

TP12 PILOTAGE D'UN SYSTÈME DE TRI PAR MICRO-CONTRÔLEUR EN LADDER ET LANGAGE ST PAR LIBRAIRIE « MACHINE À ÉTATS »

DESCRIPTION DU CONTEXTE PROFESSIONNEL :

Les secteurs professionnels : Les bâtiments (résidentiel, tertiaire et industriel), L'industrie (distribution, transport et gestion de l'énergie liés aux procédés : efficacité énergétique, conversions d'énergie, régulations et modulations d'énergie, etc.)

La nature des travaux : neuf, extension ; rénovation, adaptation, amélioration, optimisation, sécurisation; maintenance.

DESCRIPTION DES ACTIVITÉS PROFESSIONNELLES

Pôle « Conception - étude préliminaire »

Activité 1 : conception - étude préliminaire

T 1.1 : analyser et/ou élaborer les documents relatifs aux besoins du client/utilisateur

T 1.2 : élaborer un avant-projet/chantier (ou avant-projet sommaire)

T 1.3 : dimensionner les constituants de l'installation

Pôle « Conception - étude détaillée du projet »

Activité 2 : conception - étude détaillée du projet

T 2.1 : choisir les matériels

T 2.2 : réaliser les documents techniques du projet/chantier

Pôle « Analyse, diagnostic, maintenance »

Activité 4 : maintenance d'une installation électrique

T 4.3 : réaliser la maintenance corrective

Pôle « Réalisation, mise en service d'un projet »

Activité 6 : réalisation : installation – intégration

T 6.1 : organiser l'espace de travail

T 6.2 : implanter, poser, installer, câbler, raccorder les matériels électriques

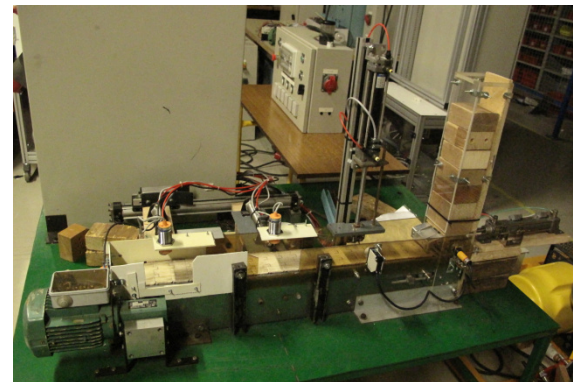
T 6.3 : programmer les applications métiers

Activité 7 : mise en service

T 7.1 : réaliser les contrôles, les configurations, les essais fonctionnels

T 7.2 : vérifier le fonctionnement de l'installation

T 7.3 : réceptionner l'installation avec le client/utilisateur



Connaissances et notions abordées :

Notion Chaine de puissance :

NATURE	LIMITE DE CONNAISSANCE
Distribution du point de livraison à la sortie du TGBT : - installations électriques BT ; - protection des personnes et des biens ; - contexte normatif et réglementaire.	Choisir⁴ des matériels de distribution et de protection Adapter⁴ les paramétrages des dispositifs de protection des personnes et des biens Argumenter⁴ une solution de distribution
Gestion et performance énergétique : - gestion automatique de la consommation d'énergie ; - optimisation de la consommation d'énergie ;	Adapter⁴ les paramétrages des matériels relatifs à la qualité, à la gestion et au comptage de l'énergie Argumenter⁴ une solution de gestion, d'optimisation de la consommation
Fonctions d'usage : - commande, modulation et conversion de l'énergie électrique dans les applications de : - l'électromécanique ; - pompage ; - traitement de l'air ;	Argumenter⁴ une solution de conversion, de modulation et de commande Dimensionner et choisir⁴ des convertisseurs, des modulateurs et des appareils de commande Adapter⁴ les paramétrages des dispositifs de conversion, de modulation et de commande Concevoir/adapter⁴ des schémas électriques de raccordement des dispositifs de conversion, de modulation et de commande

Notion Chaine d'information :

NATURE	LIMITE DE CONNAISSANCE
Traitement de l'information : - automatismes des bâtiments résidentiels, tertiaires et industriels ; - automatismes de l'industrie ; - langages de programmation normalisés	Appliquer³ et faire appliquer³ des normes et règlements

TRI DE BRIQUE MICROCONTROLEUR

(IEC61131-3, autres langages) ;

- organisation et structure d'un programme ;

Acquisition de l'information :

- capteurs y compris intelligents/connectés ;

- détecteurs.

Déterminer3 la nature de l'information (Tout ou rien, numérique, analogique)
Choisir4 des capteurs et des détecteurs
Réaliser3 les paramétrages, les réglages d'un capteur, d'un détecteur

Notion de ressources et outils professionnels :

NATURE	LIMITE DE CONNAISSANCE
Normes, règlements, certifications, labels, marques (hors habilitation électrique, voir référentiel de formation à la prévention des risques d'origine électrique)	Modifier3 un programme d'automatismes des bâtiments résidentiels, tertiaires, industriels ou de l'industrie Concevoir4 un sous-programme/une application pour ajouter une fonction Adapter4 les paramétrages des applications métiers (notamment de régulation)
Langages de description : - fonctionnelle ; - structurelle ; - temporelle.	Utiliser3 les outils de description, fonctionnelle, structurelle, temporelle Mettre en évidence3 un comportement attendu ou observé
Règles de l'art : - gestes du métier d'électricien ; - contraintes d'installation ;	Appliquer3 les règles d'installation et/ou de conception Mettre en pratique3 des procédures

Communication :

NATURE	LIMITE DE CONNAISSANCE
<i>Au-delà et en relation avec le programme de culture générale et expression, d'anglais</i> Communication orale et écrite : - techniques de communication écrite, orale adaptées au contexte professionnel. Vocabulaire technique et communication en langue anglaise	Appliquer3 les techniques de communication orale pour communiquer avec un autre intervenant et/ou le client/utilisateur Appliquer3 les principes et les techniques des écrits professionnels pour communiquer avec un autre intervenant et/ou le client/utilisateur Communiquer4 , tant à l'écrit qu'à l'oral, sur le plan technique en langue anglaise avec un autre intervenant et/ou le client/utilisateur

Objectifs :

Analyse du système

Réalisation des schémas de commande et de puissance

Câblage et mise en place d'une solution technique

Recherche documentaire sur quelques points techniques à éclaircir

Renseigner et compléter l'énoncé ou dossier technique fourni

Faire la démonstration

Donner une culture technique et une information sur le matériel existant afin d'élaborer des solutions pour l'épreuve de projet.

Conditions d'étude et réalisation :

Temps imparti : de 4 à 6 heures suivant la rapidité des étudiants

Travail individuel ou en binôme

L'énoncé, faisant office de dossier technique et de compte rendu, devra être complété.

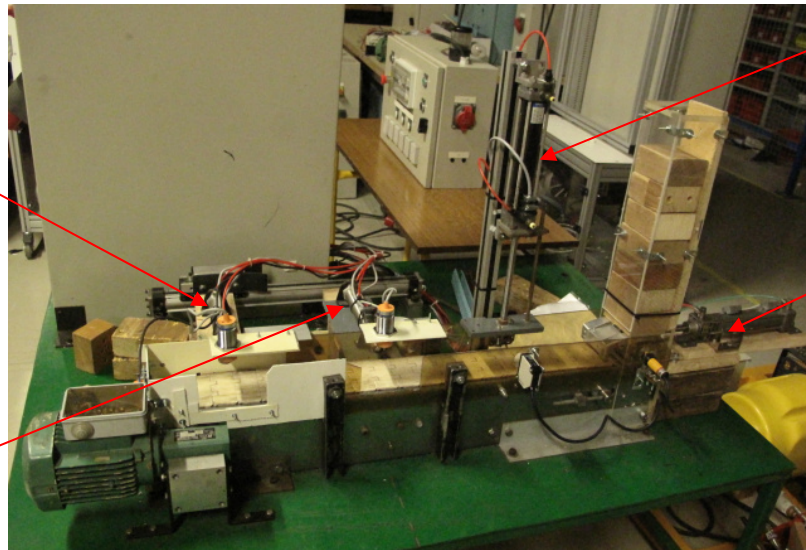
Une démonstration de fonctionnement sera présentée au professeur et devra répondre à la problématique.

DESCRIPTION DES COMPÉTENCES PROFESSIONNELLES

Principale tâche mobilisant la compétence		Critère d'observation de la compétence	nan 0 1 2 3				
Pôles « Conception - étude préliminaire » « Conception - étude détaillée du projet » « Analyse, diagnostic, maintenance » « Réalisation, installation - intégration » « mise en service d'un projet »							
C5	T 1.1 : analyser et/ou élaborer les documents relatifs aux besoins du client/utilisateur	Les besoins et les attentes du client/utilisateur sont identifiés, recensés et reformulés					
	T 1.2 : élaborer un avant-projet/chantier (ou avant-projet sommaire)	Le flux d'énergie et les transformations sont déterminés					
	T 1.3 : dimensionner les constituants de l'installation	Les solutions techniques proposées respectent les spécifications du client/utilisateur, les contraintes normatives					
C8	T 1.2 : élaborer un avant-projet/chantier (ou avant-projet sommaire)	Les solutions techniques de l'avant-projet développées sont pertinentes					
	T 1.3 : dimensionner les constituants de l'installation	Les solutions techniques respectent les spécifications du client/utilisateur, les contraintes normatives et réglementaires					
C9	T 2.1 : choisir les matériels	Les matériels choisis respectent des contraintes normatives et réglementaires et le cahier des charges du client /utilisateur La nomenclature des matériels établie est complète et exacte					
C10	T 1.2 : élaborer un avant-projet/chantier (ou avant-projet sommaire)	Les solutions techniques proposées respectent les spécifications du client/utilisateur, les contraintes normatives et réglementaires L'architecture proposée respecte les spécifications du client/utilisateur, les contraintes normatives et réglementaires Les supports de présentation utilisés sont adaptés					
C11	T 2.2 : réaliser les documents techniques du projet/chantier	Les documents de conception du projet/chantier (architecture, schémas, DOE, notes de calcul, etc.) sont établis, actualisés et archivés					
C14	T 6.1 : organiser l'espace de travail	Les risques professionnels sont identifiés					
	T 6.2 : implanter, poser, installer, câbler, raccorder les matériels électriques	L'espace de travail est approvisionné en matériels, équipements et outillages					
		Les contraintes de réalisation sont repérées					
		Les adaptations nécessaires sont déterminées					
		Les matériels électriques, les canalisations et les supports sont posés dans le respect des prescriptions et des règles de l'art					
		Les matériels électriques sont raccordés					
		Les contrôles associés sont effectués					
Les fiches d'autocontrôles sont complétées							
C15	T 6.3 : programmer les applications métiers	Les programmes sont téléchargés					
	T 7.1 : réaliser les contrôles, les configurations, les essais fonctionnels T 7.2 : vérifier le fonctionnement de l'installation	Le programme est modifié, adapté pour répondre aux attentes du client/utilisateur					
		Les matériels sont configurés et/ou interconnectés					
		L'interopérabilité des matériels est réalisée					
		Les programmes permettent d'atteindre les exigences attendues					
		Les associations, l'interopérabilité des matériels sont validées					
		Les essais sont réalisés afin de valider le fonctionnement de l'installation par rapport aux prescriptions					
Les réglages et paramétrages complémentaires sont réalisés							
C16	T 7.1 : réaliser les contrôles, les configurations, les essais fonctionnels T 7.2 : vérifier le fonctionnement de l'installation	Les conditions de la mise en service sont prises en compte					
		Les contrôles normatifs, réglementaires et spécifiques aux prescriptions sont réalisés					
		Les fiches de contrôles sont complétées					
		Les associations et l'interopérabilité des matériels sont validées					
		Les réglages et paramétrages sont validés					
		Les performances de l'installation sont mesurées					
		Le fonctionnement de l'installation est vérifié par rapport aux prescriptions					
La qualification de l'installation respecte les contraintes normatives et réglementaires							
C18	T 4.3 : réaliser la maintenance corrective	Le dysfonctionnement est diagnostiqué					
		Les opérations de dépannage sont réalisées					
		Les essais associés sont effectués					
		Le fonctionnement de l'installation est vérifié par rapport aux prescriptions					
Fiche d'aptitudes professionnelles et de comportement							
Indices de performance	AP1: faire preuve de rigueur et de précision						
	AP2: faire preuve d'esprit d'équipe						
	AP3: faire preuve de curiosité et d'écoute						
	AP4: faire preuve d'initiative						
	AP5: faire preuve d'analyse critique						
	Investissement et participation						
	Gestion et utilisation du matériel						
	Gestion et tenue de l'espace de travail						
	Respect des procédures et règlements intérieur						
	Intervention de l'enseignant/autonomie						

Problématique : mettre en service un système de tri de brique avec un micro-contrôleur : une solution « low cost » avec ses avantages et ses inconvénients.

1- Présentation sommaire :



Vérin éjecteur 2

Vérin de poinçonnage

Vérin éjecteur 1

Vérin Avance Brique

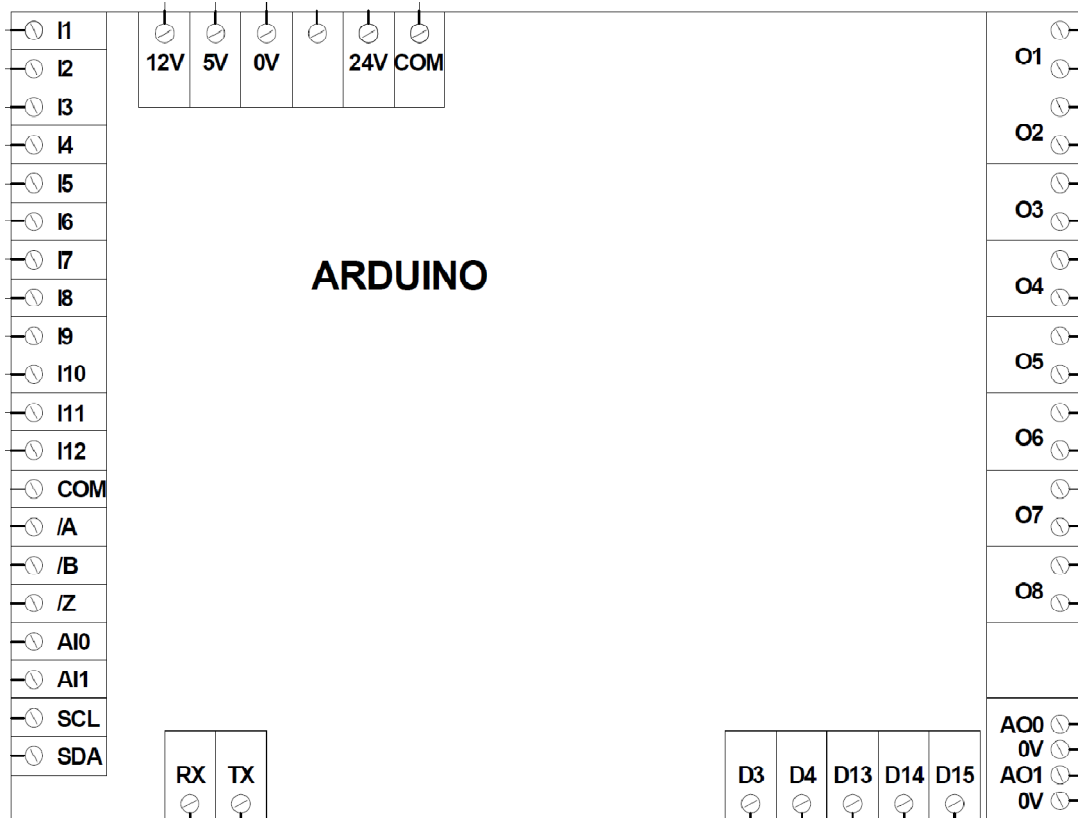
Ce système électropneumatique comporte:

- un tapis convoyeur entraîné par un moto-réducteur asynchrone et variateur de vitesse en U/F
- un magasin de stockage des briques débité par vérin double effet avec électrovannes bistables
- un vérin double effet avec électrovanne monostable pour le poinçonnage ou le marquage
- 2 vérins double effet avec électrovanne monostable d'éjection des briques pour les trier

La carte micro contrôleur (fonctionnant en 0/5V/12V) est associée à une interface comportant :

- 12entrées digitales (24V DC max) pour recueillir les informations des capteurs et divers boutons
- 8 sorties à relais NO (sorties sèches) pouvant piloter jusqu'à 230V/5A.
- 2 sortie analogique 0/10V

Cette carte dénommée Arduino comporte un micro-contrôleur de type Atmega1284P (40 pattes).

