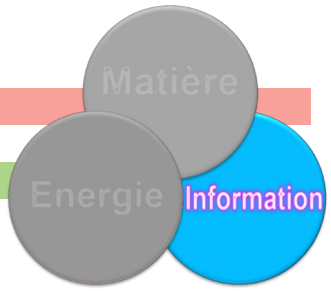
STI2D	Thème N°1.B : Réseaux et communications informatiques	LA MACHE
SIN - Terminale		



Durée prévue : 10h

Objectifs :

- Être capable de caractériser un réseau informatique
- Être capable d'analyser une communication informatique

Prérequis :

- Connaissance du binaire, décimal, ….
- Connaissances de base sur les réseaux (voir les TP)

Modalités :

- cours + TP

Documents ressources :

- site internet et fichier numérique

Plan de l'étude :

I. Introduction..... 1
II. Réseaux informatiques..... 2
1. Principes généraux..... 2
2. Éléments d'un réseau..... 5
3. Adresses des éléments d'un réseau..... 7
4. Le modèle de référence OSI..... 11
5. Comparaison des modèles OSI et TCP/IP..... 13
6. Principe de l'adressage et de l'encapsulation..... 13
7. Topologie des réseaux..... 14
III. Communications informatiques..... 20
1. Les supports de transmission..... 20
2. Exemple N°1 : la liaison série (RS232 et Arduino)..... 27
3. Exemple N°2 : le bus I^2C..... 35
IV. Exercices..... 41

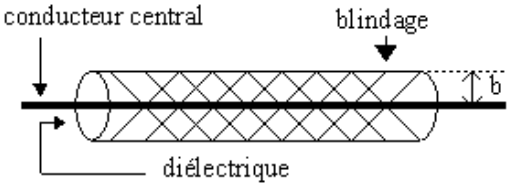
III. Communications informatiques

1. Les supports de transmission

Le cuivre : câbles coaxiaux ou à paires torsadées

Le **câble coaxial** est un câble permettant entre autres le transport de données. Il est composé, au moins, de deux conducteurs.

L'âme centrale, qui peut être mono-brin ou multi-brins (en cuivre ou en cuivre argenté, voire en acier cuivré), est entourée d'un matériau diélectrique (isolant). L'isolant est entouré d'une tresse conductrice (ou feuille d'aluminium enroulée), puis d'une gaine isolante et protectrice.



Une **paire torsadée** est formée de deux fils conducteurs enroulés en hélice l'un autour de l'autre. Un câble peut contenir plusieurs paires torsadées.

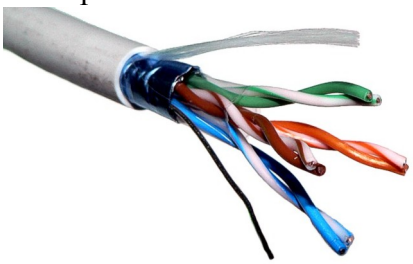
Pour limiter les interférences, les paires torsadées sont souvent blindées. Comme le blindage est fait de métal, celui-ci constitue également un référentiel de masse. Le blindage peut être appliqué individuellement aux paires, ou à l'ensemble formé par celles-ci.

Tableau récapitulatif avec les dénominations officielles (norme ISO/IEC 11801) :

Dénomination courante	**Désignation**	**Dénomination officielle**	**Blindage de l'ensemble du câble**	**Blindage des paires individuelles**
UTP	*Unshielded twisted pair* : Paire torsadée non blindée	U/UTP	aucun	aucun
STP	*Shielded twisted pair*: Paire torsadée blindée	U/FTP	aucun	feuillard
FTP	*Foiled twisted pair* : Paire torsadée écrantée	F/UTP	feuillard	aucun
FFTP	*Foiled foiled twisted pair:* Paire torsadée doublement écrantée	F/FTP	feuillard	feuillard
SFTP	*Shielded foiled twisted pair:* Paire torsadée écrantée et blindée :	SF/UTP	feuillard, tresse	aucun
SSTP	*Shielded shielded twisted pair:* Paire torsadée doublement blindée	S/FTP	Tresse	feuillard

Légende :
- TP = twisted pair ou paire torsadée
- U = unshielded ou non blindé
- F = foil shielding ou blindage par feuillard
- S = braided shielding ou blindage par tresse

Exemple d'un câble FTP :



Les câble sont aussi caractérisées par leur catégorie :

Catégorie	Classe	Impédance	Fréquence max.	Application
3	C	100-120 Ω	16 MHz	Token Ring 4 Mbit/s, 10 Base T, Fast Ethernet, 100 VG Any, LAN 100 Base T4
4	D	100 Ω	20 MHz	Token Ring 16 Mbit/s
5	D	100 Ω	100 MHz	Câble UTP et FTP, 100 Base Tx, ATM 155 Mbit/s, 1000 Base T (Cat 5E)
6	E	100 Ω	250 MHz	Câble FTP et SFTP, 1000 Base Tx
6a	E	100 Ω	500 MHz	Câble FTP et SFTP, 1000 Base Tx, 10 G Base T
7	F	100 Ω	600 MHz	Câble SFTP

Divers organismes de normalisation contribuent à la définition des propriétés physiques, électriques et mécaniques des supports disponibles pour différentes communications de données. Ces spécifications garantissent que les câbles et connecteurs fonctionnent comme prévu avec différentes mises en œuvre.

Par exemple, des normes pour les supports en cuivre sont définies pour :

- Le type de câblage en cuivre utilisé
- La bande passante de la communication
- Le type de connecteurs utilisés
- Le brochage et les codes couleur des connexions avec le support
- La distance maximale du support

Un exemple de connecteur utilisé pour les liaisons Ethernet : RJ45

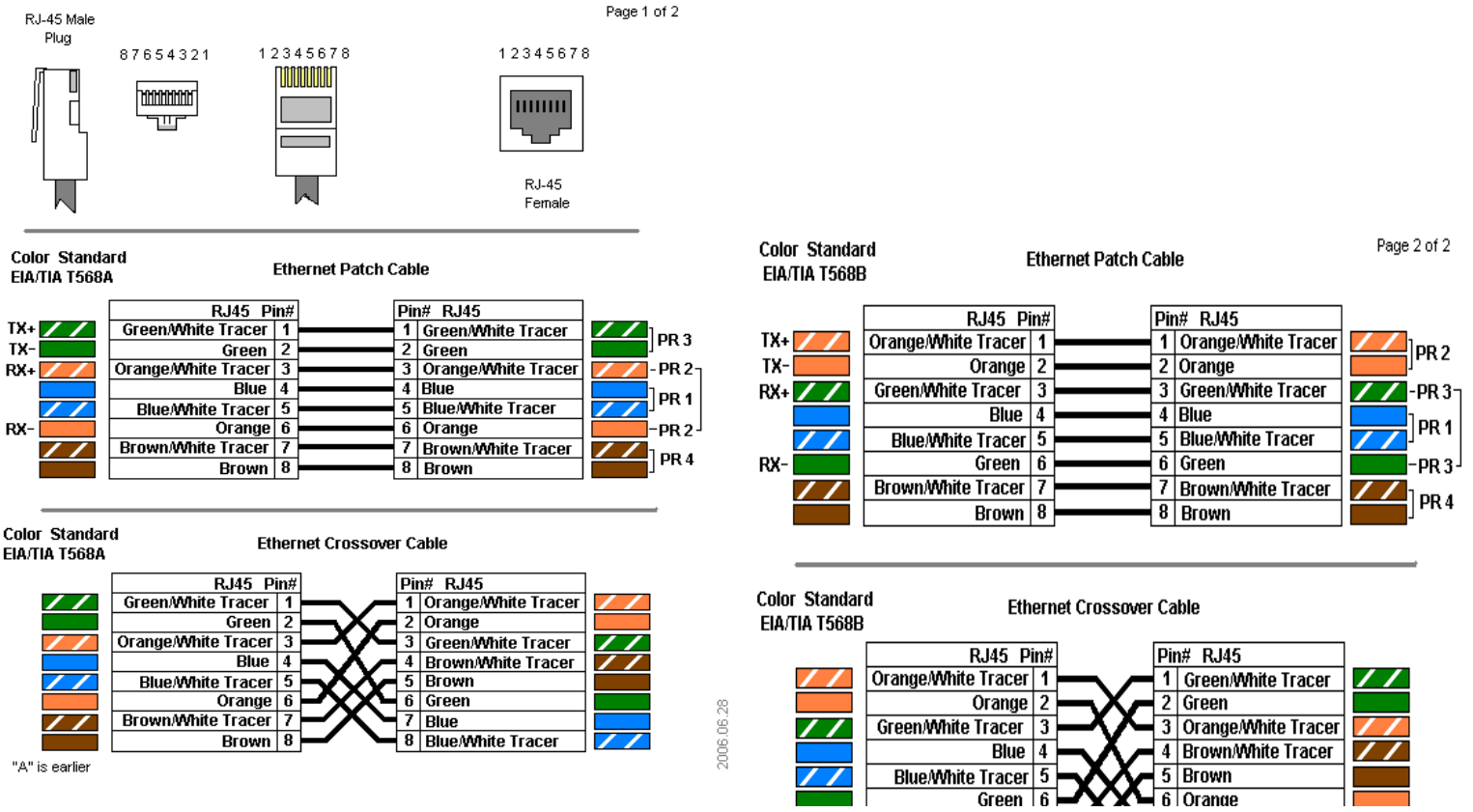
RJ45 est le nom usuel du connecteur 8P8C (8 positions et 8 contacts électriques) utilisé couramment pour les connexions Ethernet, et plus rarement pour les réseaux téléphoniques. La référence « RJ » vient de l'anglais Registered Jack (prise jack enregistrée).

Câblage :
Lors d'un câblage informatique en 10/100 Mbit/s, seules les quatre broches 1-2 et 3-6 sont utilisées pour transmettre les informations. Lors d'un câblage informatique en 1 000 Mbit/s (1 Gbit/s), les 8 broches sont utilisées.

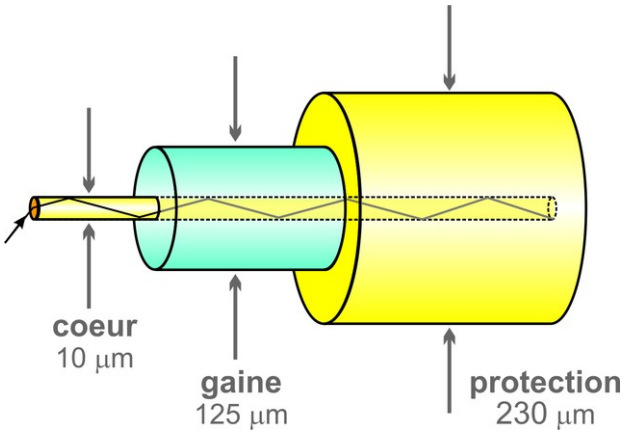
- Lorsqu'on branche un poste de travail dans un concentrateur (hub) ou un commutateur (switch), un câble droit doit être utilisé.
- Lorsqu'on doit brancher deux postes de travail ensemble, un câble croisé doit être utilisé. Dans le câble croisé, les paires utiles sont inversées, c'est-à-dire que la paire de transmission d'un côté est connectée aux broches de réception de l'autre côté.

La règle générale est la suivante : pour deux périphériques travaillant au niveau de la couche 2 (MAC) du modèle OSI comme un Hub Ethernet ou un switch sans fonction de routage, ou deux périphériques de la couche 3 (IP) comme un PC ou un routeur, on utilise un câble croisé. Dès que l'on change de couche entre deux équipements, on peut alors utiliser un câble droit (PC à Switch, Routeur à Switch, Hub à PC, etc.).

Il existe plusieurs normes (T568A et T568B) : même si ces deux normes sont déployées, la norme T568A est principalement utilisée dans le domaine du résidentiel (souvent avec du câblage simple non blindé de type UTP) alors que la norme T568B est plutôt employée dans le domaine professionnel.



Le verre : fibre optique



Le câblage en fibre optique utilise des fibres de verre ou de plastique pour guider des impulsions lumineuses de la source à la destination. Au final la fibre optique est un guide d'onde qui exploite les propriétés réfractrices de la lumière. Elle est habituellement constituée d'un cœur entouré d'une gaine.

