

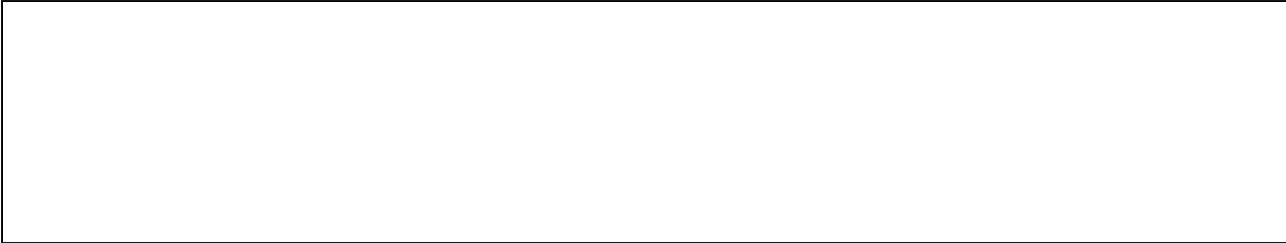
Les automates programmables industriels

1. Problématique

Les équipements de production doivent évoluer au fil des besoins de l'entreprise et de ceux des clients. Les systèmes étudiés jusqu'ici étaient basés sur le câblage de contacteurs, donc peu adaptables au besoin sauf à recâbler, au moins en partie, l'équipement. Les automates programmables industriels permettent de modifier le fonctionnement du système sur lequel ils sont implantés sans qu'il ne soit nécessaire de recâbler. Les modifications de fonctionnement qui ne nécessitent pas d'ajout d'actionneurs, se résument à des modifications de logiciels.

2. Fonction – symbole

L'automate programmable industriel se définit comme suit :



L'automate ne dispose pas de symbole normalisé au sens où ils ont été étudiés jusqu'ici. Cela est dû à la grande diversité des automates et aux fonctionnalités différentes qu'ils peuvent intégrer. La représentation graphique est donc composée comme suit :