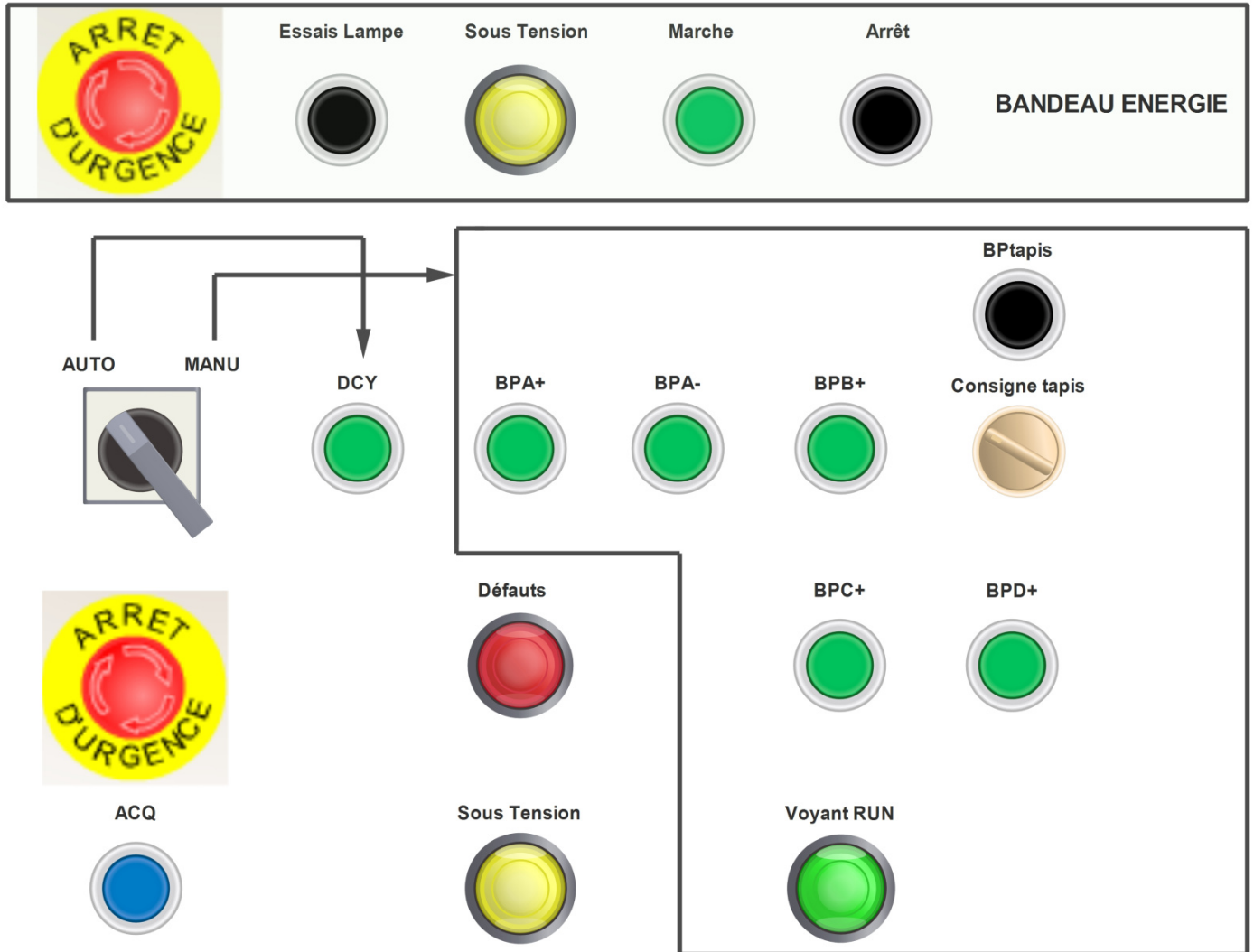


TRI DE BRIQUE MICROCONTROLEUR

Le tableau de correspondances des entrées et sorties est le suivant :

Numéro des bornes de la carte interface	Capteurs/Pré-actionneurs/ Boutons associés	Dénomination des pattes de l'Atmega1284P	Dénomination sous Arduino IDE
I1	Contact KAU NO	PD2 (pin16)	D10
I2	Contact /KA1-KA1 Commutateur AUTO-MANU	PD3 (pin17)	D11
I3	Bouton poussoir DCY	PC3 (pin25)	D19
I4	Capteur b1 tige vérin poïçonneur sortie	PC4 (pin26)	D20
I8	Capteur c1 tige vérin Ejection 1 sortie	PA3 (pin37)	D28
I9	Capteur d1 tige vérin Ejection 2 sortie	PA6 (pin34)	D30 ou A6
I7	Capteur réflexe magasin crm	PC7 (pin29)	D23
I5	Capteur a0 tige vérin Avance Brique rentrée	PC5 (pin12)	D21
I6	Capteur a1 tige vérin Avance Brique sortie	PC6 (pin28)	D22
I10	Capteur barrage poinçonnage /cbp	PA7 (pin33)	A7 ou 31
I11	Capteur présence brique Ejection 1 /cpeb1	PC0 (pin22)	D16
I12	Capteur présence brique Ejection 2 /cpeb2	PC1 (pin23)	D17
O1	Voyant vert VRUN	PB0 (pin1)	D0
O2	Pilotage A+ Avance brique	PB1 (pin2)	D1
O3	Pilotage A- Avance brique	PB2 (pin3)	D2
O4	Pilotage B+ Poinçonnage	PA0 (pin40)	A0 ou 24
O5	Pilotage C+ Ejection 1	PA1 (pin39)	A1 ou 25
O6	Pilotage D+ Ejection 2	PA2 (pin38)	A2 ou 26
O7	Pilotage relais RUN variateur	PD5 (pin19)	D13
O8		PD6 (pin20)	D14
AO0	Non utilisable sous LDmicro	PB3 (pin4)	D3
AO1	Sortie analogique 0/10V consigne variateur	PD7 (pin21)	D15
Boutons poussoirs et potentiomètre actifs en mode manuel seulement(mode manuel câblé)			
	Bouton poussoir BPA+		
	Bouton poussoir BPA-		
	Bouton poussoir BPB+		
	Bouton poussoir BPC+		
	Bouton poussoir BPD+		
	Bouton poussoir BPtapis		
	Commutateur AUTO/MANUEL		
	Potentiomètre P2		
	Bobine KA1 pilotée par le commutateur AUTO/MANUEL		
	Contacteur KM1 de mise sous tension du variateur		
Boutons/bobines/électrovannes et voyants pour la sécurité			
	Bouton coup de poing ATU général		
	Bouton coup de poing ATU1 système		
	Bouton ACQ (acquiescement défauts)		
	Bobine KAU (arrêt d'urgence)		
	Electrovanne d'arrêt VA+ (mise à l'échappement)		
	Voyant Défaut		
	Voyant Sous tension		

L'armoire comporte un ATU coupant la totalité de l'énergie et un 2^{ème} ATU propre à notre système:



Le mode manuel est câblé : les boutons poussoir permettant de piloter les électrovanne et moto-réducteur dans le désordre ne sont pas reportés en entrée du micro-contrôleur, cela permet d'économiser sur le nombre d'entrée et aussi, en cas de défaillance de la carte d'entrée, de pouvoir tout de même piloter le système.

2-Questions diverses :

Quel type de capteur est utilisé pour :

- détecter la présence brique dans le magasin ?
- de détecter la position de la tige des vérins ?
- de détecter la présence des briques avant éjection ?
- de détecter un bourrage de brique sur le tapis ?
- de détecter la présence d'une brique sous le poinçonneur ?

Qu'est-ce que la norme CEI CEI 61131-3

Comment est réalisée la sécurité machine ?

Quelle doivent être les caractéristiques de la sortie analogique de la carte ARDUINO pour piloter la consigne de vitesse du variateur ?

Comment cela se programme-t-il dans le LADDER sous LDmicro ?

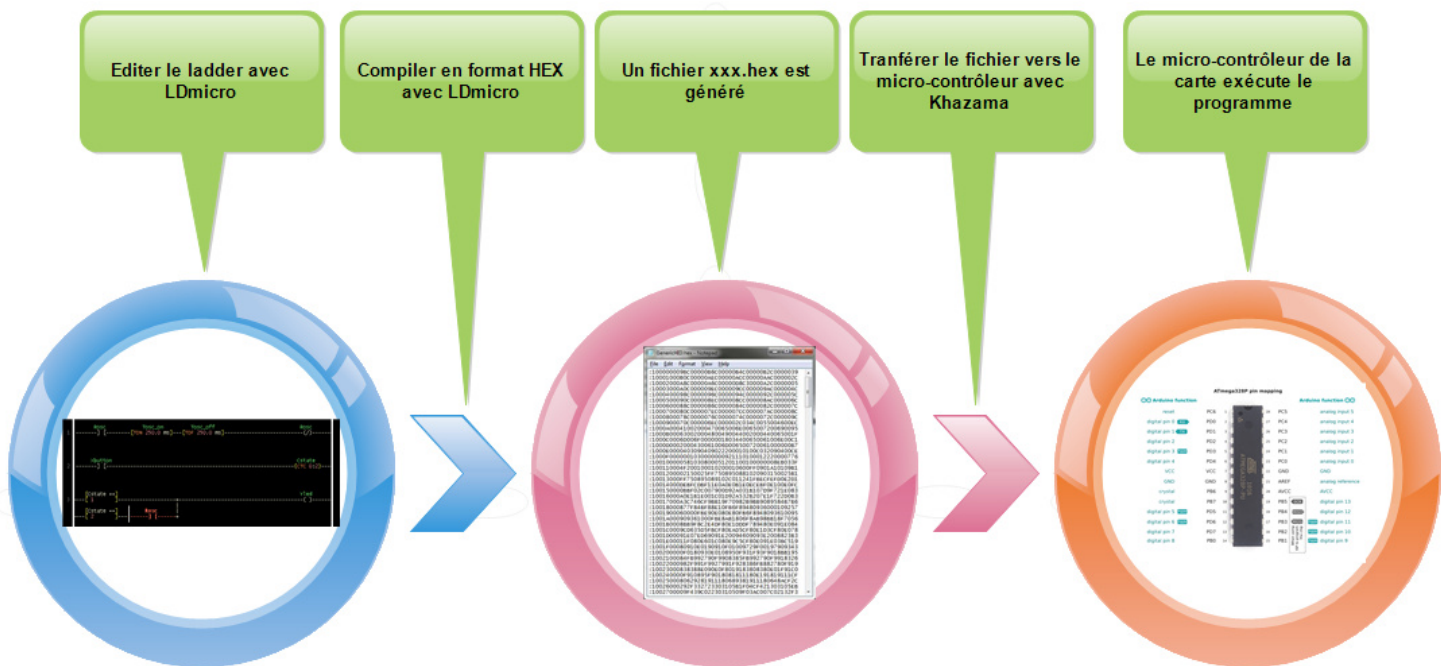


3- Travail à réaliser :

3.1. Programmation en ladder avec LDmicro :

3.1.1. Un premier programme à réaliser pour vérifier la communication avec le PC et l'inter-action entre les divers logiciels de programmation :

LDmicro est un logiciel permettant de représenter un programme en ladder et le compiler en fichier xxx.hex. Une fois ce fichier généré, un second logiciel Khazama permet de l'envoyer vers le micro contrôleur souhaité.

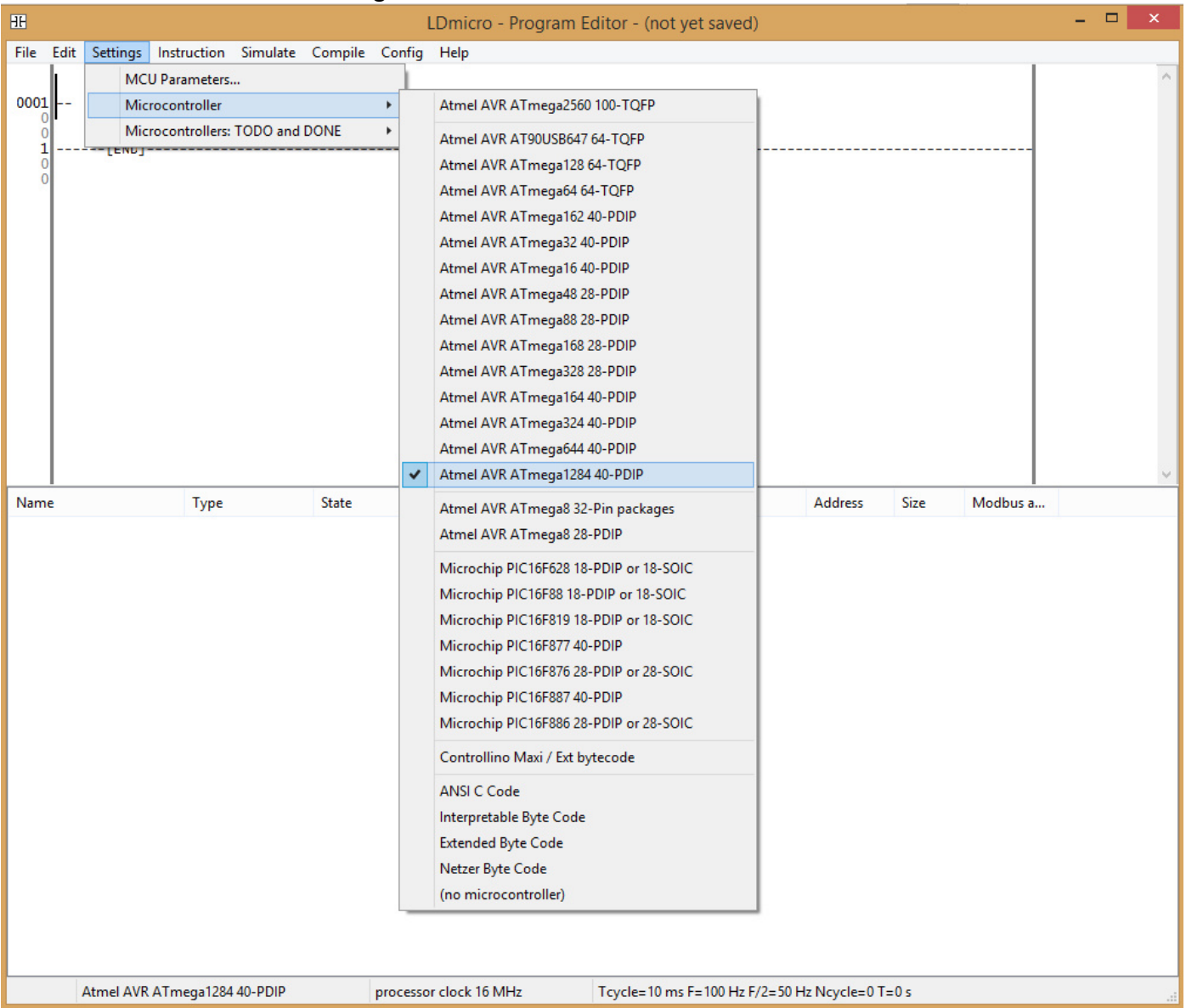


Contrairement à un automate industriel classique, dans cette configuration il n'est pas possible de récupérer un programme stocké dans la mémoire du micro-contrôleur.

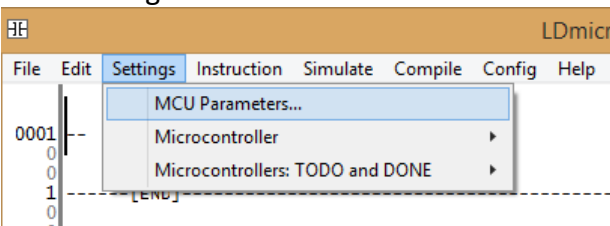
1)-Lancer LDmicro :

En appuyant sur le bouton poussoir DCY, la tige du vérin Avance brique sort et reste sortie tant que le capteur a1 n'est pas activé puis elle rentre. Représenter ci-dessous le schéma électrique :

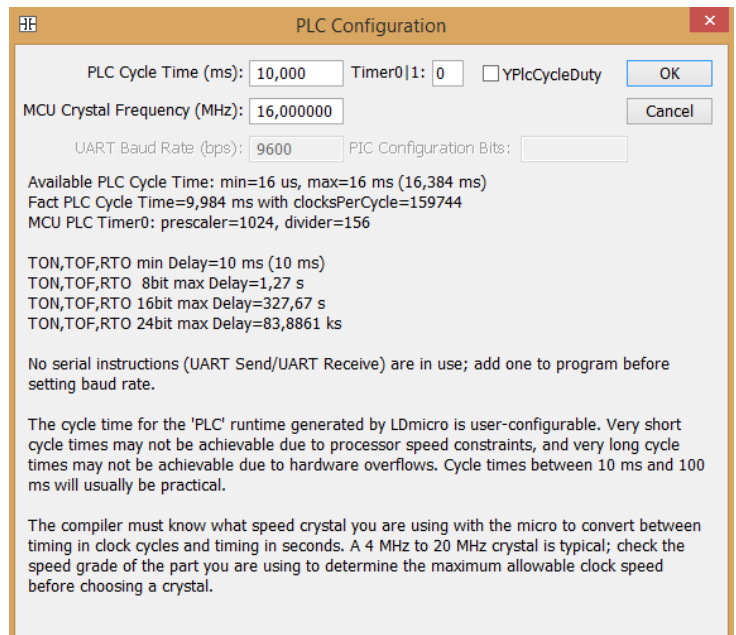
TRI DE BRIQUE MICROCONTROLEUR
Choisir le micro-contrôleur : Atmega1284P



Dans Settings :