Comme l'électronique est essentiellement basée sur l'électricité il faudra convertir les bits électriques (tension) en lumière pour la transmission et inversement pour la réception.

Les bits sont codés sur la fibre sous forme d'impulsions lumineuses. Les émetteurs utilisés sont de trois types :

- les diodes électroluminescentes (LED) qui fonctionnent dans le proche infrarouge (850 nm),
- les lasers, utilisés pour la fibre monomode, dont la longueur d'onde est 1 310 ou 1 550 nm,
- les diodes à infrarouge qui émettent dans l'infrarouge à 1 300 nm.

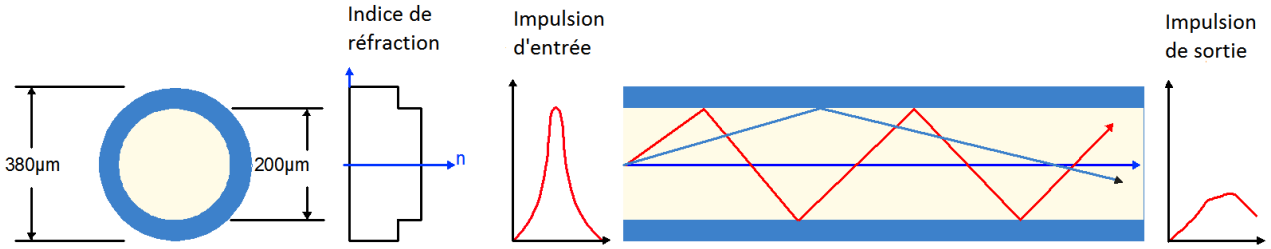
Le câblage en fibre optique prend en charge des débits de bande passante de données brutes très élevés. Des dispositifs à semi-conducteur électronique appelés photodiodes détectent les impulsions lumineuses et les convertissent en tensions qui peuvent ensuite être reconstituées en trames de données.
Au cours de son parcours, le signal est atténué et déformé : des répéteurs et des amplificateurs placés à intervalles réguliers permettent de conserver l'authenticité du message.

Fibres multimodes

Les fibres multimodes (dites MMF, pour Multi Mode Fiber), ont été les premières sur le marché. Ce sont les plus courantes. Elles ont pour caractéristique de transporter plusieurs modes (trajets lumineux). Du fait de la dispersion modale, on constate un étalement temporel du signal proportionnel à la longueur de la fibre. En conséquence, elles sont utilisées uniquement pour des bas débits ou de courtes distances. Elles sont caractérisées par un diamètre de cœur de plusieurs dizaines à plusieurs centaines de micromètres (les cœurs en multimodes sont de 50 ou 62,5 µm pour le bas débit). Cependant les fibres les plus récentes, de type OM3, permettent d'atteindre le Gbit/s sur des distances de l'ordre du km.

Il existe plusieurs modes de propagation de la lumière au sein de son cœur de silice (verre).

Multimode à saut d'indice :



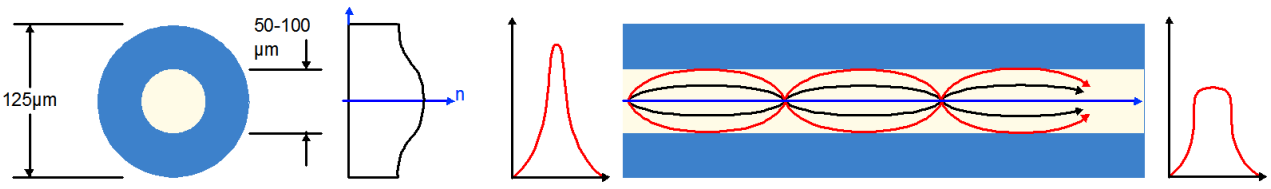
Fibre à saut d'indice

L'atténuation sur ce type de fibre est très importante comme on peut le voir sur la différence des impulsions d'entrée et de sortie.

- Débit: environ 100 Mbit/s
- Portée maximale: environ 2 Km

Multimode à gradient d'indice :

La fibre multimode à gradient d'indice est elle aussi utilisée dans les réseaux locaux. C'est une fibre multimode, donc plusieurs modes de propagation coexistent. A la différence de la fibre à saut d'indice, il n'y a pas de grande différence d'indice de réfraction entre cœur et gaine.
Cependant, le cœur des fibres à gradient d'indice est constitué de plusieurs couches de matière ayant un indice de réfraction de plus en plus élevé. Ces différentes couches de silice de densités multiples influent sur la direction des rayons lumineux, qui ont une forme elliptique.



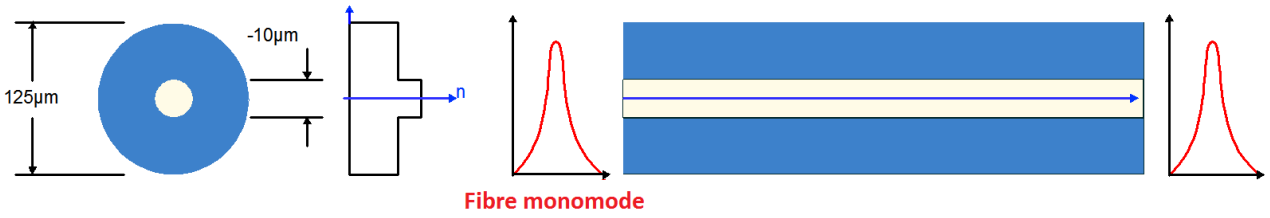
Fibre à gradient d'indice

La fibre à gradient d'indice possède un cœur de taille intermédiaire. L'atténuation sur ce type de fibre est moins importante que sur les fibres à saut d'indice.

- Débit: environ 1 Gbit/s
- Portée maximale: environ 2 Km

Fibres monomodes

Pour de plus longues distances et/ou de plus grands débits, on préfère utiliser des fibres monomodes (dites SMF, pour Single Mode Fiber), qui sont technologiquement plus avancées car plus fines (voire très fines). Ce sont les meilleures fibres optiques actuellement disponibles. Leur cœur très fin n'admet ainsi qu'un mode de propagation, le plus direct possible c'est-à-dire dans l'axe de la fibre. Les pertes sont donc minimes (moins de réflexion sur l'interface cœur/gaine) que cela soit pour de très haut débits ou de très longues distances. Les fibres monomodes sont de ce fait adaptées pour les lignes intercontinentales (câbles sous-marin).



- Débit: environ 100 Gbit/s
- Portée maximale: environ 100 Km

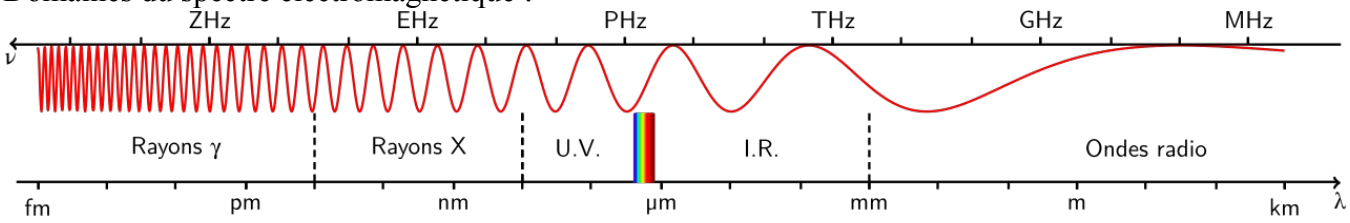
Ondes électromagnétiques : sans fil

Les supports sans fil transportent des signaux (sons, données, …) à l'aide d'ondes électromagnétiques.

Les technologies de communication de données sans fil fonctionnent bien dans les environnements ouverts. Cependant, certains matériaux de construction utilisés dans les bâtiments, ainsi que le terrain local (montagnes, collines, ...), limitent la couverture effective. De plus, la transmission sans fil est sensible aux interférences et peut être perturbée par des appareils aussi courants que les téléphones fixes sans fil, certains types d'éclairages fluorescents, les fours à micro-ondes et d'autres communications sans fil.

En outre, la couverture de communication sans fil n'exigeant aucun accès à un fil physique de support, des périphériques et utilisateurs non autorisés à accéder au réseau peuvent accéder à la transmission. La sécurité du réseau constitue par conséquent un composant essentiel de l'administration de réseau sans fil.

Domaines du spectre électromagnétique :



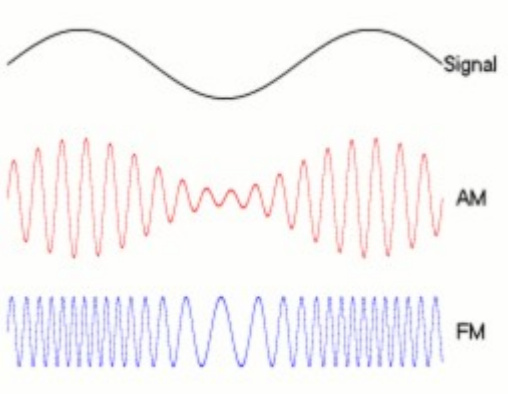
Nom	Longueur d'onde(m)	Fréquence(Hz)	Énergie du photon (eV)
Rayon gamma	< 10 pm	> 30 EHz	> 124 keV
Rayon X	10 pm – 10 nm	30 EHz – 30 PHz	124 keV – 124 eV
Ultraviolet	10 nm – 390 nm	30 PHz – 750 THz	124 eV – 3,2 eV
Visible	390 nm – 750 nm	770 THz – 400 THz	3,2 eV – 1,7 eV
Infrarouge	750 nm – 0,1 mm	400 THz – 3 THz	1,7 eV – 12,4 meV
Térahertz / submillimétrique	0,1 mm - 1 mm	3 THz - 300 GHz	12,4 meV - 1,24 meV
Micro-ondes	1 mm - 1 m	300 GHz - 300 MHz	1,24 meV - 1,24 µeV
Ondes radio	1 m – 100 000 km	300 MHz – 3 Hz	1,24 µeV – 12,4 feV

Les normes et technologies les plus courantes sont le GSM, le Wifi, le Bluetooth. Il en existe beaucoup d'autres.

Bande de fréquence	Service/Application
9 kHz-30 MHz	Radio grandes ondes, ondes moyennes et ondes courtes, détecteurs de victimes d'avalanches, systèmes RFID, applications médicales, plaques de cuisson à induction, CPL...
30 MHz-87,5 MHz	Télédiffusion (bande I), réseaux taxis, pompiers... radioamateurs, microphones sans fil, radars...
87,5 MHz - 108 MHz	Bande FM (modulation de fréquence)
108 MHz - 136 MHz	Trafic aéronautique
136 MHz - 400 MHz	Télédiffusion (bande II et III), réseaux professionnels (police, pompiers, SAMU...), vol libre (talkie-walkie), trafic amateur, trafic maritime, radiomessagerie...
400 MHz - 470 MHz	Balises ARGOS, réseaux professionnels (SNCF, EDF...), télécommandes, télémesure médicale, réseaux cellulaires
470 MHz - 860 MHz	Télédiffusion bande IV et V
704 MHz - 960 MHz	Téléphonie mobile bandes des 700, 800 et 900 MHz
960 MHz - 1 710 MHz	Radiodiffusion numérique, faisceaux hertziens
1 710 MHz - 1 880 MHz	Téléphonie mobile, bande 1 800 MHz
1 880 MHz - 1 900 MHz	Téléphonie DECT
1 900 MHz - 2 170 MHz	Téléphonie mobile UMTS
2 400 MHz - 2 500 MHz	Réseaux Wi-Fi, Bluetooth, four à micro-ondes
2 500 MHz - 2 690 MHz	Téléphonie mobile (LTE), bande des 2 600 MHz
3 400 MHz - 3 600 MHz	Boucle locale radio de type WiMAX

Exemple d'utilisation des ondes électromagnétiques pour transporter des informations

Au temps de l'analogique :



Ci-contre les 2 principales méthodes pour transmettre une information analogique (ici du son) à l'aide d'ondes électromagnétiques.
AM : modulation d'amplitude (l'amplitude du signal varie en fonction du signal à transporter)
FM : modulation de fréquence (la fréquence du signal varie en fonction du signal à transporter)

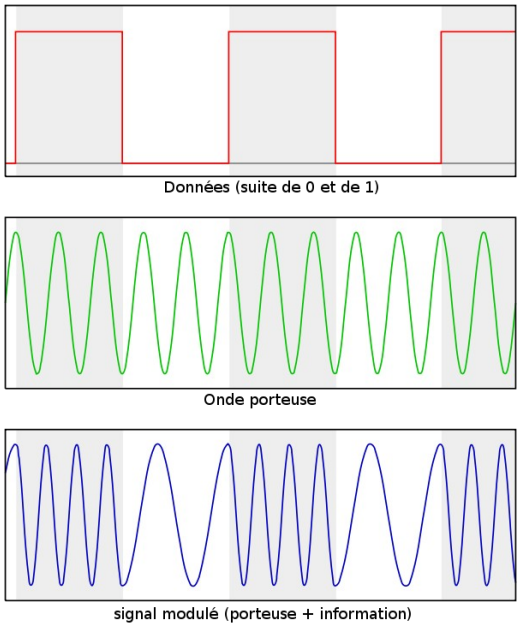
Maintenant en numérique.

