

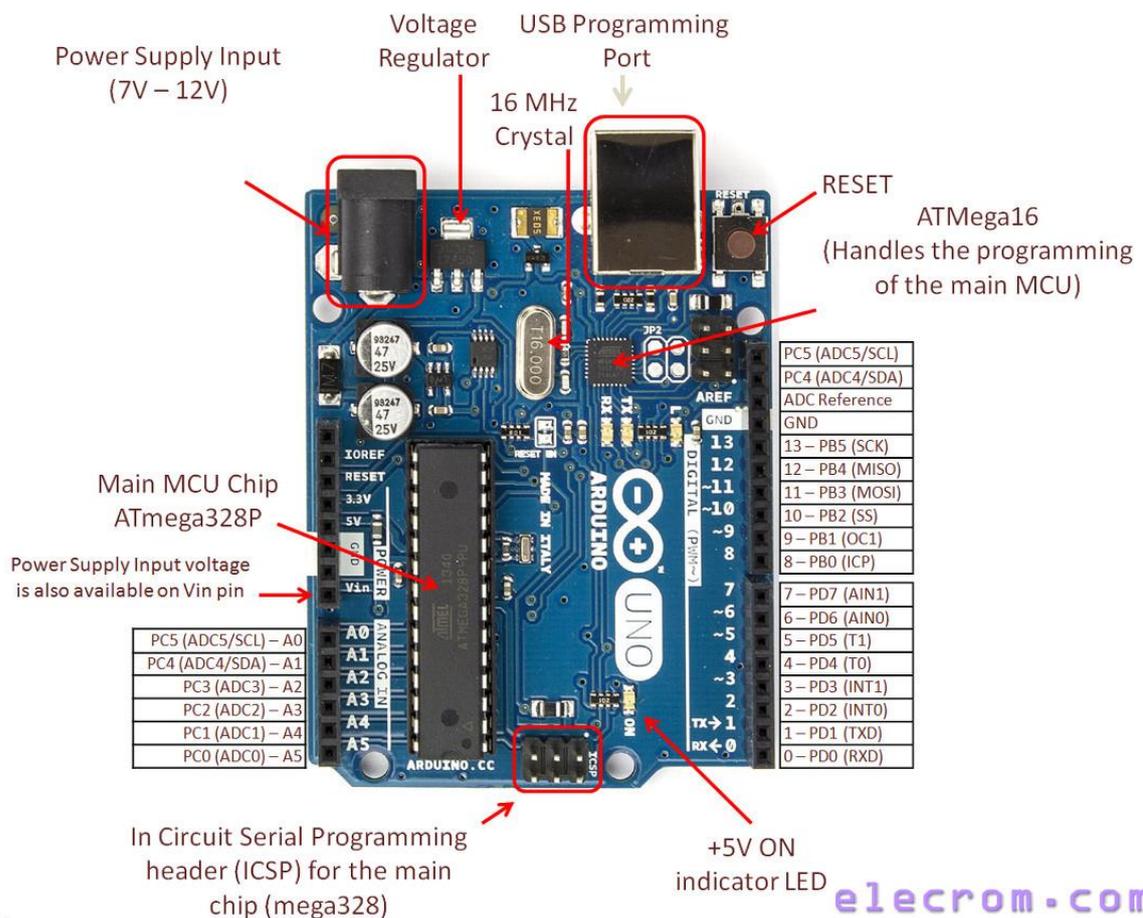
Le développement informatique

Introduction

Le développement informatique concerne tout ce qui touche à l'étude, à la conception, à la construction, au développement, à la mise au point, à la maintenance et à l'amélioration des logiciels/applications et sites web.

