

INFORMAZIONE ANALOGICA / INFORMAZIONE DIGITALE

- L'informazione è ANALOGICA se, qualunque sia la modalità con cui viene trasmessa (tramite suoni, immagini, video, multimedia...), può assumere **tutti gli infiniti valori** compresi in un determinato intervallo finito (in inglese **RANGE**), variando con continuità tra l'estremo inferiore e quello superiore del suddetto intervallo.



VU – meter : indicatore analogico del livello sonoro di un amplificatore audio



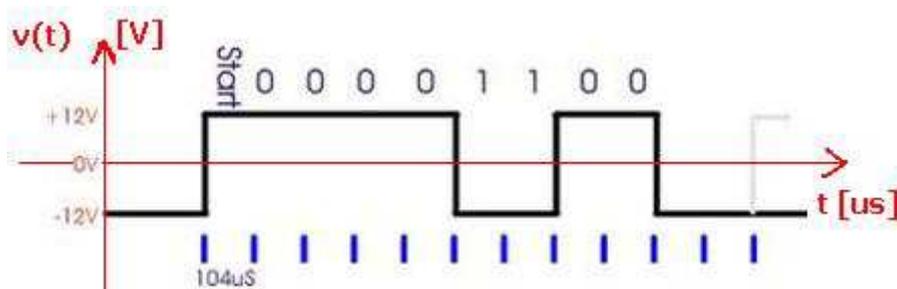
orologio analogico

- L'informazione è DIGITALE - MULTILIVELLO se può assumere solo un numero finito di valori.



Orologio digitale

- L'informazione è DIGITALE - BINARIA se può assumere solo 2 valori .



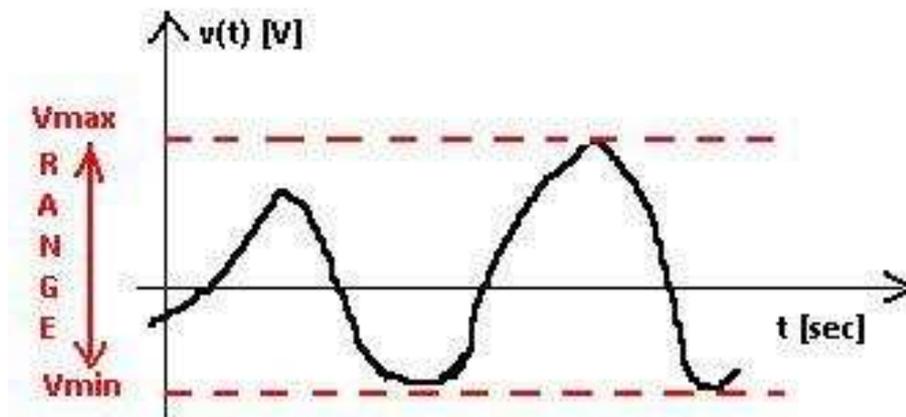
Segnale binario nello STANDARD RS 232 per la comunicazione seriale tra pc (a bassa velocità, inferiore a 112 [kbit / sec]

La differenza fondamentale tra i due tipi di segnale è che mentre nei segnali analogici l'informazione è contenuta nella "forma" stessa del segnale, nei segnali digitali l'informazione da elaborare è codificata in serie di simboli (1 e 0).

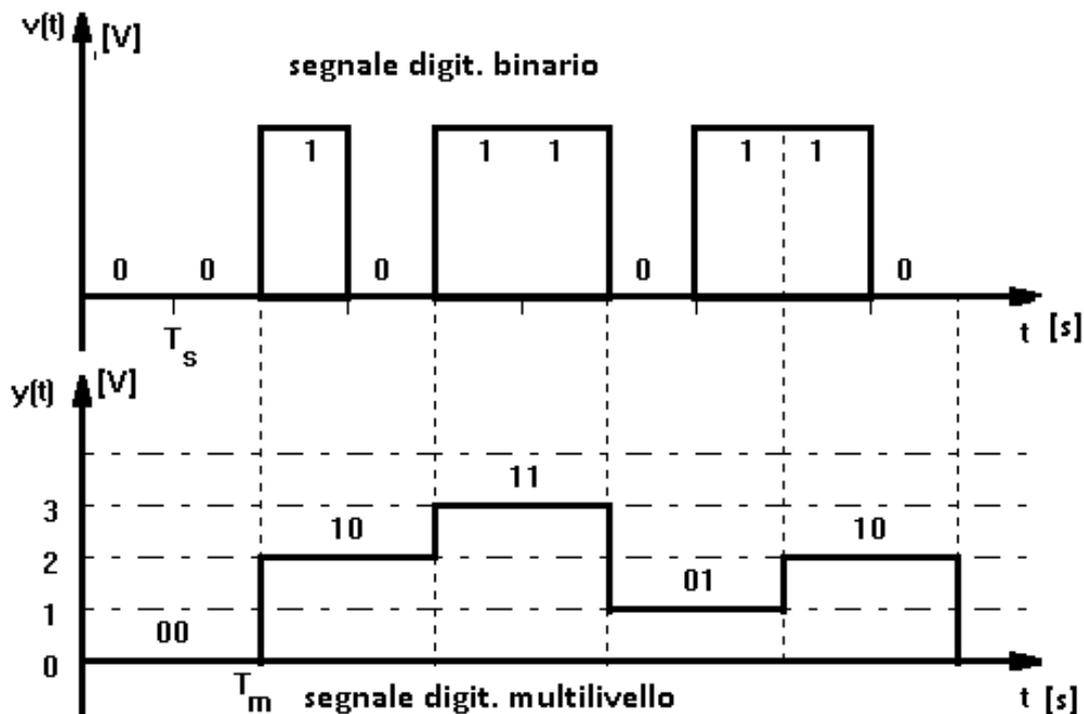
La forma del segnale quindi non ha importanza, basta che sia possibile decidere in ogni istante a quale valore logico corrisponde il segnale.

