

Programmation embarquée



Numérique et Sciences de l'Informatique

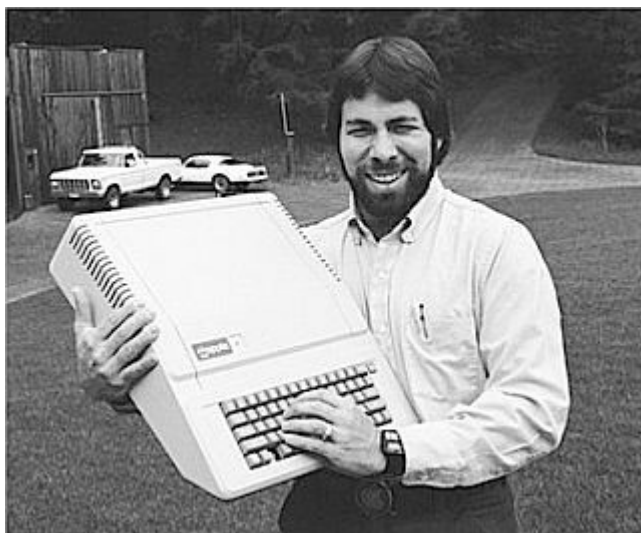
Programme (1/2)

Contenus	Capacités attendues	Commentaires
Modèle d'architecture séquentielle (von Neumann)	Distinguer les rôles et les caractéristiques des différents constituants d'une machine. Dérouter l'exécution d'une séquence d'instructions simples du type langage machine.	La présentation se limite aux concepts généraux. On distingue les architectures monoprocesseur et les architectures multiprocesseur. Des activités débranchées sont proposées. Les circuits combinatoires réalisent des fonctions booléennes.
Transmission de données dans un réseau Protocoles de communication Architecture d'un réseau	Mettre en évidence l'intérêt du découpage des données en paquets et de leur encapsulation. Dérouter le fonctionnement d'un protocole simple de récupération de perte de paquets (bit alterné). Simuler ou mettre en œuvre un réseau.	Le protocole peut être expliqué et simulé en mode débranché. Le lien est fait avec ce qui a été vu en classe de seconde sur le protocole TCP/IP. Le rôle des différents constituants du réseau local de l'établissement est présenté.

Programme (2/2)

Systèmes d'exploitation	<p>Identifier les fonctions d'un système d'exploitation.</p> <p>Utiliser les commandes de base en ligne de commande.</p> <p>Gérer les droits et permissions d'accès aux fichiers.</p>	<p>Les différences entre systèmes d'exploitation libres et propriétaires sont évoquées.</p> <p>Les élèves utilisent un système d'exploitation libre.</p> <p>Il ne s'agit pas d'une étude théorique des systèmes d'exploitation.</p>
Périphériques d'entrée et de sortie Interface Homme-Machine (IHM)	<p>Identifier le rôle des capteurs et actionneurs.</p> <p>Réaliser par programmation une IHM répondant à un cahier des charges donné.</p>	<p>Les activités peuvent être développées sur des objets connectés, des systèmes embarqués ou robots.</p>

Architecture



INSTRUCTION SET SUMMARY

INSTRUC	OPERATION	OP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
ADD	A ← M + C	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							

- Exemple de l'**Apple II**, un des premiers ordinateurs personnels :
 - Processeur MOS 6502 8 bit à 1Mhz
 - RAM : 4ko
 - ROM : 8ko

INSTRUCTIONS													ADDRESSSTATUS																																																																																																																																																																																																																																																																																																																																																																													
MANUWRITE	OPERATION	OP-1	OP-2	OP-3	OP-4	OP-5	OP-6	OP-7	OP-8	OP-9	OP-10	OP-11	OP-12	OP-13	OP-14	OP-15	OP-16	OP-17	OP-18	OP-19	OP-20	OP-21	OP-22	OP-23	OP-24	OP-25	OP-26	OP-27	OP-28	OP-29	OP-30	OP-31	OP-32	OP-33	OP-34	OP-35	OP-36	OP-37	OP-38	OP-39	OP-40	OP-41	OP-42	OP-43	OP-44	OP-45	OP-46	OP-47	OP-48	OP-49	OP-50	OP-51	OP-52	OP-53	OP-54	OP-55	OP-56	OP-57	OP-58	OP-59	OP-60	OP-61	OP-62	OP-63	OP-64	OP-65	OP-66	OP-67	OP-68	OP-69	OP-70	OP-71	OP-72	OP-73	OP-74	OP-75	OP-76	OP-77	OP-78	OP-79	OP-80	OP-81	OP-82	OP-83	OP-84	OP-85	OP-86	OP-87	OP-88	OP-89	OP-90	OP-91	OP-92	OP-93	OP-94	OP-95	OP-96	OP-97	OP-98	OP-99	OP-100	OP-101	OP-102	OP-103	OP-104	OP-105	OP-106	OP-107	OP-108	OP-109	OP-110	OP-111	OP-112	OP-113	OP-114	OP-115	OP-116	OP-117	OP-118	OP-119	OP-120	OP-121	OP-122	OP-123	OP-124	OP-125	OP-126	OP-127	OP-128	OP-129	OP-130	OP-131	OP-132	OP-133	OP-134	OP-135	OP-136	OP-137	OP-138	OP-139	OP-140	OP-141	OP-142	OP-143	OP-144	OP-145	OP-146	OP-147	OP-148	OP-149	OP-150	OP-151	OP-152	OP-153	OP-154	OP-155	OP-156	OP-157	OP-158	OP-159	OP-160	OP-161	OP-162	OP-163	OP-164	OP-165	OP-166	OP-167	OP-168	OP-169	OP-170	OP-171	OP-172	OP-173	OP-174	OP-175	OP-176	OP-177	OP-178	OP-179	OP-180	OP-181	OP-182	OP-183	OP-184	OP-185	OP-186	OP-187	OP-188	OP-189	OP-190	OP-191	OP-192	OP-193	OP-194	OP-195	OP-196	OP-197	OP-198	OP-199	OP-200	OP-201	OP-202	OP-203	OP-204	OP-205	OP-206	OP-207	OP-208	OP-209	OP-210	OP-211	OP-212	OP-213	OP-214	OP-215	OP-216	OP-217	OP-218	OP-219	OP-220	OP-221	OP-222	OP-223	OP-224	OP-225	OP-226	OP-227	OP-228	OP-229	OP-230	OP-231	OP-232	OP-233	OP-234	OP-235	OP-236	OP-237	OP-238	OP-239	OP-240	OP-241	OP-242	OP-243	OP-244	OP-245	OP-246	OP-247	OP-248	OP-249	OP-250	OP-251	OP-252	OP-253	OP-254	OP-255	OP-256	OP-257	OP-258	OP-259	OP-260	OP-261	OP-262	OP-263	OP-264	OP-265	OP-266	OP-267	OP-268	OP-269	OP-270	OP-271	OP-272	OP-273	OP-274	OP-275	OP-276	OP-277	OP-278	OP-279	OP-280	OP-281	OP-282	OP-283	OP-284	OP-285	OP-286	OP-287	OP-288	OP-289	OP-290	OP-291	OP-292	OP-293	OP-294	OP-295	OP-296	OP-297	OP-298	OP-299	OP-300	OP-301	OP-302	OP-303	OP-304	OP-305	OP-306	OP-307	OP-308	OP-309	OP-310	OP-311	OP-312	OP-313	OP-314	OP-315	OP-316	OP-317	OP-318	OP-319	OP-320	OP-321	OP-322	OP-323	OP-324	OP-325	OP-326	OP-327	OP-328	OP-329	OP-330	OP-331	OP-332	OP-333	OP-334	OP-335	OP-336	OP-337	OP-338	OP-339	OP-340	OP-341	OP-342	OP-343	OP-344	OP-345	OP-346	OP-347	OP-348	OP-349	OP-350	OP-351	OP-352	OP-353	OP-354	OP-355	OP-356	OP-357	OP-358	OP-359	OP-360	OP-361	OP-362	OP-363	OP-364	OP-365	OP-366	OP-367	OP-368	OP-369	OP-370	OP-371	OP-372	OP-373	OP-374	OP-375	OP-376	OP-377</