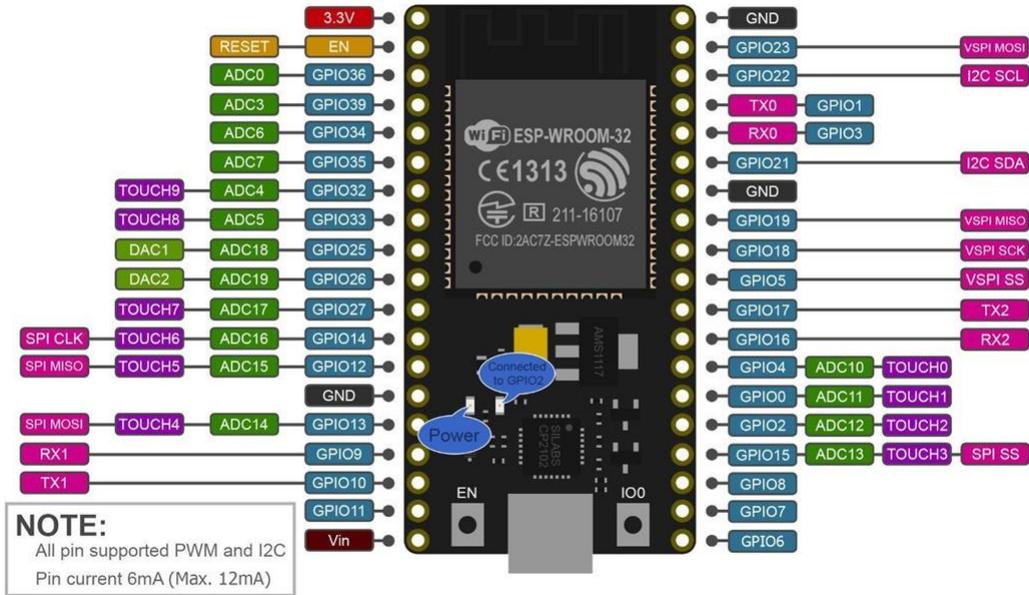
En SIN, pour effectuer du développement informatique nous étudierons plusieurs langages de programmation :

- Langages de programmation pour le développement logiciels/applications :
 - **Arduino** (très proche du C et du C++ – pour développer des programmes sur des cartes Arduino ou des microcontrôleurs de type ESP32).
Une fiche de synthèse sur les différentes commandes et spécificités du langage Arduino est disponible sur le site du professeur.
 - **Python** (pour développer des programmes sur des cartes Raspberry Pi).
Pour apprendre le langage Python une série de cours/problèmes est proposé sur le site France-ioi.org

