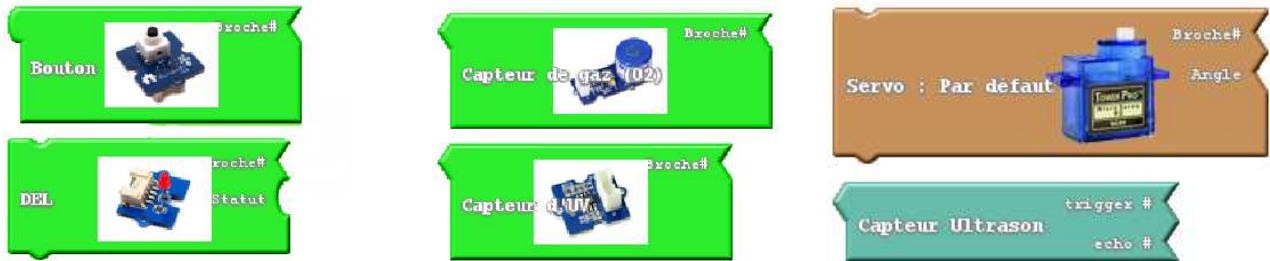




Après avoir réalisé le programme nous devons le convertir dans un langage compréhensible de l'automate. Chaque automate utilise un langage qui lui est propre et qui peut être différent des autres automates.

1) Présentation du langage

Les actions du système et les informations reçues par le système sont représentées par des blocs :



Le programme est écrit à l'intérieur d'un bloc «Boucle » :



2) L'automate ne connaît que des entrées et des sorties et pas les éléments que nous lui avons connecté. Nous allons donc convertir les différents éléments en entrée et en sortie.

Actions :

Actions	Servomoteur gauche		Servomoteur droit	
	Numéro de la sortie	Etat de la sortie (angle)	Numéro de la sortie	Etat de la sortie (angle)
Mettre en route les moteurs, sens : tout droit	D12	200	D13	200
Mettre en route les moteurs, sens : à droite	D12	200	D13	0
Mettre en route les moteurs, sens : à gauche	D12	0	D13	200
Arrêter les moteurs	D12	100	D13	100
Attendre 3 secondes	Pas de sortie utilisée			

Remarque : Les angles de sortie permettant d'arrêter les moteurs doivent être identifiés pour chaque moteur par des tests.

Informations :

Informations	Numéro de l'entrée	Etat de l'entrée
Bouton poussoir appuyé	D2	1



3) A l'aide des tableaux précédents nous convertissons notre programme dans le langage utilisable par l'automate

Actions du système	Informations reçues par le système	
	Bouton poussoir appuyé	Si Bouton poussoir D2 appuyé
Mettre en route les moteurs, sens : tout droit		Alors Servomoteur D12 angle 200 Servomoteur D13 angle 200
Attendre 3 secondes		Délai 3000 ms (millisecondes)
Mettre en route les moteurs, sens : à droite		Servomoteur D12 angle 200 Servomoteur D13 angle 0
Attendre 2 secondes		Délai 2000 ms
Mettre en route les moteurs, sens : tout droit		Servomoteur D12 angle 200 Servomoteur D13 angle 200
Attendre 3 secondes		Délai 3000 ms
Mettre en route les moteurs, sens : à gauche		Servomoteur D12 angle 0 Servomoteur D13 angle 200
Attendre 2 secondes		Délai 2000 ms
Arrêter les moteurs		Servomoteur D12 angle 100 Servomoteur D13 angle 100