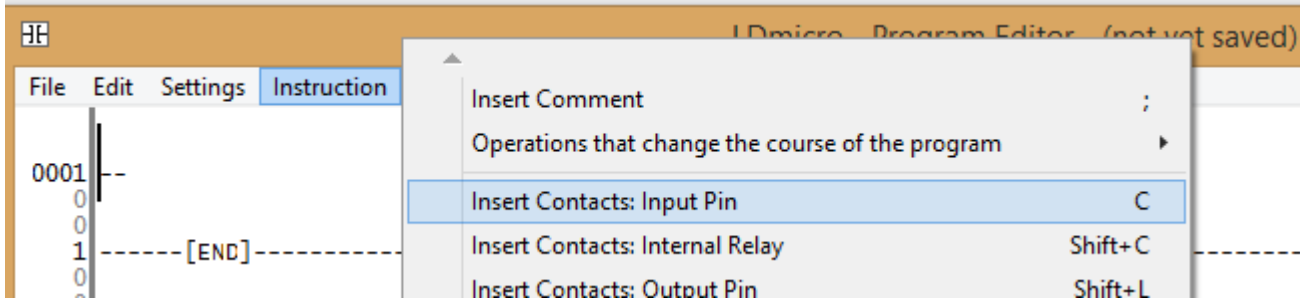


Fixer la fréquence du quartz à 16 MHz :
OK

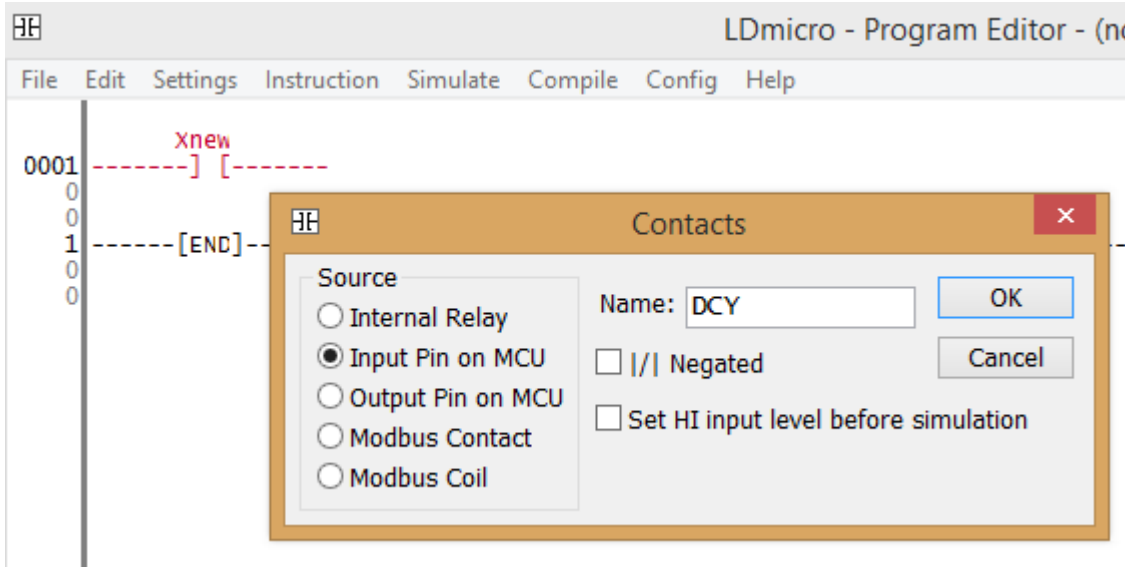


TRI DE BRIQUE MICROCONTROLEUR

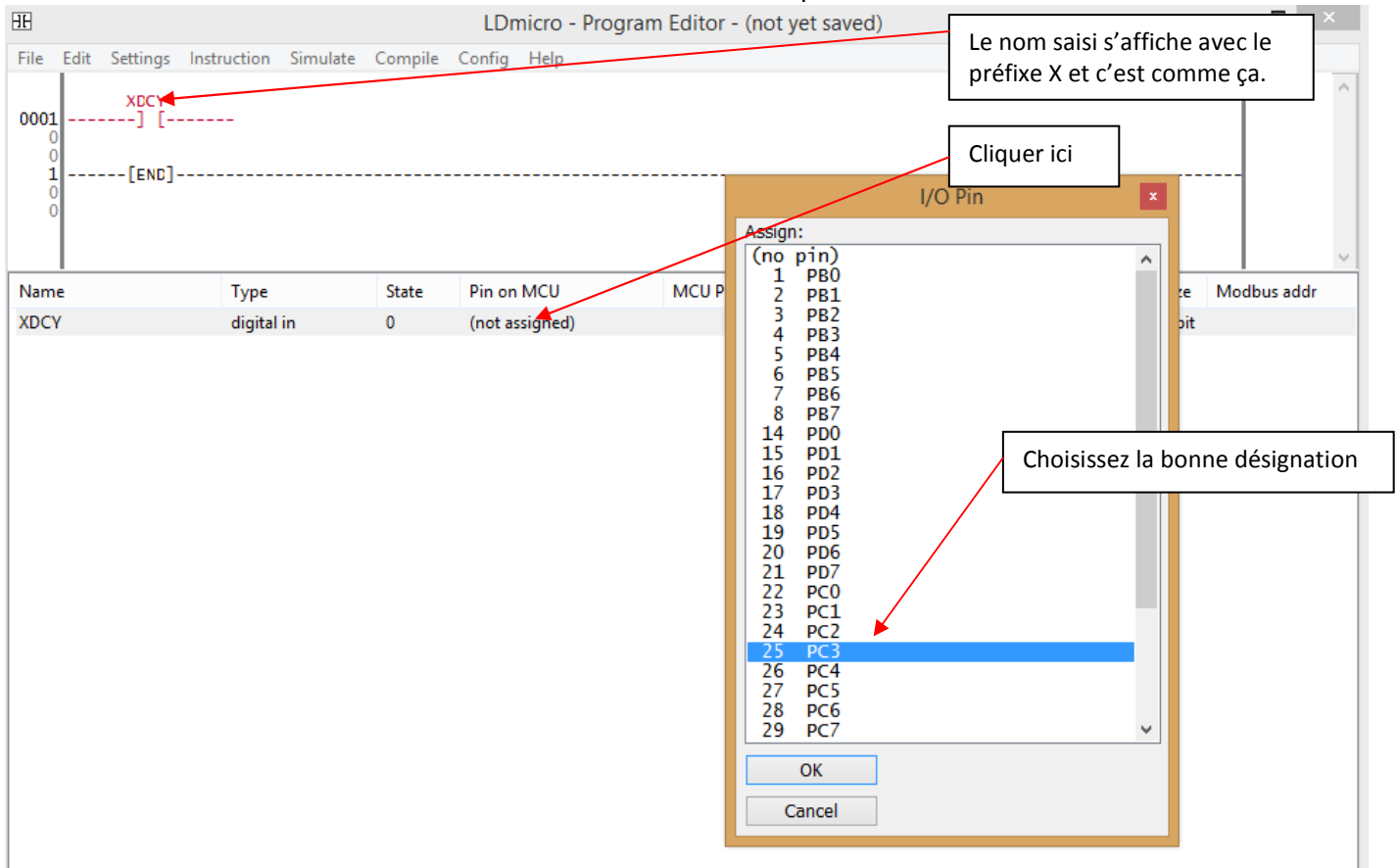
Implanter un contact :



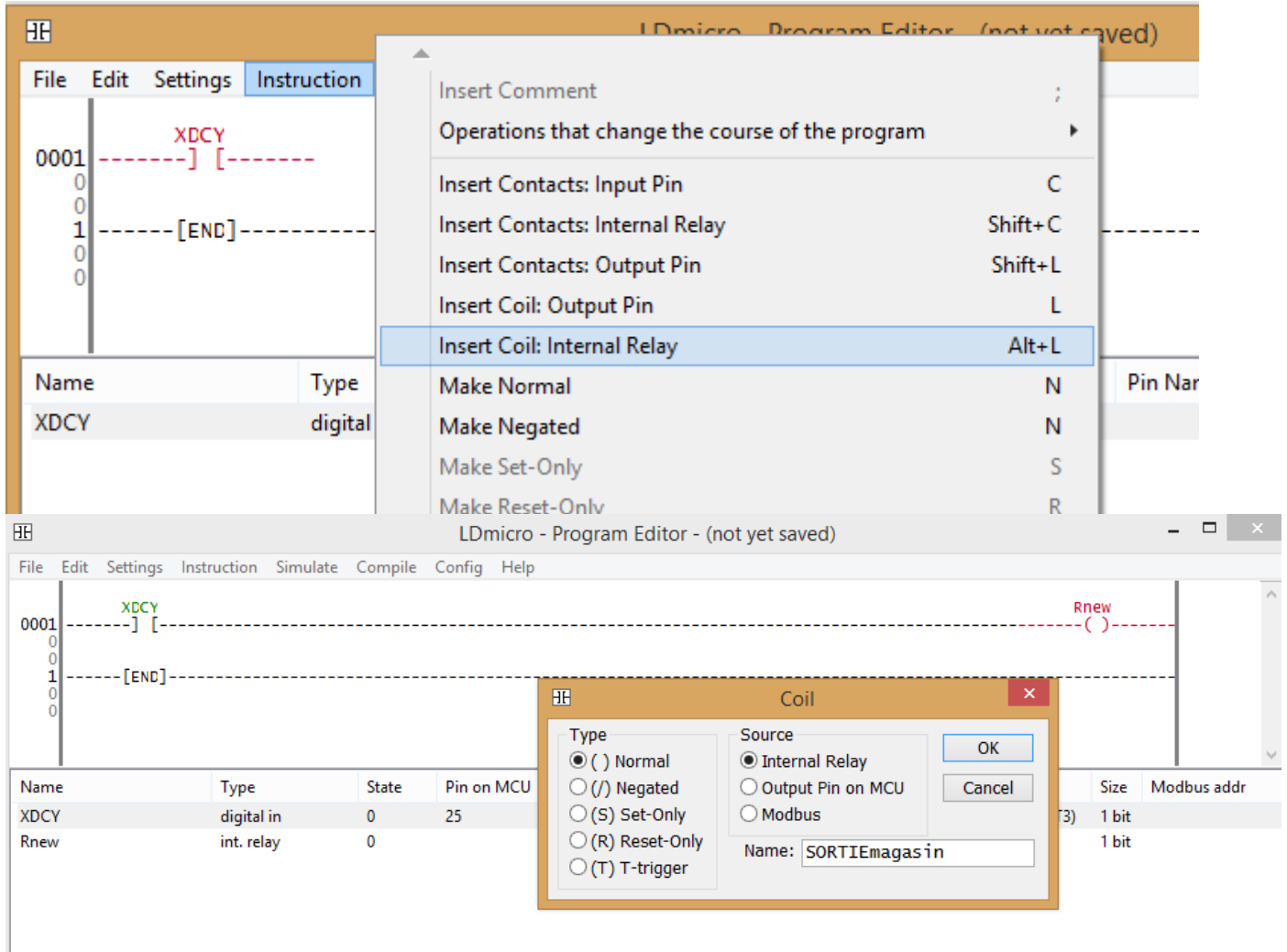
Lui donner un nom :



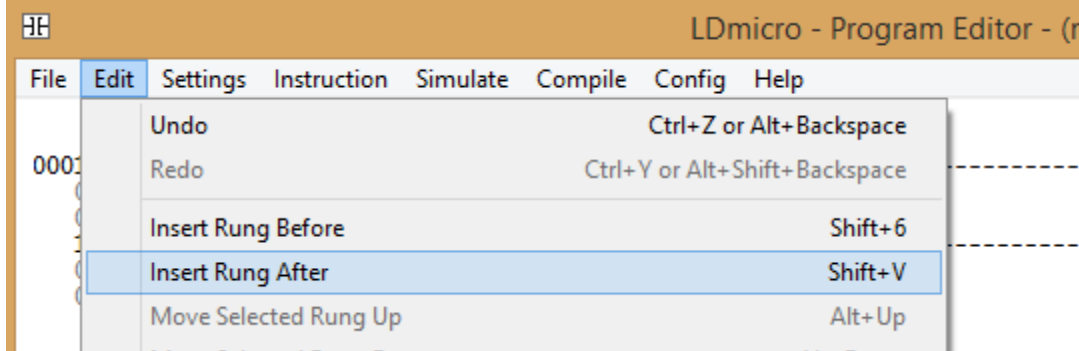
Affecter une adresse micro-contrôleur à ce nom : ici le bouton poussoir DCY est câblé sur PC3



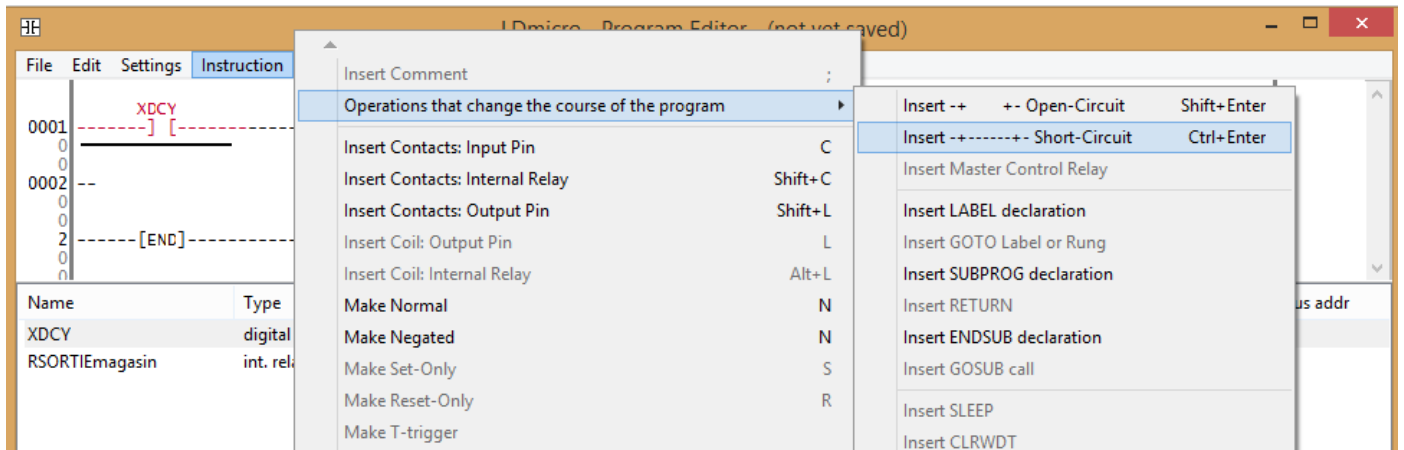
Pour insérer une bobine ou sortie : Instruction/Insert Coil



Ici on utilise un bit interne (Internal relay) qui sera dénommé RSORTIEmagasin dans le ladder. Pour insérer un rung ou ligne supplémentaire, se positionner en début de rung puis :



Pour insérer 2 contacts en fonction OU :

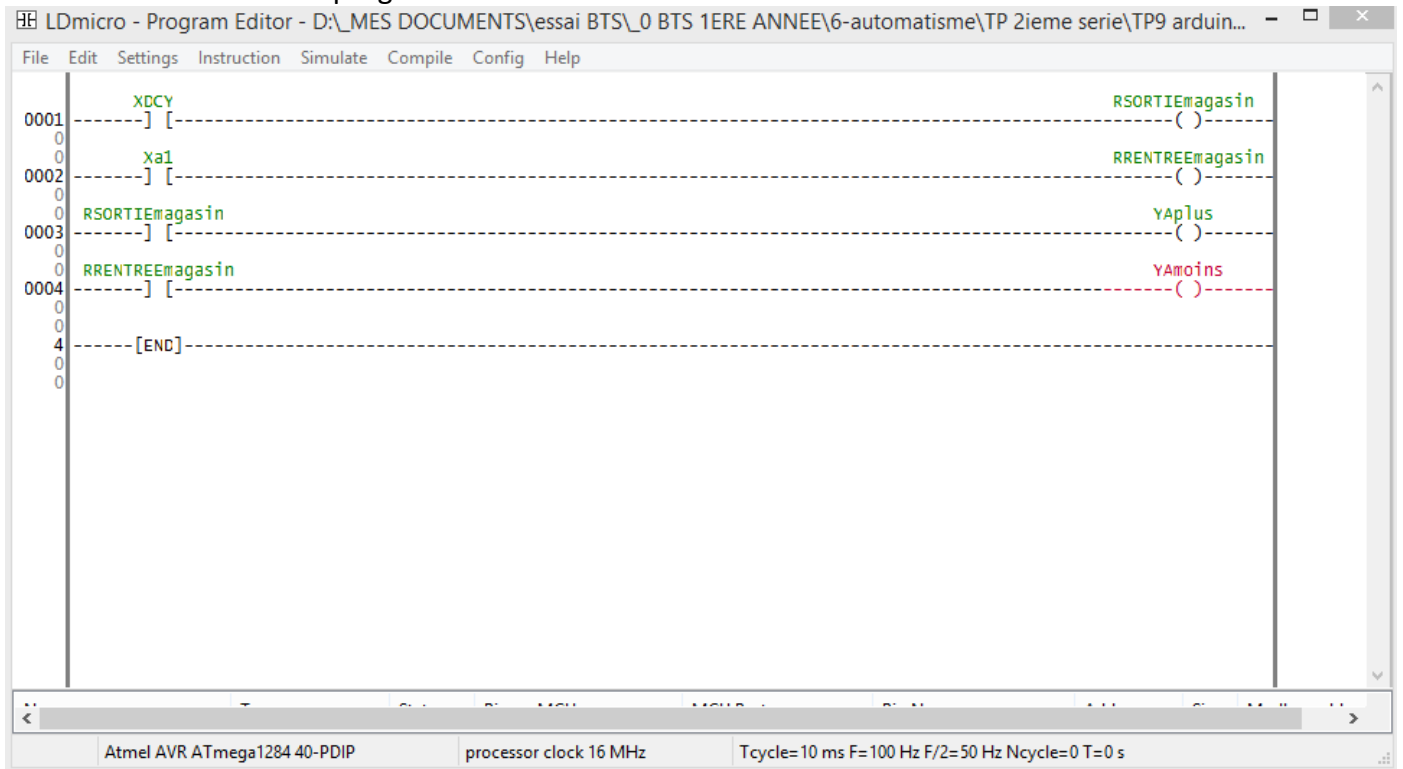


TRI DE BRIQUE MICROCONTROLEUR

Se positionner sous le contact puis insérer un court-circuit.

Insérer ensuite vos contacts dans le court-circuit.