1979 vs 2019

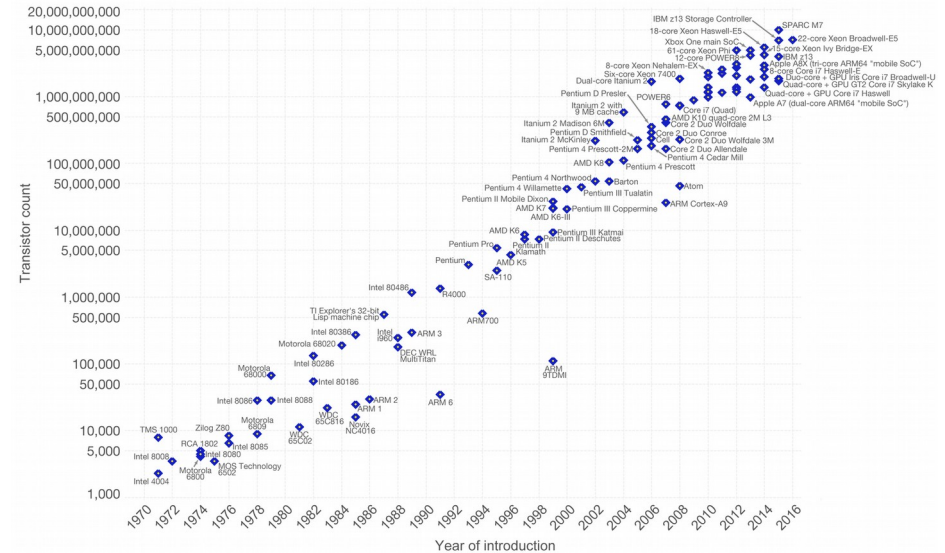
Processeur	8bit 1Mhz	64bit 2.9Ghz
RAM	4Kio	16Gio
ROM	8Ko	1To
Coeurs	1	8
Affichage	6 couleurs 280 x 192	16 millions de couleurs 1920 x 1080
Réseau	-	Ethernet GigaBit

Loi de Moore

« The complexity for minimum component costs has increased at a rate of roughly a factor of two per year »

→ Ne présage évidemment pas de l'amélioration de l'expérience utilisateur

Moore's Law – The number of transistors on integrated circuit chips (1971-2016) 
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at [OurWorldinData.org](https://www.ourworldindata.org). There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

Loi de Wirth / May

Cette croissance à créé un fossé entre les développeurs et le matériel

« Software is getting slower more rapidly than hardware becomes faster. »

« Software efficiency halves every 18 months, compensating Moore's law. »



Le logiciel pas toujours à la hauteur ?



Tom Eastman

@tveastman

Suivre



I have a Python program I run every day, it takes 1.5 seconds.

I spent six hours re-writing it in rust, now it takes 0.06 seconds.

That efficiency improvement means I'll make my time back in 41 years, 24 days :-)

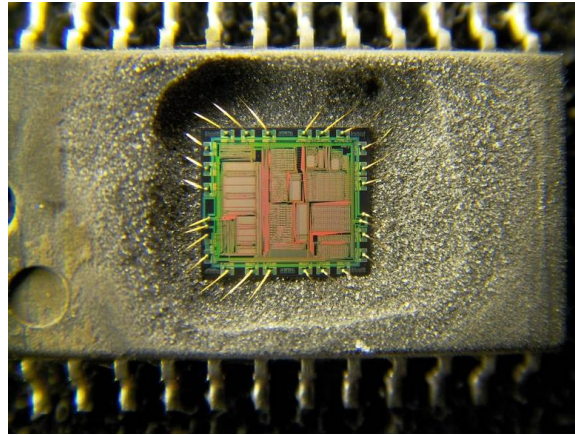
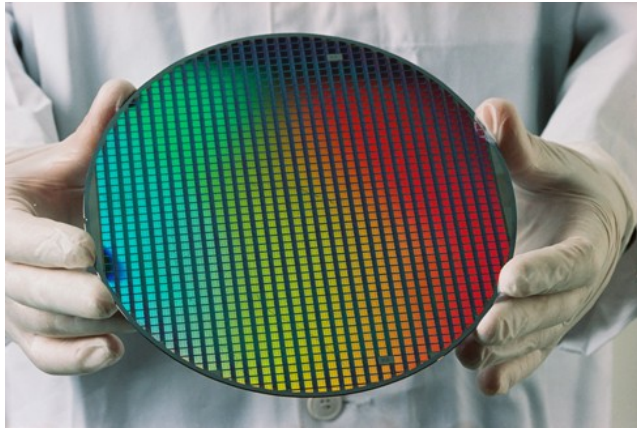
 Traduire le Tweet

21:07 - 9 sept. 2018

Le logiciel pas toujours à la hauteur ?