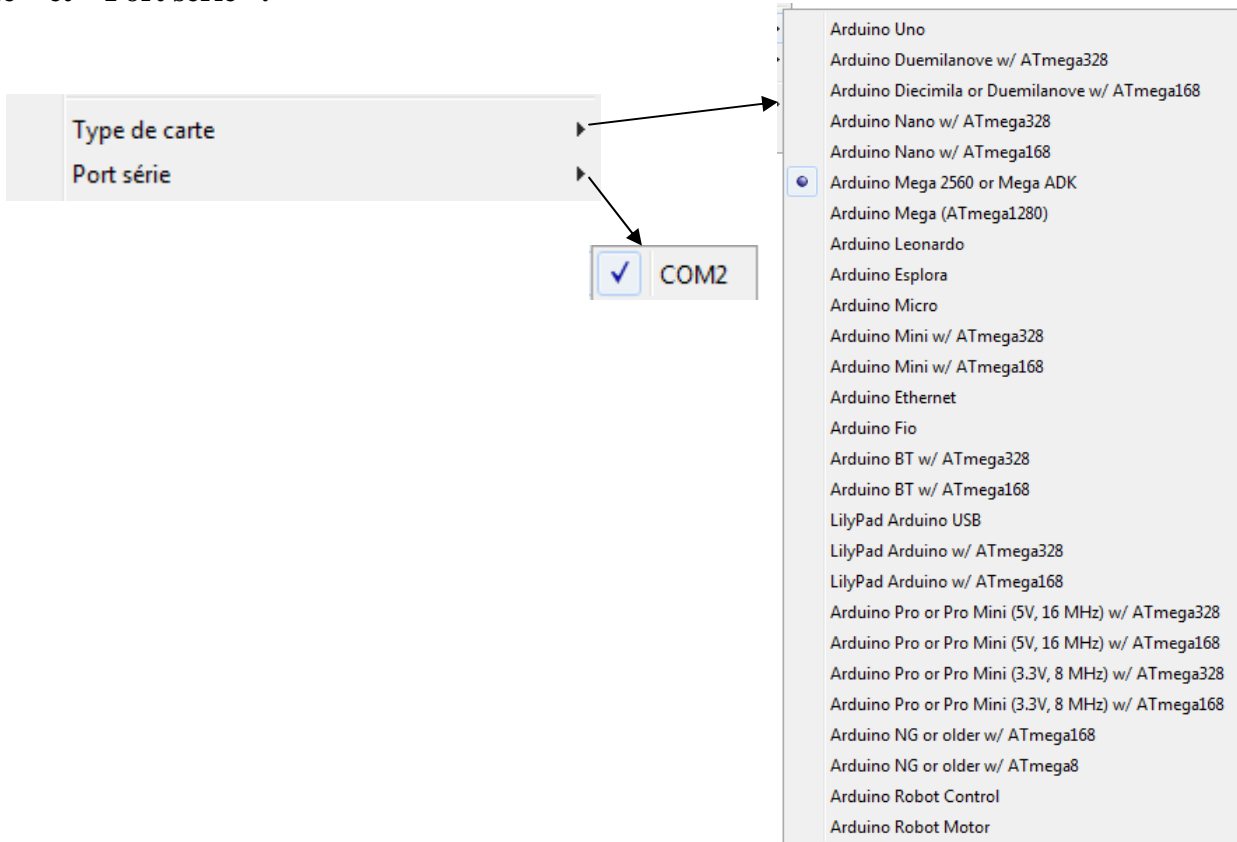
4) Nous pouvons maintenant recopier le programme dans le module « Ardublock » du logiciel « Arduino » pilotant notre carte programmable. (Ardublock se trouve dans le menu « Outils » du logiciel « Arduino »)

The screenshot shows the Ardublock programming environment with two main sections:

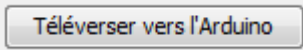
- Left Section (Main Program):** Starts with a 'Bouton' block (pin D2) connected to a 'Teste' block. Below it is an 'Alors exécute' block containing:
 - Servo : Par défaut (Broche# D12, Angle 200)
 - Servo : Par défaut (Broche# D13, Angle 200)
 - delay MILLIS (Millisecondes) 3000
 - Servo : Par défaut (Broche# D12, Angle 200)
 - Servo : Par défaut (Broche# D13, Angle 0)
 - delay MILLIS (Millisecondes) 2000
- Right Section (Loop):** A 'boucle' block labeled 'S1' containing:
 - Servo : Par défaut (Broche# D12, Angle 200)
 - Servo : Par défaut (Broche# D13, Angle 200)
 - delay MILLIS (Millisecondes) 3000
 - Servo : Par défaut (Broche# D12, Angle 0)
 - Servo : Par défaut (Broche# D13, Angle 200)
 - delay MILLIS (Millisecondes) 2000
 - Servo : Par défaut (Broche# D12, Angle 100)
 - Servo : Par défaut (Broche# D13, Angle 100)