Vous devrez aboutir à un programme ressemblant à cela :



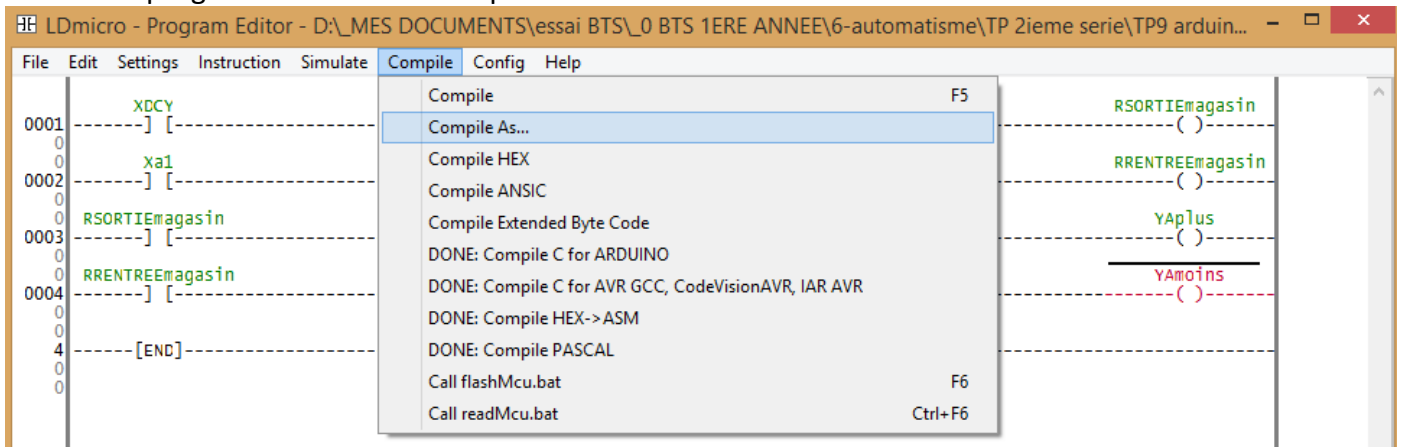
Le bit interne est associé, en fin de ladder, à la sortie à piloter.

Pourquoi cette méthode ?

On ne peut pas utiliser directement un contact NO associé à YAplus ou YAmoins dans le programme.

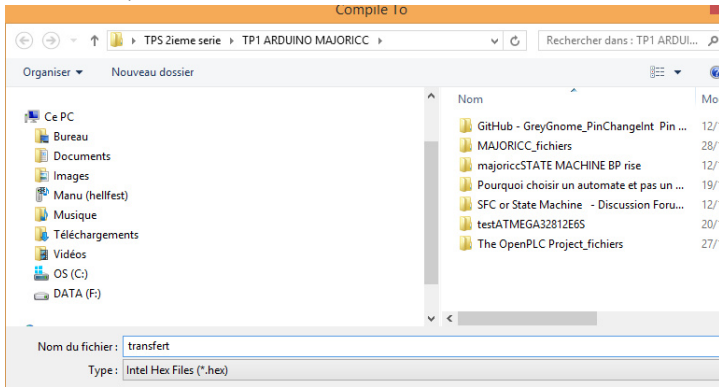
2)-Compiler le ladder édité :

Une fois le programme terminé : compiler

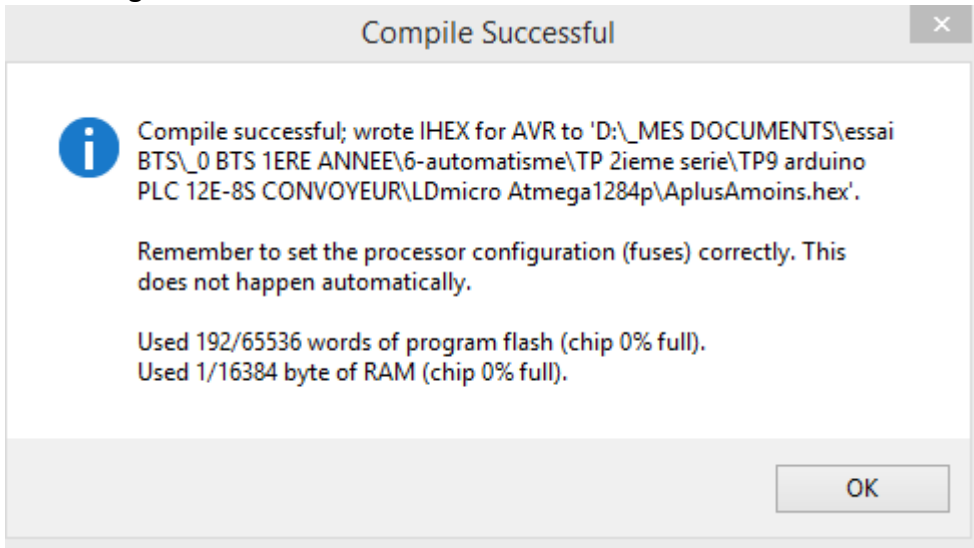


Donner un nom :

TRI DE BRIQUE MICROCONTROLEUR



Un message s'affiche : OK

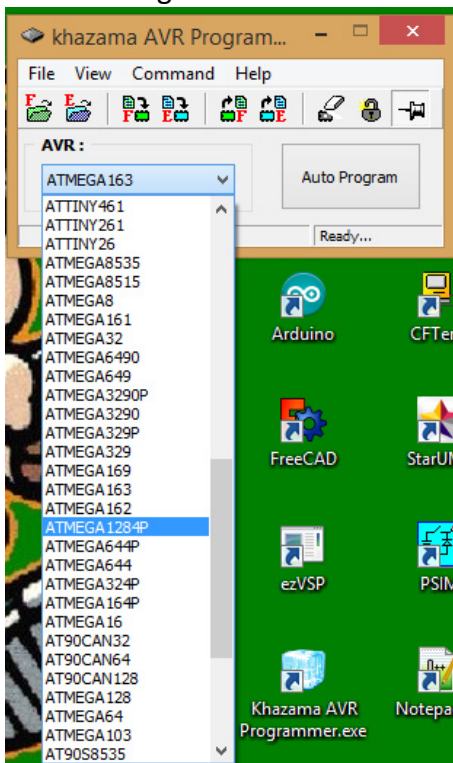


Un fichier transfert.hex a été généré.

3)-Transférer le fichier xxx.hex vers le micro-contrôleur :

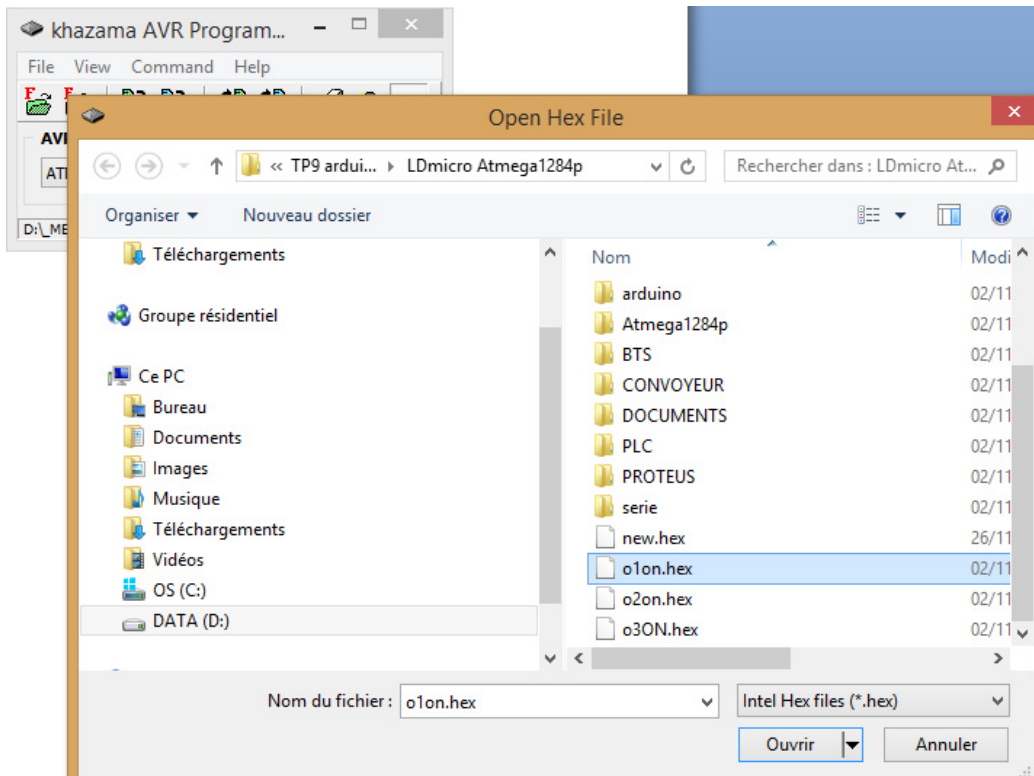
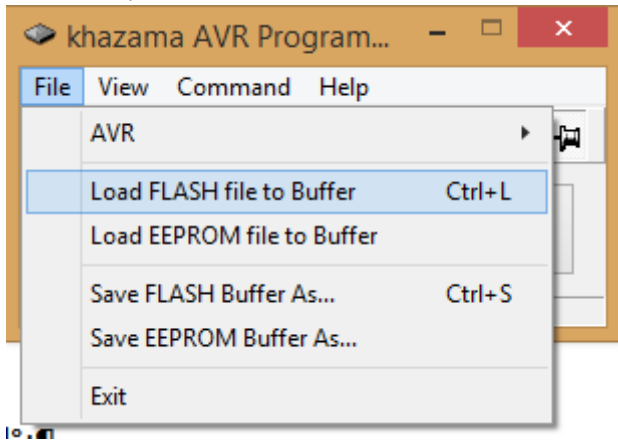
Vous utiliserez le logiciel Khazama.

Lancer le logiciel : choisir le micro-contrôleur approprié : ici l'atmega1284p

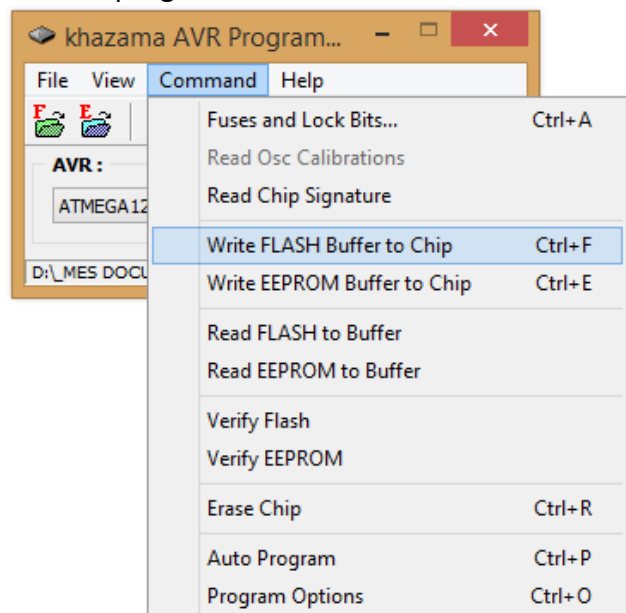


Charger le fichier xxx.hex généré précédemment dans le logiciel LDmicro:

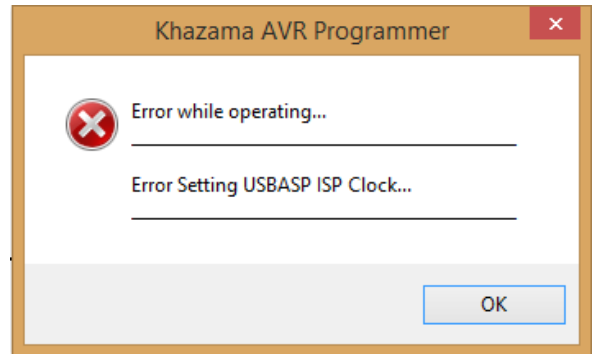
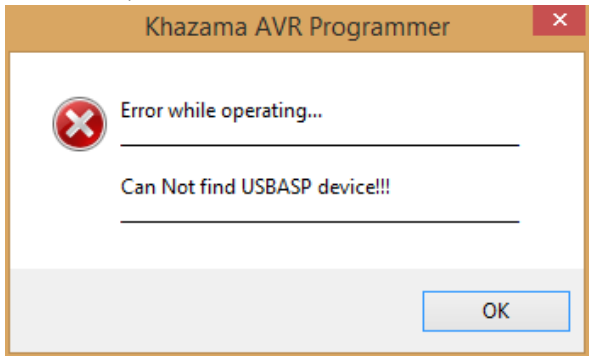
TRI DE BRIQUE MICROCONTROLEUR



Ecrire le programme dans le micro-contrôleur :

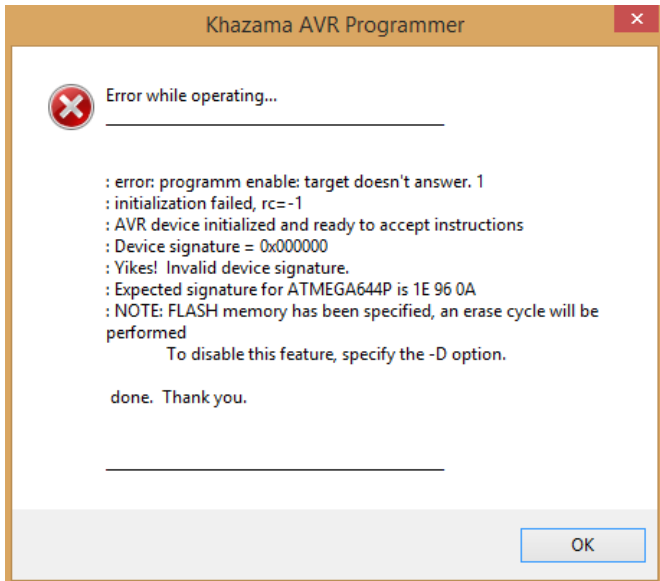


Le câble de programmation doit être correctement branché sinon une erreur survient :



Si le programme est correctement transféré une autre erreur peut survenir

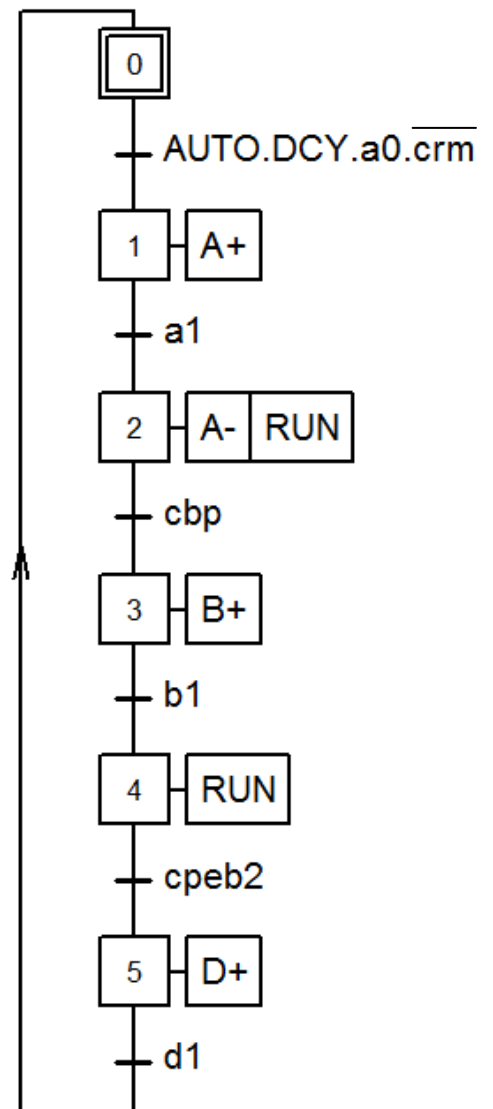
Ne pas en faire de cas, le programme a tout de même été transféré.



Tester et rêver.

3.1.2. Un deuxième programme sous LDmicro : programmer un Grafct simple à transformer en Ladder.

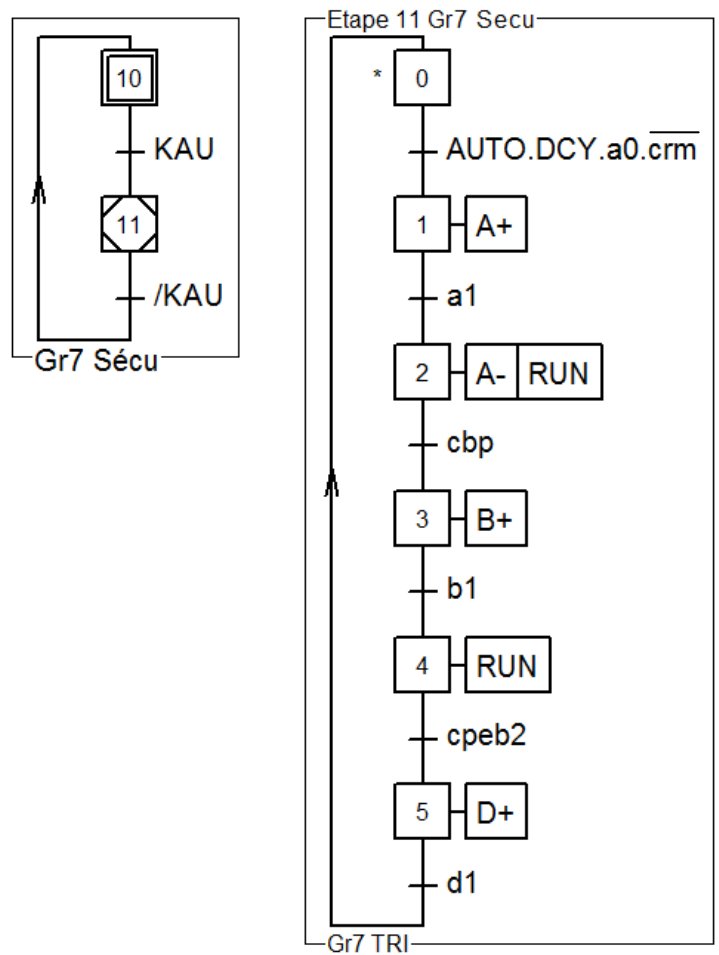
-En mode AUTO, dès l'appui sur le bouton DCY, et à condition qu'il y ait des briques dans le magasin, une brique est envoyée sur le tapis convoyeur qui se met en route puis arrivée sous le poinçonneur : poinçonnage. Puis évacuation par éjecteur 2. Faire le ladder correspondant : vous pouvez consulter le fichier exemple de la fin du TP.



