Document élève	Séquence "Transmettre des informations" Activité : **Transmettre des informations d'une carte Arduino à une autre avec la liaison série UART**	BAC S.I

Objectif

Dans le cadre de cette activité pratique, vous allez faire communiquer deux arduinos par la liaison série et analyser la trame transmise.

Matériels nécessaires

• Deux cartes Arduino Mega2560 • un bouton poussoir • une led (diode électroluminescente) • une résistance de protection de la led de 220 à 330 Ohms • quelques fils de prototypage rapide.	• Un breadboard / plaque de prototypage rapide (idéalement deux) • Un module d'oscilloscope Pycoscope (un oscilloscope numérique classique peut faire l'affaire dans un premier temps mais il ne dispose pas la fonction d'analyse de trame).

Schéma électrique des cartes Arduinos

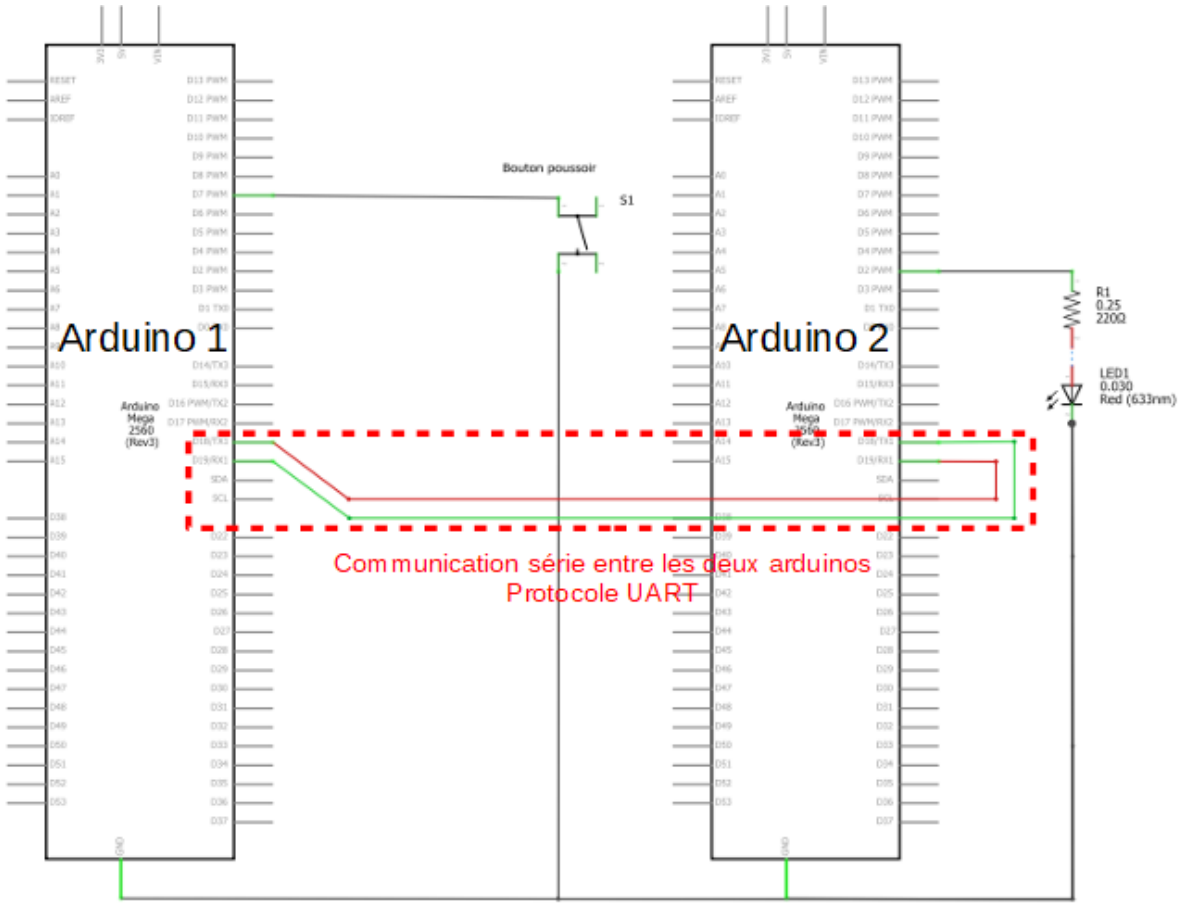
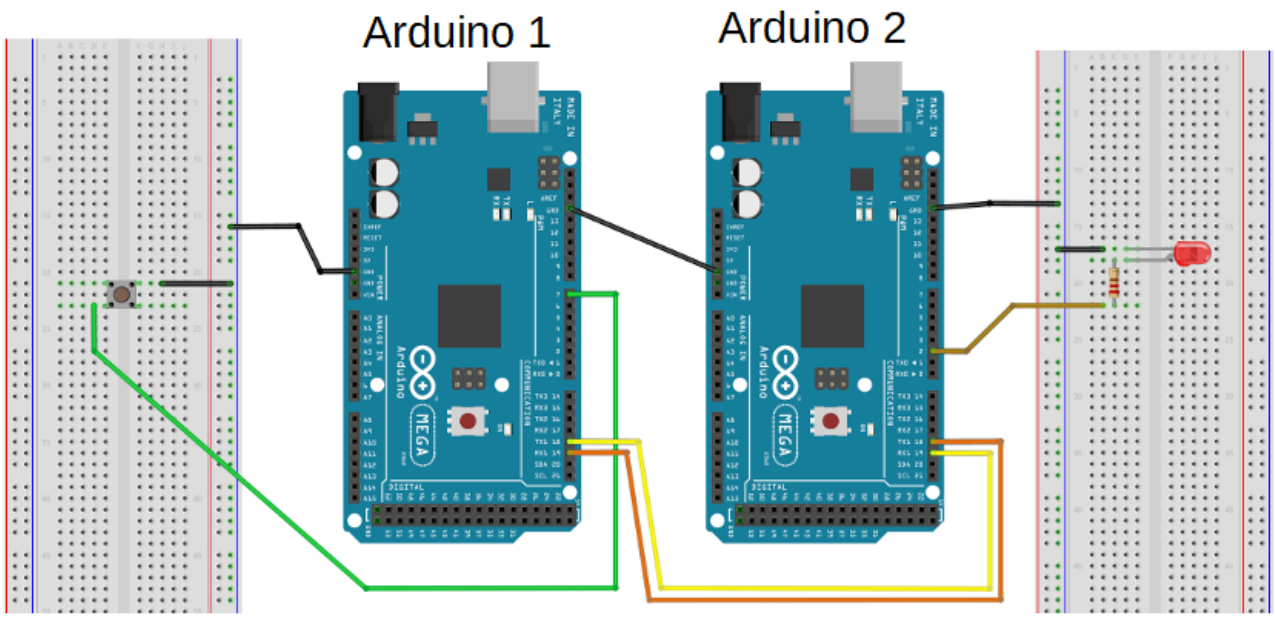