- **Logiciel lent**
 - Windows 10 met **30 minutes** à se mettre à jour ...
 - ... dans le même temps, on aurait réécrit **5 fois le SSD entier**
- **Logiciel énorme**
 - Windows 95 pesait **30 Mo** ...
 - ... l'application clavier de Google (Android) consomme 150 Mo
- **Logiciel vite obsolète**

Du silicium au logiciel



Les types de circuits intégrés

- Sur un circuit, on peut retrouver :
 - Les processeurs (CPU)
 - Les microcontrôleurs (MCU)
 - Les ASIC/ASSP
 - Les DSP
 - Les FPGA

Cartes de développement



PC Portable

Windows

3Ghz

50W

~800€



Raspberry Pi

Linux

800Mhz

5W

~50€



Arduino Uno

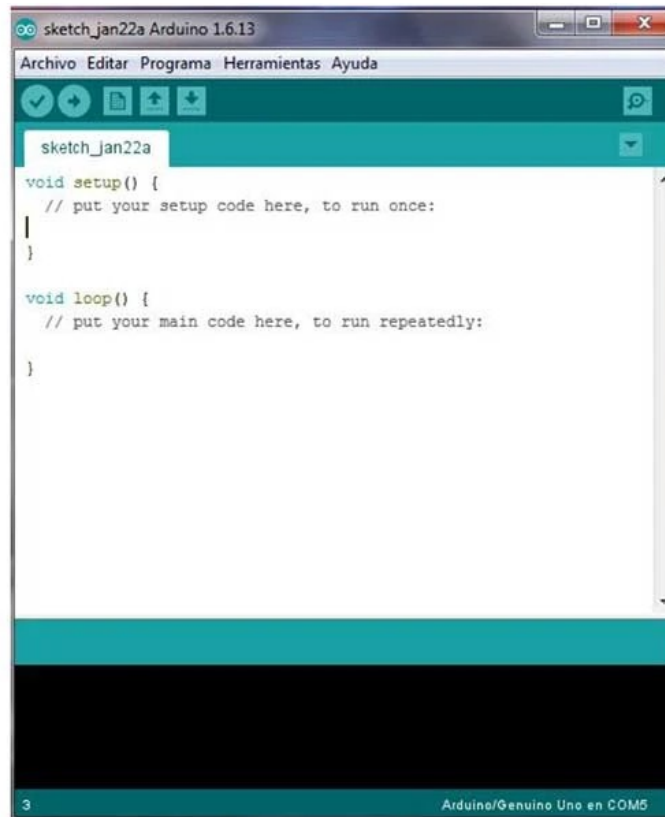
Aucun OS

16Mhz

0.2W

~5€

L'outil Arduino



Pas de système d'exploitation

- C'est le cas par exemple des cartes Arduino
- D'après vous, quels sont les avantages ? Les inconvénients ?

Pas de système d'exploitation

- On s'épargne le surcoût mémoire d'un système (raison de l'absence sur Arduino)
 - **Pas de *préemption*, avantage pour mieux maîtriser ce qui se passe, mais très peu pratique pour développer**

Pas de système d'exploitation

- On s'épargne le surcoût mémoire d'un système (raison de l'absence sur Arduino)
 - **Pas de *préemption*, avantage pour mieux maîtriser ce qui se passe, mais très peu pratique pour développer**

Comment ça marche ?



Intel core i7
x86 64 bit



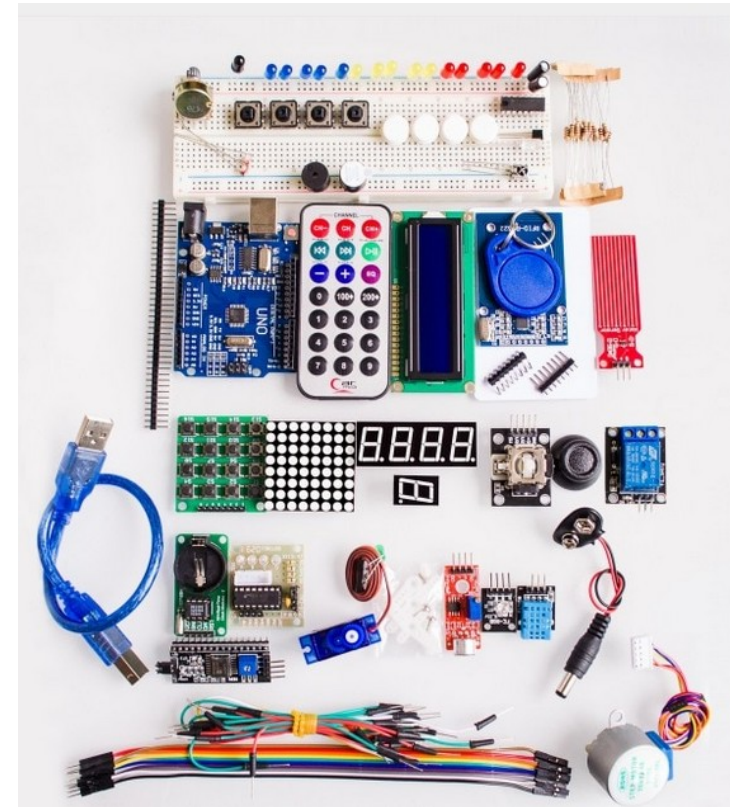
ATmega328P
AVR 8 bit

Comment ça marche ?

- L'ordinateur compile le programme pour la carte (différente architecture → on parle de cross-compilation)
- Le programme est chargé à bord à l'aide d'un *bootloader* (programme d'amorçage)
- **Contrainte :**
 - *l'interpréteur Python est trop lourd pour Arduino, nous programmons donc en C/C++ !*

Capteurs et actuateurs

- D'un point de vue pratique, on trouve dans le commerce des kits et des capteurs/actuateurs compatibles Arduino



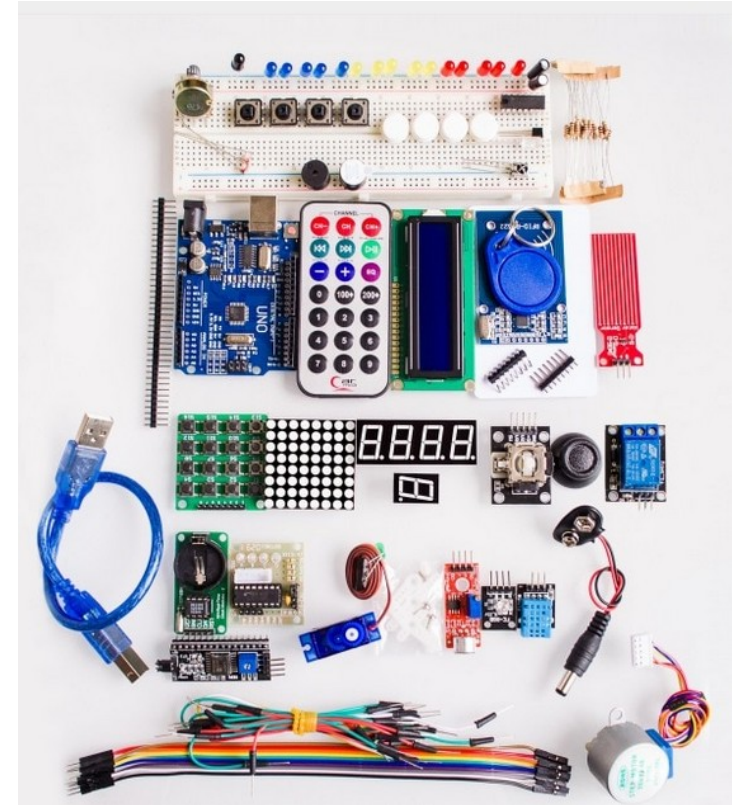
Capteurs et actuateurs

- **Capteurs**

- Température, luminosité, distance, pression, accélération, vitesse de rotation, magnétique (boussole), poids, ...

