

CODICE ASCII

ASCII è l'acronimo di **American Standard Code for Information Interchange** ed è un codice ad 8 bit per rappresentare caratteri e simboli. La prima edizione è stata pubblicata dall'ANSI (**American National Standards Institute**) nel 1963. Successivamente il codice ASCII è stato accettato dall'ISO come standard con il nome **ISO/IEC 646**.

Inizialmente il codice ASCII utilizzava solamente 7 bit, consentendo la codifica di 128 caratteri, successivamente con lo standard EXTENDED ASCII sono stati utilizzati 8 bit per ottenere la codifica di 256 caratteri.

Di seguito la tabella ASCII con i corrispondenti codici decimali associati ad ogni simbolo

NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{	 	}	~	DEL
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
€		,	f	„	…	†	‡	^	‰	Š	<	Œ		Ž	
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
	`	´	ˆ	˜	•	—	—	~	™	š	>	œ		ž	ÿ
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Una rappresentazione completa della tabella ASCII con i codici espressi in decimale, ottale ed esadecimale, è visionabile al seguente link: <https://www.asciitable.it/>

Non tutti i codici sono associati a dei caratteri, ad esempio dal numero 0 al 31 abbiamo dei caratteri di controllo che venivano utilizzati per comunicare alle stampanti determinate funzioni, come **CR Carriage Return** (13) che serve per andare accapo o **LF Line Feed** (10) che identifica una nuova linea. Oggi di questi caratteri di controllo vengono utilizzati solo i due appena nominati, che spesso troviamo abbinati insieme nelle varie istruzioni per visualizzare a schermo utilizzate nei vari linguaggi di programmazione.

Dal codice 32 al codice 127 ci sono tutti i caratteri stampabili previsti nella prima versione del codice ASCII.

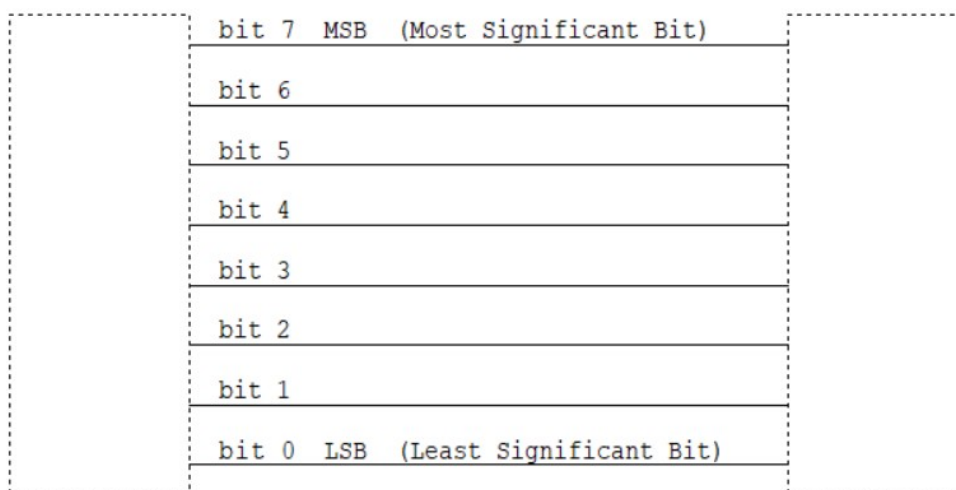
Dal codice 128 al 255 invece troviamo tutti i caratteri aggiunti con la versione EXTENDED ASCII.

COMUNICAZIONE SERIALE

Una suddivisione importante che possiamo fare nelle comunicazioni, è tra comunicazione seriale e parallela.

