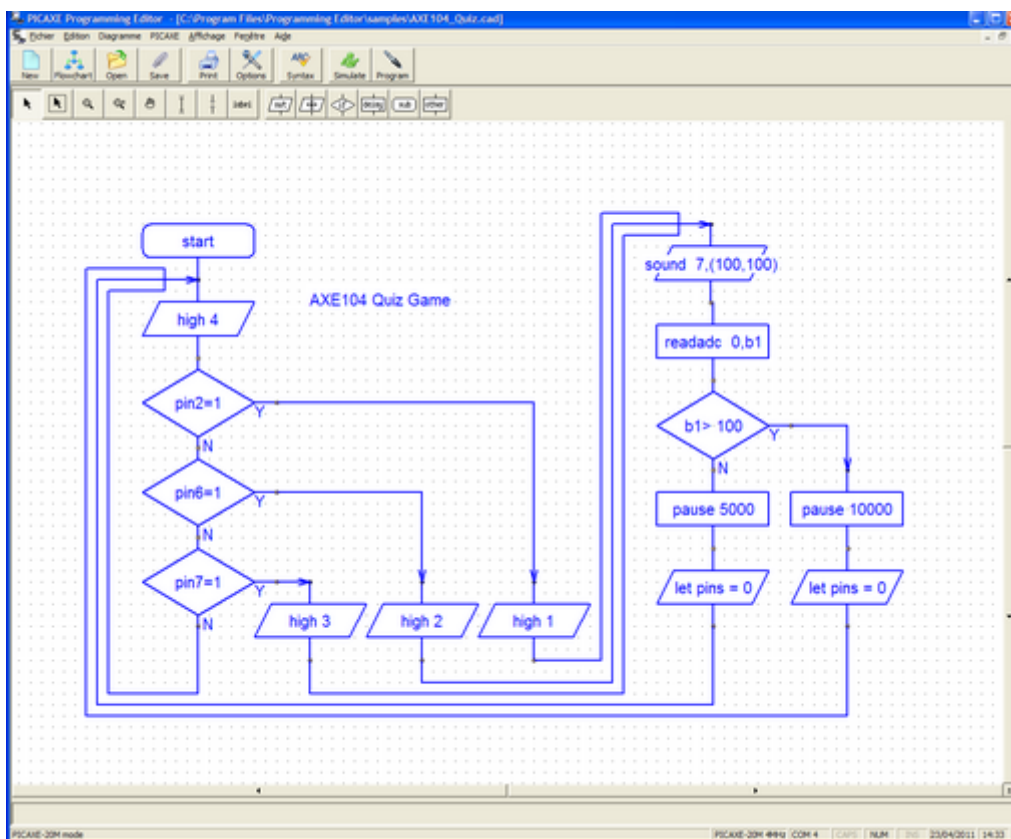


Prise en main rapide de

PICAXE Programming Editor

(© Revolution Education)
www.picaxe.co.uk



Logiciel téléchargeable gratuitement sur le site de l'auteur
ou chez www.a4.fr (Gamme PICAXE → L'indispensable pour utiliser Picaxe).

Table des matières

1Généralités.....	3
1.1Présentation du système.....	3
1.2Lancement, configuration.....	4
1.3Commandes générales.....	5
2Les modes de programmation.....	6
2.1« Fonctions logiques » (compatible automatisme câblé).....	6
2.2« Basic » (Langage de programmation).....	7
2.3« diagramme » (organigramme).....	8
3Simulation du circuit PICAXE.....	10
4Exercices de prise en main.....	11
4.1Faire clignoter une LED 0,5 s toute les secondes.....	11
4.2Allumer une LED si un bouton est appuyée.....	12
4.3Faire clignoter une LED 5 fois si un bouton est appuyée.....	13
4.4Détecter le niveau de lumière.....	14
5Cas de programmation.....	15
5.1Temporisation et minuterie (Pause ou Comptage).....	15
5.2Affichage sur l'écran LCD.....	16
5.3Utilisation de la télécommande.....	17
5.4Pilotage des moteurs.....	18
6Maquette.....	19
6.1La maquette.....	19
6.2Le simulateur de maquette.....	20

1 Généralités

1.1 Présentation du système

Définitions

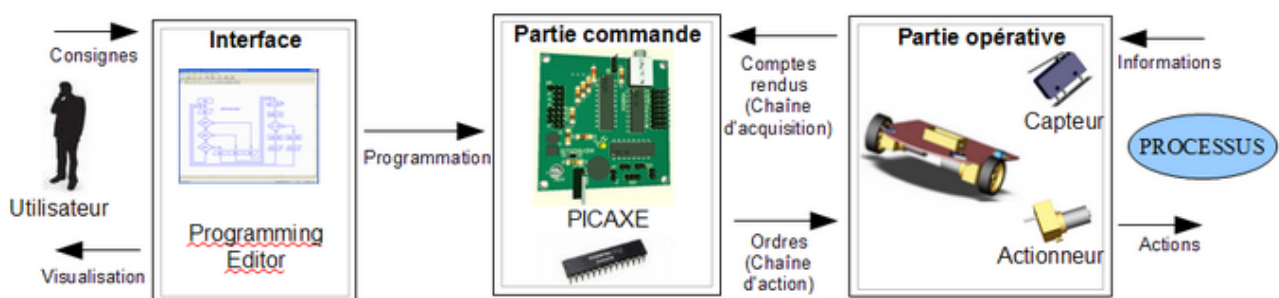
Automatisme : Machine conçu pour effectuer d'elle même un ensemble d'opération.

Partie commande : Partie qui reçoit des informations et donne des ordres.

Partie opérative : Partie qui exécute les ordres à l'aide d'actionneur et fournit des informations grâce aux capteurs.

Capteur : Organe qui transforme une grandeur physique en signal mesurable.

Actionneur : Élément qui transforme un signal en une action physique.



Notre système

Dans nos systèmes, la partie commande est réalisée par une carte électronique comportant un microcontrôleur programmable PICAXE. La partie opérative peut être des maquettes (4ème) ou un robot (3ème). L'interface de programmation est assurée par le logiciel Programming Editor.

Un capteur logique fournit une information binaire 0 ou 1. Un capteur analogique fournit comme information un nombre binaire (entre 0 et 255 en décimal).

Les informations binaires (bit) 0 ou 1 sont réalisées par la présence du signal électrique +5V ou son absence 0V. Un nombre binaire (octet) est une suite de 8 informations binaires.

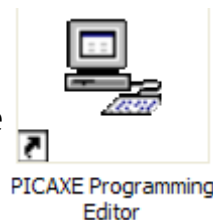
Un actionneur reçoit une information binaire 0 ou 1 pour activer ou désactiver une action (allumer une LED, arrêter un moteur, ...). Il peut aussi recevoir une suite de nombres pour indiquer un résultat (afficheur LCD).

