

Mini modules automates

« TinyPLC »

Manuel utilisateur
Version 1.1



Traduction Française Copyright Lextronic 2003

Copyrights et appellations commerciales

Windows 98™ et Windows XP™ sont des appellations commerciales appartenant à Microsoft Corporation. Toutes les autres marques, les procédés et les références des produits cités dans ce document appartiennent à leur propriétaire et Fabricant respectif. All brand names and trademarks are the property of their respective owners - Other trademarks mentioned are registered trademarks of their respective holders.

Informations techniques

Ce manuel a été conçu avec la plus grande attention. Tous les efforts ont été mis en oeuvre pour éviter les anomalies. Toutefois, nous ne pouvons garantir que ce dernier soit à 100% exempt de toute erreur. Les informations présentes dans ce manuel sont données à titre indicatif. Les caractéristiques techniques des "TINYPLC", la nature, les possibilités et le nombre de leurs instructions, ainsi que les possibilités de leurs logiciels de programmation et les caractéristiques des modules périphériques associés aux TINYPLC peuvent changer à tout moment sans aucun préavis dans le but d'améliorer la qualité et les possibilités de ces derniers. Ces produits sont protégés par des brevets à travers le monde.

Limitation de responsabilité

En aucun cas le Fabricant et LEXTRONIC ne pourront être tenus responsables de dommages quels qu'ils soient (intégrant, mais sans limitation, les dommages pour perte de bénéfice commercial, interruption d'exploitation commerciale, perte d'informations et de données à caractère commercial ou de toute autre perte financière) provenant de l'utilisation ou de l'incapacité à pouvoir utiliser les modules "TINYPLC" et leurs logiciels associés ainsi que leurs platines et modules optionnels associés, même si le Fabricant ou LEXTRONIC ont été informés de la possibilité de tels dommages.

Les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés sont destinés à être utilisés en milieu résidentiel dans les gammes de températures +10 à +50 °C. Les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés ne sont pas conçus, ni destinés, ni autorisés pour être utilisés au sein d'applications militaires, ni au sein d'applications à caractère médical, ni au sein d'applications d'alerte incendie, ni au sein d'applications pour ascenseurs, ni au sein d'applications sur machine outils, ni au sein d'applications embarquées dans des véhicules (automobiles, camions, bateaux, scooters, motos, kart, scooters des mers, avions, hélicoptères, ULM, etc...), ni au sein d'applications embarquées sur des maquettes volantes de modèles réduits (type avions, hélicoptères, planeurs, etc...).

De même, les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés ne sont pas conçus, ni destinés, ni autorisés pour expérimenter, développer ou être intégrés au sein d'applications dans lesquelles une défaillance de ces derniers pourrait créer une situation dangereuse pouvant entraîner des pertes financières, des dégâts matériels, des blessures corporelles ou la mort de personnes ou d'animaux. Si vous utilisez les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés volontairement ou involontairement pour de telles applications non autorisées, vous vous engagez à soustraire le Fabricant et LEXTRONIC de toute responsabilité et de toute demande de dédommagement.

En cas de litige, l'entière responsabilité du Fabricant et de LEXTRONIC vis-à-vis de votre recours se limitera exclusivement selon le choix du Fabricant et de LEXTRONIC au remboursement du module "TINYPLC" et/ou de ses platines et modules optionnels associés et/ou de leur réparation et/ou de leur échange. Le Fabricant et LEXTRONIC démentent toutes autres garanties, exprimées ou implicites.

L'utilisateur des modules "TINYPLC" et de ses platines et modules optionnels associés est entièrement et seul responsable des développements logiciels (de l'écriture de son programme) ainsi que de l'intégration matérielle, des modifications et ajouts de périphériques qu'il effectuera sur les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés. S'agissant de matériel "OEM", Il incombera à l'utilisateur de vérifier que l'application finie complète développée avec les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés soient conformes aux normes CEM en vigueur et que l'application finale utilise un dispositif complémentaire de sécurité en conformité avec la réglementation en vigueur (permettant le bon fonctionnement des circuits liés à la sécurité - par exemple et de façon non exhaustive la norme EN 954-1 catégorie 4) par le biais de modules de sécurité additionnel type Preventa™ (Schneider electric™) ou KNA3™ (Crouzet™) ou autres équivalents issus d'autres fabricants

Tous les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés sont testés avant expédition. Toute inversion de polarité, dépassement des valeurs limites des tensions d'alimentation, courts-circuits, utilisation en dehors des spécifications et limites indiquées dans ce document ou utilisation pour des applications non prévues pourront affecter la fiabilité, créer des dysfonctionnements et/ou endommager les modules "TINYPLC" ainsi que leurs platines et modules optionnels associés sans que la responsabilité du Fabricant et de LEXTRONIC ne puisse être mise en cause, ni que les produits puissent être échangés au titre de la garantie.

Rappel sur l'évacuation des équipements électroniques usagés

Ce symbole présent sur les modules « TINYPLC » ainsi que leurs platines et modules optionnels associés et/ou leurs emballages indique que vous ne pouvez pas vous débarrasser de ces produits de la même façon que vos déchets courants. Au contraire, vous êtes responsable de l'évacuation de ces produits lorsqu'ils arrivent en fin de vie (ou qu'ils sont hors d'usage) et à cet effet, vous êtes tenu de le remettre à un point de collecte agréé pour le recyclage des équipements électriques et électroniques usagés. Le tri, l'évacuation et le recyclage séparés de vos équipements usagés permettent de préserver les ressources naturelles et de s'assurer que ces équipements sont recyclés dans le respect de la santé humaine et de l'environnement. Pour plus d'informations sur les lieux de collecte des équipements électroniques usagés, veuillez contacter votre mairie ou votre service local de traitement des déchets.



Note for all residents of the European Union

This symbol on the product or on its packaging indicates that this product must not be disposed of with other household waste. Instead, it is your responsibility to dispose of your waste equipment by handing it over to designated collection point for the recycling of waste electrical and electric equipment. The separate collection and recycling of your waste equipment at the time of disposal will help



to conserve natural resources and ensure that it is recycled in a manner that protects human health and environment. For more information about where you can drop off your waste equipment for recycling, please contact your local city office or your local household waste disposal service.

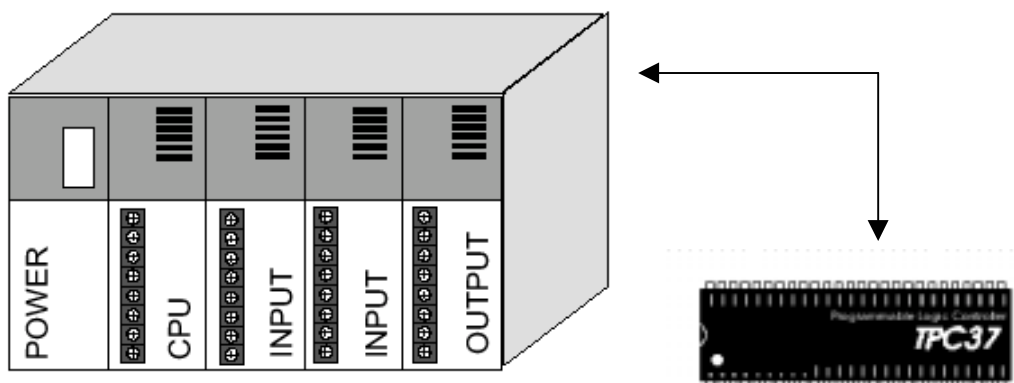
Introduction

La série TPC 3X s'apparente à une gamme de microcontrôleurs très efficaces avec beaucoup de capacité, elle peut exécuter de façon plus rapide, des fonctions plus variées que l'ancienne série TPC 0X. Vous pouvez voir ci-dessous les principales caractéristiques de la série TPC-3X ... :

- Création de programme à partir d'un programme graphique (LADDER)
- Modification en accord avec la configuration des entrées/sorties
- Sauvegarde du programme en mémoire flash, sans batterie de backup
- Avec le programme MGPL2 fourni gratuitement, création du programme sur PC et téléchargement dans le TinyPLC via l'interface RS232C.
- Suivi du programme en temps réel, en cours de mise au point

TinyPLC est d'une conception différente par rapport aux automates programmables de conception classiques (PLC). Par comparaison aux autres PLC, voici quelques différences :

- Type circuit intégré, TinyPLC peut être monté sur circuit imprimé
- Faible coût, diminue le prix de fabrication de l'application finale
- Sub-miniature, la taille du produit final est considérablement réduite



En plus des fonctions de base supportées par TinyPLC, vous pouvez utiliser des fonctions performantes, qui nécessitent des unités spécialisées dans les PLC standards :

- Compteur 16 bits 30khz installé
- Octuple convertisseur A/D 10 bits installé
- Fonction, commande de module alphanumérique LCD 2 lignes 16 caractères à 4 lignes 20 caractères, installée
- Fonction de commande de module 7 segments installée (5 digits)
- Communication multi usage avec la PC (RS232 ou 485)
- Fonction de commande à distance des entrées sorties
- Fonction pour contrôler un clavier matricé de 8x8 touches
- Fonction pour connecter un DS1820 (thermomètre digital)

TinyPLC, comme décrit ci-dessus, est un processeur monochip très performant intégrant les fonctions des PLC, dits standards, dans un circuit intégré. Comfile technology est très fier de ce nouveau produit. Vous obtiendrez de bons résultats en l'utilisant.

Chapitre 1

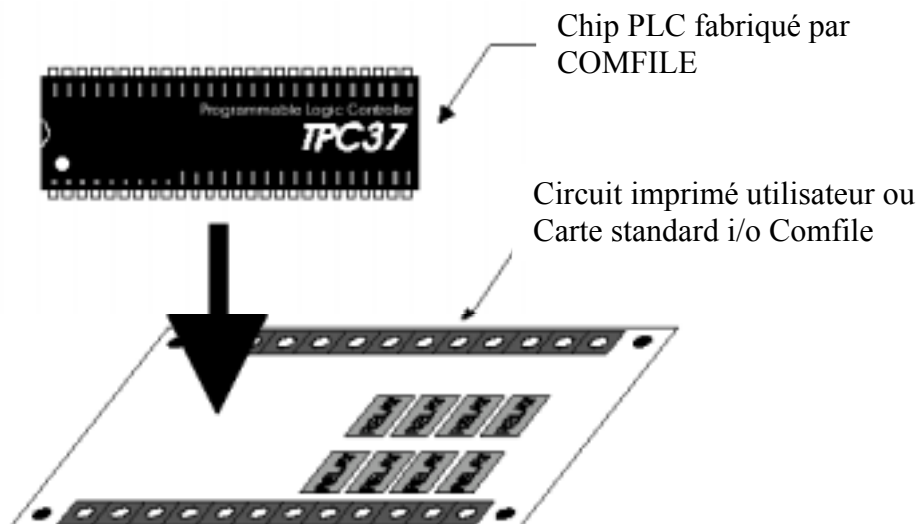
Presentation

des

TinyPLC

Introduction à TinyPLC

Comme vu précédemment le concept de TinyPLC est totalement différent des PLC (aussi appelés API, automates programmables industriels). Le montage se fait sur un circuit imprimé en ajoutant des circuits entrées/sorties pour faire un automate convenant à notre utilisation.



Cette méthode procure de nombreux avantages pour l'utilisateur :

- La majorité des entrées/sorties, exception de la CPU, est conçue et fabriquée par l'utilisateur, il peut ainsi faire un PLC à sa convenance
- Si vous choisissez un module CPU seul, comparé à un PLC standard, vous diminuez le coût, profitable pour la grande production
- Vous pouvez choisir le sens des entrées/sorties, pour la majorité d'entre elles
- Vous pouvez réduire le matériel au strict nécessaire et connecter les appareils extérieurs, le produit fini peut être très compact

Par sa conception et ses possibilités, TinyPLC a été conçu comme une machine de production de masse, où les utilisateurs développent eux même les cartes d'entrées/sorties et d'alimentation. Mais Comfile développe aussi une gamme de modules alimentation et de cartes entrées/sorties

Modèle	Description	Entrées/sorties
BASE-C20R	Carte pour série TPC33 et 37 20 entrées/sorties RS232C installé Ports pour LCD et SGN installés	12 entrées 24v DC 8 sorties relais
BASE-C64R	Carte pour TPC38 64 entrées/sorties RS232 et 485 Ports LCD et SGN installés	24 entrées 24 v DC 24 sorties relais 16 cellules I/O
EIO-DI8-24A	Module d'entrée connecté au TPC-BUS	8 entrées 24v DC
EIO-DI8-12A	Module d'entrée connecté au TPC-BUS	8 entrées 12v DC
EIO-RY8A	Module de sortie connecté au TPC-BUS	8 sorties relais
EIO-TR8A	Module de sortie connecté au TPC-BUS	8 Sorties transistors

Caractéristiques des TPC 3X

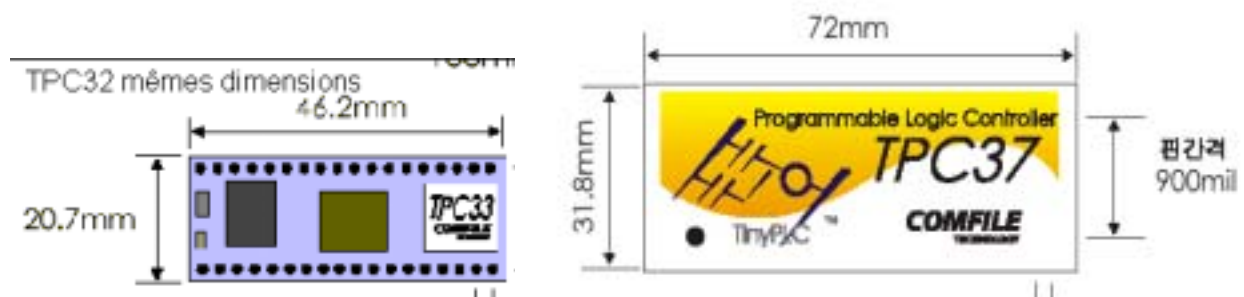
	TPC26	TPC 33	TPC 37	TPC38
Mémoire programme	16 kb	16 kb	128 kb	128 kb
Temps de cycle	2.5 ms	2.5 ou 5 ms	2.5 ou 5 ms	5 ms
Commandes de base	27	27	27	27
Commandes d'applications	84	84	84	84
E/S (P)	26	24	24 E / S 8 Ent + 8 Sort	24 E / S 20 E + 24 S
Sorties D/A 10 bits				2 PWM 10Bits
Rel. Auxil. (M)	256	256	1024	1024
Contrôle pas à pas (S)	16 de 255 pas	16 de 255 pas	32 de 255	32 de 255
Relais. Sauveg.(K)	512	512	512	512
Temporisateurs (T)	64	64	256	256
Compteurs (C)	32	32	256	256
Données (D)	220	220	1024	1024
Buffer. LCD (CH)	20 x 4	20 x 4	20 x 4	20 x 4
Buffer affichage SGN (G)	5 x 8	5 x 8	5 x 8	5 x 8
Convert A/D sur 10 bits (AD)	8	8	8	8
Compteur rapide (CNT)	1 (16 bits)	1 (16 bits)	1 (16 bits)	1 (16 bits)

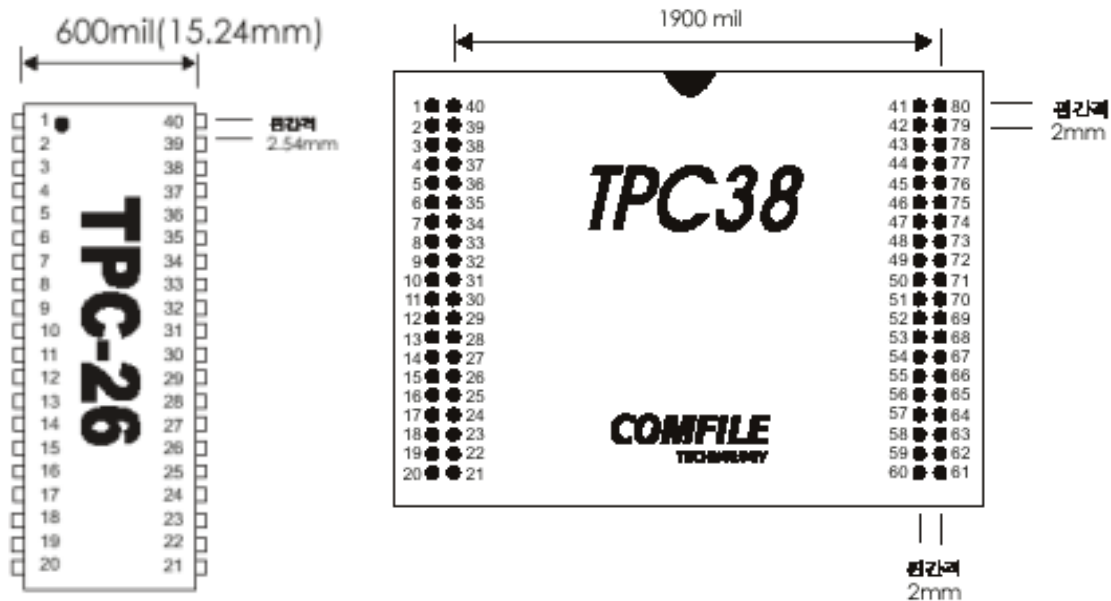
TPC32 ne figure pas dans le tableau

SGN : Module d'affichage 7 segments de Comfile

LCD : Module LCD série de Comfile

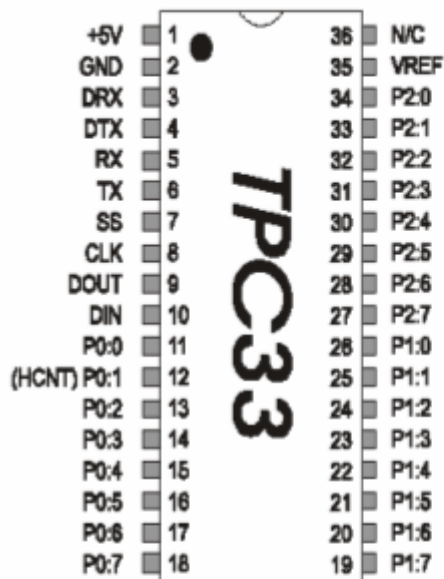
Figures :





Brochage des TPC 3X

TPC32 / 33



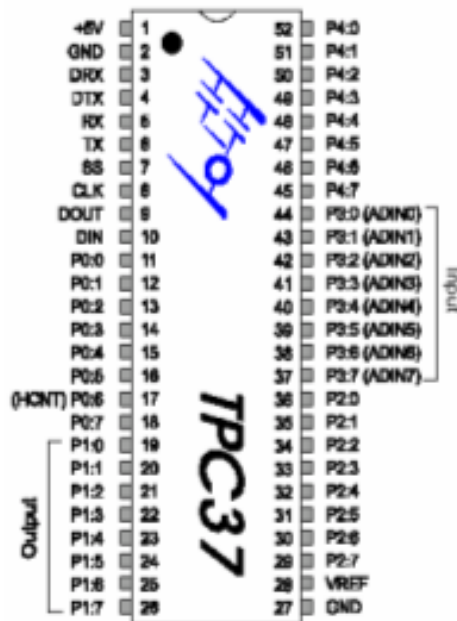
N° Pin	Nom	Fonction
1	+ 5 v	Alim
2	GND	Alim
3	DRX	Entrée
4	DTX	Sortie
5	RX	Entrée
6	TX	Sortie

N° Pin	Nom	Fonction
7	485TE	Sortie
8	CLK	Sortie
9	Dout	Sortie
10	Din	Entrée
11	P0:0	E / S
12	P0:1/HCNT	E / S
13	P0:2	E / S
14	P0:3	E / S
15	P0:4	E / S
16	P0:5	E / S
17	P0:6	E / S
18	P0:7	E / S
19	P1:7	E / S
20	P1:6	E / S
21	P1:5	E / S
22	P1:4	E / S
23	P1:3	E / S
24	P1:2	E / S
25	P1:1	E / S
26	P1:0	E / S
27	P2:7	E / S
28	P2:6	E / S

TPC 33 suite

N° Pin	Nom	Fonction
29	P2:5	E / S
30	P2:4	E / S
31	P2:3	E / S
32	P2:2	E / S

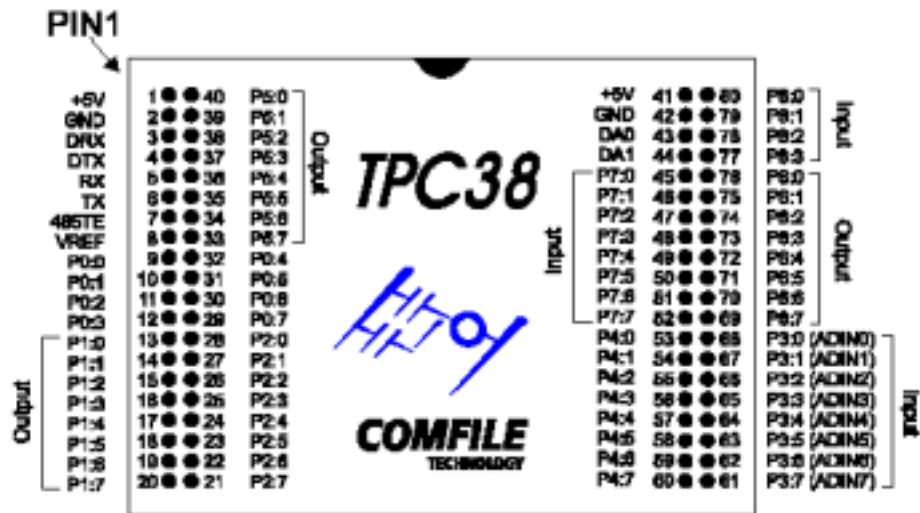
N° Pin	Nom	Fonction
33	P2:1	E / S
34	P2:0	E / S
35	V réf	Ent
36	NC	

TPC 37

N° Pin	Nom	Fonction
1	+ 5V	Alim
2	Gnd	Alim
3	DRX	Entrée
4	DTX	Sortie
5	RX	Entrée
6	TX	Sortie
7	485TE	Sortie
8	CLK	Sortie
9	Dout	Sortie
10	Din	Entrée
11	P0:0	E / S
12	P0:1	E / S
13	P0:2	E / S
14	P0:3	E / S
15	P0:4	E / S
16	P0:5	E / S
17	P0:6	E / S
18	P0:7	E / S

N° Pin	Nom	Fonction
19	P1:0	Sortie
20	P1:1	Sortie
21	P1:2	Sortie
22	P1:3	Sortie
23	P1:4	Sortie
24	P1:5	Sortie
25	P1:6	Sortie
26	P1:7	Sortie
27	GND	Alim
28	V réf	Entrée
29	P2:7	E / S
30	P2:6	E / S
31	P2:5	E / S
32	P2:4	E / S
33	P2:3	E / S
34	P2:2	E / S
35	P2:1	E / S
36	P2:0	E / S
37	P3:7	Ent ADin7
38	P3:6	Ent ADin6
39	P3:5	Ent ADin5
40	P3:4	Ent ADin4
41	P3:3	Ent ADin3
42	P3:2	Ent ADin2
43	P3:1	Ent ADin1
44	P3:0	Ent ADin0
45	P4:7	E / S
46	P4:6	E / S
47	P4:5	E / S
48	P4:4	E / S
49	P4:3	E / S
50	P4:2	E / S
51	P4:1	E / S
52	P4:0	E / S

TPC 38



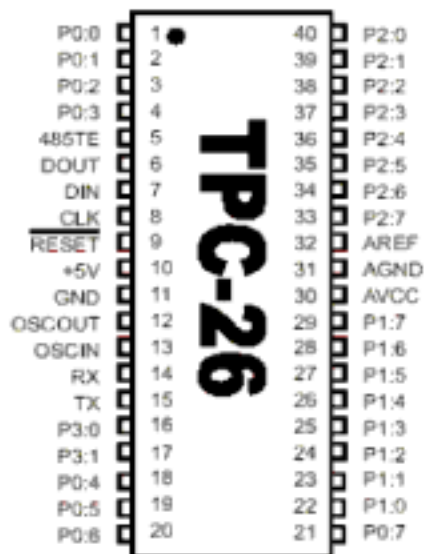
N° Pin	Nom	Fonction
1	+ 5V	Alim
2	Gnd	Alim
3	DRX	Entrée
4	DTX	Sortie
5	RX	Entrée
6	TX	Sortie
7	485TE	Sortie
8	V réf	Entrée
9	P0:0	E / S
10	P0:1	E / S
11	P0:2	E / S
12	P0:3	E / S
13	P1:0	Sortie
14	P1:1	Sortie
15	P1:2	Sortie
16	P1:3	Sortie
17	P1:4	Sortie
18	P1:5	Sortie
19	P1:6	Sortie
20	P1:7	Sortie
21	P2:7	E / S
22	P2:6	E / S
23	P2:5	E / S
24	P2:4	E / S
25	P2:3	E / S
26	P2:2	E / S
27	P2:1	E / S
28	P2:0	E / S
29	P0:7	E / S
30	P0:6	E / S

N° Pin	Nom	Fonction
31	P0:5	E / S
32	P0:4	E / S
33	P5:7	Sortie
34	P5:6	Sortie
35	P5:5	Sortie
36	P5:4	Sortie
37	P5:3	Sortie
38	P5:2	Sortie
39	P5:1	Sortie
40	P5:0	Sortie
41	+ 5 V	Alim
42	GND	Alim
43	DA 0	Sortie
44	DA 1	Sortie
45	P7:0	Entrée
46	P7:1	Entrée
47	P7:2	Entrée
48	P7:3	Entrée
49	P7:4	Entrée
50	P7:5	Entrée
51	P7:6	Entrée
52	P7:7	Entrée
53	P4:0	E / S
54	P4:1	E / S
55	P4:2	E / S
56	P4:3	E / S
57	P4:4	E / S
58	P4:5	E / S
59	P4:6	E / S
60	P4:7	E / S

TPC 38 (suite)

N° Pin	Nom	Fonction
61	P3:7	Entrée
62	P3:6	Entrée
63	P3:5	Entrée
64	P3:4	Entrée
65	P3:3	Entrée
66	P3:2	Entrée
67	P3:1	Entrée
68	P3:0	Entrée
69	P6:7	Sortie
70	P6:6	Sortie

N° Pin	Nom	Fonction
71	P6:5	Sortie
72	P6:4	Sortie
73	P6:3	Sortie
74	P6:2	Sortie
75	P6:1	Sortie
76	P6:0	Sortie
77	P8:3	Entrée
78	P8:2	Entrée
79	P8:1	Entrée
80	P8:0	Entrée

TPC 26

N° Pin	Nom	Fonction
1	P0:0	E / S
2	P0:1	E / S
3	P0:2	E / S
4	P0:3	E / S
5	485TE	Sortie
6	Dout	Sortie
7	Din	Entrée
8	Clk	Sortie
9	/Reset	Entrée
10	+ 5 v	Alim
11	Gnd	Alim
12	OSCout	Sortie

N° Pin	Nom	Fonction
13	OSCin	Entrée
14	RX	Entrée
15	TX	Sortie
16	P3:0	E / S
17	P3:1	E / S
18	P0:4	E / S
19	P0:5	E / S
20	P0:6	E / S
21	P0:7	E / S
22	P1:0	E / S
23	P1:1	E / S
24	P1:2	E / S
25	P1:3	E / S
26	P1:4	E / S
27	P1:5	E / S
28	P1:6	E / S
29	P1:7	E / S
30	A Vcc	Alim
31	A Gnd	Alim
32	A Réf	Entrée
33	P2:7	E / S
34	P2:6	E / S
35	P2:5	E / S
36	P2:4	E / S
37	P2:3	E / S
38	P2:2	E / S
39	P2:1	E / S
40	P2:0	E / S

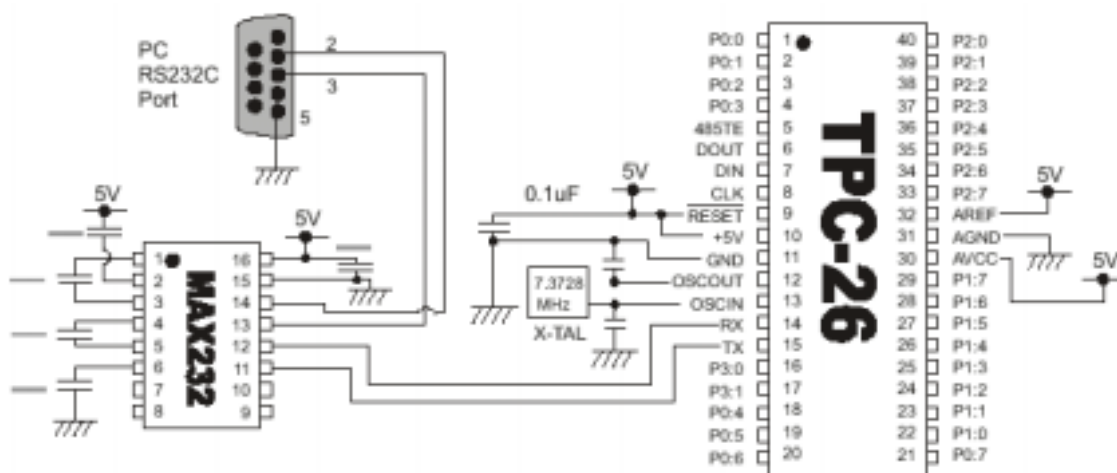
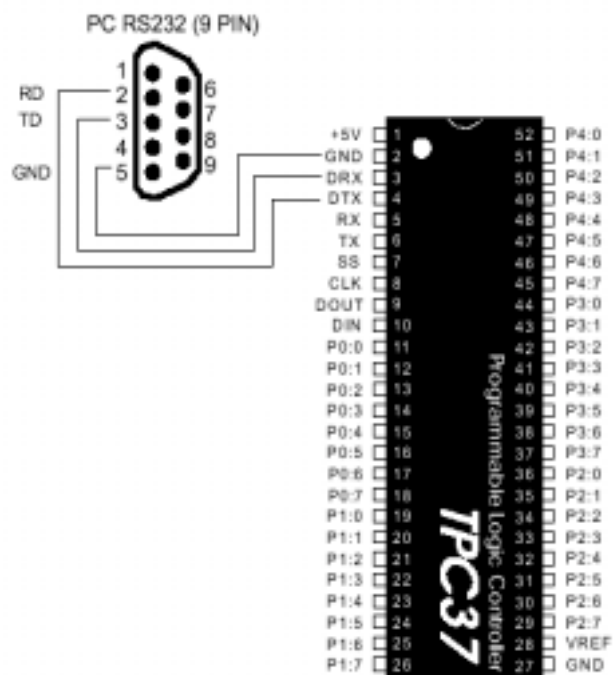
La fréquence du quartz à connecter aux broches 12 et 13 est de 7.3728 mhz

Connexion avec un PC

Broche PLC	RS232C PC 9	RS232C PC25
2 GND	5 GND	7 GND
3 DRX	3 TD	2 TD
4 DTX	2 RD	3 RD

Les broches DRX et DTX (broches 3 et 4) sont utilisées pour le téléchargement et le fonctionnement pas à pas, elles sont connectées directement aux broches RD et TD d'une RS232C d'un PC sans convertisseur spécial. (Il est installé à l'intérieur du TPC 33 37 et 38)

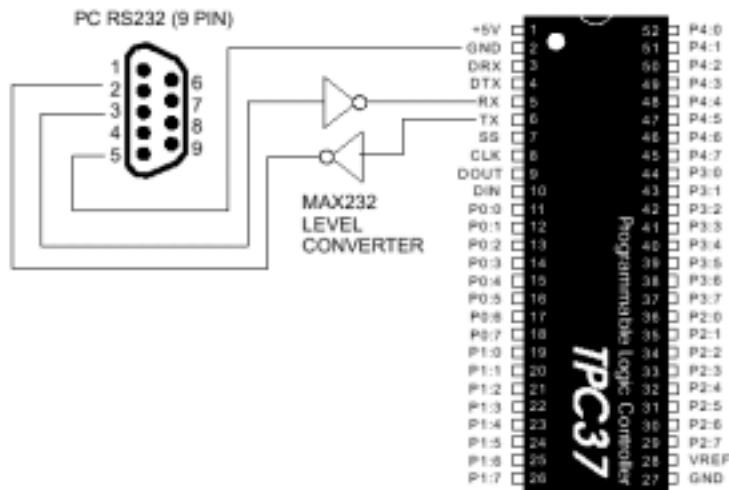
Le protocole de communication est de 19200,N,8,1



Le TPC 26 est construit sur une seule puce et ne contient pas de convertisseur de tension d'où la présence d'un convertisseur MAX 232. Les condensateurs du MAX 232 peuvent être de 1µF.

