

Invio di dati modulati sull'udibile tramite radiofrequenza

Edoardo Pilia, Claudio Mellino

19 dicembre 2024

Introduzione

Questo progetto approfondisce il sistema Radioteletype (RTTY), una tecnologia di trasmissione dati che ha avuto un ruolo fondamentale nella storia delle comunicazioni. Nato come evoluzione delle telescriventi tradizionali, il RTTY si è distinto per la sua affidabilità, in particolare in ambiti militari, diplomatici e tra i radioamatori. Nel corso della relazione verranno trattate le origini storiche, il funzionamento tecnico e le applicazioni pratiche di questa tecnologia, ponendo attenzione alla modulazione AFSK (Audio Frequency-Shift Keying) e all'integrazione con strumenti moderni come Raspberry Pi e RTL-SDR. Verranno inoltre descritti i test sperimentali svolti, evidenziando le sfide affrontate e i risultati ottenuti.

Capitolo 1

Tecnologie utilizzate

1.1 Radioteletype - RTTY

Il Radioteletype (RTTY) è un sistema di telecomunicazioni che permette la trasmissione di dati tramite telescriventi interconnessi attraverso onde radio. Nato come evoluzione delle tecnologie di comunicazione via filo, il RTTY si è affermato come uno strumento chiave per la trasmissione di informazioni, sia in ambito civile che militare. Esso consente una comunicazione affidabile su lunghe distanze, rendendosi utile in situazioni in cui altre tecnologie risulterebbero inefficaci o inaccessibili. Il termine “Radioteletype” viene utilizzato per descrivere sia l’intera famiglia di sistemi basati su telescriventi e radio, sia il sistema originale basato sul codice Baudot.

Nonostante sia una tecnologia datata, oggi il RTTY mantiene una rilevanza specifica in alcune applicazioni, malgrado l’adozione di tecnologie più avanzate.

1.1.1 Storia e sviluppi tecnologici

Le origini del RTTY risalgono al 1849, quando venne installata la prima linea di telescrivente tra Filadelfia e New York. Nel 1874, Emile Baudot introdusse un sistema basato su un codice a cinque unità, che rimane alla base di molte delle tecnologie successive. Questo codice, noto come Codice Baudot (fig. 1.1), utilizza combinazioni di cinque bit per rappresentare caratteri alfanumerici e simboli, consentendo fino a 32 configurazioni distinte.

Per superare la limitazione del numero di caratteri rappresentabili, Baudot introdusse un sistema di “cambio di stato”: due modalità operative, lettere e numeri, attivate da caratteri di controllo specifici. Questa innovazione aumentò l’efficienza del sistema, permettendo di codificare un insieme più ampio di caratteri pur mantenendo la semplicità della trasmissione.

J. M. E. BAUDOT.

PRINTING TELEGRAPH.

No. 388,244.

Patented Aug. 21, 1888.

Fig. 2A.

	1	2	3	4	5
A	+	-	-	-	-
B	+	-	+	+	-
C	+	-	+	+	-
D	+	+	+	+	-
E	+	+	-	-	-
F	+	+	+	+	-
G	-	+	-	+	-
H	+	+	+	+	-
I	-	+	+	-	-
J	+	-	-	+	-
K	+	-	-	+	+
L	+	+	-	+	+
M	-	+	-	+	+
N	-	+	+	+	+
O	+	+	+	-	-
P	+	+	+	+	+
Q	+	+	+	+	+
R	-	-	+	+	+
S	-	-	+	-	+
T	+	-	+	-	+
U	+	+	+	-	+
V	+	+	+	-	+
W	-	+	+	-	+
X	-	+	+	-	+
Y	+	+	+	-	+
Z	+	-	-	+	+
0	-	-	-	+	+
1	-	-	-	+	+
2	-	-	-	+	+
3	-	-	-	+	+
4	-	-	-	+	+
5	-	-	-	+	+
6	-	-	-	+	+
7	-	-	-	+	+
8	-	-	-	+	+
9	-	-	-	+	+

INVENTOR:

*Jean Maurice Emile Baudot***Figura 1.1:** Rappresentazione del codice Baudot

Grazie alla sua struttura compatta e alla velocità di elaborazione, il Codice Baudot si affermò rapidamente come standard per i sistemi di telescrivente, influenzando profondamente lo sviluppo delle comunicazioni digitali.