Les microcontrôleurs PICAXE utilisés au collège possèdent :

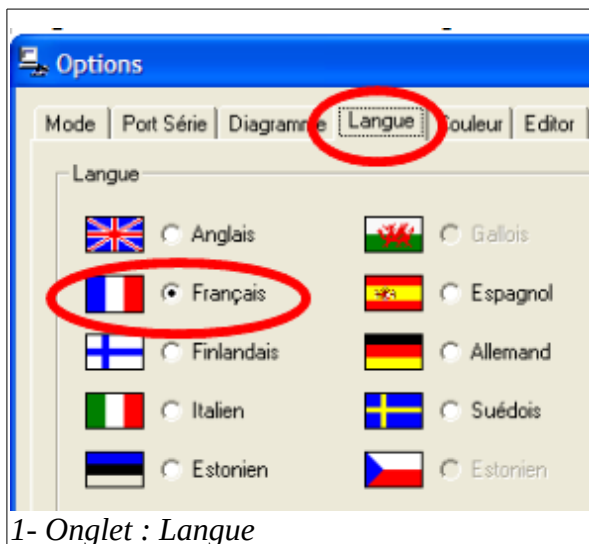
- 8 entrées pour connecter des capteurs logiques
- 4 entrées pour connecter des capteurs analogiques.
- 8 sorties logiques dont 4 sont connectées à un préactionneur permettant de piloter 2 moteurs dans les 2 sens de rotation.
- 13 variables (registres) permettant de stocker des informations (nombres).
- Un jeu d'instruction de programmation complet pour analyser les comptes rendus et générer les ordres appropriés.
- Il est programmable avec le logiciel Programming Editor.

1.2 Lancement, configuration

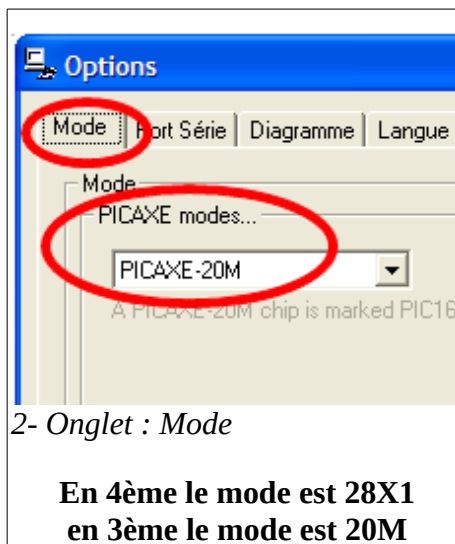
Cliquer sur l'icône PICAXE Programming Editor du bureau ou dans le menu Démarrer → Tous les programmes → Revolution Education → PICAXE Programming Editor.



Cliquer sur l'icône Options ou sur le menu Affichage → Options. Configurez les onglets suivants comme indiqués.

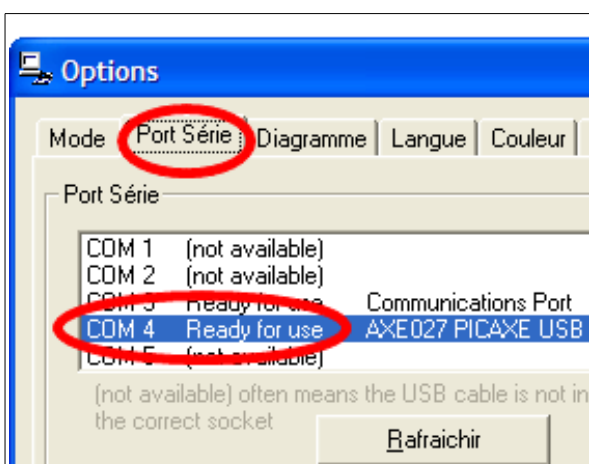


1- Onglet : Langue



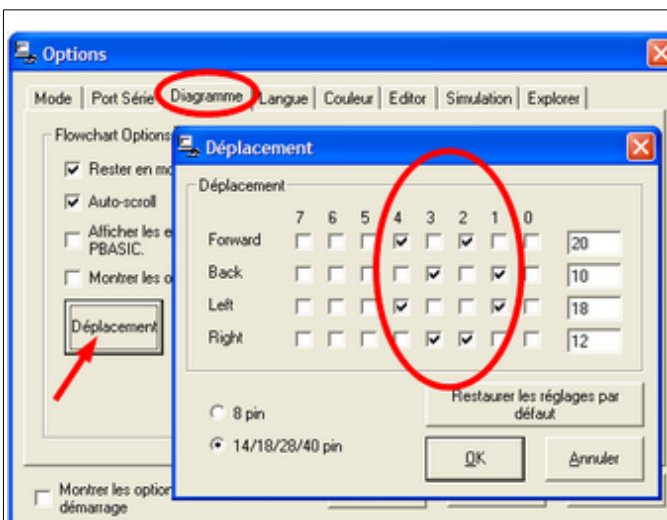
2- Onglet : Mode

**En 4ème le mode est 28X1
en 3ème le mode est 20M**



3- Onglet : Port

Le port, PICAXE USB n'est accessible que si le câble de programmation est branché en face avant (« Ready for use »).












4- Onglet : Diagramme








Les déplacements se font par les sorties 0 à 3 (3ème) ou 4 à 7 (3ème). Leur configuration dépend des actions recherchées et est spécifique à chaque programme.

1.3 Commandes générales

Barre d'outils générale

	Nouveau programme (Mode mnémonique)
	Nouveau diagramme (Mode organigramme)
	Ouvrir un fichier (.bas = mnémonique ; .cad = diagramme ; .lgc = logique)
	Sauvegarder un fichier (attention à l'emplacement choisi)
	Imprimer la fenêtre
	Boîte de dialogue Options (pour la configuration)
	Vérification de la syntaxe (pour le mode mnémonique)
	Simulation du fonctionnement (ne marche qu'en mode diagramme)
	Transfert du programme vers le microcontrôleur

Barre d'outils édition

	Outil sélection composant
	Outil sélection fenêtre
	Outil zoom fenêtre
	Outil zoom + ou – (molette souris)
	Outil déplacement schéma
	Outil tracé de liaison (un rond rouge indique les entrées ou sorties cliquables)
	Outil commentaire (permet de rendre la lecture du schéma plus explicite)

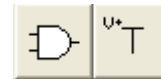
2 Les modes de programmation

2.1 « Fonctions logiques » (compatible automatisme câblé)

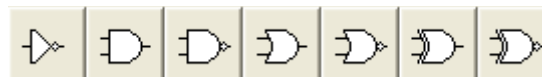
Il permet de remplacer des automatismes câblés.

Cliquer sur : Fichier → Nouveau → New logic.

La barre d'édition spécifique comporte 2 boutons.



Choix des portes logiques



Inverseuse (NON), AND (ET), NAND (NON ET), OR (OU), NOR (NON OU), XOR (OU exclusif), XNOR (NON OU exclusif).

Voir le cours sur les portes logique pour leur utilisation.

Choix des niveaux logiques



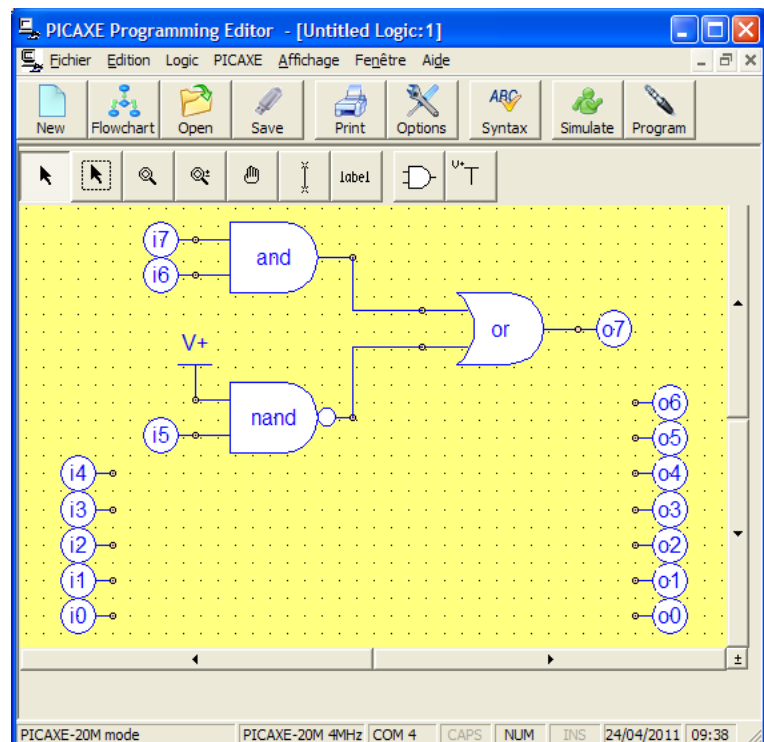
Tension d'alimentation (niveau haut ou 1), Masse de référence (niveau bas ou 0)

Il est parfois utile d'imposer un niveau particulier à l'entrée d'une porte.