Liaison avec un ordinateur

TinyPLC est aussi utilisable en liaison avec un PC pour lire ou écrire des données du TinyPLC par une RS232C ou RS485. Avec cette fonction, il est possible de centraliser le contrôle sur le PC, les broches RX et TX utilisées sur le PLC sont 5 et 6, le protocole est fixé à 9600 bauds,N,8,1

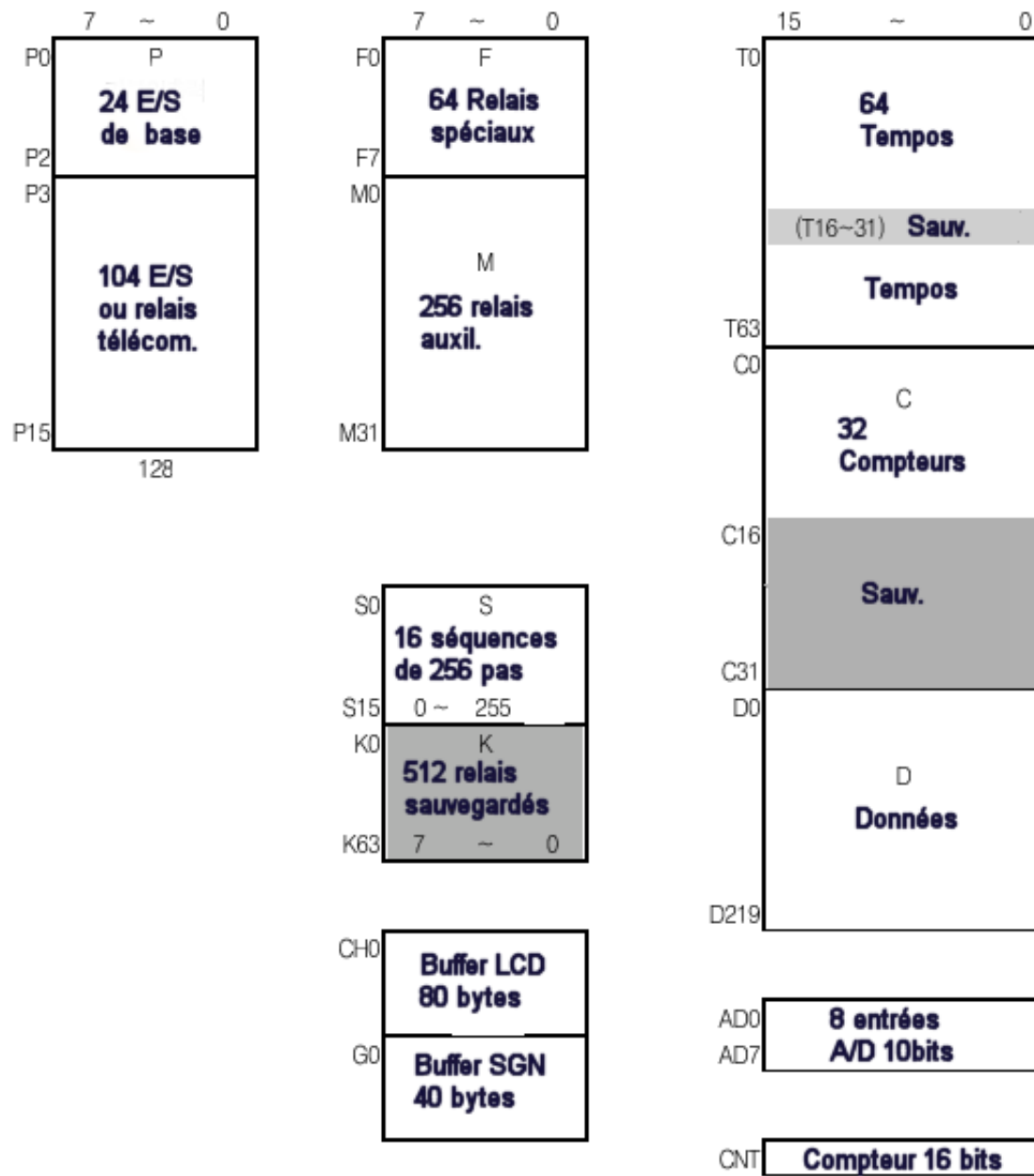


PLC	RS232 9 br	RS232 25 br
2 GND	5 GND	7 GND
5 RX	3 TD	2 TD
6 TX	2 RD	3 RD

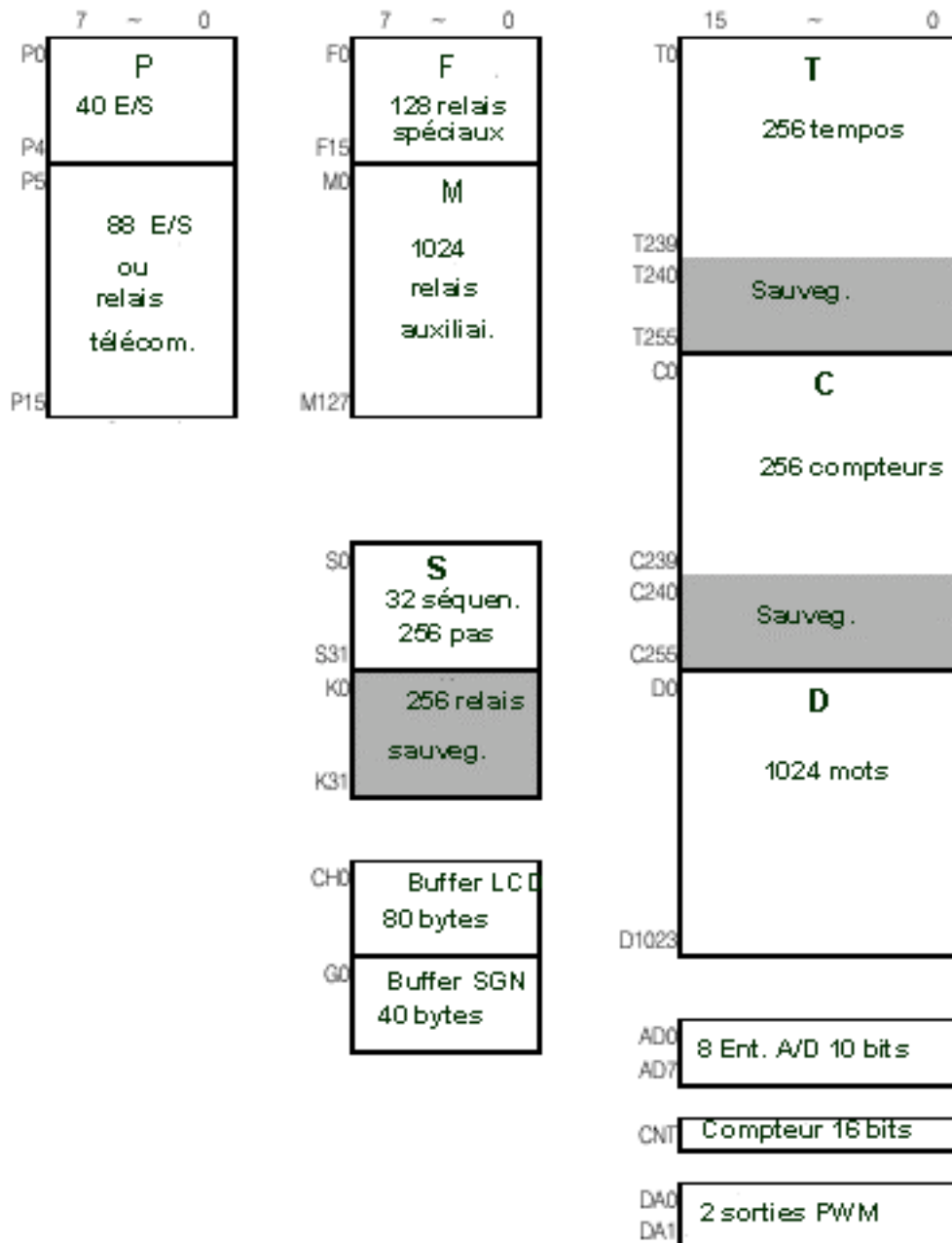
Il faut monter un convertisseur de niveau MAX 232 pour RS232C ou LTC 485 pour RS485, celui-ci n'est pas incorporé dans le circuit

Organisation mémoire de données

TPC 26 et 33



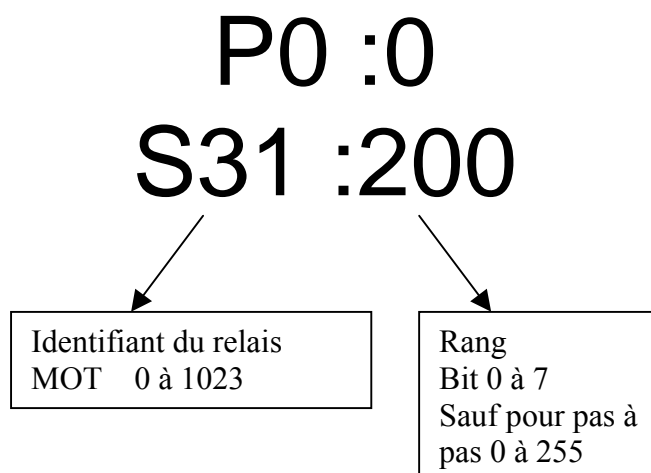
TPC 37 et 38



Les valeurs indiquées dans le tableau ci-dessus correspondent au TPC 37, sauf les DA0 et DA1 (sur 10 bits) qui sont spécifiques au TPC 38 . En outre le TPC 38 possède 64 entrées/sorties de base et donc 64 E/S télécommandables

RELAIS et RELAIS SPECIAUX

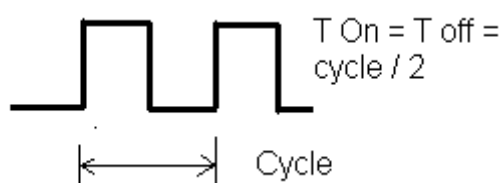
Tous les types de relais de TinyPLC sont affichés dans le format ci-dessous :



Type de relais	Ident.	Valeurs MOT	Valeur BIT	Fonction
Relais E/S	P	0-15	0-7	ŌN/OFF des ports E/S ou lecture de la broche
Relais spéciaux	F	0-15	0-7	Fonctions spéciales
Relais aux . internes	M	0-127	0-7	
Relais sauvegardés	K	0-31	0-7	Gardent leurs valeurs même alimentation coupée
Temporisateurs	T	0-255		Lecture seule
Compteurs	C	0-255		Lecture seule
Données	D	0-1023		Lecture seule
Relais pas à pas	S	0-31	0-255	Relais pour pas à pas séquentiels , 32 relais de 256 pas

RELAIS SPECIAUX

Les relais spéciaux permettent d'utiliser les signaux de la CPU
Beaucoup de signaux ont un temps de cycle spécifié de la forme suivante



Relais Spéciaux		Fonction	Relais Spéciaux		Fonction
F0	F0 :0	Toujours à OFF	F4	F4 :0	Cycles de 1 seconde 0.5 s ON 0.5 s OFF
	F0 :1	Toujours à ON		F4 :1	Cycles de 2 secondes 1 s ON 1s OFF
	F0 :2	ON pendant 1 cycle à la mise sous tension		F4 :2	Cycles de 4 secondes 2 s ON 2s OFF
	F0 :3	OFF pendant 1 cycle à la mise sous tension		F4 :3	Cycles de 8 secondes 4 s ON 4 s OFF
	F0 :4		F5	F5 :0	Cycles de 10 secondes 5 s ON 5 s OFF
	F0 :5			F5 :1	Cycles de 20 secondes 10 s ON 10 s OFF
	F0 :6			F5 :2	Cycles de 40 secondes 20 s ON 20 s OFF
	F0 :7			F5 :3	Cycles de 80 secondes 40 s ON 40 s OFF
F1					
F2	F2 :0	Cycles de 0.01 s ON/OFF à chaque scrutation du programme	F6	F6 :0 F6 :1 F6 :2 F6 :3	Pour contrôle de d'écriture sur LCD (coréen) 0 et 1 petits caractères 2 et 3 grands caractères
	F2 :1	Cycles de 0.02 seconde 0.01 s à ON, 0.01 s à OFF	F7		Adressage RS485 0 à 255
	F2 :2	Cycles de 0.04 seconde 0.02 s ON 0.02 s OFF	F8		RTC 0 à 59 (voir programme exemple)
	F2 :3	Cycles de 0.08 s 0.04 s ON 0.04 s OFF	F9		RTC 0 à 59 (voir programme exemple)
F3	F3 :0	Cycles de 0.1 s 0.05 s ON 0.05 s OFF	F10		RTC 0 à 255 (voir programme exemple)
	F3 :1	Cycles de 0.2 s 0.1 s ON 0.1 s OFF			
	F3 :2	Cycles de 0.4 s 0.2 s ON 0.2 s OFF			
	F3 :3	Cycles de 0.8 s 0.4 s ON 0.4 s OFF			

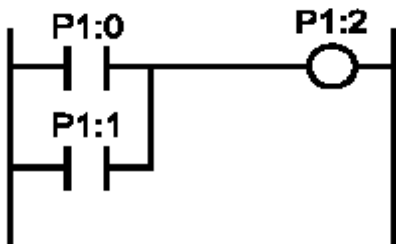
Notes :

Chapitre 2

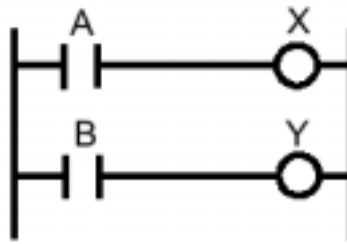
Bases des PLC

Un PLC c'est quoi, structure générale

Les PLC (Programmable Logic Controller) , ou automates programmables , sont utilisés par de nombreuses personnes, et assurent la majorité des automatismes dans l'industrie . Les PLC sont programmés souvent par un 'Relay Ladder Diagram ' qui est un langage de programmation graphique facile à comprendre . La majorité des non électroniciens le comprennent et l'applique après une formation relativement courte .



Exemple de Relay Ladder Diagram

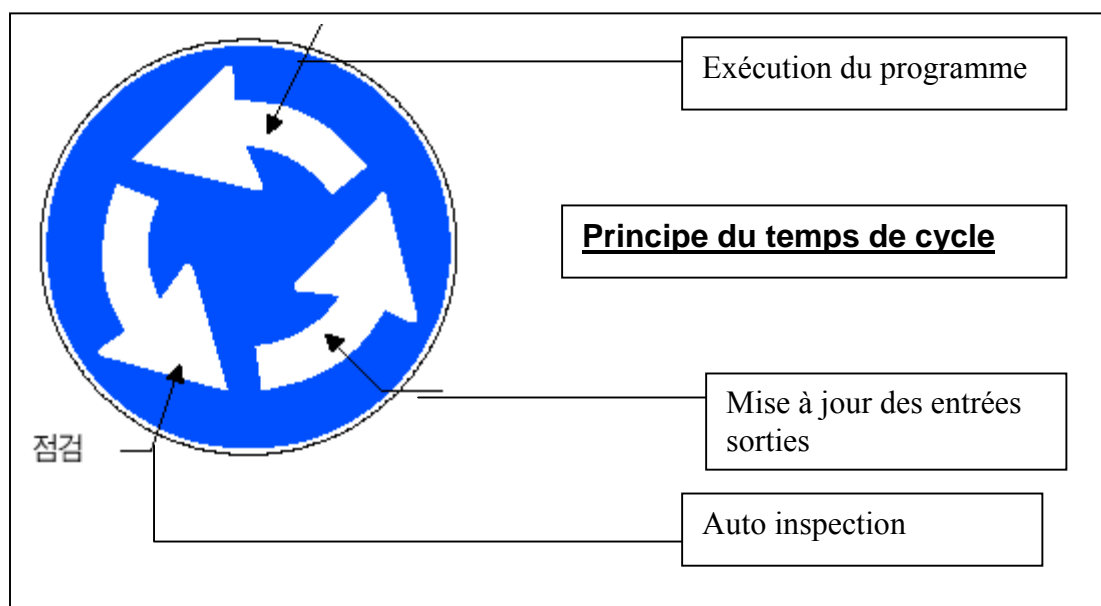


Les deux circuits travaillent de façon indépendante sans interférence .

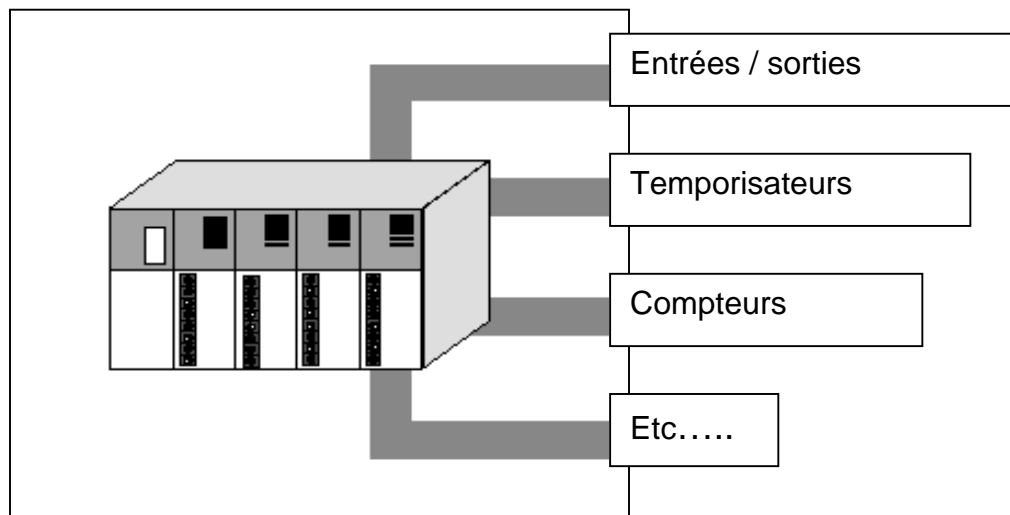


La plus grande distinction entre un PLC et des langages comme le C, l'assembleur ou le Basic est que les PLC sont multitâches, plusieurs actions sont accomplies simultanément , alors que les langages généraux sont séquentiels , les actions sont accomplies les unes à la suite des autres, une seule à la fois .

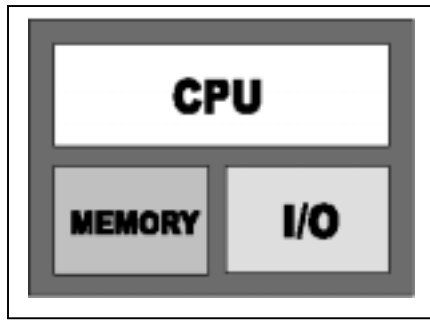
Pour travailler correctement et faire correspondre le fonctionnement des entrées et des sorties entre elles, dans des utilisations qui peuvent être complexes, il est nécessaire d'utiliser le principe de temps de cycle. Le temps de cycle est un intervalle de temps fixe pendant lequel le programme est lu du début jusqu'à la fin, quelle que soit sa longueur, le programme est lu à chaque temps de cycle .



Comme les autres processeurs, un PLC contrôle et fait fonctionner différents types de périphériques. Il se différencie des autres processeurs par le fait qu'il se substitue à des relais dans les équipements d'automatisme et qu'il peut effectivement contrôler différents équipements sur des lignes de production. Par exemple, si le PLC contrôle des convoyeurs, il valide chaque entrée (détecteur de proximité, de température ou etc...) et commande les sorties (vannes moteurs, etc....) en correspondance du programme.



Les domaines d'utilisation des PLC sont très vastes parmi lesquels l'on peut citer :
Contrôle des volets roulants automatiques, robots, machines outils etc, dans pratiquement tous les domaines, il est possible d'utiliser des PLC.



Structure générale d'un PLC

La CPU lit les instructions du programme, les interprète et les exécute, les entrées/sorties sont chargées de faire la liaison avec l'extérieur, tandis que la mémoire est chargée de stocker les informations.

Il existe différents types de mémoire ;

RAM Random access memory lecture et écriture, le contenu est perdu en cas de coupure d'alimentation

ROM Read only memory , uniquement lecture possible

EEPROM écriture/effacement électrique , le contenu peut être modifié en fonctionnement , garde l'information en cas de coupure d'alimentation

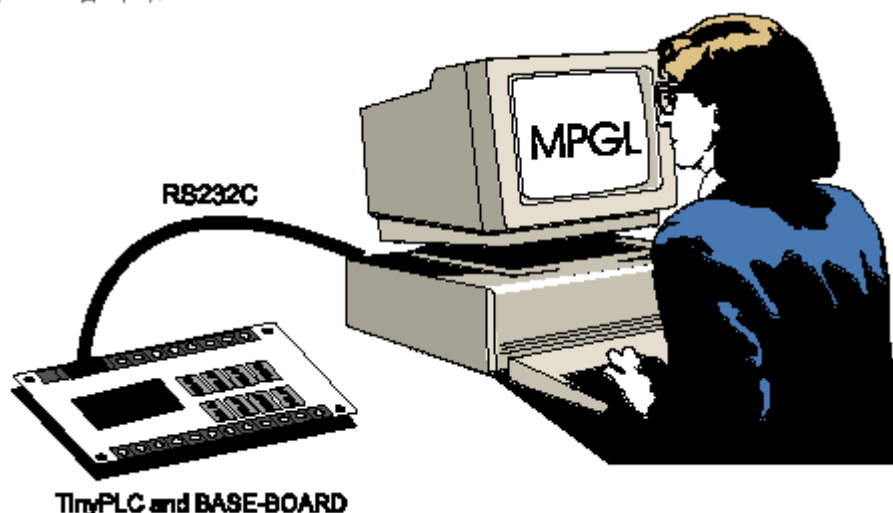
FLASH ROM fonction similaire à EEPROM , mais fabriquée de façon différente , le contenu du programme est stocké en FLASH sur le TinyPLC . Les valeurs sauvegardées sont stockées en EEPROM .

Types de relais

Les relais dans le contexte de PLC, peuvent aussi bien être des relais externes connectés aux entrées et sorties, que des relais internes et aussi les relais uniquement en mémoire, le tableau suivant décrit les relais de TinyPLC ; ils sont classés par les lettres suivantes P,M,K,T et C. Tous les relais exceptés les relais K sont à 0 à la mise sous tension ou après un reset.

Initiale	Nom	Fonction
P	Relais Entrées / Sorties (I/O relay)	Ce type de relais est connecté directement aux entrées et sorties .Les entrées peuvent être à l'état Haut ou Bas (1 ou 0, ON ou OFF et sont connectées sur les ports utilisés en entrée . Les sorties peuvent commander des électrovannes , moteurs , etc . Au total TinyPLC comporte 128 relais dans le domaine P ,
M	Relais Mémoire (support relay)	Ce type de relais n'existe uniquement dans la mémoire , il ne peut avoir directement accès sur l'extérieur . Son rôle est de garder l'information (M est le seul domaine qui peut être utilisé avec DF et DFN , front montant et descendant)
K	Relais sauvegardés (Keep relay)	Leurs rôle est similaire , ils gardent leurs valeurs même sans alimentation électrique . Ils sont utilisés dans le cas où l'on veut garder l'information hors tension
F	Relais spéciaux (Special relay)	Ce type de relay est une fonction interne du PLC , opérateurs internes ou informations temporisées . Par exemple répéter ON/OFF toutes les secondes
T	Temporisateurs (Timers)	Ce type de relais contrôle les temporisations , les tempos peuvent être ON ou OFF . La base de temps est de 10 ou 100 ms , voir l'instruction pour plus de détail
C	Compteurs (Counter)	Ce type de relais compte un nombre d'impulsions . Les impulsions plus courtes que le temps de cycle sont comptées par le compteur rapide CNT
D	Données (Data)	Le type de relais stocke les données d'opération sous forme de mots de 16 bits , plusieurs instructions de TinyPLC stockent l'information dans ce domaine
S	Relais pas à pas (Step controller relay)	Relais est utilisé pour le fonctionnement séquentiel . Comme les PLC fonctionnement sur le principe du temps de cycle , permet de résoudre simplement le problème de fonctionnement en séquentiel (ce que les autres langages permettent), qui pose des problème parfois.
CH	Affichage LCD	Buffer pour l'affichage LCD , si l'on écrit un code ASCII dans ce domaine , qui sera ensuite affiché sur le LCD, si l'on écrit un blanc (20H) le caractère est effacé
G	Affichage SGN	Buffer pour l'affichage 7 segments , quand l'on écrit un code ASCII , il apparaît ensuite sur l'afficheur
AD	Stockage du résultat d'une conversion A/D	Le résultat d'une conversion A/D est enregistrée dans ce domaine , l'utilisateur peut utiliser simplement les résultats en se référant à ce domaine
CNT	Compteur rapide	La valeur du compteur rapide est sauvegardée dans ce domaine. L'utilisateur peut utiliser cette fonction en se référant à ce domaine

Chargement et exécution du programme LADDER



Quand vous utilisez un PLC , vous devez avoir un programme LADDER , un câble de téléchargement et un PLC .

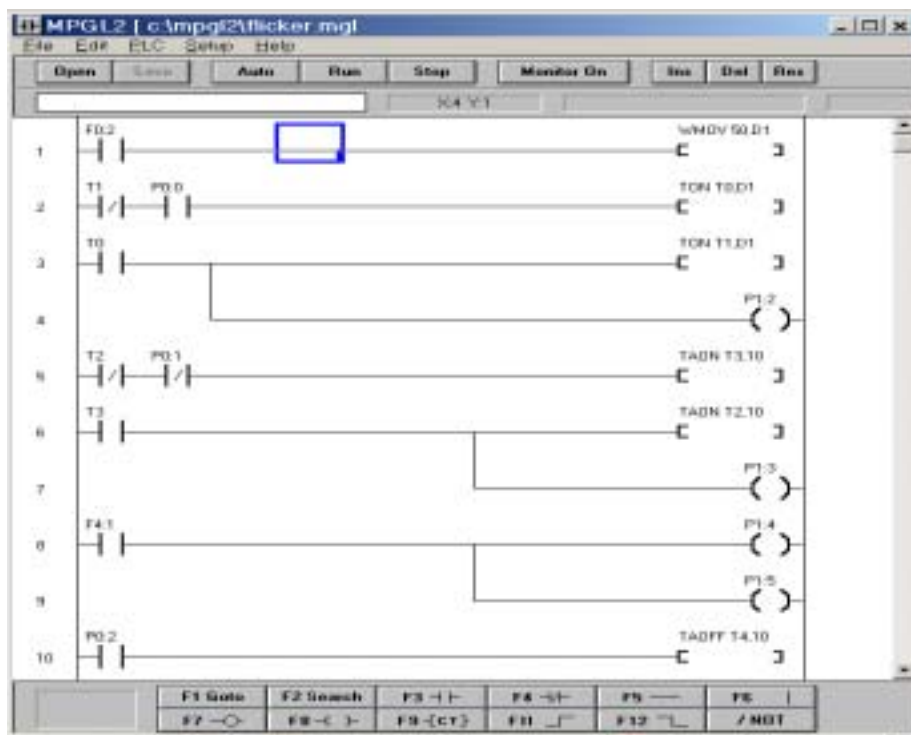
Pour le programme Ladder , il y a la possibilité d'utiliser beaucoup de type de micro-ordinateur ou une console dédiée. Mais le PC est très utilisé maintenant . Le Ladder est décrit plus simplement avec les graphismes du PC . Les diagrammes Ladder dessinés sur le PC peuvent être téléchargés en utilisant un câble RS232C . Après téléchargement le programme attend une coupure d'alimentation , puis remise sous tension .

Le développement de programmes utilisant TinyPLC s'effectue en répétant trois phases

Ecriture (dessin) du programme

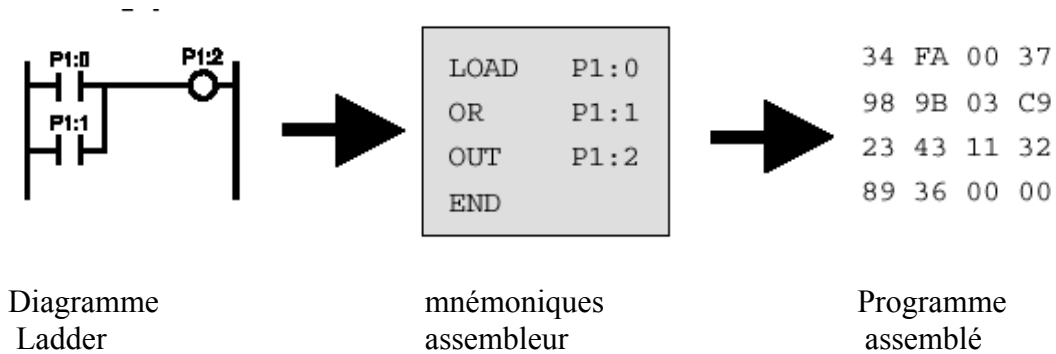
Compilation téléchargement

Vérification du fonctionnement

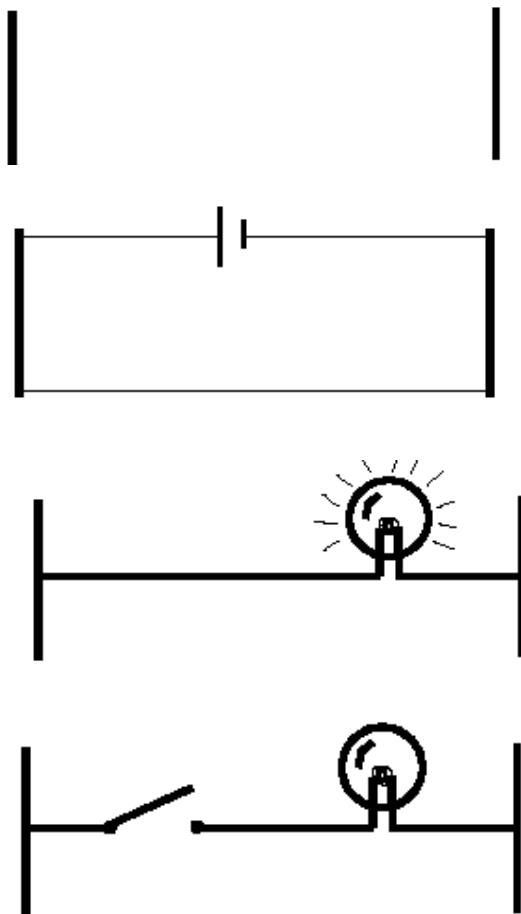


Copie d'écran de MGPL2 sous Windows

Dans TinyPLC , le Ladder procède en trois étapes pour effectuer le téléchargement , en premier il compile le dessin Ladder en mnémoniques et ensuite l'assembleur les transforme en code puis le téléchargement est effectué, par l'intermédiaire de câble RS232 . Si une erreur de dessin est présente, le processus est arrêté pour y remédier.