Le informazioni analogiche associate a determinate grandezze fisiche (temperatura, pressione, velocità, luminosità, suono, ...) sono trasformate dai Trasduttori (Sensori) in segnali analogici di tensione o di corrente, successivamente convertiti in segnali digitali.

Nella pratica , con segnale digitale s'intende praticamente sempre un segnale binario.



Segnale analogico



INTRODUZIONE ALLA TEORIA DELL' INFORMAZIONE

- Ogni messaggio informativo utilizza un **codice** che deve essere **comune** a chi lo trasmette e a chi lo riceve e sulla cui base il messaggio deve essere interpretato.
- I codici utilizzano di norma un certo numero di **simboli** : si costruiscono messaggi complessi combinando fra loro i simboli, sulla base di apposite **regole di combinazione**.
- Un messaggio viene codificato attraverso una successione di **scelte** effettuate fra i simboli e le varie combinazioni possibili.
- Utilizziamo il concetto di scelta (o decisione) per misurare la quantità di informazione contenuta in un messaggio.
- Ogni scelta può essere pensata come una successione di **scelte binarie**, infatti una scelta fra più alternative può sempre essere ridotta a **più scelte fra due alternative**.

Esempio: **il semaforo**.

Ha **3** lampade, ciascuna delle quali può essere **accesa o spenta**. La scelta è dunque fra **8** alternative :

1. tutte e tre le lampade spente
2. verde acceso
3. giallo acceso
4. rosso acceso
5. verde e giallo accesi
6. giallo e rosso accesi
7. verde e rosso accesi
8. verde, giallo e rosso tutti accesi

Il semaforo rappresenta una scelta fra **8** alternative possibili riguardo **3** variabili (le 3 lampade), ciascuna avente **2** soli stati possibili : accesa o spenta.

Qui interviene il concetto di logaritmo : **il logaritmo in base 2 di un numero n non è altro che il numero al quale bisogna elevare 2 per ottenere n .**

La scelta fra n alternative diverse, nel nostro esempio **8** (le alternative devono essere anche **egualmente probabili**) corrisponde a $\log_2 n$, nel nostro esempio **3** scelte binarie.

Sappiamo in effetti che $8=2^3$, e che dunque il logaritmo in base 2 di 8 è proprio 3.

Per capire meglio questo meccanismo pensate a un gioco : supponiamo che dobbiate indovinare il numero, intero, pensato da un amico, sapendo che questo numero è compreso fra **1 e 8** (8 possibili alternative) e avendo a disposizione **3** domande , alle quali il vostro amico può rispondere solo '**sì**' o '**no**'.

Potete innanzitutto chiedere (**1° domanda**) se il numero pensato dal vostro amico è compreso fra **1 e 4**.

Supponiamo che la risposta sia '**no**' : è allora chiaro che il numero è compreso fra **5 e 8**.

Vi sono rimaste **4** alternative.

Dimezzatele di nuovo, chiedendo come **2° domanda** se il numero è compreso fra **5 e 6**. Supponiamo che la risposta sia nuovamente '**no**'. Sono rimaste **2** sole alternative: il numero sarà **7**, oppure **8**.

La **3° domanda** potrà essere " **il numero che hai pensato è 7 ?** "

In caso di risposta affermativa, avrete individuato il numero che cercavate.

Ma anche in caso di risposta negativa l'avrete individuato: poteva essere solo o 7 o 8, non è 7, dunque è **8**.

Il lettore potrà verificare, costruendo un semplice **albero binario**, che **qualunque** sia il numero pensato dal suo ipotetico amico, se esso è compreso fra 1 e 8 , il processo di progressivo "dimezzamento" delle alternative porta sempre a individuarlo attraverso **3** domande.