Les entrées disponibles sont représentées à droite et les sorties à gauche. Elles peuvent être déplacées à volonté.

exemple : Le voyant vert (o7) est allumé s'il n'y a pas de pièce (i5) ou s'il y a une pièce (i7) et que la sécurité est en place (i6).

Une fois la (ou les) fonction(s) logique(s) représentée(s), transférez le schéma dans le microcontrôleur après avoir alimenté et relié la carte de commande. Vérifiez ensuite le fonctionnement.



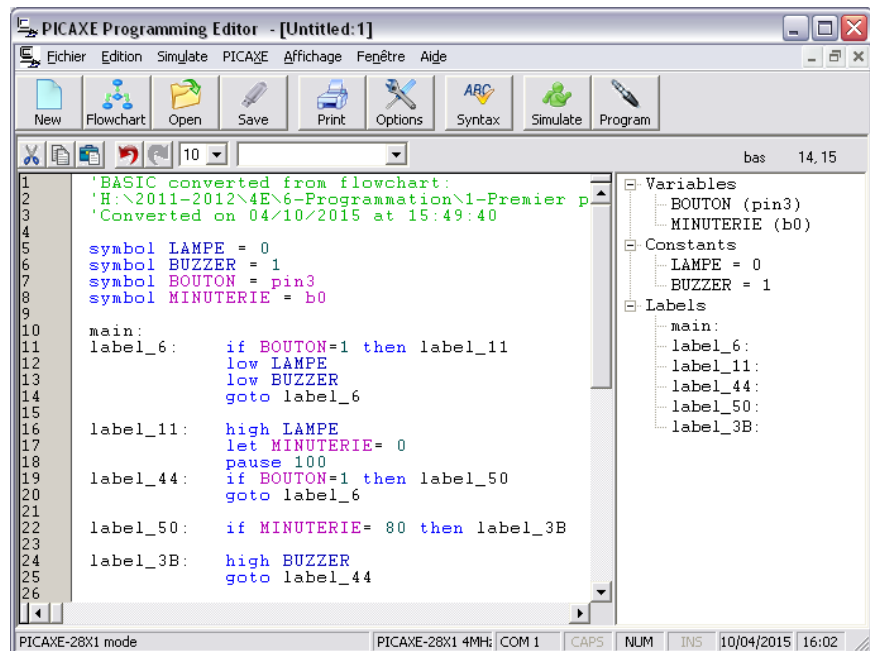
2.2 « Basic » (Langage de programmation)

Il permet la programmation directement en langage PICAXE BASIC.

Cliquer sur : Fichier → Nouveau → Basic ou directement sur l'icône (New).

La barre d'édition spécifique est celle d'un éditeur de texte.

Chaque ligne correspond à une instruction et porte un numéro d'ordre.



Symbol Nom = Valeur

Symbol permet d'indiquer le nom donné à un capteur (pin), une variable (b) ou un actionneur (). Il est suivi d'un espace ou plus généralement d'une tabulation.

Les noms ne peuvent pas posséder d'espace ni de lettre accentuée ou de caractère spécial.

La valeur est composée du préfixe (entre parenthèse ci-dessus) et du numéro correspondant.

Label Commande Paramètres

**Pour utiliser ce mode, consulter le manuel
« PICAXE manual 2 – BASIC Commands »
dans le menu « Aide ».**

Label (optionnel) est le point de branchement d'une instruction de saut. Il est suivi d'un « : » pour le séparer de la partie commande.

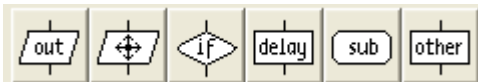
Les commandes sont décrites (fonctionnement et liste des paramètres) dans le manuel.

2.3 « diagramme » (organigramme)

Cliquer sur Fichier → Nouveau → Diagramme ou directement sur l'icône (Flowchart)

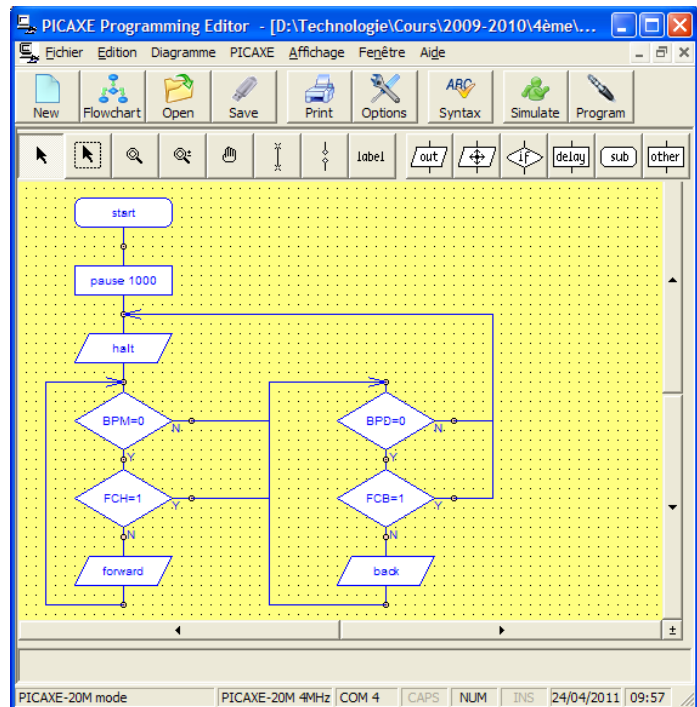
L'emplacement du début du programme est indiqué par START.

La barre de boutons d'édition spécifique comprend 6 boutons.



Ne sont présentées ci-dessous que les commandes les plus utilisées. Voir Aide → Manuel2 pour les autres.

Une fois en place et sélectionnée, il est possible de paramétrer la commande dans la barre d'état (bas de l'écran).



Commandes des actionneurs

	Activation de la sortie indiquée (ex. Allumer LED)	high 0
	Désactivation de la sortie indiquée (ex. Éteindre LED)	low LED
	Modification simultanée de plusieurs sorties	let pins = 0 7 6 5 4 3 2 1 0
	Inversion de l'état d'une sortie.	toggle LED
	Envoi d'un message à un afficheur	serout 0 N2400 .()

Commandes des moteurs



					Avant, Arrière, Gauche, Droite et Arrêt. L'action réelle sur les moteurs est fonction de la configuration de la boîte de dialogue « Options – Diagramme - Déplacement ». (Voir fiche spécifique)

Choix en fonction d'un capteur (PIN) ou d'une variable (VAR)

								if pin1 = 0
								if CNT > 8


Les réponses Oui et Non sont orientées de différentes façon.