TRI DE BRIQUE MICROCONTROLEUR

-Ajouter l'arrêt d'urgence qui empêchera tout mouvement.

Apporter les modifications au ladder précédent.

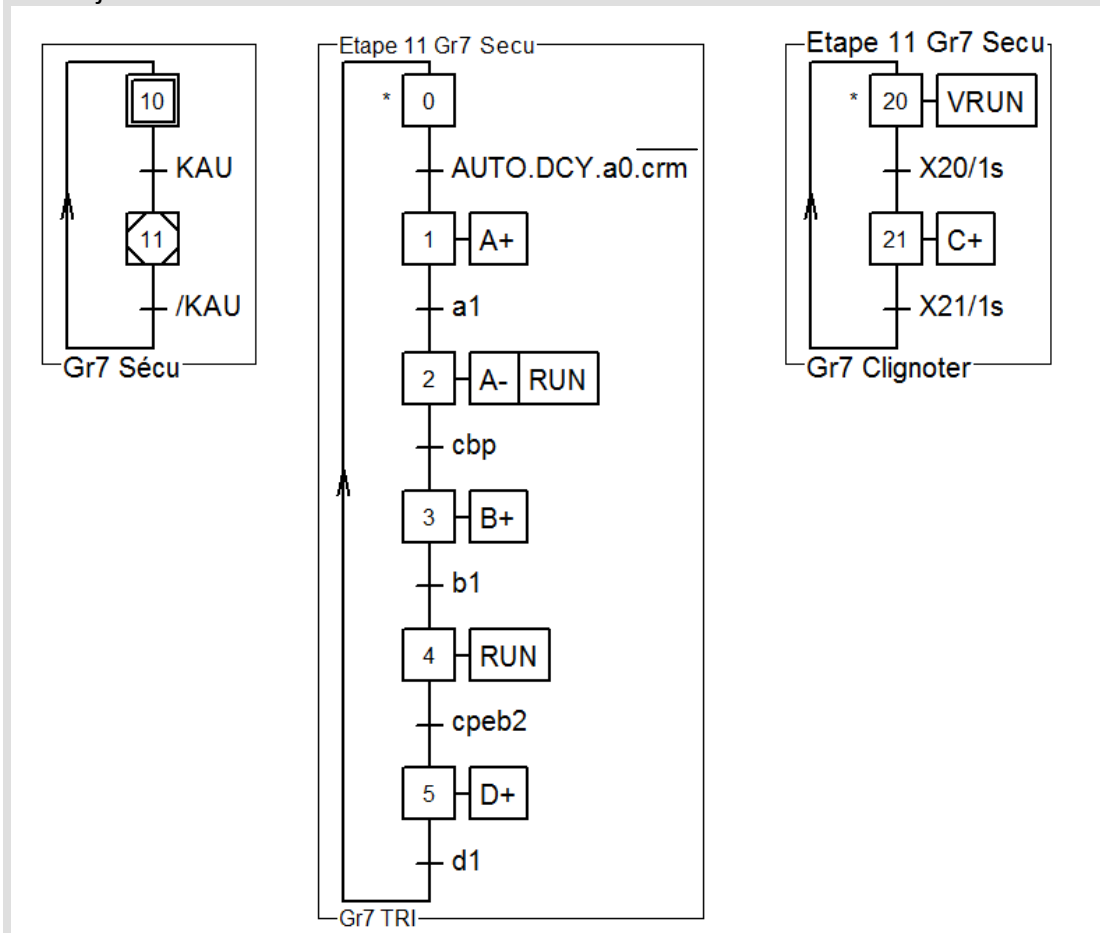


3.1.3. En option : un troisième programme sous

LDmicro : multitâche

En conservant le Grafcet précédent, faire un autre

programme fonctionnant en parallèle et faisant fonctionner alternativement le voyant vert VRUN et le vérin éjecteur 1.

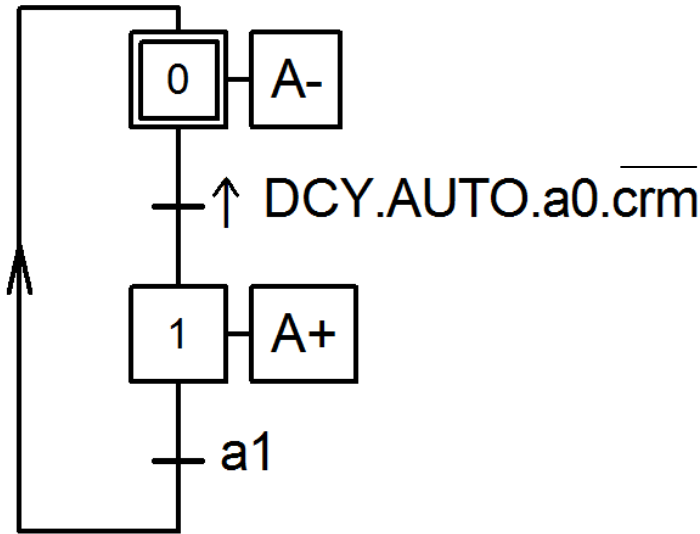


Apporter les modifications au ladder précédent.

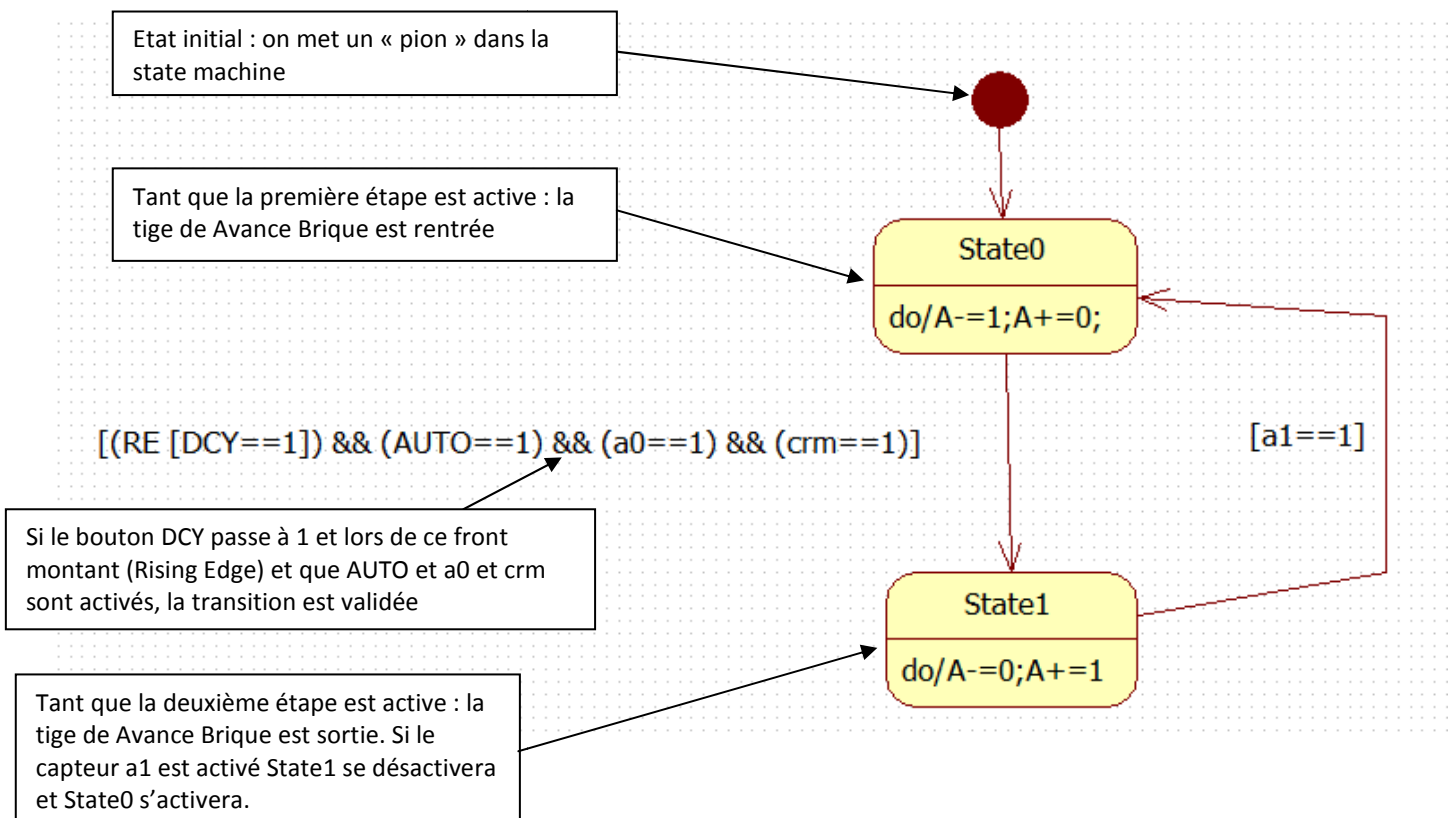
3.2. Programmation en langage ST d'un state machine: sous Arduino 1.8.2

3.2.1. Premier programme :

En appuyant sur le bouton poussoir DCY, la tige du vérin Avance brique sort et reste sortie tant que le capteur a1 n'est pas activé puis elle rentre.



La traduction dans le formalisme de la state machine est simple à comprendre :



Définitions :

- == : test d'égalité
- = : affectation
- && : fonction ET

On ne peut pas dessiner ce diagramme et le télécharger vers l'Arduino. Vous allez donc traduire ce diagramme en langage C très simplement avant de le télécharger.

1) Lancer Arduino 1.8.2 :

Un programme sous Arduino s'écrit en langage C et comporte 3 zones :

```
Blink | Arduino 1.6.1
Fichier Édition Croquis Outils Aide
Blink$
#define Tsortie 13

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(Tsortie, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(Tsortie, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(Tsortie, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Compilation terminée.
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2 039 octets pour les variables locales. Le maximum est de 2 048 octets.
1 Arduino Uno on COM11
```

Zone de déclaration des variables à utiliser dans le programme, de librairie ou option spéciale à utiliser (périphériques de communication, extension d'entrée/sorties...).

void setup() : Zone de déclaration des directions des pattes du micro-contrôleur : entrée ou sortie (input ou ouput).

void loop() :Zone de programme dans une boucle infinie par défaut.

Le programme permettant de traduire la state machine en langage C est le suivant :

```
trieuse | Arduino 1.8.2
Fichier Édition Croquis Outils Aide
trieuse$
#include <SM.h>
SM Trieuse(&State0); //création de la machine "Trieuse" et déclaration
// de State0 comme l'état initial recevant le pion

//Boutons poussoirs, capteurs et sorties
#define DCY 19
#define a0 21
#define a1 22
#define AUTO 11
#define crm 23
#define Aplus 1
#define Amoins 2

int etat = 0; //variable utilisée pour la détection de front montants

void setup() {
  pinMode(DCY, INPUT); //les capteurs et boutons en entrées, les électrovannes
  pinMode(a0, INPUT); //et relais et voyants en sortie
  pinMode(a1, INPUT);
  pinMode(AUTO, INPUT);
  pinMode(crm, INPUT);
  pinMode(Aplus, OUTPUT);
  pinMode(Amoins, OUTPUT);
}
```

Utilisation d'une librairie spéciale ou option SM (State Machine)

Association des noms des capteurs et des pattes de l'arduino

Déclarations des entrées et des sorties

TRI DE BRIQUE MICROCONTROLEUR

```
void loop() {  
  EXEC(Trieuse); //mise en marche de la state machine: un pion est déposé dans  
  // State0 et les séquences peuvent se dérouler  
}  
  
State State0() {  
  digitalWrite(Amoins, HIGH); //A- mis à 1  
  digitalWrite(Aplus, LOW); //A+ mis à 0  
  if ((RE(digitalRead(DCY), etat) == 1) && (digitalRead(a0) == 1) &&  
      (digitalRead(AUTO) == 0) && (digitalRead(crm) == 1)) Trieuse.Set(State1);  
}  
  
State State1() {  
  digitalWrite(Amoins, LOW); //A- mis à 0  
  digitalWrite(Aplus, HIGH); //A+ mis à 1  
  if (digitalRead(a1) == 1) Trieuse.Set(State0);  
}
```

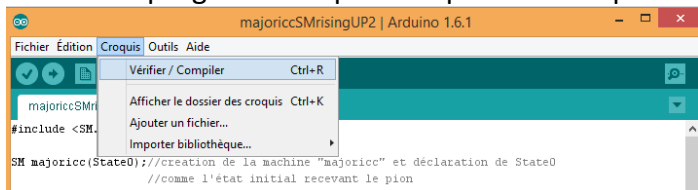
Le programme principal lançant la state machine

Les 2 étapes State0 et State 1 que l'on trouve dans la représentation graphique ainsi que les transitions. Pour des étapes supplémentaires, il suffit de les rajouter à la suite de ce programme

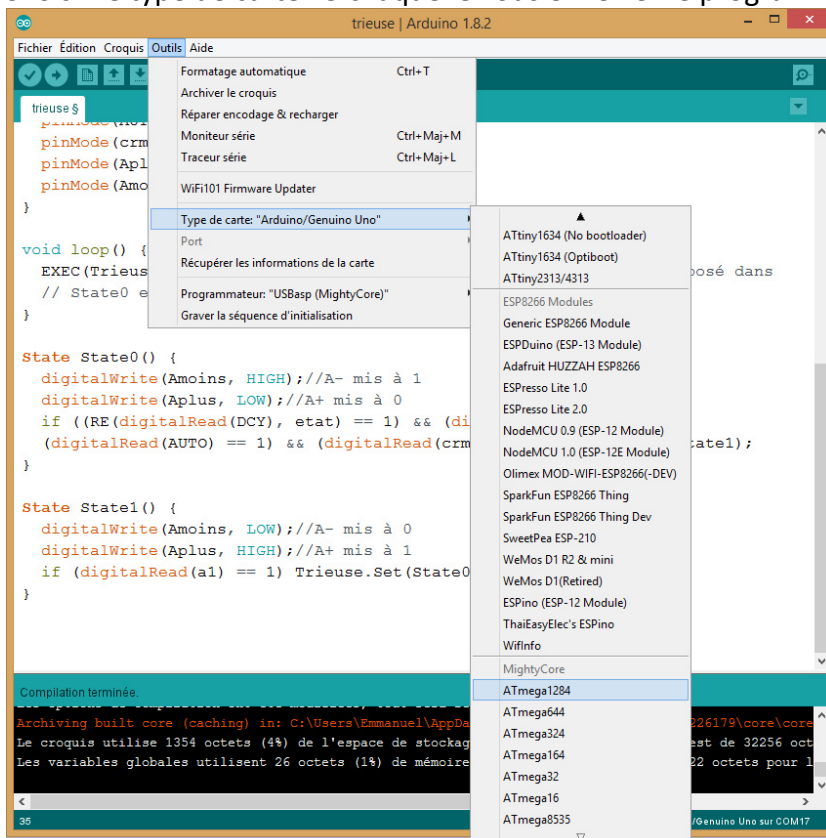
Ce programme utilise une librairie spéciale qui permet d'éviter d'écrire plus de de ligne de code. On trouve derrière le symbole « // » les commentaires que vous pouvez apporter au programme

2) Compiler- téléverser :

Une fois le programme tapé : compiler le code pour vérifier d'éventuelles erreurs.

