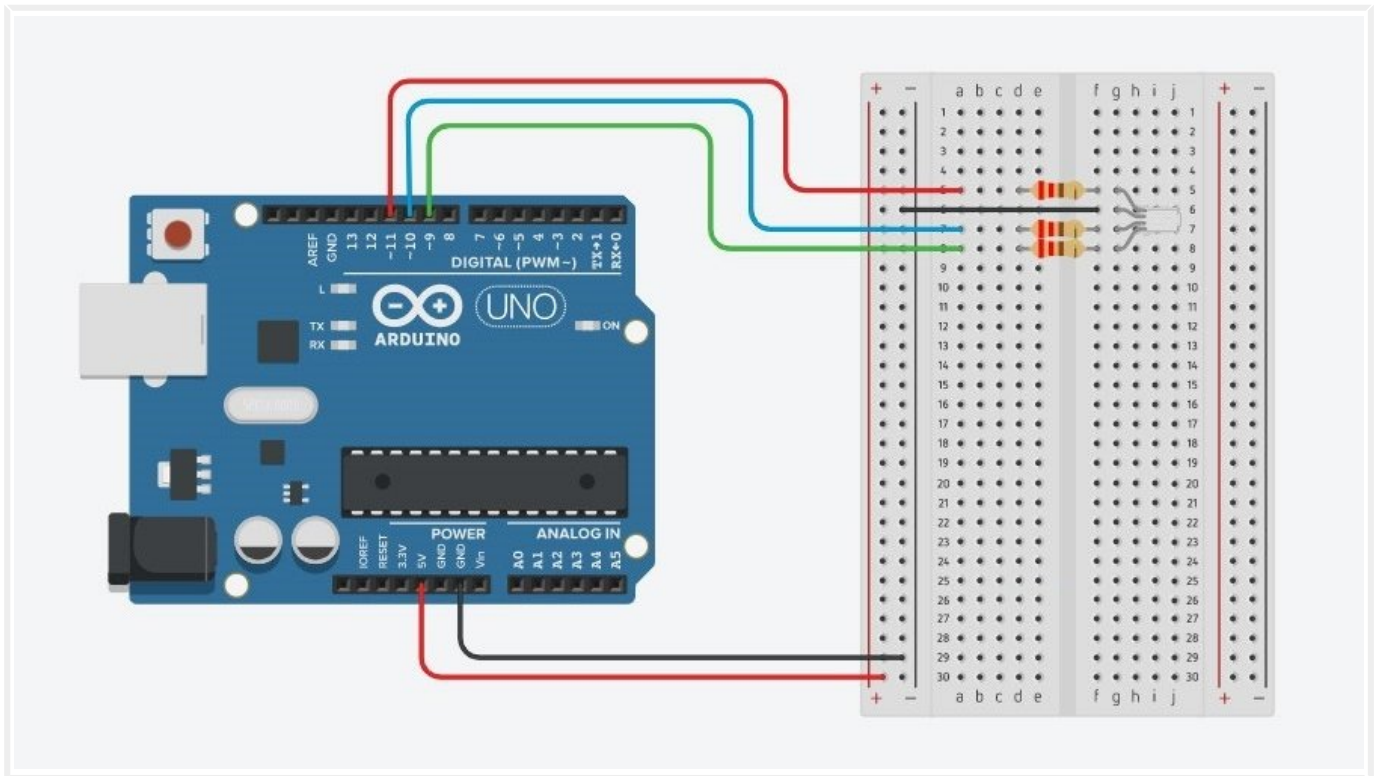


# *Serial - Envoi de caractères*

## *(du moniteur série vers l'Arduino)*



### Liste des composants

- . 1 DEL RVB
- . 3 résistances de 220  $\Omega$  (résistances de protection des DELs)
- . 1 plaque d'essai
- . Fils de connexion

### Objectif

La fonction "print()" de la classe "Serial" permet d'envoyer des données depuis la carte Arduino vers le moniteur série.

Il est également possible d'envoyer des données du moniteur série vers la carte Arduino.

Dans cette activité, le code demande à l'utilisateur de saisir un caractère dans le moniteur série:

- . 'R', 'V' ou 'B' pour allumer respectivement la DEL rouge, verte ou bleue
- . '0' pour éteindre les DELs

Le caractère envoyé est lu et l'action correspondante est effectuée. Le programme ne fait rien si le caractère lu n'est pas un des caractères attendus.