5) Nous devons configurer le logiciel pour lui indiquer le port de communication et la carte utilisée. Pour cela dans le menu outil du logiciel « Arduino » nous allons utiliser les commandes « Type de carte » et « Port série ».



6) Nous pouvons maintenant téléverser le programme vers l'Arduino dans le module « Ardublock ».

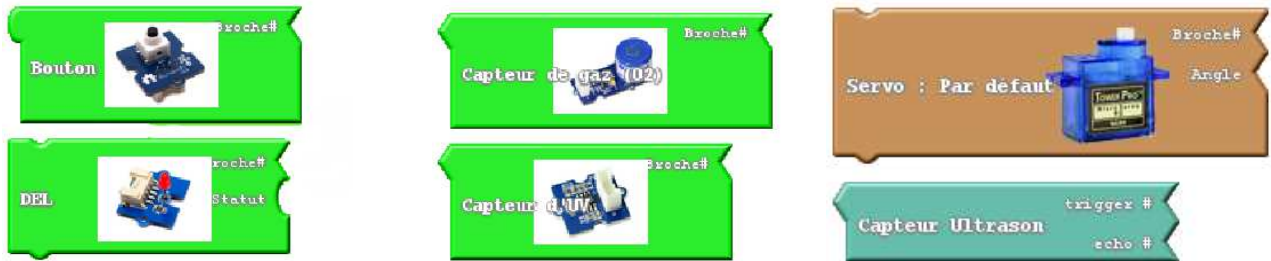




Après avoir réalisé le programme nous devons le convertir dans un langage compréhensible de l'automate. Chaque automate utilise un langage qui lui est propre et qui peut être différent des autres automates.

1) Présentation du langage

Les actions du système et les informations reçues par le système sont représentées par des blocs :



Le programme est écrit à l'intérieur d'un bloc «Boucle » :



2) L'automate ne connaît que des entrées et des sorties et pas les éléments que nous lui avons connectés. Nous allons donc convertir les différents éléments en entrée et en sortie.

Actions :

Actions	Numéro de la sortie
Mettre en route les moteurs, sens : tout droit	Les moteurs sont branchés sur un module moteur. Un seul peut être connecté par Arduino, il n'est donc pas nécessaire ni possible d'indiquer le numéro de la sortie.
Mettre en route les moteurs, sens : à droite	
Mettre en route les moteurs, sens : à gauche	
Arrêter les moteurs	Stop Driver I2C
Attendre 3 secondes	Pas de sortie utilisée

Informations :

Informations	Numéro de l'entrée	Etat de l'entrée
Bouton poussoir appuyé	D2	1



3) A l'aide des tableaux précédents nous convertissons notre programme dans le langage utilisable par l'automate

Actions du système	Informations reçues par le système
	Bouton poussoir appuyé
Mettre en route les moteurs, sens : tout droit	
Attendre 3 secondes	
Mettre en route les moteurs, sens : à droite	
Attendre 2 secondes	
Mettre en route les moteurs, sens : tout droit	
Attendre 3 secondes	
Mettre en route les moteurs, sens : à gauche	
Attendre 2 secondes	
Arrêter les moteurs	

Si	Bouton poussoir D2 appuyé
Alors	Avancer tout droit
	Délai 3000 ms (millisecondes)
	Pivoter droite
	Délai 2000 ms
	Avancer tout droit
	Délai 3000 ms
	Pivoter gauche
	Délai 2000 ms
	Stop Driver I2C