- **Actuateur**

- Moteurs, servomoteurs, moteurs à pas, moteurs linéaires, électro-aimants ...



Programmer dans Arduino IDE

- Il est possible d'utiliser l'interface série pour afficher des messages :

```
void setup()
{
    // Initialisation de la communication série à
    // 115200bauds
    Serial.begin(115200);
}

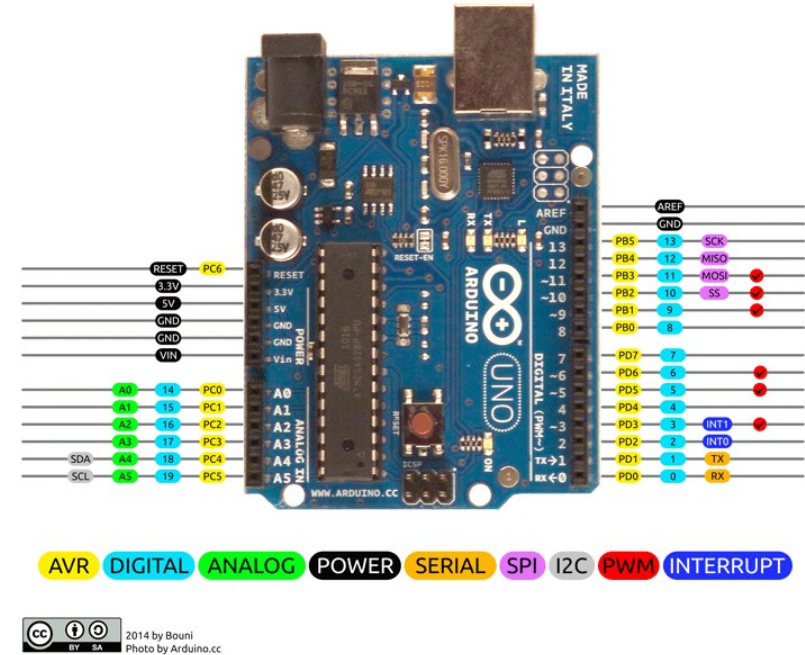
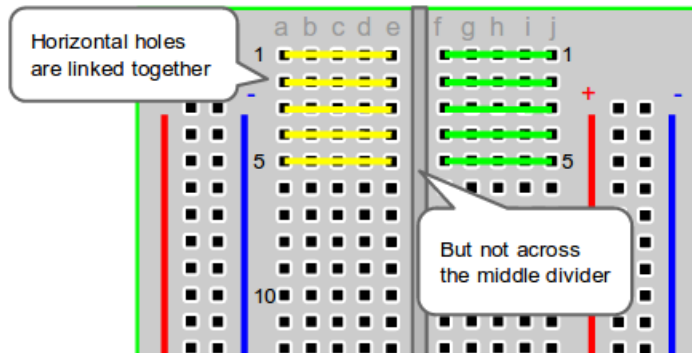
void loop()
{
    // Affichage d'un message
    Serial.println("Bonjour!");
    // On attend 1s
    delay(1000);
}
```

Programmer dans Arduino IDE

- Nous vous proposons un exemple de code commenté ici, qui contient des commentaires, variables, boucles etc :
- [\[exemple.ino\]](#)

Broches et branchements

- Les cartes de développement sont équipées de broches (*pin* en anglais)
- On trouve aussi des platines « *breadboard* », qui permettent des branchements sans soudure :



Manipulation 1

Entrées/sorties analogiques et digitales

Entrées/sorties générales

- Toutes les broches peuvent être utilisées en entrée/sortie **digitale**
- On appelle cela **GPIO**
 - *General Purpose Input/Output*
- Le niveau « 0 » (bas) et « 1 » (haut) correspondent à un niveau voltage (par exemple 0V et 5V)

Entrées/sorties générales

- Dans Arduino, il faut paramétrer la broche (en sortie *OUTPUT* ou en entrée *INPUT*) avec la fonction *pinMode*, puis on peut contrôler son niveau logique à l'aide de *digitalWrite* :

```
// La broche 3 est en sortie
pinMode(3, OUTPUT);
// On écrit le niveau "haut" (5V)
digitalWrite(3, HIGH);
```

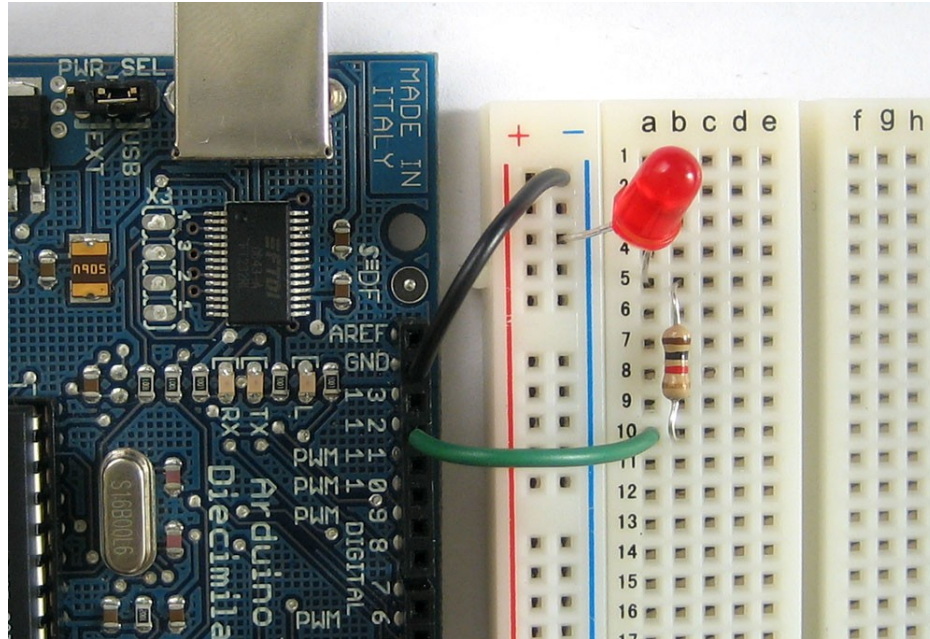
Entrées/sorties générales

- De la même manière, on peut mettre une broche en entrée et la lire à l'aide de *digitalRead* :

```
// La broche 3 est en entrée
pinMode(3, INPUT);
// Si on lit 5V, on affiche un message
if (digitalRead(3) == HIGH) {
    Serial.println("Niveau haut!")
}
```