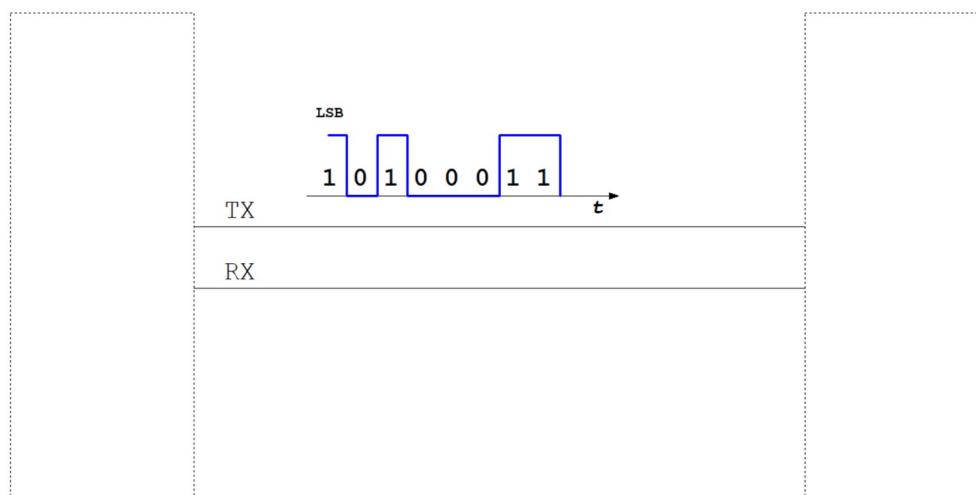
COMUNICAZIONE PARALLELA

In questo caso i bit vengono inviati contemporaneamente tra i due dispositivi comunicanti. Per questo tipo di comunicazione occorrono più linee, una per ogni bit.



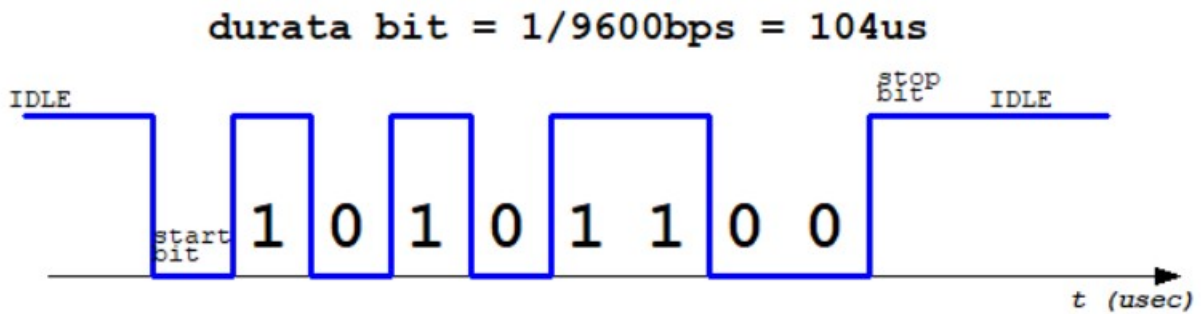
COMUNICAZIONE SERIALE

In questo caso i bit transitano su una linea per la Trasmissione e su una linea per la Ricezione in maniera temporalmente consecutiva.



Nella gran parte dei casi oggi viene utilizzata una comunicazione SINCRONA, come nel caso delle reti Ethernet o della comunicazione USB.

Uno degli standard più conosciuti per comunicare in maniera seriale, è lo standard RS232 TTL che prevede l'invio di un byte alla volta.



Lo standard prevede l'invio di 8 bit preceduti da uno start bit e da uno o più stop bit, come in figura.

