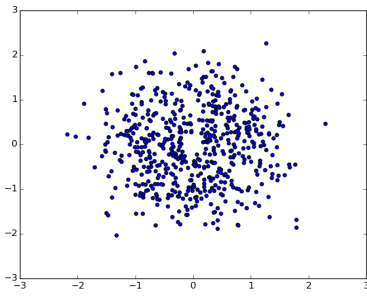


Figura 2.8: Costellazione con dati grezzi ricevuti 16QAM

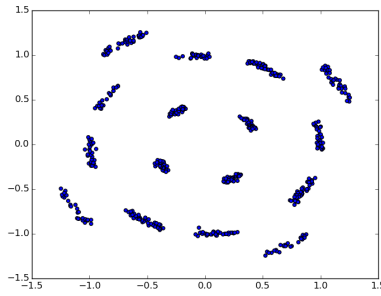


Figura 2.9: Corretti da CFO con preamble corto

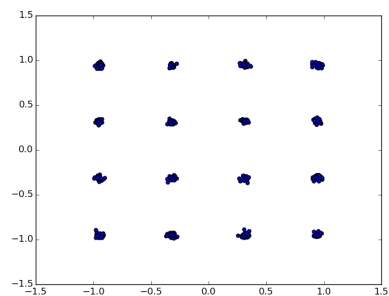


Figura 2.10: Corretti da CFO e SFO

[12]

Doppler. OFDM soffre inoltre di un elevato PAPR (peak to average power ratio) causato dalla caratteristica di avere le sottoportanti in quadratura. L'ortogonalità garantisce di non avere sovrapposizioni sulla frequenza dove si trasmette un simbolo ma ciò non si verifica nelle frequenze intermedie fra una sottoportante e l'adiacente. Accade così che in particolari circostanze le sfasature fra i simboli su sottoportanti adiacenti finiscano per sommarsi creando un picco ben superiore alla media. Questo fenomeno influisce nel dimensionamento degli apparati che devono essere scelti per non saturare il segnale ma allo stesso tempo che non siano sprecati funzionando sotto alla metà della potenza. [18]

- **ADSL** la connessione adsl avviene mediante doppioli lunghi anche qualche chilometro. Il doppiolo di rame presenta fisicamente una resistenza (prevedibile con la seconda legge di ohm) che è posizionata in serie al segnale, inoltre il doppiolo possiede un induttanza. Quando è presente un segnale (corrente alternata) si comporta come un filtro che attenua il segnale, tali effetti si amplificano all'aumentare della distanza e della frequenza. OFDM viene utilizzato in questo campo proprio perchè non risente molto della differenza di attenuazione nelle sottoparti del canale trasmissivo.
- **Powerline** i dispositivi powerline utilizzano l'impianto elettrico come mezzo trasmissivo, viene utilizzato OFDM per la presenza di un canale molto variabile e soggetto da disturbi esterni non prevedibili.
- **Wlan, WiMAX** OFDM viene utilizzato in due fra i principali standard per la trasmissione di internet senza fili. fornisce robustezza contro il problema della multipath propagation oltre ad un ottimo range di scelta sulle modulazioni che si presta bene sia per situazioni di bassa qualità del canale sia in situazioni di stabilità dove si vuole ottenere un buon bandwidth.
- **Radio e televisione digitali** La televisione pubblica italiana come molte di quelle europee vengono trasmesse secondo lo standard DVB-T(Digital Video Broadcasting-Terrestrial) che sfrutta OFDM per inviare un flusso contenente i vari canali televisivi già compressi e provvisti di trame per la decodifica. La radio digitale DAB(Digital Audio Broadcasting) suddivide invece le stazioni in blocchi contenenti una decina di radio ognuno. Ogni blocco viene poi trasmesso utilizzando OFDM

SDR (Software Defined Radio)

Tradizionalmente gli apparati per le telecomunicazioni vengono implementati in hardware, lo sviluppo risulta molto costoso finendo per essere svolto da poche persone. Progettare in hardware richiede molto tempo ed il risultato è un sistema affidabile ma con un compito specifico e difficile da aggiornare o modificare una volta prodotto. Recentemente con l'aumento della potenza di calcolo è finalmente possibile svolgere con il software compiti precedentemente svolti da hardware specializzato. L'SDR è una scheda che contiene l'hardware aggiuntivo necessario ad un computer per poter ricevere e/o trasmettere informazioni. Un grande vantaggio dell' utilizzo di queste piattaforme è la possibilità di implementare la tecnica di trasmissione favorita con inoltre la possibilità di variare tutti i parametri tecnici (es. frequenza, larghezza di banda, frequenza di campionamento, ecc.). Un applicazione interessante di questa tecnologia è la creazione di un sistema dinamico in grado di far variare la frequenza ed il metodo di trasmissione per adattarsi alla situazione presente nel migliore dei modi. Esistono diverse tipologie di SDR, i più economici costano appena una decina di euro e seppure con qualche limitazione riescono a ricevere fino a quasi 2GHz, versioni più costose sono in grado anche di trasmettere contemporaneamente su un range di frequenza e sample rate più elevati. Da sottolineare che ogni diversa frequenza su cui si intende trasmettere-ricevere richiede una specifica antenna e che non esiste un' antenna generica. Il costo per una scheda SDR da laboratorio si aggira dai 500 ai 2000 euro ma è destinato a scendere visto che il suo vero valore tenendo conto anche ricerca e progettazione è stimato essere un quarto. [23]

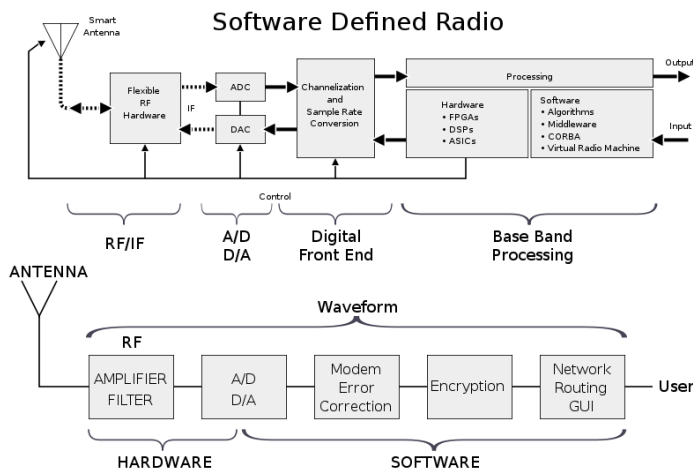


Figura 3.1: Diagramma blocchi funzionamento SDR [7]

