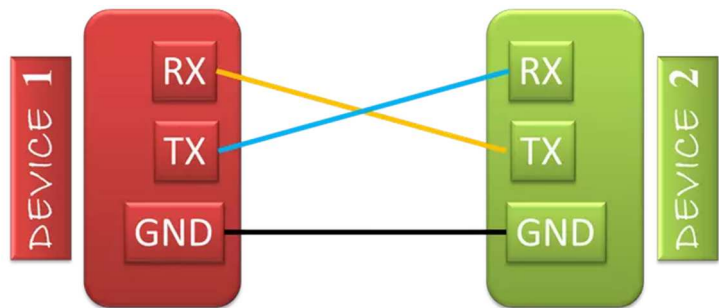
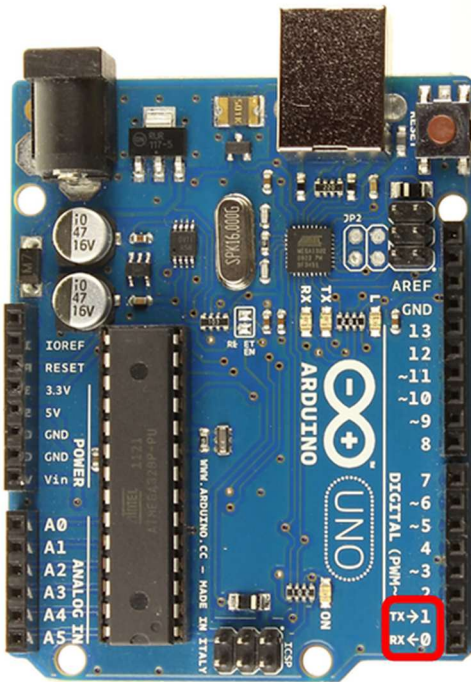


COMMUNICATION SERIE



INSTRUCTIONS PERMANENTES DE SÉCURITÉ



1. Avant toute mise sous tension, le professeur vérifie le montage et contrôle le calibrage des éventuels appareils de mesure.
2. La mise sous tension et hors tension du poste (consignation, déconsignation) est effectuée en présence du professeur.
3. Toute intervention nécessitant l'ouverture d'un circuit électrique (installation d'un appareil) est effectuée hors tension.
4. Pendant la phase où le poste est sous tension, l'élève travaille sans modifier le câblage du circuit (relevés de mesures ...).
5. En cas de problèmes sur un poste de travail voisin, vous devez impérativement couper l'alimentation du poste en activant le bouton d'arrêt d'urgence le plus proche.

**C'EST LE PROFESSEUR QUI DONNE, APRÈS AVOIR
PROCÉDÉ À LA CONSIGNATION DU POSTE,
L'AUTORISATION DE DÉMONTAGE**

Informations



Vous devez rédiger un compte rendu par élève.

Objectifs

L'objectif de l'activité est d'observer et de décoder la transmission d'une information sur un liaison de communication série grâce à une carte Arduino UNO.

1. Envoi d'un caractère

Q1. Proposer un schéma de principe permettant de relever le signal de la broche 1 (TxD - transmission série de données - pin 1) de la carte arduino directement sur un oscilloscope ou un piscope.

Q2. Saisir le code suivant sur le logiciel arduino IDE :

```
Liaison_serie.ino
1 void setup() {
2
3   Serial.begin(9600); // Ouvrir une discussion sur port serie
4 }
5
6 void loop() {
7
8   Serial.print('U'); // Envoie d'un caractère
9   delay(200);
10 }
```

Q3. Compiler et téléverser le programme dans la carte arduino.

Q4. Observer la trame à l'oscilloscope, la reporter sous forme de chronogramme ou de copie d'écran propre sur votre copie et à l'aide de l'annexe :

- identifier le bit de start et le bit de stop,
- lire la donnée en binaire,
- convertir la donnée en hexadécimal,
- donner le caractère ASCII correspondant : Voir annexe

Q5. Y a-t-il un bit de parité ? Si oui, est-il pair ou impair ?

Q6. Modifier le programme comme suit : `Serial.begin(9600, SERIAL_7E1)`. Simuler, relever le chronogramme et commenter la différence.

Q7. Mesurer la durée de la trame. En déduire le débit de transmission. Quel paramètre permet de régler ce débit dans le programme ?

2. Envoi d'une chaîne de caractère

- Q8.** Modifier votre programme pour envoyer cette fois-ci non plus un caractère mais une chaîne de caractère : « UN » → `Serial.print("UN")`
- Q9.** Relever le chronogramme de TxD en identifiant les bits à 1 (5V) et les bits à 0 (0 V).
- Q10.** Identifier sur le chronogramme chaque bit transmis en les nommant : S (start), P (stop), b7, b6, b5, b4, b3, b2, b1 et b0.
- Q11.** Identifier les deux caractères envoyés et préciser lequel est le premier transmis.
- Q12.** Quelle sera la longueur en bits de la transmission série du message « 2I2D-EE » ?