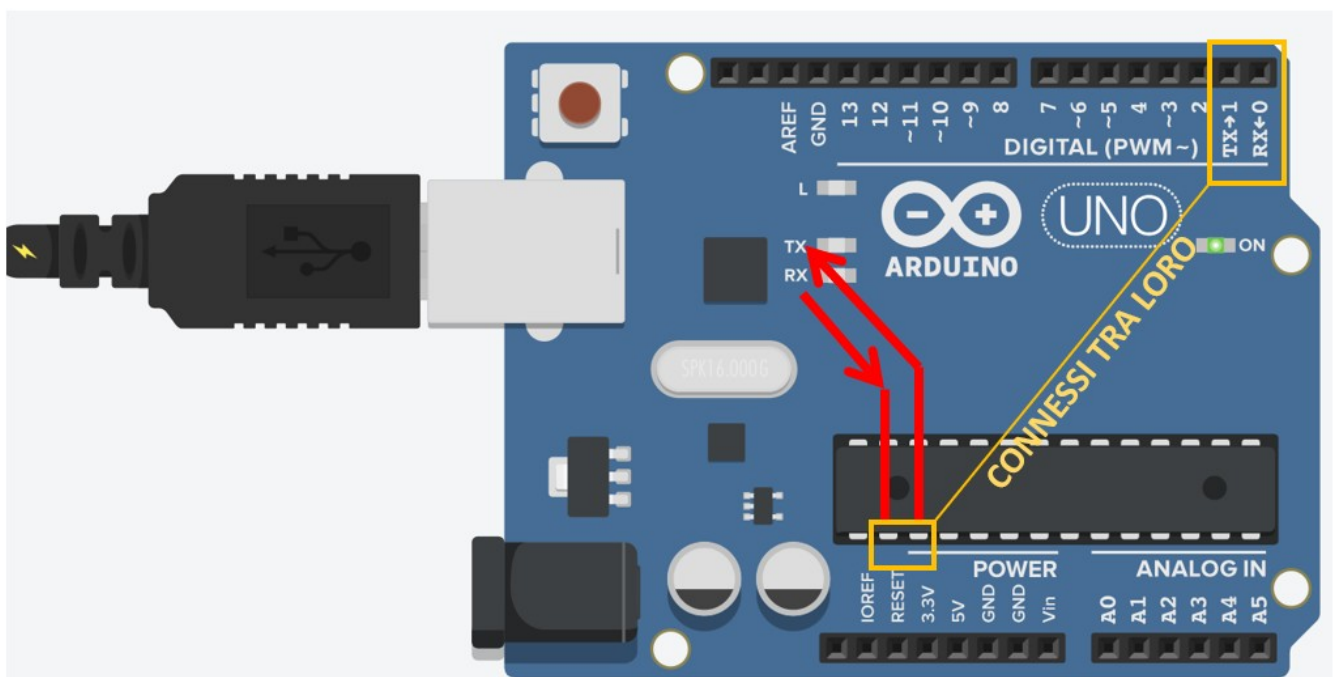
La durata di ogni bit dipende dalla velocità utilizzata, ad esempio utilizzando una velocità di 9600 bit per secondo, ogni bit avrà una durata di $1/9600$ cioè $104\mu\text{sec}$.

COMUNICAZIONE CON LE SCHEDE A MICROCONTROLORE ARDUINO UNO O ARDUINO NANO

Nelle schede ARDUINO UNO o ARDUINO NANO, ma anche in tutte quelle appartenenti a questa piattaforma, c'è la possibilità di far comunicare il Computer con la scheda ad esso collegata.

All'interno della scheda ci sono due pin del microcontrollore collegati alle linee 0 ed 1. Questi pin vengono utilizzati anche per ricevere (RX) e trasmettere (TX) dei dati tramite la connessione USB del PC (freccie rosse). Il segnale seriale viene convertito dallo standard RS232 TTL allo standard USB e viceversa.

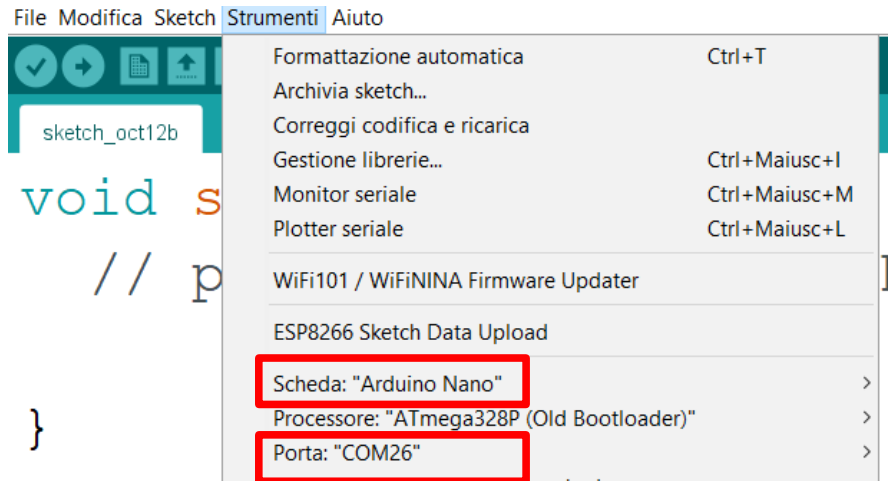
Pertanto se vogliamo far comunicare la nostra scheda con il PC non potremo utilizzare le linee 0 ed 1.



