

# TP SIN

## Codage de l'information

### Support : Carte Arduino Méga

**Pré requis (l'élève doit savoir) :**

- Savoir utiliser un ordinateur
- Avoir réalisé : TP fonctions logiques de base

**Programme**

4: Gestion de l'information

Etude des Structures matérielles et logicielles associées au traitement de l'information

Mise en œuvre d'outils de programmation graphique Traitement programmé et composants programmables

Implémentation d'un programme dans un composant programmable

**Objectif terminale :**

L'élève doit être capable de coder une information

**Matériels :**

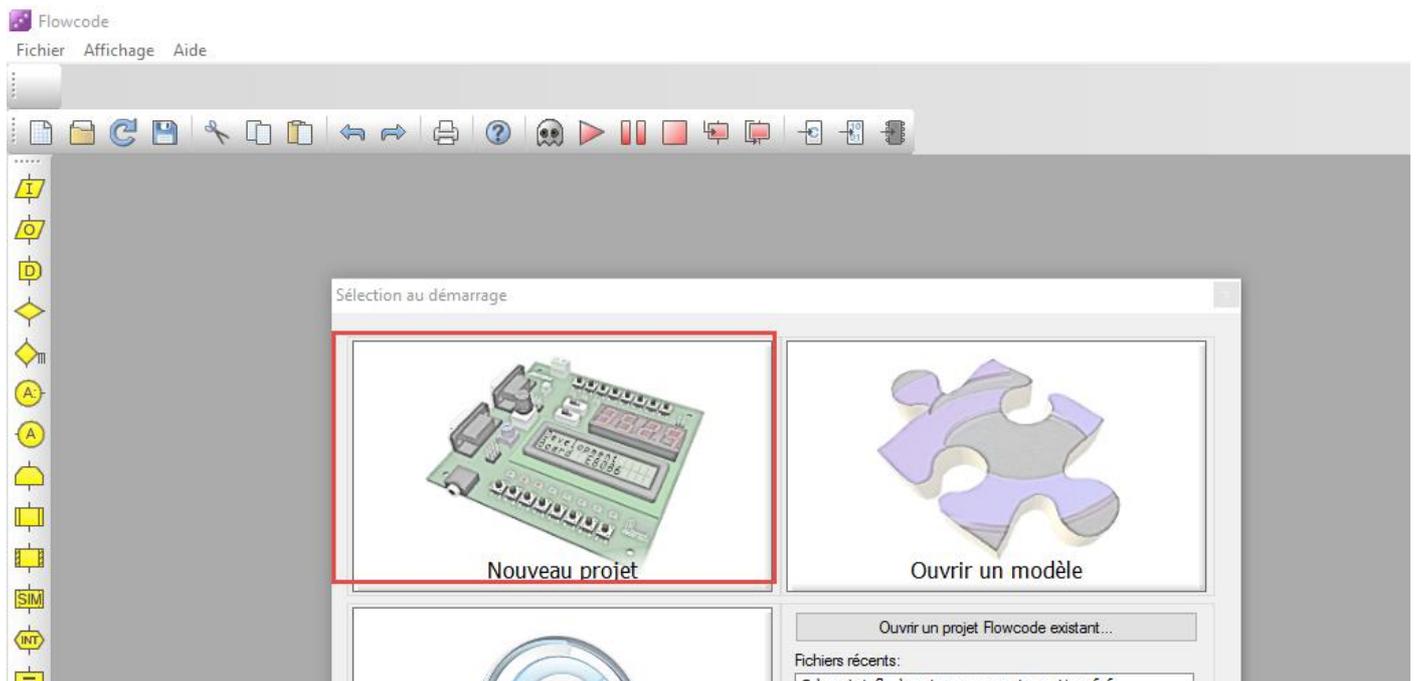