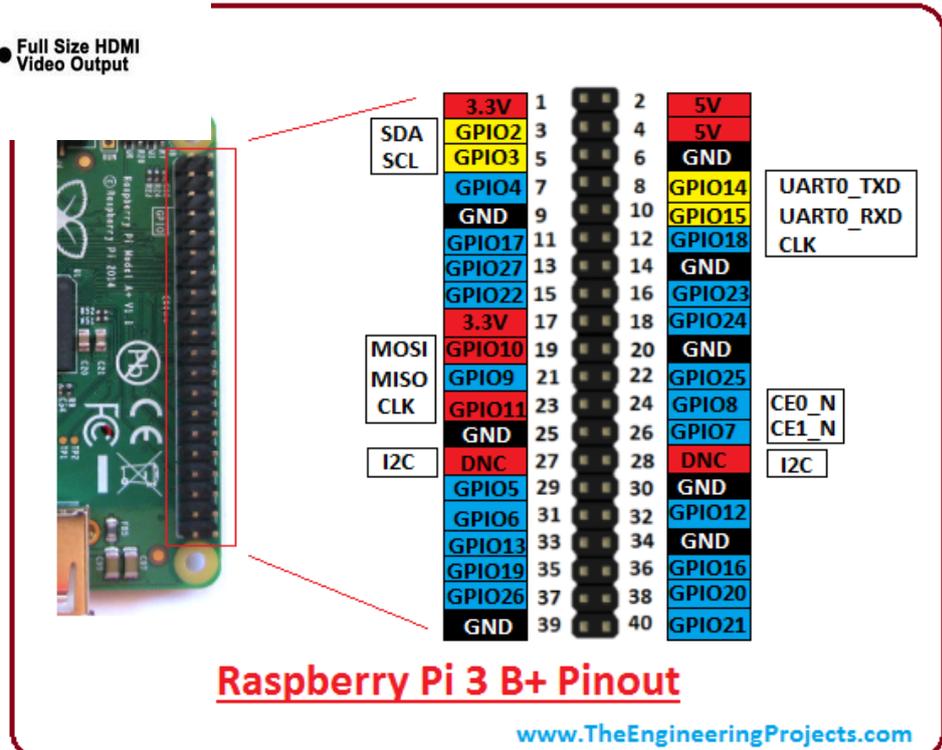
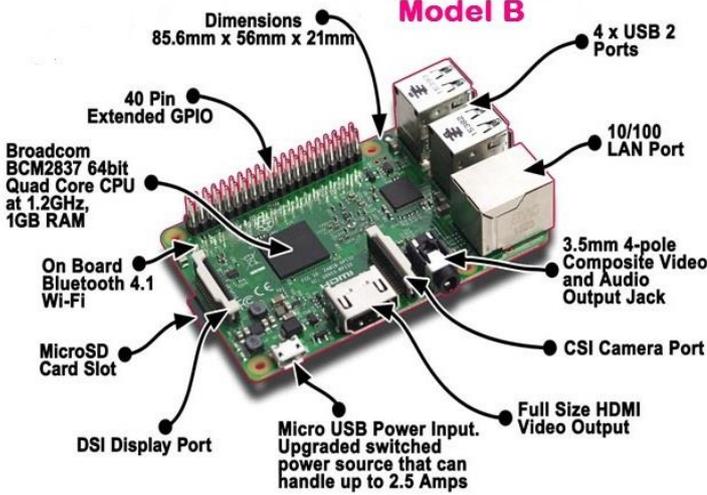


NodeMCU-32S

PINOUT



Raspberry Pi 3 Model B



Raspberry Pi 3 B+ Pinout

www.TheEngineeringProjects.com

- Langages de programmation pour le développement WEB :
 - **HTML** (HyperText Markup Language – Langage de balisage d’hypertexte – permet de représenter des pages web).
 - **CSS** (Cascading Style Sheet – feuille de style en cascade – Langage permettant de mettre en forme une page web).
Vous pouvez retrouver l’activité HTML/CSS effectué en classe de première sur le site du professeur.
 - **PHP** (Hypertext Preprocessor – permet de créer des pages web dynamiques).
Le langage PHP sera étudié en activité.

Encodage de l’information

Principe d'une base

La base est le nombre qui sert à définir un système de numération. La base du système décimal est dix alors que celle du système hexadécimal est 16 et la base du système binaire est 2. Quelque soit la base numérique employée, elle suit la relation suivante :

$$\sum_{i=0}^{i=n} (b_i a^i) = b_n a^n + \dots + b_5 a^5 + b_4 a^4 + b_3 a^3 + b_2 a^2 + b_1 a^1 + b_0 a^0$$

Où b_i est le chiffre de la base de rang i et a_i est la puissance de la **base a** d'exposant de rang i .

Exemple : base 10

$$1986 = (1 \times 10^3) + (9 \times 10^2) + (8 \times 10^1) + (6 \times 10^0)$$

Le système décimal

Le système décimal est celui dans lequel nous avons le plus l'habitude d'écrire. Chaque chiffre peut avoir 10 valeurs différentes : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. De ce fait, le système décimal a pour **base 10**. Tout nombre écrit dans le système décimal vérifie la relation suivante :

$$745 = 7 \times 100 + 4 \times 10 + 5 \times 1$$

$$745 = 7 \times 10 \times 10 + 4 \times 10 + 5 \times 1$$

$$745 = 7 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

Chaque chiffre du nombre est à multiplier par une puissance de 10 : c'est ce que l'on nomme le **poinds du chiffre**.