Commandes de temporisations


  Temporisations en 1000ème de seconde (pause) ou en seconde (wait)


pause	1000
wait	1

Commandes de programmes

 Arrêt du programme

 Début de sous programme


 Fin de sous programme


 Aller à un sous programme.


sub0	sub0
------	------


gosub	sub0
-------	------

Autres commandes

 Définition ou calcul avec d'une variables

 Lecture d'une entrée analogique vers une variable

 Lecture de température (DS18B20) vers une variable

 Autres commandes à saisir manuellement

let	b0	=	b0	+	1
-----	----	---	----	---	---

readadc	0	,	CNT
---------	---	---	-----

readtemp	0	,	CNT
----------	---	---	-----

BASIC	irin [100],0,b0
-------	-----------------

Utilisation de noms explicites

Les noms prédéfinis des entrées 0 à 7, des sorties 0 à 7 et des variables 0 à 13 peuvent être modifiés pour être plus explicites.

Commande Diagramme → Table des symboles

La table des symboles doit être définie en premier sous peine de devoir paramétrer à nouveau les commandes.

Si l'entrée 6 correspond au **bouton poussoir monté**, on peut écrire

if	BPM	=	0
----	-----	---	---

au lieu de

if	pin6	=	0
----	------	---	---

Symboles...	
Sorties...	Variables...
output 0 0	b0 Lumiere
output 1 MONTE	b1 b1
output 2 DESCEND	b2 b2
output 3 3	b3 b3
output 4 4	b4 b4
output 5 5	b5 b5
output 6 6	b6 b6
output 7 7	b7 b7
	b8 b8
	b9 b9
	b10 b10
	b11 b11
	b12 b12
	b13 b13
Entrées...	Restaurer les réglages par défaut
input 0 pin0	
input 1 pin1	
input 2 FCB	
input 3 FCH	
input 4 pin4	
input 5 BPD	
input 6 BPM	
input 7 pin7	
	OK Annuler

3 Simulation du circuit PICAXE

Il est possible de simuler le programme avant son transfert pour le tester.

Commande Diagramme → Simuler ou l'icône correspondante.

Le déroulement des instructions apparaît en rouge dans la fenêtre d'édition.

Un clic sur les entrées change leur état (sombre = 0, clair = 1).

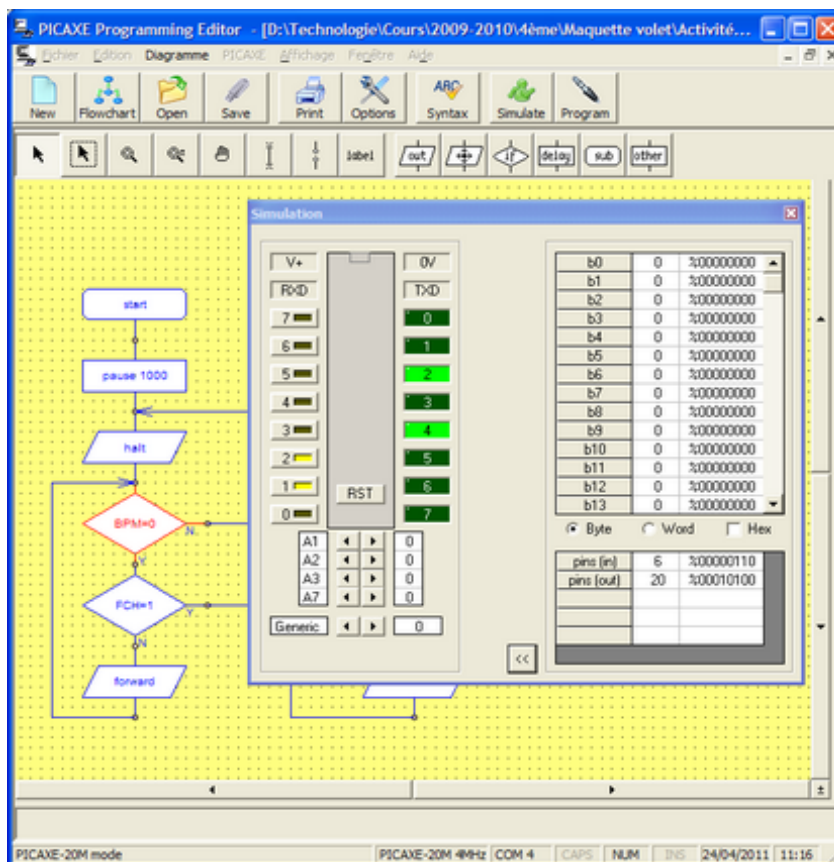
Les entrées analogiques peuvent être incrémentées ou décrémentées ou fixées directement.

L'état des sorties et des variables est indiqué. Les variables peuvent être modifiées en double cliquant dessus.

Pour arrêter la simulation il faut cliquer dans une zone libre de l'éditeur.

Les paramètres de la simulation sont fixés dans la boîte de dialogue « Options - Simulation ».

L'afficheur LCD ou la télécommande peuvent être simulés.



Une simulation plus réaliste comprenant les capteurs et les actionneurs est possible avec le programme « Simulateur de maquette »

4 Exercices de prise en main

4.1 Faire clignoter une LED 0,5 s toute les secondes

La LED rouge est connectée sur la sortie 2 (fil noir à l'intérieur).

Écriture du programme

Démarrer le logiciel : « Programming Editor ». Le configurer (Options, Mode = 28X1, Langue = Français). Ouvrir une fenêtre diagramme (Flowchart).

Nommer l'actionneur : (Diagramme – Table des symboles) : Output 2 → LED

Écrire le programme :

1- Début

2- Éteindre la LED (désactiver).

Bouton « Commande de sortie » puis « État bas ». Sélectionner dans la liste de choix en bas de l'écran la sortie 2 (LED) . Retour au niveau supérieur.

3- Attendre 0,5 s (= 500 ms).

Bouton « Delais » puis « Pause (ms) ». Inscrivez la durée dans la case en vas de l'écran et validez. Retour au niveau supérieur.

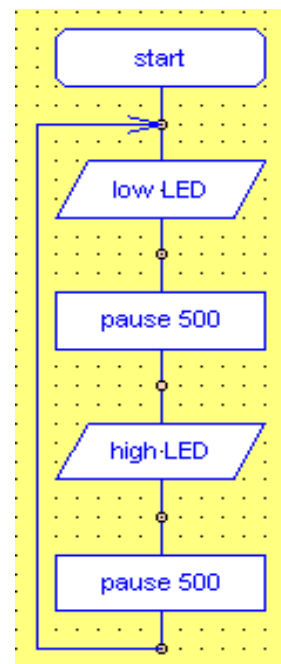
4- Allumer la LED (activer).

Bouton « Commande de sortie » puis « État haut». Sélectionner dans la liste de choix en bas de l'écran la sortie 2 (LED) . Retour au niveau supérieur.

5- Attendre 0,5 s (= 500 ms).

Bouton « Delais » puis « Pause (ms) ». Inscrivez la durée dans la case en vas de l'écran et validez. Retour au niveau supérieur.

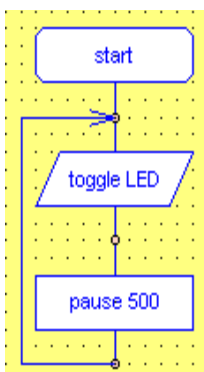
6- Aller à l'étape 2 (recommencer le programme).



Simulation et transfert

Simulez le fonctionnement puis transférez le programme sur la maquette après avoir connecté le câble USB (Options - Port série = PICAXE USB (Ready for use))

Autre écriture du programme



1 - Début

2- Inverser l'état de la LED

passer de désactivée à activée ou de activée à désactivé

3- Attendre 0,5 s (= 500 ms)

4- Aller à l'étape 2 (continuer à alterner les états haut et bas)

4.2 Allumer une LED si un bouton est appuyée

La LED rouge est connectée sur la sortie 2 (fil noir à l'intérieur).