- Logiciel Flowcode

**a) Travail demandé****a) Introduction**

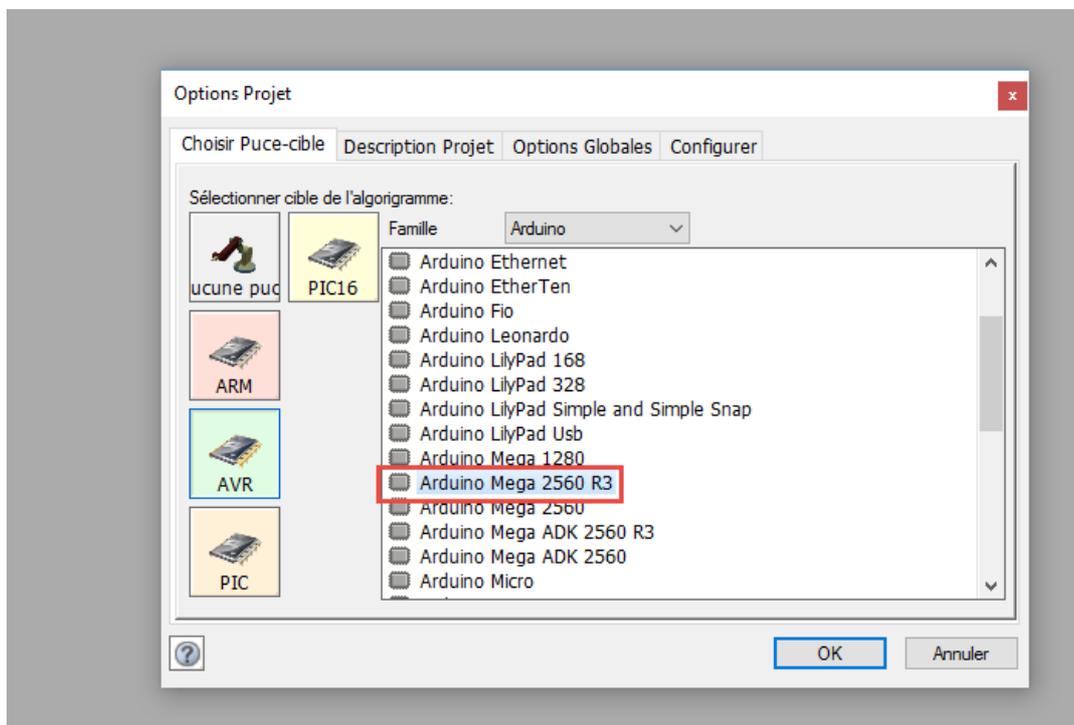
Les microcontrôleurs modernes, comme le PIC, sont capables de faire des tâches simples de mathématiques avec des nombres sur 8 bits rapidement. Lorsque les calculs deviennent plus complexes ou qu'il y a dépassement des possibilités de comptage sur 8 bits(255). Flowcode 6 nous permet d'utiliser des calculs complexes sur des nombres à 16 bits au maximum (toute la complexité de manipulation est transparente). Nous devons cependant tenir compte que cela ralentira la vitesse d'exécution du programme.