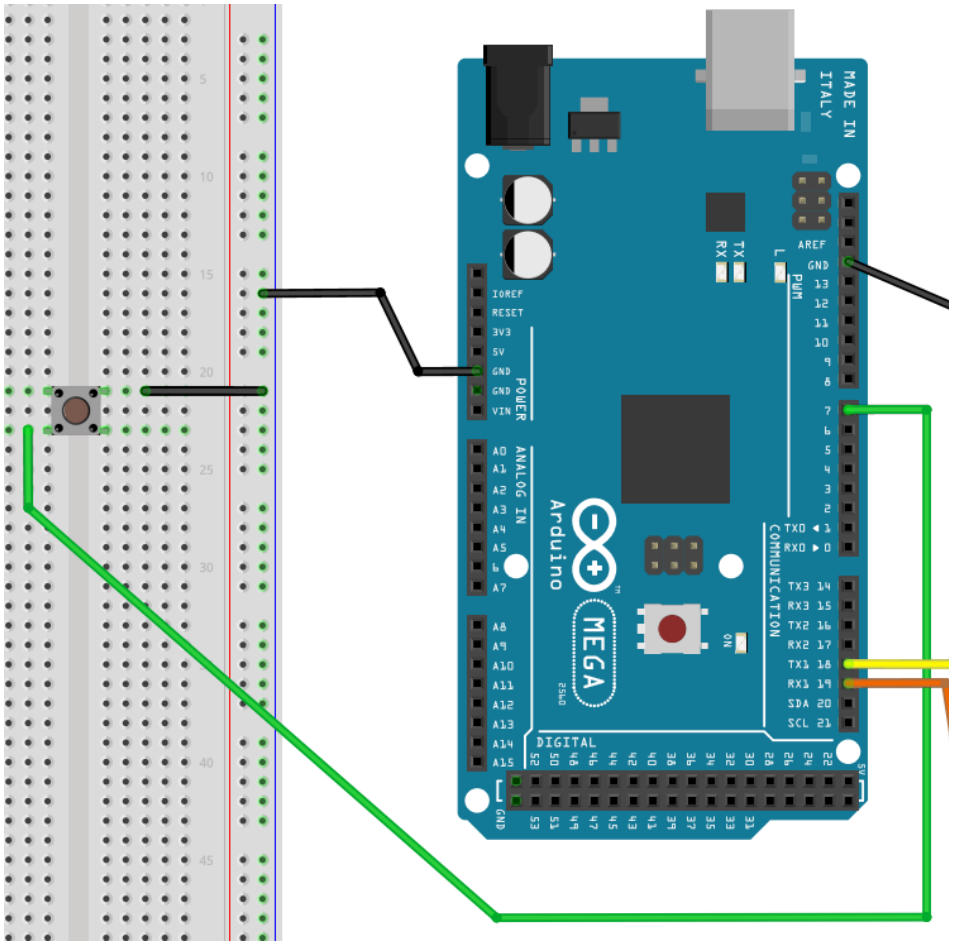
Schéma de câblage du montage précédent sur plaque de prototypage