Il RTTY è stato sviluppato come una naturale evoluzione delle comunicazioni via filo, trovando applicazione inizialmente in ambito commerciale. Già nel 1932, erano operativi collegamenti RTTY tra San Francisco e Honolulu, e nel 1934 tra San Francisco e New York. Durante la Seconda Guerra Mondiale, il sistema fu adottato su larga scala dall'esercito statunitense. La Marina sviluppò il sistema RATT (Radio Automatic Teletype), mentre l'Esercito utilizzò il sistema SCRT (Single-Channel Radiotelegraph), basati entrambi su tecnologie di modulazione a spostamento di frequenza (FSK). Questi sistemi garantirono comunicazioni sicure e affidabili su lunghe distanze, contribuendo in modo significativo agli sforzi bellici.

Negli anni '80, i telescriventi meccanici vennero gradualmente sostituiti da computer dotati di software di emulazione. Ciò ha semplificato notevolmente l'utilizzo del RTTY, riducendo il costo e la complessità dell'hardware

necessario. L'integrazione con i computer a quindi permesso l'introduzione di nuove funzioni, migliorando l'efficienza e la versatilità del sistema. Questi progressi hanno favorito l'adozione del RTTY anche in contesti amatoriali, mantenendo vivo l'interesse per gli appassionati anche nell'era digitale.

1.1.2 Struttura e funzionamento del sistema

Un sistema RTTY è composto da tre elementi fondamentali:

- Telescrivente: dispositivo elettromeccanico o elettronico per l'immissione e la visualizzazione di testo. Nei modelli più vecchi, venivano utilizzati lettori e perforatori di nastro per la memorizzazione dei dati. Questi strumenti garantiscono affidabilità anche in condizioni più estreme.
- Modem: dispositivo che converte i segnali digitali del telescrivente in segnali analogici modulati (AFSK) o segnali a spostamento di frequenza diretta (FSK). Questo componente è essenziale per interfacciare il telescrivente con il sistema radio.
- Radio: utilizzata per trasmettere e ricevere i segnali modulati. Le radio impiegate nei sistemi RTTY sono progettate per garantire una trasmissione stabile ed una ricezione chiara anche in presenza di interferenze.

Il RTTY utilizza un sistema asincrono di trasmissione basato sul codice Baudot, che rappresenta ogni carattere con una sequenza di 5 bit. A causa del numero limitato di combinazioni possibili (32 codici), vengono utilizzati due stati (lettere e numeri) per estendere l'insieme di caratteri disponibili. La modulazione più comune è l'FSK, in cui due toni audio rappresentano rispettivamente i segnali "mark" e "space". La velocità di trasmissione standard varia da 45.45 a 300 baud, con spostamenti di frequenza comuni tra 170 e 850 Hz. Questa configurazione garantisce un compromesso tra robustezza del segnale e larghezza di banda utilizzata, rendendo il sistema efficace in un'ampia varietà di contesti.

Con l'avvento dei computer moderni, le funzionalità del telescrivente e del modem sono state integrate in software e hardware digitali. Questo ha reso possibile l'utilizzo del RTTY su piattaforme più flessibili e accessibili, come i computer dotati di schede audio e software specifici (es. MMTTY). La digitalizzazione ha inoltre aperto la strada a ulteriori ottimizzazioni, inclusa l'automazione di alcune funzioni e l'integrazione con altre tecnologie di comunicazione.

1.1.3 Applicazioni e utilizzatori

Il RTTY è stato ampiamente utilizzato dai reparti militari e dai servizi diplomatici per le comunicazioni sicure. La robustezza del sistema contro le interferenze lo rende ideale per l'uso in condizioni operative difficili. Grazie alla sua semplicità e affidabilità, il RTTY è stato adottato per garantire il coordinamento in scenari strategici, dimostrando la sua utilità anche in situazioni di emergenza.

Un'applicazione significativa è la trasmissione di dati meteorologici. Ad esempio, il Deutscher Wetterdienst (DWD) trasmette regolarmente bollettini meteorologici utilizzando il RTTY su frequenze LF e HF. Questi dati sono essenziali per la navigazione marittima e aerea, oltre che per il monitoraggio delle condizioni climatiche in tempo reale.