On utilise aussi les 2 principes utilisés du temps de l'analogique (modulation d'amplitude et modulation de fréquence) parfois en les combinant.

Exemple :

La modulation par déplacement de fréquence (MDF), plus connue sous sa dénomination anglophone de frequency-shift keying (FSK) est un mode de modulation de fréquence numérique dans lequel le signal modulé varie entre des fréquences prédéterminées.

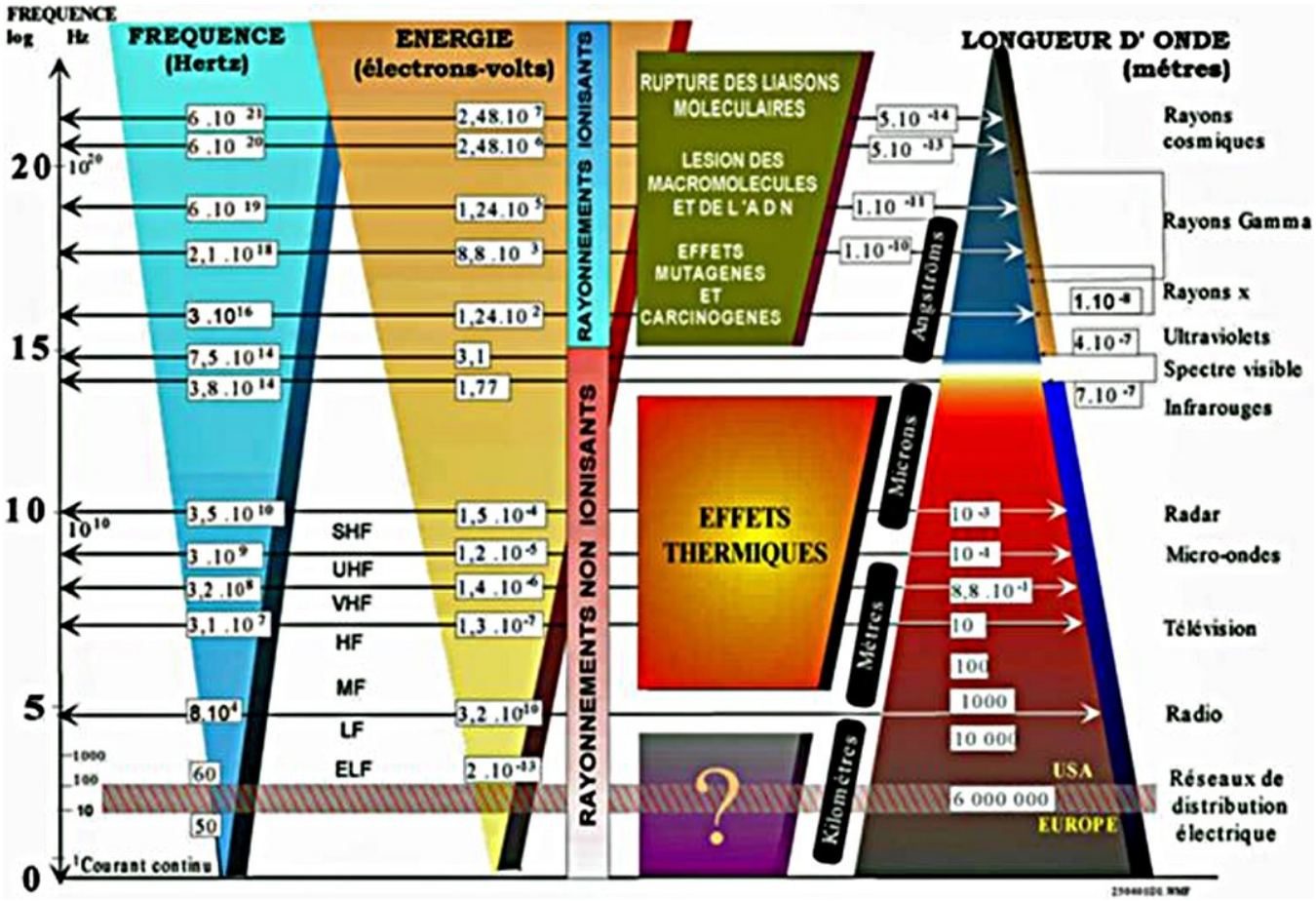
Il y a bien sûr de nombreuses autres manières de transporter des données numériques mais elles n'entrent pas dans le cadre de ce cours.



Risques sanitaires des télécommunications

Les champs électromagnétiques créées par les télécommunications hertziennes (téléphone mobile, le téléphone domestique sans fil, le Wi-Fi, ou encore les antenne-relais de téléphonie mobile) sont perçus par certains comme une « pollution électromagnétique » dangereuse pour la santé, alors que les études scientifiques de ces dernières années ont donné des résultats contradictoires. Il semble quand même y avoir un risque que les études scientifiques n'ont pas encore permis de confirmer totalement ni d'évaluer (il faut dire que les études sont assez rares et celles qui existent sont souvent financées par les sociétés qui émettent ces ondes électromagnétiques donc leurs résultats ne sont pas toujours fiables).

Bref il s'agit d'un sujet à controverse et les impacts sur le très long terme de l'exposition au rayonnement électromagnétique issus des appareils modernes sont encore relativement méconnus.



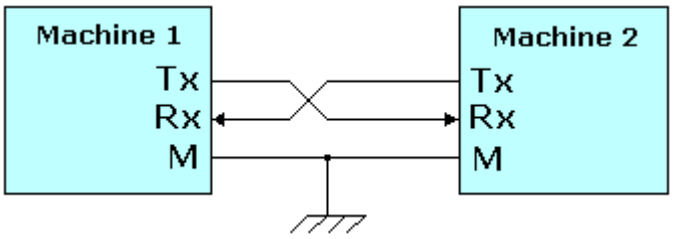
2. Exemple N°1 : la liaison série (RS232 et Arduino)

La liaison série va permettre de faire communiquer 2 appareils (ordinateurs, capteurs, …).

Des liaisons séries, il en existe beaucoup (RS-232, Universal Serial Bus (USB), Serial ATA, SPI, …). Nous allons nous intéresser à la RS-232, qui est très répandue ainsi qu'à la version présente sur les cartes micro-contrôleur Arduino.

Pour communiquer via la voie série, deux dispositifs doivent avoir 3 câbles minimum.

- le premier câble est la référence électrique, communément appelée masse électrique. Cela permet de prendre les mesures de tension en se fixant un même référentiel. Dans notre cas, on considérera que le 0V sera notre référentiel électrique commun.
- Les deux autres câbles permettent la transmission des données. L'un sert à l'envoi des données pour un émetteur, mais sert aussi pour la réception des données venant de l'autre émetteur. Idem pour l'autre câble. Il permet l'émission de l'un et la réception de l'autre.



Il s'agit du strict minimum. La norme n'interdit pas l'utilisation d'autres câbles qui servent à faire du contrôle de flux et de la gestion des erreurs.

La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en BAUDS (1 baud correspond à 1 bit par seconde). Il existe différentes vitesses normalisées: 9600, 4800, 2400, 1200... bauds
La communication peut se faire dans les deux sens (duplex), soit émission d'abord, puis réception ensuite (half-duplex), soit émission et réception simultanées (full-duplex)
La transmission étant du type asynchrone (pas d'horloge commune entre l'émetteur et le récepteur), des bits supplémentaires sont indispensables au fonctionnement: bit de début de mot (start), bit(s) de fin de mot (stop).

Le principal intérêt de la communication série est le nombre de fils réduits: la communication la plus simple peut être faite sur 3 fils (Tx, Rx et masse).

Le signal électrique et le protocole

Avant tout, il faut savoir que pour communiquer, deux dispositifs électronique ou informatique utilisent des données sous forme de bits. Ces bits sont des états logiques (vrai ou faux, 0 ou 1) qui peuvent être regroupés pour faire des ensembles de bits. Quand ces ensembles de bits sont constitués de 8 bits ils forment alors un octet. Jusqu'ici rien de nouveau !

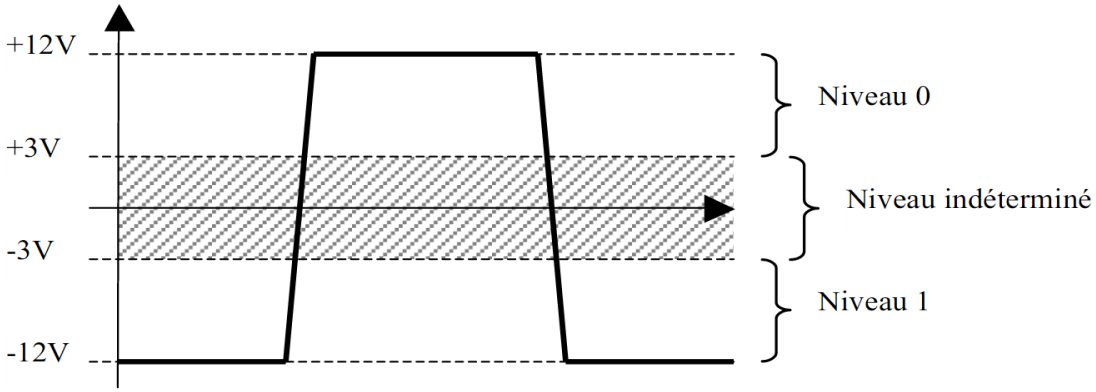
Les tensions utilisées

Ces bits sont en fait des niveaux de tension électrique. Et la norme RS-232 définit quelles tensions doivent être utilisées (ici norme V28 présente sur les ordinateurs) :

	Niveau logique 0	**Niveau logique 1**
Tension électrique minimale	+3V	-3V
Tension électrique maximale	+25V	-25V

Ce qui donne : -3v > 1 logique > -25v et +3v < 0 logique < +25v

En général on obtient :



On note que les tensions comprises entre +3V et -3V sont ignorées car c'est dans ces zones là que se trouvent la plupart des parasites. C'est un moyen permettant d'éviter un certain nombre d'erreurs de transmissions.
C'est aussi un moyen de contrôler que la liaison physique (les fils) est correcte : lorsqu'il n'y a pas de communication sur la voie série, il y a ce qu'on appelle un état de repos. C'est à dire un niveau logique toujours présent. Il s'agit du niveau logique 1. Soit une tension comprise entre -3V et -25V. Si cet état de repos n'est pas présent, c'est qu'il peut y avoir un problème de câblage (fil coupé, …).

Les données

Les données qui transitent par la voie série sont transmises sous une forme numérique (binaire). C'est à dire avec des niveaux logiques 0 et 1. Prenons une donnée que nous voudrions envoyer, par exemple la lettre « P » majuscule. Il faut savoir qu'une lettre du clavier est codé sur un nombre de 8 bits, donc un octet. Réellement c'est en fait sur 7 bits qu'elle est codée, mais en rajoutant un 0 devant le codage, cela conserve sa valeur et permet d'avoir un codage de la lettre sur 8 bits. Ces codes sont définis selon la table ASCII. Ainsi, pour chaque caractère du clavier, on retrouve un codage sur 8 bits.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

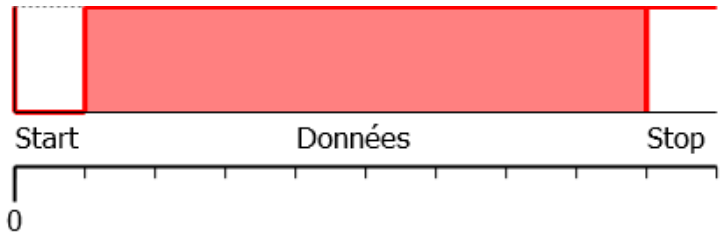
En observant la table, on tombe sur la lettre « P » majuscule et l'on voit que sa correspondance en décimal est 80 donc en binaire : 01010000.