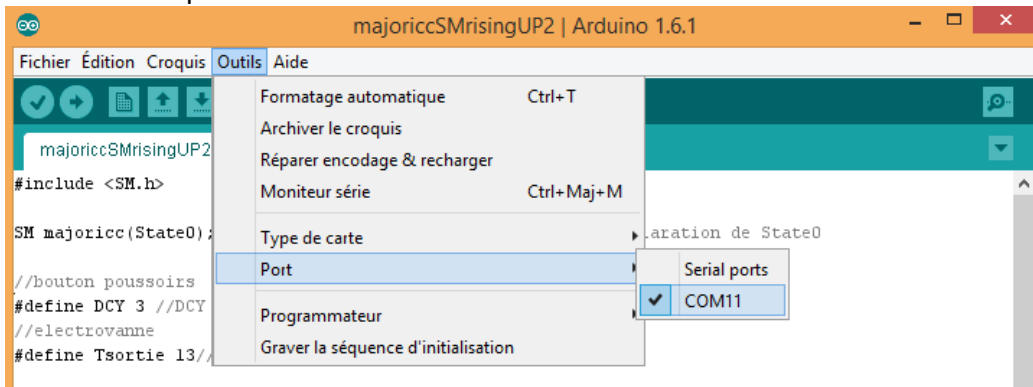


Choisir le type de carte vers laquelle vous enverrez le programme : choisir atmega1284

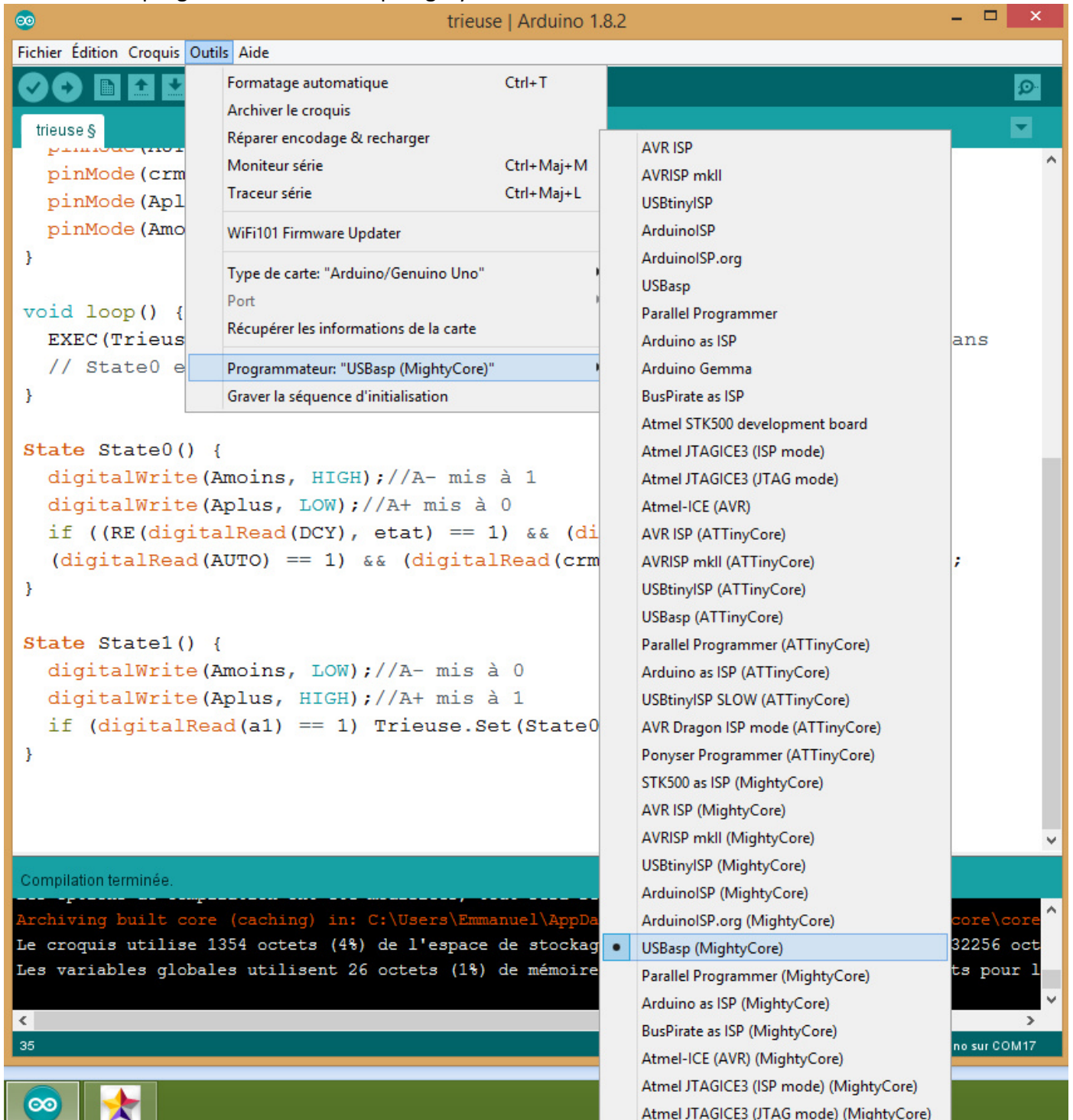


TRI DE BRIQUE MICROCONTROLEUR

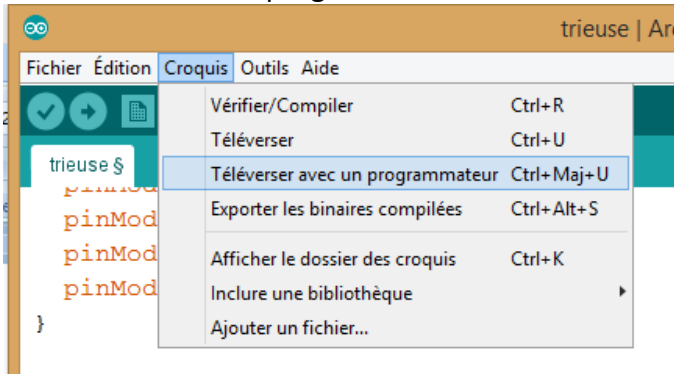
Choisir le bon port COM de communication :



Choisir le bon programmeur : USBasp MightyCore

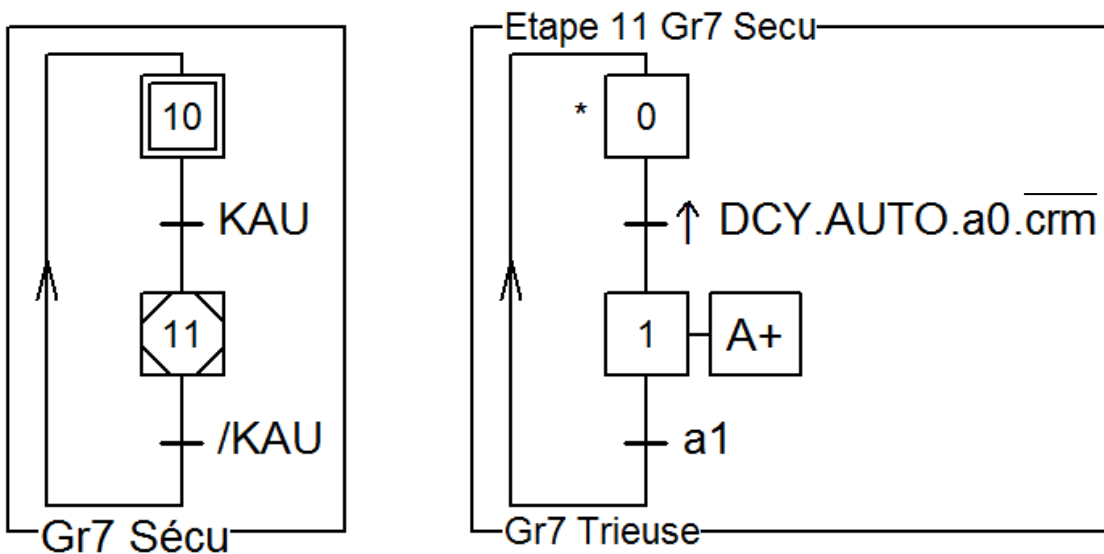


TRI DE BRIQUE MICROCONTROLEUR
Téléverser : avec un programmeur.

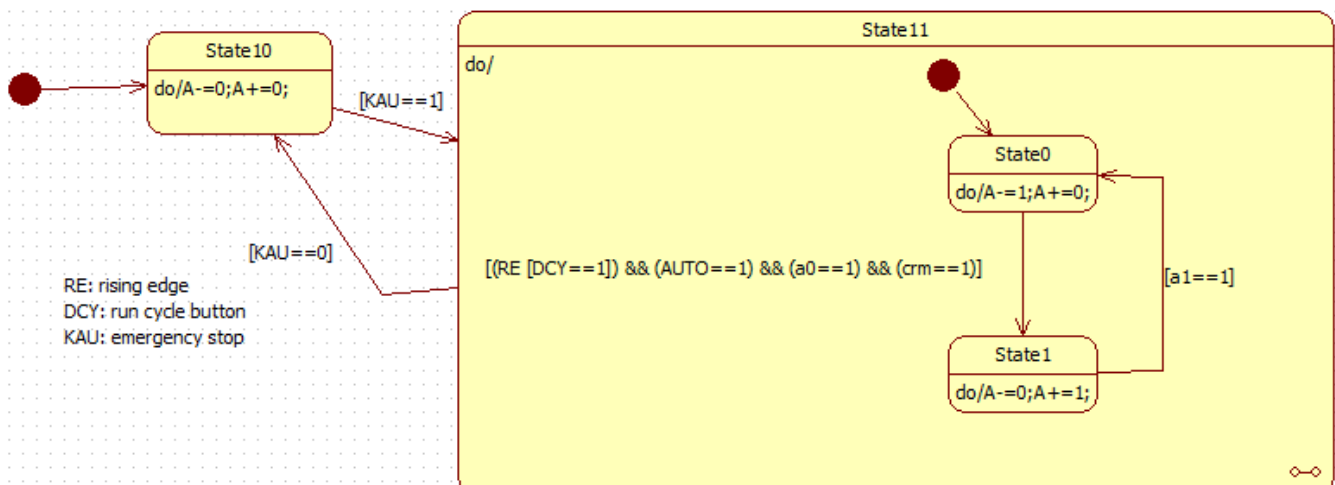


3.2.2. Deuxième programme :

-En cas d'arrêt d'urgence, cela empêchera tout mouvement.



La traduction en formalisme State machine (très facile à comprendre n'est-ce pas ?) :



Définitions :

- == : test d'égalité
- = : affectation
- && : fonction ET
- || : fonction OU

TRI DE BRIQUE MICROCONTROLEUR
Le programme correspondant :

```
trieuseATU | Arduino 1.8.2
Fichier Édition Croquis Outils Aide
trieuseATU
#include <SM.h>

SM Gr7Secu(&State10); //création de la machine "Gr7Secu" et déclaration
// de State10 comme l'état initial recevant le pion
SM Trieuse(&State0); //création de la machine "Trieuse" et déclaration
// de State0 comme l'état initial recevant le pion

//Boutons poussoirs, capteurs et sorties
#define DCY 19
#define a0 21
#define a1 22
#define AUTO 11
#define crm 23
#define Aplus 1
#define Amoins 2
#define KAU 10

int etat = 0; //variable utilisée pour la détection de front montants

void setup() {
  pinMode(DCY, INPUT); //les capteurs et boutons en entrées, les électrovannes
  pinMode(a0, INPUT); //et relais et voyants en sortie
  pinMode(a1, INPUT);
  pinMode(AUTO, INPUT);
  pinMode(crm, INPUT);
  pinMode(Aplus, OUTPUT);
  pinMode(Amoins, OUTPUT);
  pinMode(KAU, INPUT);
}

void loop() {
  EXEC(Gr7Secu); //mise en marche de la state machine: un pion est déposé dans
  // State10 et les séquences peuvent se dérouler
  if (digitalRead(KAU) == 0) {
    digitalWrite(Amoins, LOW); //A- mis à 0
    digitalWrite(Aplus, LOW); //A+ mis à 0
    Trieuse.Finish();
  }
  if (digitalRead(KAU) == 1 && Trieuse.Finished) {
    EXEC(Trieuse);
    Trieuse.Set(State0);
  }
}
```

TRI DE BRIQUE MICROCONTROLEUR

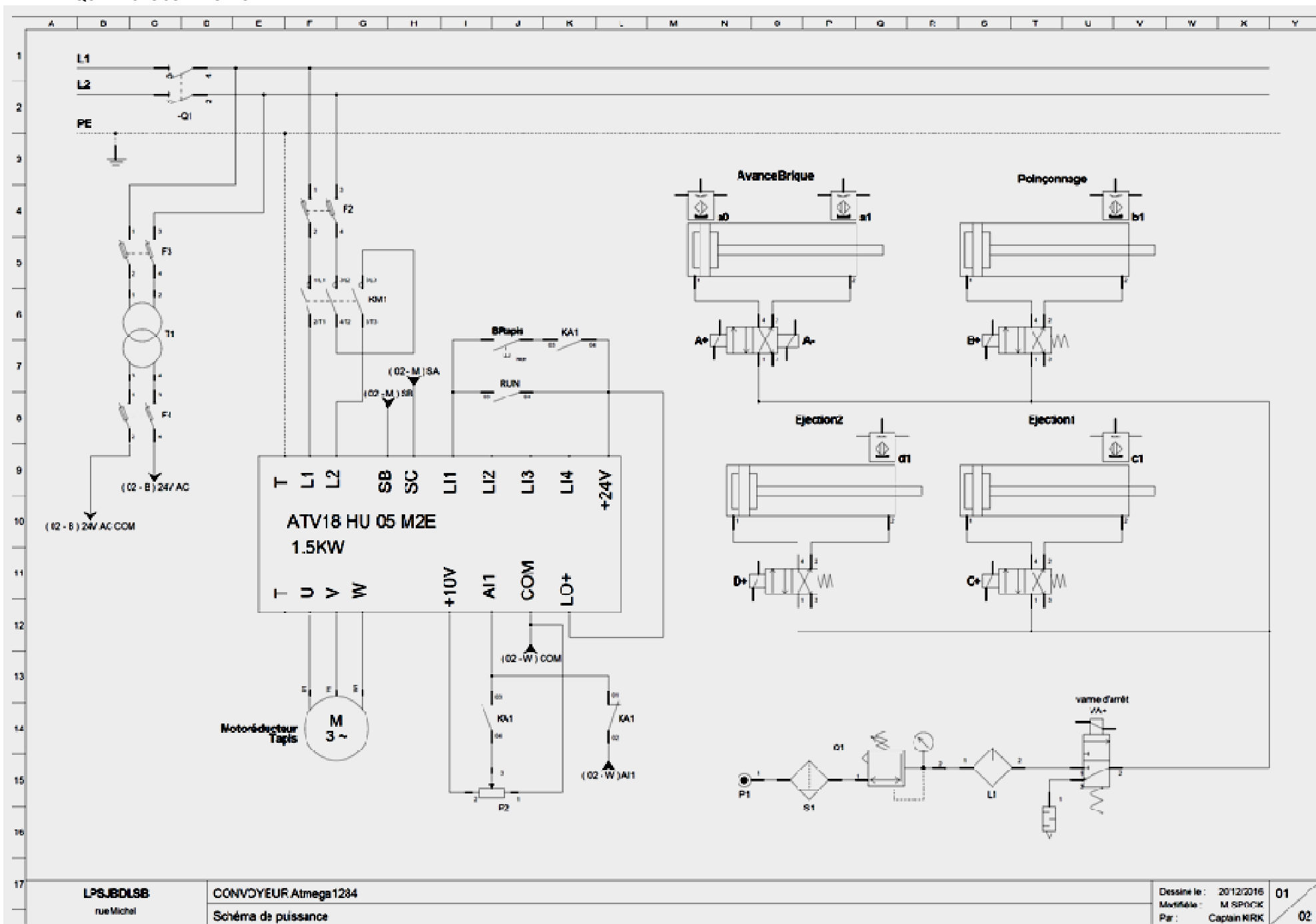
```
State State10() {  
    digitalWrite(Amoins, LOW); //A- mis à 0  
    digitalWrite(Aplus, LOW); //A+ mis à 0  
    if (digitalRead(KAU) == HIGH) Gr7Secu.Set(State11);  
}
```

```
State State11() {  
    EXEC(Trieuse);  
    if (digitalRead(KAU) == LOW) Gr7Secu.Set(State10);  
}
```

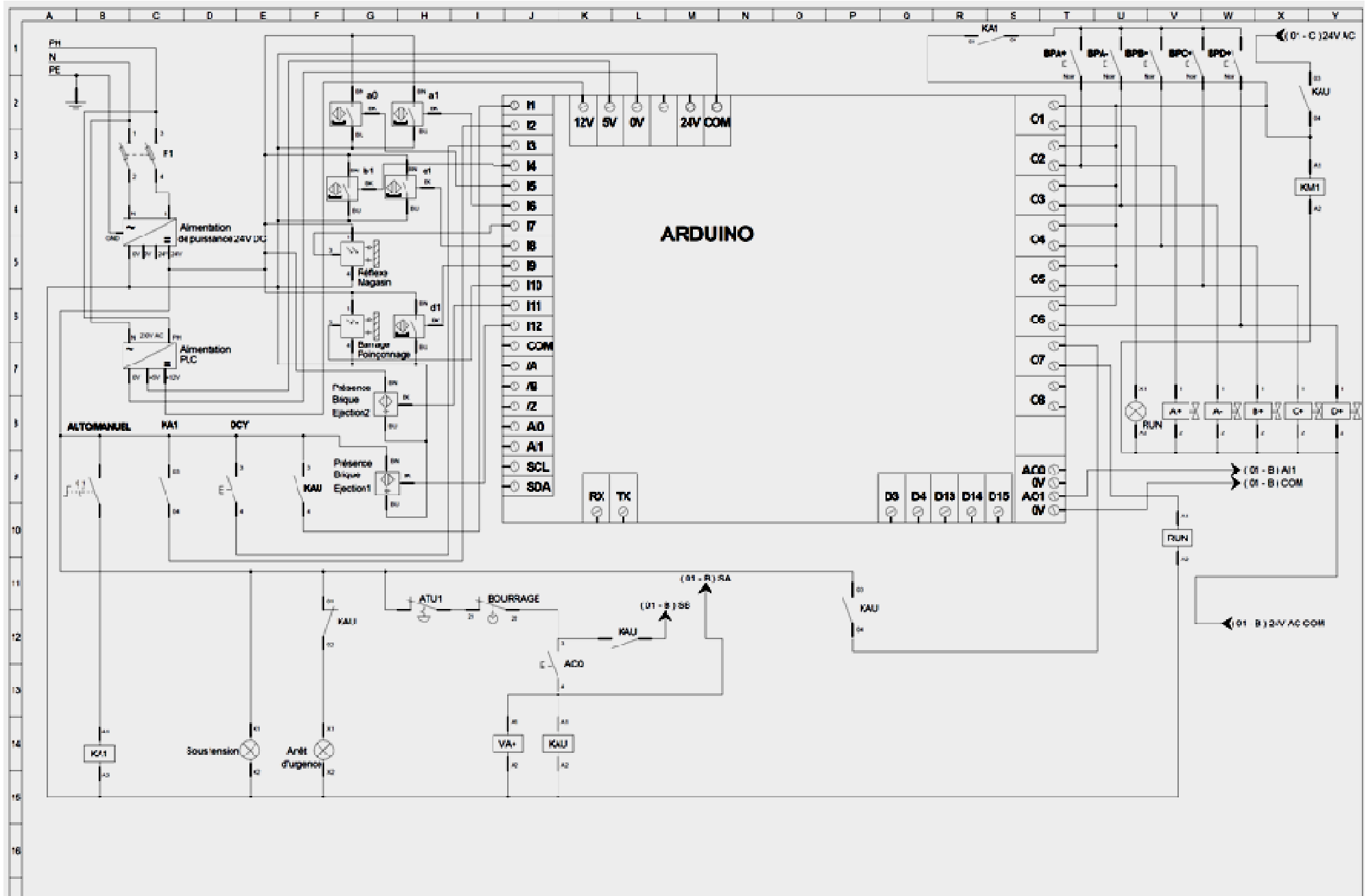
```
State State0() {  
    digitalWrite(Amoins, HIGH); //A- mis à 1  
    digitalWrite(Aplus, LOW); //A+ mis à 0  
    if ((RE(digitalRead(DCY), etat) == 1) && (digitalRead(a0) == 1) &&  
        (digitalRead(AUTO) == 0) && (digitalRead(crm) == 1)) Trieuse.Set(State1);  
}
```

```
State State1() {  
    digitalWrite(Amoins, LOW); //A- mis à 0  
    digitalWrite(Aplus, HIGH); //A+ mis à 1  
    if (digitalRead(a1) == 1) Trieuse.Set(State0);  
}
```