MONITOR SERIALE

Il monitor seriale è una parte dell'IDE di Arduino, e viene utilizzato per inviare e ricevere caratteri alla scheda.

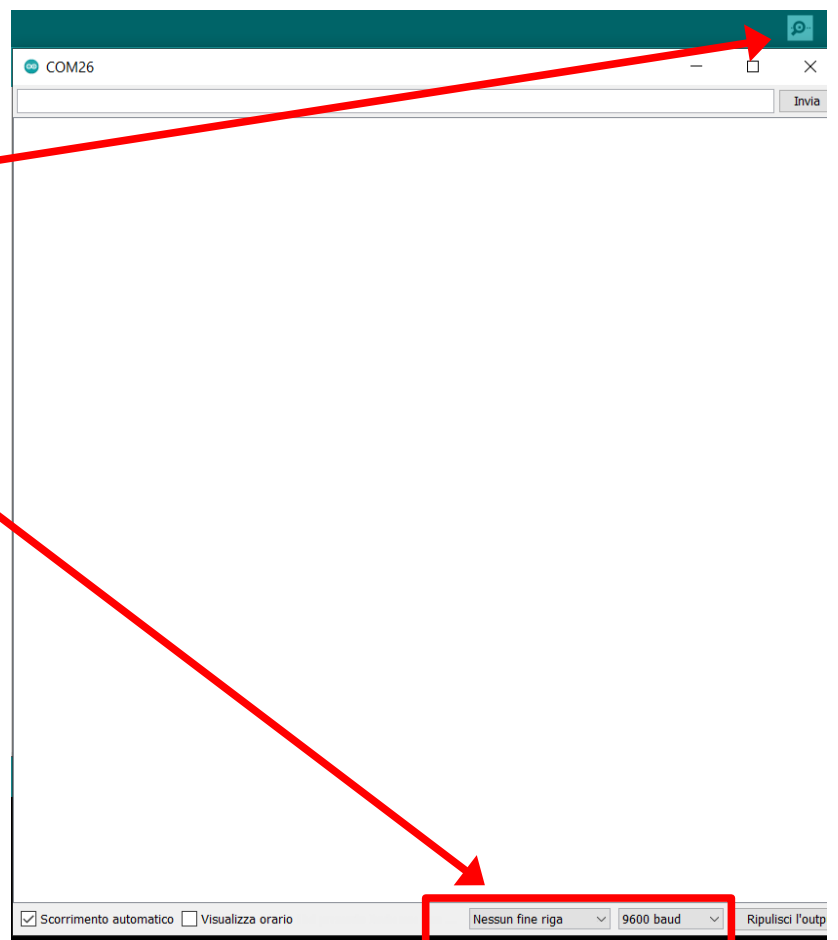
Per attivare il monitor seriale occorre prima collegare la scheda al PC e selezionare nel menù "strumenti" dell'IDE il tipo di scheda e la porta COM.



Una volta collegata la scheda ed effettuate le impostazioni, si potrà aprire il **monitor seriale** con l'apposito pulsante.

E' importante anche impostare la stessa velocità che utilizzeremo nel programma, ed impostare "nessun fine riga" in basso della schermata.

Ricordiamoci che 9600 baud, corrisponde a 9600 bit per secondo, e quando scriveremo il programma dovremo ricordarci di impostare questa velocità.