Symboles de base LADDER

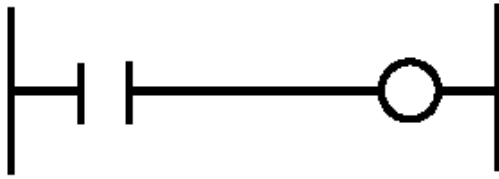


Un symbole Ladder est très simple, il consiste en deux lignes au départ

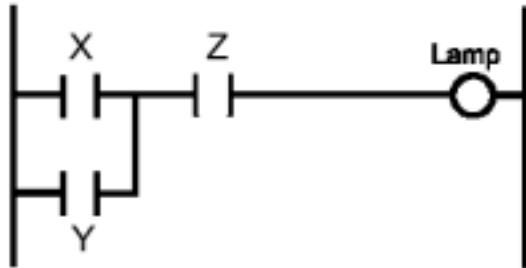
Ces deux lignes correspondent à l'alimentation

Si une lampe est connectée sur ces lignes, elle s'allume

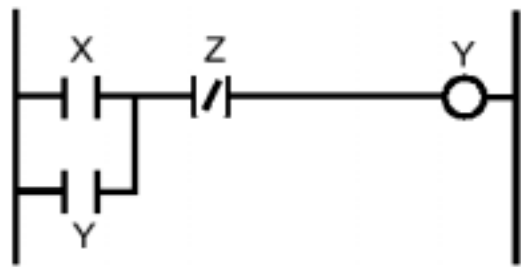
On peut ajouter un interrupteur, si l'interrupteur est fermé la lampe s'allume , si il est ouvert la lampe est éteinte



Quand le circuit est dessiné par symboles, c'est un diagramme de symboles à relais



Quand plusieurs symboles sont mis comme ci-contre, il s'agit de conditions, X et Y font un OU et Z forme un ET avec X ou Y. Pour avoir la lampe allumée il faut Z et X ou Z et Y.



Dans ce circuit si X est appuyé, Y se colle car la condition AND avec Z est à ON, ensuite Y (OU avec X) maintient le circuit sur ON. Si l'on appuie sur Z la sortie passe à OFF

Notes


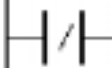


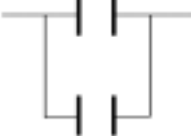
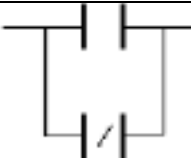
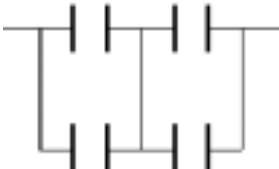
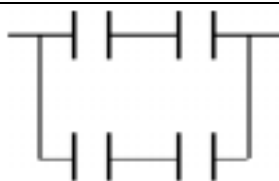
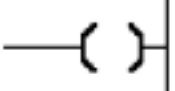
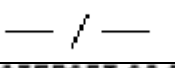
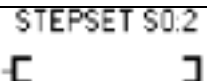
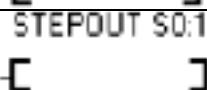
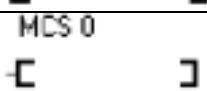
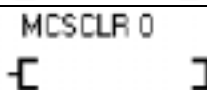
Chapitre 3

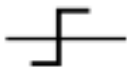








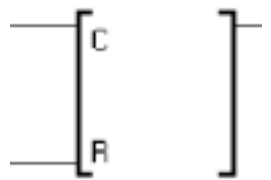
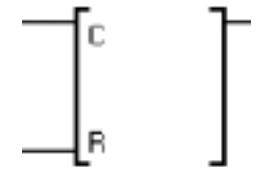
Instructions de base

De

TinyPLC

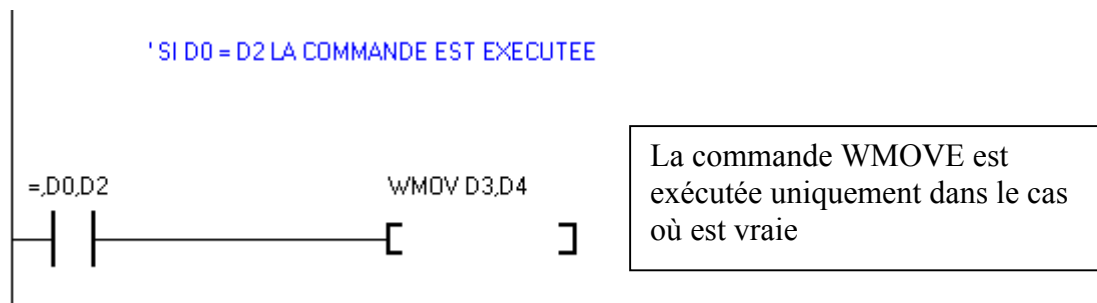
Tableau des instructions**Instructions de base**

Nom	Ins- truction	Symbole	Description
Load	LOAD		Fonction qui commence au point de contact A, NO
Load Not	LOADN		Fonction qui commence au point de contact B , NF
And	AND		Interface série de contacts
And Not	ANDN		Interface série entre NO et NF
Or	OR		Interface parallèle
Orn	ORN		Interface parallèle entre contacts NO et NF
And Stack	ANDS		Interface AND entre blocs
Or Stack	ORS		Interface OR entre blocs
Output	OUT		Sortie , résultat d'une fonction
Not	NOT		Inverse le résultat de la fonction qui la précède
Step sequential set	STEPSET		Sortie séquentielle d'un contrôle pas à pas
Step Output	STEPOUT		Sortie d'un pas (LIFO)
Master control set	MCS		Commence le contrôle de boucles
Master control clear	MCSCLR		Termine le contrôle de boucles

Differential	DF		Front montant (différentiel)
Differential Not	DFN		Front descendant (différentiel)
Set Output	SETOUT	SETOUT P0:0 	Met une sortie à 1
Reset Output	RSTOUT	RSTOUT P0:1 	Met une sortie à 0
Save Status	SAVES	Pas de symbole	Sauve l'état de la fonction en cours
Read Status	RDS	Pas de symbole	List l'état sauvegardé
End	END	END 	Fin du programme
Tempos et compteurs			
On Timer Base 10 ms	TON	TON T1,100 	Relais tempo valeur maxi 32767 soit 327.67 s . Quand elle est validée , la tempo démarre , la sortie passe à ON en fin de tempo, sinon reste à OFF
Off Timer Base 10 ms	TOFF	TOFF T1,100 	Relais tempo valeur maxi 32767 Passe à 1 dès validation, se maintient à 1 après dé validation pendant le temps programmé
On Timer Base 100 ms	TAON	TAON T1,100 	Base 0.1 s max 3276.7 secondes Fonct. Idem TON
Off Tmer Base 100 ms	TAOFF	TAOFF T1,100 	Idem TOFF mais base de 100 ms Max 3276.7 secondes
Up Counter	CTU	CTU C1,100 	Compteur (maxi 65535). Augmente de 1 a chaque lois que l'entrée passe à 1 . remis à 0 par le reset. La sortie passe à ON quand la valeur programmée est atteinte
Down counter	CTD	CTD C1,100 	Dé compteur , utilisé pour décompter depuis la valeur programmée (65535 mxi) . A chaque fois que l'entrée passe à 1 , la valeur diminue de 1 . La valeur est réinitialisée à la valeur programmée par le reset

Commandes de comparaisons

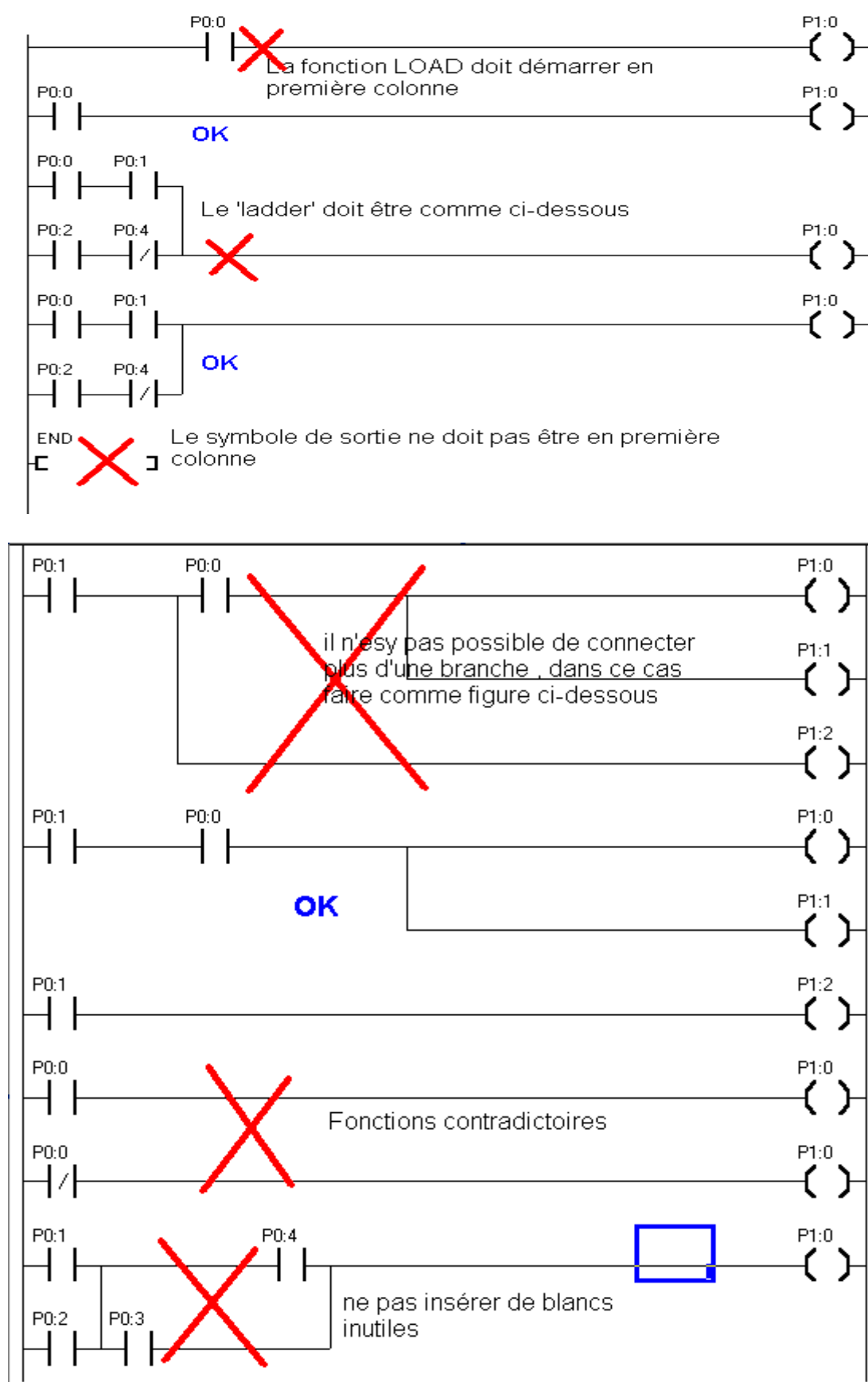
Les commandes de comparaisons implémentées et utilisables de TinyPLC sont décrites dans le tableau suivant et commencent en général par le signe d'inégalité.



Classification	Format de l'instruction	Description
Commandes de comparaisons Mots 16 bits	=,S1,S2	Dans le cas où S1 = S2 , le point de connexion passe à 1 . Il compare une valeur de 16 bits
	>,S1,S2	Si S1 > S2 le point de connexion passe à 1
	<,S1,S2	Si S1 < S2 le point de connexion passe à 1
	>=,S1,S2	Si S1 >= S2 le point de connexion passe à 1
	<=,S1,S2	Si S1 <= S2 le point de connexion passe à 1
	<>,S1,S2	Si S1 >= S2 le point de connexion passe à 1
Commandes de comparaisons Mots long 32 bits	D=,S1,S2	Dans le cas où S1 = S2 , le point de connexion passe à 1 . Il compare une valeur de 32 bits
	D>,S1,S2	Si S1 > S2 le point de connexion passe à 1
	D<,S1,S2	Si S1 < S2 le point de connexion passe à 1
	D>=,S1,S2	Si S1 >= S2 le point de connexion passe à 1
	D<=,S1,S2	Si S1 <= S2 le point de connexion passe à 1
	D<>,S1,S2	Si S1 >= S2 le point de connexion passe à 1

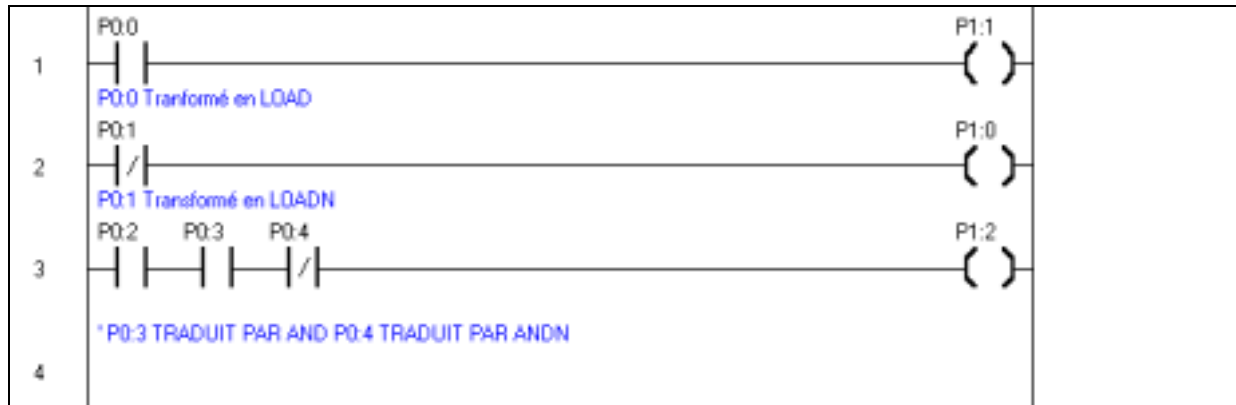
Les interfaces AND et OR peuvent être utilisées avec les commandes de comparaisons, de façon identiques en général aux points de connexions.

Précautions pour les entrées LADDER



LOAD, LOADN

AND, ANDN



Commandes	P	M	Relais			Comp.	Timer	Etc ..			Constantes
			F	K	S	C	T	AD	CH	G	
LOAD	■	■	■	■	■	■	■				
LOADN	■	■	■	■	■	■	■				
AND	■	■	■	■	■	■	■				
ANDN	■	■	■	■	■	■	■				

Description en détail

LOAD charge le point de connexion P0:0 , si l'entrée passe à 1 la sortie passe à 1

LOADN charge le point de connexion P0 :1 , si l'entrée passe à 1 , la sortie passe à 0

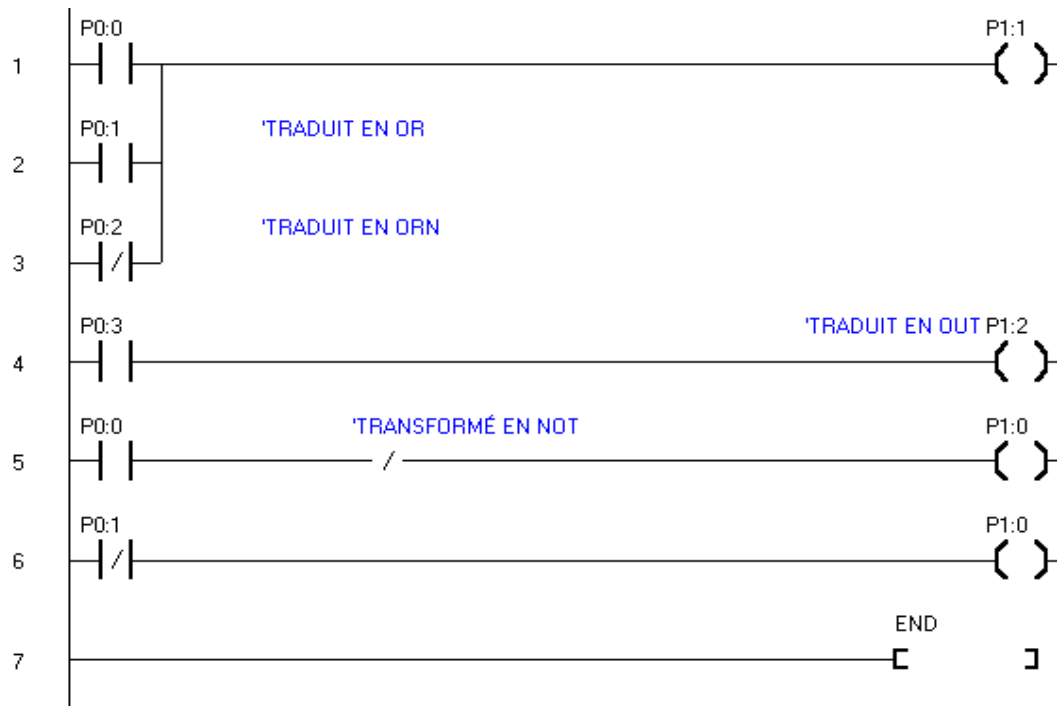
AND et ANDN sont des commandes qui permettent d'utiliser les entrées mises en séries

Si P0 :2 et P0 :3 passent à 1 , la sortie P1 :1 passe à 1 , si le point de connexion P0 :4 passe à 1 la sortie P1 :1 passe à 0

Très important : Les entrées et sorties doivent être configurées dans la programme MPGL2 .Si les entrées et sorties ne sont pas configurées correctement, le montage ne fonctionnera pas correctement.

OR,ORN,OUT

NOT, END



Commandes	Relais					Comp.	Timer	Etc ..			Constantes
	P	M	F	K	S	C	T	AD	CH	G	
OR	■	■	■	■	■	■	■				
ORN	■	■	■	■	■	■	■				
OUT	■	■		■							

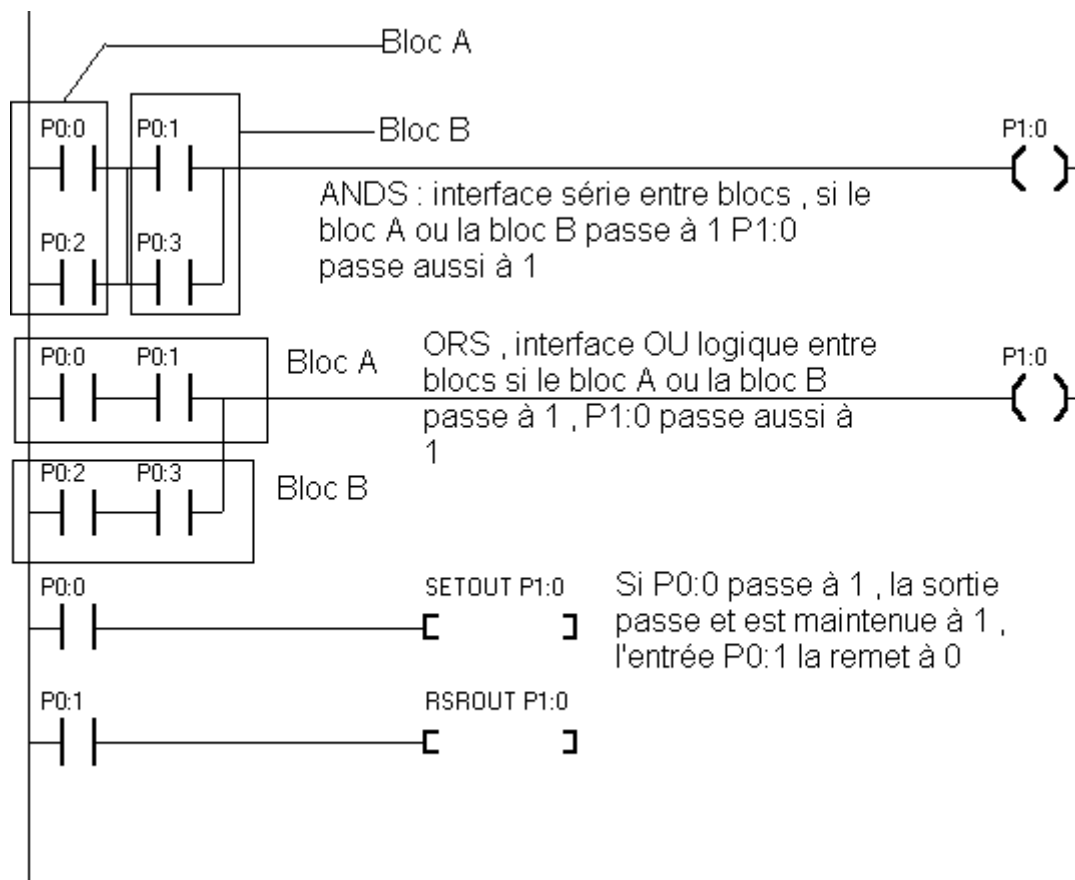
Lignes 1,2 et 3 ,fonction OR et ORN il suffit qu'une seule des conditions soit vraie pour avoir la sortie à 1 , si P0 :0 = 1 ou P0 :1 = 1 P1 :1 passe à 1 , si P0 :2 passe à 1 P1 :1 passe à 0 (sous réserve que P0 :0 et P0 :1 soient à 0)

Ligne 4 si P0 :3 passe à 1 P1 :2 passe à 1

Lignes 5 et 6 si P0 :0 passe à 1 la sortie P1 :0 passe à 0 , les lignes 5 et 6 ont des effets équivalents

Ligne 7 END , affiche la fin du programme , à utiliser avec précaution toute instruction placée après END est ignorée par MGPL2.

ANDS, ORS, SETOUT, RSTOUT

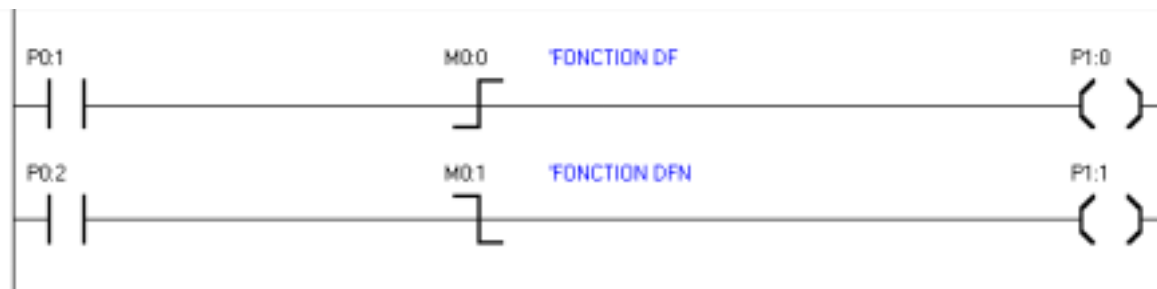


Commandes	Relais					Comp.	Timer	Etc ..			Constantes
	P	M	F	K	S			AD	CH	G	
SETOUT	■	■		■							
RSTOUT	■	■		■							

DF , DFN

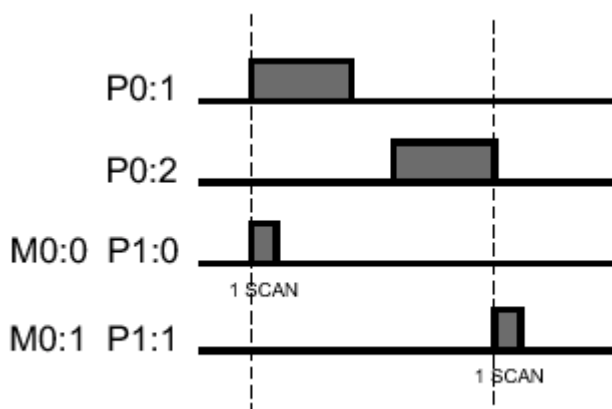
DF , si le signal d'entrée passe du niveau 0 au niveau 1 (front montant) la sortie de la condition passe au niveau 1 pour un cycle 2.5 ou 5 ms

DFN , si le signal d'entrée passe du niveau 1 au niveau 0 (front descendant) la sortie de la condition passe au niveau 1 pour cycle .



Commandes	Relais					Comp. C	Timer T	Etc ..		Constantes
	P	M	F	K	S			AD	CH	
DF		■								
DFN		■								

Attention : DF et DFN ne sont utilisables uniquement que dans le domaine M

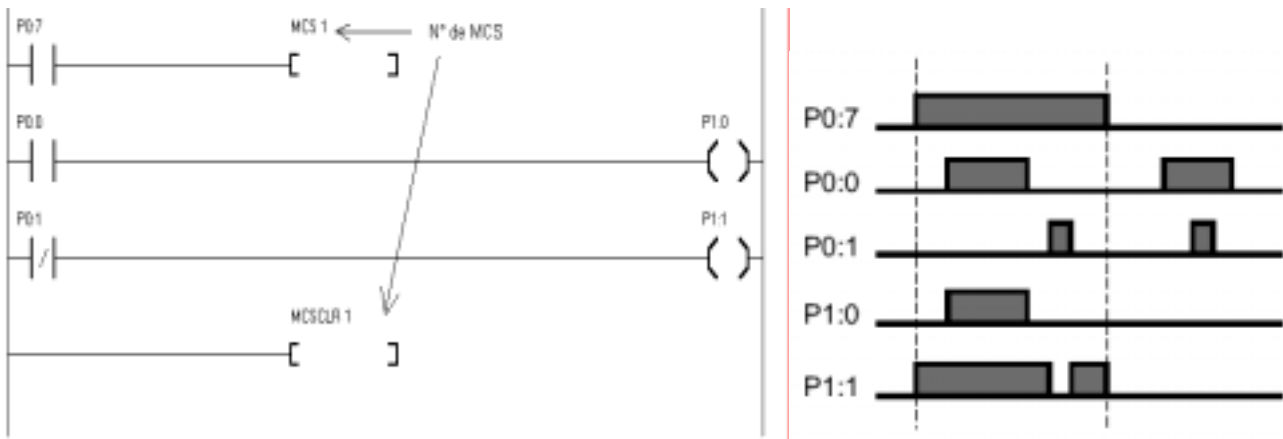


Au moment que P0 :1 passe au niveau 1 , P1 :0 passe au niveau 1 pendant 1 cycle du programme (2.5 ou 5 ms)

Au moment que P0 :2 passe au niveau 0 , P1 :1 passe au niveau 1 pendant un 1 cycle

MCS, MCSCLR

Abréviation de Master Control Relay. Quand la condition initiale de MCS est vraie ,exécution des fonctions jusqu'à MCSCLR du même numéro. Si MCSCLR est à 0, les instructions ne sont pas exécutées et les sorties entre MCS et MCSCLR restent à 0 .

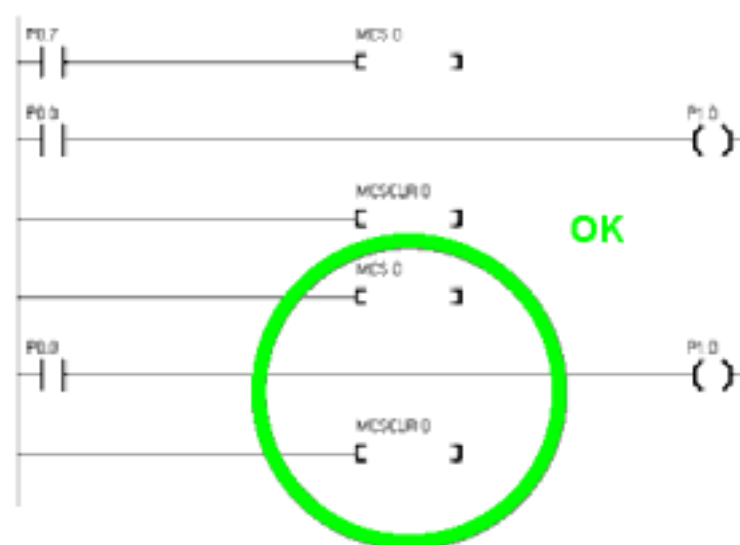
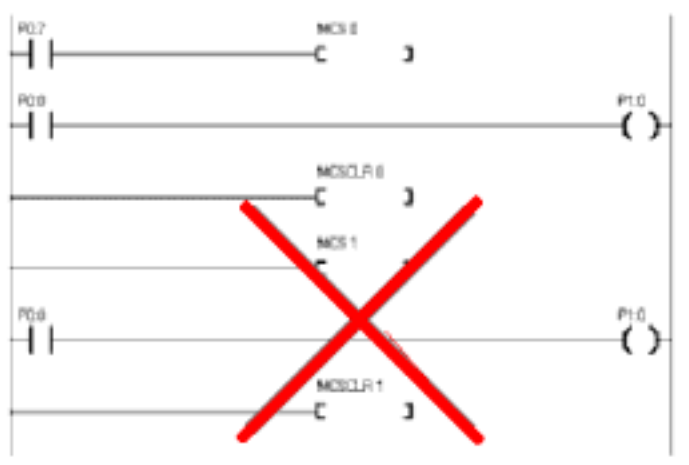
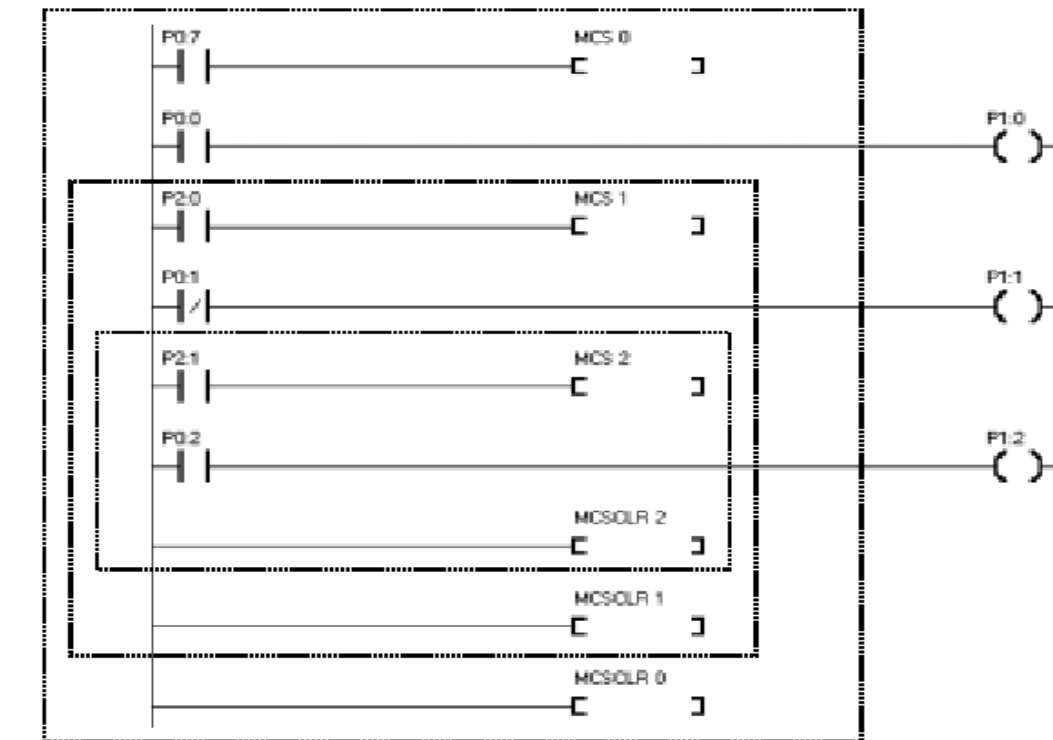


Si P0 :7 passe à 1 , les instructions entre MCS 1 et MCSCLR 1 seront exécutées , si MCS1 reste à 0, les instructions entre MCS 1 et MCSCLR 1 resteront à 0 .

Les valeurs utilisables avec MCS et MCSCLR vont de 0 (la plus forte priorité) à 7 (la plus faible priorité). Si l'on fini une boucle MCS MCSCLR , toutes les boucles de plus faible priorité sont aussi libérées ., voir aussi le tableau suivant :

Instruction	Condition MCS à ON (1)	Condition MCS à OFF (0)
OUT	Exécution normale	OFF sans conditions
SETOUT	Exécution normale	Maintenu constamment dans l'état avant que MCS soit à ON
RSTOUT	Exécution normale	Maintenu constamment dans l'état avant que MCS soit à ON
TON TOFF	Exécution normale	Remis à la valeur initiale
CTU CTD	Exécution normale	Maintenu constamment dans l'état avant que MCS soit à ON
Autres instructions.	Exécution normale	Non exécuté

Voir exemple page suivante pour l'imbrication de boucles MCS , MCSCLR



STEPSET

Contrôle pas à pas Contrôle séquentiel

Dans le cas ou beaucoup de valeurs précédentes sont à ON , le passage de la séquence suivante à ON va provoquer , la passage à OFF des séquences précédentes, ceci est appelé contrôle séquentiel parce qu'il passe à ON de façon séquentielle. Les pas de 0 à 255 sont utilisables .