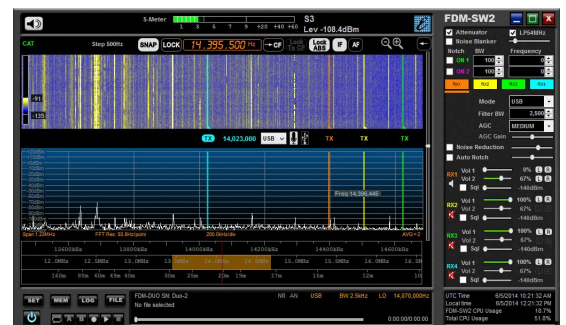


Figura 3.2: Software generico per l'analisi dello spettro [24]

• 3.0.1 USRP (Universal Software Radio Peripheral)

USRP è una tipologia di SDR venduta dal marchio Ettus Research pensata per essere accessibile a tutti. Alcune versioni contengono un processore su cui può essere caricato del software per funzionare in autonomia. Sono disponibili anche modelli con una scheda ethernet integrata per il controllo remoto attraverso una rete locale o remota. Gli USRP permettono la trasmissione e la ricezione contemporaneamente e dispongono di un ampio range di frequenze che varia in relazione al modello ma è di gran lunga più elevato rispetto agli sdr-rtl. Le schede USRP sono

utilizzabili con il driver UHD disponibile come due blocchi uno per la ricezione ed uno per la trasmissione pronti per essere integrati nel proprio flusso su GNURadio.

Ettus USRP-B210

Questo modello del costo di 1100 euro in dotazione ai laboratori dell' università permette di variare la frequenza da 70MHz a 6GHz e raggiungere una frequenza di campionamento di 56MHz. Supporta MIMO in full-duplex

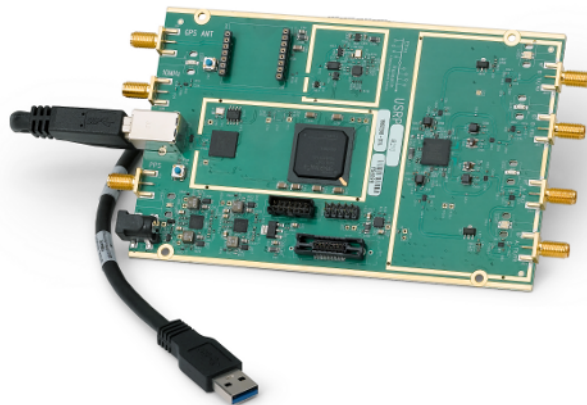


Figura 3.3: USRP modello B210[27]

• 3.0.2 RTL-SDR

Questa particolare tipologia di SDR sono i più economici presenti sul mercato, vengono venduti come decoder per lo standard DVB-T della televisione digitale e per DAB e FM della radio. Utilizzando driver alternativi è possibile ricevere un flusso di campionamenti dalla scheda. Questa tipologia di SDR è solo in grado di ricevere il segnale, inoltre presenta limitazioni sia sulla frequenza di ricezione (minore di 2GHz) che sulla frequenza di campionamento (minore di 2MHz per una lettura affidabile, fino a 3Mhz).



Figura 3.4: RTL-SDR venduto come ricevitore DVB-T, DAB e FM[22]

GNURADIO

GNU Radio è una piattaforma gratis e open-source per lo sviluppo di codice finalizzato all'implementazione di software radio, lo sviluppo può essere eseguito sia utilizzando schede hardware esterne oppure simulando lo scenario virtualmente. Il progetto GnuRadio è nato nel 2001 con l'idea di portare il concetto di free-software anche nel mondo delle Software Defined Radios visto che in precedenza questo settore era dominato solo da software proprietari. Attualmente è molto utilizzato sia in ambito accademico che commerciale, negli ultimi anche nel mondo hobbistico. Lo sviluppo in GnuRadio consiste nella disposizione di una serie di blocchi ciascuno dei quali svolge un'operazione ben precisa, il collegamento fra i blocchi è monodirezionale e rappresenta il flusso delle informazioni dalla sorgente al pozzo finale.

Il programma ufficiale dispone di molti moduli per l'elaborazione delle informazioni e l'interazione esterna, tuttavia è possibile creare un proprio blocco da inserire nel flusso in linguaggio python oppure C++. Per creare un blocco personalizzato GnuRadio mette a disposizione pyBOMBS il cui compito è quello di preparare l'ambiente per lo sviluppo e l'integrazione del blocco scaricando e configurando le librerie necessarie senza che l'utente debba preoccuparsene. Per la creazione di tutti i file necessari per il funzionamento di un blocco viene fornita l'utility `gr_modtool` che attraverso il terminale chiede all'utente i parametri necessari. Un modulo Gnu-radio deve possedere delle caratteristiche per il funzionamento. Ogni blocco deve specificare che tipologia di valori si aspetta in ingresso ed in uscita (Int, Float, Complex, ecc.), quanti ingressi e uscite fornirà, la lista dei parametri da richiedere all'utente per l'utilizzo e il rapporto fra il numero di campioni in ingresso e quello in uscita. I blocchi dunque devono avere un rapporto costante fra input e output rappresentato da una costante. `gr_modtool` può creare se lo sviluppatore lo desidera un file python per il testing (Quality assurance), eseguendo il testing viene simulato un semplice diagramma come se fosse stato creato nell'ambiente grafico, questo diagramma è personalizzabile e permette di aggiungere tutti i blocchi necessari. Solitamente è sufficiente avere un diagramma composto da tre blocchi: il primo per fornire le informazioni al blocco di test, il blocco di test stesso ed infine un blocco che ritorna i risultati ottenuti permettendo al codice di confrontarli con quelli desiderati specificati dal programmatore. Una volta completato lo sviluppo è possibile compilare il blocco per renderlo utilizzabile all'interno dell'ambiente GNURadio.

Crittografia RSA

l'algoritmo RSA (Rivest-Shamir-Adleman) è un algoritmo per la crittazione ampiamente utilizzato che basa il suo funzionamento sulla difficoltà di fattorizzare un numero generato moltiplicando due numeri primi grandi. La base matematica dell'algoritmo venne pubblicata nel 1976 da due matematici Diffie e Hellman famosi per aver inventato l'algoritmo Diffie-Hellman utilizzato ancora oggi per instaurare una crittografia a chiave simmetrica ma senza la trasmissione della chiave. L'algoritmo RSA venne pubblicato nel 1977. L'algoritmo era già stato segretamente documentato da un militare britannico qualche anno prima ma venne mantenuta la notizia segreta fino al 1997.