Successivamente se scriveremo un programma che invia dati dalla scheda Arduino Uno, vedremo i dati inviati al centro della schermata. E viceversa se vogliamo inviare dati dal PC alla scheda dovremo scrivere nella barra in alto e premere il pulsante INVIA.

INVIO DI DATI DALLA SCHEDA AL PC

Per fare in modo che la scheda Arduino Uno possa ricevere o inviare dati, occorre innanzitutto scrivere nel SETUP la seguente funzione:

```
void setup() {  
  Serial.begin(9600);  
}
```

In questo modo diremo alla scheda che la velocità dei dati sarà di 9600 bit al secondo.

Successivamente per inviare dei dati esistono tre funzioni:

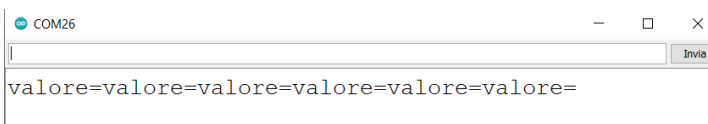
- **Serial.print**
- **Serial.println**
- **Serial.write**

Di seguito tratteremo in maniera molto sintetica le prime due funzioni, per approfondire l'argomento si consiglia la lettura della dispensa al seguente link:

<https://danielepostacchini.it/wp-content/uploads/2021/01/Arduino-Monitor-Seriale.pdf>

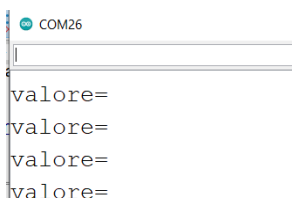
Per capire il funzionamento vediamo 4 esempi di utilizzo delle prime due funzioni:

1. *In questo caso verrà inviata la stringa contenente il codice ASCII delle lettere che compongono la parola "valore" insieme al codice ASCII del simbolo "=". L'invio avverrà ogni secondo. Il risultato sarà il seguente.*



```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print("valore=");  
  delay(1000);  
}
```

2. *In questo caso verrà inviata la stringa come prima, ma alla fine verranno inviati i due caratteri ASCII CR e LN, cioè accapo e nuova linea. Il risultato pertanto sarà il seguente.*



```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("valore=");  
  delay(1000);  
}
```

3. In questo caso invece utilizzeremo `Serial.print` per scrivere la stringa “**Valore=**” senza andare accapo, e poi scriveremo il contenuto della variabile `cont`.

```
int cont=10;

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.print("valore=");
  Serial.println(cont);
  delay(1000);
}
```

Deve esser chiaro che in questo caso oltre al codice ASCII dei caratteri della stringa `Valore=` vengono inviati anche i codici ASCII dei simboli 1 e 0, ma non viene inviato un byte contenente il valore 10.

Andando a rivedere la tabella ASCII nell'esempio precedente verranno inviati i seguenti byte.

	Codice ASCII in decimale	Codice ASCII in binario
V	86	0101 0110
a	97	0110 0001
l	108	0110 1100
o	111	0110 1111
r	114	0111 0010
e	101	0110 0101
=	61	0011 1101
1	49	0011 0001
0	48	0011 0000