Commandes	P	M	Relais F K S	Comp. C	Timer T	Etc .. AD CH G	Constantes
STEPSET				■			

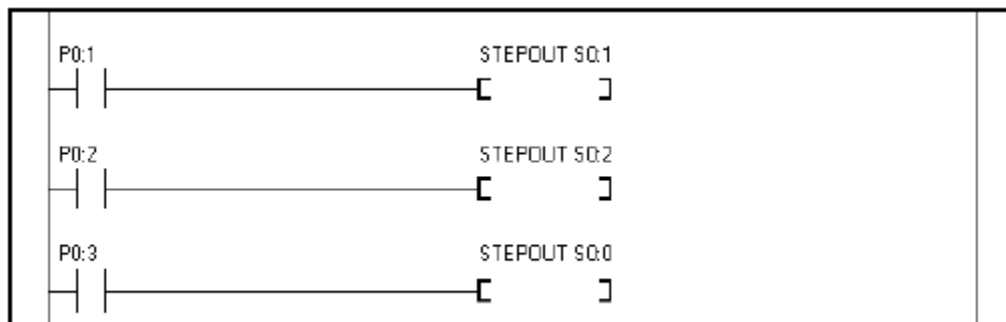
Time Chart



Si P0:2 passe à ON , le pas 2 du groupe 0 (même groupe) attend et fait passer le pas 1 à OFF pour passer à ON .Si P0 :3 passe à 1 , il fait passer sans condition au pas 0 (le pas 0 est utilisé pour faire un reset)

STEPOUT

Si plusieurs entrées sont dans le même groupe , uniquement le dernier pas passe à ON et les autres pas passent à OFF . Le dernier pas à la priorité sur tous les autres pas , ceci est appelé LIFO (last in , first out , dernier entré , premier sorti) , le nombre de pas utilisables est de 255 .



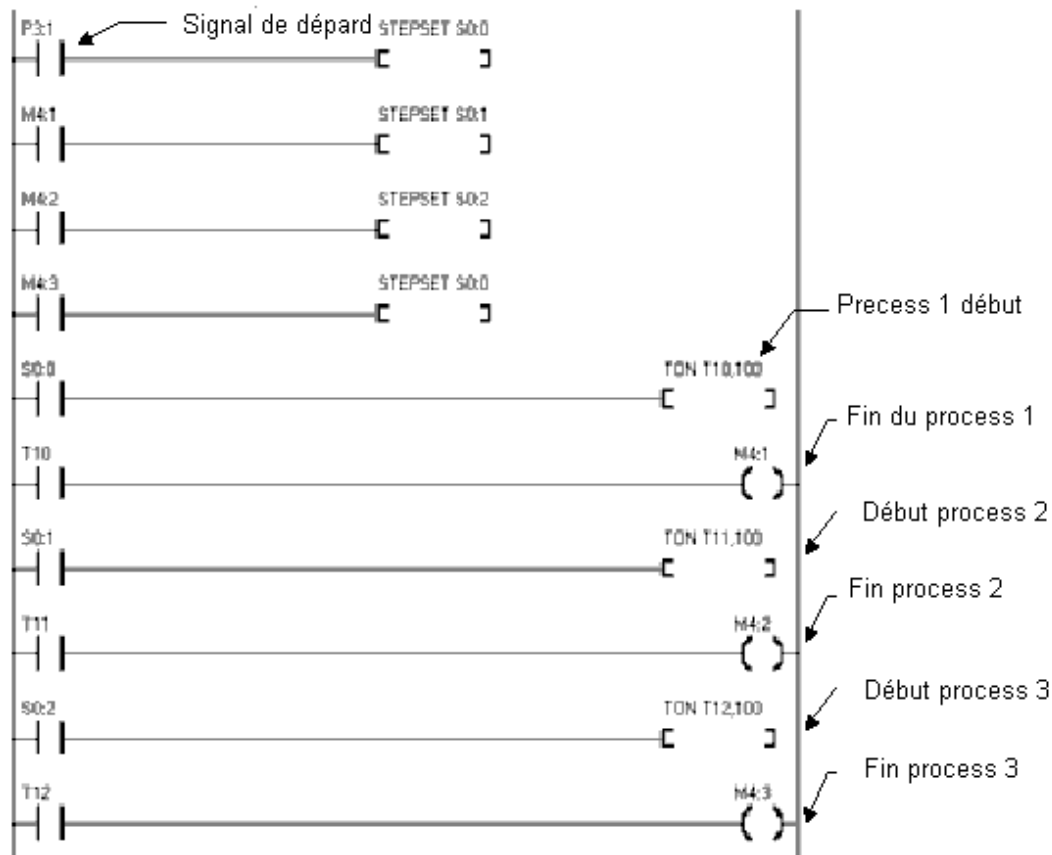
Commandes	Relais					Comp.	Timer	Etc ..			Constantes
	P	M	F	K	S	C	T	AD	CH	G	
STEPOUT					■						

Si P0 :1 passe à ON , le premier pas passe à ON , après si P0 :3 passe à ON , la pas 0 passe à ON , après si P0 :2 passe à ON , le pas 2 passe à ON .
 Sans condition , le dernier pas passe à ON et les autres pas passent à OFF

Description supplémentaire

Le langage C ou l'assembleur font du contrôle séquentiel facilement mais il n'est pas implémenté dans TinyPLC pour des raisons de temps de cycle . Ce mode est implémenté pour cette raison , il est utile exécuter un process en séquences .

Voir exemple page suivante :



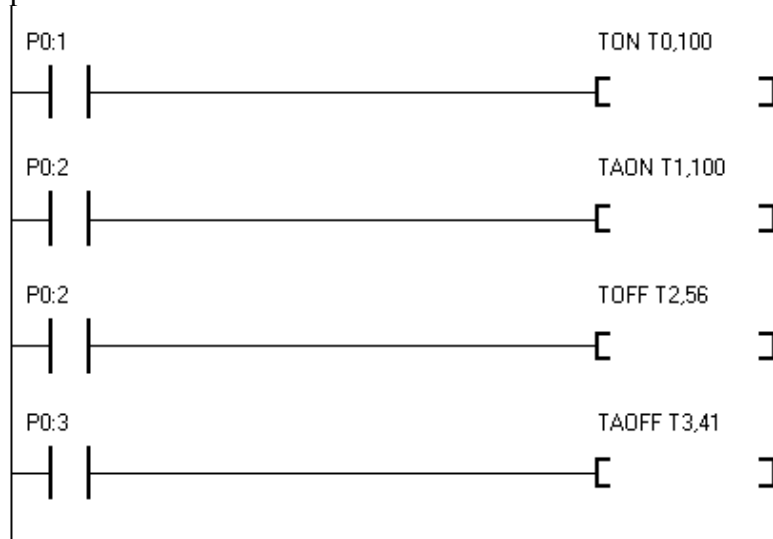
Les relas STEP ont une fonction automaintien (avant une autre entrée , il sont maintenu dans l'état)
 Dans un même groupe , uniquement 1 seul peut être à ON
 En cas de contrôle séquentiel , il est possible de revenir un pas en arrière , dans le cas où la valeur précédente est à ON

TON , TAON ,TOFF,TAOFF

TON et TAON , si la condition d'entrée passe à ON , la temporisation commence , si la condition passe à OFF , la tempo revient à zéro (reset) . Si la tempo atteint sa valeur de réglage , la sortie passe à ON .

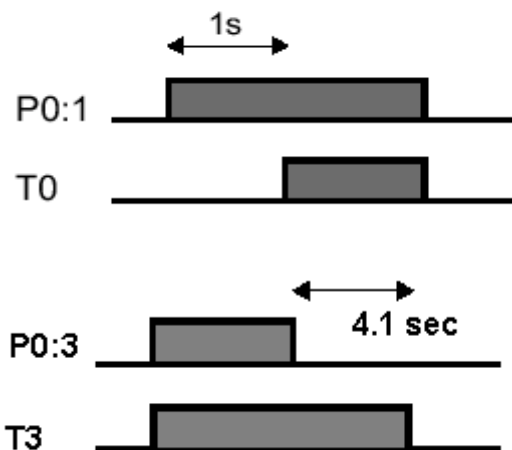
TOFF et TAOFF si la condition de la tempo passe à ON , la sortie passe aussi à ON , quand l'entrée passe à OFF , la sortie ne passera à OFF que lorsque le temps réglé par la tempo sera écoulé.

Les tempos TON et TOFF ont une base de temps de 0.01 seconde et donc un maximum de 327.68 secondes , tandis que TAON et TAOFF ont une base de temps de 0.1 seconde soit un temps maxi de 3276.8 secondes



Opérandes

Commandes	Relais					Comp.	Timer		Etc ..			Constantes
	P	M	F	K	S	C	T	D	AD	CH	G	
T (tempo)							■					
Point de connexion								■				
N (valeur)								■				■



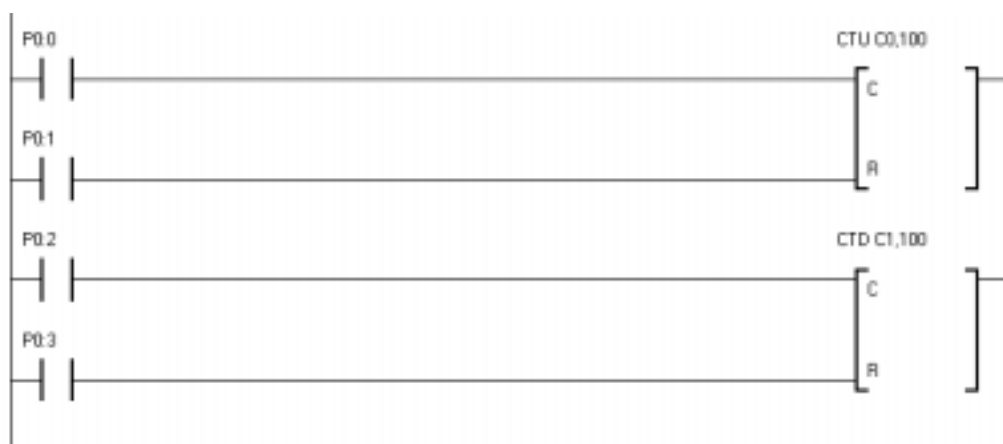
TON , 1 seconde après la validation de P0 :1 à ON , la sortie T0 passe à ON jusqu'à la condition de reset de l'entrée

TAOFF , T3 démarre en même temps que P0 :3 et passe à OFF 4.1 secondes après le passage à OFF de l'entrée

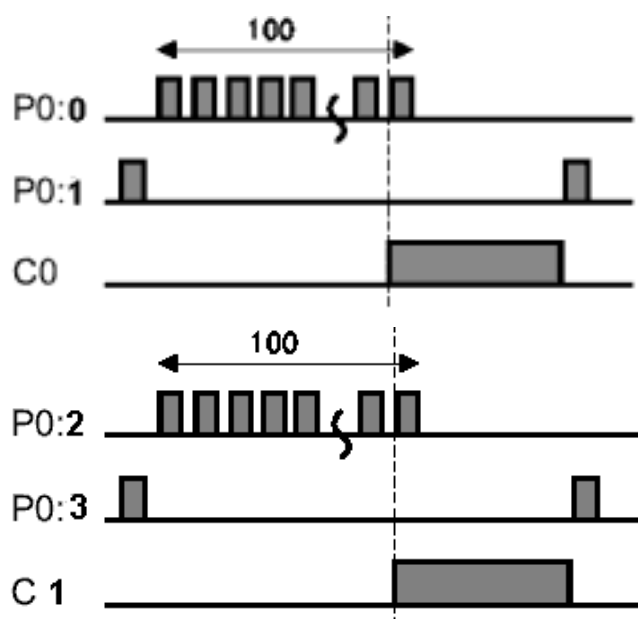
CTU , CTD

CTU , chaque fois que l'entrée du compteur passe à ON, la valeur augmente de 1 , si la valeur atteint la valeur de réglage , le contact de sortie passe à ON . Si des entrées sont toujours présentes à l'entrée ,il continue à compter jusqu'à 65535 , ensuite il repasse à zéro mais en maintenant la sortie . Si l'entrée reset est mise à ON , le compteur revient à zéro .

CTD , chaque fois qu'une entrée est présente sur le compteur , la valeur est décrémentée de 1 , lorsque la valeur atteint 0 la sortie passe à ON . Une entrée reset remet le compteur à sa valeur de présélection .



Commandes	Relais					Comp.	Timer		Etc ..			Constantes
	P	M	F	K	S	C	T	D	AD	CH	G	
T (tempo)							■					
Point de connexion												
N (présélection)								■				■

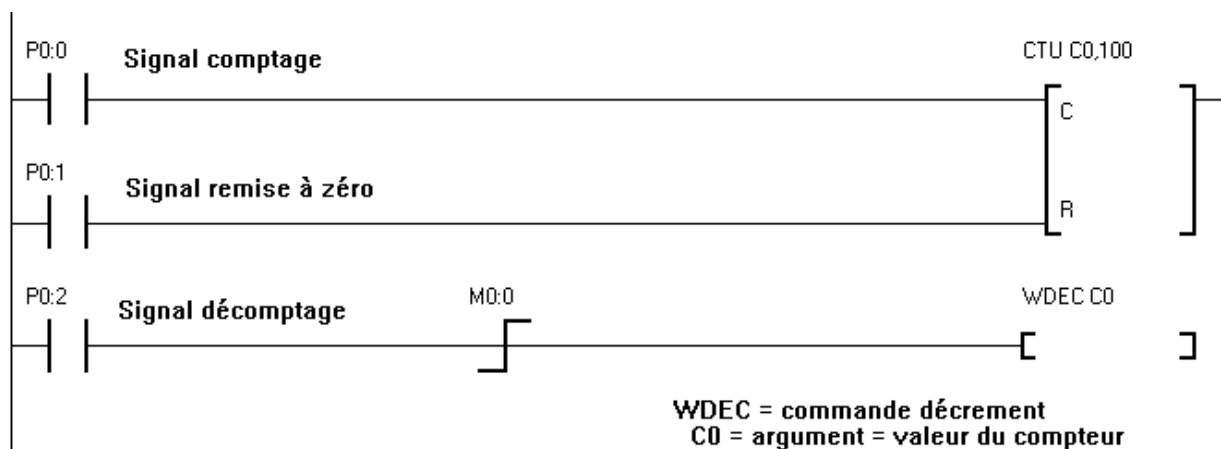


CTU : Après un reset quand le compteur a compté 100 unités , la sortie C0 passe et reste à ON , jusqu'à un reset provoqué par le passage à ON de P0 :1, ou elle revient à 0

CTD , la valeur du compteur décompte jusqu'à 0 , qui fait passer la sortie à ON . Un reset remet le compteur a sa valeur de présélection et remet la sortie C1 à OFF

Implémentation d'un compteur /dé compteur

TinyPLC ne possède pas de compteur/dé compteur spécial, mais le compteur dé compteur ci dessus est utilisable par implémentation .



La valeur de C0 est décrémentée de 1 par la passage de P0 :2 à ON , et le front montant (pour ne compter qu'une unité , sinon une unité par cycle)

Domaine K des temporisateurs et compteurs

En regardant l'espace mémoire l'on peut voir qu'une partie des tempos et compteurs sont dans l'espace sauvegardé en cas de coupure d'alimentation . Si vous utilisez cet espace pour les tempos et les compteurs , leur valeur va rester pendant la coupure d'alimentation et après remise sous tension le processus va continuer.

Attention : Si vous utilisez un compteur dans le domaine K (sauvegardé) , vous devez utiliser un compteur CTU , pour continuer à compter après remise sous tension , car la commande CTD est initialisée à sa valeur correspondante à la mise sous tension .

Commandes de comparaisons de mots

= , <> , < , > , >= , <=

Si le résultat de la condition de comparaison de mots (16 bits) est vrai, la connexion de sortie passe à ON. IL existe 6 commandes de comparaisons

Commandes de comparaisons	Exécution
= , s1,s2	Si s1 = s2 la sortie passe à ON
<>,s1,s2	Si s1 est différent de s2 , la sortie passe à ON
>,s1,s2	Si s1 est plus grand que s2 , la sortie devient ON
<,s1,s2	Si s1 est plus petit que s2 , la sortie devient ON
>=,s1,s2	Si s1 est égal ou plus grand que s2 , la sortie devient ON
<=,s1,s2	Si s1 est égal ou plus petit que s2 , la sortie devient ON

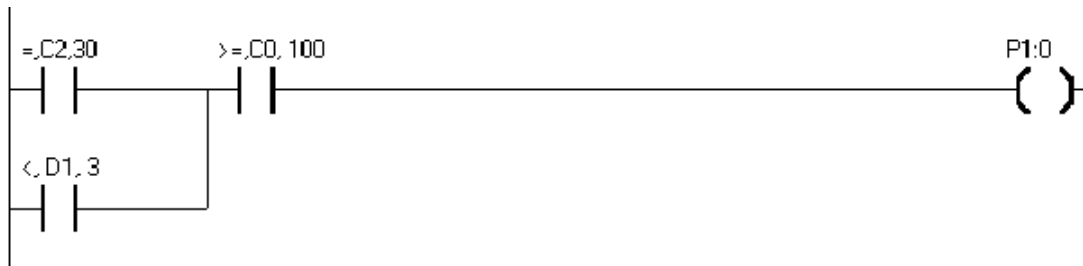


Si D0 est égal à D1 , P1 :0 passe à ON
 Si D0 est différent de D1 , P1 :1 passe à ON
 Si D0 est plus grand que D1 , P1 :2 passe à 1
 Si D0 est plus petit que D1 , P1 :3 passe à 1
 Si D0 est plus grand ou égal à D1 , P1 :4 passe à 1
 Si D0 est plus petit ou égal à D1 , P1 :5 passe à 1

Opérandes

Commandes	P	M	Relais			Comp.	Timer	AD	Etc ..	Data	Constantes
			F	K	S	C	T		CH		
S1						■	■	■	■	■	■
S2						■	■	■	■	■	■

Les fonctions AND et OR sont utilisables , exemple :



Si C2 = 30 et C0 >= 100 P1 :0 passe à ON

Aussi si D1 est plus petit que 3 et C0 plus grand ou égal à 100, P1 :0 passe à ON

Com. de comparaisons de mots doubles

D= , D<> , D< , D> , D>= , D<=

Si le résultat de la condition de comparaison de mots longs (32 bits) est vrai, la connexion de sortie passe à ON. IL existe 6 commandes de comparaisons

Commandes de comparaisons	Exécution
D= , s1,s2	Si s1 = s2 la sortie passe à ON
D<>,s1,s2	Si s1 est différent de s2 , la sortie passe à ON
D>,s1,s2	Si s1 est plus grand que s2 , la sortie devient ON
D<,s1,s2	Si s1 est plus petit que s2 , la sortie devient ON
D>=,s1,s2	Si s1 est égal ou plus grand que s2 , la sortie devient ON
D<=,s1,s2	Si s1 est égal ou plus petit que s2 , la sortie devient ON



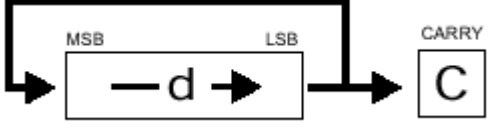
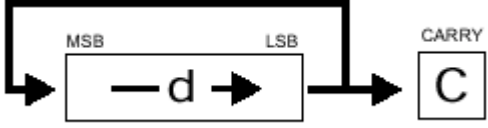
Le fonctionnement et l'exécution des commandes de comparaison de mots longs est identique que pour les mots de 16 bits , voir les explications pages précédentes .



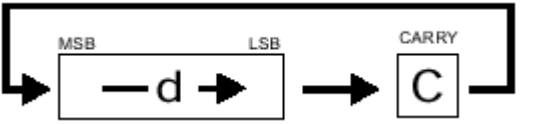
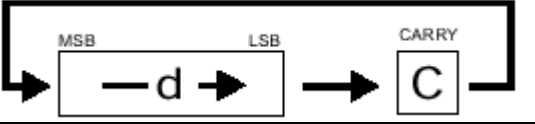
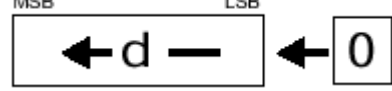
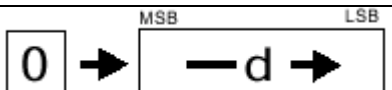
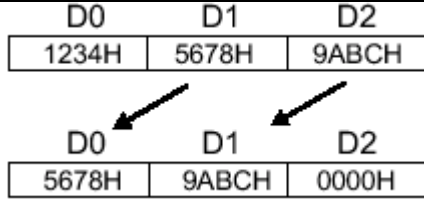
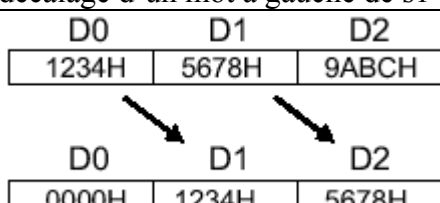
Chapitre 4

Commandes d'applications TinyPLC

Commandes d'applications

Class e	Fonction	Format de la commande	Description
Commandes de déplacement	Byte move	MOVE s,d	Déplace une donnée (8 bits) (s) → (d)
	Byte complement move	CMOVE s,d	Déplace le complément d'une donnée (8 bits) (s) → (d)
	Word move	WMOV s,d	Déplacement d'une donnée de 16 bits (s) → (d)
	Double word move	DWMOV s,d	Déplacement d'une donnée de 32 bits (s,s+1) → (d, d+1) Pour 32 bits le mot le plus significatif est sauvegardé dans s et le – significatif dans s+1
	Word complement move	WCMOV s,d	Déplace le complément (16 bits) de la donnée Complément(s) → (d)
	Double word complement move	DWCMOV s,d	Déplace le complément d'une donnée 32 bits Complément (s,s+1) → (d ,d+1)
	Word negative move	WNEG s,d	Déplace le négatif de la donnée (16 bits) Negative 2(s) → (d)
	Double word negative move	DWNEG s,d	Déplace le négatif de la donnée (32 bits) Negative 2(s,s+1) → (d,d+1)
	Word exchange move	WXCHG s,d	Echange des données 16 bits (s) ↔ (d)
	Double word exchange move	DWXCHG s,d	Echange de données 32 bits (s,s+1) ↔ (d,d+1)
	Fill move	FMOV s,d,n	Commande de remplissage de données (s) → Nombre de n données à partir de (d)
	Group Move	GMOV s,d,n	Déplacement d'un groupe de données Nombre de n données depuis (s) → Nombre de n données depuis (d)
Commandes de conversion	Word binary to BCD code	WBCD s,d	Convertit une donnée binaire 16 bits en code BCD 4 digits (16 bits binary) → (4 digit BCD)
	Double word binary To BCD code	DWBCD s,d	Convertit une donnée 32 bits en 8 digits BCD (32 bits binary) → (8 digit BCD)
	Word BCD code to binary	WBIN s,d	Convertit 4 digit BCD en une donnée binaire 16 bits (4 digits BCD) → (16 bits binaires)
	Double word BCD code to binary	DWBIN s,d	Convertit 8 digits BCD en une donnée binaire 32 bits (8 digits BCD) → (32 bits binaire)
Incréments décréments	Word Increment	WINC d	Incrémene une valeur 16 bits de 1 (d)+1 → (d)
	Double word increment	DWINC d	Incrémene une valeur de 32 bits de 1 (d,d+1)+1 → (d,d+1)
	Word decrement	WDEC d	Décrémene de 1 un mot de 16 bits (d)-1 → (d)
	Double word decrement	DWDEC d	Décrémene une donnée de 32 bits de 1 (d,d+1)-1 → (d,d+1)

Class e	Fonction	Format de la commande	Description
Calculs arithmétiques	Word addition	WADD s1,s2,d	Addition de mots 16 bits $(s1)+(s2 \rightarrow d)$
	Double word addition	DWADD s1,s2,d	Addition de mots longs 32 bits $(s1,s1+1) + (s2,s2+1) \rightarrow (d,d+1+1)$
	Word soustraction	WSUB s1,s2,d	Soustraction de mots 16 bits $(s1)-(s2 \rightarrow d)$
	Double Word soustraction	DWSUB s1,s2,d	Soustraction de mots longs 32 bits $(s1,s1+1) - (s2,s2+1) \rightarrow (d,d+1+1)$
	Word multiplication	WMUL s1,s2,d	Multiplications de mots 16 bits , résultat 32 bits $(s1)*(s2) \rightarrow (d,d+1)$
	Double Word multiplication	DWMUL s1,s2,d	Multiplications de mots longs (32 bits) , résultat 64 bits $(s1,s1+1) *(s2,s2+1) \rightarrow (d,d+1,d+2,d+3)$
	Word division	WDIV s1,s2,d	Division de mots 16 bits $(s1)/(s2) \rightarrow (d)$ pour une partie , (d+1) pour le reste
	Double Word division	DWDIV s1,s2,d	Division de mots longs (32 bits) $(s1,s1+1) / (s2,s2+1) \rightarrow (d,d+1)$ pour une partie (d+2,d+3) pour le reste
Opérations logiques	Word and	WAND s1,s2,d	AND sur mots 16 bits $(s1) \text{ AND } (s2) \rightarrow (d)$
	Double word and	DWAND s1,s2,d	AND sur mots longs de 32 bits $(s,s+1) \text{ AND } (s2,s2+1) \rightarrow (d,d+1)$
	Word Or	WOR s1,s2,d	OR sur mots de 16 bits $(s1) \text{ OR } (s2) \rightarrow (d)$
	Double Word OR	DWOR S1,s2,d	OR sur mots longs de 32 bits $(s1,s1+1) \text{ OR } (s2,s2+1) \rightarrow (d,d+1)$
	Word Xor	WXOR s1,s2,d	XOR sur mots de 16 bits $(s1) \text{ XOR } (s2) \rightarrow (d)$
	Double word XOR	DWXOR S1,s2,d	XOR sur mots longs de 32 bits $(s1,s1+1) \text{ XOR } (s2,s2+1) \rightarrow (d,d+1)$
Calculs	Word Rotate left	WROL d	
	DoubleWord Rotate left	DWROL d	
	Word rotate right	WROR d	
	Double rotate right	DWROR d	

Commandes de rotation	Word rotate left with carry	WRCL d	
	Double Word rotate left with carry	DWRCL d	
	Word rotate right with carry	WRCR d	
	Double Word rotate right with carry	DWRCR d	
Commande de décalage	Bit shift left	BSHL d,n	 <p>Commande de décalage 'd' à gauche de 'n' bits</p>
	Bit shift right	BSHR d,n	 <p>Commande de décalage 'd' à droite de 'n' bits</p>
	Word shift left	WSHL d,n	 <p>Commande de décalage d'un mot à gauche de s1 vers s2</p>
	Word shift right	WSHR d,n	 <p>Commande de décalage d'un mot à droite de s1 vers s2</p>
Com	Convert for 7 segments	SEG s,d	Commande de conversion en 7 segments, converti un mot 16 bits de s en une valeur 7 segments et 4 digits
	LCD area clear	LCDCLS	Insère des espaces (code ASCII 20H) dans l'espace CH (efface le LCD)
	LCD data output	LCDOUT port,mode	Affiche sur le LCD les données de l'espace CH (buffer LCD)
	SGN data output	SGNOUT port	Affiche sur les afficheurs 7 segments le contenu de l'espace G (buffer G)
	Word bin to ASCII HEX format	HEX s,d	Convertit une valeur binaire 16 bits en code hexadécimal (s,s+1) → 4 bytes à partir de (d)

	Double Word bin to ASCII HEX format	DHEX s,d	Convertit une valeur binaire 32 bits en code hexadécimal(s,s+1) → 4 bytes à partir de (d)
Affi-Cha-ge	Word bin to ASCII DEC format	ASC s,d,n	Convertit une valeur 16 bits en code ASCII décimal (s) → n bytes depuis (d)
	Double Word bin to ASCII DEC format	DASC s,d,n	Convertit une valeur 32 bits en code ASCII décimal (s,s+1) → n bytes depuis (d)
Sauts	Goto Label	GOTO Label LABEL Label	Commande de saut Commande de création de label (étiquette)
	Subroutine CALL RET	CALLS label SBRT label RET	Appelle un sous programme Définit un sous programme Commande RETURN
	Looping	LOOP label ,n	Répète la commande de saut au label jusqu'à n = 0
Autres commandes	Key metrix scan	KEYSCAN	Commande pour scanner un clavier matriciel maximum 8x8 touches .Utilisé pour les claviers à membrane
	Output all off	OUTOFF	Toutes les sorties sont à OFF
	Distribute	DIST s,d,n	Divise une valeur 16 bits par 4 bits vers d
	Combine	UNIT s,d,n	Fait une valeur 16 bits en prenant les 4 bits de poids faible pour faire un mot
	Encode	ENCO s,d	Code les 4 bits de poids faible de s et le stocke à l'emplacement d
	Decode	DECO s,d	Décode les 16 bits de s et stocke les 4 bits de poids faible
	Tremometer input	THIN port ,d	Mesure la température d'un thermomètre DS1820 et stocke la valeur à l'emplacement d

Explication des nombres traités par TinyPLC

Avant d'expliquer les commandes, les systèmes numériques et les différents types de code doivent être expliqués, le tableau suivant donne les correspondances dans les différentes bases.

Dec.	Binaire	Hexa	BCD
0	0000	0	0000
1	0001	1	0001
2	0010	2	0010
3	0011	3	0011
4	0100	4	0100
5	0101	5	0101
6	0110	6	0110
7	0111	7	0111
8	1000	8	1000
9	1001	9	1001
10	1010	A	
11	1011	B	
12	1100	C	
13	1101	D	
14	1110	E	
15	1111	F	

Un nombre binaire de 4 caractères peut être exprimé par un nombre hexadécimal d'un seul caractère

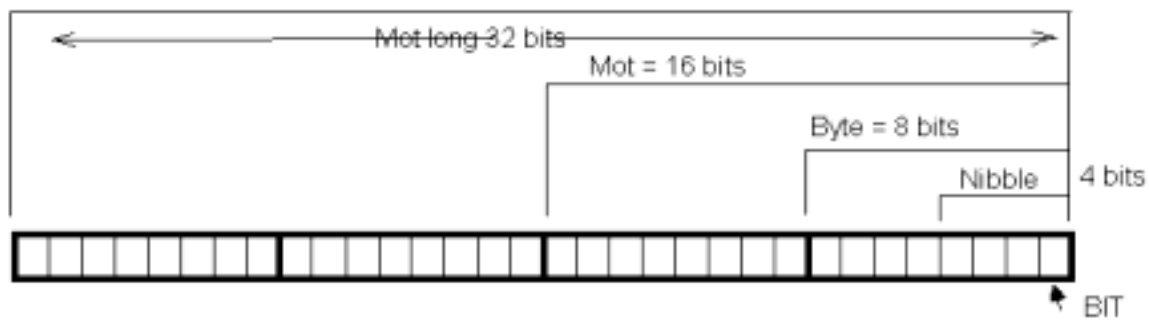
Le code BCD est identique au code binaire, mais se limite de 0 à 9 Exemple :
100 décimal à l'origine = 01100100 (64H) en hexa
= 0001 0000 0000 (100H) en BCD

A à F sont utilisés pour exprimer en hexa les valeurs 10 à 15.