3. Liaison asynchrone : notion de vitesse de transmission

- Q13.** Modifier le programme afin d'envoyer le message « STI2D ». Vérifiez que le message s'affiche bien dans le moniteur série d'arduino (Outils → Moniteur série).
- Q14.** Modifier la valeur 9600 par 4800 du `Serial.begin`. Mesurer la durée de la trame. En déduire le débit de transmission.
- Q15.** Commenter le message reçu dans le moniteur série.
- Q16.** Arrêter la simulation, double-cliquer dans le moniteur série et modifier la valeur du « Baud rate » à 4800 également. Relancer la simulation et commenter le message reçu dans le moniteur série
- Q17.** Quelle est la condition indispensable dans une liaison **asynchrone** entre un **émetteur** et un **récepteur** pour que les messages soient lisibles ? (« Baud rate » correspond au paramètre *vitesse de transmission*, exprimé en bit/s.)

4. Liaison entre 2 Arduino Uno - BONUS

- Q18.** Proposer un schéma de principe pour faire communiquer 2 Arduino par liaison série : l'un disposant d'une LED sur une sortie quelconque, l'autre d'un bouton poussoir sur une entrée ou les 2 !
- Q19.** Réaliser le raccordement et mettre en œuvre une solution pour prouver l'échange.

https://www.tinkercad.com/things/47W2qRZqcqJ-powerful-snaget/editel?sharecode=xPHOqoC-g_m8E7buYMG093ooN7ZKGo45aCoJtJpMOMw

ANNEXE

STRUCTURE D'UNE TRAME SERIE protocole UART

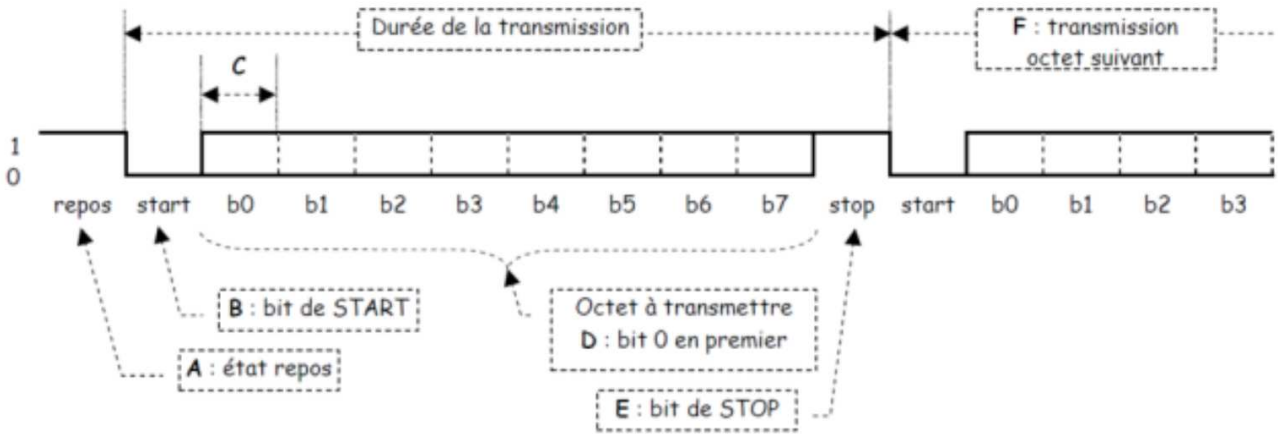


TABLE DE CODES ASCII

Binaire					HEXA	0	1	2	3	4	5	6	7			
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0									
0	0	0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p	
0	0	0	0	1	1	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	0	1	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	0	1	1	0	0	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	0	0	1	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	0	0	1	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	0	0	1	0	6	ACK	SYN	8	6	F	V	f	v
0	1	1	1	0	0	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	0	0	1	0	8	BS	CAN	(8	H	X	h	x
1	0	0	0	1	0	1	0	9	HT	EM)	9	I	Y	i	y
1	0	1	0	0	0	1	0	A	LF	SUB	*	:	J	Z	j	z
1	0	1	1	0	0	1	0	B	VT	ESC	+	;	K	[k	{
1	1	0	0	0	0	1	0	C	FF	FS	,	<	L	\	l	
1	1	0	1	0	0	1	0	D	CR	GS	-	=	M]	m	}
1	1	1	0	0	0	1	0	E	SO	RS	.	>	N	^	n	~
1	1	1	1	0	0	1	0	F	SI	US	/	?	O	_	o	DEL

Exemple : le mot « Test » sera codé de la manière suivante

Codage	T	e	s	t
Binaire	1010100	1100101
Hexadécimal	54	65