L'exposant de cette puissance est nul pour le chiffre situé le plus à droite et s'accroît d'une unité pour chaque passage à un chiffre vers la gauche.

$$12\ 435 = 1 \times 10^4 + 2 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 5 \times 10^0.$$

Cette façon d'écrire les nombres est appelée **système de numération de position**. Dans notre système conventionnel, nous utilisons les puissances de 10 pour pondérer la valeur des chiffres selon leur position, cependant il est possible d'imaginer d'autres systèmes de nombres ayant comme base un nombre entier différent.

Le système binaire

Dans le système binaire, chaque chiffre peut avoir 2 valeurs différentes : 0, 1. De ce fait, le système a pour base 2. Tout nombre écrit dans ce système vérifie la relation suivante :

$$(10\ 110)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$(10\ 110)_2 = 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1$$

$$\text{donc } (10110)_2 = (22)_{10}$$

Tous les systèmes de numération de position obéissent aux règles que nous venons de voir.

Le système hexadécimal

Le système hexadécimal utilise les 16 symboles suivants : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. De ce fait, le système a pour **base 16**.

Un nombre exprimé en base 16 pourra se présenter de la manière suivante : $(5AF)_{16}$.

Le nombre $(5AF)_{16}$ peut se décomposer comme suit :

$$(5AF)_{16} = 5 \times 16^2 + A \times 16^1 + F \times 16^0$$

En remplaçant A et F par leur équivalent en base 10, on obtient :

$$(5AF)_{16} = 5 \times 16^2 + 10 \times 16^1 + 15 \times 16^0$$

$$(5AF)_{16} = 5 \times 256 + 10 \times 16 + 15 \times 1$$

$$\text{donc } (5AF)_{16} = (1455)_{10}$$

Le système octal

Le système octal utilise un système de numération ayant comme base 8. Il faut noter que dans ce système nous n'aurons plus 10 symboles mais 8 seulement : 0, 1, 2, 3, 4, 5, 6, 7.

Ainsi, un nombre exprimé en base 8 pourra se présenter de la manière suivante : $(745)_8$

Lorsque l'on écrit un nombre, il faudra bien préciser la base dans laquelle on l'exprime pour lever les éventuelles indéterminations (745 existe aussi en base 10). Ainsi le nombre sera mis entre parenthèses (745 dans notre exemple) et indicé d'un nombre représentant sa base (8 est mis en indice).

Cette base obéira aux mêmes règles que la base 10, vue précédemment, ainsi on peut décomposer $(745)_8$ de la façon suivante :

$$(745)_8 = 7 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$$

$$(745)_8 = 7 \times 64 + 4 \times 8 + 5 \times 1$$

$$(745)_8 = 448 + 32 + 5$$

Nous venons de voir que : $(745)_8 = (485)_{10}$.

Base 10	Base 16	Base 2
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Système décimale	Système octal	Système binaire
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111
16	20	10000
17	21	10001
-	-	-
-	-	-
63	77	111111
64	100	1000000
65	101	1000001

ASCII

Le code ASCII (pour American Standard Code for Information Interchange) est une norme informatique de codage de caractères alphanumériques. ASCII comprend 128 combinaisons de 7 bits historiquement qui définissent 95 caractères imprimables, dont les chiffres arabes de 0 à 9, les lettres de l'alphabet latin non accentuées, minuscules et majuscules, des symboles de ponctuation et quelques caractères mathématiques simples.

Il convient bien pour coder les textes en anglais, mais est insuffisant pour le français, notamment en raison de l'**absence des caractères accentués**. Il est à noter aussi que les 32 premières combinaisons ne sont pas imprimables ainsi que la dernière, 127 : toutes correspondent à des commandes de terminal informatique. Ainsi, la combinaison 127 correspond à la commande pour effacer.