Expression des nombres dans un programme LADDER

Binaire : 00011010B
Décimal : 123, 80
Hexadécimal : 0ABH, 12345H

Les unités utilisées sont : le bit, le byte (8 bits), le mot (16 bits) et le mot long (32 bits)

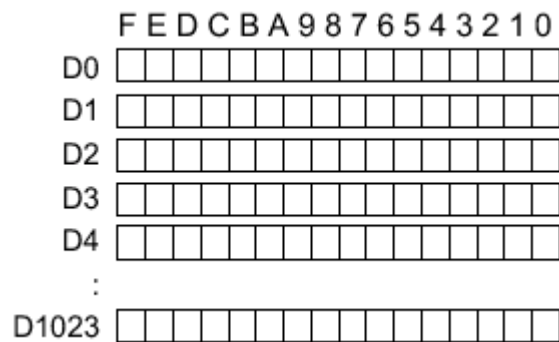


Champs par Byte	Champs par Mot
P (entrées / sorties)	T (tempos)
K (valeurs sauvegar.)	C (compteurs)
M (relais aux)	D (données)
CH (Affichage LCD)	AD (résultat)
G (affichage SGN)	CNT (compt rapide)

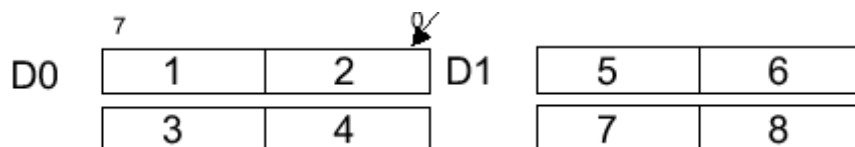
	7 6 5 4 3 2 1 0
P0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> A
P1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
P2	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
:	
P15	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Le champ P est séparé par l'unité BYTE, P0 :0 se réfère au bit marqué A. L'expression P0 :8 est donc invalide, les valeurs vont de 0 à 7

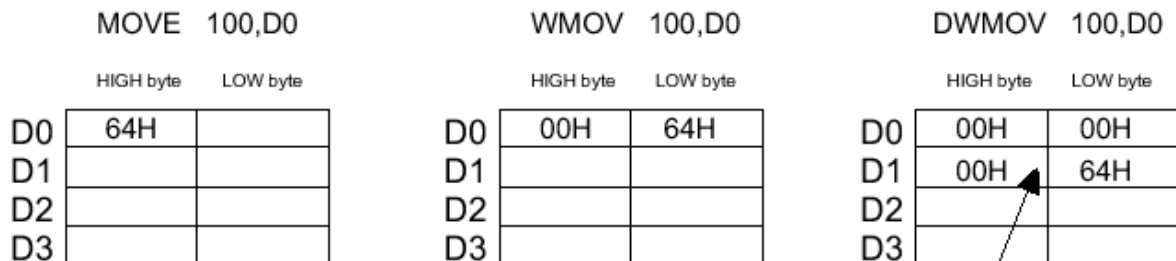
Dans le champ D, les données sont stockées par mot (16 bits)



Les commandes logiques et arithmétiques de TinyPLC sont en général par mots , certaines commandes sont en mots longs (32 bits) . Un mot long est stocké en utilisant, l'adresse de base et l'adresse suivante . Par exemple si vous faites la commande DWMOV 12345678H,D0 (mettre 12345678H à l'adresse D0) Les adresses D0 et D1 seront utilisées pour stocker la donnée



Exemple de transfert d'une valeur par Byte , Mot ou Mot long



Code ASCII

Bits de poids faible

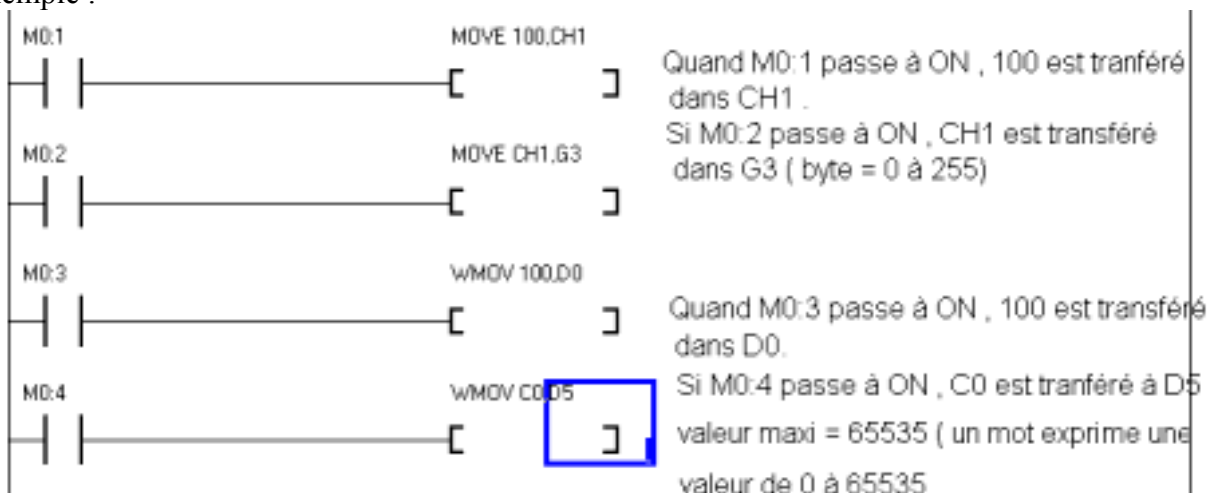
Bits de poids forts		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
	3		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
	4		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	5		P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^
	6		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
	7		p	q	r	s	t	u	v	w	x	y	z	{		}	

MOVE , WMOV

Syntaxe : MOVE s, d
 WMOV s,d

Déplacement de 8 bits, transfère la valeur de s vers d
Déplacement d'un mot 16 bits , la valeur (ou constante) est transférée vers d

Exemple :



Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	

Précautions en utilisant les champs C, T et D avec MOVE

La commande MOVE transfère les données par byte et C T D utilisent des données par mots, il faut alors prendre certaines précautions . La commande MOVE 100,D0 va stocker 100 dans le poids fort de D0 et la valeur de D0 sera modifiée , il vaut mieux donc utiliser WMOVE

Précautions en utilisant les champs P, M et K avec WMOVE

La commande WMOVE transfère les données par mots (16 bits) et les champs P, M et K travaillent par Byte , si vous faites WMOV 1234H,CH0 , 12h est stocké en CH0 mais 34H est stocké en CH1

DWMOV

Syntaxe : DWMOV s,d Commande de déplacement de donnée mot long 32 bits de s vers d



Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	

Si M0:1 passe à ON , 0 est transféré dans D0 et 100 dans D1

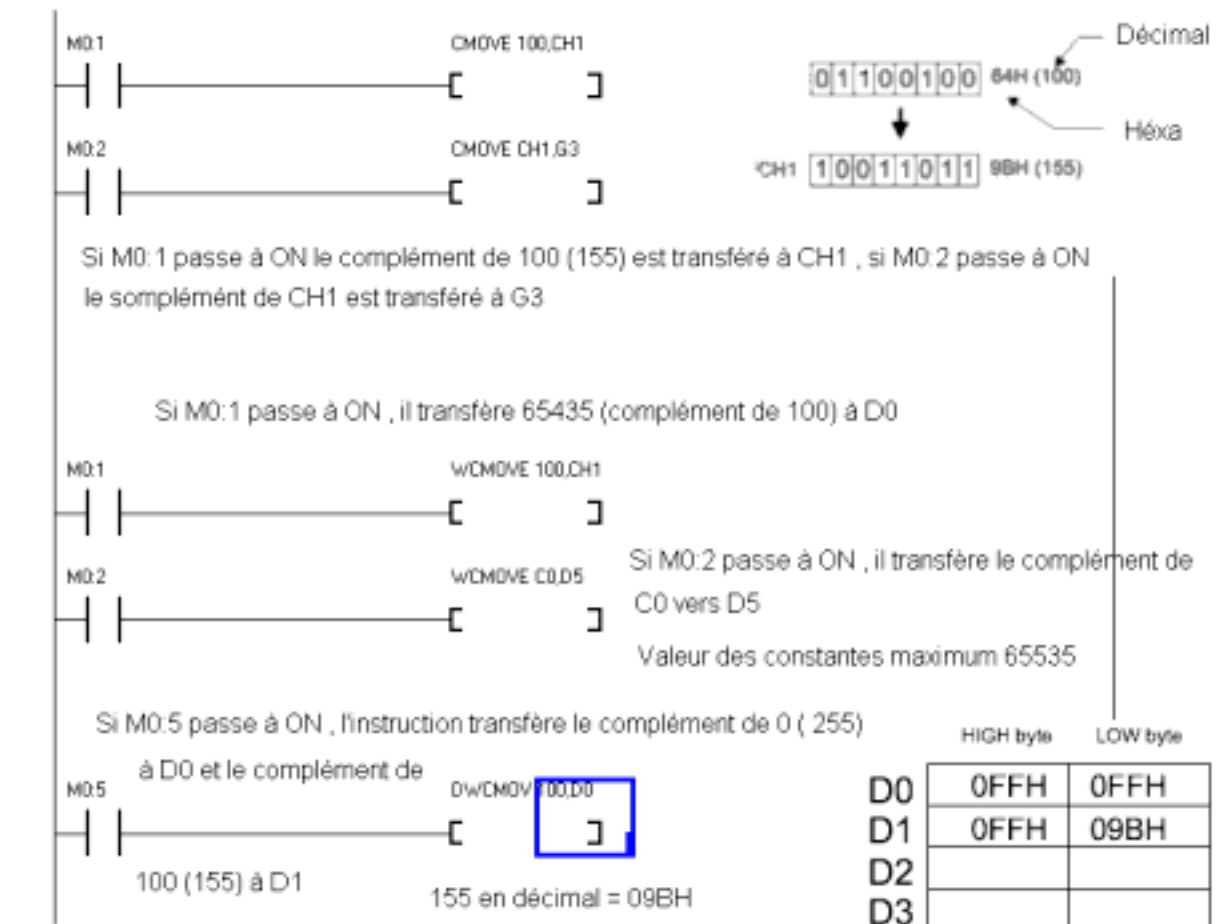
Vous ne pouvez pas utiliser de valeur supérieure à 2 147 418 112 (7FFF0000H) pour les constantes

	HIGH byte	LOW byte
D0	00H	00H
D1	00H	64H
D2		
D3		

CMOVE , WCMOV , DWCMOV

Syntaxe : CMOVE s,d
WCMOV s,d
DWCMOV s,d

Transfert le complément de la donnée 8 bits (CMOVE) ,16 bits (WCMOV) ou 32 bits de s vers d .



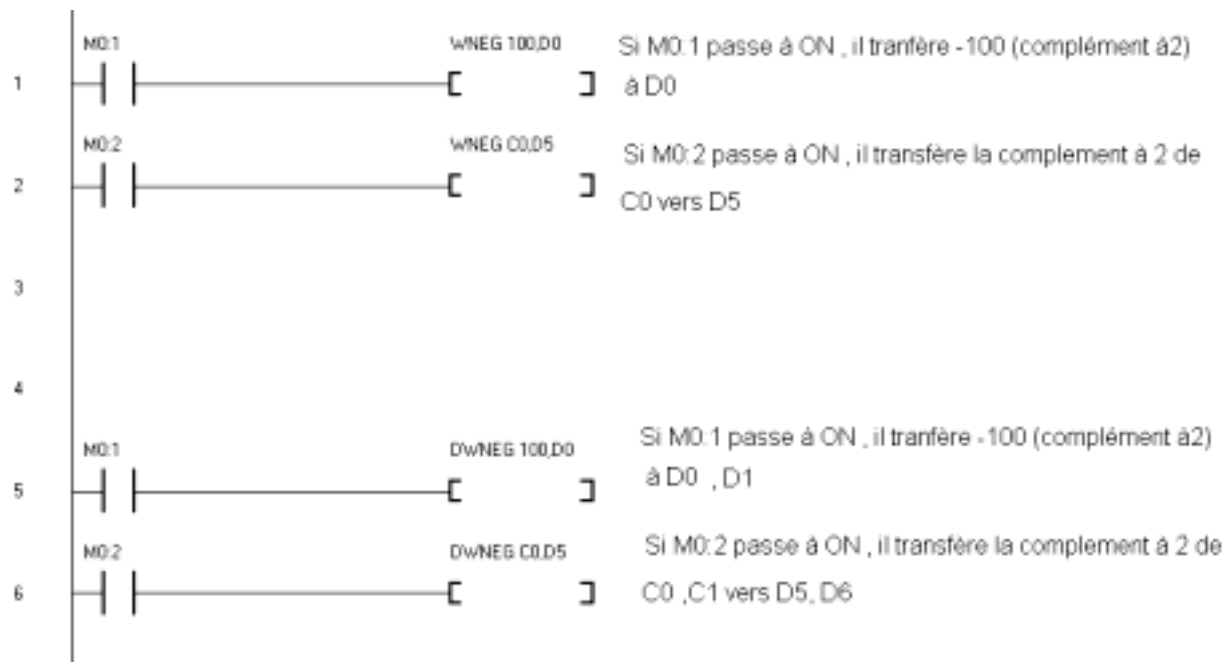
Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	

WNEG , DWNEG

Syntaxe : WNEG s, d
 DWNEG s, d

Effectue le transfert d'un complément à 2 d'un mot 16 bits (WNEG) ou d'un mot long 32 bits (DWNEG) de l'emplacement s vers l'emplacement d



Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	

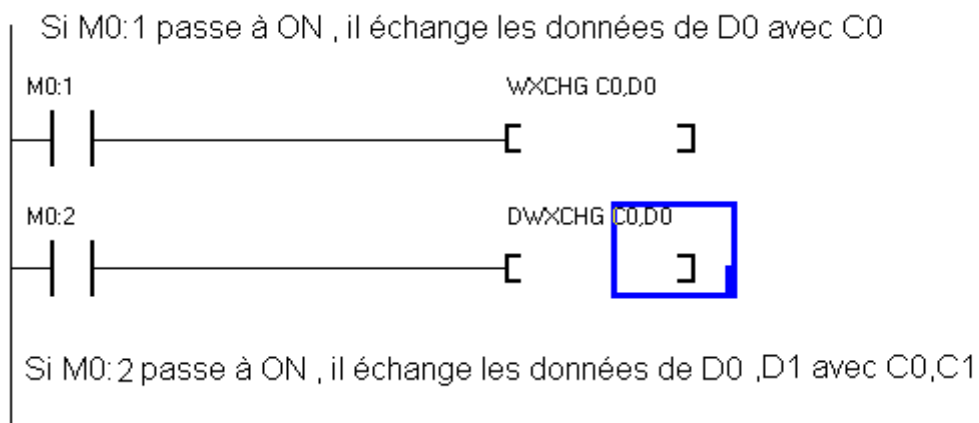
Qu'est le complément à 2

Sur les ordinateurs pour faire un nombre négatif, on peut faire un complément à 2, on fait d'abord un complément normal et l'on ajoute ensuite 1. Dans TinyPLC, il n'existe pas de commande spécifique pour faire un nombre négatif, nous devons utiliser le complément à 2

WXCHG , DWXCHG

Syntaxe : WXCHG s,d
 DWXCHG s,d

Effectue un échange de la valeur de s avec la valeur de d, mot de 16 bits (WXCHG) ou mot long de 32 bits (DWXCHG)



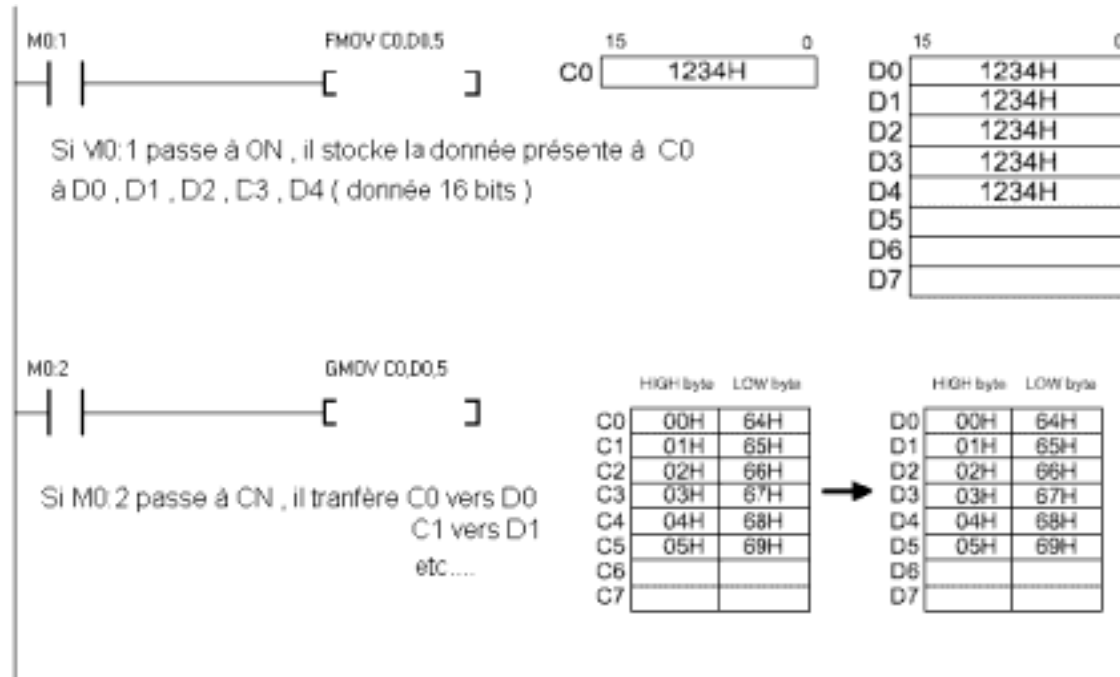
Commandes	Relais					Comp	Timer	Data	Etc ..				Constantes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	
D (destina.)	■	■		■		■	■	■	■	■	■	■	

WXCHG C0,D0



FMOV , GMOV

Syntaxe : FMOV s,d,n FMOV écrit le mot 16 bits (ou une constante) présent à l'adresse s, à l'adresse d , n nombre de fois
 GMOV s,d,n GMOV :commande de transfert de groupe de données diverses
 .Elle transfère le nombre n de valeurs à partir de s vers d

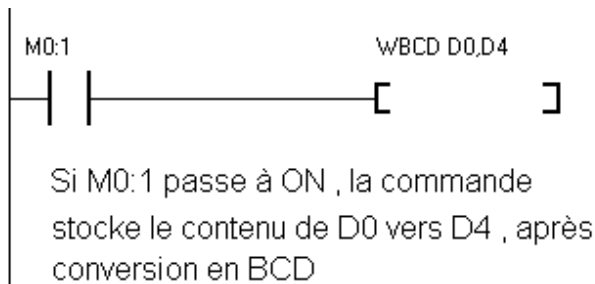


Opérandes

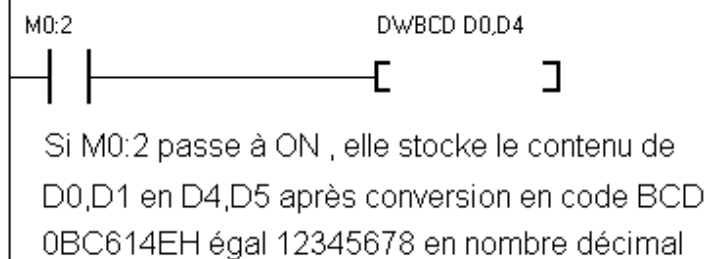
Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	
N (nombre)													■

WBCD , DWBCD

Syntaxe : WBCD s, d , conversion d'une donnée 16 bits en une donnée BCD 4 digits
 DWBCD s, d , conversion d'une donnée 32 bits en une donnée BCD 8 digits



	HIGH byte	LOW byte
D0	00H	64H
D1		
D2		
D3		
D4	01H	00H
D5		
D6		
D7		



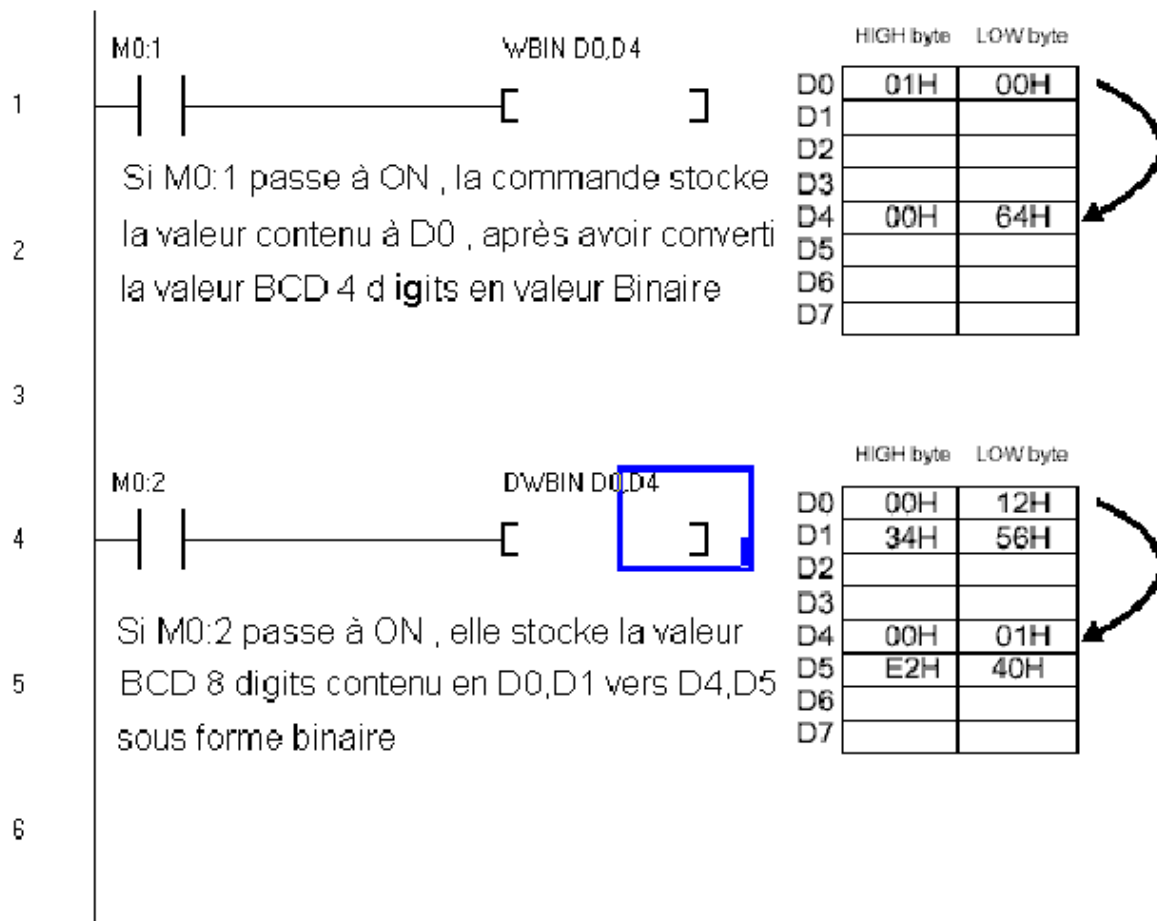
	HIGH byte	LOW byte
D0	00H	BCH
D1	61H	4EH
D2		
D3		
D4	12H	34H
D5	56H	78H
D6		
D7		

Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	

WBIN , DWBIN

Syntaxe : WBIN s,d converti une valeur BCD 4 digits en une valeur binaire 16 bits
 DWBIN s,d converti une valeur BCD 8 digits en une valeur binaire 32 bits

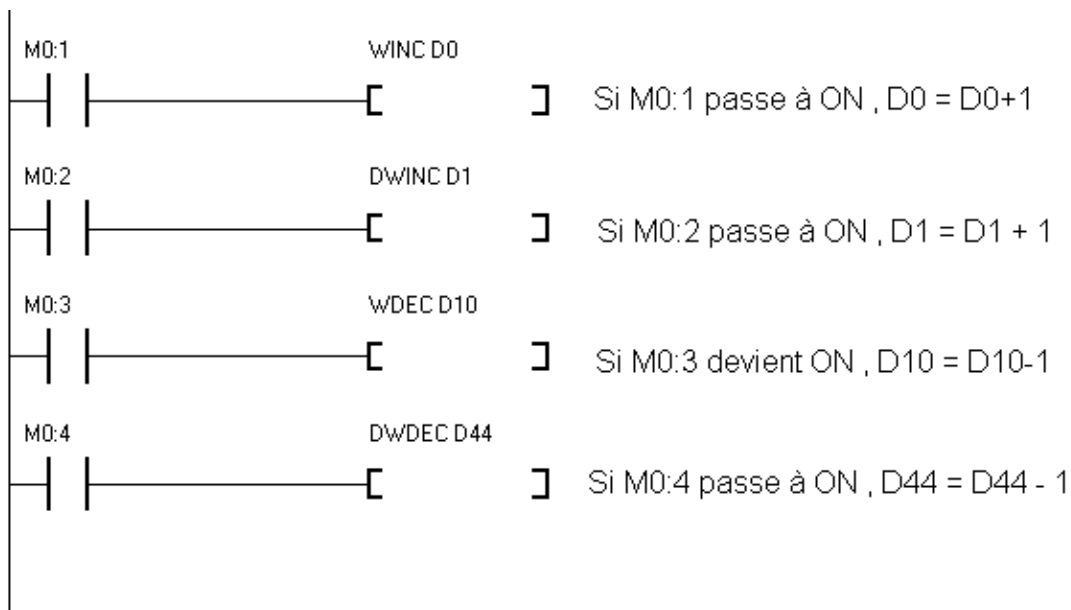


Opérandes

Commandes	Relais					Comp	Timer	Data	Etc ..				Constantes
	P	M	F	K	S				AD	Cnt	CH	G	
S (source)	■	■		■		■	■	■	■	■	■	■	■
D (destina.)	■	■		■		■	■	■	■	■	■	■	

WINC , DWINC , WDEC , DWDEC

Syntaxe : WINC d ajoute 1 à la valeur 16 Bits contenue à l'adresse d
 DWINC d ajoute 1 à la valeur 32 Bits contenu à l'adresse d
 WDEC d diminue de 1 la valeur 16 bits contenue à l'adresse d
 DWDEC d diminue de 1 la valeur 32 bits contenue à l'adresse d

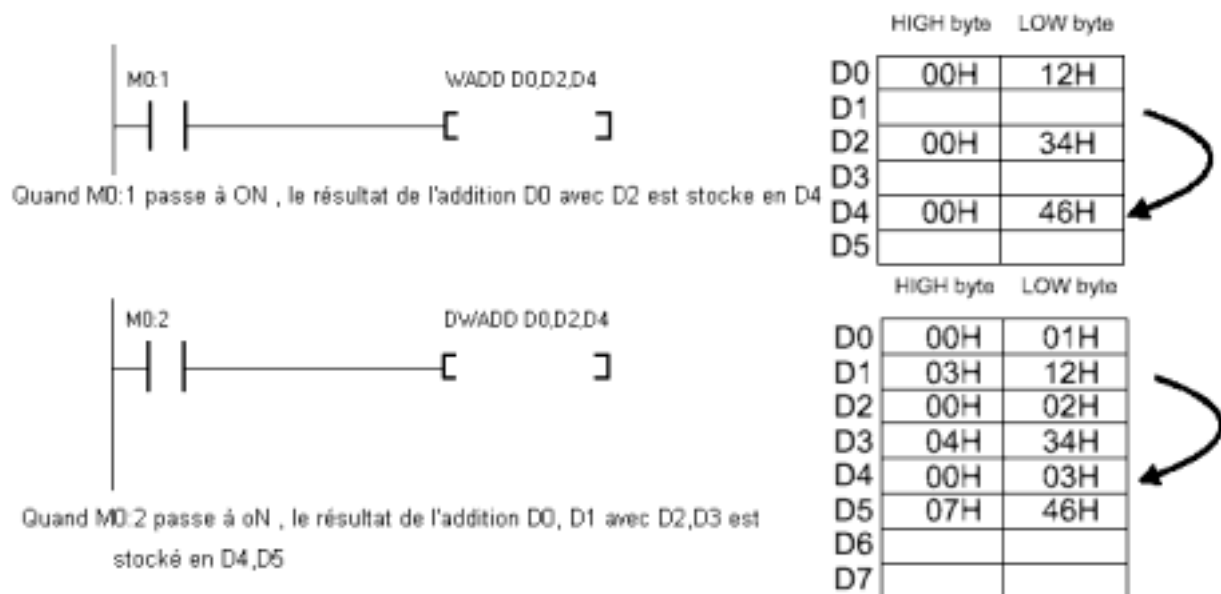


Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
d						■	■	■	■	■			

WADD , DWADD

Syntaxe : WADD s1,s2,d commande addition 16 bits, le résultat de l'addition de s1 et s2 est stocké en d
 DWADD s1,s2,d commande addition 32 bits, le résultat de l'addition de s1 et s2 est stocké en d

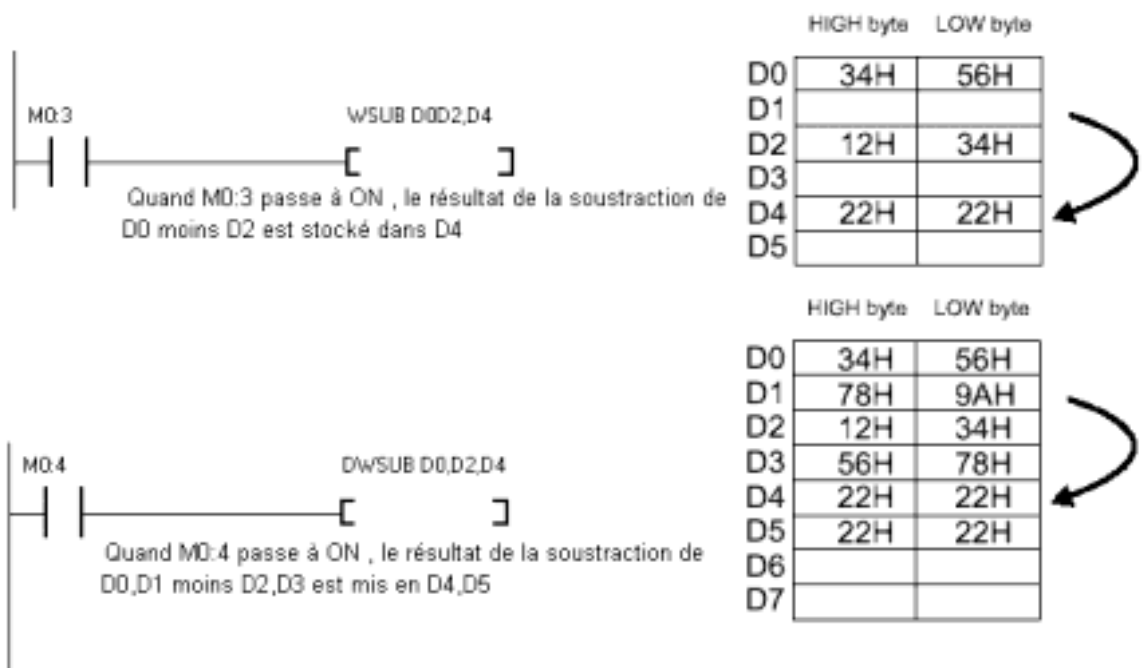


Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WSUB , DWSUB

Syntaxe : WSUB s1,s2,d commande soustraction 16 bits , le résultat de la soustraction de s1 et s2 est stocké en d
 DWSUB s1,s2,d commande soustraction 32 bits , le résultat de la soustraction de s1 et s2 est stocké en d

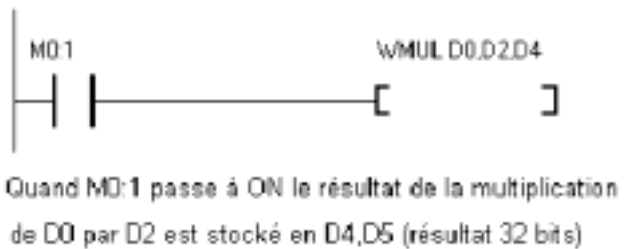


Opérandes

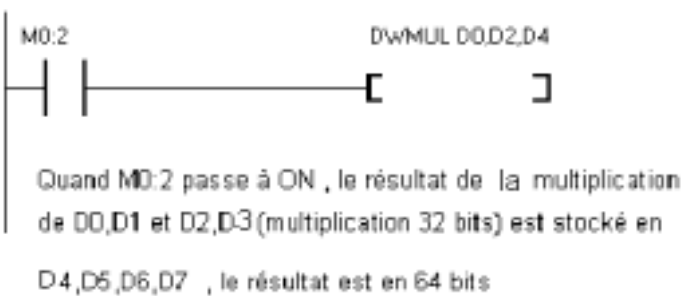
Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WMUL , DWMUL