Réaliser le câblage des deux arduinos comme indiqué ci-après.

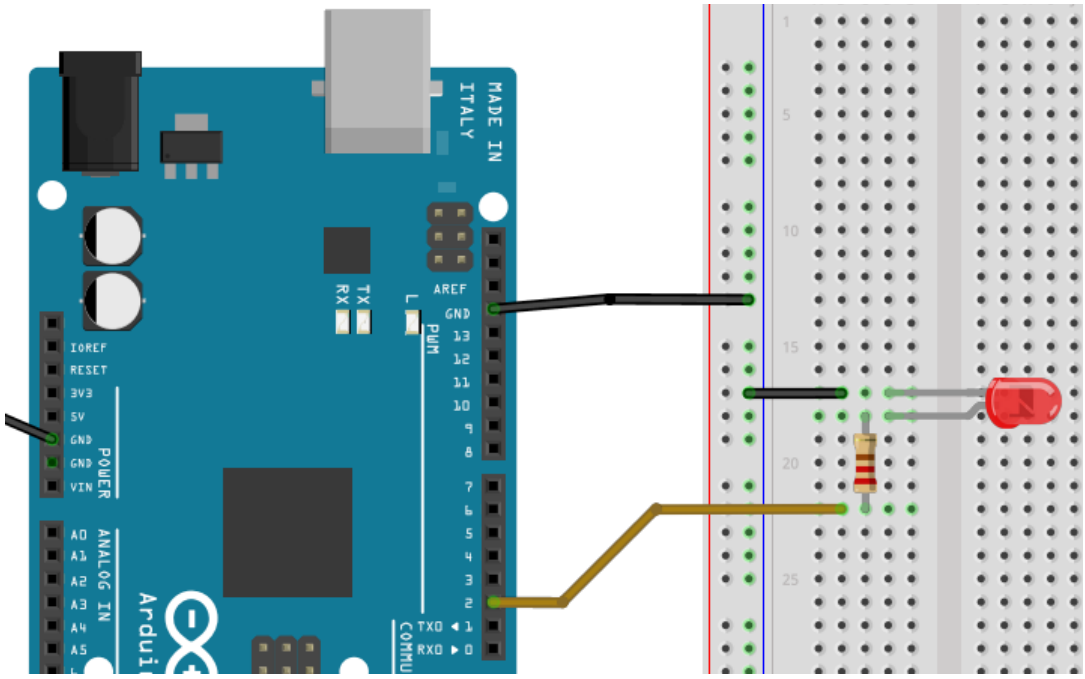


Détails des connexions :

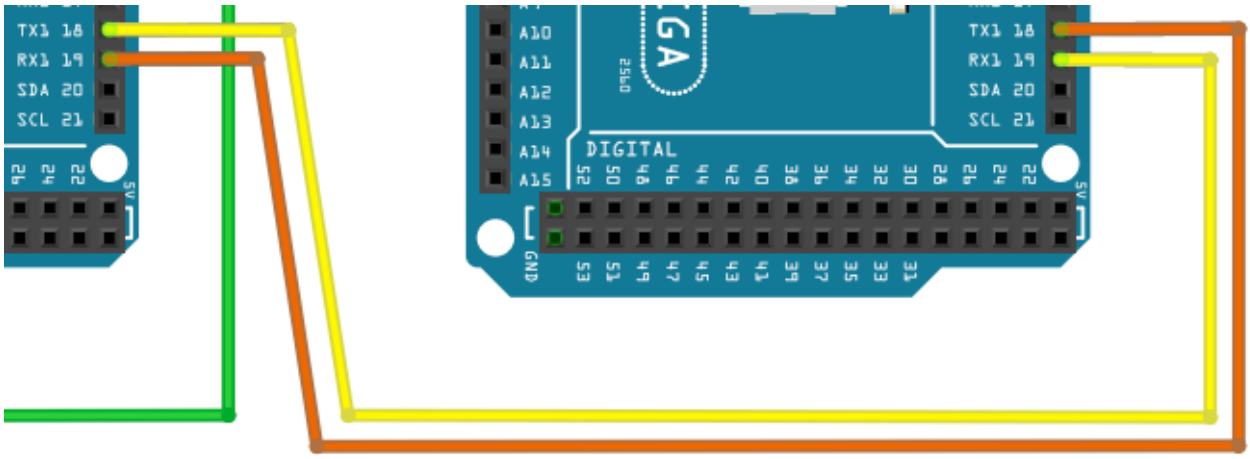
Arduino 1



Arduino 2



Connexion liaison série UART entre les deux Arduinos



Une fois le câblage réalisé, **faire** vérifier par l'enseignant.

Programme à téléverser dans les micro-contrôleurs Arduino

Les programmes sont opérationnels et donnés ci-après.
Réaliser un simple copier coller dans l'IDE Arduino procéder au téléversement des programmes dans chacune des cartes.
Attention : ne pas oublier de choisir le type de carte et le bon port COM pour les téléversements.

Un fois téléversé **procéder** aux essais.

Programme à téléverser dans l'arduino 1 – Emetteur

```
// ARDUINO 1 EMISSION

char lettreRecue;
const byte button=7;
byte etatButton;

void setup()
{
	pinMode(button , INPUT_PULLUP); // configuration de la broche 7
	Serial.begin(9600);//initialisation de la communication série matérielle Tx0/Rx0
	Serial1.begin (9600);//initialisation de la communication série matérielle Tx1/Rx1

	Serial.println ("Bienvenue l'activité \"Communication entre Arduinos\"");
	Serial.println ("Carte d'envoi de la commande de la led sur l'autre carte");
	delay(1000);
	Serial.println("");
}

void loop()
{
	etatButton=digitalRead(button);
	if (etatButton==LOW)
	{
		Serial.println("bouton appuyé");
		Serial1.print('1');
	}

	else
	{
		Serial.println("bouton relaché");
		Serial1.print('0');
	}
}
```