Caratteristiche

– 5.1.1 Codifica asimmetrica con doppia chiave

Le due chiavi sono dette privata e pubblica e vengono generate dallo stesso dispositivo, poi viene pubblicata solo quella pubblica. RSA è un algoritmo a chiave asimmetrica che utilizza quindi due chiavi distinte per la procedura di codifica e decodifica, a differenza degli algoritmi a chiave privata condivisa in rsa la chiave privata è posseduta solo da uno dei due soggetti della comunicazione rendendone più difficile l'ottenimento da parte di un eventuale attaccante.[20]

– 5.1.2 Algoritmo unico

RSA utilizza lo stesso algoritmo per la codifica e la decodifica delle informazioni.[20]

– 5.1.3 Chiavi intercambiabili

E' possibile utilizzare le chiavi nell'ordine preferito, ad esempio durante l'invio di un messaggio il trasmettitore lo codificherà con la chiave pubblica del ricevitore mentre per le firme digitali il trasmettitore cripterà un hash del messaggio con la propria chiave privata permettendo al ricevitore di verificare l'identità.[20]

– 5.1.4 Procedimento creazione chiave monodirezionale

E' computazionalmente semplice generare la chiave pubblica partendo da quella privata mentre il contrario è proibitivo. Non è matematicamente dimostrata l'impossibilità della scoperta di un algoritmo che renda la procedura inversa efficiente.[20]

Algoritmo

Generazione delle chiavi

Il primo passo consiste nel generare la chiave privata e quella pubblica. Tutta la seguente procedura viene effettuata solo su un dispositivo.

- vengono scelti due numeri primi molto grandi con lunghezza simile e ne viene eseguita il prodotto $n = p * q$, n sarà il modulo utilizzato nell' algoritmo di codifica/decodifica
- viene calcolato $f(n) = (q - 1) * (p - 1)$
- viene scelto un numero e tale che $1 < e < f(n)$ e che $MCD(e, f(n)) = 1$
- viene calcolato $d = e^{-1} \bmod f(n)$ utilizzando l'algoritmo di euclide

La chiave pubblica sarà formata dalla coppia (e,n) mentre quella privata (d,n). [21]

Cifratura

La cifratura verrà eseguita dal mittente utilizzando la chiave pubblica resa nota dal destinatario (e,n) calcolando $C = M^e \bmod n$. [21]

Decifratura

La decifratura verrà eseguita dal destinatario con la propria chiave privata (d,n) decodificando il messaggio $M = C^d \bmod n$. [21]

OFDM in Gnu Radio

La prima parte del lavoro svolto è consistita nell'implementazione di OFDM nell'ambiente Gnu-Radio. Il funzionamento di OFDM necessita di vari meccanismi complessi come l'assegnazione di informazioni alle sottoportanti, le correzioni in frequenza, l'equalizzazione e la correzione d'errore come precedentemente spiegato nella parte teorica. Lo svolgimento di queste operazioni sono rese possibili dalla presenza sia di blocchi generici utili ad esempio per la correzione d'errore che di blocchi disponibili specificamente per l'implementazione di OFDM. Per una comunicazione standard OFDM in Gnuradio non è necessaria la scrittura di algoritmi, il lavoro consiste nel collegamento e nella configurazione dei parametri al fine di fare comunicare tutto nella maniera corretta. La comunicazione è divisa in due parti, una per la trasmissione che ha il compito di generare campioni per il driver dell'usrp-sdr ed una per la ricezione che partendo dai campionamenti effettuati dall'rtl-sdr decodifica le informazioni. Di seguito verranno documentate parte per parte tutte le fasi necessarie. E' stato scelto di trasferire un file audio in formato wma perchè non effettuando compressioni non viene influenzato dalla mancanza di campionamenti. L'unico problema si verifica quando i campioni persi fanno parte della piccola sezione riservata ai metadati. Un pacchetto perso risulterà in circa mezzo millisecondo di audio mancante. La frequenza scelta per la trasmissione è 915Mhz che risulta disponibile per uso civile [1], la scelta non ha potuto comprendere 2.4ghz e 5ghz visto che l'Sdr economico utilizzato in ricezione arriva a massimo 2Ghz.

Trasmittitore

La trasmissione è composta da quattro sezioni principali connesse fra loro.

– 6.1.1 Formazione pacchetti dal flusso in ingresso

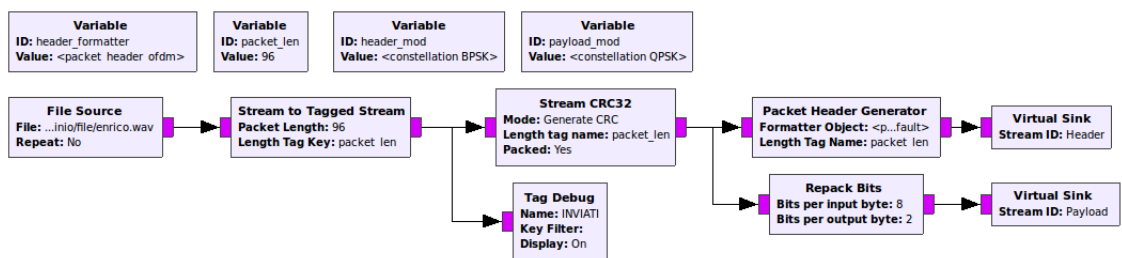


Figura 6.1:

Le informazioni vengono lette da file e fornite dal blocco "File Source" 8bit alla volta, questi campioni poi vengono aggregati 96 alla volta tramite l'aggiunta di un tag. A questo punto il blocco "CRC32" appende il codice di controllo per permettere al ricevitore di verificare la presenza di errori. I campioni a questo punto vengono processati sia dal blocco "Repack bits" che ha il compito di generare quattro nuovi byte con soli due bit dell'input (la costellazione qpsk permette l'invio di 2 bit alla volta quindi i 6 restanti lasciati a zero e verranno ignorati) che dal blocco "Packet Header generator". Quest'ultimo ha il compito di generare gli header dei pacchetti che permettono al ricevitore di riconoscere l'inizio di un blocco di 96byte di informazione. Gnuradio mette a disposizione degli oggetti "costellation" che contengono informazioni utili relative alle caratteristiche delle costellazioni, ad esempio il numero di bit significativi per la costellazione oppure la mappatura fra bit e punti delle costellazioni (Ad esempio l'oggetto digital.constellation_qpsk viene utilizzato nel modulo

”Repack bits” chiamando la funzione `payload_mod.bits_per_symbol()` che restituisce 2). Le informazioni vengono poi trasferite a due ”Virtual Sink” che fungono solamente da collegamento logico verso i ”Virtual Source”, il loro scopo è solamente quello di rendere la composizione più leggibile.