Le bouton poussoir (BP) est connecté sur l'entrée 0 (fil noir à l'intérieur).

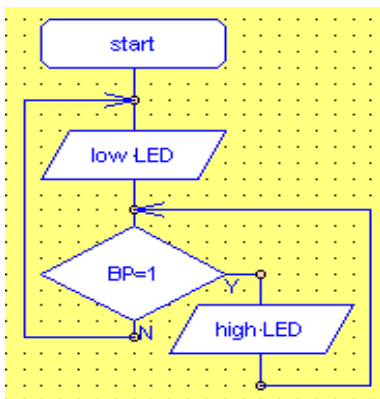
Écriture du programme

Démarrer le logiciel : « Programming Editor ». Le configurer (Options, Mode = 28X1, Langue = Français). Ouvrir une fenêtre diagramme (Flowchart).

Nommer les composants : (Diagramme – Table des symboles) :

Output 2 → LED ; Input 0 → BP

Écrire le programme :



1- Début

2- Éteindre la LED (désactivé par défaut) .

Bouton « Commande de sortie » puis « État bas ». Sélectionner dans la liste de choix en bas de l'écran la sortie 2 (LED) . Retour.

3- Choix en fonction du BP (appuyé = 1).

Bouton « Commande de choix » puis « Pin sortie oui à droite ». Sélectionner dans la liste de choix en bas de l'écran l'entrée 0 (BP) et l'état actif (1). Retour.

3a (N) - Si le bouton n'est pas appuyé (= 0), Mettre une liaison (sortie Non) pour aller à l'étape 2 (La LED reste éteinte tant-que le bouton (BP) n'est pas appuyé).

3b (Y) - Si le bouton est appuyé (= 1), on passe à l'étape 4.

4- Allumer la LED (activer).

5- Mettre une liaison pour aller à l'étape 3 (La LED reste allumé tant-que le bouton (BP) est appuyé).

Simulation et transfert

Simulez le fonctionnement (un clic sur l'entrée 0 la fait changer d'état), puis transférez le programme sur la maquette pour l'essayer.

4.3 Faire clignoter une LED 5 fois si un bouton est appuyée

La LED rouge est connectée sur la sortie 2 (fil noir à l'intérieur).

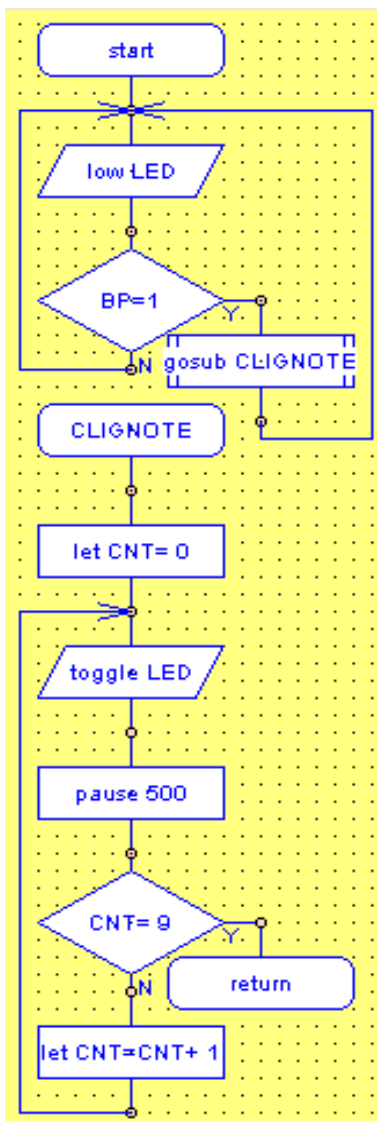
Le bouton poussoir est connecté sur l'entrée 0 (fil noir à l'intérieur).

Écriture du programme

Nommer les composants : (Diagramme – Table des symboles) :

Output 2 → LED ; Input 0 → BP ; Variable 0 → CNT

Écrire le programme :



On peut reprendre le programme précédent en remplaçant l'allumage de la LED par un sous-programme clignotement.

Sélectionner l'instruction « High LED » et la supprimer.

(4)- Insérer un appel à sous-programme.

(5)- Mettre une liaison pour aller à l'étape 1 à la fin du clignotement.

6- Mettre un début de sous-programme et le nommer « CLIGNOTE ».

7- Mettre le chiffre 0 dans la variable « CNT ».

8- Inverser l'état de la LED (clignotement).

9- Attendre 500ms (0,5s).

10a (Y)- Si la variable « CNT » égale 9 aller à l'étape 13.

10b (N)- Si la variable « CNT » est différente de 9 continuer.

11- Ajouter 1 dans la variable « CNT » (comptage).

12- Aller à l'étape 8 pour continuer le clignotement.

13- Quitter le sous-programme et reprendre le programme principal (si la variable « CNT » = 9, la LED a clignoté 5 fois).

14- Sélectionner l'appel à sous-programme et affectez lui le nom « CLIGNOTE ».

Simulation et transfert

Simulez le fonctionnement (un clic sur l'entrée 0 la fait changer d'état), puis transférez le programme sur la

maquette pour l'essayer.

Attention. Il faut penser à relâcher rapidement le bouton.

4.4 Détecter le niveau de lumière

Les LED rouge, jaune et verte sont connectées respectivement sur les sorties 2 à 4.

La LDR (détecteur de lumière) est connecté sur l'entrée analogique 0 (fil bleu à l'intérieur).

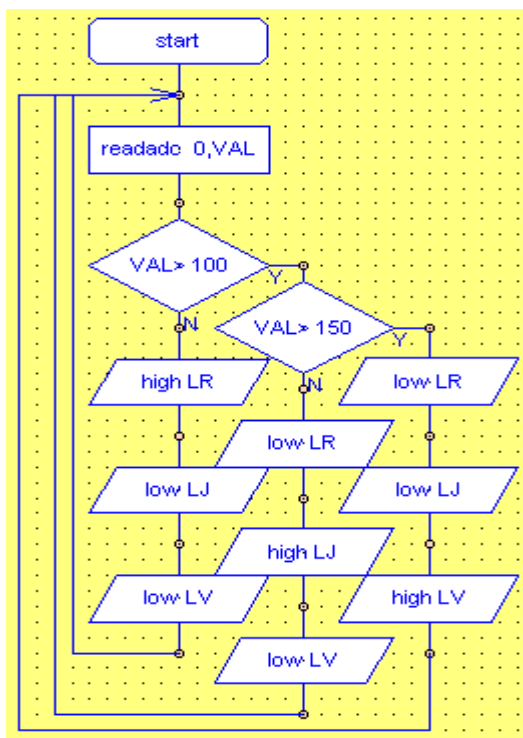
Écriture du programme

Nommer les composants :
(Diagramme – Table des symboles) :

Les entrées analogiques ne peuvent pas être nommées.

Écrire le programme :

Sorties...		Variables...	
output 0	LCD	b0	CNT
output 1	1	b1	VAL
output 2	LED R	b2	b2
output 3	LED J	b3	b3
output 4	LED V	b4	b4



1- Début

2- Lire l'entrée analogique 0 et mettre le résultat dans VAL.

3a (N)- Si le contenu de VAL est plus petit ou égal à 100, allumer la LED rouge et éteindre les autres avant de retourner au début.

4a (N)- Si le contenu de VAL est plus petit ou égal à 150 (entre 101 et 150), allumer la LED jaune et éteindre les autres avant de retourner au début.

4b (O)- Si le contenu de VAL est plus grand que 150, allumer la LED rouge et éteindre les autres avant de retourner au début.