b) Flowcode et paramétrage de programmation du microcontrôleur

- Nouveau projet



- Sélectionner la bonne carte

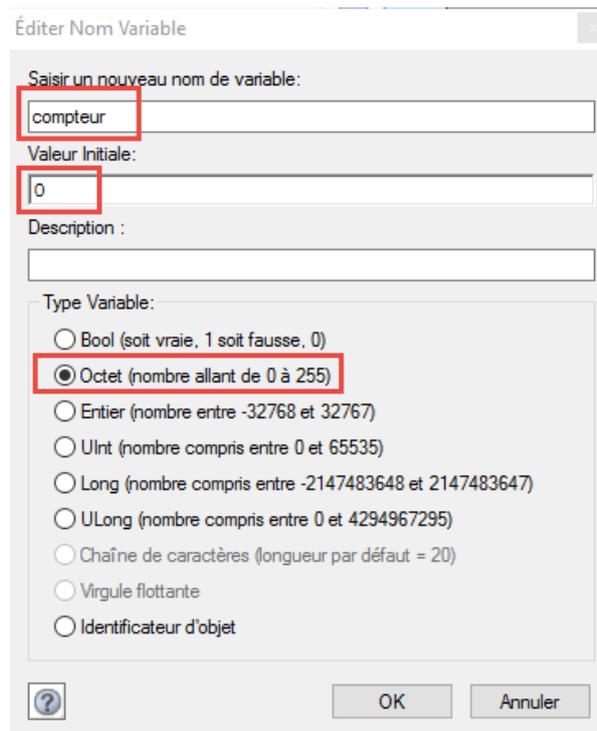


c) Instructions

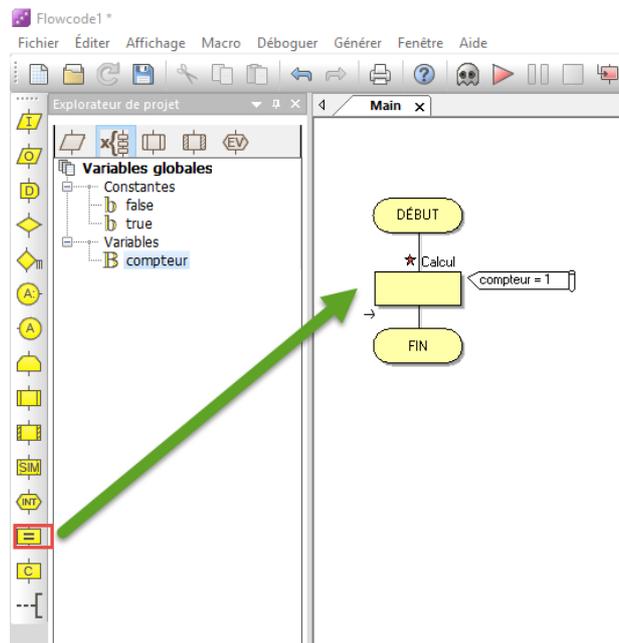
Utiliser l'aide de Flowcode pour obtenir plus d'informations sur le calcul et les possibilités d'utilisation de variables. Au cours de ces exercices, vous allez envoyer différents codes 8-bit au port B de votre microcontrôleur. Vous apprendrez également comment utiliser des variables et effectuer des calculs sur les nombres de 8 bits.

d) Travaux pratiques

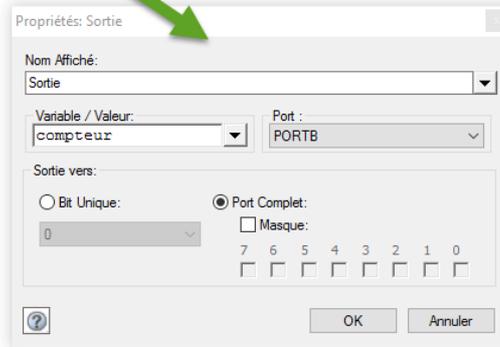
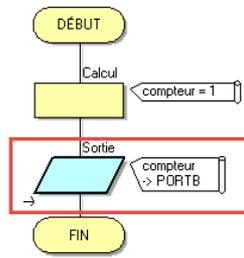
- Créer la variable (8 bit dit octet) que l'on appellera ' compteur ' (**MENU Editer=>Variables=>Ajouter Variable**)



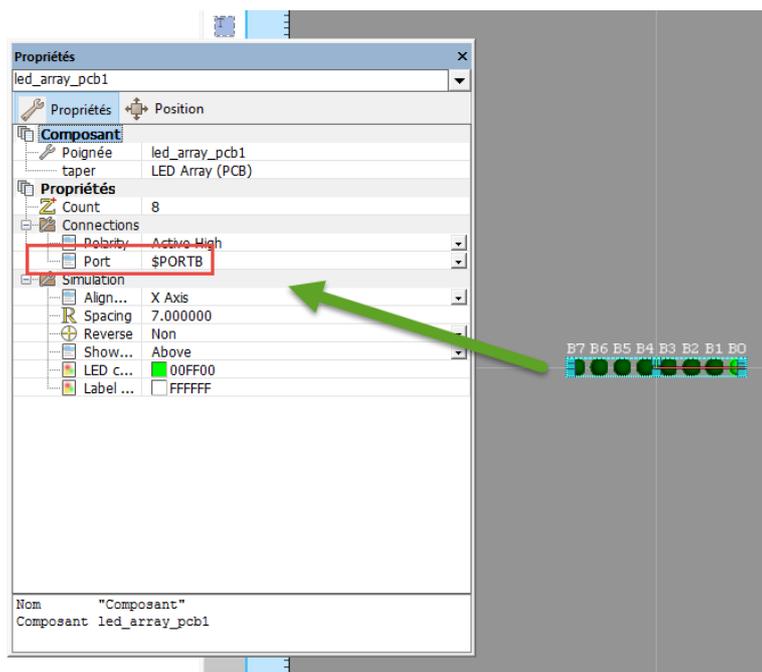
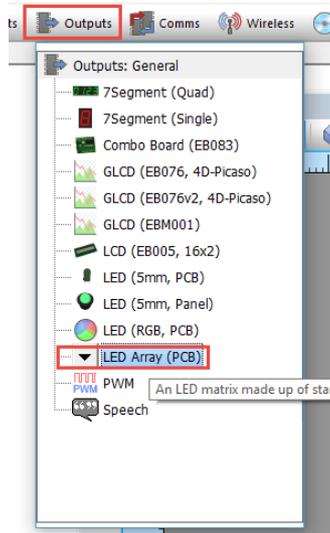
- Charger la valeur '1' dans cette variable Pour ce faire, ajouter un icône de calcul avec : 'compteur = 1'