– 6.1.2 Codifica informazioni in Simboli

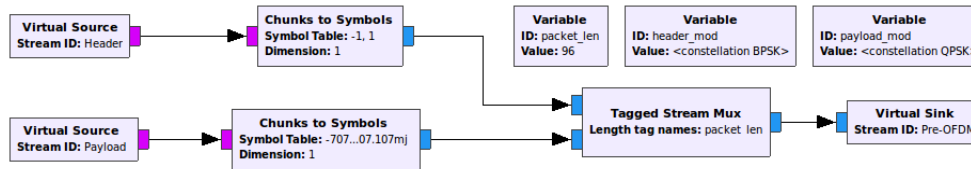


Figura 6.2:

Questa sezione del trasmettitore riceve due flussi dalla precedente contenenti i bit divisi fra header e payload. Il blocco ”Chunks to Symbols” convertono i bit (che ricordiamo sono già nel formato 1 bit significativo per byte per l’ header e 2 bit significativi per byte per il payload) in numeri complessi della rispettiva costellazione. La mappatura fra bit significativi e numeri complessi viene fornita dall’oggetto di appoggio disponibile per le varie modulazioni fornito da gnuradio descritto brevemente nel punto precedente utilizzato chiamando rispettivamente le funzioni `payload_mod.points()` e `header_mod.points()`. A questo punto è necessario unire in un solo flusso (mantenendo la divisione fedele ai blocchi iniziali) i punti delle costellazioni ottenuti ricalcolando il tag della lunghezza, questo lavoro viene eseguito dal blocco ”Tagged Stream Mux”.

– 6.1.3 Allocazione simboli sulle sottoportanti e creazione campioni per l’SDR

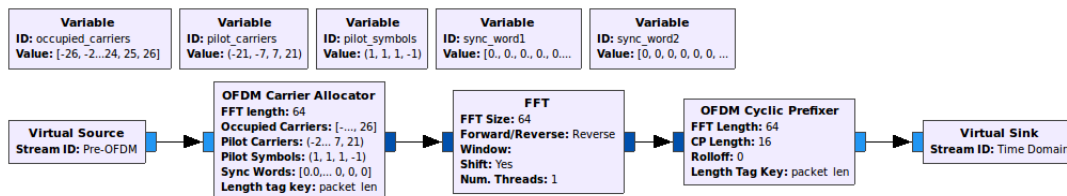


Figura 6.3:

L’ obiettivo di questa sezione è quello di allocare sulle sottoportanti di OFDM le informazioni e creare i campioni da spedire nel dominio del tempo. Il blocco ”OFDM Carrier Allocator” si occupa di allocare i punti delle costellazioni e i simboli pilota (pilot_symbols) rispettivamente alle sottoportanti definite come `occupied_carriers` e `pilot_carriers`. Inoltre aggiunge all’inizio di ogni blocco trasmesso i `sync_words` utilizzati dal ricevitore per sincronizzazione ed equalizzazione. L’output rappresenta un simbolo OFDM pronto per essere portato nel dominio del tempo dal blocco ”FFT” che esegue appunto la trasformata di Fourier veloce. Il blocco ”OFDM Cyclic Prefixer” aggiunge il prefisso ciclico spiegato in precedenza.

– 6.1.4 Passaggio dati all’SDR

L’ultima operazione consiste nel passare il simbolo OFDM al driver dell’SDR configurandolo con i parametri necessari per la trasmissione. Il blocco ”WX GUI FFT Sink” serve per avere un feedback sulla trasmissione mostrando un grafico sul dominio delle frequenze.

Ricevitore

La ricezione è composta da tre sezioni principali connesse fra loro.

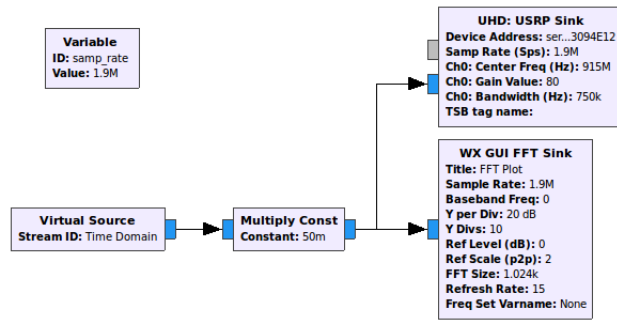


Figura 6.4:

– 6.2.1 Ricezione informazioni, sincronizzazione

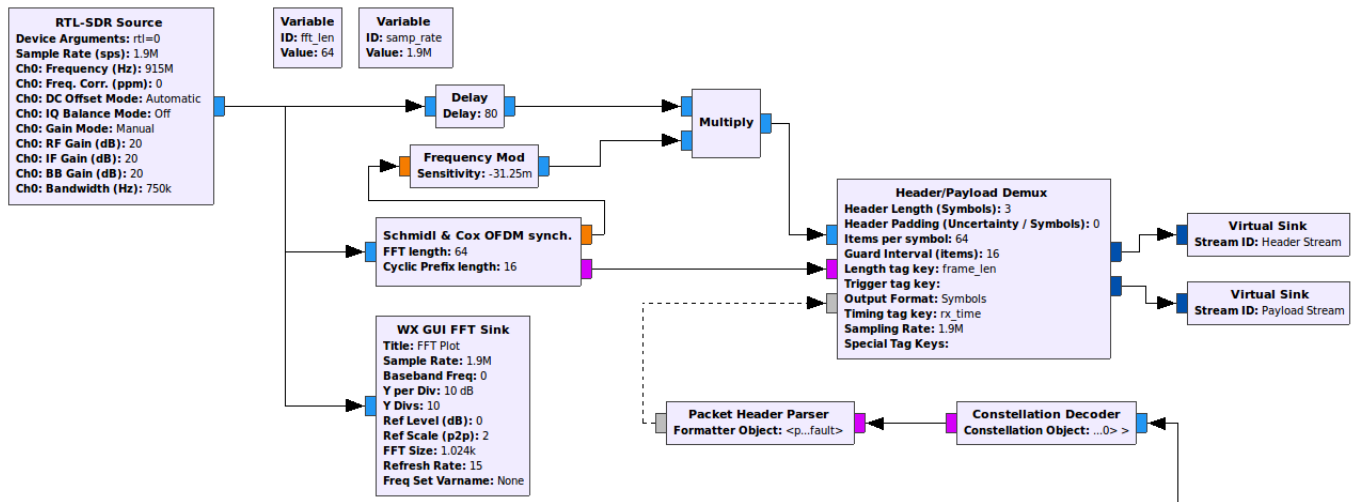


Figura 6.5:

Questa prima sezione del ricevitore ha il compito di leggere i campionamenti forniti dal driver per l'rtl-sdr e ottenere le informazioni nuovamente divise fra header e payload. Il blocco centrale di questa sezione è senzadubbio "Header/payload Demux" che ha proprio questo compito strettamente necessario per poi poter essere analizzati in seguito. I campionamenti sono forniti in forma complessa (I/Q) e rappresentano l'ampiezza del segnale in quell'istante ma anche l'andamento della funzione che lo ha generato. Il processo di ricezione ha inizio quando il blocco "Schmidl & Cox OFDM sync" individua l'inizio della trasmissione inviando un segnale di comando (trigger) verso il demuxer. Il suo funzionamento è molto complesso e permette assieme al blocco "Frequency mod" la sincronizzazione necessaria per l'inizio della lettura. Il blocco "Header/payload Demux" inoltra sulla prima uscita (Header Stream) i primi 3 simboli OFDM (composti da 64 campioni ognuno) e rimane in attesa fino a che non gli vengono fornite le informazioni contenute nell'header. L'analisi dei campioni contenenti le informazioni sul payload verrà approfondita nella sezione successiva, per ora è importante notare che "Header/payload Demux" riceve queste informazioni sul terzo ingresso. A questo punto il demuxer grazie alle informazioni dell'header può inoltrare il numero corretto di simboli OFDM sulla seconda uscita (Payload Stream).