L'ordre et les délimiteurs

On va à présent voir comment est transmit un octet sur la voie série en envoyant notre exemple, la lettre « P ».

Pour comprendre suivons l'analogie d'un appel téléphonique :

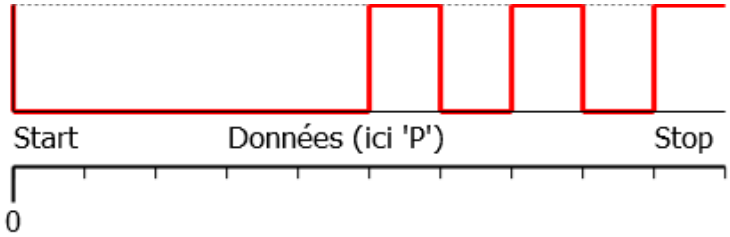
- Lorsque l'on passe un coup de fil, bien généralement on commence par dire « Bonjour » ou « Allo ». Ce début de message permet de faire l'ouverture de la conversation. En effet, si l'on reçoit un appel et que personne ne répond après avoir décroché, la conversation ne peut avoir lieu. Dans la norme RS-232, on va avoir une ouverture de la communication grâce à un bit de départ. C'est lui qui va engager la conversation avec son interlocuteur. Dans la norme RS-232, ce dernier est un état 0.
- Ensuite, vous allez commencer à parler et donner les informations que vous souhaitez transmettre. Ce sera les données. L'élément principal de la conversation (ici notre lettre 'P').
- Enfin, après avoir renseigné tout ce que vous aviez à dire, vous terminez la conversation par un « Au revoir ». Cela termine la conversation. Il y aura donc un bit de fin ou bit de stop qui fera de même sur la voie série. Dans la norme RS-232, c'est un état 1.



C'est de cette manière là que la communication série fonctionne. D'ailleurs, savez-vous pourquoi la voie série s'appelle ainsi ? En fait, c'est parce que les données à transmettre sont envoyées une par une, à la queue leu-leu. Exactement comme une conversation entre deux personnes : la personne qui parle ne peut pas dire plusieurs phrases en même temps, ni plusieurs mots ou sons. Chaque élément se suit selon un ordre logique. L'image précédente résume la communication que l'on vient d'avoir, il n'y a plus qu'à la compléter pour envoyer la lettre « P ».

Donc il y a le bit de start, notre lettre P et le bit de stop. D'après ce qu'on a dit, cela donnerait, dans l'ordre, ceci : 0 (Start) 01010000 (Données) et 1 (Stop).

Eh bien… c'est presque ça. Sauf que les ingénieurs qui ont inventé ce protocole ont eu la bonne idée de transmettre les données à l'envers… Par conséquent, la bonne réponse était : 0(Start)00001010(données)1(Stop) c'est à dire 0000010101. Avec un chronogramme, on observerait ceci :

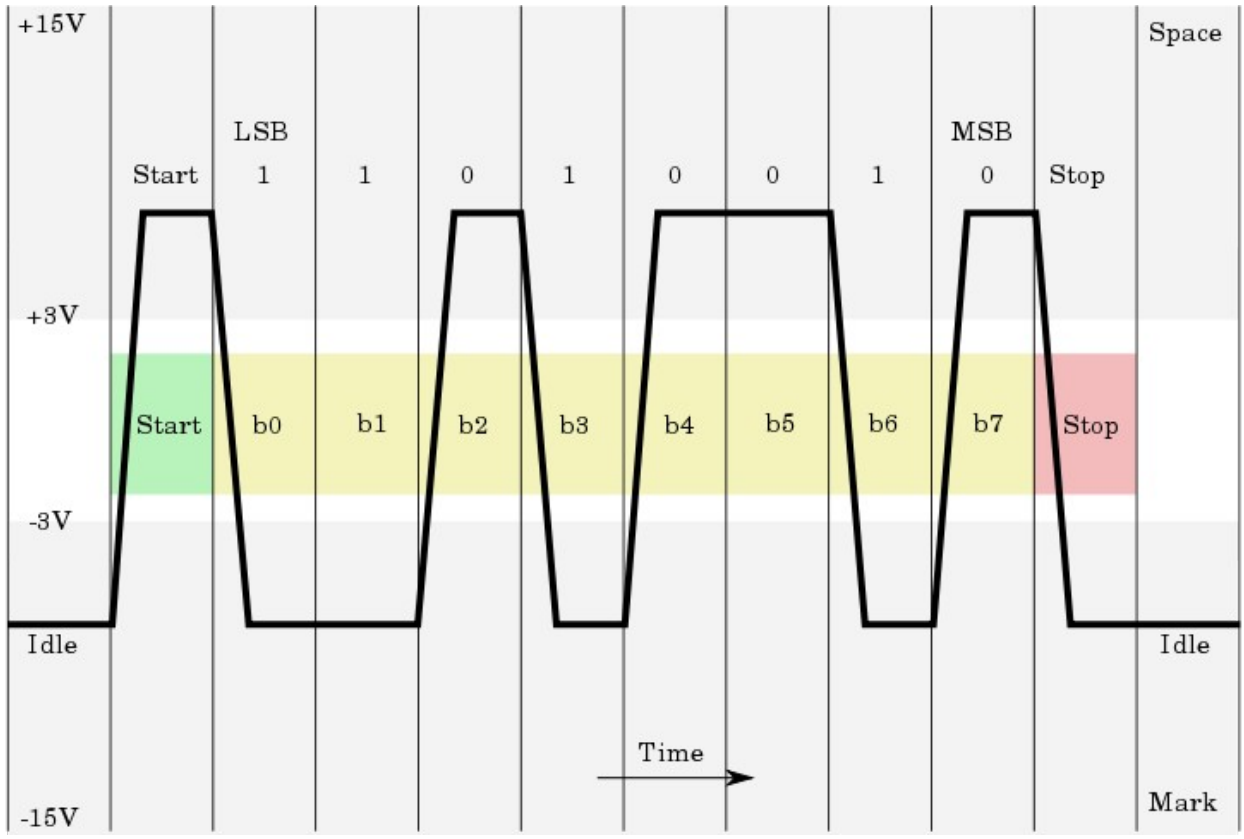


Un peu de vocabulaire

Les données sont donc envoyées à l'envers. Ce qu'il faut savoir c'est que le bit de donnée qui vient après le bit de start s'appelle le bit de poids faible ou LSB en anglais pour Less Significant Bit. C'est un peu comme un nombre qui a des unités (tout à droite), des dizaines, des centaines, des milliers (à gauche), etc. Par exemple le nombre 6395 possède 5 unités (à droite), 9 dizaines, 3 centaines et 6 milliers (à gauche). On peut faire référence au bit de poids faible en binaire qui est donc à droite. Plus on

s'éloigne et plus on monte vers… le bit de poids fort ou MSB en anglais pour Most Significant Bit. Et comme les données sont envoyées à l'envers sur la liaisons série, on aura le bit de poids faible juste après le start, donc à gauche et le bit de poids fort à droite.
Il est donc essentiel de savoir où est le bit de poids faible pour pouvoir lire les données à l'endroit. Sinon on se retrouve avec une donnée erronée !

Pour regrouper un peu tout ce que l'on a vu sur le protocole de la norme RS-232, voici une image d'une trame :



Vous devrez être capable de trouver quel est le caractère envoyé sur cette trame… alors ? Indice : c'est une lettre… On lit les niveaux logiques de gauche à droite, soit 11010010 ; puis on les retourne soit 01001011 ; enfin on compare à la table ASCII et on trouve la lettre « K » majuscule. Attention aux tensions négatives qui correspondent à l'état logique 1 et les tensions positives à l'état logique 0.

La vitesse

La norme RS-232 définit la vitesse à laquelle sont envoyée les données. Elles sont exprimés en bit par seconde (bit/s). Elle préconise des vitesse inférieures à 20 000 bits/s. Sauf qu'en pratique, il est très courant d'utiliser des débits supérieurs pouvant atteindre les 115 200 bits/s. Quand on va utiliser la voie série, on va définir la vitesse à laquelle sont transférées les données. Cette vitesse dépend de plusieurs contraintes que sont : la longueur du câble utilisé reliant les deux interlocuteurs et la vitesse à laquelle les deux interlocuteurs peuvent se comprendre. Voilà ce que dit la norme RS-232 :

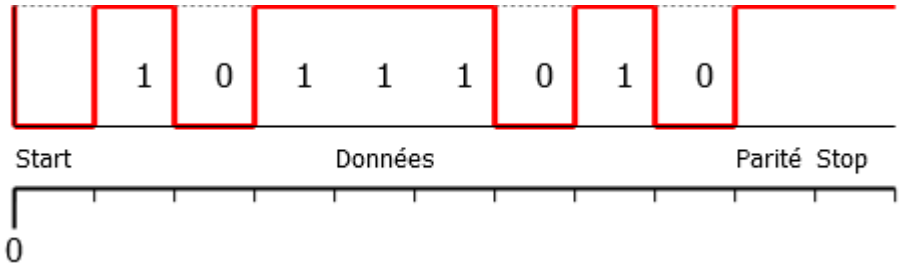
Débit (bit/s) - (bauds)	**Longueur (m)**
2 400	60
4 800	30
9 600	15
19 200	7,6
38 400	3,7
56 000	2,6

Plus le câble est court, plus le débit pourra être élevé car moins il y a d'affaiblissement des tensions et de risque de parasites.

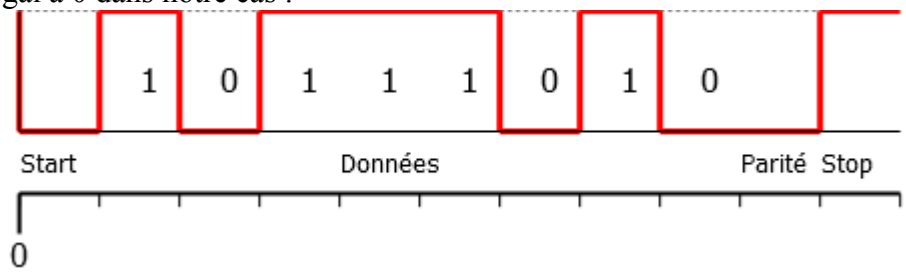
La gestion des erreurs

Malgré les tensions imposées par la norme, il arrive qu'il y ai d'autres parasites et que des erreurs de transmission surviennent. Pour limiter ce risque, il existe une solution. Elle consiste à ajouter un bit de contrôle appelé bit de parité. Juste avant le bit de stop, on va ajouter un bit qui sera pair ou impair. Donc, respectivement, soit un 0 soit un 1. Le paramétrage de la liaison va permettre de choisir une parité paire ou impaire.

Si l'on choisi une **parité paire**, alors le nombre de niveaux logiques 1 dans les données plus le bit de parité doit donner un nombre paire. Donc, dans le cas ci-dessous où il y a 5 niveaux logiques 1 sans le bit de parité, ce dernier devra prendre un niveau logique 1 pour que le nombre de 1 dans le signal soit paire. Soit 6 au total :



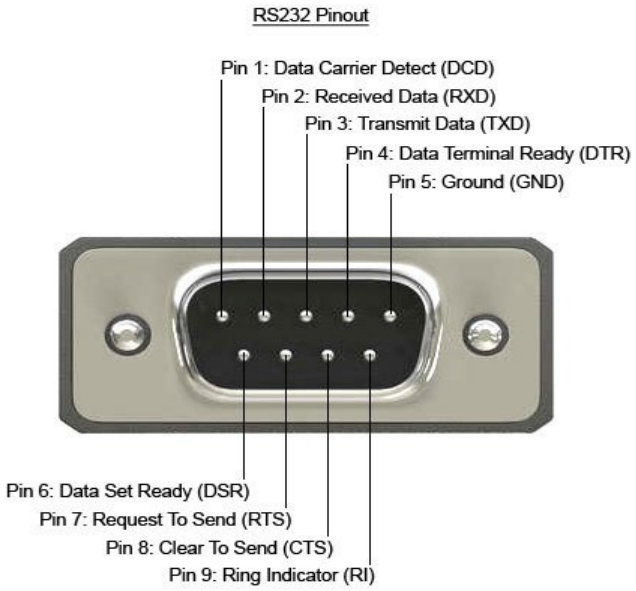
Dans le cas où l'on choisirait une **parité impaire**, alors dans le même signal où il y a 5 niveaux logiques 1, eh bien le bit de parité devra prendre la valeur qui garde un nombre impaire de 1 dans le signal. Soit un bit de parité égal à 0 dans notre cas :



Après, c'est le récepteur qui va vérifier si le nombre de niveaux logiques 1 est bien égale à ce que indique le bit de parité. Dans le cas où une erreur de transmissions serait survenu, ce sera au récepteur de traiter le problème et de demander à son interlocuteur de répéter.