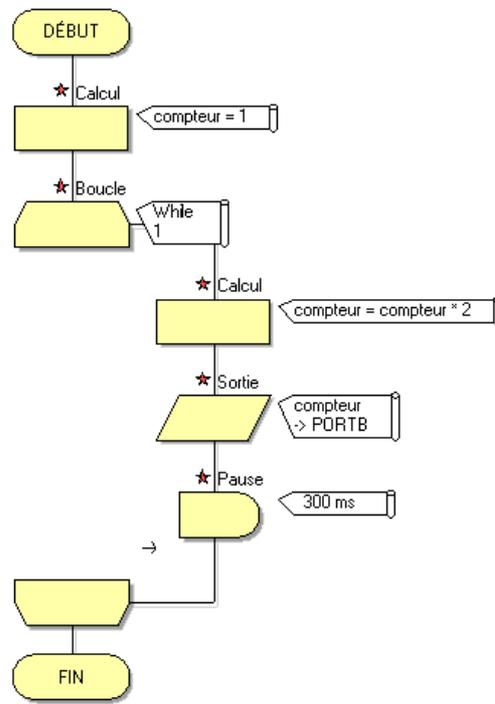
- Afficher la valeur de la variable ' compteur ' sur les LEDs (PORT B)



- Construire l'interface virtuelle à LED



- Lancer la simulation, quelle(s) LED(s) s'allume(nt) ?
- Multiplier la valeur de 'compteur' par 2 et afficher sur les 8 DEL. Pour ce faire, ajouter un icône de calcul avec : 'compteur = compteur \* 2'
- Répétez ces étapes dans une boucle programme infinies et ajouter un retard de 300ms entre ces étapes. Qu'observez-vous ? Comment expliquez-vous cela ? Faire valider par votre professeur.



- Remplacer la 'multiplication par 2' par l'instruction 'compteur = compteur + 1'... Qu'observez-vous? Vous avez programmé un compteur binaire.

- Pour chaque valeur décimale, remplacer les valeurs décimales en valeur binaire et faire le calcul. Puis comparer au résultat en décimal. Ensuite afficher le résultat des calculs suivants sur les LED du port B, en utilisant Flowcode et la commande de calcul et contrôler le résultat en binaire.

(Détaillez les calculs)

$$45 + 52 =$$

$$60 - 23 =$$

$$12 \times 9 =$$

- Pour chaque valeur décimale, remplacer les valeurs décimales en valeur binaire et faire le calcul. Puis comparer au résultat en décimal. Ensuite afficher le résultat des calculs suivants sur les LED du port B, en utilisant Flowcode et la commande de calcul et contrôler le résultat en binaire.

$$45 \text{ AND } 52 =$$

$$45 \text{ OR } 52 =$$

$$\text{NOT } 45 =$$

$$45 \text{ XOR } 52 =$$

- Ecrire l'équation ci-dessous en code ASCII base 16

Exemple :

4 correspond à 34 d'après le tableau ci-dessous

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	Space	Space	64	40	100	0	0	96	60	140		`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141		a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142		b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143		c
4	4	004	EOT (end of transmission)	36	24	044	\$	\$	68	44	104	D	D	100	64	144		d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145		e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146		f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147		g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150		h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151		i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152		j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153		k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154		l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155		m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156		n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157		o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160		p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161		q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162		r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163		s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164		t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165		u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166		v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167		w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170		x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171		y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172		z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173		{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174		
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175		}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176		~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

$$(1 - 2) \times (4/5) =$$