Il RTTY è popolare tra i radioamatori, che lo utilizzano per stabilire comunicazioni. La semplicità del sistema e la sua resistenza alle interferenze lo rendono particolarmente apprezzato in questo ambito. Inoltre, il RTTY offre un'opportunità unica per esplorare le potenzialità delle comunicazioni radio, favorendo la sperimentazione e la condivisione di conoscenze tra appassionati.

1.1.4 Vantaggi e limiti del RTTY

Il RTTY si distingue per la sua resistenza alle interferenze e per la capacità di funzionare in condizioni radio difficili. Questo lo rende una scelta preferita in contesti dove l'affidabilità è cruciale. La sua semplicità strutturale e la capacità di operare su lunghe distanze senza una significativa perdita di qualità lo rendono una tecnologia ancora rilevante in ambiti specifici.

La principale limitazione del RTTY è la sua bassa velocità di trasmissione, tipicamente 45.45 baud (circa 60 parole al minuto). Questa caratteristica lo rende obsoleto rispetto alle tecnologie moderne più veloci, limitandone l'applicazione a contesti specifici. Tuttavia, in scenari dove la priorità è la robustezza del segnale piuttosto che la velocità, il RTTY mantiene un ruolo rilevante.

1.2 AFSK

Nell'ambito dei sistemi di comunicazione digitale, il Radioteletype (RTTY) e la modulazione Audio Frequency-Shift Keying (AFSK) rappresentano due tecnologie fondamentali per la trasmissione di dati in ambito radioamatoriale e oltre. Entrambe offrono soluzioni specifiche per esigenze di trasmissione affidabile, anche in condizioni tecnicamente complesse.

1.2.1 L'AFSK: Fondamenti e Applicazioni

L'Audio Frequency-Shift Keying è una tecnica che converte dati digitali in segnali audio attraverso due toni distinti: il "mark" e lo "space". In particolare, la specifica più nota, Bell 202, utilizza questo schema:

- Il bit '0' viene rappresentato da una frequenza di 2200 Hz.
- Il bit '1' viene rappresentato da una frequenza di 1200 Hz.

Questa modulazione è comunemente utilizzata nei sistemi radio amatoriali, come l'Automatic Packet Reporting System (APRS), e in vecchi modem per linee telefoniche. Nella Bell 202 si ha una comunicazione half-duplex a 1200 bit/s, che si adatta bene alle esigenze di trasmissione di dati di telemetria.

1.2.2 L'accoppiatore acustico

Un elemento che utilizza l'AFSK è l'accoppiatore acustico, un dispositivo che consentiva di interfacciare terminali digitali con linee telefoniche analogiche. L'accoppiatore convertiva segnali digitali in toni udibili e viceversa, anticipando l'approccio adottato dall'AFSK. Sebbene inizialmente utilizzato nei modem, questo metodo ha influenzato lo sviluppo delle tecnologie di modulazione digitale adattabili a infrastrutture esistenti, come il RTTY su radio analogiche.

1.3 Software-Defined Radio (SDR)

Il Software-Defined Radio (SDR) è una tecnologia innovativa che consente di implementare tramite software gran parte delle funzionalità tradizionalmente affidate a componenti hardware nei sistemi radio.

Un sistema SDR utilizza un hardware minimale per digitalizzare segnali analogici, demandando l'elaborazione dei segnali e la modulazione/demodulazione a un software eseguito su processori generici o DSP (Digital Signal Processor). Questo approccio permette una flessibilità senza precedenti, consentendo di:

- Supportare diversi protocolli di comunicazione semplicemente modificando il software.
- Sperimentare con tecniche avanzate di modulazione e demodulazione.
- Adattare rapidamente il sistema alle esigenze di nuove applicazioni o standard.

Grazie alla sua versatilità, l'SDR rappresenta una naturale evoluzione nel campo delle comunicazioni digitali, permettendo integrazione e interoperabilità tra tecnologie come RTTY, AFSK e altri protocolli radio.

Capitolo 2

Sperimentazione

2.1 Specifiche Tecniche

L'esperimento è stato condotto utilizzando la seguente strumentazione:

- Raspberry Pi 4, equipaggiato con un filtro passa-banda autocostruito (Figura 2.1);
- Ricevitore RTL-SDR V4 collegato a un computer per la ricezione del segnale;
- Sistema di trasmissione implementato tramite il Raspberry Pi, con uscita PWM (Pulse Width Modulation);
- Radio a valvole.

Il segnale trasmesso viene elaborato in ricezione tramite un software sviluppato in *Python*.