Branchement d'une LED

- Voici à quoi ressemble le branchement d'une LED et de sa résistance:

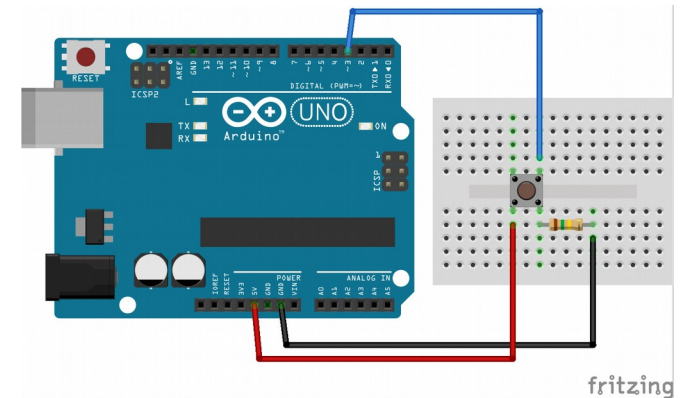


Branchement d'un bouton

- Pour lire un bouton, on peut utiliser un branchement similaire à l'image ci-contre.
- La résistance ici permet de s'assurer que l'on lit un « 0 » lorsque le bouton n'est pas appuyé (résistance de tirage)

Note : une autre solution est d'activer la résistance de tirage interne :

```
pinMode(3, INPUT);  
digitalWrite(3, HIGH);
```



Entrées/sorties analogiques

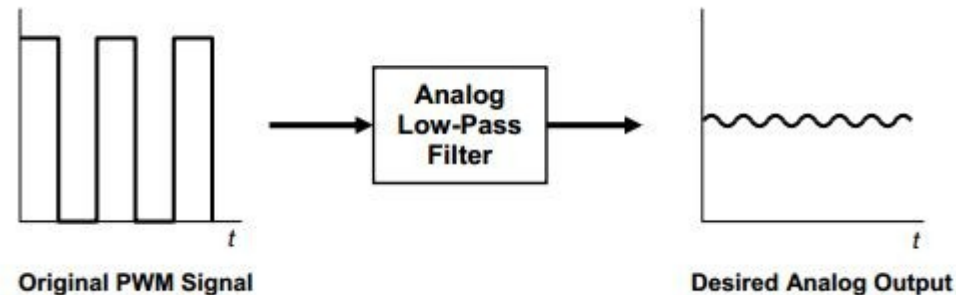
- Les entrées analogiques permettent de mesurer une valeur non-binaire, par exemple entre 0V et 5V. On appelle cela l'échantillonnage par un convertisseur analogique/numérique
- Sur Arduino, cette valeur est sur 10 bits (de 0 à 1023) :

```
// La broche "A1" est configurée en entrée
pinMode(A1, INPUT);

if (analogRead(A1) > 512) {
    Serial.println("Le voltage est plus haut que 2.5V!");
}
```

Entrées/sorties analogiques

- Dans l'autre sens, il n'est pas possible de produire une valeur analogique à proprement parler, car la carte n'est pas équipée de convertisseur numérique/analogique
- Cependant, il est possible de produire un signal carré, dont on règle le *rapport cyclique*. On nomme ce principe **PWM** (pulse width modulation):



Entrées/sorties analogiques

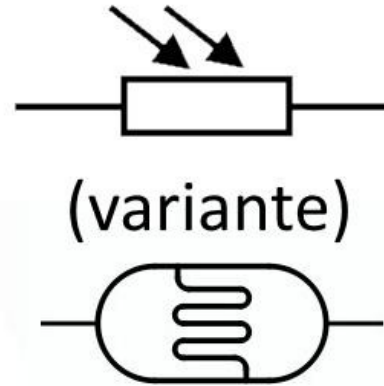
- Arduino permet de faire cela à l'aide de *analogWrite* :

```
// La broche D9 est configurée en sortie  
pinMode(9, OUTPUT);
```

```
// On règle le rapport cyclique à 200 sur 255  
analogWrite(9, 200);
```

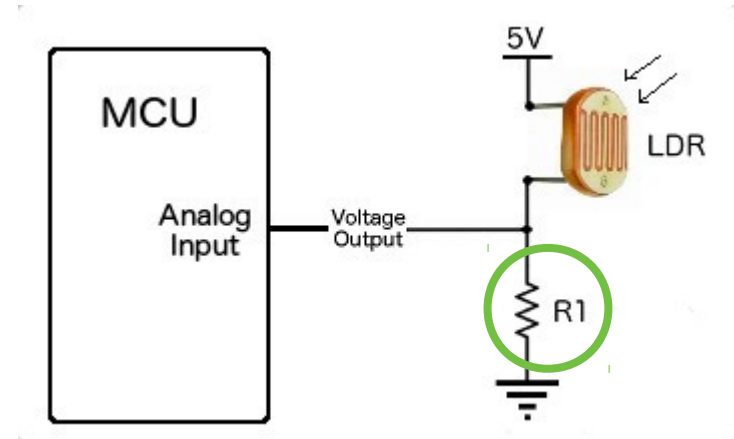
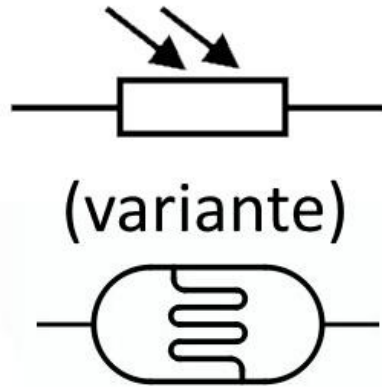
Échantillonnage analogique d'une résistance

- Certains capteurs sont en fait des résistances qui varient selon la température ou la luminosité. Comment les échantillonner ?