– 6.2.2 Equalizzazione e ottenimento simboli

L'obiettivo di questa sezione è quello di ottenere dai campionamenti ricevuti le costellazioni delle sottoportanti OFDM. La procedura consiste nell'applicazione della trasformata di Fourier veloce per passare al dominio delle frequenze. Nella decodifica dell'header è presente il blocco "Channel Estimator" che sfruttando i primi due sync_words ha lo scopo di ottenere informazioni di partenza sulle caratteristiche di sfasatura CFO (Carrier Frequency Offset).

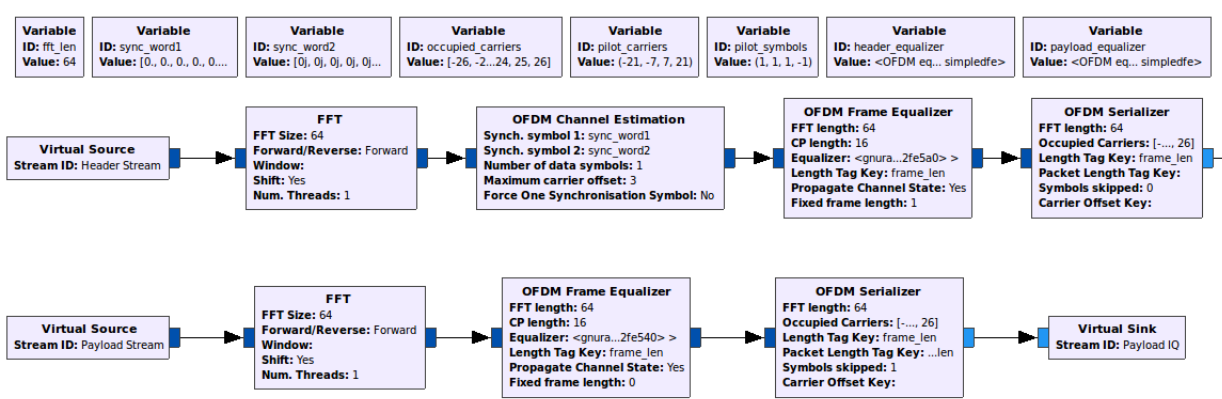


Figura 6.6:

e di attenuazione canale. Il blocco successivo "OFDM Frame Equalizer" utilizza queste informazioni per effettuare la prima equalizzazione, le informazioni sulle caratteristiche del canale vengono poi aggiornate alla ricezione di ogni simbolo OFDM grazie ai simboli pilota contenuti nelle apposite sottoportanti. L'ultima operazione della sezione viene svolta dal blocco "OFDM Serializer" e consiste nell'invertire il lavoro svolto dall'allocatore nella fase di trasmissione al fine di ottenere un flusso contenente solo i punti delle costellazioni che contengono informazioni in modo ordinato e raggruppati secondo pacchetto di trasmissione (mediante l'aggiunta di un tag).

– 6.2.3 Decodifica payload e scrittura file

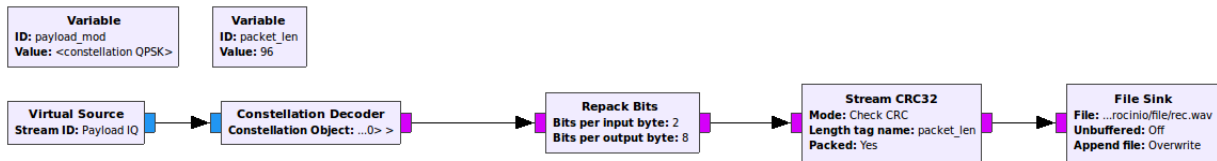


Figura 6.7:

Il blocco "Constellation Decoder" effettua la decodifica dei numeri complessi ricevuti. Le informazioni ottenute dalla decodifica di una singola costellazione in qpsk sono rappresentabili in 2 bit, è dunque necessario l'utilizzo del blocco "Repack bits" che preleva solo i due bit significativi di quattro decodifiche successive per completare un byte. Le informazioni ottenute contengono ancora informazioni sul blocco di appartenenza, "Stream CRC32" esegue la verifica pacchetto per pacchetto scartandolo se contenente errori. In conclusione il blocco "File Sink" si occupa della scrittura su file completando il processo di ricezione.

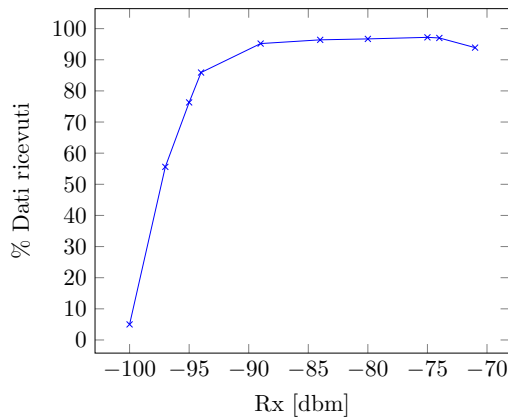
Testing

Sono stati effettuati alcuni test per osservare il comportamento in un ambiente reale. DA FARE!!! (decibel non calibrati, sdr con poco range su frequenza e sample rate)

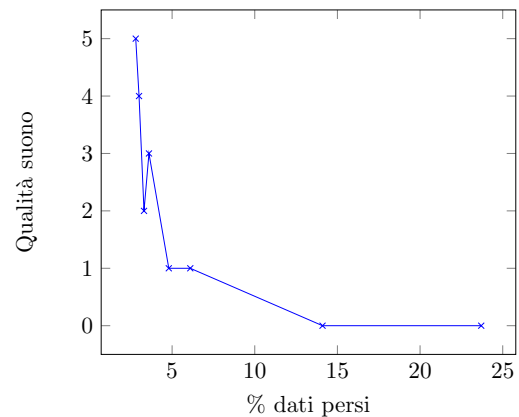
Implementazione Rsa in Gnuradio

La seconda parte del lavoro svolto è stata quella di valutare la possibilità dell'aggiunta di una crittografia RSA al sistema di trasmissione OFDM. Il primo passo è stato sviluppare un modulo che criptasse le informazioni passando poi all'implementazione di un secondo blocco per