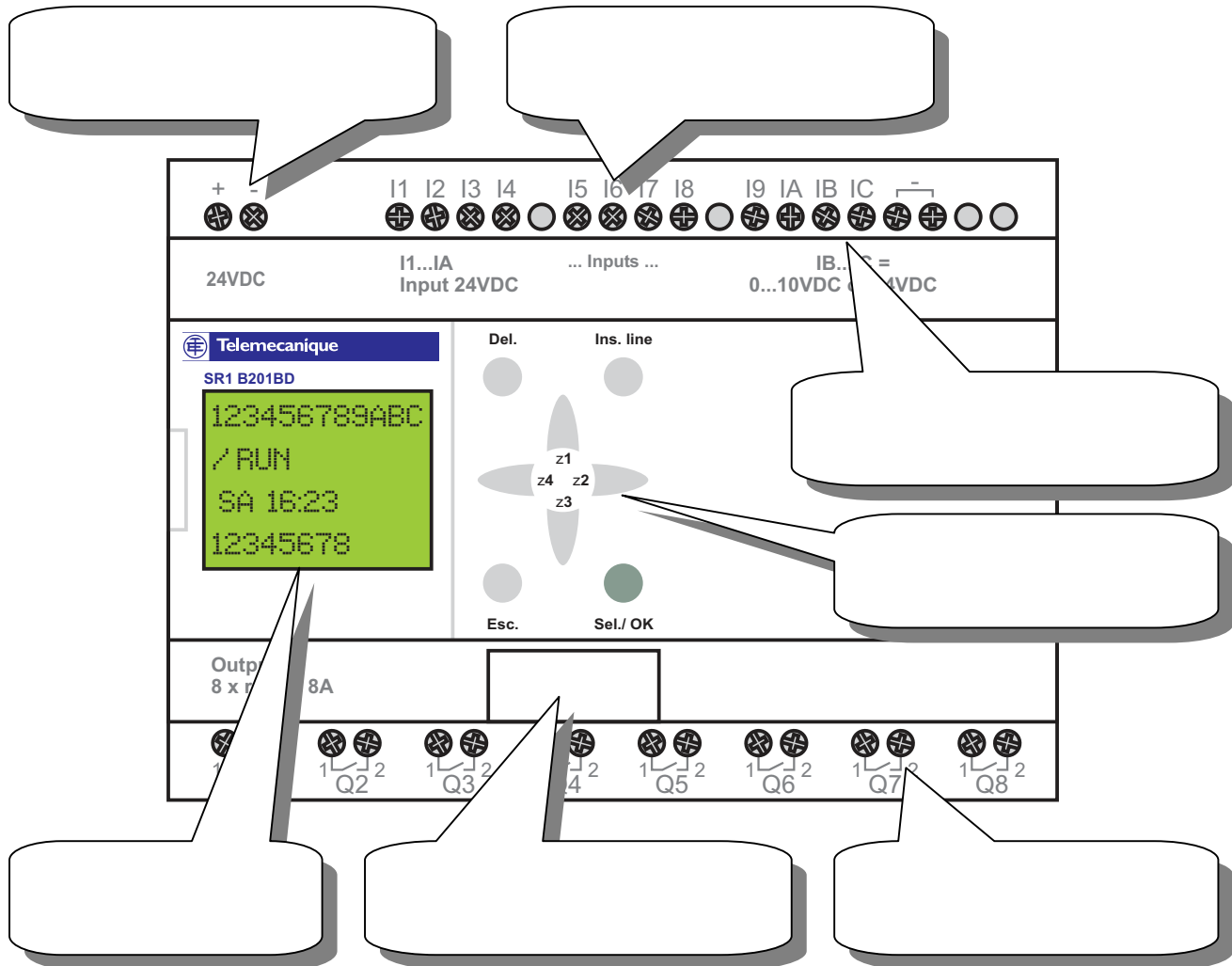


L'automate programmable industriel représenté ci-dessus comporte quatre entrées numériques (I1.1 à I1.4) avec commun (CI1-4), quatre sorties numériques (Q1.1 à Q1.4) avec commun (CQ1-4) et deux entrées analogiques (AI1 et AI2) elles aussi avec un commun (CAI1-2).

3. Constitution

Les automates programmables industriels sont de véritables petits ordinateurs industriels spécialisés pour l'exécution de programmes destinés à commander des pré-actionneurs reliés aux sorties (contacteurs, électrovannes, variateurs de vitesse...) en fonction de l'état des entrées (détecteurs, capteurs, interface homme - machine (IHM)...) et éventuellement de l'instant auquel s'effectuent ces changements (systèmes séquentiels). Ils comprennent un microprocesseur, de la

mémoire vive (mémoire de travail), de la mémoire « morte » contenant le système d'exploitation, de la mémoire programme qui contient les « règles » liant l'évolution des sorties en fonction de l'état des entrées, des interfaces d'entrée et de sortie TOR et éventuellement des entrées / sorties analogiques. Des fonctions diverses telles que des interfaces pour sondes de températures, entrées – sorties en courant (4 – 20 mA ou autre) ou autres existent aussi.



L'automate repris ci-dessus comporte des entrées qui sont « liées » à son alimentation, les sorties sont dites à « contact sec », c'est à dire qu'elles ne sont liées à aucun potentiel. L'API repris sur le paragraphe fonction - symbole disposait d'entrées qui n'étaient pas liées à l'alimentation de l'automate mais à une borne appelée « commun » des entrées (C11). De même, les sorties étaient toutes reliées à un « commun » des sorties (CQ1). Cela démontre la grande diversité des configurations permises avec les automates.

4. Critères de choix

Le choix d'un automate programmable industriel se fait en fonction des critères suivants :

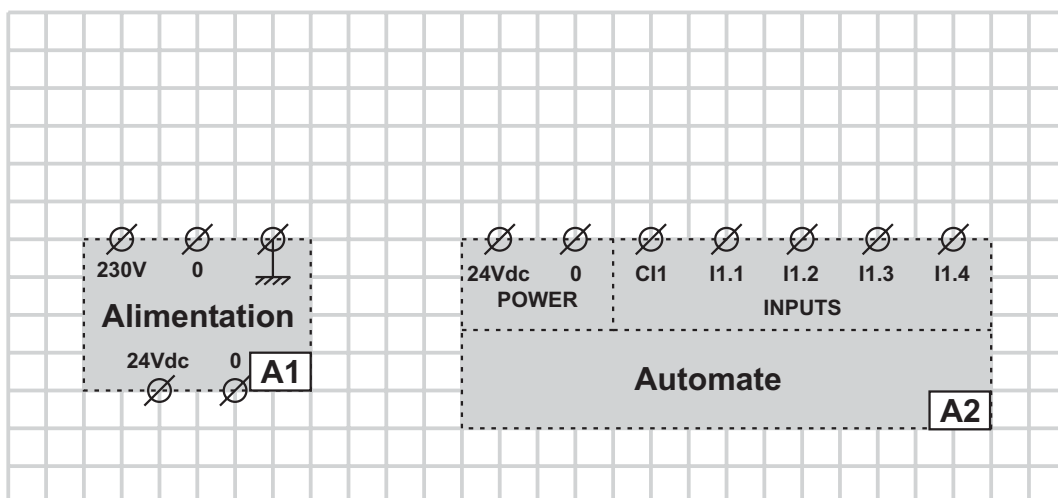
Les automates fabriqués aujourd'hui sont extrêmement fiables. Toutefois, comme pour les ordinateurs individuels, ils ne sont pas à l'abri d'un « plantage ». C'est pourquoi, les sécurités électriques seront câblées matériellement (arrêts d'urgence, verrouillages électriques...) afin de rester opérantes même en cas de plantage de l'automate.