Simulation et transfert

Simulez le fonctionnement (Un clic sur le curseur de l'entrée analogique 0 change la valeur lue. On peut aussi indiquer directement la valeur dans la case correspondante.).

Transférez le programme sur la maquette pour l'essayer.

5 Cas de programmation

5.1 Temporisation et minuterie (Pause ou Comptage)

Une action est conditionnée à l'écoulement d'un délai depuis une action précédente.

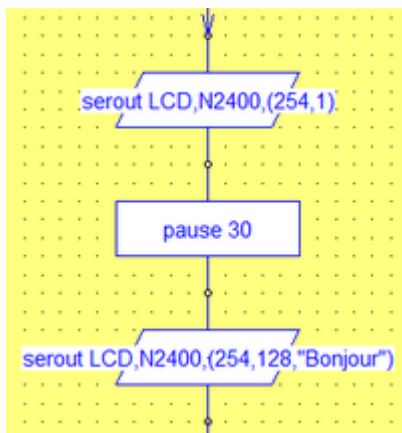
Courte durée

Il est parfois nécessaire de retarder une action pour laisser le temps à un capteur ou un actionneur de changer d'état.

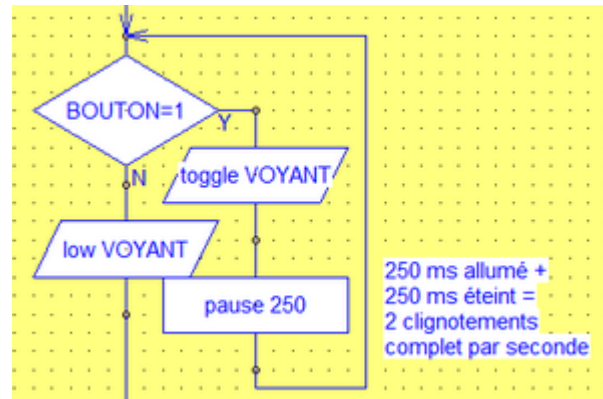
Cela peut aussi être nécessaire pour permettre à l'opérateur de se rendre compte d'un événement (le processeur fonctionne à la fréquence de 4 MHz alors que l'œil ne réagit que pour des fréquences inférieures à 25 Hz).

Ici, la commande appropriée est PAUSE. Elle interrompt le fonctionnement du programme pendant la durée indiquée. Il continue normalement à l'issue.

Inconvénient, durant ce délai, le changement d'état d'un capteur ne peut plus être détecté.



Exemple 1 : L'effacement correct de l'afficheur LCD nécessite qu'aucune autre commande ne lui soit envoyée avant un délai de 30 ms.



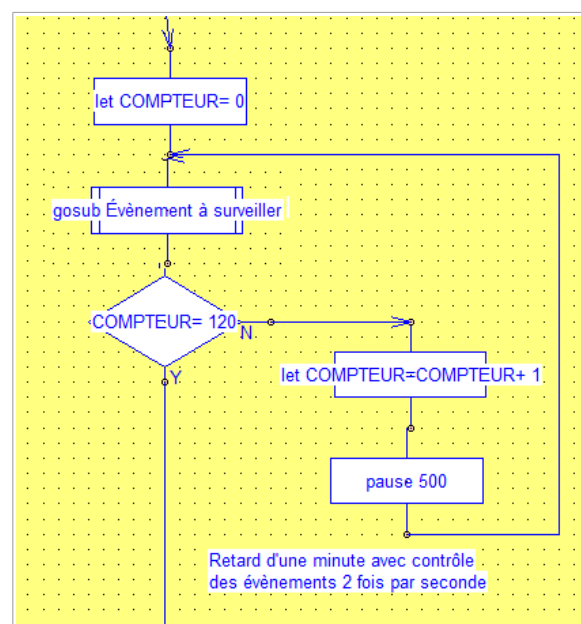
Exemple 2 : Le clignotement du voyant est limité à 2 fois par seconde (2 Hz) pour éviter d'avoir l'air d'être allumé en continu mais faiblement.
Le bouton doit donc rester appuyé au moins ¼ de seconde afin d'être sûr d'être pris en compte.

Longue durée

Ici les capteurs doivent être interrogés en continu mais l'action doit attendre un certain temps avant d'être effectuée.

Exemple 3: Un délai d'une minute est nécessaire pour l'action suivante mais un événement doit être contrôlé durant tout ce temps.

Un compteur est initialisé à 0, puis incrémenté régulièrement jusqu'au délai voulu. L'événement à surveiller est placé dans la boucle afin de continuer à être surveillé régulièrement.



5.2 Affichage sur l'écran LCD

Une LDR (capteur de lumière) est branché sur l'entrée analogique 0 et la valeur lue placée dans la variable VAL (b1) avant d'être visualisée sur l'afficheur LCD qui est connecté sur la sortie directe 1.

L'afficheur LDC

L'afficheur dispose de 2 lignes de 16 caractères affichables (noté de 0 à 15). L'affichage d'un caractère à une position remplace le caractère déjà présent. Les caractères suivant en font de même avec les positions suivantes.

Il peut recevoir un code de commande ou afficher un texte ou une valeur :

- (254,1) = Efface l'afficheur (doit être suivi d'une pause de 30ms)
- (254,128) = Aller au début de la 1ère ligne (ajouter à 128 le n° du caractère pour les autres positions de la 1ère ligne)
- (254,192) = Aller au début de 2ème ligne (ajouter à 192 le n° du caractère pour les autres positions de la 2ème ligne)
- ("Texte à afficher") = Affiche le texte indiqué entre guillemets (attention pour les accents et les symboles)
- (#bx) = Affiche le contenu de la variable x (0 à 13)

Les commandes, textes et valeurs dans une même commande doivent être séparés par des virgules

Écriture du programme

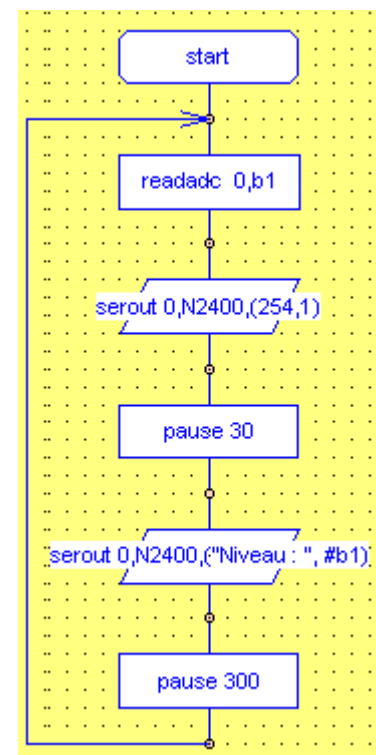
La lecture d'une valeur analogique se fait avec l'instruction READADC.

L'écriture d'une information sur l'afficheur se fait avec l'instruction SEROUT.

La syntaxe est la suivante "SEROUT 0, N2400, (xxxx)" avec :

- 1 = n° de la sortie utilisée,
- N2400 = paramètres de transmission
- (xxxx) = message.

Écrire le programme, le faire contrôler et tester par le professeur qui dispose de l'afficheur.



5.3 Utilisation de la télécommande

Un récepteur de code de télécommande infra/rouge doit être utilisé pour détecter les codes émis.

La télécommande

Elle doit être configurée à chaque changement de pile.

TVR010A (Transparent Blue colour, with arrow keys)

1. Insert 2 AAA size batteries, preferably alkaline.
2. Press 'S' and 'B' at the same time. S is in the centre of the arrows. The top left red LED should light.
3. Press '0'. The LED should flash.
4. Press '1'. The LED should flash.
5. Press '3'. The LED should go out.
6. Press the red power button (top right).