TRI DE BRIQUE MICROCONTROLEUR



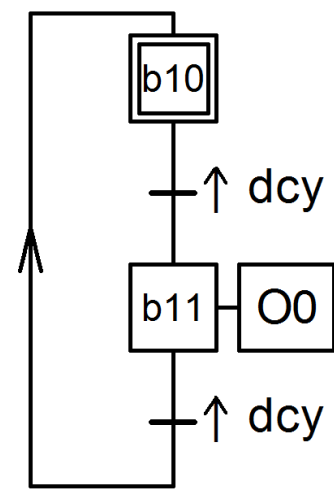
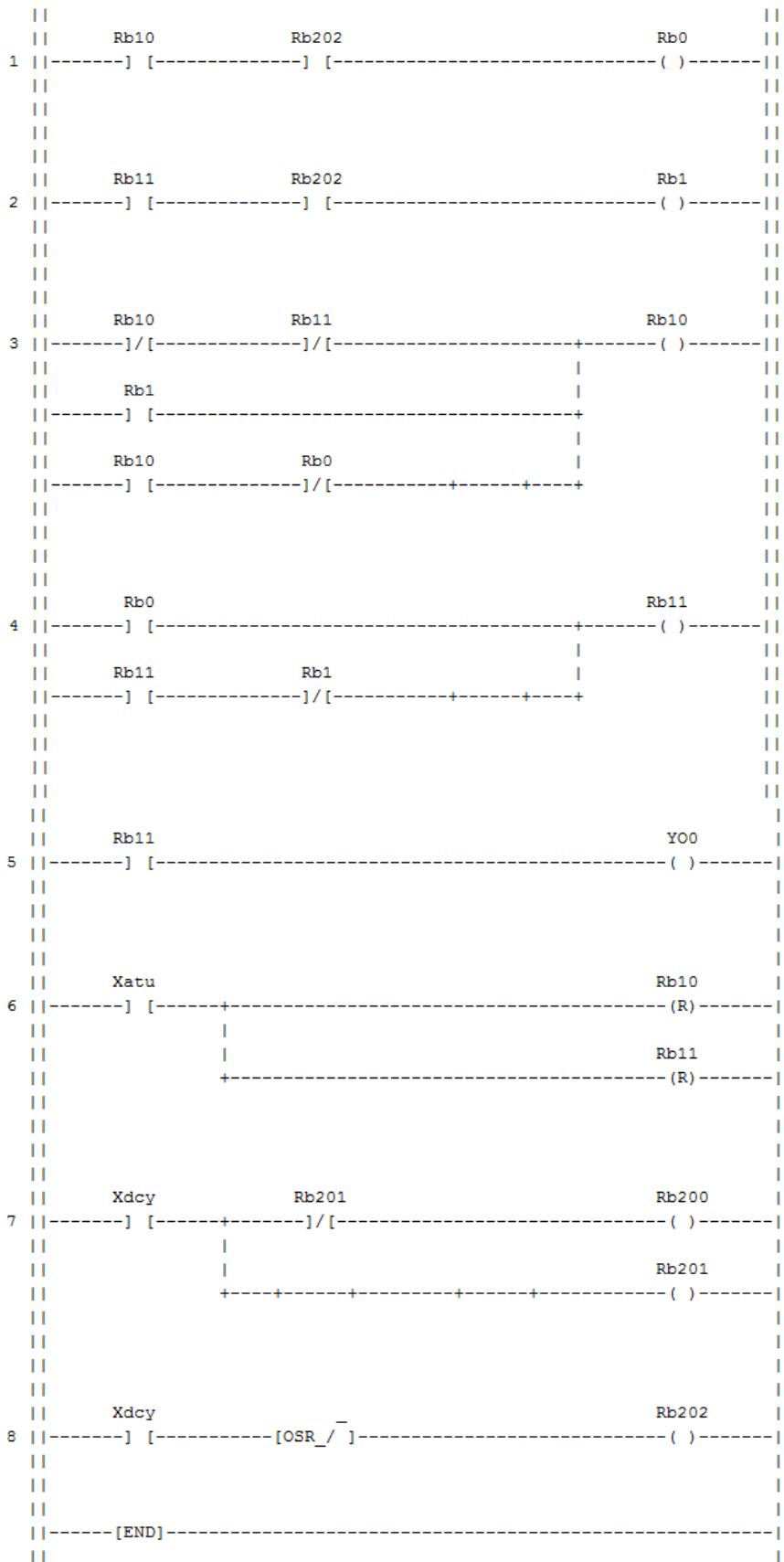
TRI DE BRIQUE MICROCONTROLEUR



Exemple de programme LDmicro :Atmel AVR ATmega328 28-PDIP, 16.000000 MHz

crystal, 10.0 ms cycle time, 2 methodes pour fronts montants

LADDER DIAGRAM:



I/O ASSIGNMENT:

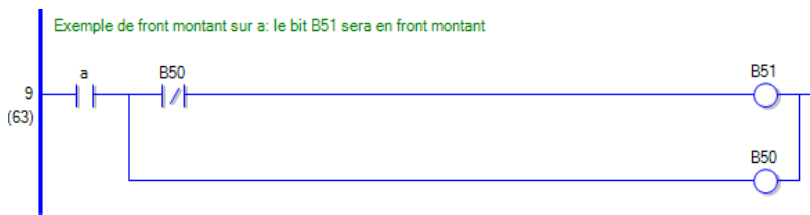
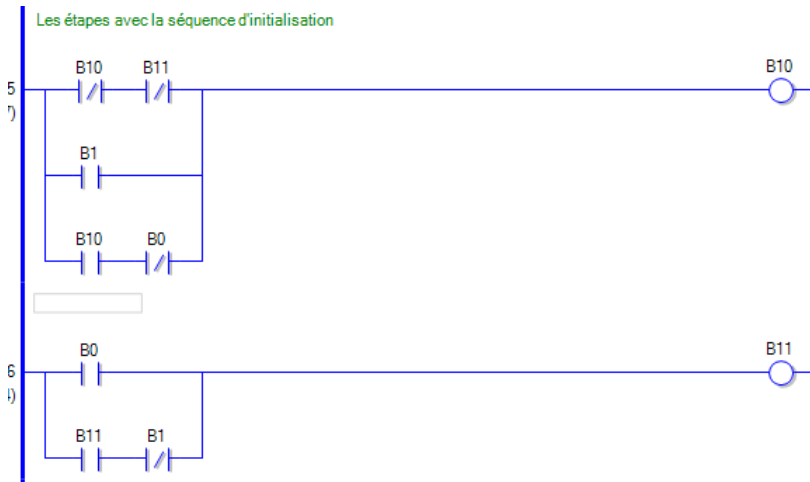
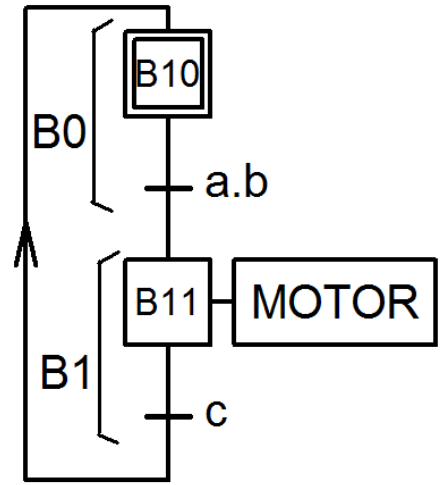
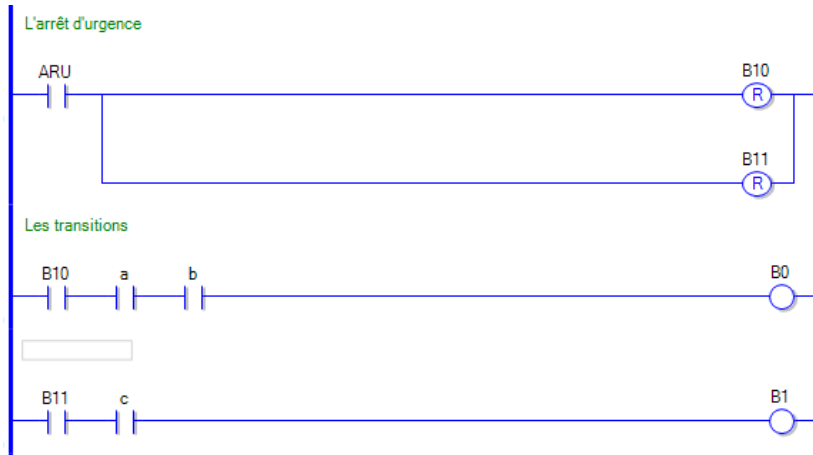
Name	Type	Pin
Xatu	digital in	5
Xdcy	digital in	4
Y00	digital out	19
Rb0	int. relay	
Rb1	int. relay	
Rb10	int. relay	
Rb11	int. relay	
Rb200	int. relay	
Rb201	int. relay	
Rb202	int. relay	

Méthode 1 ladder classic : b200 est à utiliser comme bit relais image du front montant sur dcy.

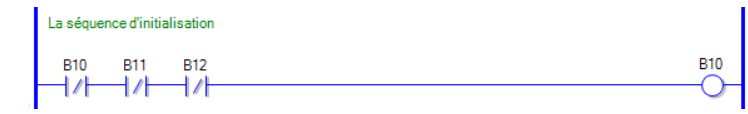
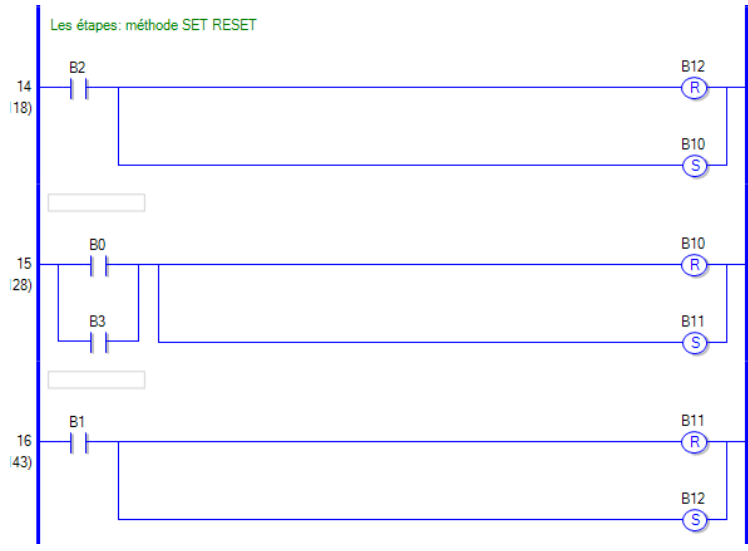
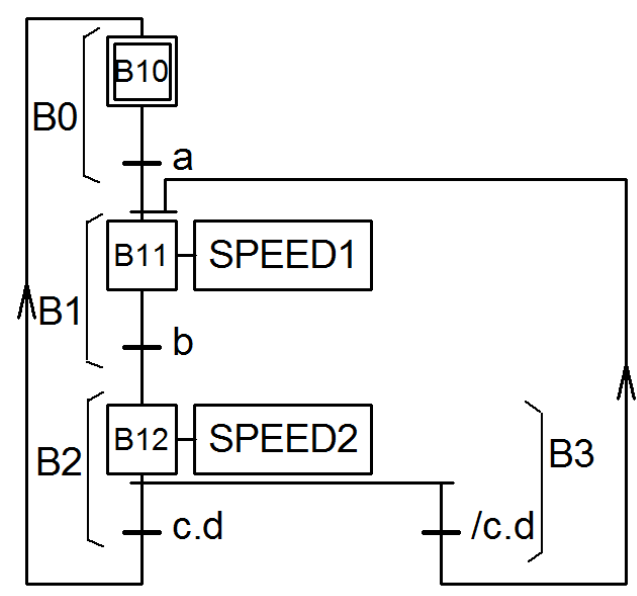
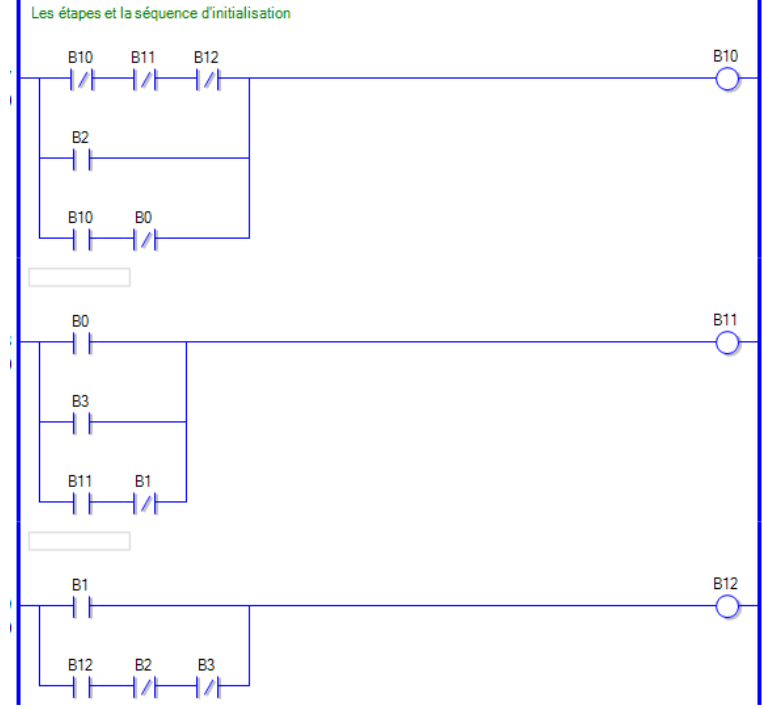
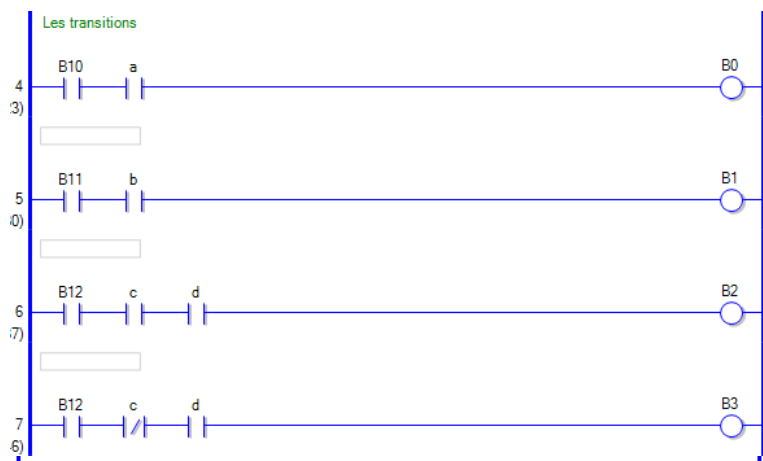
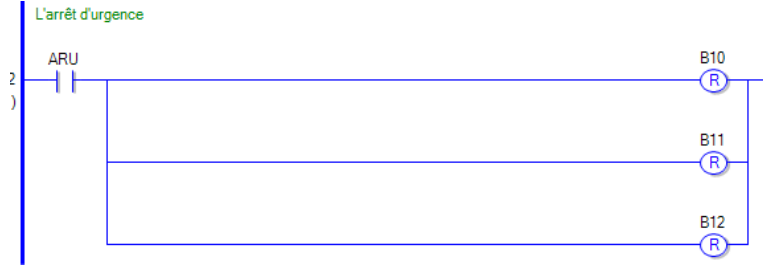
Méthode 2 fonction interne Rising Edge : b202 est à utiliser comme bit relais image du front montant

Tutoriels de codage : GRFACET vers LADDER par 2 méthodes

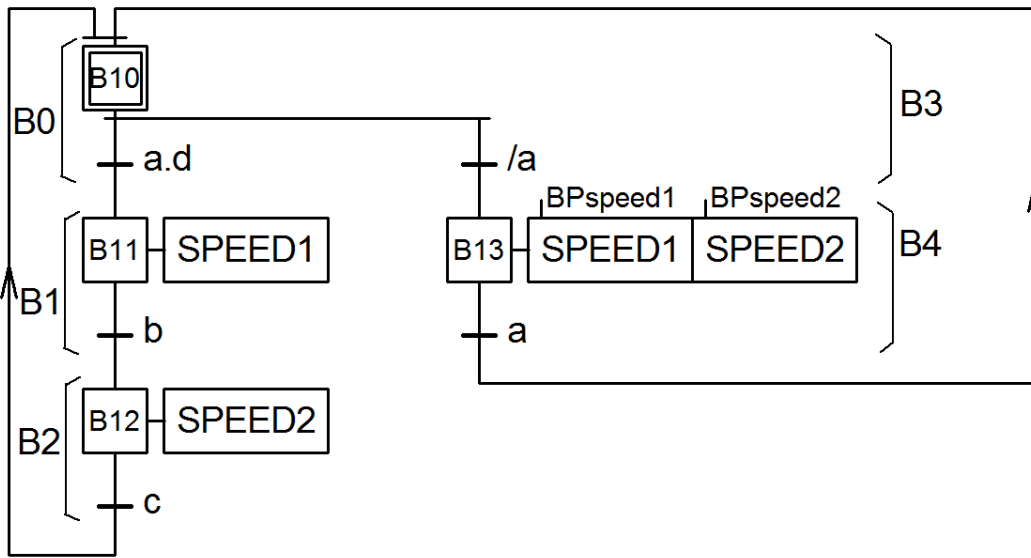
Grafcet 2 étapes :