5. Raccordement électrique

Le raccordement d'un automate programmable industriel se fait en fonction de ses caractéristiques (présence de communs en entrées, en sorties...).

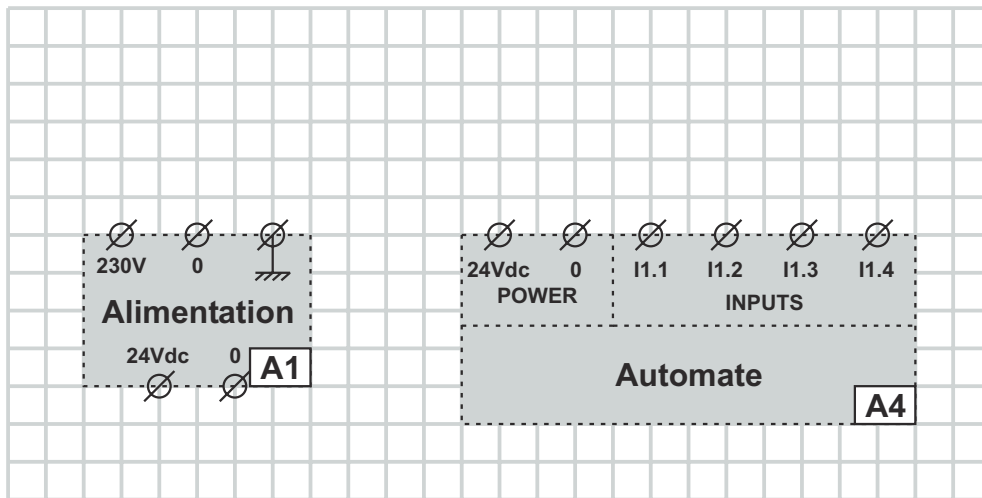
5.1. Entrées avec commun

Complétez le schéma suivant en reliant un bouton poussoir *NO* repéré *S3* sur l'entrée *I1.2* et un contact *NC* du disjoncteur *D1* sur l'entrée *I1.4* de l'automate alimenté en 24 Vdc. Les sorties ne sont pas représentées à ce niveau, elles peuvent l'être sur un autre folio du schéma. L'automate aura alors le même repère sur les différents folios sur lesquels il figure (ici *A2*). Le calibre du disjoncteur de protection de l'alimentation (D2) sera choisi conformément aux recommandations du fabricant de l'automate.