Le code ASCII a été étendu sur 8 bits pour incorporer certains caractères accentués (notamment les accents français). On voit ci-dessous la deuxième partie du code ASCII dit *étendu* codé sur 8 bits, la première partie étant le code ASCII historique sur 7 bits qui code en décimal de 0 à 127, en hexadécimal de 00h à 7Fh. L'extension ci-dessous code donc en décimal de 128 à 255 et en hexadécimal de 80h à FFh.

ASCII control characters			ASCII printable characters			Extended ASCII characters										
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	õ
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	ö
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	Ł	228	ø
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	ł	229	ő
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	á	166	ª	198	Ł	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	ł	231	þ
08	BS	(Backspace)	40	(72	H	104	h	136	ê	168	¿	200	Ł	232	þ
09	HT	(Horizontal Tab)	41)	73	I	105	i	137	ë	169	®	201	ł	233	Û
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	Ł	234	Ü
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ì	171	½	203	ł	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	í	172	¼	204	Ł	236	Ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	î	173	»	205	ł	237	Ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ë	174	«	206	Ł	238	–
15	SI	(Shift In)	47	/	79	O	111	o	143	Ä	175	»	207	ł	239	ˆ
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	»	208	Ł	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	»	209	ł	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	»	210	Ł	242	±
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ø	179	»	211	ł	243	¼
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	»	212	Ł	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	»	213	ł	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	»	214	Ł	246	§
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	»	215	ł	247	˚
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	»	216	Ł	248	˚
25	EM	(End of medium)	57	9	89	Y	121	y	153	ÿ	185	»	217	ł	249	˚
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	»	218	Ł	250	˚
27	ESC	(Escape)	59	;	91	[123	{	155	ø	187	»	219	ł	251	˚
28	FS	(File separator)	60	<	92	\	124		156	£	188	»	220	Ł	252	˚
29	GS	(Group separator)	61	=	93]	125	}	157	Ø	189	»	221	ł	253	˚
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	»	222	Ł	254	˚
31	US	(Unit separator)	63	?	95	_			159	f	191	»	223	ł	255	nbsp
127	DEL	(Delete)														

UNICODE

Unicode est un standard informatique qui permet des échanges de textes dans de **très nombreuses langues**, à un niveau mondial. Il est développé par le **Consortium Unicode**, qui vise au codage de texte écrit en donnant à tout caractère de n'importe quel système d'écriture un nom et un identifiant numérique, et ce de manière unifiée, quelle que soit la plateforme informatique ou le logiciel utilisés.

À ce jour, Unicode référence plus de **137 000 caractères** dans une centaine d'écritures.

La principale caractéristique d'UTF-8 (variante d'Unicode) est qu'elle est **rétro compatible** avec la norme ASCII, c'est-à-dire que tout caractère ASCII se code en UTF-8 sous forme d'un unique octet, identique au code ASCII. Par exemple, « A » (A majuscule) a pour code ASCII 65 et se code en UTF-8 par l'octet 65.

Exercices :

- 1) Convertir $(156)_{10}$ en binaire.
- 2) Convertir $(10100110)_2$ en décimal.
- 3) Convertir $(120)_{10}$ en hexadécimal.
- 4) Convertir $(AF)_{16}$ en décimal.
- 5) Convertir $(11101100)_2$ en hexadécimal.
- 6) Convertir $(7C)_{16}$ en binaire.
- 7) Convertir $(162)_{10}$ en octal.
- 8) Convertir $(55)_8$ en décimal.
- 9) Convertir $(10011010)_2$ en octal.
- 10) Convertir $(63)_8$ en binaire.
- 11) Coder en hexadécimal le mot « Bienvenue » grâce au codage ASCII.
- 12) En ASCII, que faut-il faire pour passer une majuscule en minuscule ?