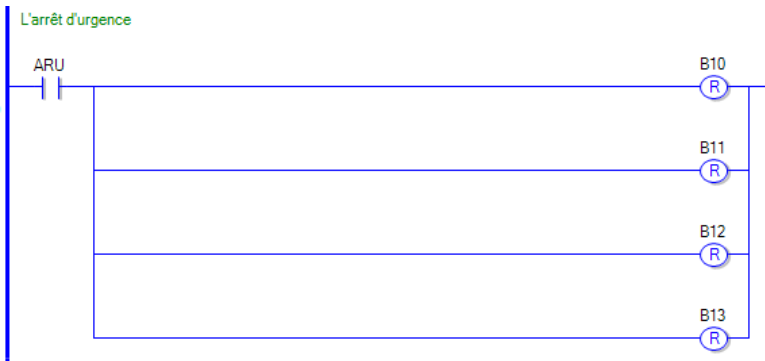
Grafcet 3 étapes :



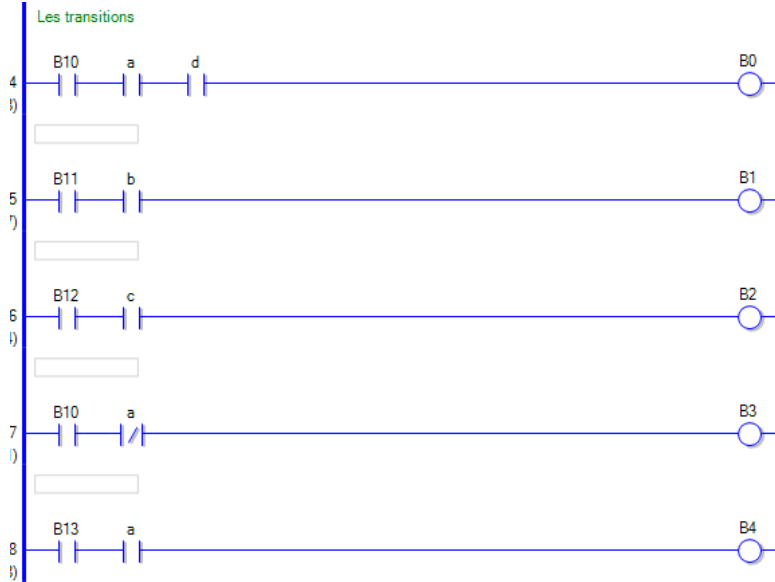
Grafset 4 étapes :



L'arrêt d'urgence

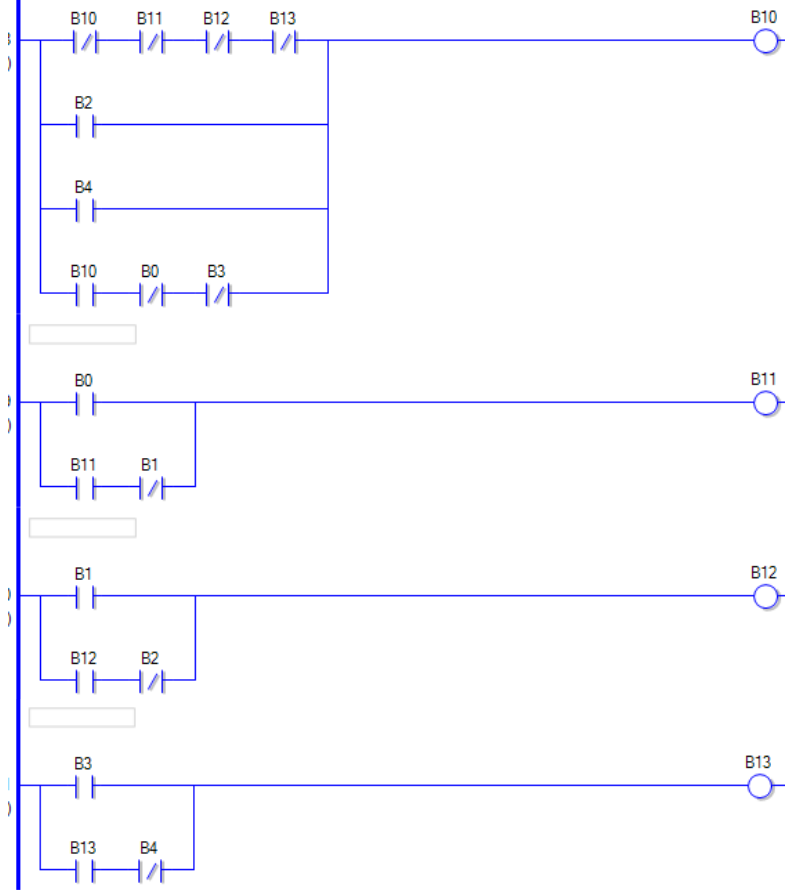


Les transitions

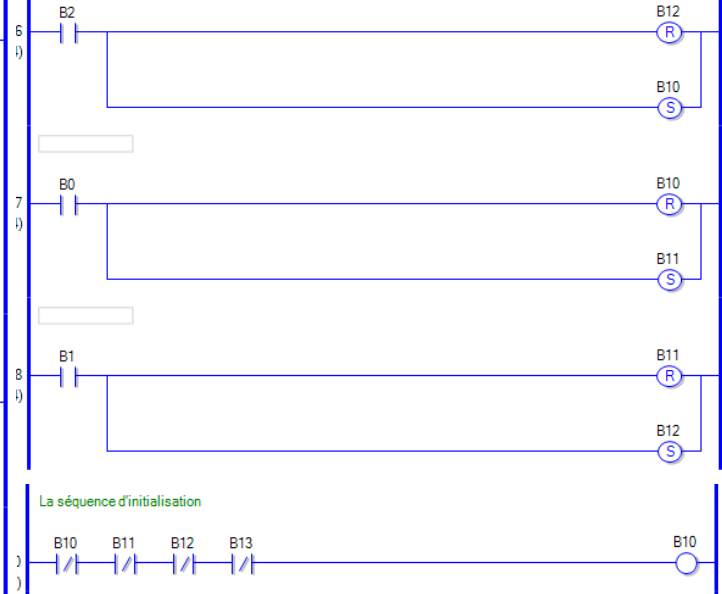


TRI DE BRIQUE MICROCONTROLEUR

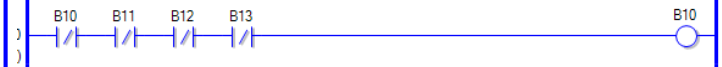
Les étapes et la séquence d'initialisation



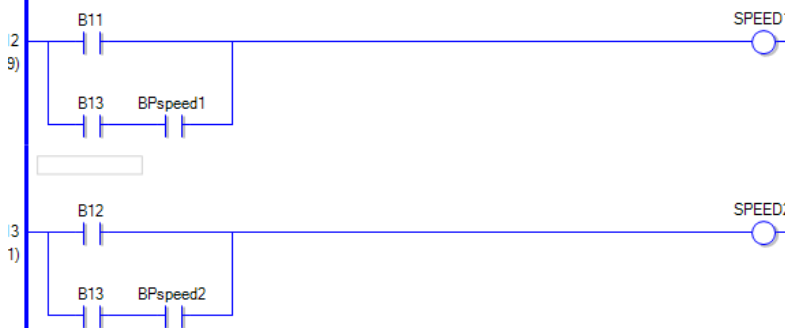
Les étapes: méthode SET RESET



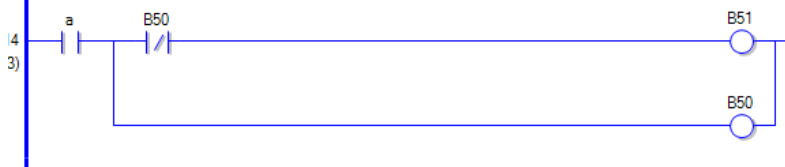
La séquence d'initialisation



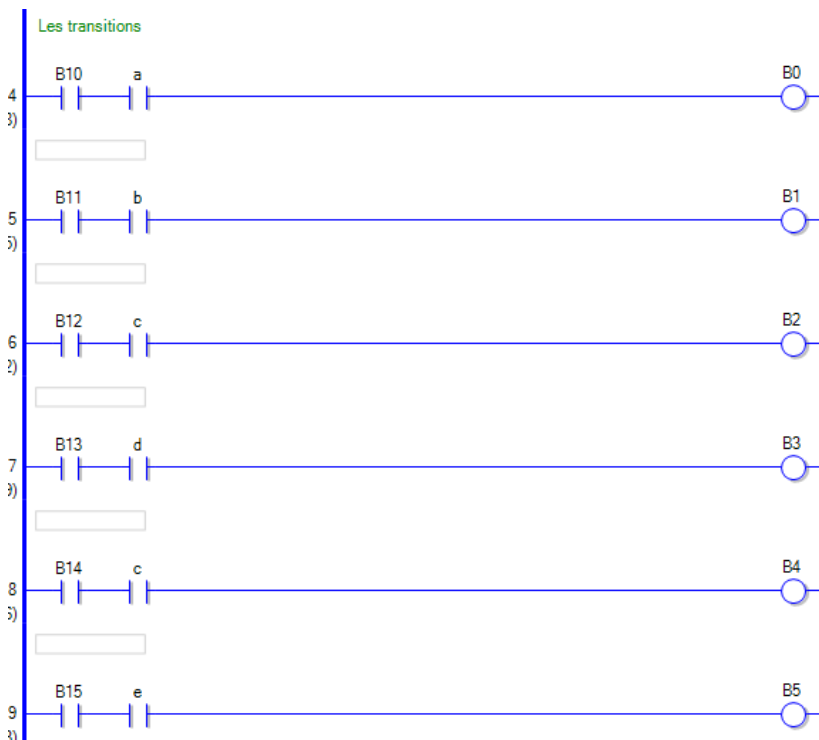
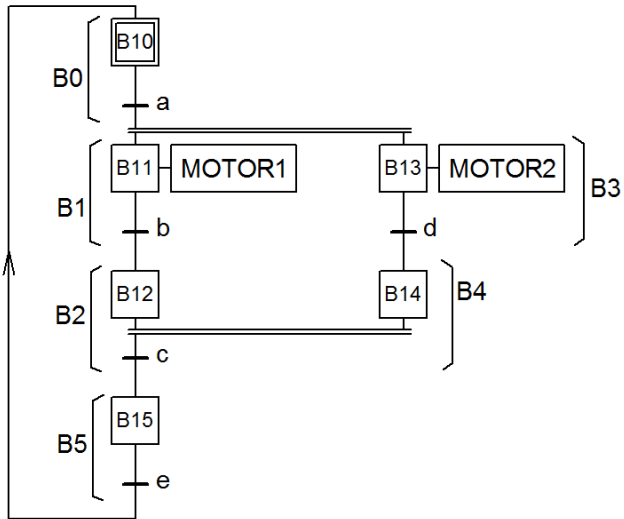
Les actions associées aux étapes



Exemple de front montant sur a: le bit B51 sera en front montant

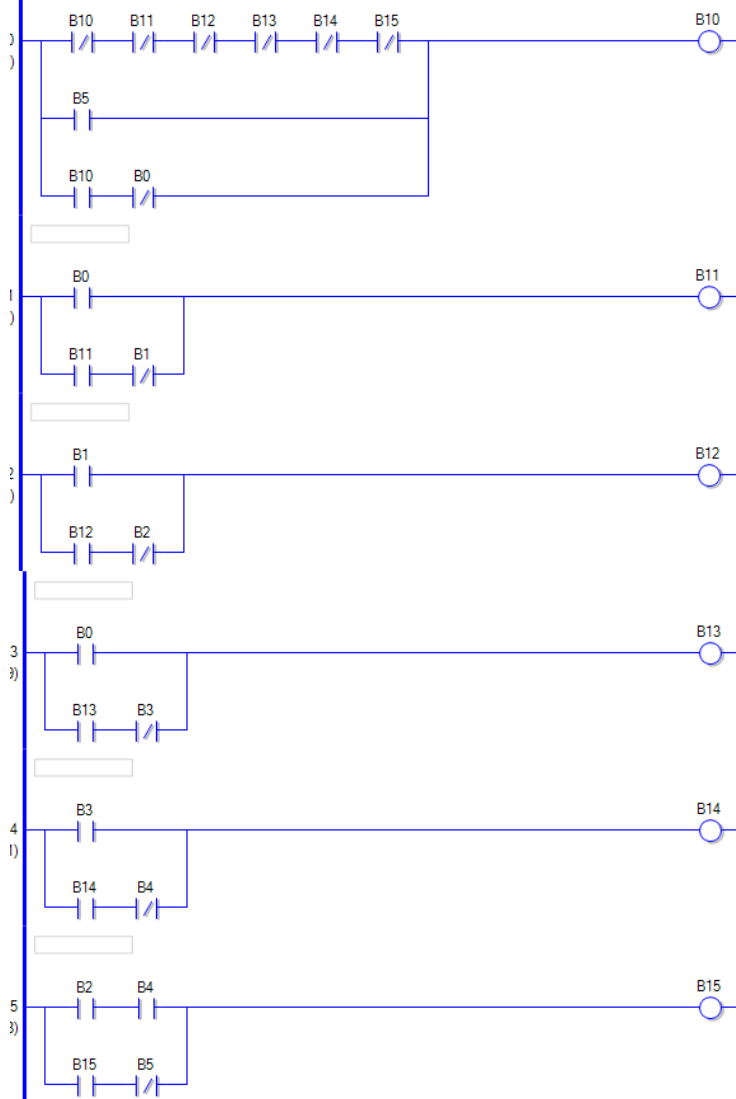


Grafset 6 étapes :

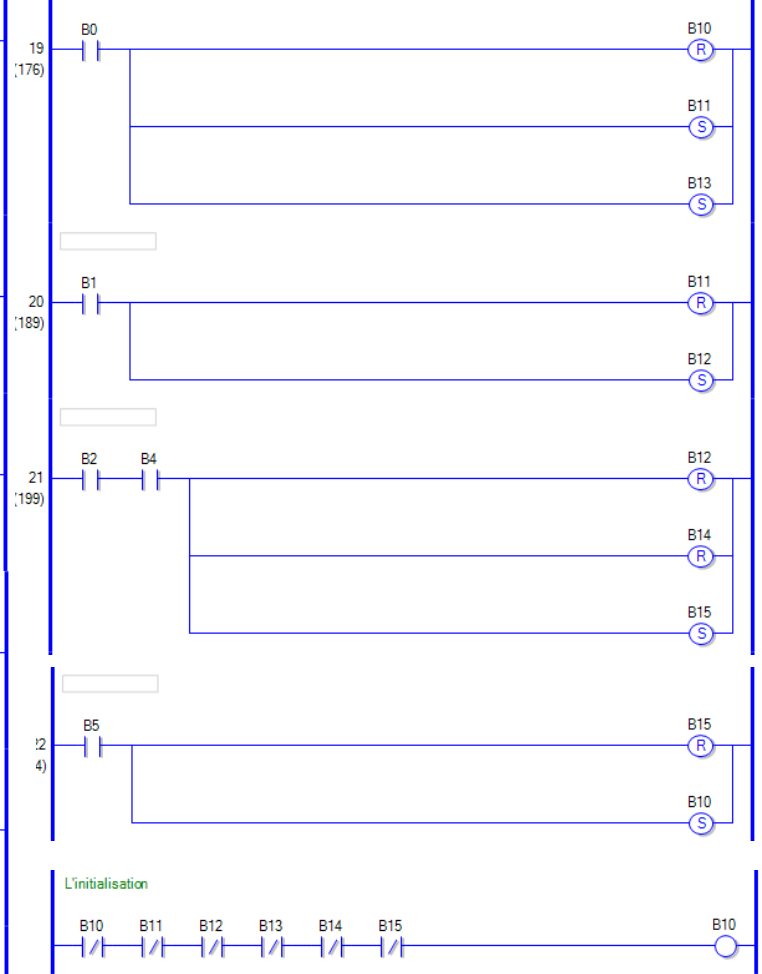


TRI DE BRIQUE MICROCONTROLEUR

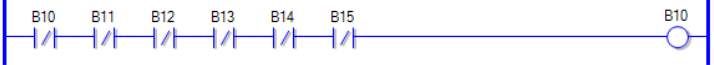
Les étapes et la séquence d'initialisation



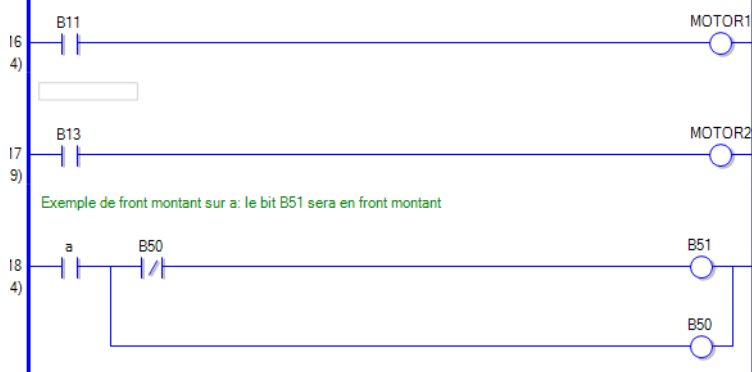
Les étapes: méthode SET RESET



L'initialisation



Les actions associées aux étapes



Exemple de front montant sur a: le bit B51 sera en front montant