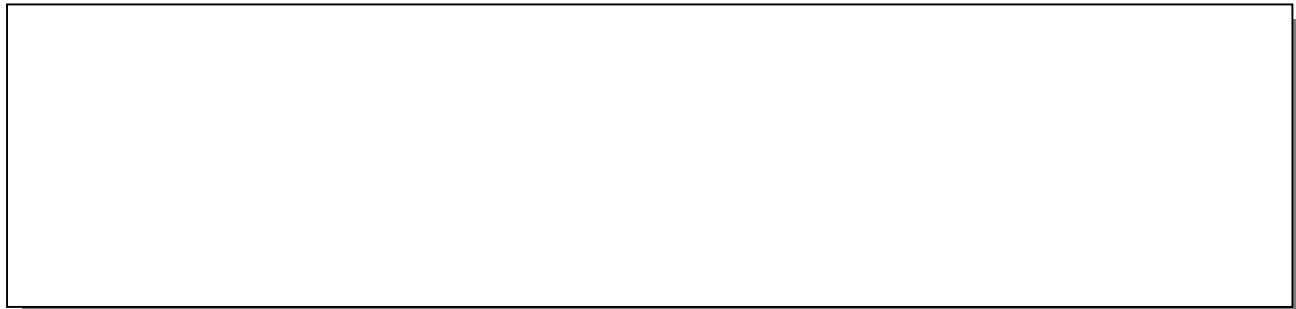


5.2. Entrées sans commun

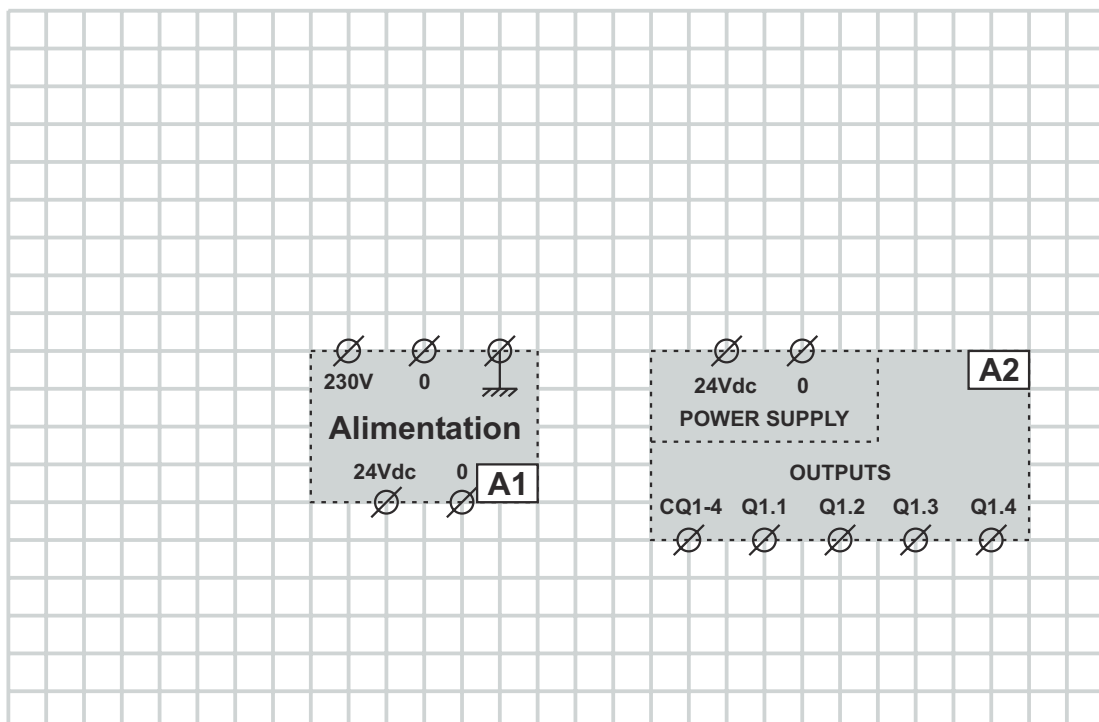
Complétez le schéma ci-dessous en reliant le contact *NC* d'un disjoncteur repéré *D1* sur l'entrée *I1.1* et le contact *NC* d'un relais thermique *F2* sur l'entrée *I1.3* de l'automate *A4* alimenté en 24 Vdc.