Programme à téléverser dans l'arduino 1 – Emetteur

```
//ARDUINO 2 RECEPTION

const byte led=2;
char donneeRecue;

void setup()
{
	pinMode(2,OUTPUT);
	Serial.begin(9600);
	Serial1.begin(9600);
	Serial.println ("Bienvenue l'activité \"Communication entre Arduinos\"");
	Serial.println ("Carte de réception de l'envoi l'autre carte");
	delay(1000);
	Serial.println("");
}

void loop()
{
	if (Serial1.available()>0)
	{
		donneeRecue=Serial1.read();
	}
	if(donneeRecue=='1')
	{
		digitalWrite(led,HIGH);
	}

	if(donneeRecue=='0')
	{
		digitalWrite(led,LOW);
	}
}
```

Analyse des programmes

Quelques précisions :

Deux communications séries sont utilisées :

- Serial (Tx0/Rx0) pour la communication avec l'ordinateur
- Serial1(Tx1/Rx1) pour la communication entre les deux Arduinos

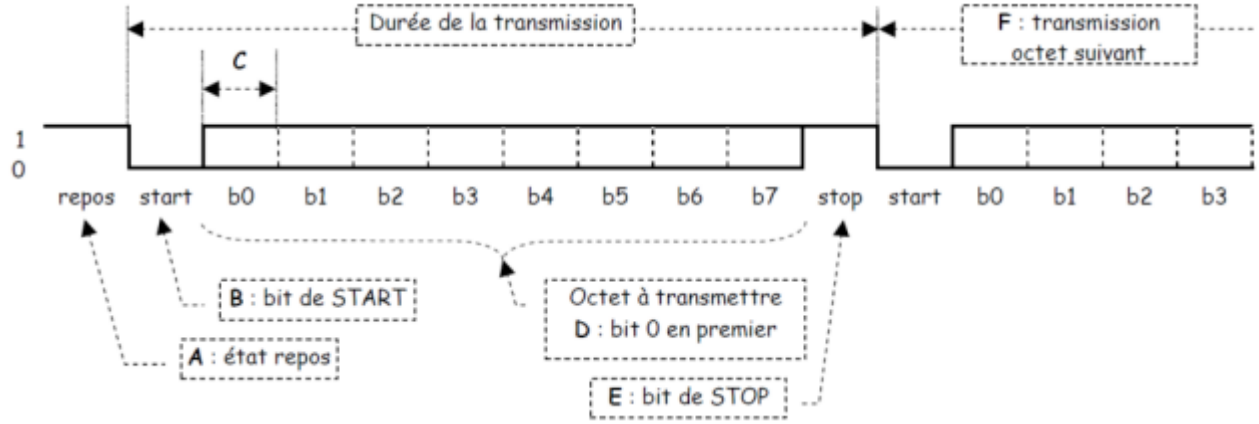
Tx est la broche Transmitter (Transmission de l'information)
Rx est la broche Receiver (Réception de l'information)
Le Tx de l'un doit être branché sur le Rx de l'autre et réciproquement !

Explication simple des codes

Le programme de l'Arduino 1 récupère l'état du bouton poussoir et envoie un « 1 » sur la liaison série Serial1 lorsque l'on appuie sur le bouton poussoir. Il envoie en « 0 » si l'on n'appuie pas sur le bouton poussoir.
Prendre le temps d'analyser le code des deux micro-contrôleurs !

Analyse de la trame transmise

On rappelle la structure d'une trame série transmise avec le protocole UART.



Visualisation de la trame Série

Nous avons vu que le caractère « 1 » est envoyé sur la liaison Serial1 lorsque l'on appuie et le caractère « 0 » lorsque l'on relache le bouton poussoir.

Proposer un schéma de raccordement de l'oscilloscope numérique (classique ou Picoscope) pour visualiser la trame transmise.