Le connecteur de la liaison série RS232

Le connecteur utilisé par la norme RS232 est du type DB9 :



On notera la présence des 3 fils obligatoires au minimum : RxD (receive Data), TxD et la masse GROUND mais aussi d'autres possibilités de fils permettant une gestion plus efficace de la liaison série (on ne traitera pas de leur utilisation dans ce cours).

La liaison série de la carte microcontrôleur Arduino.

Prenons l'exemple suivant : le but est de connecter deux microcontrôleurs Arduino ensemble pour qu'ils puissent s'échanger des données.

La tension des microcontrôleurs :

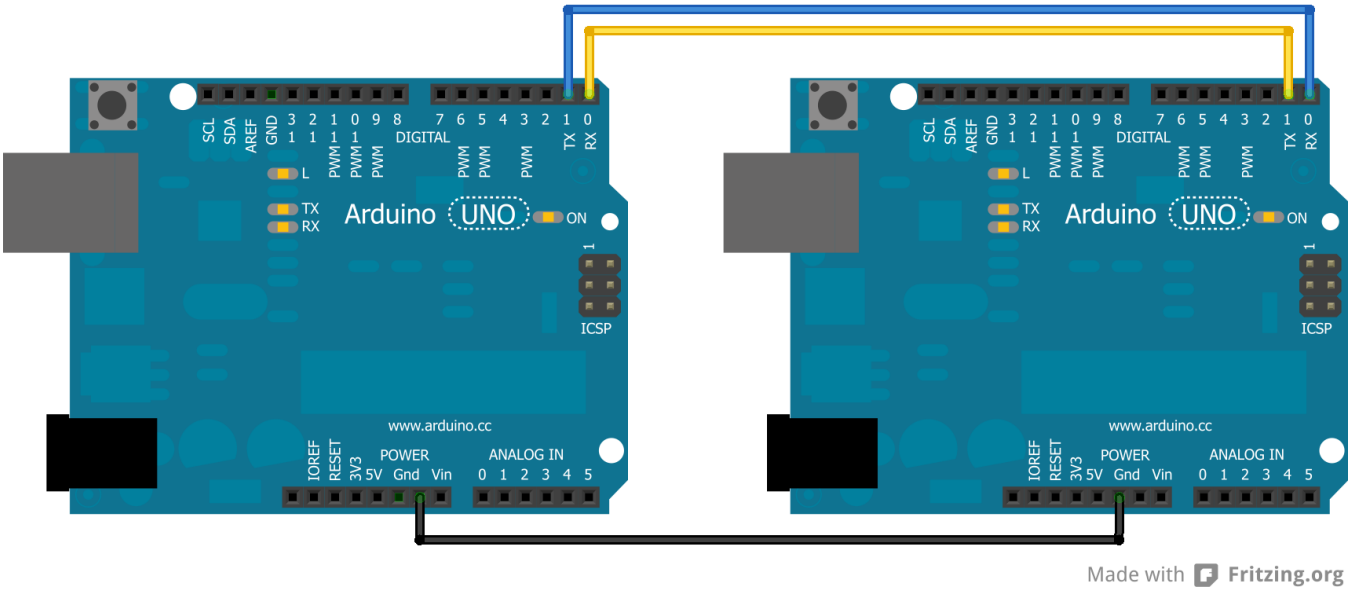
	Tension
Niveau Logique 0	0V
Niveau Logique 1	+5V

Contrairement à ce qu'impose la norme RS-232, les microcontrôleur ne peuvent pas utiliser des tensions négatives. Du coup, ils utilisent les seuls et uniques tensions qu'ils peuvent utiliser, à savoir le 0V et le +5V. Il y a donc quelques petits changement au niveau de la transmission série. Un niveau logique 0 correspond à une tension de 0V et un niveau logique 1 correspond à une tension de +5V. Fort heureusement, comme les microcontrôleurs utilisent quasiment tous cette norme, il n'y a aucun problème à connecter deux microcontrôleurs entre-eux même s'ils ne sont pas de même marque. Cette norme s'appelle alors UART pour Universal Asynchronous Receiver Transmitter plutôt que RS232. Hormis les tensions électriques et le connecteur, c'est la même chose !

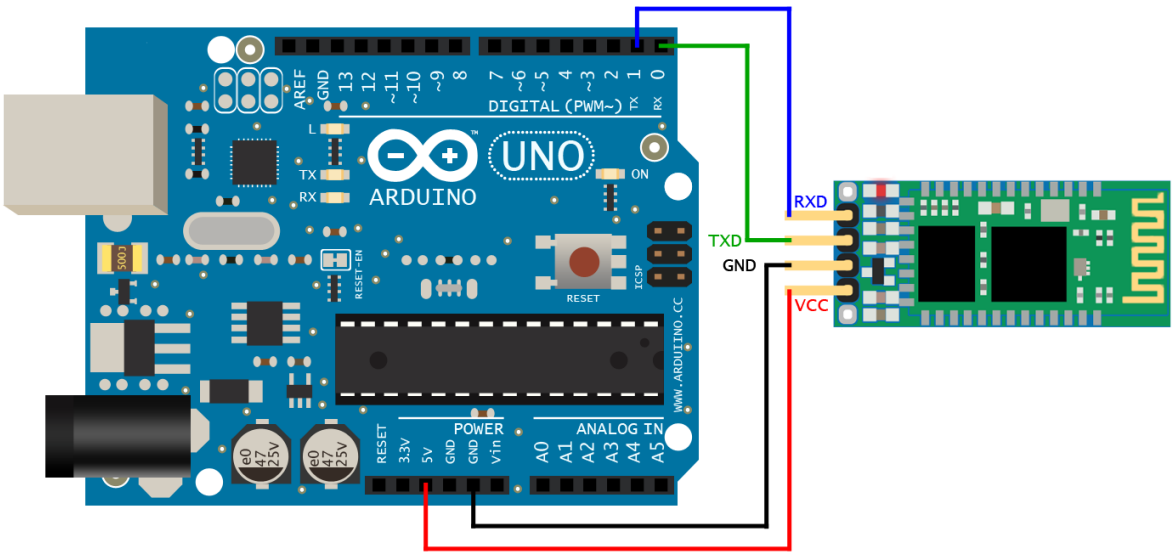
Croisement de données

Il va simplement falloir faire attention à bien croiser les fils. On connecte le Tx (broche de transmission) d'un microcontrôleur au Rx (broche de réception) de l'autre microcontrôleur. Et inversement, le Tx de l'autre au Rx du premier. Et bien sûr, la masse à la masse pour faire une référence commune.

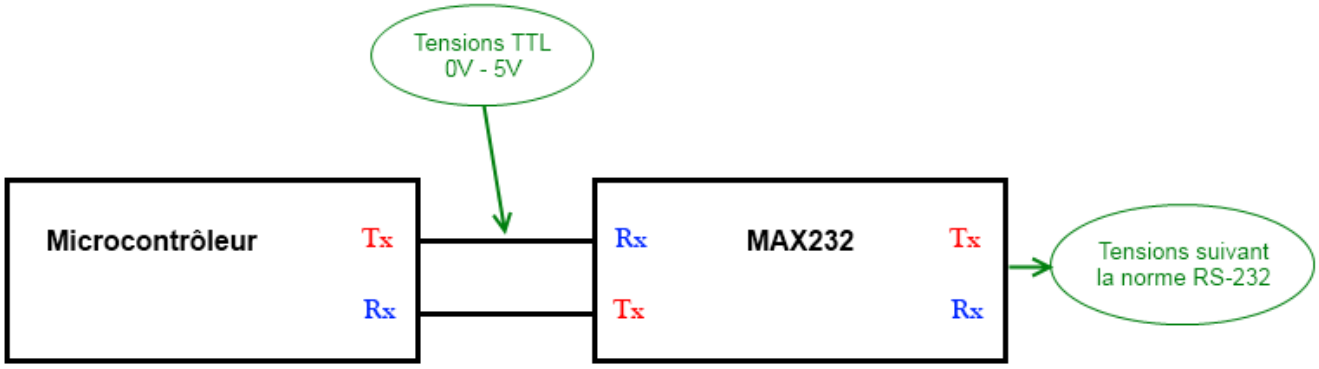
Exemple 1 : entre 2 cartes arduino



Exemple 2 : entre une carte Arduino et un capteur équipé d'une liaison série



Exemple 3 : entre une carte arduino et un ordinateur



Le circuit électronique MAX232 va adapter les tensions de la liaison série de l'arduino à la liaison série (RS232) de l'ordinateur.

Remarque : lorsque l'on utilise le moniteur série du logiciel Arduino, la carte et l'ordinateur dialoguent à l'aide de la liaison série. Ils utilisent alors le port USB.

Remarque N°2 : il existe différentes librairies Arduino permettant de gérer les liaisons série (celle qui est standard s'appelle « serial »).

Remarque N°3 : sur la carte Arduino Uno précédente, on a utilisé les broches 0 et 1 qui sont les broches standard de la liaison série. Si on a besoin de plusieurs liaisons série, certaines autres broches peuvent devenir des broches « série » (Tx et Rx). Il faut consulter la documentation pour vérifier lesquelles sont compatible. Attention, tout est différent avec la carte Arduino Mega.

3. Exemple N°2 : le bus I²C

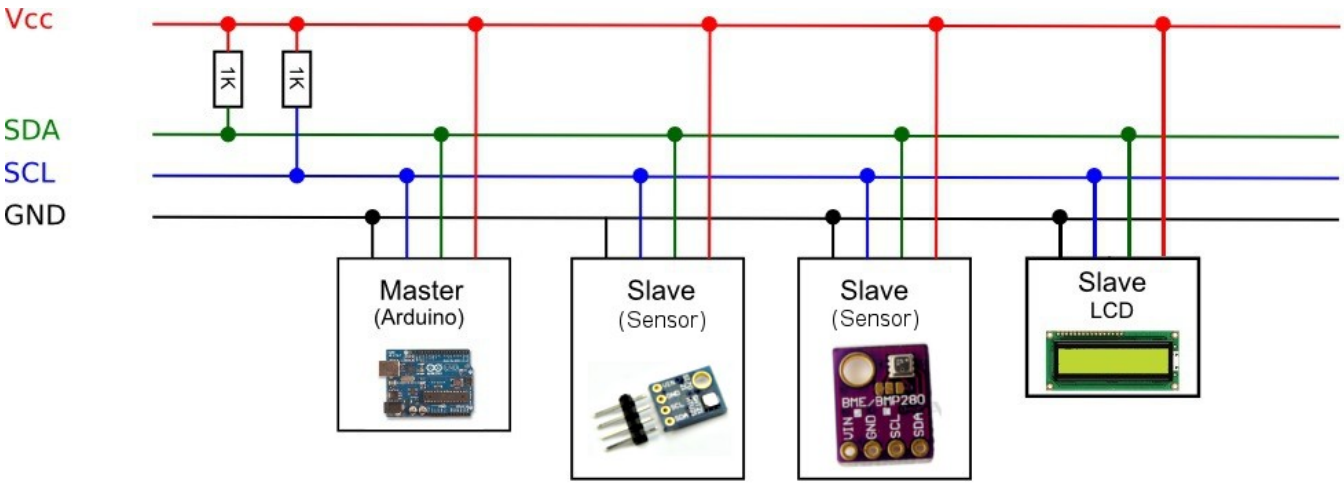
Le bus I²C (I²C signifie : Inter-Integrated Circuit) a été créé au début des années 80 par RTC Philips afin d'apporter une solution simple et peu coûteuse à la communication entre les circuits intégrés numériques à l'intérieur des appareils grand public (téléviseurs, magnétoscopes, jouets, ...). Le principal avantage du bus I²C est de limiter le nombre de liaisons entre circuits intégrés.