5.3. Sorties sur contacts secs avec commun

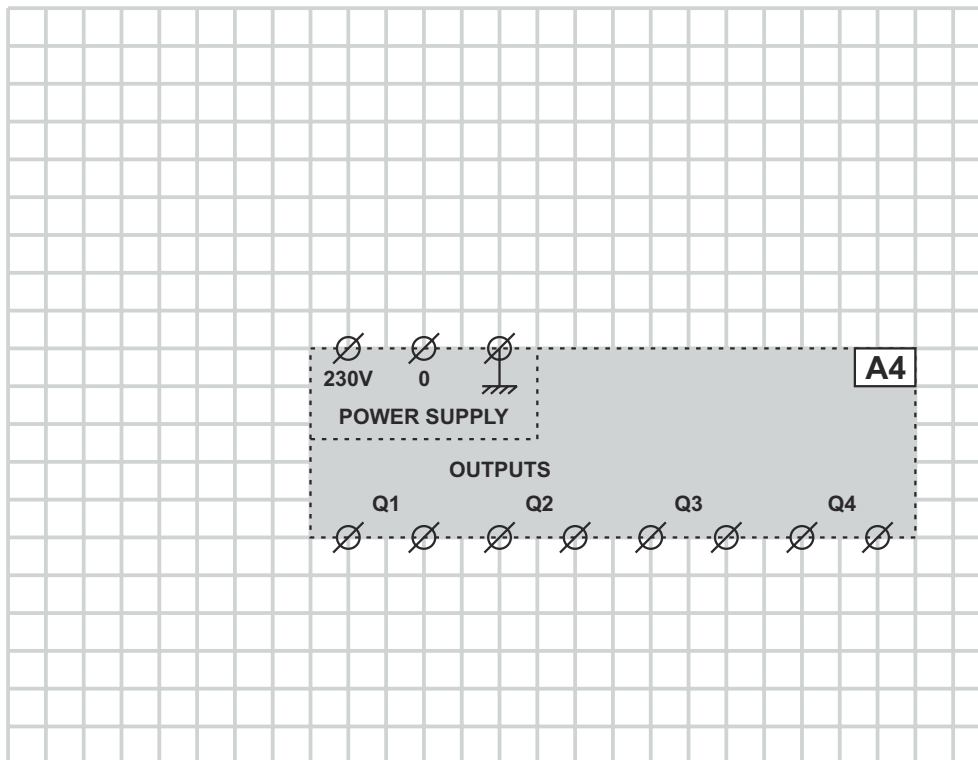


Complétez le schéma ci-dessous en reliant une bobine de contacteur *KM1* sur la sortie *Q1.2* et le voyant *VI* la sortie *Q1.4* (les entrées ne seront pas représentées, elles peuvent l'être sur un autre folio du schéma). La bobine et le voyant seront alimentés à partir d'un transformateur monophasé (*T1*) 230 / 24 Vac protégé par un disjoncteur *D1* au primaire et *D3* au secondaire.



5.4. Sorties sur contacts secs sans commun

Complétez le schéma ci-dessous en reliant une bobine de contacteur **KM1** sur la sortie **Q1**, le voyant **V1** la sortie **Q3** et le voyant **V2** la sortie **Q4**. La bobine et les voyants seront alimentés à partir d'un transformateur monophasé (**TI**) 230 / 24 Vac protégé par un disjoncteur **D1** au primaire et **D3** au secondaire, l'automate est alimenté sous 230 Vac via le disjoncteur **D2**.



5.5. Sorties statiques

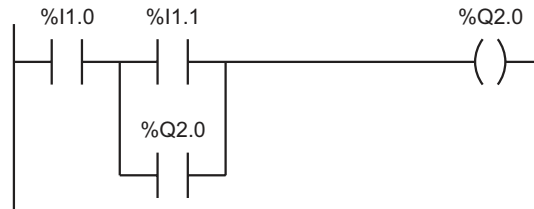
Les sorties statiques n'utilisent pas de relais mais des transistors ou autre composant électronique adapté. Leur raccordement n'est pas plus compliqué, mais il est ici important de respecter les polarités de l'alimentation. L'avantage des sorties statiques vient du fait qu'il n'y a pas d'usure mécanique car il n'y a aucune pièce en mouvement au niveau de l'étage de sortie. Pour remédier à cette usure, certains automates à sorties sur relais disposent de relais interchangeable en sortie.

6. Programmation

La programmation des automates se fait soit par l'intermédiaire d'une « console » (le plus souvent un PC) ou d'un afficheur et d'un clavier intégrés sur l'automate. La richesse des langages de programmation est liée aux automates eux-mêmes. On retrouve le plus souvent le Ladder (**LD**, Ladder Diagram) et la liste d'instruction (**IL**, Instruction List) qui ont été normalisés et se retrouvent chez la plupart des constructeurs d'automates. On retrouve aussi des langages qui se rapprochent du GRAFCET (**SFC**, Sequential Function Chart), des blocs logiques (**FBD**, Function Bloc Diagram) ou se rapprochant des langages informatiques tels que le **C** (**ST**, Structured Text). Tous les automates ne reprennent pas la totalité des langages, c'est l'un des critères de choix.