Après vérification du câblage à faire hors tension **procéder** à la visualisation des deux trames (trame 1 pour l'envoie du « 1 », trame 0 pour l'envoie de « 0 ») lorsque vous appuyez ou pas sur le bouton poussoir.

Décodage des trames (sans fonctionnalité de décodage de l'oscilloscope)

On rappelle le code ASCII utilisé dans la communication UART (annexe 1).

Donner la traduction en binaire des deux caractères envoyés :

Caractère envoyé sur la liaison série	*Codification en binaire nature*
Caractère 0	
Caractère 1	

Sachant que la configuration de la liaison série - #ligne Serial1.begin(9600) – est de 9600 bits/s, pas de parité, un bit de stop, **dessiner** ci-après les trames envoyées pour les caractères « 0 » et « 1 ».

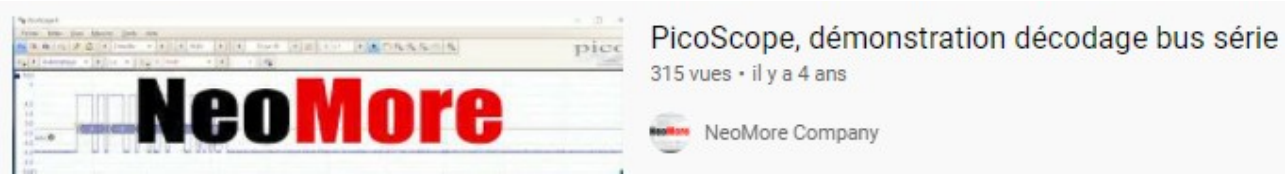
Comme vu ci-avant, la communication série Serial1 est configurée à 9600 bits/s.
Indiquer combien de temps il faut à la communication pour transférer le caractère « 0 » ou « 1 » ?

Faire l'acquisition à l'oscilloscope et **montrer** à l'enseignant que vous arrivez à faire le décodage du caractère tranmis.

Décodage des trames avec fonctionnalité de décodage de l'oscilloscope

Pour cette partie vous devez absolument utiliser les oscilloscopes Picoscope !

Afin de réussir parfaitement l'acquisition du signal et son décodage, prendre connaissance de la vidéo suivante (cliquer sur l'image)



Une fois la vidéo visualisée, **réaliser** l'acquisition de la trame et **procéder** au décodage.
À l'aide de la Référence en ligne Arduino, vous aller modifier les paramètres de la communication série entre les deux Arduinos.

La ligne à modifier dans le code est l'une des suivantes :
Attention à modifier la bonne communication série, pas les deux !;)

```
9   Serial.begin(9600);
10  Serial1.begin(9600);
```

Paramétrer la communication entre les arduinos de la manière suivante :

- 7 bits de données
- 1 bit de parité
- 2 bits de stop

Une fois les paramétrages effectués et les programmes téléversés dans les bonnes cartes, **réaliser** l'acquisition de de la trame et de son décodage. **Relever** la trame et son décodage sur papier libre. **Vérifier** l'adéquation du bit de parité avec la donnée transmise (« 0 » ou « 1 »).

Envoi d'un message Fox par liaison série et décodage de trame

Téléverser les deux programmes ci-après dans les cartes.
Arduino 1 ⇒émission, Arduino 2 ⇒réception

```
//ARDUINO 1 EMISSION
void setup()
{
        Serial.begin(9600);
        Serial1.begin(9600);
}

void loop()
{
        Serial.println("THE QUICK BROWN FOX JUMPS OVER A LAZY DOG...");
        Serial1.println("THE QUICK BROWN FOX JUMPS OVER A LAZY DOG...");
        delay(1000);
}
```

```
//ARDUINO 2 RECEPTION

String donneeRecue;

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop()
{
  if (Serial1.available()>=0)
  {
    donneeRecue=Serial1.readStringUntil('\n');
    Serial.println(donneeRecue);
  }
}
```

Procéder à la capture de trame et à son décodage.

ANNEXE 1

Table de codes ASCII

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