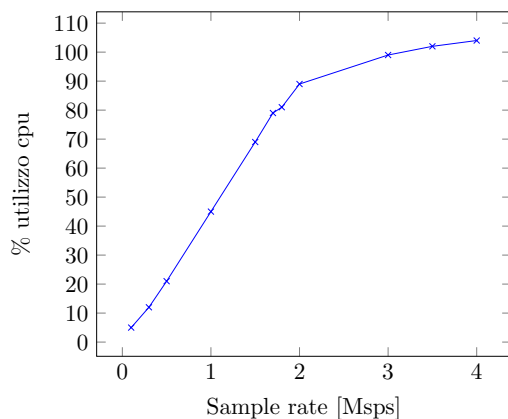
Dati ricevuti in base alla potenza del segnale ricevuto



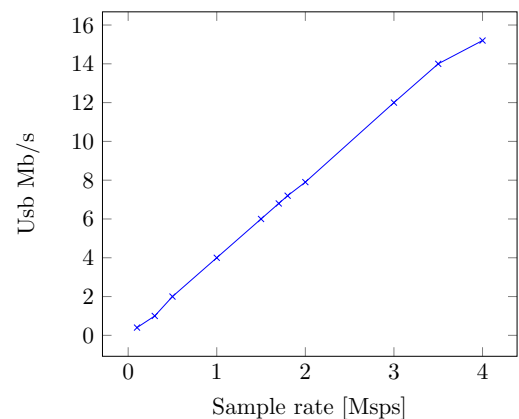
Qualità suono in relazione alla quantità di dati



Utilizzo cpu all' aumento del sample-rate



Utilizzo Usb all' aumento del sample-rate



la decodifica. Le chiavi utilizzate non sono assolutamente sicure, sono state scelte piccole solo per rendere facile la comprensione del funzionamento. In un applicazione reale sono necessarie chiavi di almeno 1024bit.

Sviluppo dei moduli

Lo sviluppo dei moduli è diviso in tre parti fondamentali descritte in seguito, per la creazione dei file di supporto e l'importazione delle librerie necessarie Gnuradio mette a disposizione un tool chiamato `gr.moodtool` utilizzabile da riga di comando.

Definizione proprietà moduli per l'ambiente grafico

Di seguito sono mostrati i file contenenti la definizione dei parametri necessari per l'interazione con l'ambiente. I blocchi sono costituiti da un ingresso per ricevere il flusso dati dal programma "Sink" ed un uscita per permettere la continuazione "Source", inoltre possono richiedere parametri aggiuntivi che devono essere forniti dall'utente nel formato corretto. Al codificatore è richiesto sapere la coppia di valori che costituiscono la chiave pubblica del decodificatore mentre per la decriptazione è sufficiente la chiave privata.

Algoritmo per cifratura e decifratura

La codifica e la decodifica rsa si effettua utilizzando lo stesso algoritmo e fornendo la chiave corretta. Nell'implementazione di rsa è importante non effettuare subito l'elevamento a potenza in quanto può originare numeri enormi, l'approccio migliore consiste nel moltiplicare la base n volte calcolando il modulo ad ogni passaggio. Il collegamento logico fra i blocchi all'interno dell'ambiente gnuradio al momento dell'esecuzione viene tradotto in un file python il cui compito è quello di gestire il flusso delle informazioni istanziando le classi dei blocchi necessari e succes-

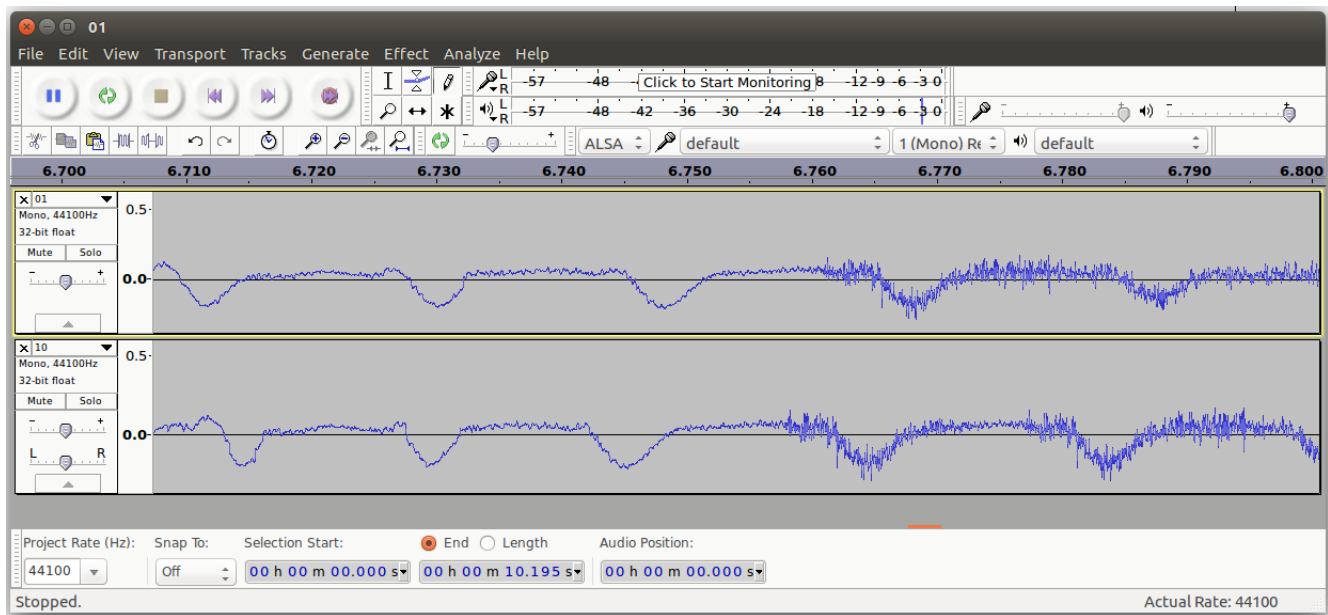


Figura 6.8: Confronto fra l'audio originale e quello ricevuto

sivamente chiamare il loro metodo "work" presente in ogni modulo. Lo sviluppo del proprio algoritmo deve essere implementato proprio all'interno della funzione work. Le informazioni vengono trasferite attraverso un vettore di vettori. il primo indice rappresenta la numerazione dell'ingresso o dell'uscita mentre il secondo permette di gestire l'arrivo di più informazioni alla volta. Il blocco codificatore riceve 8 bit alla volta che vengono tradotti in python con il complemento a due. il risultato della cifrazione non risulta più rappresentabile con 8bit e per la nostra analisi di fattibilità effettuata con chiavi deboli è stato sufficiente l'utilizzo di una variabile intera. L'applicazione in uno scenario reale implica la risoluzione di problematiche discusse nella sezione delle conclusioni. Il blocco Decodificatore riceve le informazioni codificate e le riporta nel formato originale.