2.2 Tecnologia

2.2.1 Trasmissione

Il sistema di trasmissione utilizza un segnale AFSK (*Audio Frequency Shift Keying*) a due simboli, generato attraverso un programma Python dedicato. Il procedimento segue i seguenti passaggi:

1. **Generazione della forma d'onda:** Utilizzando la libreria **numpy**, viene generato un file **.wav** contenente la forma d'onda del segnale AFSK. Le frequenze scelte per la modulazione sono 2000 Hz per il

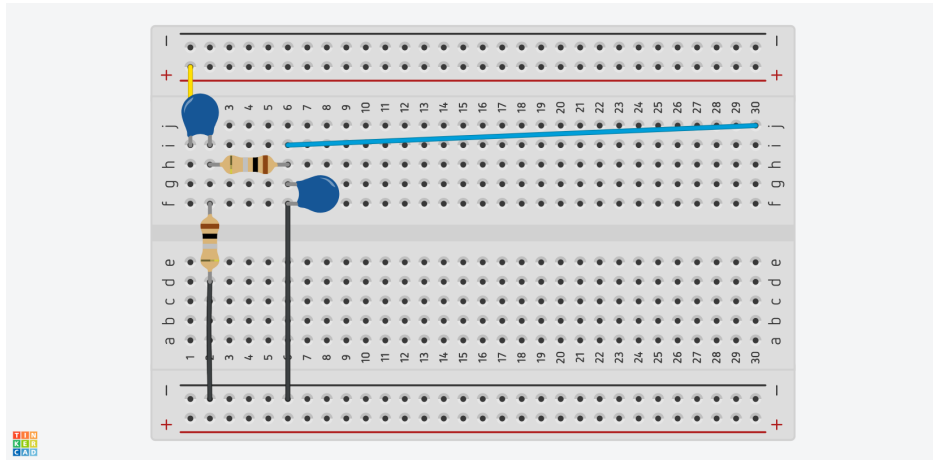


Figura 2.1: Filtro passa-banda

simbolo '0' e 6000 Hz per il simbolo '1', entrambe appartenenti alla gamma udibile. La frequenza di campionamento utilizzata è 22050 Hz e la durata di ogni simbolo è di 0.8 ms, con un symbol rate teorico di 1250 baud.

2. **Associazione bit-tono:** Ad ogni bit della sequenza da trasmettere viene associato un tono specifico. Questi toni vengono concatenati in un array e successivamente salvati nel file `.wav`.
3. **Trasmissione tramite PWM:** Il file audio viene trasmesso in modulazione di frequenza FM utilizzando il Raspberry Pi 4. L'uscita viene generata tramite PWM sul pin GPIO 4.

Tuttavia, l'utilizzo del PWM introduce una forma d'onda quadra, che costituisce solo un'approssimazione della sinusoide desiderata. Di conseguenza, il segnale trasmesso non filtrato è caratterizzato da armoniche indesiderate, le quali possono interferire con la ricezione.

2.2.2 Ricezione

La ricezione del segnale FM avviene attraverso il ricevitore RTL-SDR V4, collegato al computer. Il segnale viene quindi elaborato con il seguente metodo:

1. **Analisi del segnale tramite FFT:** Il programma Python esegue una trasformata di Fourier veloce (FFT) sul segnale ricevuto. La FFT consente di individuare quali componenti di frequenze sono presenti nel segnale.

2. **Decodifica del segnale:** In base alla frequenza rilevata dall'analisi FFT, viene associato il corrispondente valore di bit (1 o 0). Tale processo permette di ricostruire la sequenza trasmessa.

2.2.3 Codifica

Durante le prove, sono emerse difficoltà con l'utilizzo dei caratteri ASCII per quanto riguarda la conversione da binario a testo e viceversa. Dunque, allo scopo, si è preferito sviluppare una codifica a 8 bit così composta:

- I primi due bit rappresentano la tipologia del carattere;
- Il terzo bit rappresenta la parità;
- Gli ultimi cinque bit rappresentano il carattere.

Tramite questa codifica è possibile identificare errori singoli tramite il controllo del bit di parità.