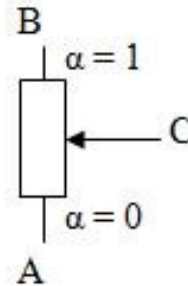
Échantillonnage analogique d'une résistance

- On utilise simplement un pont diviseur de tension à l'aide d'une résistance fixe que l'on ajoute :

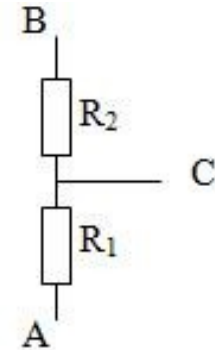


Échantillonnage d'un potentiomètre

- Le potentiomètre est un type de capteur qui fonctionne en rotation et qui équivaut au circuit suivant :

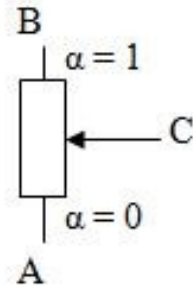


équivalent à

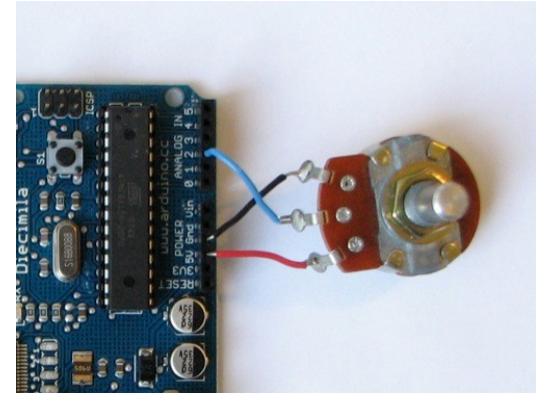
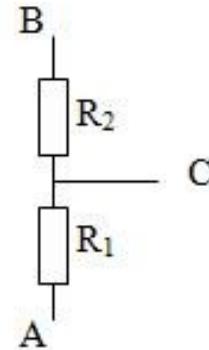


Échantillonnage d'un potentiomètre

- Pour l'échantillonner, on applique donc un voltage aux extrémités (par ex. 5V) et on mesure le voltage au point central, qui variera de 0V à 5V :

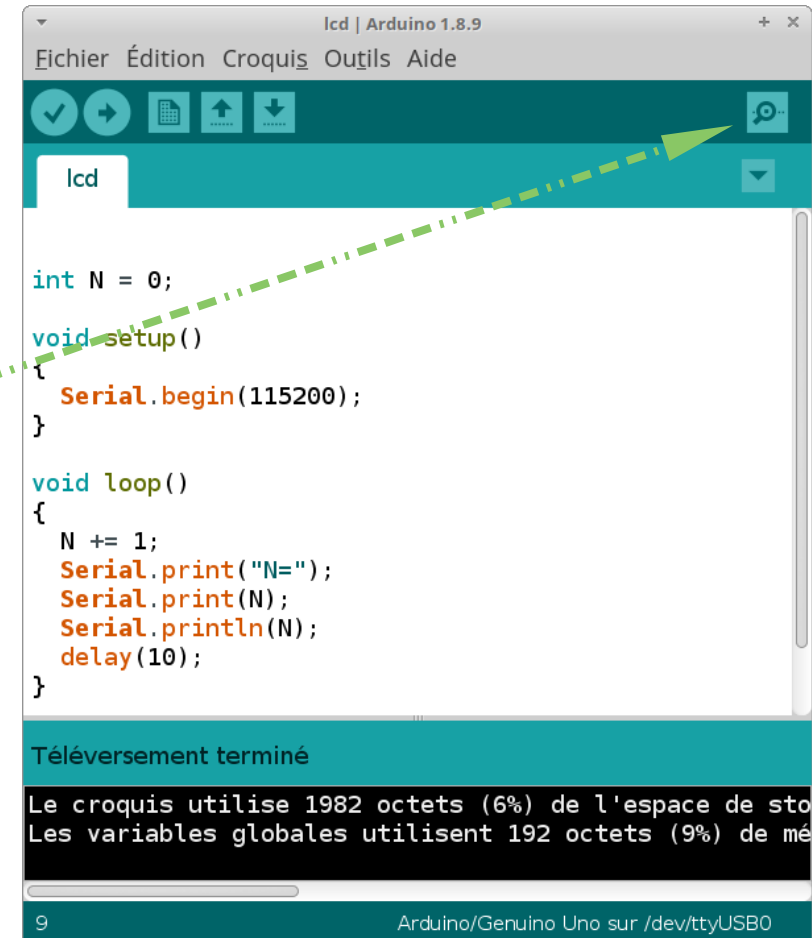


équivalent à



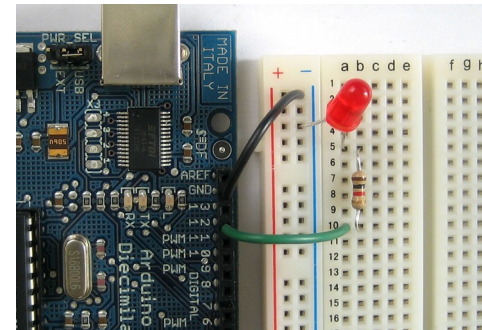
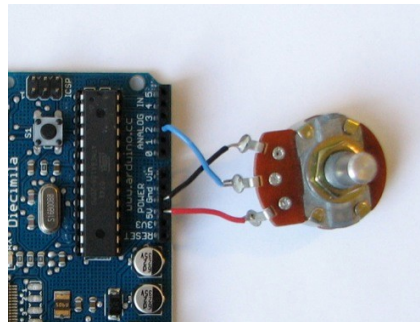
Moniteur série

- La carte peut communiquer pendant qu'elle tourne à l'aide du moniteur série
- Pour cela, il faut utiliser la même fréquence (*baudrate*) de communication
- [\[serial-print.ino\]](#)



A vous de jouer !

- Affichez dans le moniteur série les valeurs du potentiomètre
- A l'aide de *analogWrite*, faites varier l'intensité de la LED en fonction du potentiomètre !
 - Attention : le potentiomètre est échantillonné entre 0 et 1023 (10 bits) alors que *analogWrite* attend une valeur entre 0 et 255 (8 bits)