Syntaxe : WMUL s1,s2,d multiplication 16 bits , le résultat de la multiplication de s1 par s2 est mis en d
 DWMUL s1,s2,d multiplication 32 bits , le résultat de la multiplication de s1 par s2 est mis en d



	HIGH byte	LOW byte
D0	01H	23H
D1		
D2	04H	56H
D3		
D4	00H	04H
D5	EDH	C2H
D6		
D7		



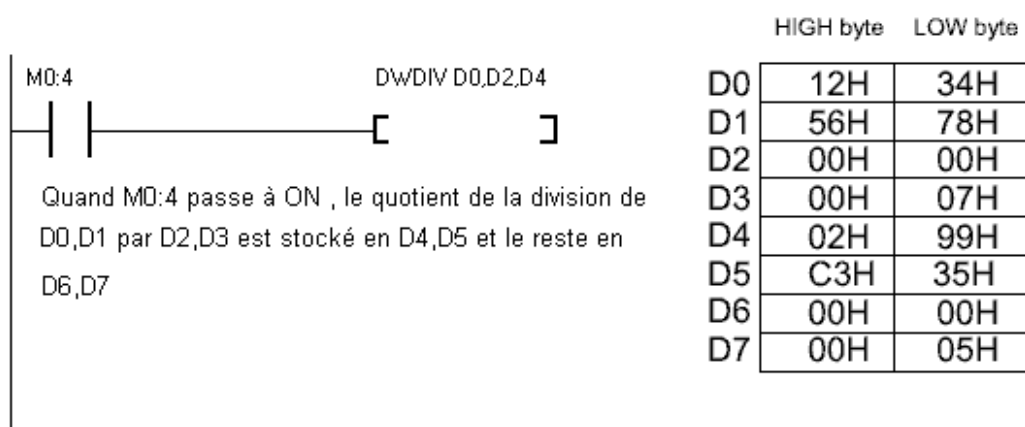
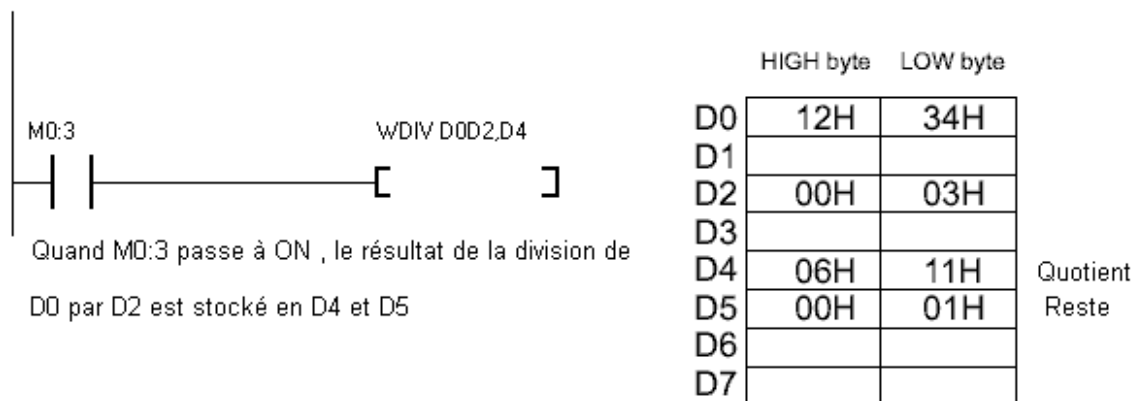
	HIGH byte	LOW byte
D0	00H	12H
D1	34H	56H
D2	00H	33H
D3	33H	33H
D4	00H	00H
D5	00H	00H
D6	7FH	70H
D7	FDH	6CH

Opérandes

Commandes	Relais					Comp	Timer	Data	Etc ..				Constantes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WDIV , DWDIV

Syntaxe : WDIV s1,s2,d division 16 bits , le quotient de la division de s1 par s2 est mis en d et le reste en d+1
 DWDIV s1,s2,d division 32 bits , le quotient de la division de s1 pas s2 est mis en d et le reste en D+1

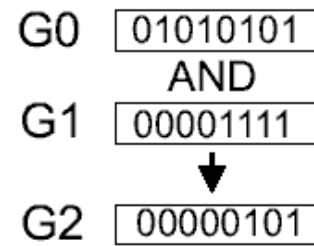
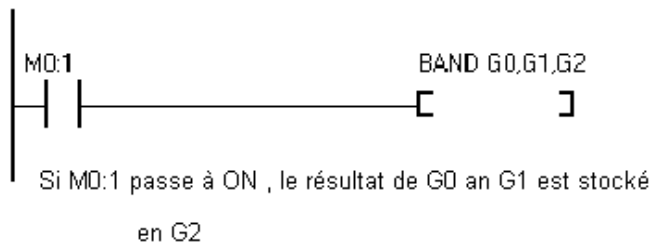


Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					



Syntaxe : BAND s1,s2,d Commande AND 8 bits , le résultat de s1 AND s2 est mis en d

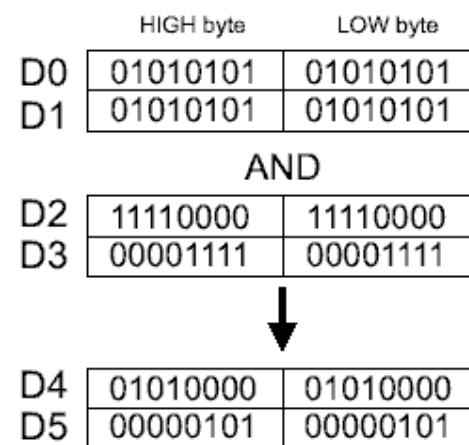
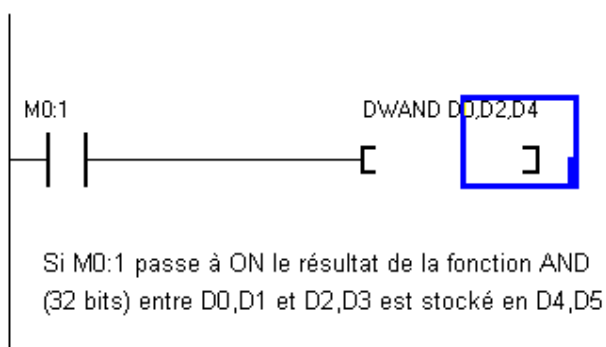
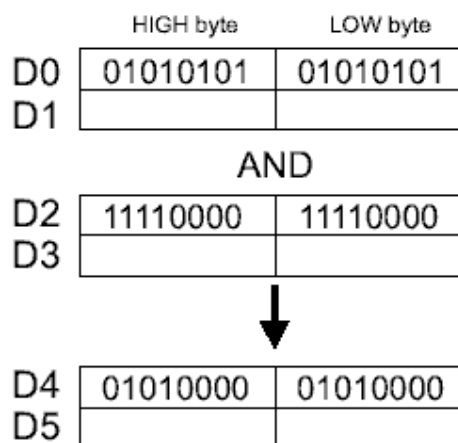
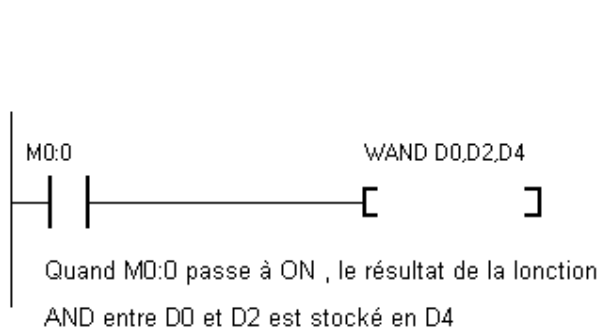


Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WAND , DWAND

Syntaxe : WAND s1,s2,d Commande AND 16 bits , le résultat de s1 AND s2 est mis en d
 DWAND s1,s2,d commande AND 32 bits , le résultat de s1 AND s2 est mis en d

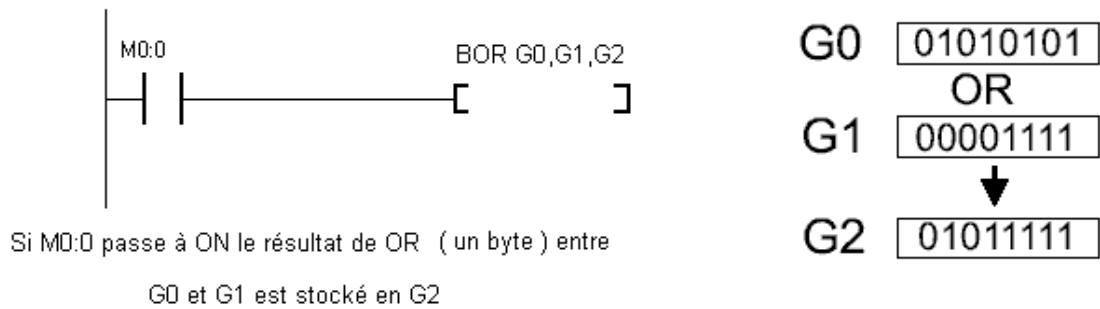


Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

BOR

Syntaxe : BOR s1,s2,d Commande OR 8 bits , le résultat de s1 OR s2 est mis en d

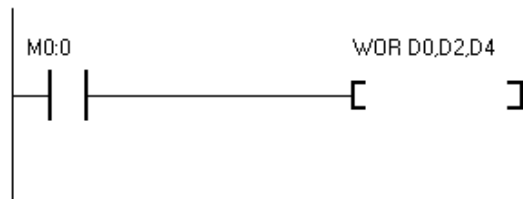


Opérandes

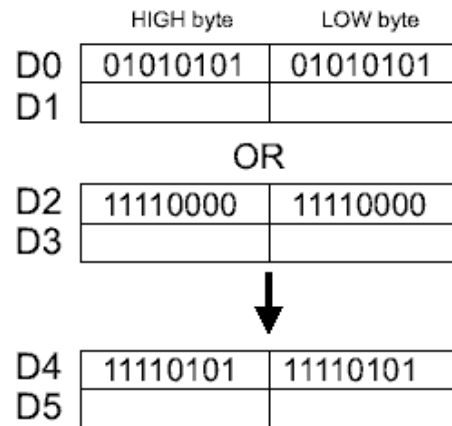
Commandes	Relais					Com p	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WOR , DWOR

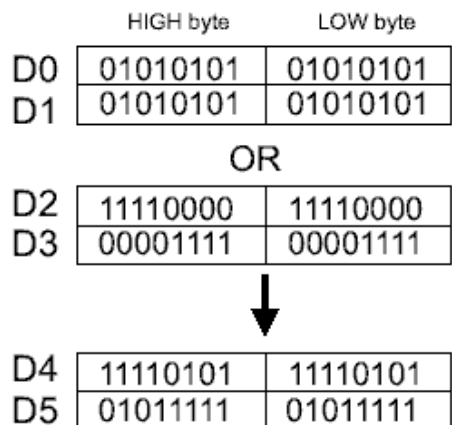
Syntaxe : WOR s1,s2,d Commande OR 16 bits , le résultat de s1 OR s2 est mis en d
 DWOR s1,s2,d commande OR 32 bits , le résultat de s1 OR s2 est mis en d



Si M0:0 passe à ON le résultat de OR (16 bits) entre D0 et D1
 est stocké en D4



Quand M0:1 passe à ON , le résultat de OR (32 bits)
 entre D0,D1 et D2,D3 est stocké en D4,D5

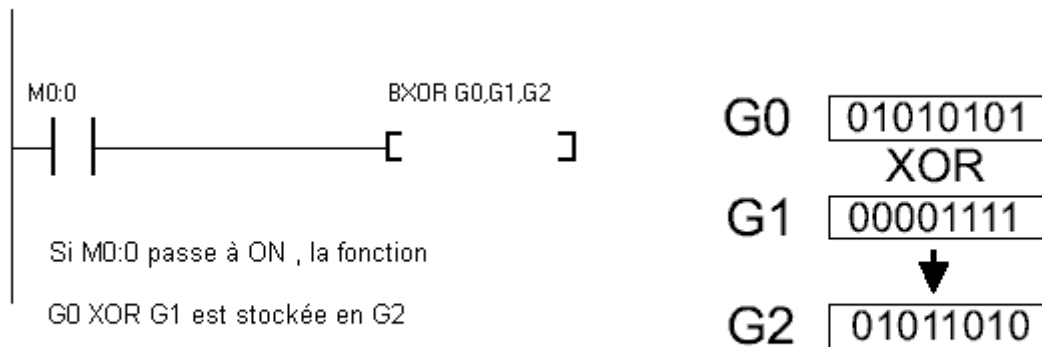


Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

BXOR

Syntaxe : BXOR s1,s2,d commande XOR 8 bits entre s1 et s2 résultat en d



Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WXOR , DWXOR

WXOR s1,s2,d commande XOR 16 bits entre s1 et s2 résultat en d

DWXOR s1,s2,d commande XOR 32 bits entre s1 et s2 résultat en d



Quand M0:0 passe à ON , le résultat de
D0 XOR D1 est transféré en D4

	HIGH byte	LOW byte
D0	01010101	01010101
D1		

XOR

D2	11110000	11110000
D3		



D4	10100101	10100101
D5		



Quand M0:1 passe à ON ,le résultat de la fonction
D0,D1 XOR D2,D3 est stocké en D4,D5

	HIGH byte	LOW byte
D0	01010101	01010101
D1	01010101	01010101

XOR

D2	11110000	11110000
D3	00001111	00001111



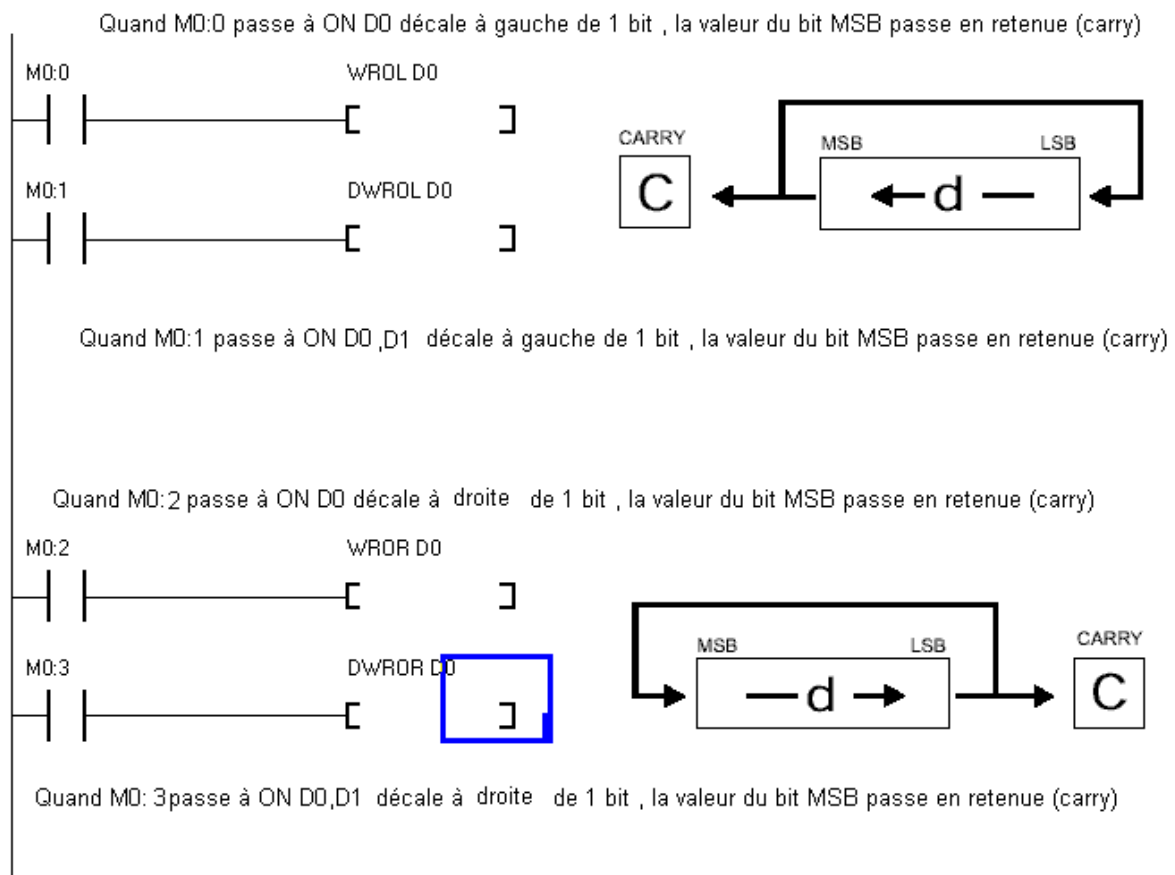
D4	10100101	10100101
D5	01011010	01011010

Opérandes

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S1						■	■	■	■	■			■
S2						■	■	■	■	■			■
D						■	■	■					

WROL ,DWROL, WROR , DWROR

Syntaxe : WROL d , $d=d \ll 1$, commande de décalage à gauche de 1 bit (commande sur 16 bits)
 DWROL d , $d=d \ll 1$, commande de décalage à gauche de 1 bit (commande sur 32 bits)
 WROR d , $d=d \ll 1$, commande de décalage à droite de 1 bit (commande sur 16 bits)
 DWROR d , $d=d \ll 1$, commande de décalage à droite de 1 bit (commande sur 32 bits)

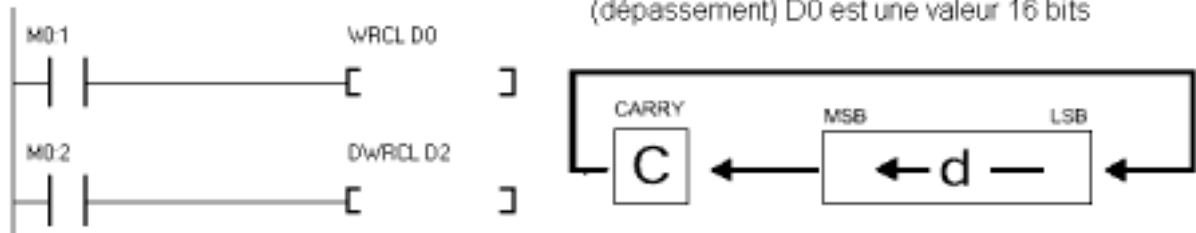


Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
D	■	■		■		■	■	■	■	■	■	■	

WRCL , DWRCL

Syntaxe : WRCL d d = d << 1 with c Commande de décalage à gauche de 1 bit (sur 16 bits) en incluant le bit Carry (dépassement)
 DWRCL d d = d << 1 with c Commande de décalage à gauche de 1 bit (sur 32 bits) en incluant le bit Carry (dépassement)

Si M0:1 passe à ON , D0 est décalé de 1 bit à gauche , le bit MSB se décale en Carry
 (dépassement) D0 est une valeur 16 bits

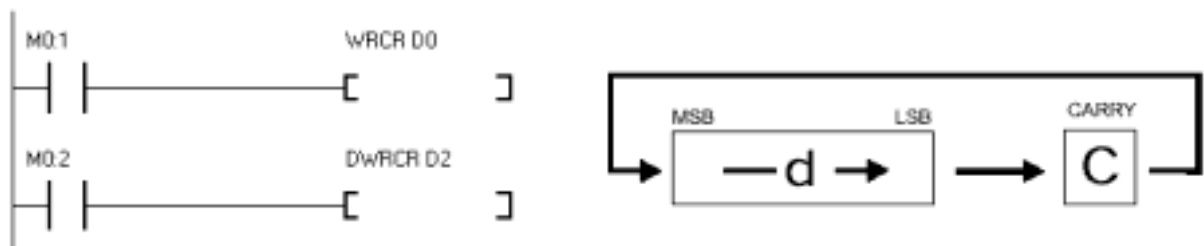


Si M0:2 passe à ON , la valeur 32 bits D2,D3 est décalée de 1 bit vers la gauche en incluant le drapeau dépassement

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
D	■	■		■		■	■	■	■	■	■	■	

WRCR , DWRCR

Syntaxe : WRCR d d = d >> 1 with c Commande de décalage à droite de 1 bit (sur 16 bits) en incluant le bit Carry (dépassement)
 DWRCR d d = d >> 1 with c Commande de décalage à droite de 1 bit (sur 32 bits) en incluant le bit Carry (dépassement)



Si M0:1 passe à ON , la donnée 16 bits est décalée de 1 bit vers la droite en incluant C

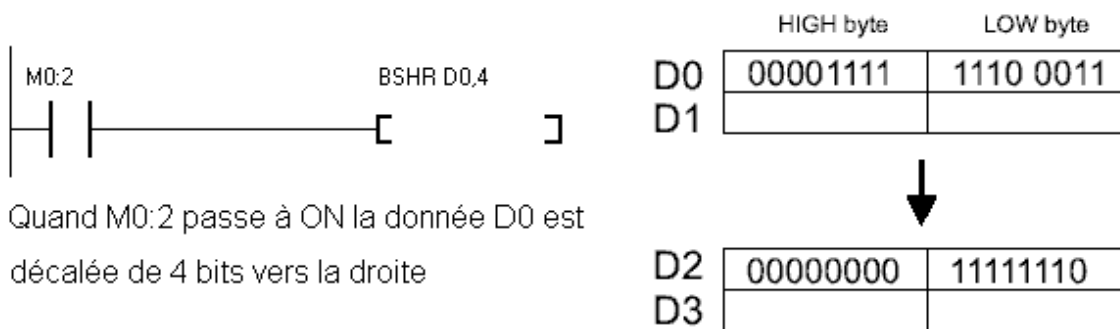
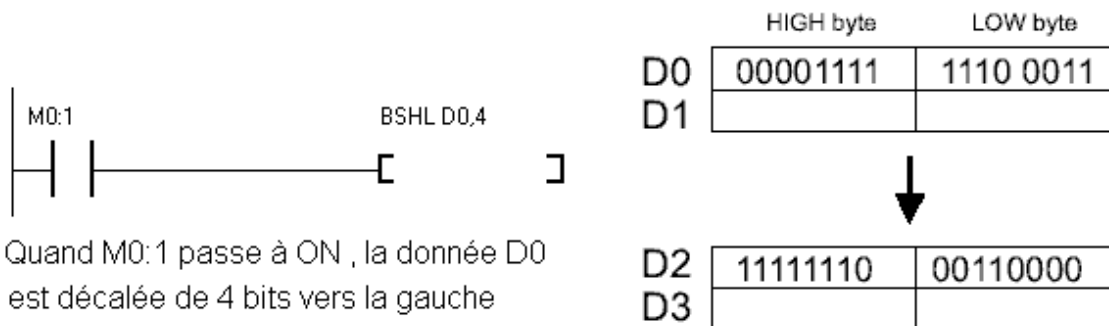
Si M0:2 passe à ON , la donnée 32 bits D2,D3 est décalée de 1 bit vers la droite en incluant le bit Carry

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
D	■	■		■		■	■	■	■	■	■	■	

BSHL ,BSHR

Syntaxe : BSHL d,n d = d << n Commande de décalage de la donnée 16 bits à gauche de n bits , les nouveaux bits insérés sont mis à zéro

BSHR d,n d = d >> n Commande de décalage de la donnée 16 bits à droite de n bits , les nouveaux bits insérés sont mis à zéro



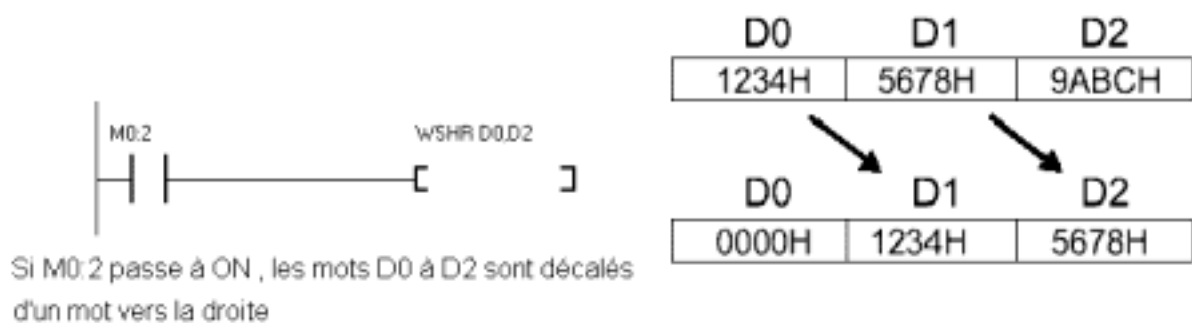
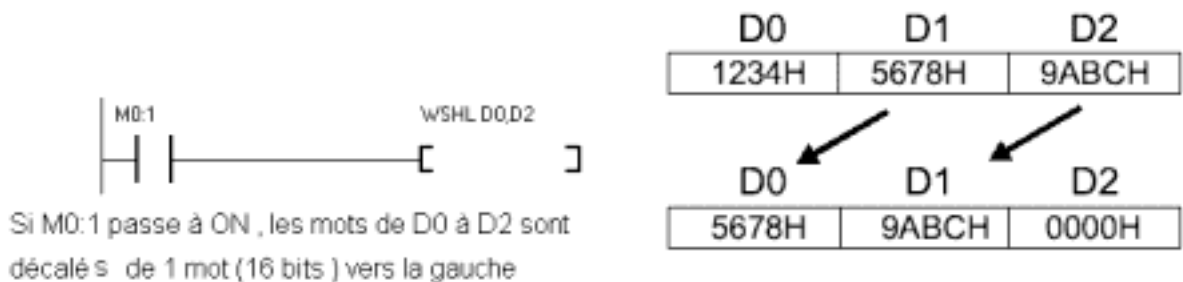
BSHL et BSHR sont utilisables sur des données 16 bits

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
D	■	■		■		■	■	■	■	■	■	■	
n												■	

WSHL , WSHR

Syntaxe : WSHL s1,s2 commande de décalage à gauche de la donnée s1 à la donnée s2 de un mot (16 bits) , le nouveau mot inséré est mis à zéro

WSHL s1,s2 commande de décalage à droite de la donnée s1 à la donnée s2 de un mot (16 bits) , le nouveau mot inséré est mis à zéro



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
D	■	■		■		■	■	■	■	■	■	■	

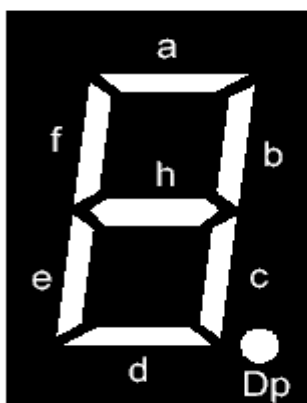
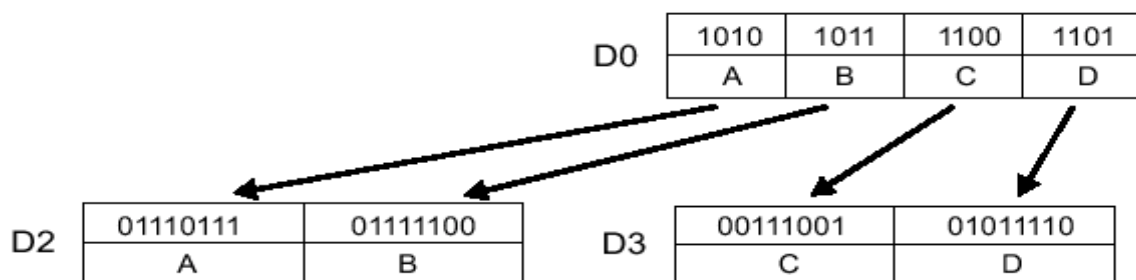


Syntaxe : SEG s,d , Convertit une donnée 16 bits (mot) , en 4 digits 7 segments



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
s	■	■		■		■	■	■	■	■	■	■	■
d	■	■		■		■	■	■			■	■	

Si M0 :1 passe à ON , la donnée 16 bits présente en D0 est convertie en donnée pour 4 digits de 7 segments , stockée en deux mots (4 bytes) en D2 et D3



	Dp	g	f	e	d	c	b	a	Affich.
0	0	0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	1	0	1
2	0	1	0	1	1	0	1	1	2
3	0	1	0	0	1	1	1	1	3
4	0	1	1	0	0	1	1	0	4
5	0	1	1	0	1	1	0	1	5
6	0	1	1	1	1	1	0	1	6
7	0	0	1	0	0	1	1	1	7
8	0	1	1	1	1	1	1	1	8
9	0	1	1	0	1	1	1	1	9
A	0	1	1	1	0	1	1	1	A
B	0	1	1	1	1	1	0	0	B
C	0	0	1	1	1	0	0	1	C
D	0	1	0	1	1	1	1	0	D
E	0	1	1	1	1	0	0	1	E
F	0	1	1	1	0	0	0	1	F

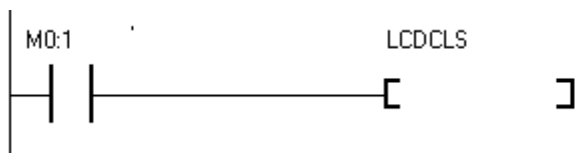
SEG (SUITE)

Il existe deux possibilités pour connecter des afficheurs 7 segments sur les TinyPLC , la première est d'utiliser des afficheurs 7 segments standards directement sur les sorties du PLC en utilisant la commande SEG , la seconde est d'utiliser des modules SGN série commercialisés par Comfile .

Il est plus économique d'utiliser la commande SEG seulement si vous avez seulement un ou deux afficheurs à contrôler, ou si vous avez des sorties qui ne sont pas utilisées. Le module SGN utilise une seule ligne de sortie et laisse toutes les autres disponibles.

LCDCLS

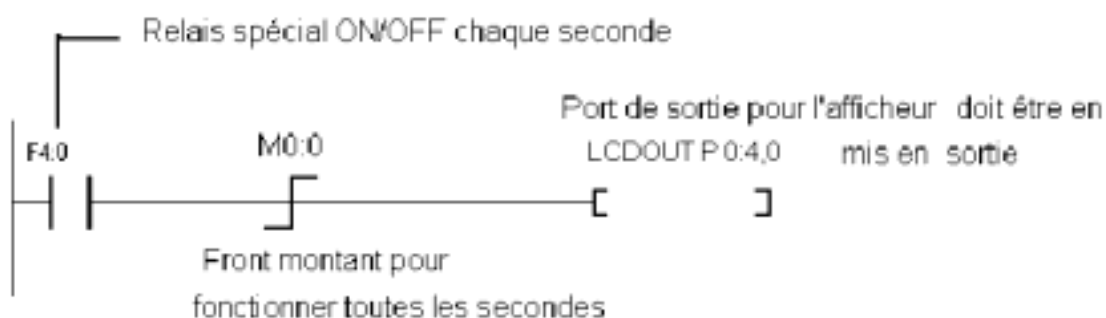
Syntaxe : LCDCLS Remplit le champ du buffer LCD avec le code 20H (code ASCII de ESPACE)



Sans effet sur l'afficheur sans la commande LCDOUT . Cette commande efface seulement le contenu du champ CH .