## . Le programme

### Serial\_Char\_Send

```
const int PinLEDR = 11;
const int PinLEDB = 10;
const int PinLEDV = 9;
const int PinPOT = A0;

const char LedR = 'R';
const char LedV = 'V';
const char LedB = 'B';
const char Off = '0';
char c=' ';

void setup() {
  Serial.begin(9600);
  pinMode (PinLEDR, OUTPUT);
  pinMode (PinLEDB, OUTPUT);
  pinMode (PinLEDV, OUTPUT);
  Serial.println ("Pour allumer la DEL rouge, envoyer: R");
  Serial.println ("Pour allumer la DEL verte, envoyer: V");
  Serial.println ("Pour allumer la DEL bleue, envoyer: B");
  Serial.println ("Pour eteindre les DELs, envoyer: 0");
}

void loop() {
  if (Serial.available() > 0) {
    c = Serial.read();
  }

  if (c == LedR) {
    Serial.println (c);
    digitalWrite(PinLEDR,1);
    digitalWrite(PinLEDV,0);
    digitalWrite(PinLEDB,0);
  }
  if (c == LedV) {
    Serial.println (c);
    digitalWrite(PinLEDR,0);
    digitalWrite(PinLEDV,1);
    digitalWrite(PinLEDB,0);
  }
  if (c == LedB) {
    Serial.println (c);
    digitalWrite(PinLEDR,0);
    digitalWrite(PinLEDV,0);
    digitalWrite(PinLEDB,1);
  }
  if (c == Off) {
    Serial.println (c);
    digitalWrite(PinLEDR,0);
    digitalWrite(PinLEDV,0);
    digitalWrite(PinLEDB,0);
  }
  Serial.flush();
  c=' ';
}
```

Dans ce programme, on demande à l'utilisateur de saisir un caractère dans le moniteur série.

Une fois, la saisie effectuée, le nombre d'octets (caractères) disponible pour lecture dans la file d'attente du port série n'est plus nul (**Serial.available() > 0**), le caractère envoyé est alors lu à l'aide de la fonction **"read()"** et stocké dans la variable **"c"** définie comme étant un caractère.

### Remarques :

- L'instruction **"char c"** déclare une variable, **"c"** d'un octet de mémoire (8 bits) qui contient une valeur correspondant à un caractère. Les caractères uniques sont écrits entre des guillemets simples, comme ceci : **'A'**
- Les caractères sont stockés de la même façon que les nombres : à chaque caractère correspond une valeur numérique comprise entre 0 et 127 (code ASCII). Ceci signifie également qu'il est possible de faire des opérations sur les caractères, dans lesquelles la valeur ASCII du caractère est utilisée (par exemple **'A'+1** a la valeur 66, car la valeur ASCII de la lettre capitale A est 65).
- Si la donnée envoyée contient plus d'un caractère (par exemple, le nombre « 123 »), la donnée est considérée comme une suite de 3 caractères qui seront affichés à la suite tant que la fonction **"Serial.available()"** retournera une valeur non nulle.
- Le code ASCII (American Standard Code for Information Interchange) est la façon standard de coder un texte numériquement :

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

Source: [www.LookupTables.com](http://www.LookupTables.com)

Le code ASCII ne contient pas de caractères accentués parce qu’il a été mis au point pour la langue anglaise. Pour coder ce type de caractère, il faut recourir à un autre code. Le code ASCII a donc été étendu à 8 bits (un octet) pour pouvoir coder plus de caractères (on parle d’ailleurs de code ASCII étendu...).

Ce code attribue les valeurs 0 à 255 aux lettres majuscules et minuscules, aux chiffres, aux marques de ponctuation et aux autres symboles :

128	Ç	144	É	161	í	177	⌘	193	⌞	209	⌞	225	ß	241	±
129	ü	145	æ	162	ó	178	⌘	194	⌞	210	⌞	226	Γ	242	≥
130	é	146	Æ	163	û	179		195	⌞	211	⌞	227	π	243	≤
131	â	147	ô	164	ñ	180	†	196	—	212	⌞	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	‡	197	†	213	⌞	229	σ	245	∫
133	à	149	ò	166	²	182	‡	198	†	214	⌞	230	μ	246	÷
134	å	150	û	167	°	183	¶	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	168	¿	184	¶	200	⌞	216	‡	232	Φ	248	°
136	ê	152	—	169	—	185	‡	201	⌞	217	‡	233	⊖	249	·
137	ë	153	Ö	170	¬	186		202	⌞	218	⌞	234	Ω	250	·
138	è	154	Û	171	½	187	¶	203	⌞	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	¶	204	‡	220	■	236	∞	252	—
140	î	157	¥	173	¡	189	¶	205	=	221	■	237	φ	253	²
141	ì	158	—	174	«	190	¶	206	‡	222	■	238	ε	254	■
142	Ä	159	ƒ	175	»	191	¶	207	⌞	223	■	239	∩	255	
143	Å	160	á	176	⌘	192	⌞	208	⌞	224	α	240	≡		

Source : [www.LookupTables.com](http://www.LookupTables.com)