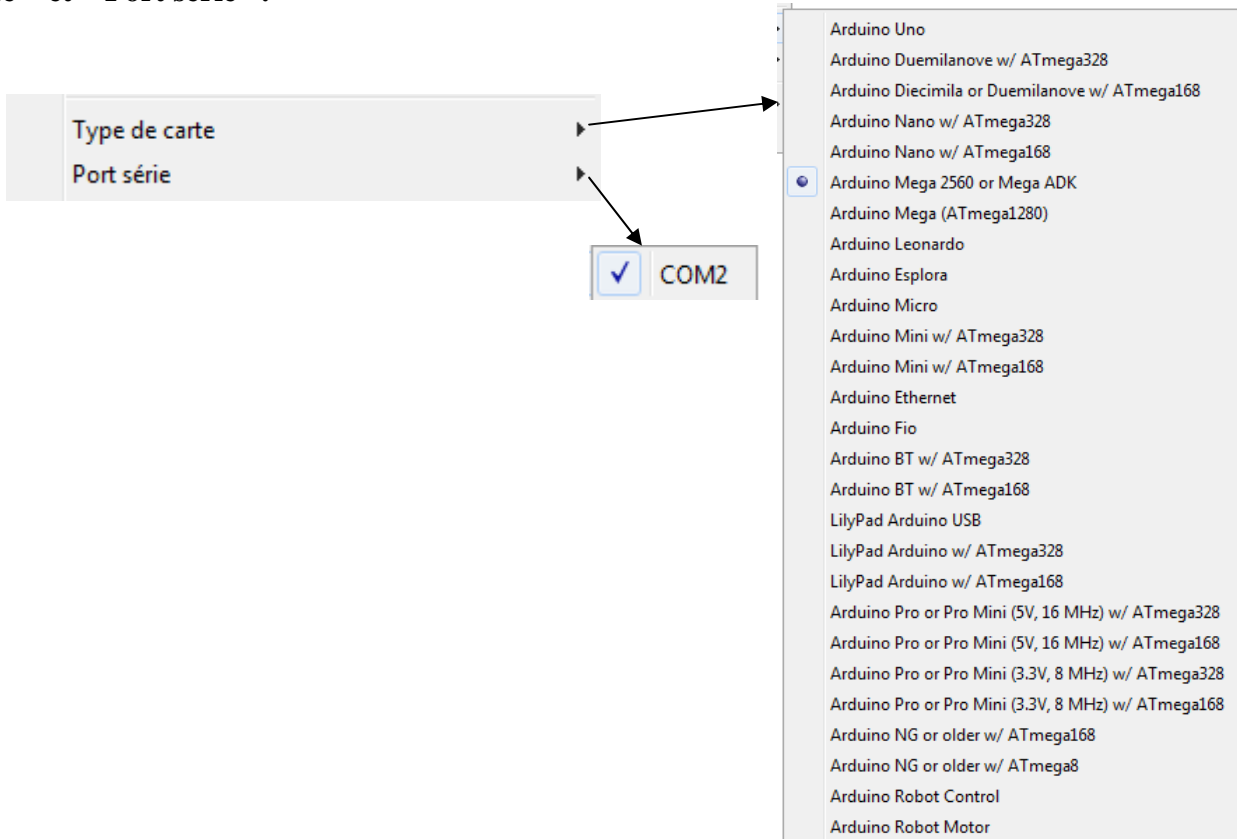


4) Nous pouvons maintenant recopier le programme dans le module « Ardublock » du logiciel « Arduino » pilotant notre carte programmable. (Ardublock se trouve dans le menu « Outils » du logiciel « Arduino »)





5) Nous devons configurer le logiciel pour lui indiquer le port de communication et la carte utilisée. Pour cela dans le menu outil du logiciel « Arduino » nous allons utiliser les commandes « Type de carte » et « Port série ».



6) Nous pouvons maintenant téléverser le programme vers l'Arduino dans le module « Ardublock ».