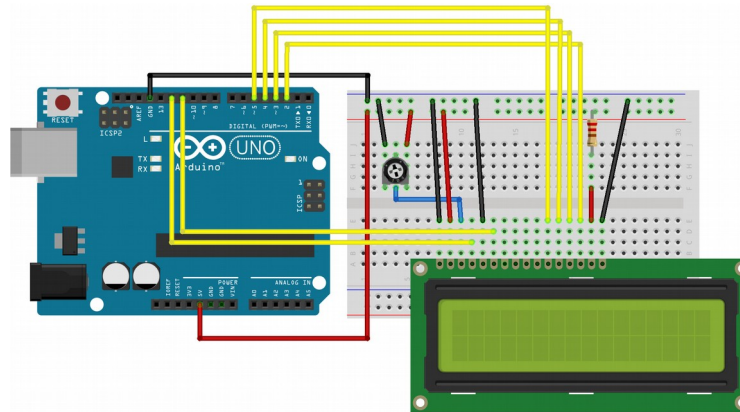


Manipulation 2

Pilotage de moteurs

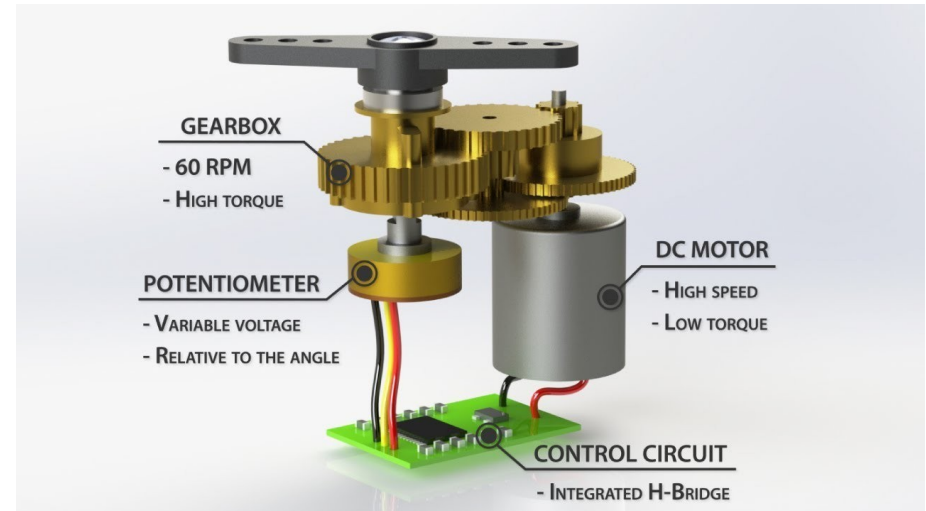
Bus de communication

- La carte peut également dialoguer avec des capteurs via des bus de communication (UART, I²C, SPI, ...)
- Il existe de nombreuses bibliothèques, et la possibilité d'en télécharger des supplémentaires sur internet !



Servomoteurs

- Un servomoteur est un actuateur équipé :
 - D'un moteur DC
 - D'un étage de réduction (engrenages / *gearbox*)
 - D'un potentiomètre



Servomoteurs

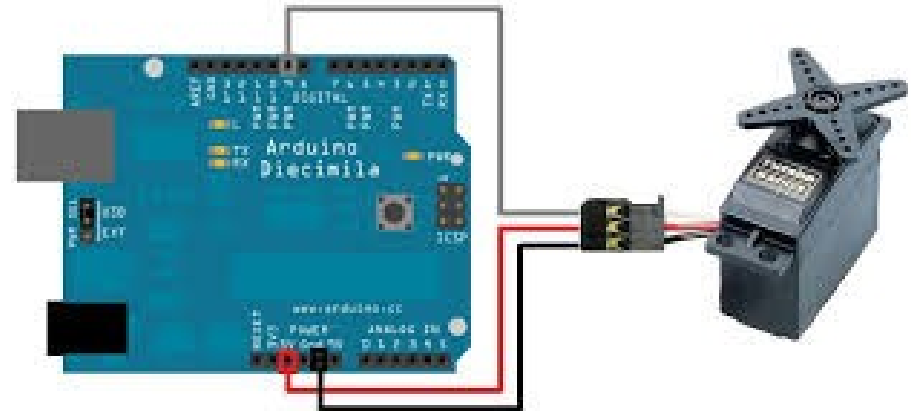
- La connexion se fait simplement (2 broches d'alimentation + 1 broche de signal), et la communication peut se faire via la bibliothèque « Servo.h » :

```
#include <Servo.h>

Servo servo;

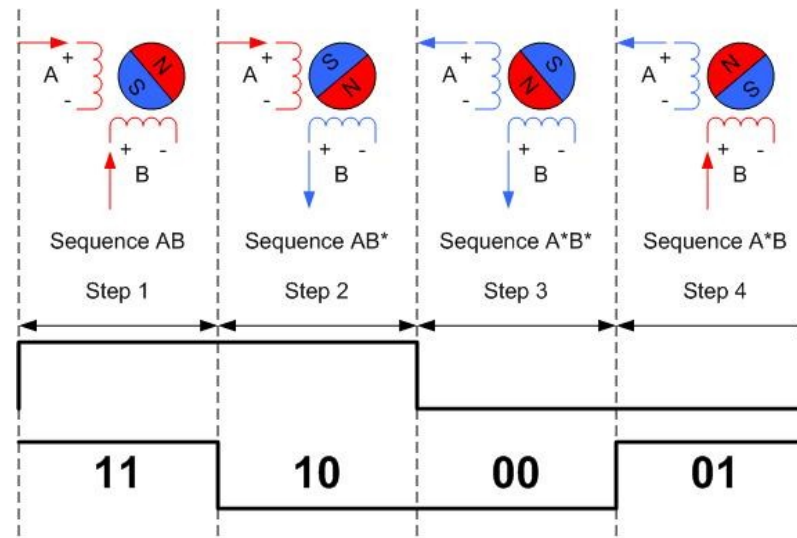
void setup()
{
  servo.attach(9)
}

void loop()
{
  servo.write(0); // 0°
  delay(500);
  servo.write(90); // 90°
  delay(500);
}
```



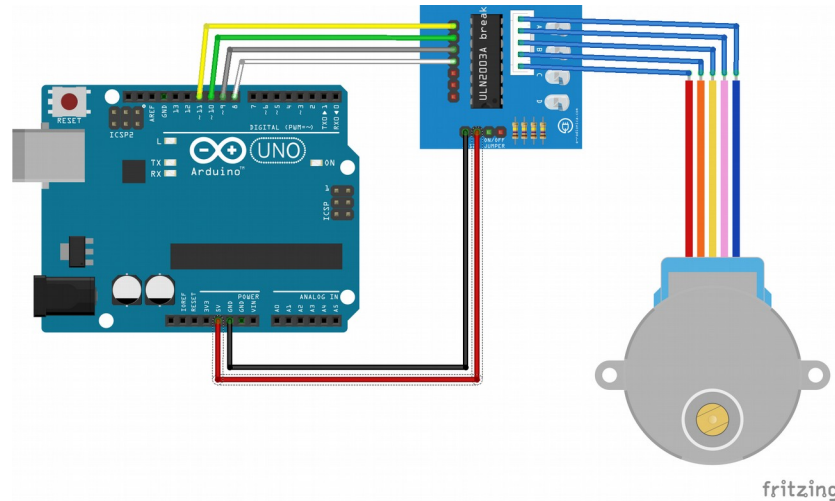
Moteurs à pas

- Les moteurs à pas sont basés sur des pôles, en général deux, qui sont activés avec des polarités différentes selon une certaine séquence, par exemple :



Moteurs à pas

- Nous utiliserons une carte de contrôle pour piloter le moteur, qui se branche comme cela :



A vous de jouer !