LCDOUT

Syntaxe : LCDOUT port , mode
 Le port indique où est connecté le module LCD par exemple P0 :4
 Mode indique le type de LCD connecté :
 0 : module LCD 2 lignes 16 caractères (anglais) ELCD162
 1 : module LCD 4 lignes 16 caractères (anglais) ELCD164
 2 : module LCD 4 lignes 20 caractères (anglais) ELCD204
 3 : module LCD Coréen HLCD112 ou HLCD114

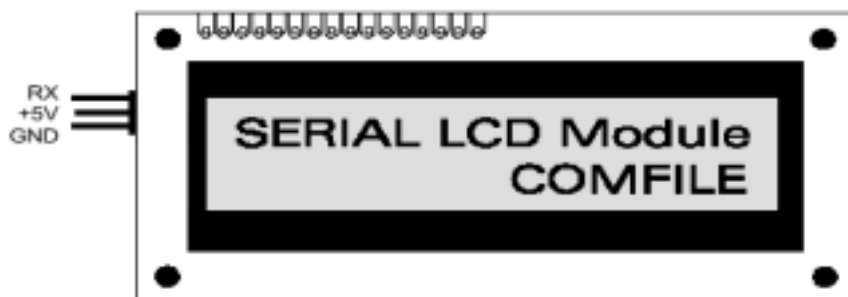


F0 :4 est un relais spécial qui répète ON/OFF toutes les secondes, il permet en utilisant la front montant de faire fonctionner l’affichage du LCD toutes les secondes . L’instruction LCDOUT transmet les données du domaine CH à l’afficheur (ceci prend de 200 à 500 ms)

Uniquement les produits LCD produit par COMFILE sont utilisables par cette commande . Le mode 19200 bauds doit être sélectionné par jumpers

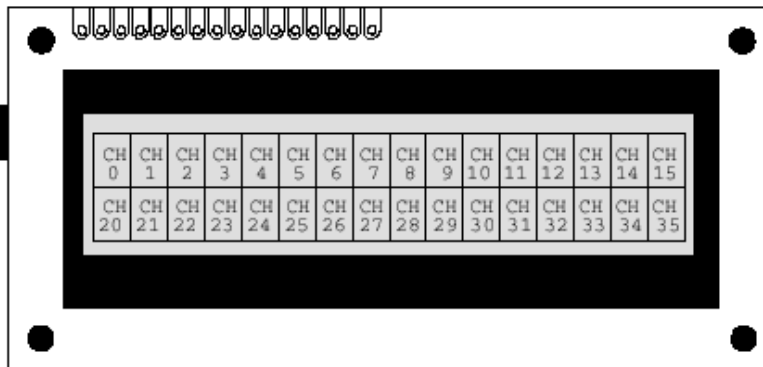
A propos des modules LCD (anglais)

Les modules LCD produits par Comfile sont différents des autres modules du fait qu’il sont alimentés uniquement par 3 fils (0v , +5v et Rx) contrairement aux modules LCD qui sont alimentés en 14 fils . RX = transmission de données , 19200 bauds .L’instruction LCDOUT transmet toutes les données du domaine CH par la RS232 (code ASCII)

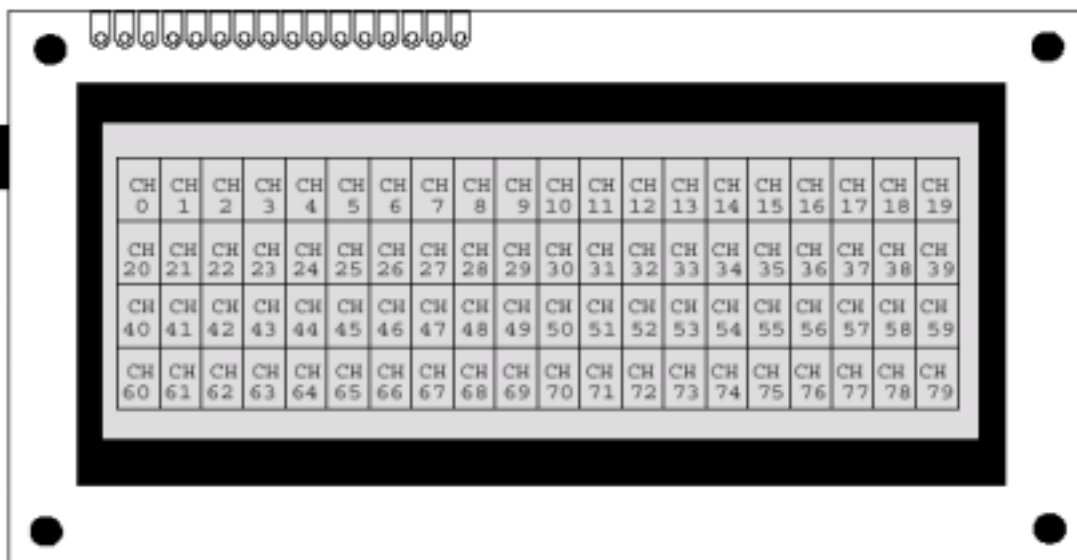


La position sur l’afficheur est indiquée par la position CH , si vous mettez CH0 , la position correspond à la première ligne caractère de gauche . Figure pour indiquer les relations CH du module 2 lignes de 16 caractères.

Première ligne CH0 , deuxième CH20



Module 4 lignes 20 caractères

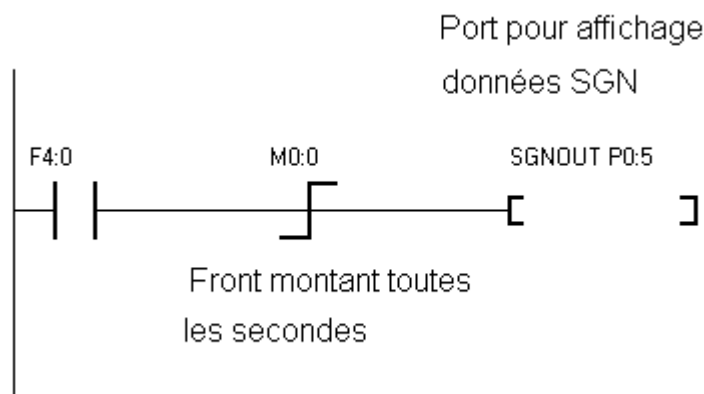


Si vous commandez plusieurs modules simultanément , il est nécessaire de respecter le temps de transfert de 200 à 500 ms pour éviter tout conflit entre les différents modules .Ceci est aussi applicable à l'instruction SGNOUT

Modèles	Produits	Affichage
ELCD162	2 lignes 16 caract.	64.5 * 13.8
ELCD162-BL	2 lignes 16 caract rétro-eclair	64.5 * 13.8
ELCD164	4 lignes 16 caract	61.8 * 25.2
ELCD164-BL	4 lignes 16 caract retro-éclair	61.8 * 25.2
ELCD204	4 lignes 20 caract	76 * 25.2
ELCD204-BL	4 lignes 20 caract rétro-éclair	76 * 25.2
ELCD162-BIG-BL	2 lignes 16 grands caractèr rétro-éclairés	99 * 36
HLCD112-BL	Modeles pour caractères coréen 2 lignes 11 caract	60.5 * 18.5
HLCD114-BL	Modèles pour caractères coréen 4 lignes 11 caract	70.7 * 38.8

SGNOUT

Syntaxe : SGNOUT port , les données du domaine G sont transmises au module série 7 segments (SGN) .Port indique le port où est connecté le module (ex :P0 :5)



F4 :0 est un relais spécial qui commute ON/OFF toutes les secondes et ainsi fait exécuter l'instruction SGNOUT toutes les secondes . Cette instruction transmet les données du buffer G à l'afficheur SGN . Quand SGNOUT est exécuté toutes les données sont transmises (ce qui prend de 50 à 100 ms) Faire attention de ne pas transmettre d'autre informations en même temps .

Un afficheur SGN peut avoir au maximum 5 digits et 8 afficheurs maxi peuvent être connectés à un module .Un port peut donc contrôler un maximum de 40 digits .

Au sujet des modules SGN

Les modules d'affichage 7 segments sont fabriqués par Comfile Technology (SGN est l'abréviation de Seven seGment Network) . Ces modules sur circuits imprimé comporte un microcontrôleur qui allume les segments correspondants aux données transmises par une ligne RS232C à 9600 bauds alimentée en +5 volts. Les modules comportent 5 digits au maximum.



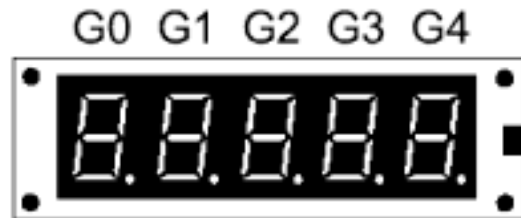
TinyPLC contient les instructions permettant de commander les modules

Un port I/O peut commander au maximum 8 modules de 5 digits



L'arrière des modules SGN comporte des DIP qui permettent de sélectionner , les adresses du domaine G concernées .

DIP	Adresse domaine G
0000	G0 ~ G4
0001	G5 ~ G9
0010	G10 ~ G14
0011	G15 ~ G19
0100	G20 ~ G24
0101	G25 ~ G29
0110	G30 ~ G34
0111	G35 ~ G39



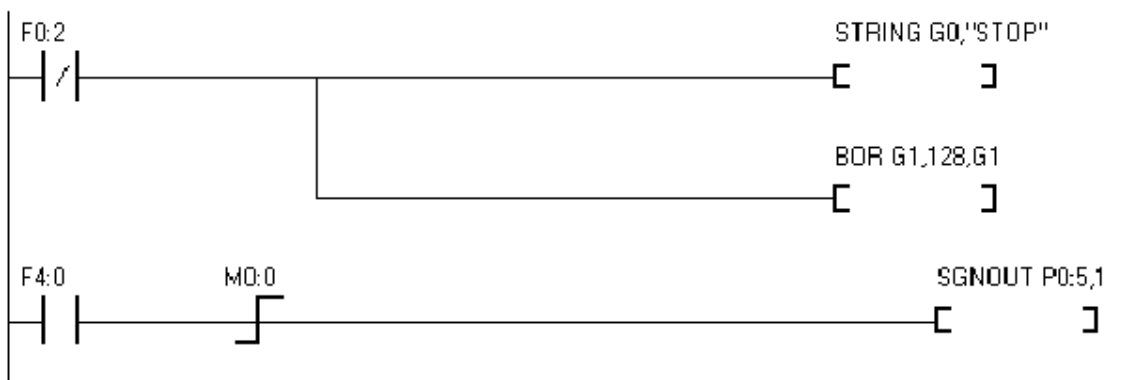
Les adresses de chaque module commencent par la gauche

Par exemple si 31H (code ASCII 1) est en G0 , 32H en G1 , 33H en G2 ,34H en G3 et 35H en G4 l'affichage correspondant sera :



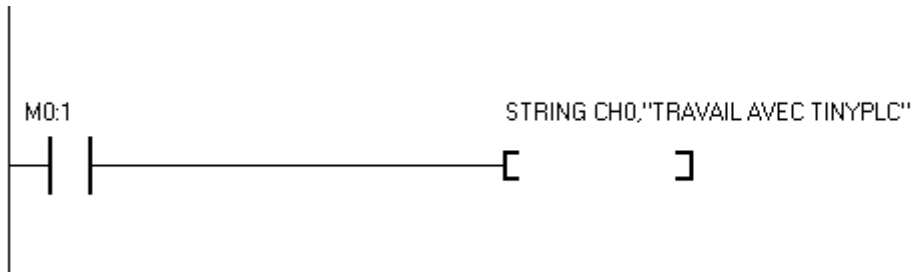
Pour transmettre une donnée au SGN , vous devez sauvegarder la donnée (en code ASCII) dans le domaine G en utilisant ASC , HEX ou STRING et la transmettre par SGNOUT

Exemple de programme :



STRING

Syntaxe : STRING d, 'chaîne' Sauvegarde les caractères de la chaîne à l'adresse spécifiée



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S	C	T	D	AD	Cnt	CH	G	
d								■			■	■	

Quand M0 :1 passe à ON la chaîne est passée en code ASCII à partir de CH0

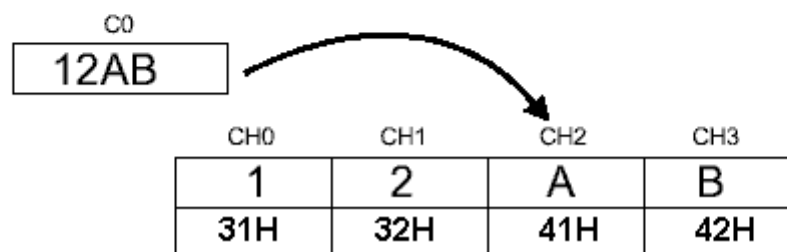
CH 0	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8	CH 9	CH 10	CH 11	CH 12	CH 13	CH 14	CH 15	CH 16	CH 17	CH 18	CH 19
T	R	A	V	A	I	L		A	V	E	C		T	I	N	Y	P	L	C
54	52	41	56	41	49	4C	20	41	56	45	43	20	54	49	4E	59	50	4C	43

HEX , DHEX

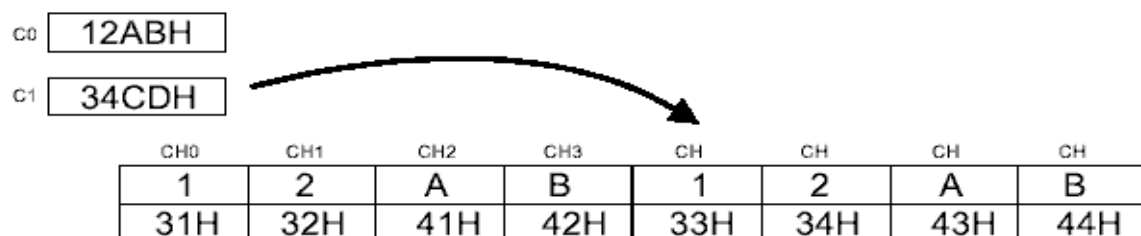
Syntaxe : HEX s,d convertit une donnée 16 bits (mot) en code ASCII hexadécimal
 DHEX s,d , convertit une donnée 32 bits en ASCII hexadécimal



Cette commande convertit en code ASCII , avec des caractères qui peuvent être affiché sur LCD ou SGN , quand M0:1 passe à ON , il stocke la valeur de C0 après conversion en ASCII à partir de CH0



Quand M0:2 passe à ON , C0,C1 sont stocké (mot long 32 bits) après conversion en ASCII hexadécimal



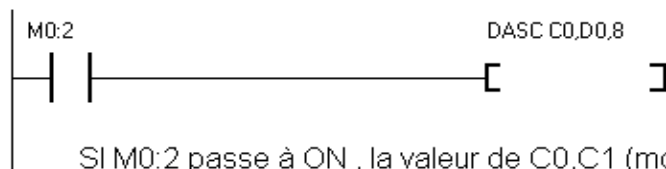
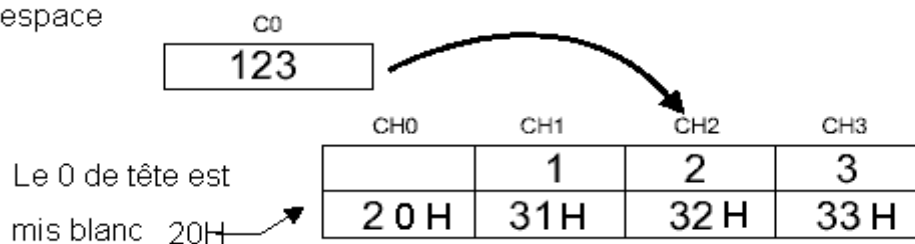
Commandes	Relais					Comp	Timer	Data	Etc ..				Constantes
	P	M	F	K	S				AD	Cnt	CH	G	
S	■	■	■	■		■		■	■	■	■	■	
d								■			■	■	

ASC , DASC

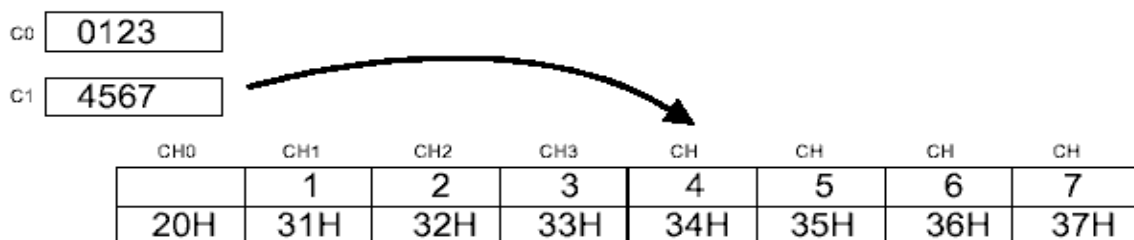
Syntaxe : ASC s,d,n convertit une donnée 16 bits (mot) en code ASCII décimal
 DASC s,d,n convertit une donnée 32 bits (mot long) en code ASCII décimal



Si M0:1 passe à ON , la valeur de C0 est stockée , après conversion en code ASCII affichable , à partir de CH0 en 4 bytes , le 0 de tête est mis en espace



SI M0:2 passe à ON , la valeur de C0,C1 (mot long 32 bits) est stockée sous forme ASCII décimale , à partir de CH0 et 8 bytes (le 0 de tête est mis en espace)



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
S	■	■	■	■		■	■	■	■	■	■	■	■
d								■			■	■	
N (nombre converti)													■

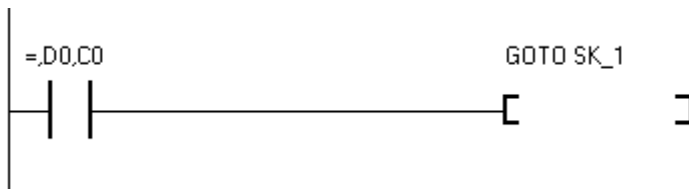
GOTO , LABEL

Syntaxe : GOTO label Instruction de transfert , saut à l'étiquette indiqué par GOTO ,
 LABEL est la commande où est déclarée l'étiquette (format de l'instruction
 LABEL label(étiquette)

Exemple de programme :

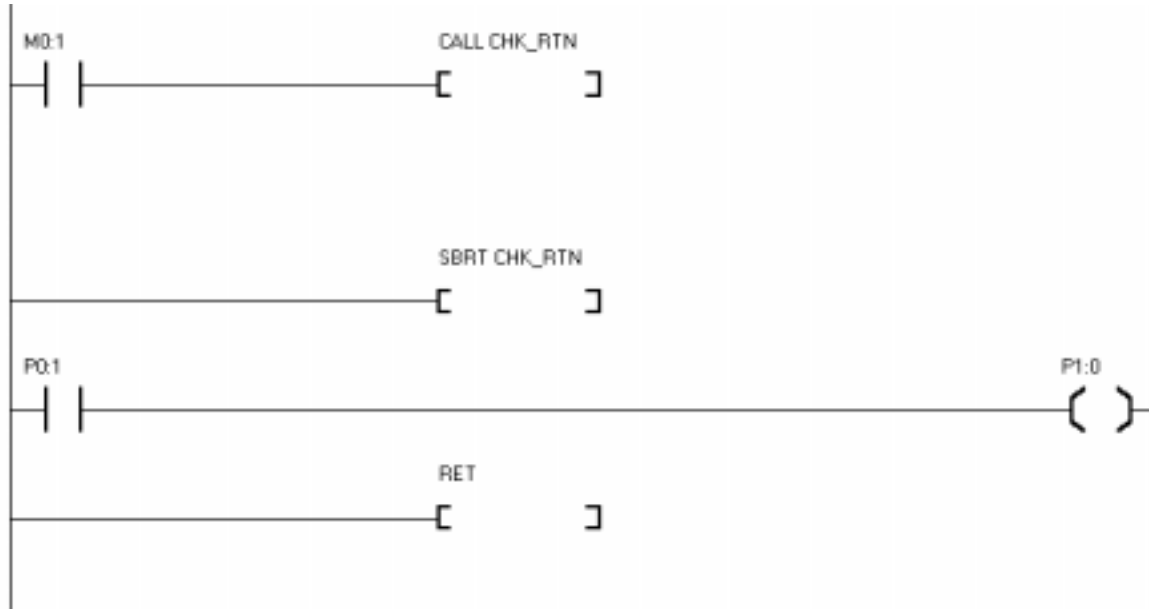


Si M0 :1 passe à ON , un saut est effectué à SK_1 , il est possible d'utiliser un branchement conditionnel (si D0=C0 , saut à SK_1)

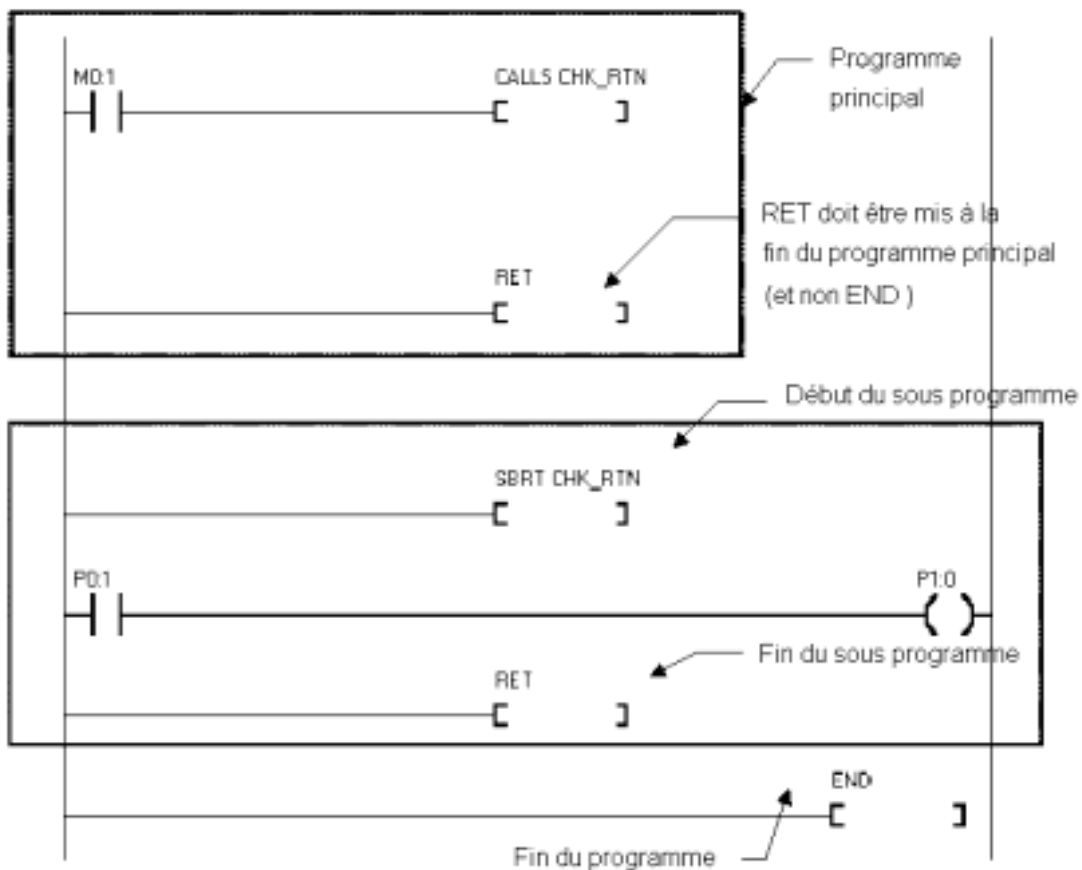


CALLS , SBRT , RET

Syntaxe : CALLS label . Cette instruction appelle un sous programme
 SBRT label Instruction de définition de sous programme
 RET doit être placé à la fin du sous programme



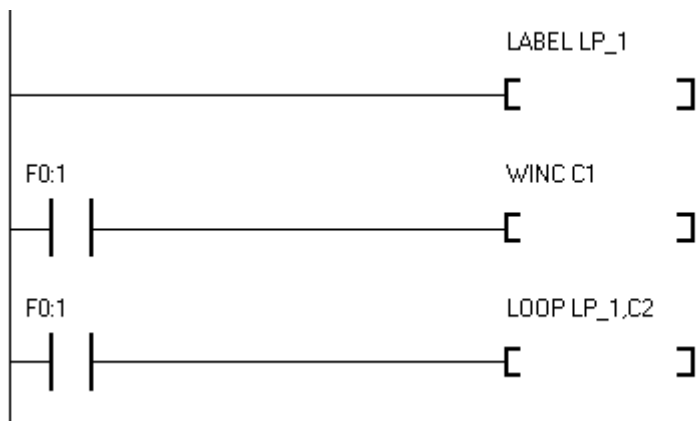
Si M0 :1 passe à ON , appel du sous programme , 16 niveaux de sous-programmes peuvent être utilisés (un sous programme peut appeler un autre sous programme sur 16 niveaux)



LOOP

Syntaxe : LOOP label,d , saut à l'étiquette jusqu'à ce que la valeur d = 0

Exemple de programme :



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S	C	T	D	AD	Cnt	CH	G	
d						■	■	■					■

La commande LOOP est exécutée inconditionnellement , F0 :1 est toujours à ON , le saut à LP_1 est fait jusqu'à ce que la valeur C2 soit à 0

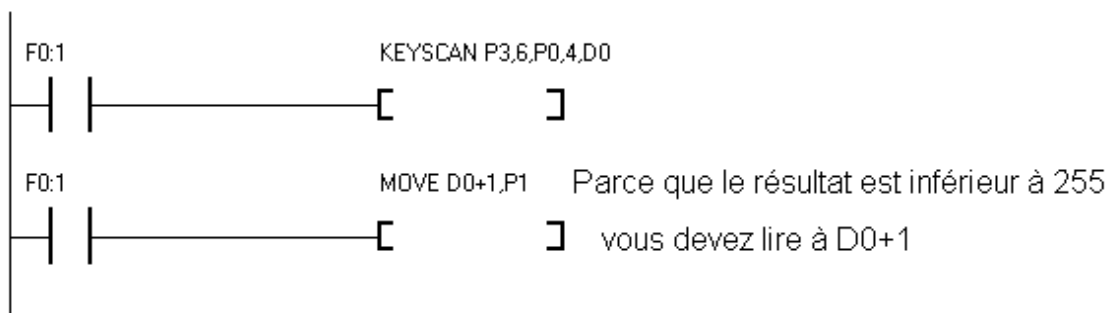


Syntaxe : KEYSCAN a,b,c,d,e

Cette instruction lit les touches de clavier par scrutation

- a : Numéro du port d'entrée par ex : P3
- b : Nombre d'entrées utilisées
- c : Numéro du port de sortie par ex :P0
- d : Nombre de sorties utilisées
- e : Endroit en mémoire où le code du mot est sauvegardé

Exemple de programme :



Parce que F0 :1 est toujours à 1 , l'instruction KEYSCAN peut fonctionner .Elle sauvegarde en permanence le code de la touche pressée d'un clavier matricé de 6x4 touches raccordé aux ports P0 et P3 . L'instruction suivante fait sortir la valeur de la touche sur le port P1, si des LED sont connectées à ce port , il est facile de voir quelle touche à été pressée .

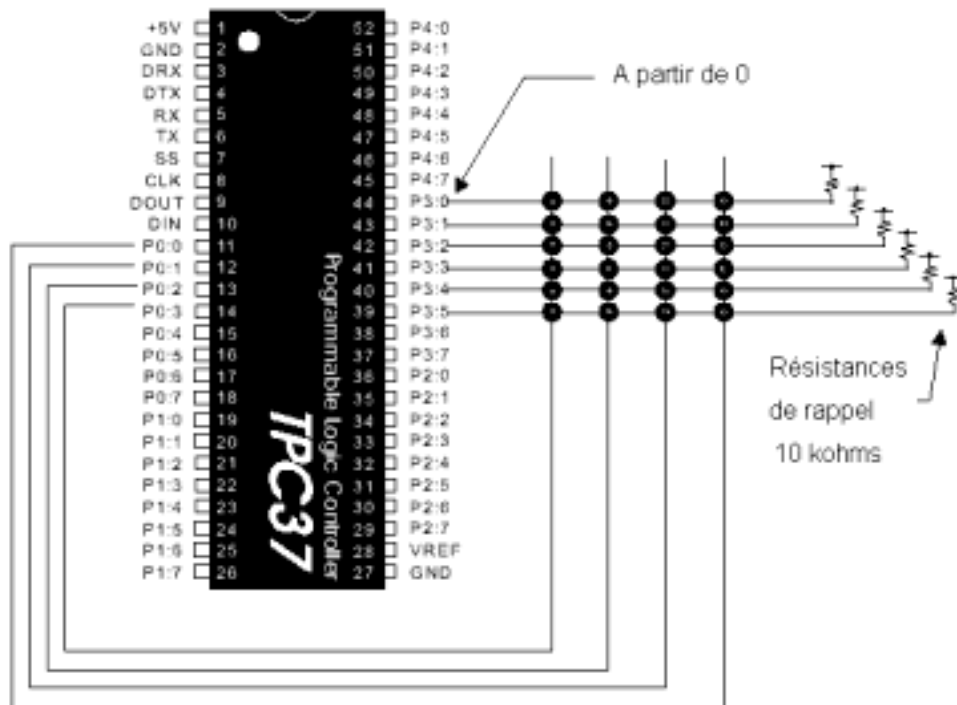
Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
d						■	■	■	■	■	■	■	

Cette instruction à été créée pour lire des claviers à membrane matricés (ce claviers sont très utilisés dans beaucoup de domaines, machines automatiques , etc) . Il faut prendre quelques précautions pour faire fonctionner cette instruction :

- 1 - Utiliser à partir de Px :0 , exemple si vous utilisez P0 , commencer à P0 :0
- 2 – Mettre des résistance de rappel connectées au port d'entrée

KEYSCAN peut lire au maximum un clavier matricé de 8x8 touches et le résultat est sauvegardé au format MOT , mais la valeur est toujours inférieure à 256 . Sans touche d'appuyée le résultat reste à 0 . Si le clavier n'utilise pas le port en entier , le reste doit être utilisé en entrée

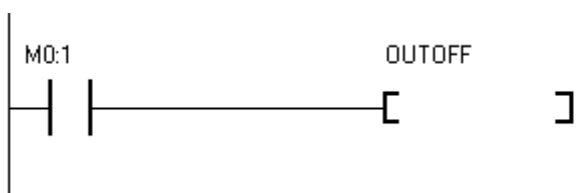
Schéma de raccordement : page suivante



Ports utilisables : TPC33 ----- P1 P2
 TPC37----- P1 P2 P3 P4
 TPC38----- P1 P2 P3 P4

OUTOFF

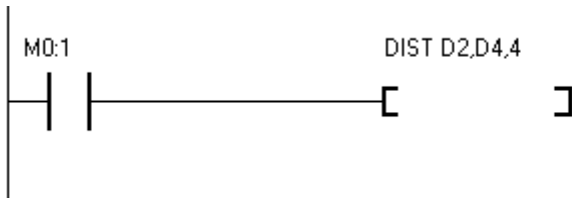
Syntaxe : Met tous les ports configurés en sortie à OFF



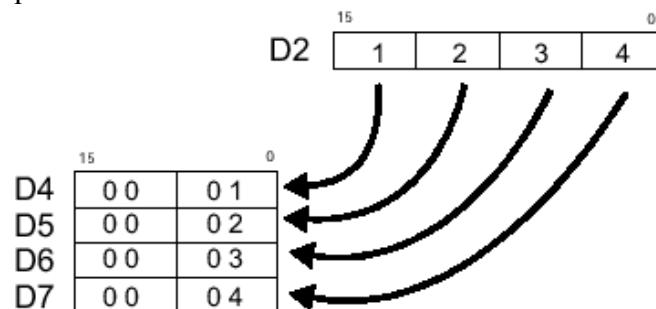
Si M0 :1 passe à ON , toutes les sorties passe à OFF

DIST

Syntaxe : DIST s,d,n , Divise un mot 16 bits en valeur n bits et le stocke en d



Si M0 :1 passe à ON , la valeur contenue dans D2 est divisée en 4 bits et stockée de D4 à D7 (n indique par quel nombre la valeur est divisée et doit être inférieur à 5)



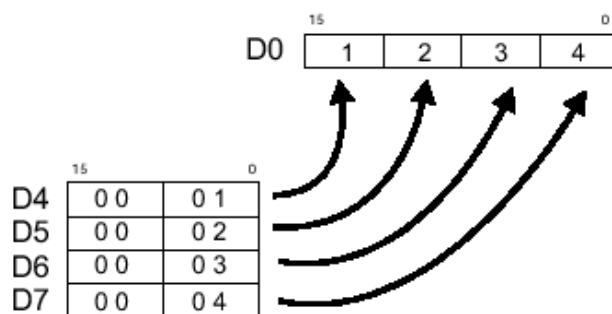
Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
s	■	■	■	■		■	■	■	■	■	■	■	■
d	■	■		■		■	■	■			■	■	
n						■	■	■					■