Les codes retournés sont indiqués ci-contre. Le sous programme de lecture ajoute 1 à tous les codes afin que 0 signifie une absence de réception.

Le programme

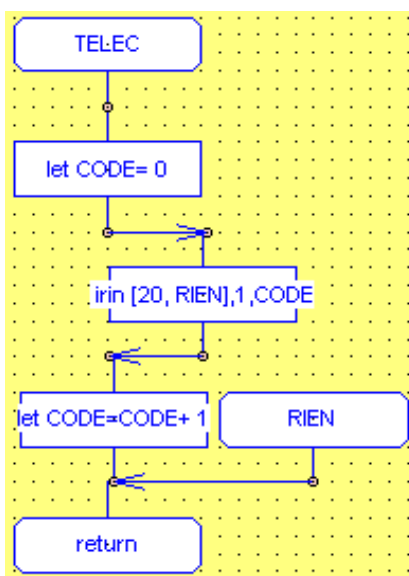
Il n'y a pas d'instruction spécifique en mode organigramme. Il faut donc utiliser l'instruction BASIC et saisir le texte de la commande.

irin [20, RIEN], 1, CODE

- irin : commande de lecture du récepteur
- 20 : durée d'attente (ms) pour la réception
- RIEN : instruction suivante si pas de réception
- 1 : numéro de l'entrée utilisée par le récepteur. (Il n'est pas possible d'utiliser un nom spécifique pour l'entrée)
- CODE : variable recevant le code si réception

TVR010A Key Values for PICAXE irin commands

Symbol KEY_POWER	= 21
Symbol KEY_1	= 0
Symbol KEY_2	= 1
Symbol KEY_3	= 2
Symbol KEY_4	= 3
Symbol KEY_5	= 4
Symbol KEY_6	= 5
Symbol KEY_7	= 6
Symbol KEY_8	= 7
Symbol KEY_9	= 8
Symbol KEY_0	= 9
Symbol KEY_UP	= 16
Symbol KEY_DOWN	= 17
Symbol KEY_LEFT	= 19
Symbol KEY_RIGHT	= 18
Symbol KEY_BAR	= 96
Symbol KEY_TENT	= 54
Symbol KEY_VERT_CROSS	= 37
Symbol KEY_DIAG_CROSS	= 20
Symbol KEY_MINUS	= 98
Symbol KEY_PLUS	= 11



La lecture du code est effectuée dans un sous-programme (TELEC).

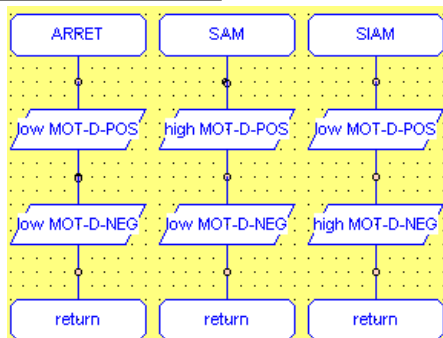
Le code lu est placé dans la variable CODE (b13) et incrémenté.

Si rien n'est reçu au bout de 20 ms, le programme continue à l'instruction RIEN. Le code 0 correspond donc à une non réception de code.

5.4 Pilotage des moteurs

La carte de commande permet de piloter individuellement des moteurs ou de les associer au déplacement d'un véhicule.

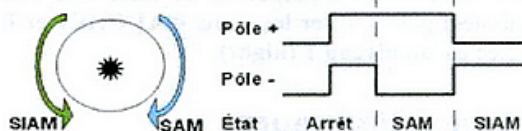
Les moteurs



Différence de tension	Pôle +	Pôle -	État
Pas de différence	Niveau bas	Niveau bas	Arrêt
	Niveau haut	Niveau haut	
Positive	Niveau haut	Niveau bas	SAM
Négative	Niveau bas	Niveau haut	SIAM

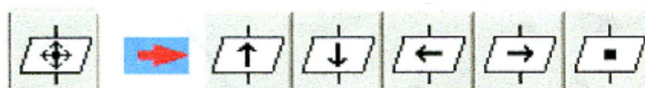
SAM = rotation dans le sens des aiguilles d'une montre

SIAM = rotation dans le sens inverse des aiguilles d'une montre

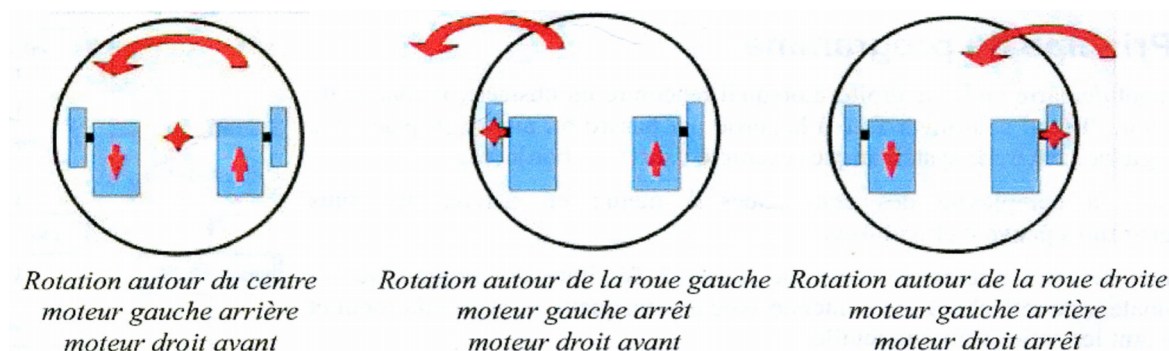


Un moteur à courant continu possède 2 entrées et réagit en fonction du niveau de celles-ci.

Les déplacements



L'utilisation de 2 moteurs implique d'avoir le même sens de rotation des roues pour la marche avant et la marche arrière. En fonction de la disposition des moteurs, la connexion d'un moteur à la carte de commande pourra être inversée pour obtenir le résultat souhaité.

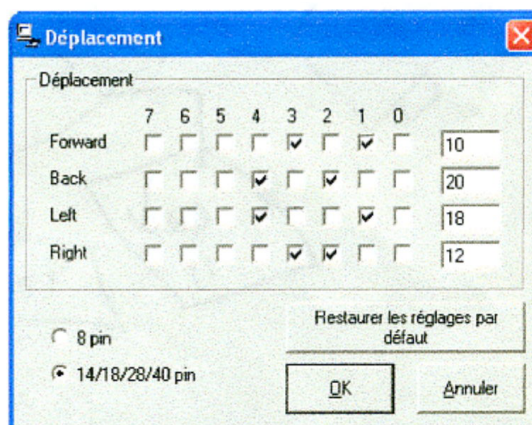


Pour tourner dans une direction, il existe 3 possibilités à choisir en fonction de la stratégie désirée.

Mise en pratique

En fonction de la stratégie choisie et des sorties utilisées pour les moteurs, les différentes combinaisons pour les commandes moteur sont définies dans la boîte « Options – Diagramme - Déplacement ». Une sortie cochée correspond à son activation (état haut) pour la commande correspondante.

ICI, le moteur gauche est connecté en S3 -S4 et le droit en S1-S2. La stratégie choisie est la rotation autour du centre.