- Cette carte est dotée de 4 broches, IN1, IN2, IN3 et IN4
- A partir du tableau ci dessous, faites tourner le moteur dans un sens, puis dans l'autre !

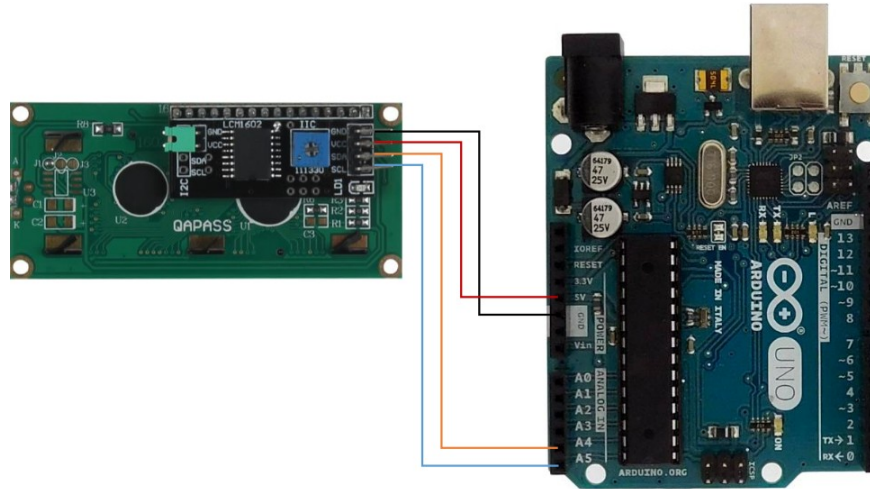
Phase	IN1	IN2	IN3	IN4
1	HIGH	LOW	LOW	LOW
2	LOW	HIGH	LOW	LOW
3	LOW	LOW	HIGH	LOW
4	LOW	LOW	LOW	HIGH

Manipulation 3

Écran LCD

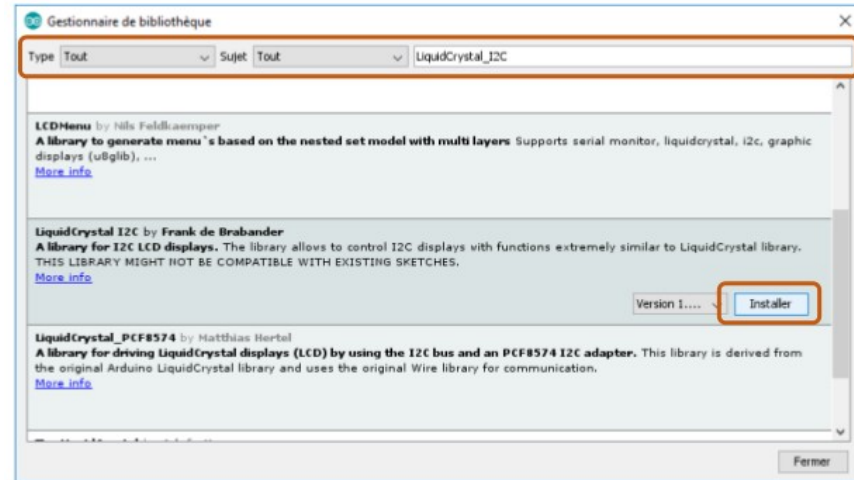
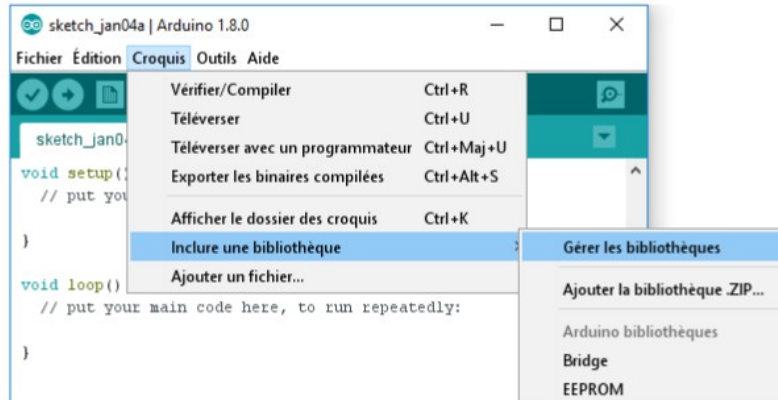
Ecran LCD en I²C

- Le kit fourni contient un écran LCD en I²C, pour le brancher nous allons utiliser le câblage suivant :



Écran LCD en I²C

- Rendez-vous ensuite dans « Gérer les bibliothèques » et installez LiquidCrystal_I2C :



Écran LCD en I²C

- Voici un exemple de programme qui affiche le temps écoulé :
 - 0x27 est l'adresse I²C du périphérique
 - 16 et 2 sont le nombre de lignes et colonnes
 - *setCursor* permet de se placer sur une ligne et une colonne données
- [\[lcd.ino\]](#)

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{
    lcd.init();
}

void loop()
{
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Temps:");
    lcd.setCursor(0, 1);
    lcd.print(millis());
    lcd.print(" ms");
}
```

A vous de jouer !

- Le kit contient un joystick qui fournit une position X, Y et un bouton appuyable (qui sont des entrées sorties)
- Faites le fonctionner, et faites un programme qui déplace un « X » sur l'écran en fonction du joystick !



Manipulation 4

Communication avec Python

Communication avec Python

- Il est possible de communiquer avec Python à travers l'interface série
- Un premier exemple qui permet d'envoyer des messages du code python vers l'écran LCD :
 - [\[lcd-receive.ino\]](#)
 - [\[send.py\]](#)

Communication avec Python

- Voici un exemple qui fonctionne dans l'autre sens, qui échantillonne le JoyStick et le lis depuis Python :
 - [\[joystick-send.ino\]](#)
 - [\[joystick.py\]](#)

Récupérer des valeurs

- Le programme [\[read.py\]](#) permet de récupérer des valeurs envoyées par *Serial.println()* et les stocke finalement dans un fichier CSV
- Le programme [\[plot.py\]](#) utilise la bibliothèque *matplotlib* pour dessiner une courbe des valeurs échantillonnées

A vous de jouer !

- En vous aidant des exemples précédent, ainsi que de [\[date.py\]](#), un programme Python qui affiche la date, écrivez un programme qui affiche la date sur la première ligne de l'écran LCD, et l'heure sur la deuxième ligne

On devrait voir les secondes bouger !



*Il y a encore plein de choses
dans le kit que vous n'avez pas
utilisé, n'hésitez pas à les
découvrir !*