2.3 Protocollo

Al fine di avere usi più ampi, è stata costruita una struttura a pacchetti per migliorare invio e ricezione dei messaggi. Il pacchetto è così composto:

- 8 bit di start (00001010);
- 10 bit di indirizzo sorgente;
- 10 bit di indirizzo destinazione;
- 8 bit di fine blocco (00011011);
- 2 bit di tipologia pacchetto;
- 10 bit di ID pacchetto;
- 8 bit di fine blocco;
- Payload di dimensione variabile;
- 8 bit di fine file (00011000).

Le tipologie di pacchetto previste sono due:

1. Modalità RTTY, in cui i dati vengono inviati secondo la codifica sopracitata;

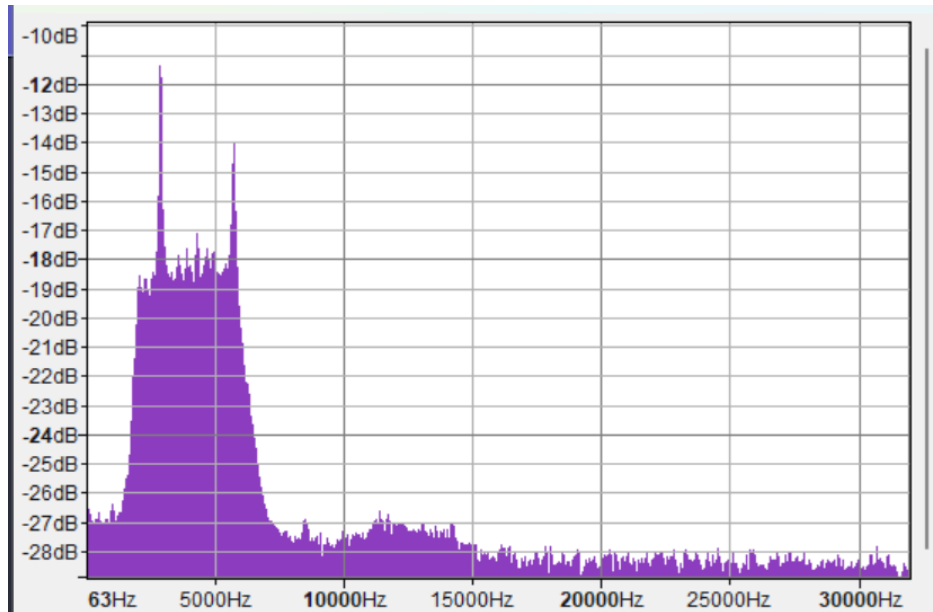


Figura 2.2: Spettro non filtrato

2. Modalità binaria, in cui i dati vengono inviati in forma binaria.

Il numero massimo di ID è 1024 (10 bit), il cui ruolo è identificare il pacchetto nella sequenza.

2.4 Risultati

La prima fase è stata svolta con l'obiettivo di comprendere i parametri ottimali della codifica. La stringa di testo viene codificata e convertita in un file `.wav` contenente la forma d'onda equivalente alla sequenza di toni. Il file `.wav` viene quindi decodificato per verificare l'integrità del dato contenuto nella forma d'onda. In questa fase sono stati settati gli attuali parametri di campionamento e durata del tono.

Nella seconda fase, il file `.wav` veniva generato allo stesso modo con il Raspberry e trasmesso in FM sui 90 MHz. La ricezione è stata inizialmente eseguita con una radio a valvole e successivamente con un ricevitore SDR collegato al PC. Tramite l'uso del software SDRSharp, il segnale è stato registrato e analizzato. La registrazione mostrava errori in decodifica, attribuibili alla presenza di componenti di rumore. Con l'ausilio di **Audacity**, è stato analizzato lo spettro del segnale audio (riferimento in fig.2.2), che mostrava componenti di rumore. Quindi, sempre con lo stesso software, è stato applicato un filtro passa-banda centrato sui 2000 Hz e 6000 Hz (Spettro filtrato in