In effetti, quella che abbiamo appena descritto è una *ricerca binaria* (o *logaritmica*) di **1 dato in un insieme: si tratta di uno degli algoritmi più importanti - e più usati - dell'informatica!**

Sappiamo che ogni scelta binaria corrisponde a **1 bit** di informazione: nell'esempio visto sopra, dunque, la scelta del vostro amico - che avveniva fra **8** alternative diverse e poteva essere quindi individuata o codificata attraverso tre scelte binarie – "peserà" **3 bit**.

In generale: se mi trovo davanti a una scelta fra n alternative egualmente probabili, la **quantità di informazione** (ovvero il numero di scelte binarie) corrispondente sarà uguale al **logaritmo in base 2 di n** (in formula, $\log_2 n$).

CODIFICA DELLE INFORMAZIONI

1. TESTI : la Codifica dei Caratteri

I caratteri sono quei simboli **alfabetici o numerici** che possono essere introdotti da una tastiera.

Vi possono essere tastiere diverse per ogni paese in quanto alcuni paesi usano tipi di caratteri diversi: ad esempio in Italia si usano anche i caratteri corrispondenti alle vocali accentate.

In Francia o Spagna si usa ad esempio la **cedilla** cioè il carattere ç e nelle tastiere di quei paesi é presente un tasto relativo.

Se un computer deve funzionare in questi paesi l'unica cosa che verrà sostituita é la tastiera mentre ovviamente il computer deve essere costruito per poter funzionare indipendentemente dal paese.

Ma quanti bit sono necessari per rappresentare un carattere ?

Su questo aspetto negli anni 60 e 70 vi è stata molta discussione. Alcuni costruttori suggerivano 6 bit, altri 7 infine si sono affermati gli 8 ed i 16 bit.

Un numero di bit uguale a **6** fornisce solo $2^6 = 64$ combinazioni.

Con 64 combinazioni si potevano memorizzare caratteri alfabetici e numerici, i vari segni di interpunzione come la virgola ed il punto ma **non** si potevano inserire i caratteri minuscoli o i caratteri speciali dei vari paesi.

La soluzione a **7** bit con le sue **128** possibilità permette di inserire tutti i tipi di carattere maiuscoli e minuscoli ma non i caratteri dei vari paesi.

La soluzione più diffusa attualmente é quella a **8** bit con la quale si possono rappresentare **256** caratteri.

Con questa scelta è possibile inserire anche **caratteri speciali** che servono per funzioni particolari come le comunicazioni o per rappresentare segni grafici speciali.

Un secondo problema é come disporre i caratteri tra le varie combinazioni, ad esempio la combinazione $(33)_{10} = (00100001)_2$ corrisponde normalmente alla **A** (maiuscola), ma chi lo stabilisce?

Ovviamente serve uno **standard**, cioè un modo uniforme di considerare i caratteri. In caso contrario se alla A un computer associa la combinazione 65 e alla B la combinazione 66 ed un altro computer fa l'opposto, quando trasferiamo un testo da un computer all'altro le A diventano B e viceversa, ad esempio BABA diventa ABAB.

Lo standard oggi più diffuso é quello **ASCII (American Standard Code for Information Interchange) a 7 bit**.

Lo standard ASCII non dà purtroppo una definizione precisa dei caratteri dalla combinazione 128 fino alla 255 e vi sono varie possibilità, una di queste la codifica **ASCII ESTESA a 8 bit**.

Nei moderni sistemi operativi, per i caratteri è utilizzata la codifica **UNICODE a 16 bit**.

Il numero di possibili simboli rappresentabili é **65536** ; si possono utilizzare le varie combinazioni binarie anche per rappresentare caratteri ideografici come ad esempio il Kanij dei giapponesi.

1. CODICE ASCII : American Standard Code for Information Interchange (a 7 bit)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	ˆ
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.pubblinet.com

Esempio: Roberto → in HEX : **R** **o** **b** **e** **r** **t** **o**
in BIN : **1010010** **1101111** **1100010** **1100101** **1110010** **1110100** **1101111**

2. ASCII esteso

Il termine **ASCII esteso** (in inglese *extended ASCII*) designa una codifica a 8 bit o più, in grado di rappresentare molti altri caratteri oltre ai tradizionali 128 dell'ASCII a 7 bit.

L'utilizzo di questo termine è stato spesso criticato in quanto potrebbe lasciar intendere (erroneamente) che l'ASCII originario sia stato aggiornato o che si tratti della stessa codifica.

Ragioni per l'espansione

Poiché il numero dei simboli usati nelle lingue naturali è di molto più grande dei caratteri codificabili col vecchio ASCII è stato necessario espanderne il set di codifica. Negli anni, nei paesi che non utilizzano l'alfabeto latino (o comunque caratteri non presenti nel set ASCII), come i paesi dell'estremo oriente o del mondo slavo, sono nati metodi di codifica per i caratteri non-standard afflitti però da seri problemi di compatibilità verso gli altri set.