6.1. Le langage « Ladder » (LD)

Le langage Ladder est aussi appelé « langage à contacts ». On lit l'état électrique de l'entrée (sous tension, hors tension), on effectue des opérations logiques avec l'état des différentes entrées puis on affecte le résultat logique aux sorties de l'automate. On retrouve des bobines particulières telles que les bobines à accrochage et à décrochage (fonctions **SET** et **RESET**), des fonctions horloges, temporisations, compteurs, comparateurs... Se référer à la documentation technique de l'automate utilisé pour en avoir une description détaillée.



Exemple de programme en langage LD.

Le programme ci-dessus lit l'état des entrées. Il effectue un ET logique entre l'entrée %I1.0 et le résultat du OU logique entre l'état de l'entrée %I1.1 et l'état de la sortie %Q2.0. Le résultat de ces opérations logiques est ensuite affecté à la sortie %Q2.0. Ce programme effectue l'équivalent d'un démarrage direct tel que celui étudié dans le cours « démarrage direct des moteurs ».

6.2. Le langage « liste d'instructions » (IL)

Le langage liste d'instructions se rapproche du langage machine. On décrit le fonctionnement du système à l'aide d'opérations logiques écrites selon une syntaxe particulière.

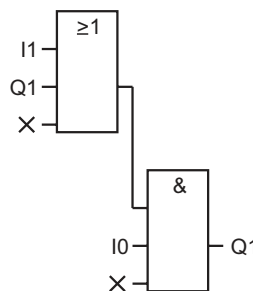
```
LD    %I1.0
AND(  %I1.1
OR    %Q2.0
)
ST    %Q2.0
```

Exemple de programme en langage IL.

Le programme ci-dessus est la traduction de celui du paragraphe précédent en langage **LD**, il a exactement les mêmes fonctionnalités.

6.3. La programmation à partir de portes logiques (FBD)

Le langage FBD est la traduction en portes logiques (ET, OU...) du fonctionnement d'un système en logique combinatoire (l'état des sorties est lié uniquement à la combinaison des entrées et ne dépend pas de la chronologie entre celles-ci).



Exemple de programme en langage FBD.

Le programme ci-dessus est la traduction de celui du paragraphe sur le langage **LD**. La différence principale vient du repérage des entrées qui s'effectue selon une autre « notation » que pour les paragraphes précédents. Les notations sont fonction du choix du fabricant de l'automate. Il est parfois possible de démarrer la programmation en Ladder, de basculer en **LI** puis en **FBD** et

inversement. La traduction s'effectue automatiquement sauf pour certaines portions de programme qui restent alors dans le langage d'origine car elles n'ont pas d'équivalent dans le langage de destination.

6.4. La programmation GRAFCET (SFC)

La programmation se fait par une représentation graphique similaire au GRAFCET. On commence par représenter les étapes, on programme les transitions sous forme **LD**, **LI** ou **FBD** puis on affecte les sorties sous l'une des trois formes précédentes. Tous les automates ne sont pas compatibles avec la programmation **SFC**. Il est possible de programmer un GRAFCET en langage **LD** ou **LI** à l'aide de variables internes, mais la chose est beaucoup moins aisée.

6.5. Le langage « texte structuré » (ST)

On se rapproche ici beaucoup de la programmation des ordinateurs individuels, type langage C. La puissance de ce langage est sans commune mesure par rapport au précédents, mais sa complexité le réserve à des spécialistes de l'informatique industrielle.

7. Application

Nous allons repartir de l'application développée dans le cours « démarrage direct, deux sens de rotation » qui avait pour but de faire fonctionner le malaxeur. Le schéma de puissance ne change pas, seul le schéma de commande est modifié. Le fonctionnement n'est pas modifié et répond au cahier des charges suivant :