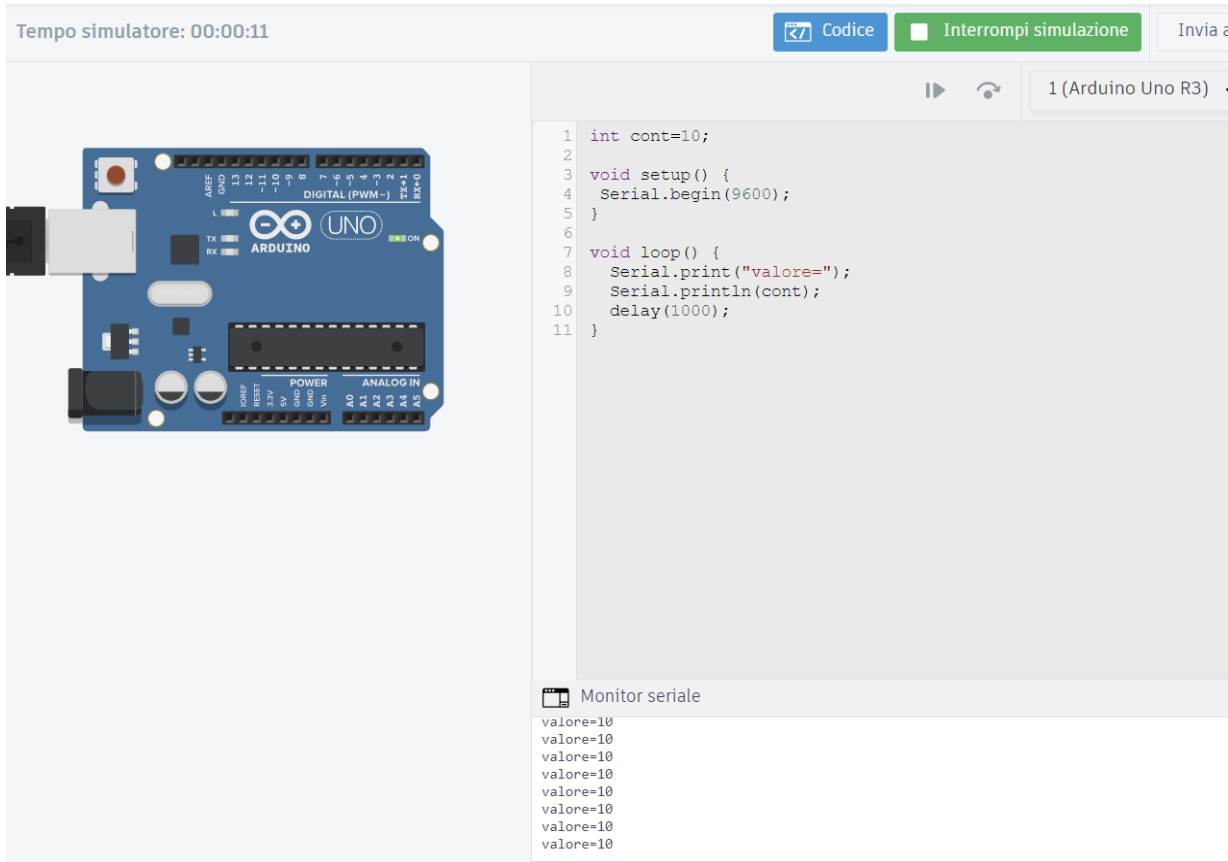
Come si può vedere non c'è traccia del valore 10 scritto in binario che corrisponderebbe alla sequenza di bit **0000 1010**.

Le funzioni `Serial.print` e `Serial.println` pertanto inviano solamente il codice ASCII dei caratteri che compongono la cifra.

Per inviare un byte contenente un valore numerico si utilizza la funzione `Serial.write` descritta nella dispensa indicata precedentemente nel link.

MONITOR SERIALE SU TINKERCAD

E' possibile utilizzare il monitor serial anche su tinkercad, è sufficiente cliccare sul pulsante CODICE ed in basso troveremo il monitor seriale.



ESERCIZIO

Scrivere un programma su tinkercad che realizzi la seguente funzione:

- *all'avvio del programma sul monitor seriale visualizzare la scritta "conteggio=" seguita da un valore che va da 0 a 10 e da 10 a 0 continuamente, con l'intervallo di un secondo tra ogni valore.*

```
COM26
conteggio=0
conteggio=1
conteggio=2
conteggio=3
conteggio=4
conteggio=5
conteggio=6
conteggio=7
conteggio=8
conteggio=9
conteggio=10
conteggio=9
conteggio=8
conteggio=7
conteggio=6
conteggio=5
conteggio=4
conteggio=3
conteggio=2
conteggio=1
conteggio=0
conteggio=1
conteggio=2
conteggio=3
```