Le bus I²C est un bus de type série synchrone ne nécessitant que deux signaux.

- SDA (Serial Data Line), le signal de données bidirectionnelle.
- SCL (Serial Clock Line), le signal d'horloge bidirectionnel.

Ce bus permet la communication entre un circuit maître et un circuit esclave. Le montage peut comporter plusieurs maîtres et plusieurs esclaves. Le maître est le circuit qui émet le signal d'horloge de synchronisation, un seul maître peut envoyer ce signal. Les données peuvent circuler dans les deux sens sur le fil des données, de sorte que chaque circuit, qu'il soit maître ou esclave peut servir d'émetteur ou de récepteur (de données).

Les différents circuits sont placés en parallèle sur les lignes SDA et SCL comme sur le schéma suivant :



Remarques :

- Les 2 lignes SDA et SCL sont tirées au niveau de tension du Vcc (niveau haut) à travers des résistances de pull-up (ici 1KΩ).

- Cartes Arduino : I²C est disponible sur une carte Arduino Uno sur la broche analogique 5 pour SCL qui fournit un signal d'horloge, et la broche analogique 4 pour SDA, qui s'occupe du transfert des données (sur la Mega, il faut utiliser la broche 20 pour SDA et la broche 21 pour SCL). La bibliothèque utilisée est WIRE.
- Raspberry Pi : pour type B, il faut utiliser les broches GPIO2 et GPIO3 pour SDA et SCL repérées sur le « Pi Cobbler ».
- En général il n'y a qu'un seul « maître » mais il est toutefois possible d'en avoir plusieurs.

Nombre d'éléments connectables :
Le nombre maximal d'équipements est limité par le nombre d'adresses disponibles, 7 bits pour l'adresse et un bit pour définir si on écrit ou on lit, soit 128 périphériques, mais il dépend également de la caractéristique physique (capacité) du bus. Il faut savoir que des adresses sont réservées par les fabricants ce qui limite grandement le nombre d'équipements.

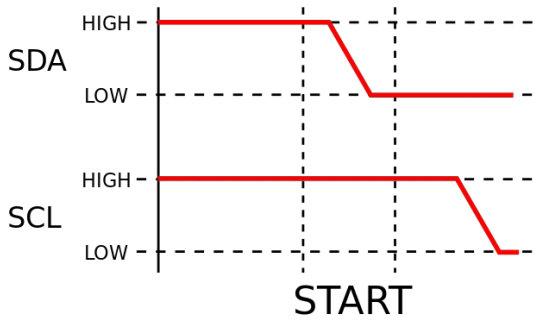
Protocole :

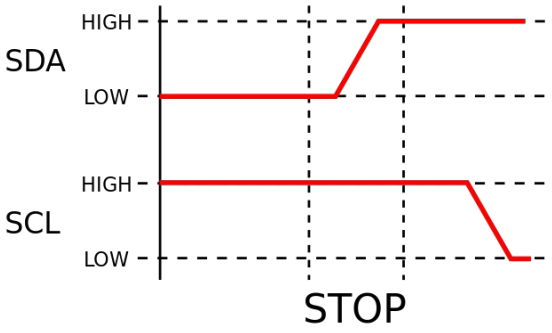
Au repos, c'est à dire lorsque aucun circuit n'émet, les signaux SDA et SCL sont au niveau logique haut. Pour éviter les conflits, un maître qui veut émettre doit attendre que le bus soit au repos.

Principe :
- Quand une ligne (SDA ou SCL) est au repos (niveau 1), on peut la forcer à 0.
- Quand une ligne (SDA ou SCL) est au niveau 0, on ne peut pas la forcer à 1.

Comme la transmission s'effectue sous forme série, une information de début et de fin doit être prévue. L'information de début se nomme START et l'information de fin STOP.

Un maître prend le contrôle du bus en effectuant un START : il met SDA à 0, SCL restant à 1. Au cours de la communication, l'horloge SCL est envoyée par le maître et SDA ne peut changer d'état que lorsque SCL est à 0 :



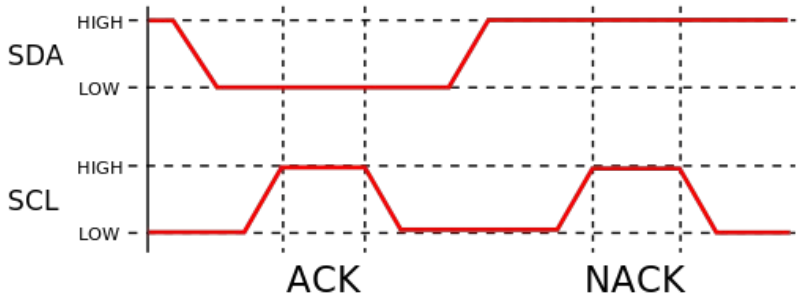


Les données sont envoyées par paquets de huit bits (ou octets). Le bit de poids fort est envoyé le premier, chaque octet est suivi par un bit d'acquittement (ACK) de la part du destinataire.

Le message peut être décomposé en deux parties :

1. Le maître est l'émetteur, l'esclave est le récepteur :
   - émission d'une condition de START par le maître (« S »),
   - émission de l'octet ou des octets d'adresse par le maître pour désigner un esclave, avec le bit $R/\overline{W}$ à 0

- réponse de l'esclave par un bit d'acquittement ACK (ou de non-acquittement NACK),
- après chaque acquittement, l'esclave peut demander une pause (« PA »).
- émission d'un octet de commande par le maître pour l'esclave,
- réponse de l'esclave par un bit d'acquittement ACK (ou de non-acquittement NACK),
- émission d'une condition de RESTART par le maître (« RS »),
- émission de l'octet ou des octets d'adresse par le maître pour désigner le même esclave, avec le bit R/W à 1.

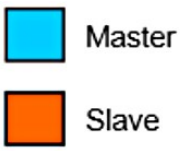


2. Le maître devient récepteur, l'esclave devient émetteur :
   - émission d'un octet de données par l'esclave pour le maître,
   - réponse du maître par un bit d'acquittement ACK (ou de non-acquittement NACK),
   - (émission d'autres octets de données par l'esclave avec acquittement du maître),
   - pour le dernier octet de données attendu par le maître, il répond par un NACK pour mettre fin au dialogue,
   - émission d'une condition de STOP par le maître (« P »).

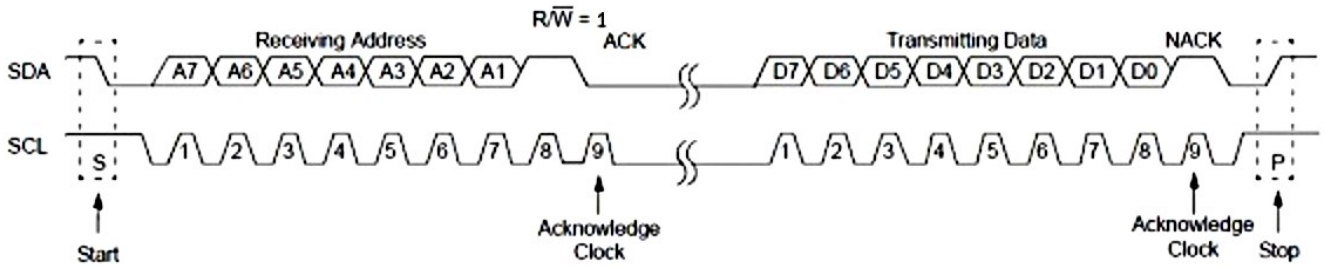
Des exemples de trame I^2C :

Cela donne dans le cas où le maître ne fait que lire ce que l'esclave envoi:

avec :

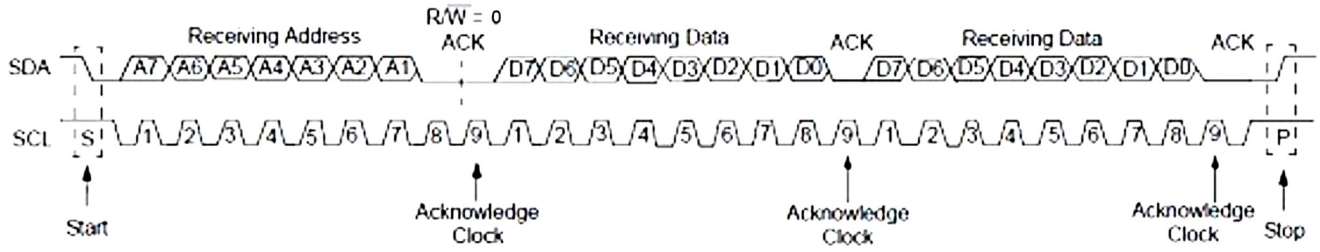


et en signal électrique :



Cela donne dans le cas où le maître écrit à l'esclave envoi:

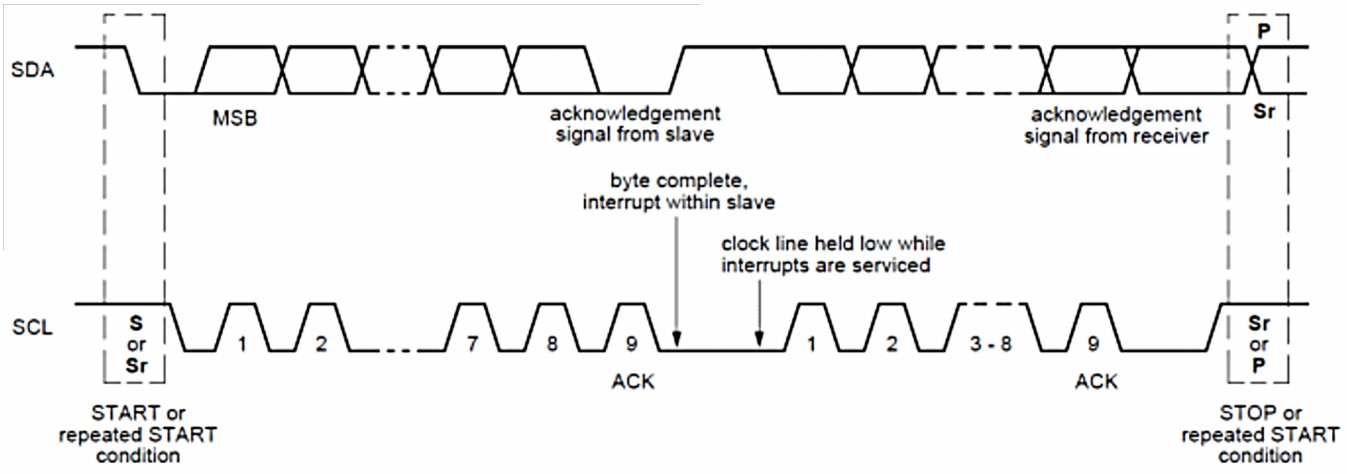
et en signal électrique :



Cela donne dans où le maître écrit (envoie une commande) puis lit ce que l'esclave envoi:

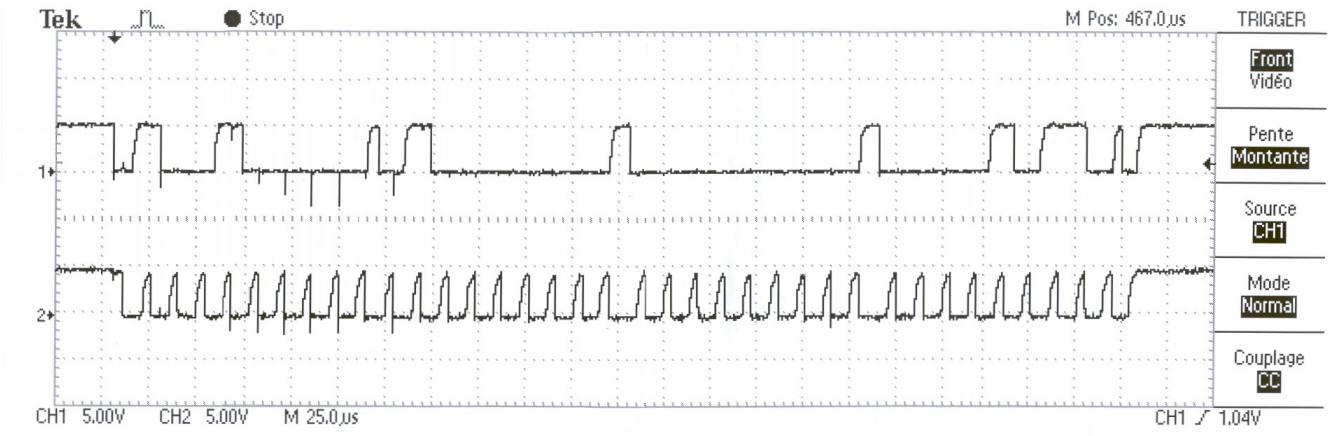
On remarque que pour que le maître envoie une commande il faut mettre le bit R/W à 0 et pour que le maître lise les données envoyées par un esclave il faut mettre le bit R/W à 1

et en signal électrique :



On remarque que SDA est "préparé" avant que SCL soit à l'état haut. Ainsi s'explique le décalage entre le changement d'état de SCL et celle de SDA pendant la transmission.