Codice di test

Gnuradio permette la creazione di casi di test come spiegato nella sezione di introduzione teorica, il testing è particolarmente utile in scenari come quello presente dove un eventuale errore nell'algoritmo verrebbe propagato ai blocchi successivi senza produrre segnalazioni. Le funzioni di testing sono utili anche durante lo sviluppo in quanto permettono di provare il codice scritto senza dover compilare tutto il modulo per renderlo disponibile nell'ambiente Gnuradio. Il testing viene effettuato istanziando la classe del blocco in analisi collegandolo ad una sorgente dati che si occuperà di passare i dati specificati ed un pozzo per riceverli una volta processati. Vengono inoltre impostati i parametri richiesti dal modulo (nel nostro caso riguardano i due valori di cui sono composte le chiavi). Infine vengono confrontati i risultati utilizzando delle funzioni apposite dette "assert" che restituiscono un messaggio di errore se falliscono. Nel testing del blocco di codifica i valori scelti (128,255,0,7,127) coprono tutti i casi limite del complemento a due.

Importazione in Gnuradio

Il passo finale consiste nel compilare i moduli e farli funzionare nell'ambiente grafico Gnuradio. Nella console inclusa in basso nell'interfaccia utente vengono stampati dei feedback generati dai moduli durante il loro lavoro. L'esempio riportato trasferisce un testo cifrandolo. Questo modello non è tuttavia ancora applicabile ad una comunicazione reale per le motivazioni discusse nelle conclusioni.

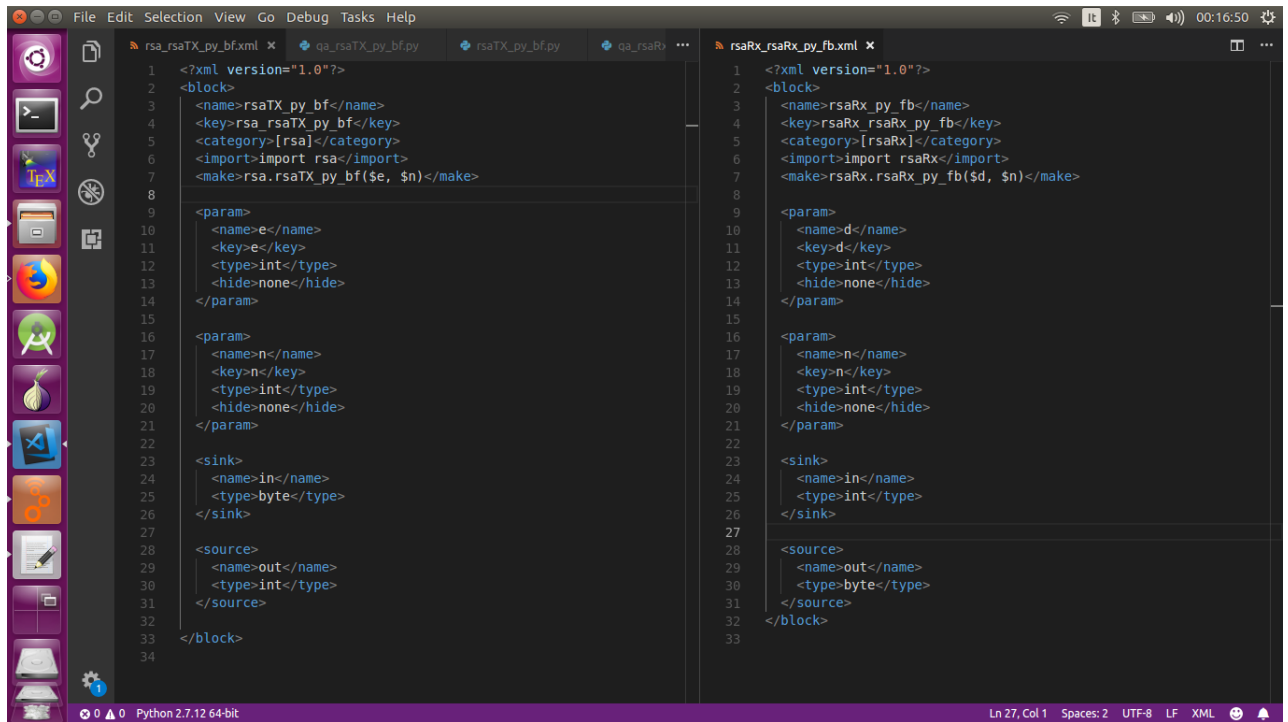


Figura 7.1: File contenenti la definizione dei parametri nell' ambiente grafico

```
def work(self, input_items, output_items):
    in0 = input_items[0]
    out = output_items[0]
    for j in range(0, len(in0)):
        init = int(0)
        init = long(in0[j])
        if long(in0[j]) < 0:
            init = ((init)%self.n_public_key) + 256
        res = init
        for count in range(1, self.e_public_key):
            res = (res*init)%self.n_public_key
        print ("messaggio codificato (tx): " + str(res) + " da: " + str(in0[j]+256 if in0[j] < 0 else in0[j]))
        out[j] = int(res)
    return len(output_items[0])
```

Figura 7.2: Algoritmo per il blocco di codifica

Conclusioni

Also, the worst performance issue with arithmetic cryptography like RSA is not about CPU, but about size. When you encrypt a piece of data with RSA, you get some overhead. E.g. with a 1024-bit RSA key, you can encrypt at most 117 bytes at a time, but this yields 128 bytes. When transmitting a long message, a fixed overhead of a few hundred bytes is much more tolerable than a +10% size increase.

CPU efficiency is often quoted as the reason we do hybrid encryption, but it is rarely as


```
def work(self, input_items, output_items):
    in0 = input_items[0]
    out = output_items[0]

    for i in range(0, len(in0)):
        resi = long(in0[i])%self.n
        res = resi
        for count in range(1, self.d):
            res = (res*(in0[i]%self.n))%self.n
        print ("messaggio decodificato (rx): " + str(res) + " da " + str(resi))
        out[i] = int(res)
    return len(output_items[0])
```

Figura 7.3: Algoritmo per il blocco di decodifica

```
def test_001_t (self):
    src_data = (128,255,0,7,127)
    expected_result = (863,606,0,343,187)
    src = blocks.vector_source_b (src_data)
    mult = rsaTX_py_bf (3,943)
    snk = blocks.vector_sink_i ()
    self.tb.connect (src, mult)
    self.tb.connect (mult, snk)
    self.tb.run ()
    result_data = snk.data ()
    self.assertEqual(expected_result, result_data)
```

Figura 7.4: Codice di test per il blocco di codifica

compelling as the two problems explained above.