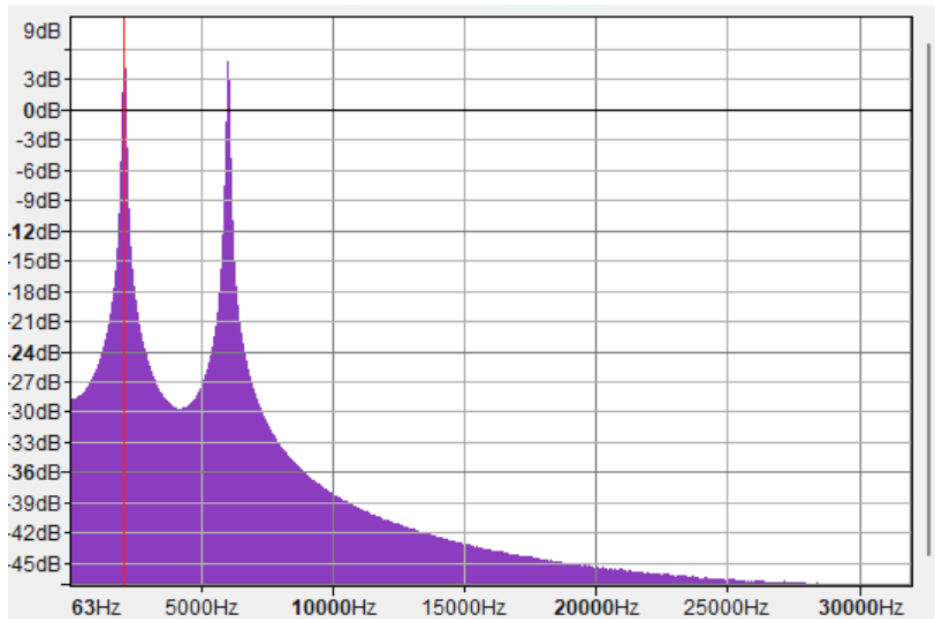


Figura 2.3: Spettro filtrato

fig. 2.3). Dopo il filtraggio, la decodifica ha avuto successo. Durante questa fase, è stata utilizzata provvisoriamente una durata del tono pari a 0,1 s per migliorare la percezione del segnale.

Nella terza fase, la durata del tono è stata ridotta secondo le specifiche tecniche. La ricezione è stata effettuata tramite il modulo `pyrtlsdr` integrato nel programma. Tuttavia, il risultato è stato negativo, in quanto non si è riusciti a tradurre correttamente alcun pacchetto trasmesso. La causa principale potrebbe risiedere in un errore nella demodulazione FM effettuata da `pyrtlsdr` o in un filtro software non ottimizzato nella libreria Python `scipy`.

Conclusione

L'ipotesi iniziale di questo progetto era quella di integrare il sistema RTTY con un sistema DMR, sfruttando la versatilità delle tecnologie moderne. Tuttavia per incompatibilità è stato impossibile sfruttare quest'ultimo protocollo, portando il progetto verso altri orizzonti.

Dai test condotti emergono alcune difficoltà tecniche come la gestione del segnale PWM e la presenza di rumore attribuibile alla demodulazione FM, gestita dal modulo `pyrtlsdr`. Tuttavia, l'ausilio di software come SDR# e Audacity mostra che si possono ottenere dei risultati accettabili.

Tramite tale sperimentazione si è quindi evidenziato come, attraverso ulteriori miglioramenti nella trasmissione e filtraggio del segnale, con il RTTY e l'uso di strumenti più accurati si possa raggiungere una soluzione.

Appendice A: Codifica RTTY

Numeri

01100000: 0, 01000001: 1, 01000010: 2, 01100011: 3, 01000100: 4,
01100101: 5, 01100110: 6, 01000111: 7, 01001000: 8, 01101001: 9

Lettere

10010000: a, 10001000: b, 10000100: c, 10000010: d, 10000001: e,
10011111: f, 10101110: g, 10001110: h, 10110111: i, 10111011: j,
10111000: k, 10110100: l, 10110010: m, 10110001: n, 10101100: o,
10101010: p, 10101001: q, 10100110: r, 10100101: s, 10100011: t,
10111110: u, 10011100: v, 10011010: w, 10011001: x, 10010110: y,
10010101: z,
11110000: A, 11101000: B, 11100100: C, 11100010: D, 11100001: E,
11111111: F, 11001110: G, 11101110: H, 11010111: I, 11011011: J,
11011000: K, 11010100: L, 11010010: M, 11010001: N, 11001100: O,
11001010: P, 11001001: Q, 11000110: R, 11000101: S, 11000011: T,
11011110: U, 11111100: V, 11111010: W, 11111001: X, 11110110: Y,
11110101: Z

Simboli speciali

00100001: [, 00100010:], 00100100: {, 00101000: }, 00110000: :,
00110001: , 00110010: /, 00111111: .

Simboli riservati

00011000: °, 00011011: #, 00001010: §, 00000000: ?, 00000011: '