Exemple d'un relevé réel (oscilloscope) :



Calcul des résistances de Pull-up : R_P

Les temps et les niveaux de tension dépendent de la capacité du bus (CB) et de la valeur des résistances de pull-up (RP). Il est difficile de modifier la valeur de la capacité du bus, mais on peut choisir la valeur des résistances pull-up.

La valeur minimale des résistances de pull-up est limitée par le courant des sorties SDA et SCL (I_{OL}) lorsqu'elles sont à l'état LOW (V_{OL}) :

$$R_{Pmin} = \frac{V_{DD} - V_{OLmax}}{I_{OL}}$$

.

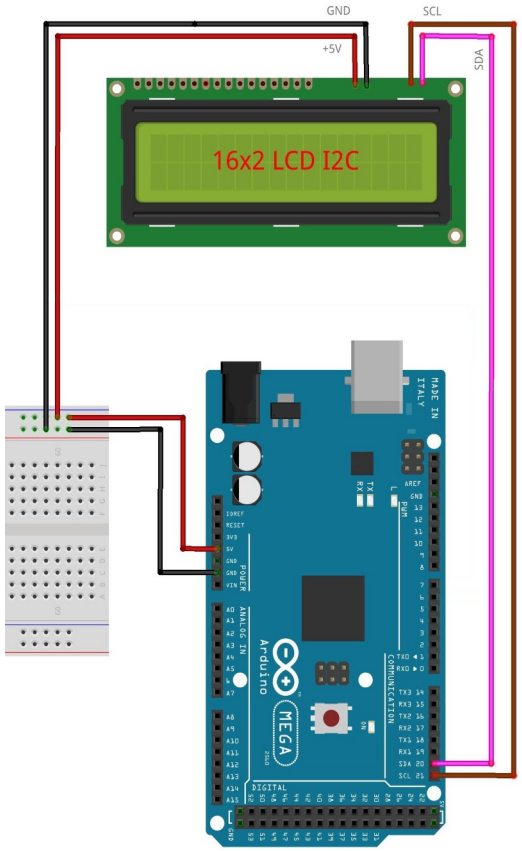
Mode	**V_{OLmax}**	**I_{OL}**	**R_{Pmin} pour V_{DD}=5V**
Standard	0,4V	3mA	1534Ω
Fast	0,6V	6mA	733Ω
Fast plus	0,4V	20mA	230Ω

Les valeurs max (avec t_r, le temps de montée des signaux SDA et SCL):

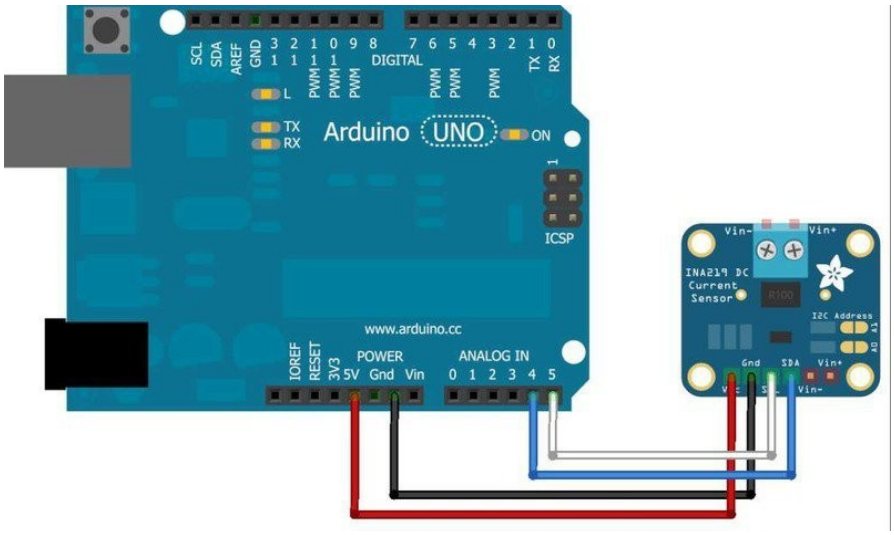
Mode	**t_r**	**C_B**	**R_{Pmax}**
Standard	1µs	400pF	2950Ω
Fast	300ns	400pF	885Ω
Fast plus	120ns	550pF	257Ω

Des exemples avec Arduino :

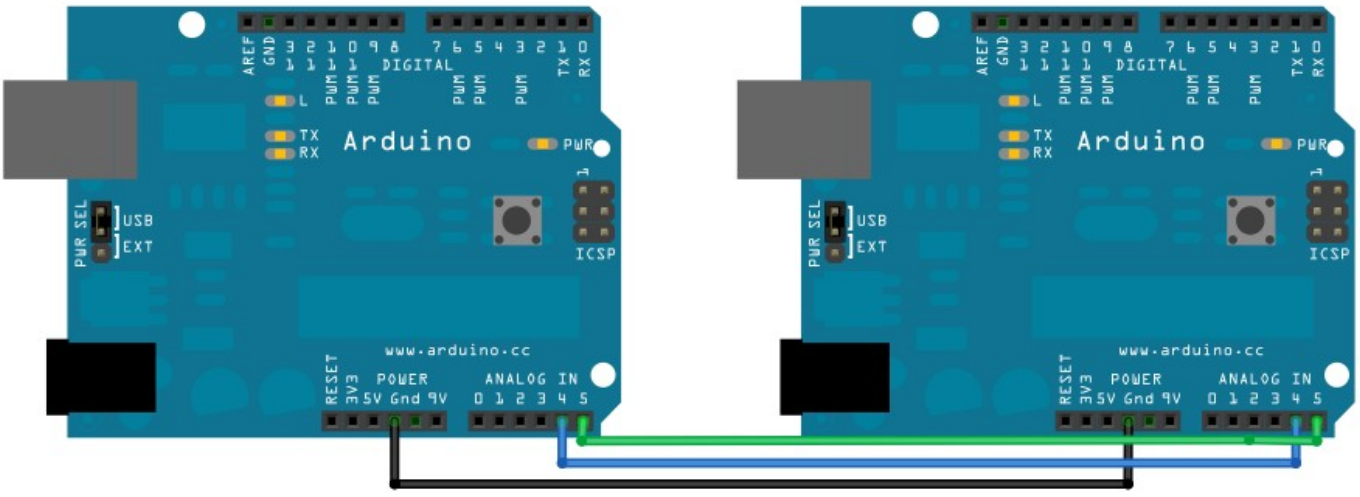
Arduino et un afficheur LCD I²C:



Arduino et un capteur de courant I²C:



Deux Arduino (un maître, un esclave) :

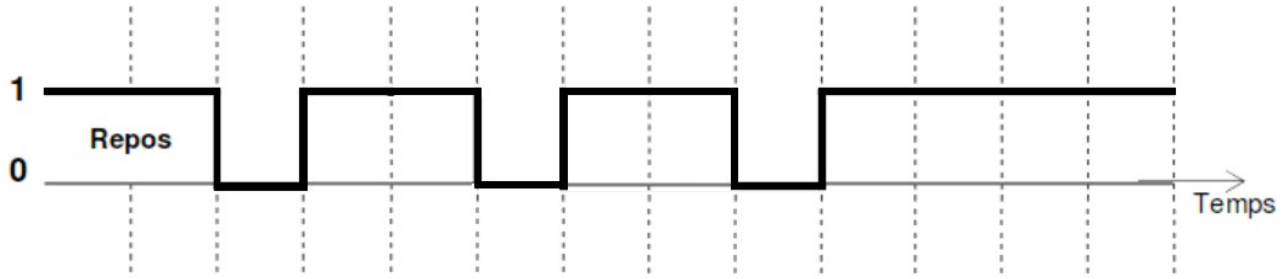


IV. Exercices

Exercice N°7 Analyse d'une trame RS232

La liaison série est paramétrée de la manière suivante :
- Donnée sur 7 bits
- Parité paire
- 2 bits de stop

On relève la trame suivante :

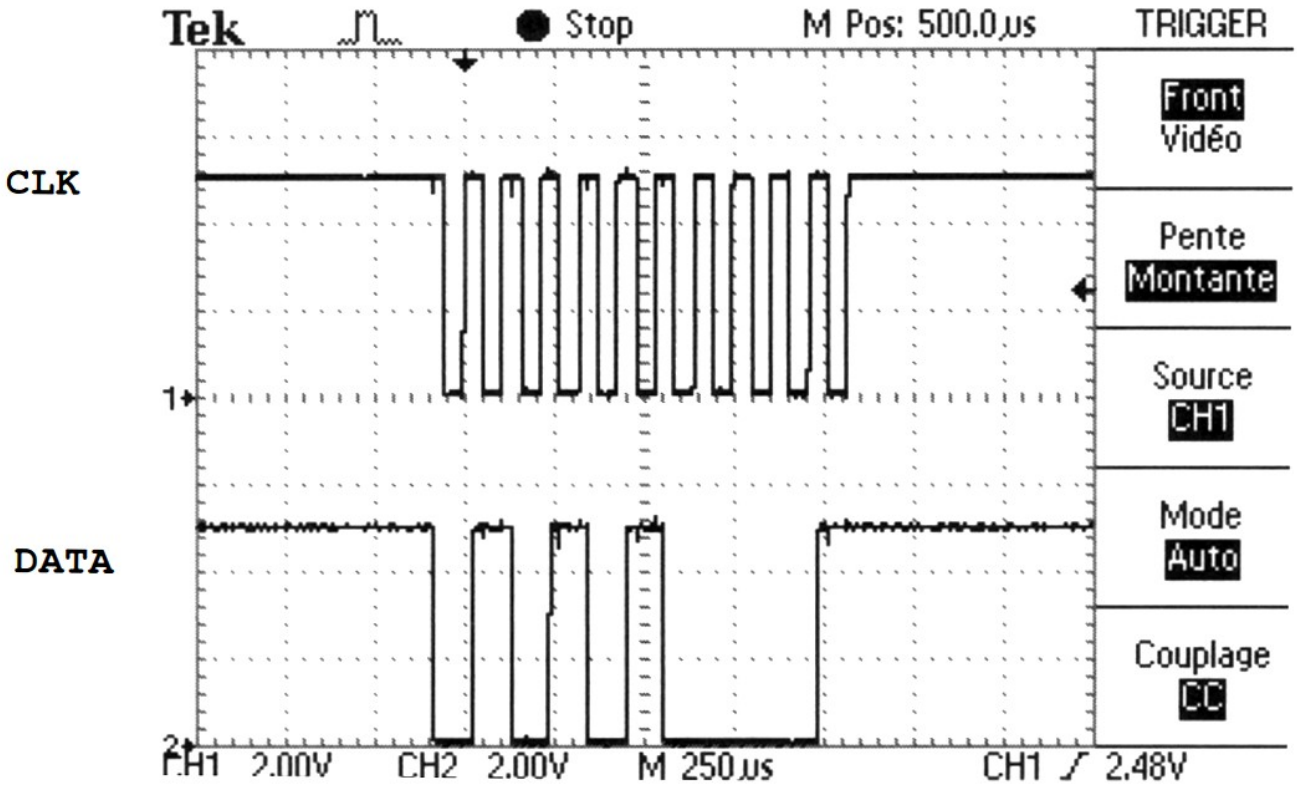


1. ***Repérez sur la trame précédente :***
   - ***le bit de start***
   - ***les bits de donnée***
   - ***le bit de poids faible (LSB)***
   - ***le bit de poids fort (MSB)***
   - ***les bits de stop***
   - ***le bit de parité***
2. ***La donnée transporté est un caractère. Quel est-il ?***
3. ***La parité est-elle bonne ? Justifiez votre réponse.***