Bibliografia

- [1] Allocazione-frequenze. <http://www.mise.gov.it/index.php/it/comunicazioni/radio/pnrf-piano-nazionale-di-ripartizione-delle-frequenze>.
- [2] Ask. https://it.wikipedia.org/wiki/Amplitude-shift_keying.
- [3] Constellation. <https://aerospaceresearch.net/?p=825>.
- [4] Convolutional-coding. <http://www.tdm.uni-oldenburg.de/2004/Material/faltung.htm>.
- [5] Crc. https://it.wikipedia.org/wiki/Cyclic_redundancy_check.
- [6] cyclic-prefix. http://www.sharetechnote.com/html/Communication_OFDM.html.
- [7] Diagramma-sdr. https://it.wikipedia.org/wiki/Software_defined_radio#/media/File:SDR_et_WF.svg.
- [8] Dpsk. https://www.tutorialspoint.com/digital_communication/digital_communication_differential_phase_shift_keying.htm.
- [9] Effetto-doppler. <https://soundwavesreillymckennaaly.weebly.com/doppler-effect.html>.
- [10] Efficienza-spettrale. https://it.wikipedia.org/wiki/Efficienza_spettrale.
- [11] Equalizzazione. <https://pdfs.semanticscholar.org/d8f6/7c4e4a42164bed115af4610d22adc78afec3.pdf>.
- [12] Frequencyoffsetofdm. https://openofdm.readthedocs.io/en/latest/freq_offset.html.
- [13] Fsk. <http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-FSK.html>.
- [14] Modulazioni digitali. <http://www.digitaler-bos-funk.de/digital/digital.htm>.
- [15] Ofdm. <http://www.revolutionwifi.net/revolutionwifi/2015/3/how-ofdm-subcarriers-work>.
- [16] Ofdm. https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing#Orthogonality.

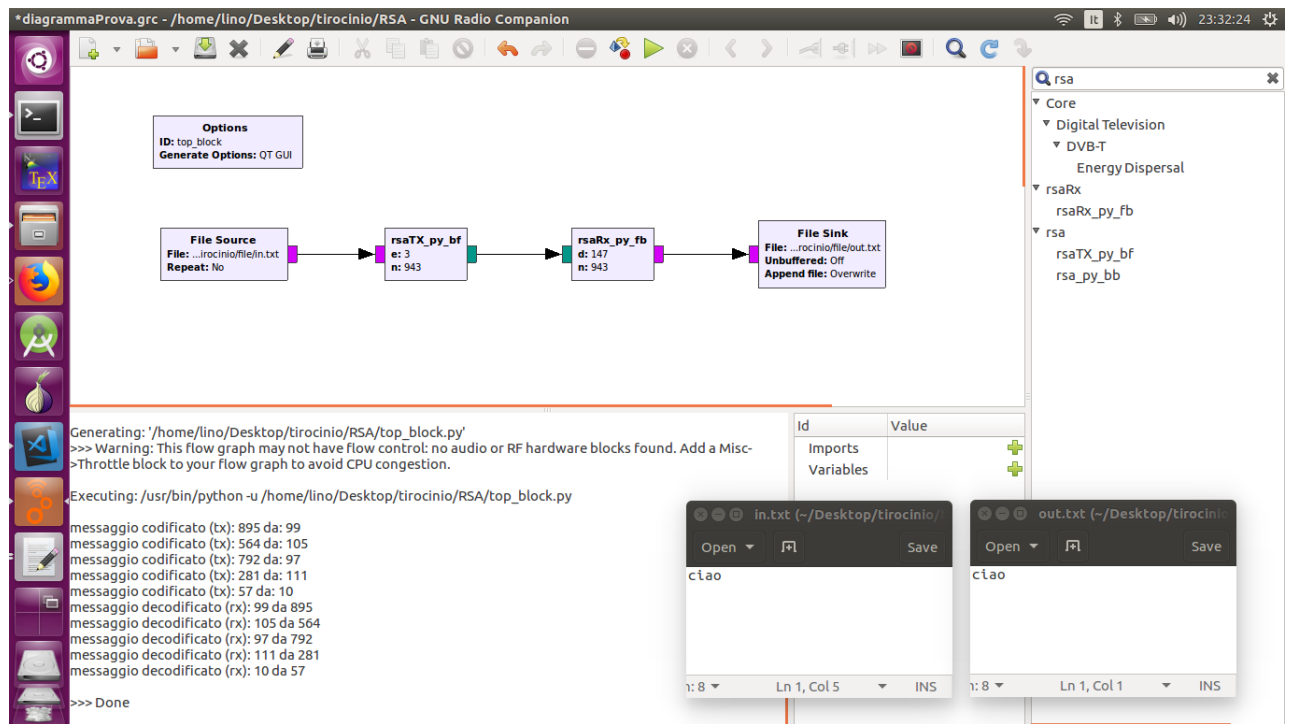


Figura 7.5: Codice di test per il blocco di codifica

- [17] Ofdm. https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing#Guard_interval_for_elimination_of_intersymbol_interference.
- [18] Papr. <http://www.techplayon.com/papr-peak-average-power-ratio-matters-power-amplifier/>.
- [19] Qam. <https://www.radio-electronics.com/info/rf-technology-design/quadrature-amplitude-modulation-qam/what-is-qam-tutorial.php>.
- [20] Rsa. <https://retiavanzate.eu/>.
- [21] Rsa. <http://www.di-srv.unisa.it/~ads/corso-security/www/CORSO-9900/rsa/testo.htm>.
- [22] Rtl-immagine. <https://shyamjos.com/assets/img/noaa/rtlsdr-dongle.jpg>.
- [23] Sdr-più-redditizio-della-droga. <https://zeptobars.com/en/read/AD9361-SDR-Analog-Devices-DAC-ADC-65nm>.
- [24] Sdr-software. https://www.wimo.com/elad-fdm-duo-sdr-transceiver_e.html.
- [25] sottoportanti-ofdm. <https://www.webnews.it/2008/11/24/tecnica-ofdm/>.
- [26] Subcarriers. <http://www.teletopix.org/4g-lte/lte-sub-carrier/>.
- [27] Usrp-b210. <https://www.ettus.com/product/details/UB210-KIT>.

Titolo primo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Titolo secondo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.