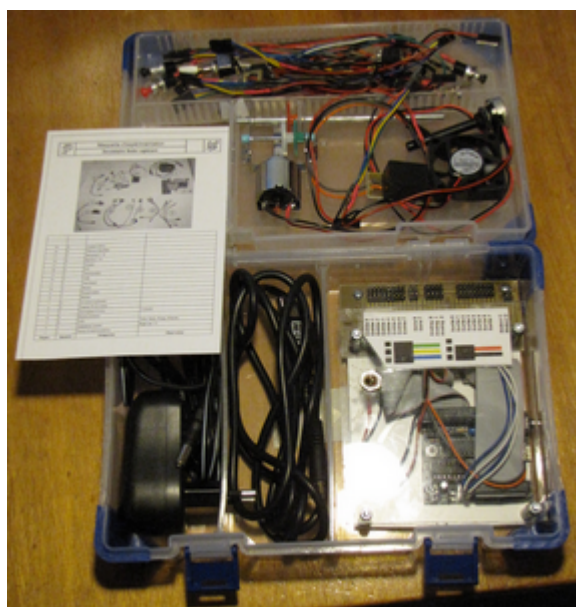
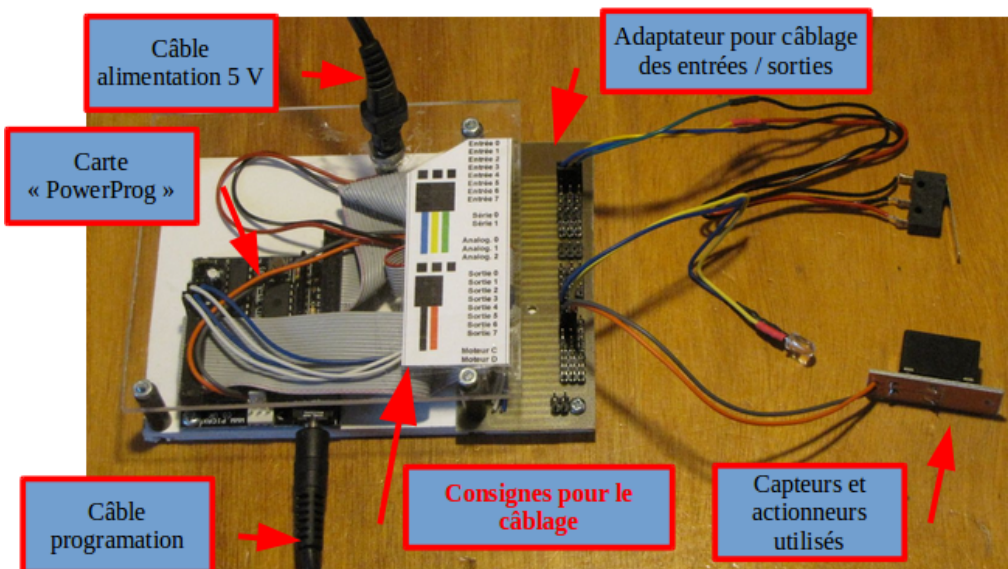


ATTENTION : Le programme s'exécutant très rapidement et les moteurs ayant une inertie, des changements trop fréquents de direction peuvent mener à un échauffement du circuit pilote, de son blocage, voire de sa destruction. Il est donc nécessaire de placer un arrêt et une pause d'au moins 200 ms avant tous changement de direction.

6 Maquette

6.1 La maquette

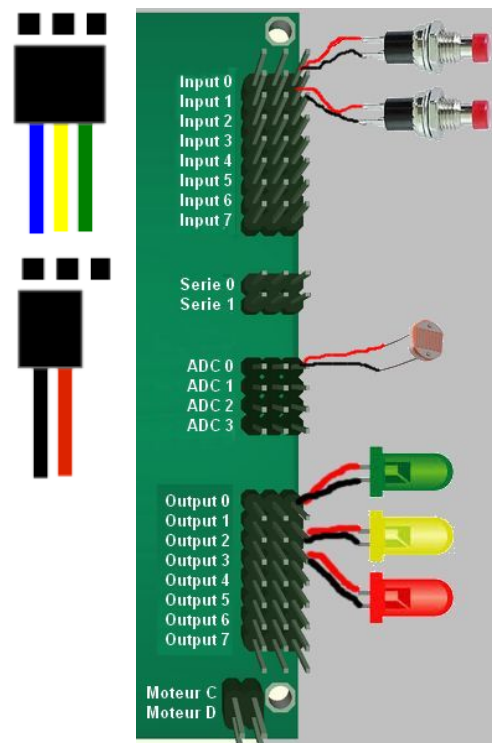
ATTENTION
au sens de branchement
des composants



Valise de rangement

Les sorties Série 0 et 1 correspondent respectivement à Output 0 et 1. Elles sont spécifiques pour les actionneurs travaillant avec une liaison série.

Les sorties Moteur C et D correspondent respectivement à Output 4-5 et Output 6-7



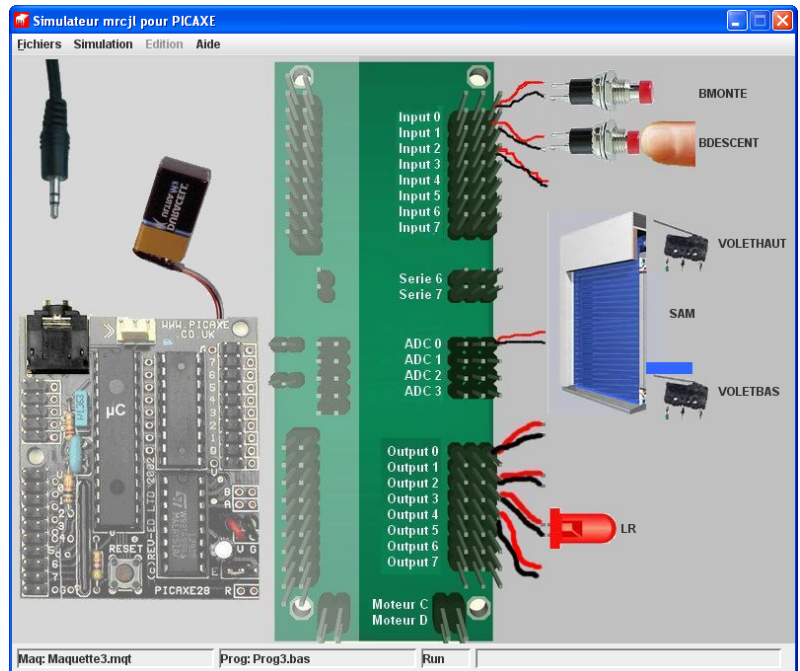
Détail de la platine de connexion et sens de branchement des connecteurs

6.2 Le simulateur de maquette

Les programmes obtenus peuvent être testés sur une maquette physique (mallette PICAXE) ou virtuelle (programme SIMULPICAXE).

Une fois le simulateur lancé, cliquer sur « Fichier - Ouvrir » pour charger la maquette (*,mqt) désirée. S'il ne vous l'a pas déjà indiqué, demandez au professeur les connexions utilisées par les composants afin de faire correspondre celles indiquées dans le programme.

Conversion d'un organigramme



Le programme à tester doit être en version « BASIC » (.bas). Une fois l'organigramme terminé, cliquer sur « Picaxe – Convertir le diagramme en Basic » et enregistrer cette version.

Chargement et test d'un programme

1. Cliquer sur la pile pour la connecter
2. Cliquer sur le Jack pour le connecter.
3. Charger le programme (*,bas) désiré.
4. Tester le fonctionnement du programme.

A tout moment, le programme en test peut être réinitialisé en déconnectant la pile (clic dessus) et la reconnectant.

Modification de l'état d'un capteur

L'état des capteur peut être modifié en cliquant dessus :

- Capteurs « tout ou rien » - clic gauche = appui fugitif, clic droit = appui maintenu
- Capteurs proportionnels – clic gauche = décrétement, clic droit = incrément (avec touche [MAJ] = X 10).