3. ISO 8859 e adattamenti proprietari

In seguito al proliferare di codifiche proprietarie, l'ISO rilasciò uno standard denominato ISO 8859 contenente un'estensione a 8 bit del set ASCII. Il più importante fu l'ISO 8859-1, detto anche *Latin1*, contenente i caratteri per i linguaggi dell'Europa Occidentale. Furono standardizzate codifiche per gli altri linguaggi: ISO 8859-2 per i linguaggi dell'Europa Orientale, ISO 8859-5 per i caratteri cirillici e molti altri.

4. UNICODE

Una nuova codifica chiamata [Unicode](#) fu sviluppata nel [1991](#) per poter codificare più caratteri in modo standard e permettere di utilizzare più set di caratteri estesi (es. greco e cirillico) in un unico documento; questo set di caratteri è oggi largamente diffuso.

Inizialmente prevedeva 65.536 caratteri ed è stato in seguito esteso a 1.114.112 ($= 2^{20} + 2^{16}$) e finora ne sono stati assegnati circa 101.000.

I primi 256 *caratteri* ricalcano esattamente quelli dell'[ISO 8859-1](#). La maggior parte dei codici sono usati per codificare lingue come il [cinese](#), il [giapponese](#) ed il [coreano](#).

2. Codifica delle immagini

Le immagini si suddividono in **raster** e **vettoriali**.

Le immagini **raster** sono idealmente suddivise in punti (**pixel**), disposti su una griglia a righe orizzontali e verticali.

In origine, in un'immagine raster di tipo **bitmap** (mappa di bit) era possibile rappresentare solo due colori :

- **bianco (bit 0)**
- **nero (bit 1)**

Attualmente si prevedono diverse **profondità di colore** per i pixel dell'immagine.

Con **8** bit è possibile rappresentare **256** colori (es. **le tonalità di grigio**), con **16** più di **32000** colori, e così via.

Essendo le immagini un insieme di punti colorati disposti su una griglia, esse sono rappresentate con un certo livello di approssimazione o meglio, di **risoluzione**.

La risoluzione è data dal prodotto delle colonne e delle righe della griglia.

Ad esempio, quando si parla di un'immagine con risoluzione di **640×480 pixel**, significa che i pixel totali sono appunto **640×480 = 307200** e che essi sono disposti su una griglia di 640 colonne (larghezza) e 480 righe (altezza).

Un'immagine di **640×480 pixel** con **256** colori occupa **640*480*8 = 2.457.600 bit = 307.200 byte = 307.2 Kbyte**.

E' possibile risparmiare spazio nella rappresentazione delle immagini. Ad esempio, le aree dello stesso colore si possono rappresentare in modo "**abbreviato**": si specifica il colore e la lista delle coordinate dei pixel di quel colore.

Esistono vari formati di codifica **bitmap compressa** : **gif, jpg, pict, tiff** e altri.

In genere, è possibile passare da un formato ad un altro.

Le immagini possono essere anche rappresentate in forma "**vettoriale**". In questo caso si specificano le istruzioni su come disegnare l'immagine, per esempio del tipo:

```
...  
colore (255,128,0)  
spessore 1  
alpha = 1  
linea dal punto (10,12) al punto (20,30)  
colore (0,0,125)  
linea dal punto (20,30) al punto (20,35)  
...
```

Le immagini vettoriali sono comunque tradotte in formato bitmap al momento della visualizzazione (si ricordi che anche il monitor è una griglia di pixel).

Il vantaggio di questi formati è che non si perde di qualità nello “scaling” (zoom) dell’immagine.

Il formato è particolarmente idoneo per immagini di tipo geometrico (per esempio progetti ingegneristici) e generalmente richiede poco spazio.

Il formato più diffuso è il PostScript (ps, eps), anche per la stampa dei testi.

Un altro esempio di formato vettoriale è il cdr di CorelDraw.

3. Codifica dei filmati

Sono sequenze di immagini compresse : (ad esempio si possono registrare solo le variazioni tra un fotogramma e l’altro).

Esistono vari formati (compresi i suoni) :

- avi (microsoft)
- mpeg (il più usato)
- quicktime mov (apple)

4. Codifica dei suoni

L’onda sonora viene misurata (**campionata**) a intervalli regolari.

Minore è l’intervallo di campionamento, maggiore è la qualità del suono.

Per i CD musicali standard si fanno **44100** campionamenti al secondo (**frequenza di campionamento $F_c = 44100$ [Hz]**),

cioè si preleva **1** campione ogni **22,5 [micro sec]** (**Periodo di campionamento $T_c = 22,5$ [µsec]**)
e ogni campione è codificato con **16 bit**.