Exercice N° 8 Analyse d'une trame RS232

Une liaison série entre un capteur de température et une carte Arduino est paramétrée de la manière suivante :
- Donnée sur 8 bits
- 1 bit de stop

On relève, à l'oscilloscope, la trame suivante (l'horloge semble être active sur le front descendant du signal CLK):



1. ***Repérez sur la trame précédente (entourer sur l'oscillogramme précédent) :***
   - ***le bit de start***
   - ***les bits de donnée***
   - ***le bit de poids faible (LSB)***
   - ***le bit de poids fort (MSB)***
   - ***le bits de stop***
   - ***le bit de parité***
2. ***Quelle est le type de parité utilisée? Justifiez votre réponse.***
3. ***La donnée transportée représente une température (nombre entier). Le bit de poids fort représente le signe de cette température. Quelle est la température mesurée?***

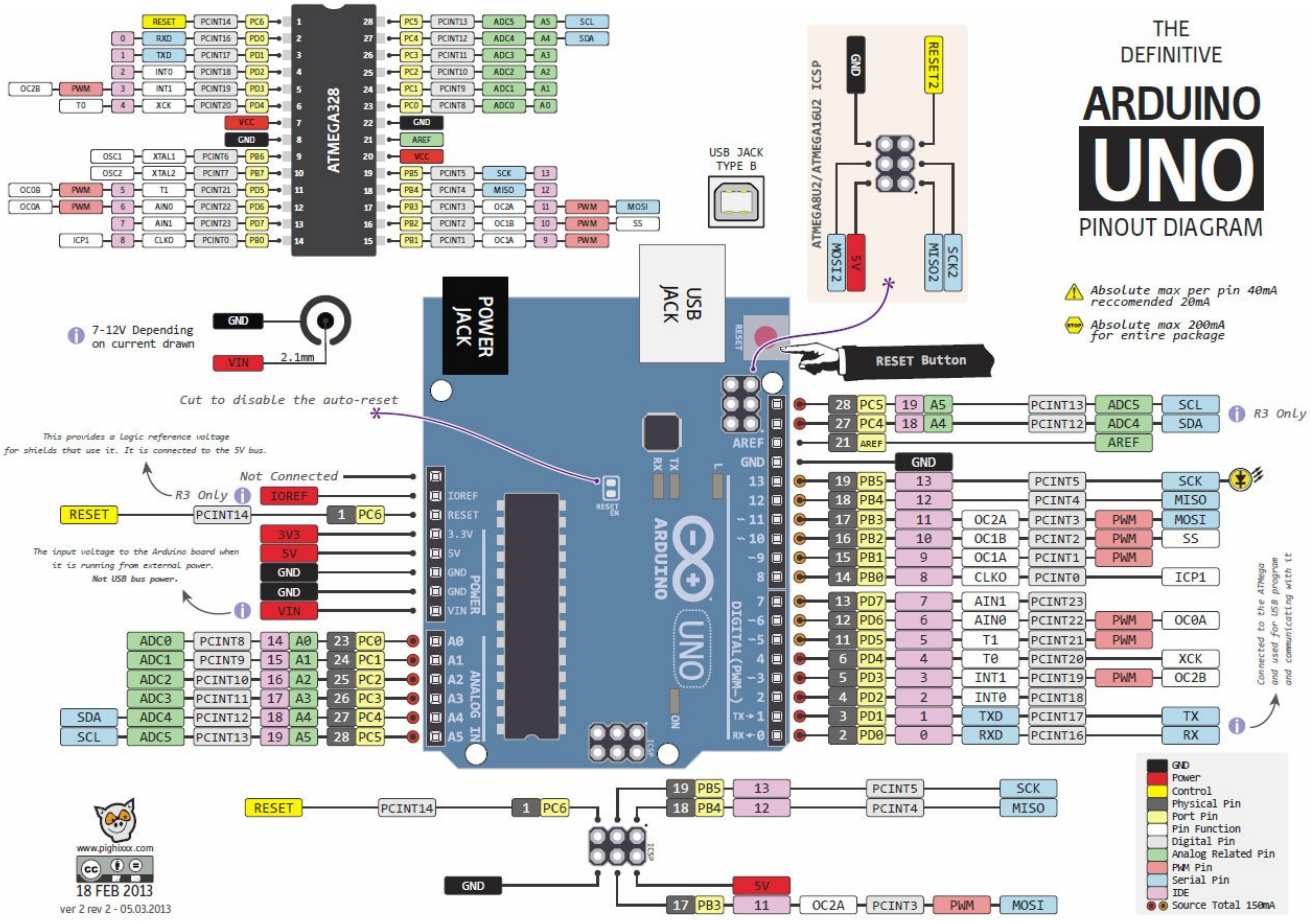
Exercice N°9 Écriture d'une trame RS232

La liaison série est paramétrée de la manière suivante :
- Donnée sur 7 bits
- Parité impaire
- 1 bits de stop

Dessinez la trame si la donnée transportée est le caractère "R"

Exercice N°10 Câblage sur carte Arduino UNO

Description des affectations possibles des broches (pin) d'une carte Arduino uno :



Description des capteurs du système :

TSL2561 Digital Luminosity/Lux/Light Sensor :

Specification :

- Approximates Human eye Response
- Precisely Measures Illuminance in Diverse Lighting Conditions
- Temperature range: -30 to 80 *C
- Dynamic range (Lux): 0.1 to 40,000 Lux
- Voltage range: 2.7-3.6V
- Interface: I2C
- This board/chip uses I2C 7-bit addresses 0x39, 0x29, 0x49, selectable with jumpers

Connect the VCC pin to a 3.3V or 5v power source
Connect GND to the ground pin.
Connect the i2c SCL clock pin to your i2c clock pin.
Connect the i2c SDA data pin to your i2c data pin.
Don't connect the INT pin

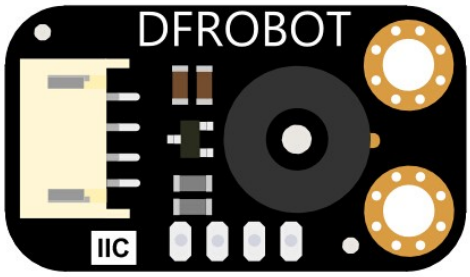
Accéléromètre 3 axes BMA220 SEN0168

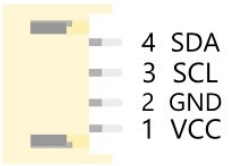
This Triple Axis Accelerometer with Bosch BMA220 is an ultra small triaxial, low-g acceleration sensor breakboard with I2C interface, aiming for lowpower consumer market applications. It allows measurement of accelerations in 3 perpendicular axes and thus senses tilt, motion, shock and vibration in cell phones, handhelds, computer peripherals, man-machine interfaces, virtual reality features and game controllers.

Specification :

- Power supply: 2.0-3.6V
- Interface: I2C
- Acceleration range:±2g/±4g/±8g/±16g
- LED power indication

Capteur de température IR SEN0206





Specification

- Operating Voltage: 3.3V - 5V
- Operating Current: 1.2mA
- Temperature: -70.01℃ to +382.19℃, (0.01 ℃ resolution)
- Interface Type: I2C
- Interface Line Sequence: VCC, GND, SCL, SDA

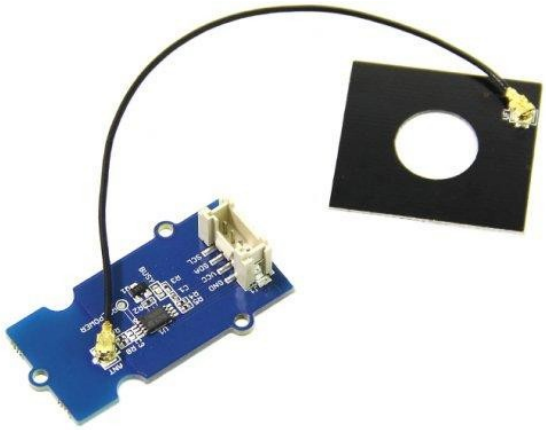
Module Bluetooth Grove 113020008

Grove - Serial Bluetooth is an easy to use module compatible with the existing Grove Base Shield, and designed for transparent wireless serial connection setup. The serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR(Enhanced Data Rate) 2Mbps Modulation with complete 2.4GHz radio transceiver and baseband.

Specifications

- Operating Voltage: 5.0VDC
- Data Rate: 2Mbps
- Fully Qualified Bluetooth V2.0+EDR 3Mbps Modulation
- Selectable baud rate
- Communication : serial port

Module NFC Tag 101020070

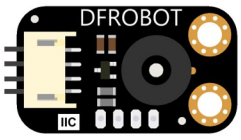


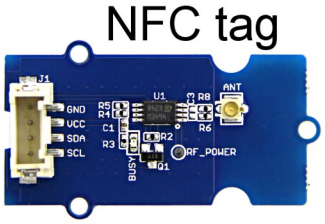
Grove - NFC Tag is a highly integrated Near Field Communication Tag module,this module is I2C interface,which base on M24LR64E-R,M24LR64E-R have a 64-bit unique identifier and 64 -Kbit EEPROM.Grove - NFC Tag attach an independent PCB antenna which can easily stretch out of any enclosure you use, leaving more room for you to design the exterior of your project.

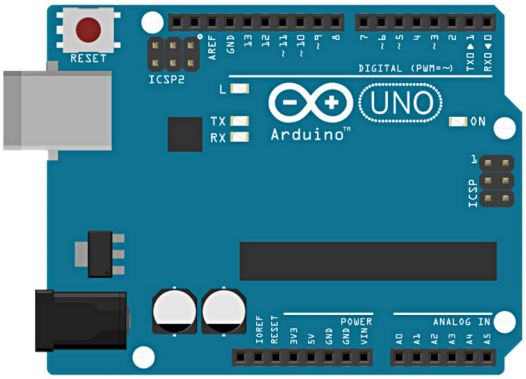
Specifications

- Working Voltage:5V or 3,3V
- Effective range<2cm
- Serve for contactless communication at 13.56MHz
- 64-bit unique identifier (UID)
- Read Block & Write (32-bit blocks)
- Grove I2C Interface

Faites le câblage des capteurs sur la carte Arduino (signaux et alimentations):







bluetooth

Exercice N°11 Trame bus I²C

Nous allons brancher sur une carte Arduino un capteur de température infrarouge avec communication par bus I²C
Caractéristiques du capteur:

- Le capteur fonctionne sur 12 bits
- plage de mesure : -40°C à 85°C
- la mesure est linéaire
- le capteur envoi d'abord les bits de poids faible (B7 à B0) puis ceux de poids plus forts (B11 à B8)

La liaison I²C est classique donc les adresses sont sur 7 bits

On relève la trame I2C suivante :

1. ***Décodage de la trame :***
   - ***Entourez sur la trame le bit de START***
   - ***Relevez l'adresse du capteur. La mettre en hexadécimal***
   - ***Entourez sur la trame le bit de R/W***
   - ***Quel est son état logique et que cela signifie-t-il ?***
   - ***Entourez sur la trame les bits d'acquittement (ACK)***
   - ***Entourez sur la trame les bits de données transmis par le capteur***
   - ***Entourez sur la trame le bits de non-acquittement (NACK)***
   - ***Entourez sur la trame le bit de STOP***

2. ***Analyse des données :***
   - ***Calculez la résolution du capteur***
   - ***Donnez la valeur des 12 bits de mesure que le capteur a envoyé (lus sur la trame)***
   - ***En déduire la température mesurée par le capteur***