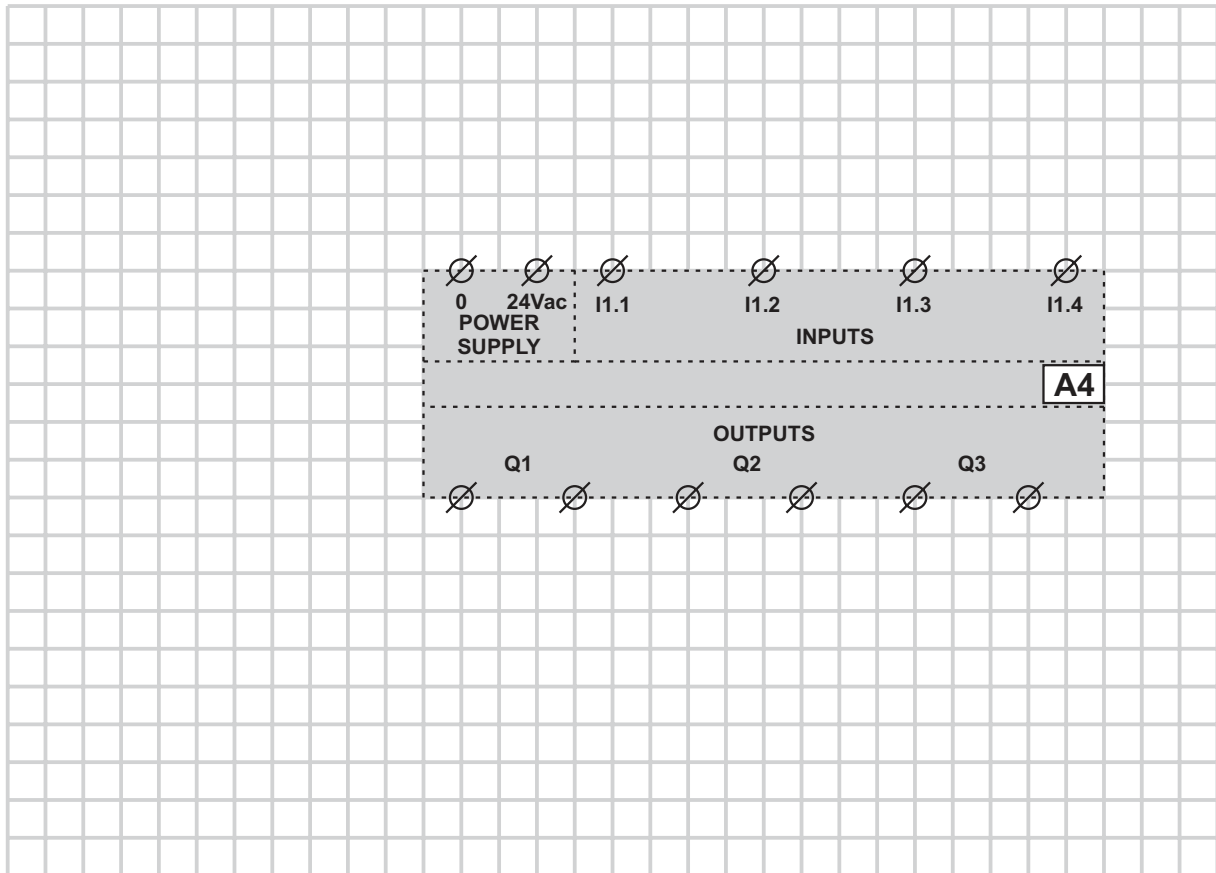
- € La mise sous tension du moteur dans le sens direct se fait par l'appui sur le bouton poussoir **S2**,
- € La mise sous tension du moteur dans le sens de rotation inverse se fait par l'appui sur le bouton poussoir **S3**,
- € L'arrêt du moteur, quel que soit son sens de rotation, se fait par l'appui sur le « coup de poing » **SI**,
- € Suite à une panne de courant, lors du retour de l'alimentation, le moteur ne doit pas redémarrer seul,
- € La fonction arrêt est prioritaire sur la fonction marche (l'appui simultané sur les boutons poussoirs de mise en marche (**S2** ou **S3**) et d'arrêt (**SI**) ne doit pas mettre le moteur sous tension,
- € Un voyant **HI** permettra de visualiser le déclenchement du relais thermique **F1**.

Nous allons dans un premier temps choisir l'automate dans la famille « Zelio Logic » de chez Schneider Electric. Ce dernier sera alimenté sous 24 V alternatifs et doit disposer d'un afficheur et d'un clavier. Donnez la référence de l'automate retenu ainsi que les critères retenus pour son choix.

La mise en œuvre d'un automate programmable industriel ne modifie en rien le schéma de puissance utilisé. Seul le schéma de commande est modifié.

Etant donné la simplicité de l'application étudiée, un automate programmable ne se justifierait pas ici. Cette application a été retenue afin de se concentrer sur le raccordement des entrées - sorties de l'automate.

L'automate retenu dispose d'entrées et de sorties sans commun. Afin de ne pas surcharger inutilement le schéma, seules les entrées et sorties utilisées par le système seront représentées. Complétez le schéma de commande suivant en respectant les règles vues précédemment.



Le programme entré dans l'automate ne sera pas étudié ici.