Syntaxe : UNIT s,d,n Combine des données du domaine D , où uniquement des données de 4 bits sont présentes et fait un mot de 16 bits (fonction inverse de DIST)



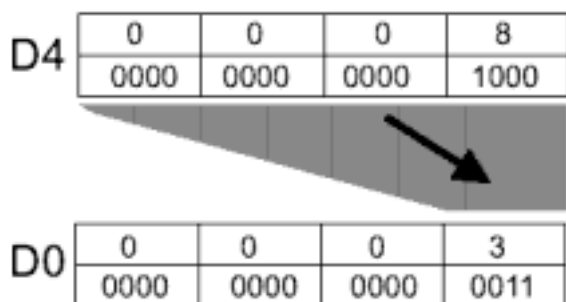
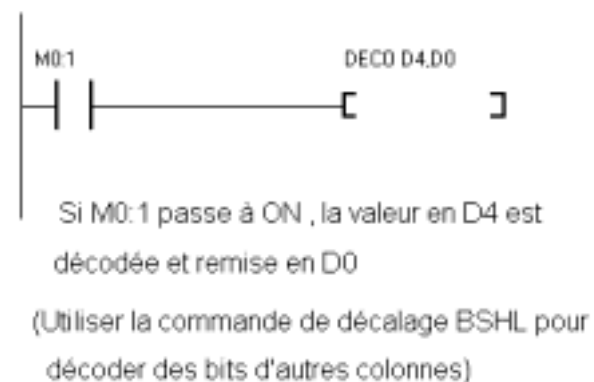
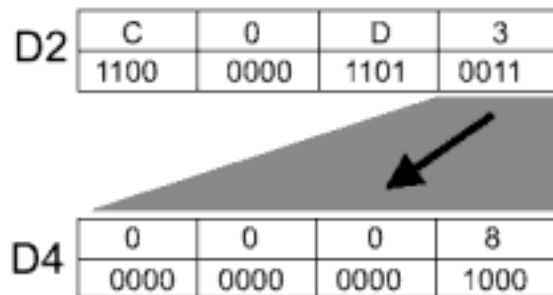
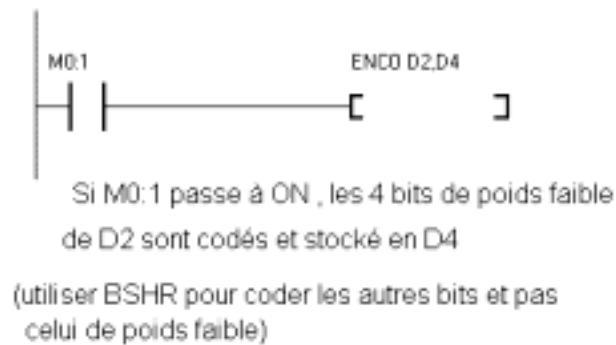
Si M0 :1 passe à ON , l'instruction UNIT va combiner les 4 bits poids faible , contenus aux adresses données D4 à D7 en un mot 16 bits si sera stocké en D0 (n désigne le nombre de groupe de 4 bits qui doivent être combiné et est inférieur à 5)



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
s	■	■		■		■	■	■			■	■	
d	■	■		■		■	■	■			■	■	
n						■	■	■					■

ENCO DECO

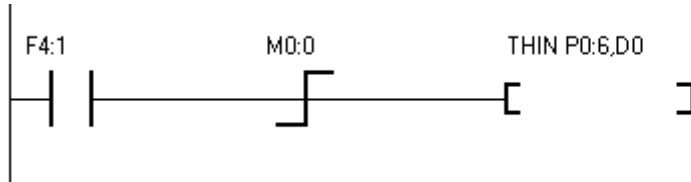
Syntaxe : ENCO s,d Code les 4 bits de poids faible de s et les déposent en d
 DECO s,d Décode les 4 bits de poids faible de s et les dépose en d



Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S				AD	Cnt	CH	G	
s	■	■		■		■	■	■	■	■	■	■	■
d	■	■		■		■	■	■			■	■	



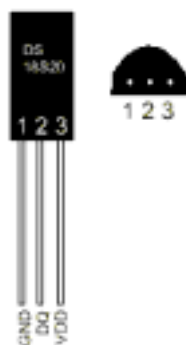
Syntaxe : THIN port,d Lit la température d'un capteur thermique DS1820 , connecté sur un port et sauvegarde la valeur en d



F4 :1 répète un créneau ON/OFF toutes les 2 secondes , le signal est mis en forme par un front montant (M0 :0) , Un cycle de lecture est donc effectué toutes les deux secondes .THIN lit la données fournie par le DS1820 connecté en P0 :6 et la stocke en D0

Commandes	Relais					Co mp	Ti mer	Da ta	Etc ..				Const antes
	P	M	F	K	S	C	T	D	AD	Cnt	CH	G	
d								■					

- Le DS1820 possède un capteur de température interne
- THIN doit être exécuté toutes les 1 ou 2 secondes (la conversion interne du DS1820 prend environ 500 ms , 750 ms pour le DS18S20, données Dallas/Maxim)
- Utilisable de -55 à +125°C (pas de 0.5 °C)
- Précision 1°C de 0 à 70°C
- Le port connecté au DS1820 doit être configuré en sortie (2 fils sont connectés au DS1820 . Sur le TPC37 ,P1 et P3 ne peuvent pas être utilisés)



Le DS 1820 est un capteur de température fabriqué par DALLAS co
Il peut lire la température sur un seul fil , deux avec la masse .

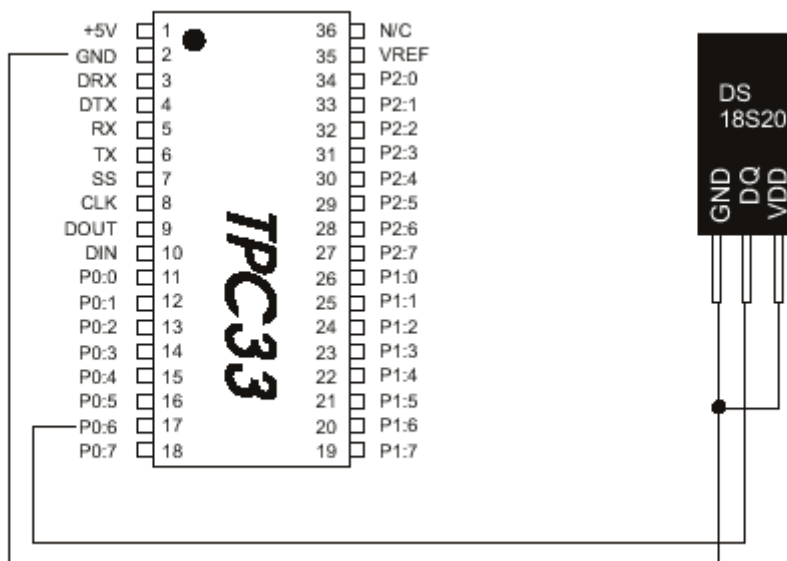
Il transmet la température mesurée sous forme binaire , le recepteur n'a pas de calcul ou transformation à faire pour l'utiliser

Température	Entrée digitale (hex)	Valeur décimale
		Négative = complément à 2
+125 °C	00FAH	250
+25 °C	0032H	50
+0.5 °C	0001H	1
0 °C	0000H	0
-0.5 °C	FFFFH	-1
-25 °C	FFC3H	-50
-55 °C	FF92H	-110

Diviser par 2 pour
avoir la température réelle

Le DS1820 en boîtier transistor TO92 et peut être installé à l'endroit de la mesure (comme un capteur PT100) et nécessite uniquement deux fils (distance maxi deux mètres) , le boîtier comporte trois fils , mais Vdd est inutilisé . La ligne DQ (qui sert de mesure) sert aussi à l'alimentation . Il est possible si l'on veut faire des mesures à différents points , d'utiliser plusieurs DS1820 simultanément . La figure suivante montre comment connecter un DS1820 sur un TinyPLC (ici le port P0 :6 est utilisé)

Le datasheet du DS1820 est disponible sur le site du fabricant à l'adresse www.dalsemi.com



Le DS 18S20 est une version plus récente du DS1820 , le temps de conversion est de 750 ms . La liaison de Vdd et GND permet une amélioration de l'immunité aux parasites .

Chapitre 5

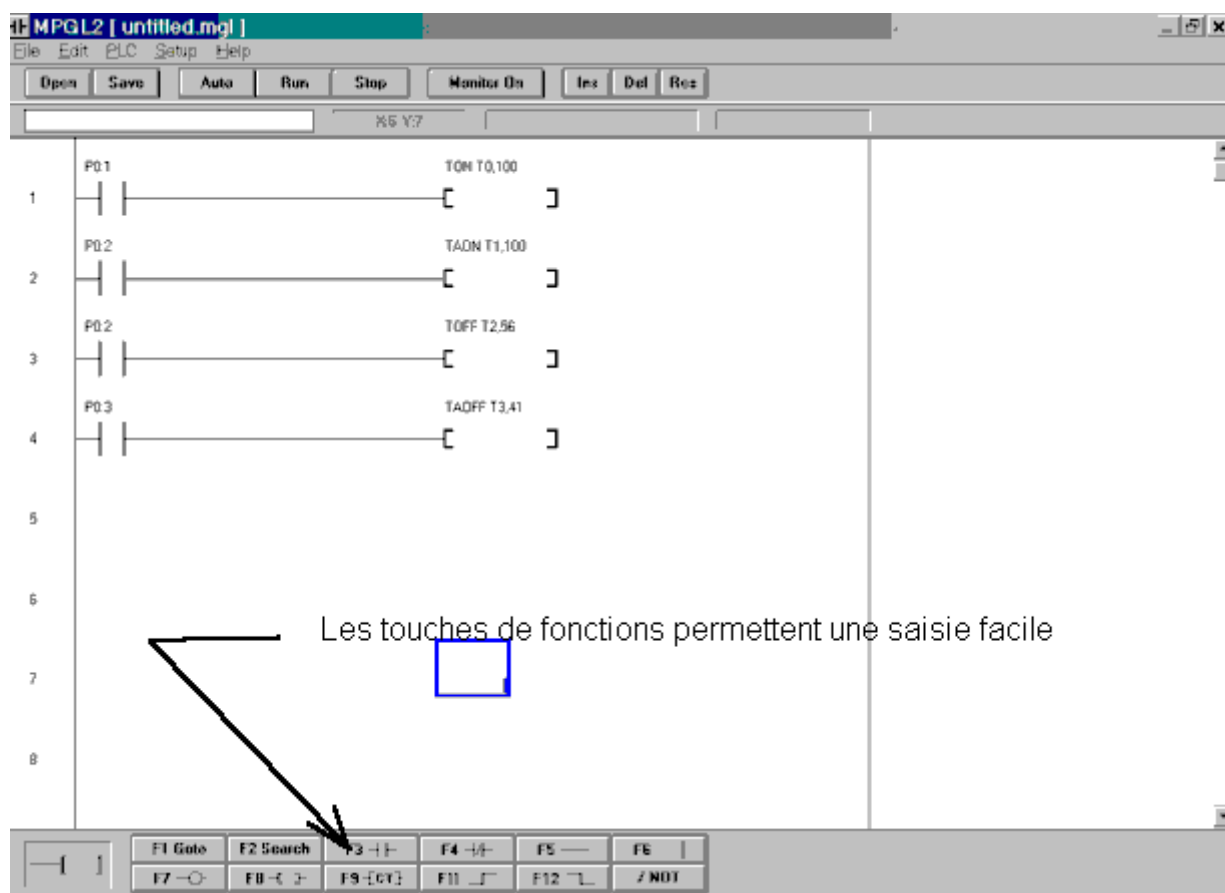
Comment utiliser

MPGL2

Présentation de MGPL2

MGPL2 est un logiciel d'application fait pour TinyPLC . Il est possible de saisir le programme LADDER , le compiler et le charger dans le PLC , il sert aussi de moniteur. Il fonctionne sous WINDOWS (3.1 , 95 et 98) . Il utilise l'interface RS232C pour dialoguer avec le PC .

L'écran de présentation est le suivant :



MGPL2 est l'abréviation de Micro PLC Loader
 MGPL2 supporte les séries TPC3X mais pas l'ancienne série TPC0X
 Ce logiciel est libre de droit , mais avec des restrictions de commercialisation

MGPL est un programme développé COMFILE technology pour les
 développements individuels

MGPL2 n'est pas utilisable sous DOS

INSTALLATION DE MGPL2 (pour Windows98™)

Pour pouvoir utiliser le logiciel , il est nécessaire de l'installer sur le disque dur de l'ordinateur . La procédure décrite ici est très légèrement différente par rapport à la notice COMFILE , qui est à partir de diskettes), en général le programme est zippé , il provient du téléchargement du site COMFILE pour avoir la dernière version , actuellement 5.8A.

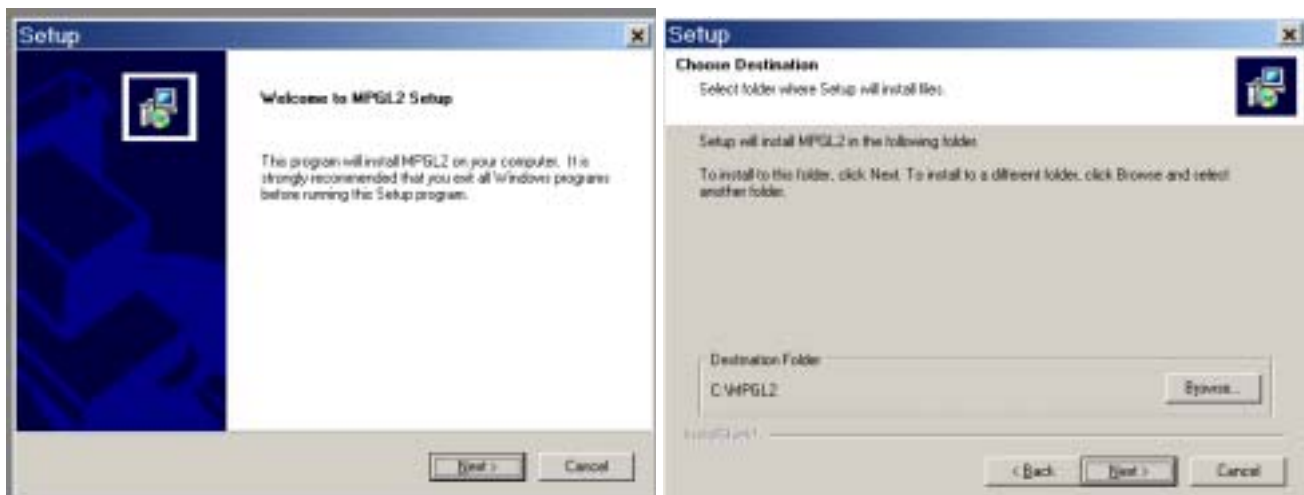
L'adresse du site est www.comfile.co.kr/ (site en langue coréenne) ou www.comfile.co.kr/english/ (site en anglais mais pas à jour)

Si vous êtes sous WindowsXP™, il vous faudra utiliser le logiciel TinyPLC Studio (vous le trouverez sur votre CD-ROM)

1. Il faut commencer par extraire le fichier ZIP dans un répertoire temporaire ,
- 2 . Se placer dans ce répertoire et exécuter le fichier SETUP

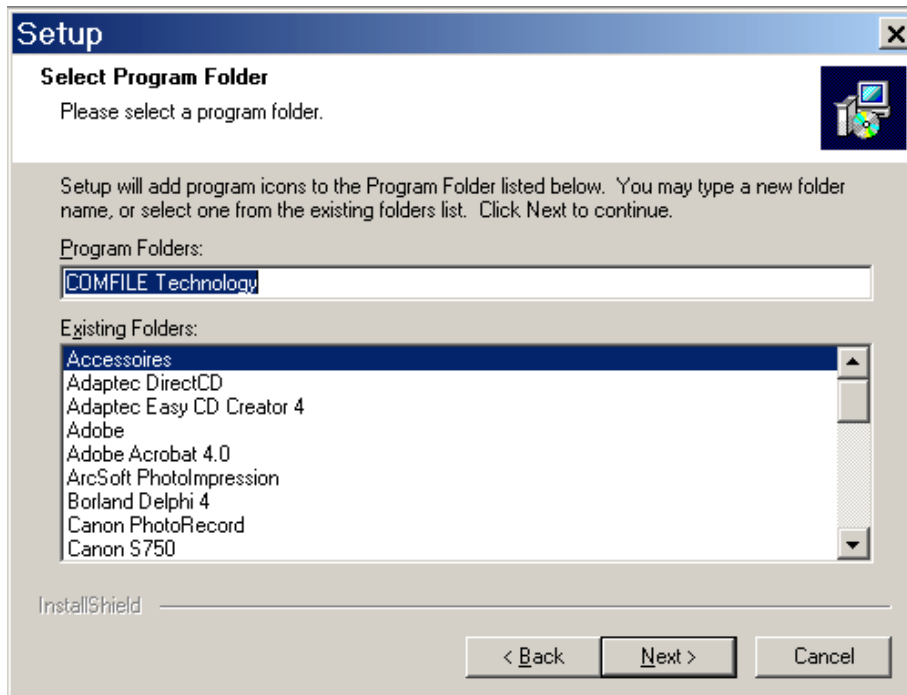


Ce qui nous conduit à l'écran suivant , il suffit de cliquer sur Next

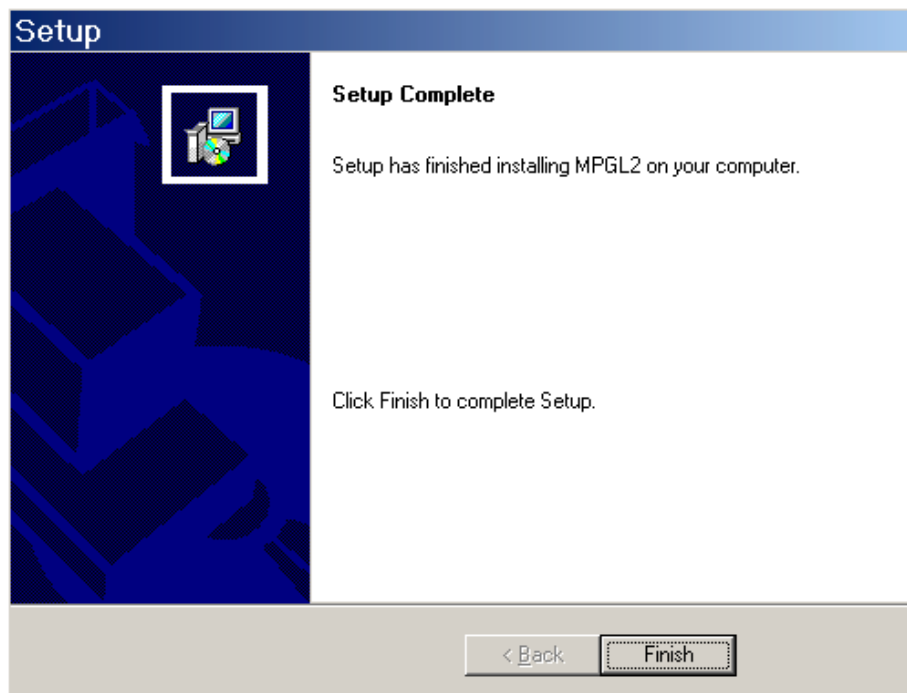


Nous pouvons ensuite choisir le répertoire d'installation , MGPL2 est proposé par défaut , si nous le gardons , il suffit de presser next

3. Choisir ensuite le groupe de démarrage ou garder celui proposé par défaut (qui peut aussi être utilisé par PICBASIC, du même fabricant).

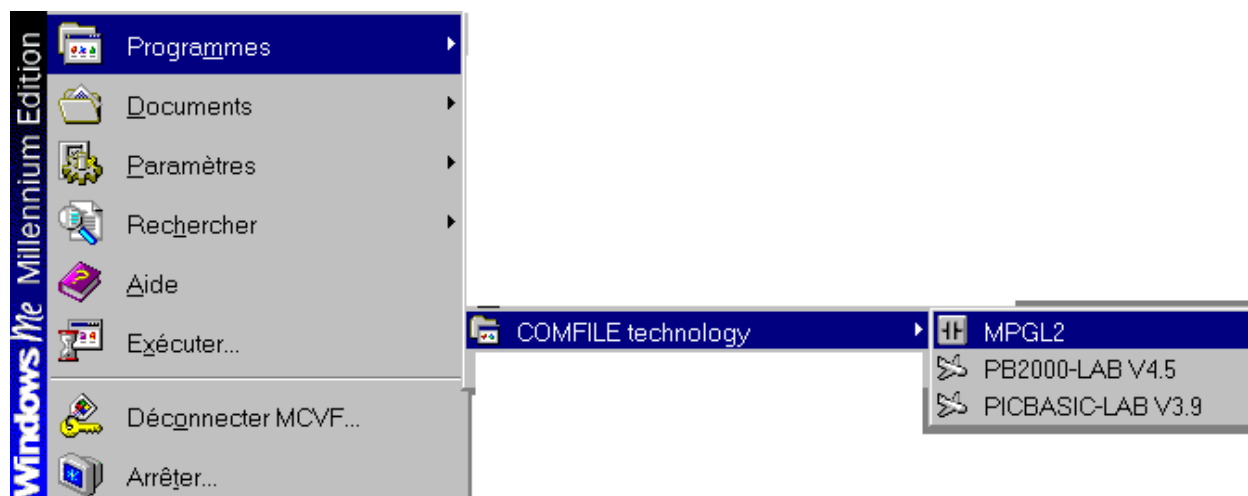


4. l'installation s'exécute rapidement puis l'on vient sur l'écran suivant



où il suffit de cliquer sur Finish pour terminer l'installation .

Pour démarrer le programme, il suffit de le choisir dans la barre des tâches .



Système minimum requis : 486 et au dessus
Windows 3.1
4 MO ram
100 MO disque dur

Il est nécessaire de connaître le port de communication qui sera utilisé pour assurer la liaison avec l'ordinateur , ou procéder par tâtonnement , en général le port 1 ou 2 est disponible (quelquefois la souris série utilise un des ports, non utilisé par une souris avec connecteur PS2)

Messages d'erreur de MPGL2

Erreur : 'END COMMAND NOT FOUND'

Dans ce cas la commande END n'est probablement pas dans le programme source

Erreur : 'COMPILE ERROR'

Erreur de syntaxe LADDER , pas en relation avec la communication , à chercher dans le programme LADDER

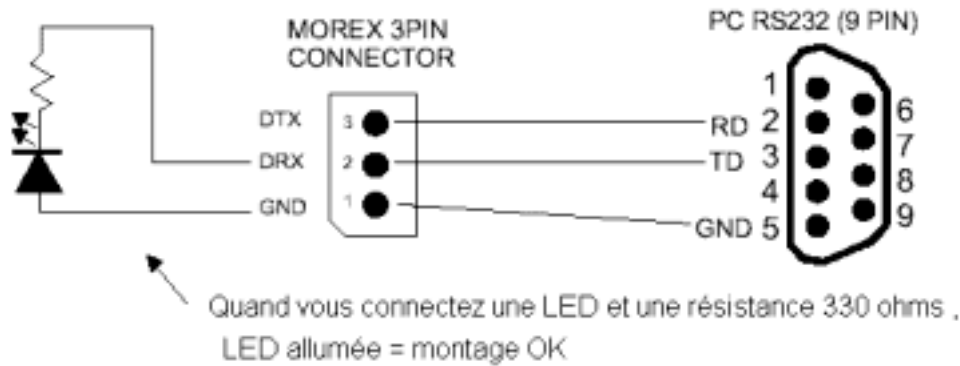
Erreur : 'Invalid RS232C port error or device unavailable error'

Problème de communication avec le PC , essayer les autres ports de l'ordinateur . Sinon en cas de problème grave s'adresser à un professionnel

Erreur : 'TinyPLC module not found'

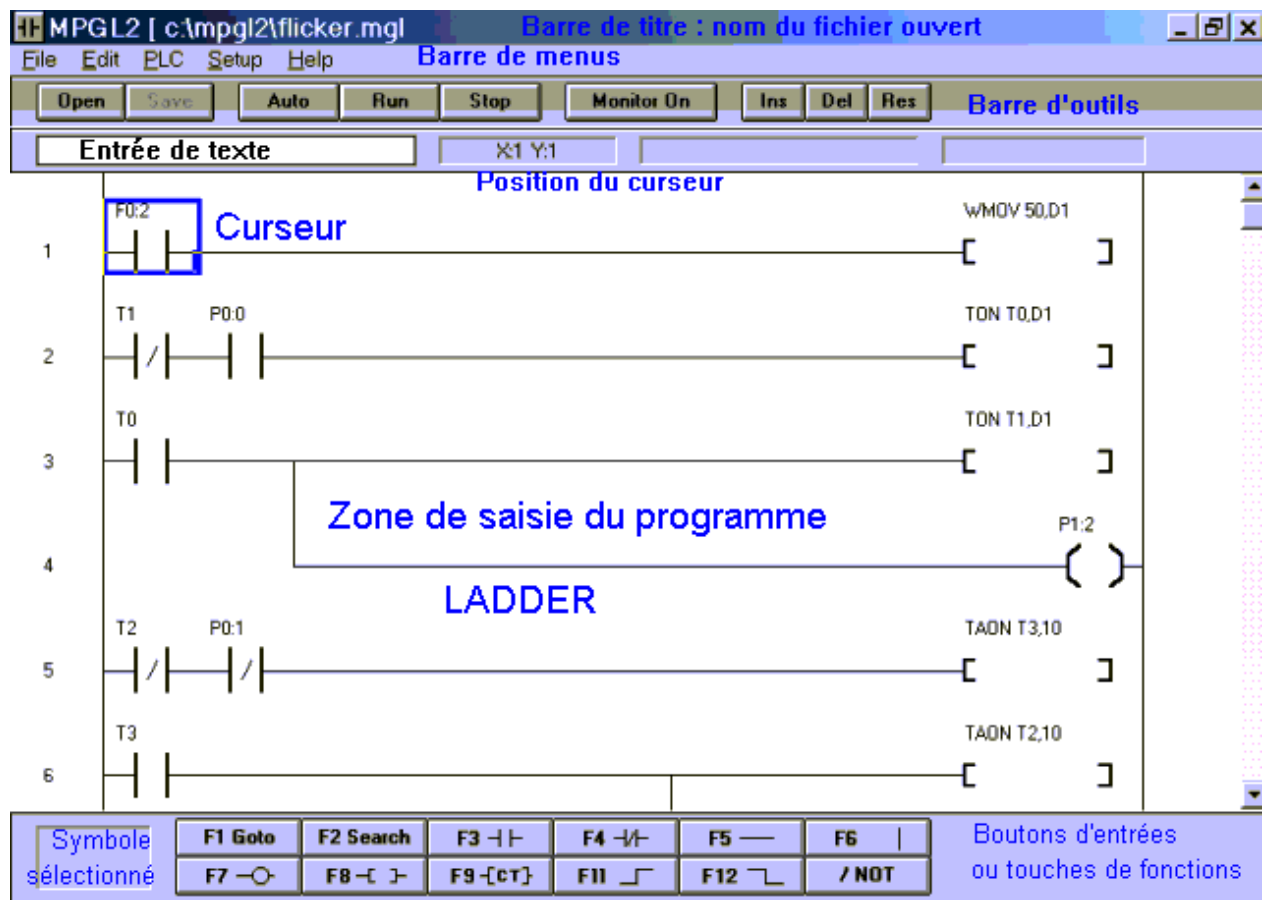
Dans ce cas TinyPLC n'est pas reconnu . Vérifier le câble l'alimentation , vérifier les contacts

Ci-dessous le dessin du câble fabriqué par COMFILE . Si vous faites votre câble vous même , vérifiez que le câblage est bien correspondant .





La méthode finale pour vérifier et d'essayer sur un autre PC , si la communication se fait sans erreurs , faire vérifier par un professionnel le fonctionnement et le paramétrage des ports série .Si le défaut est persistant contacter COMFILE pour faire remplacer le câble .

Fenêtre de MGPL2



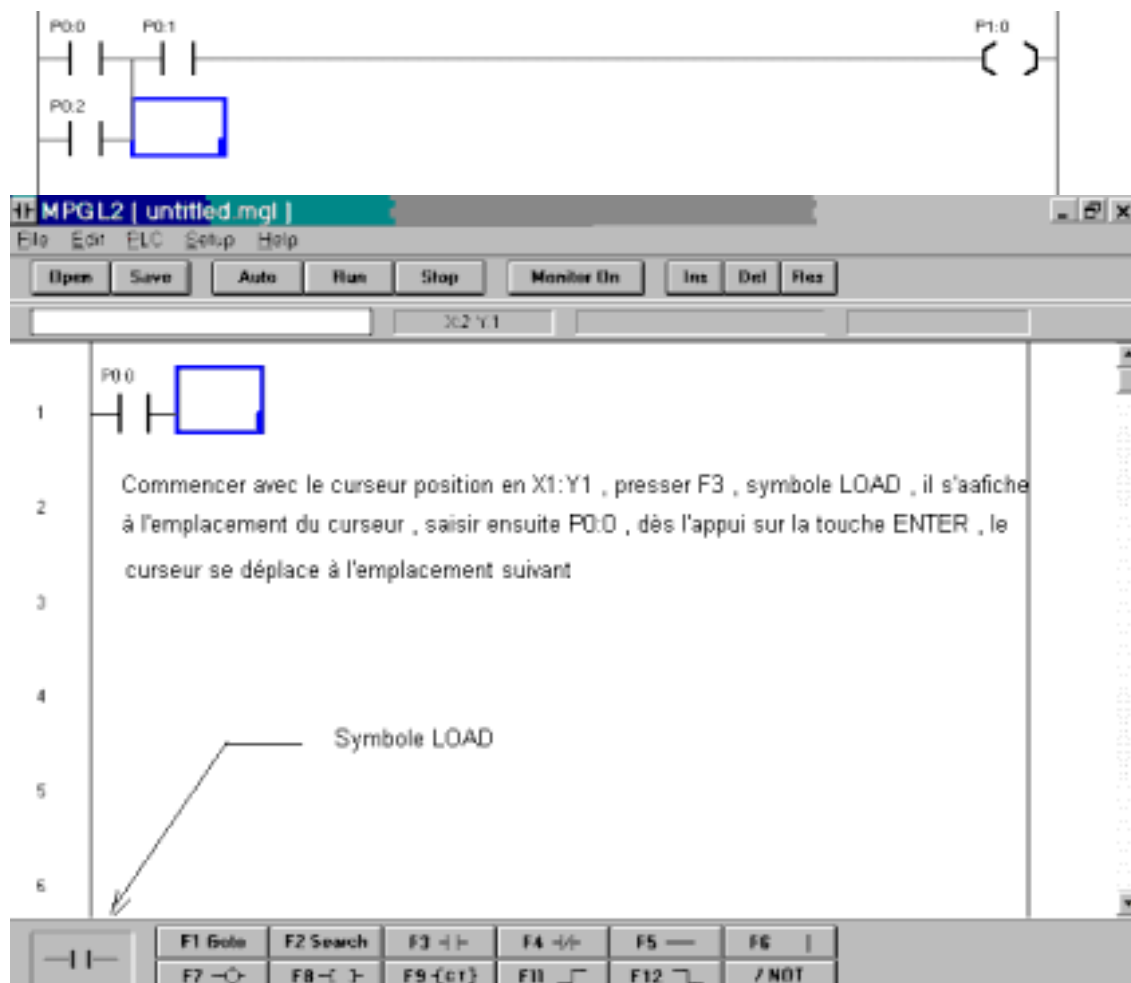
Entrée des symboles LADDER

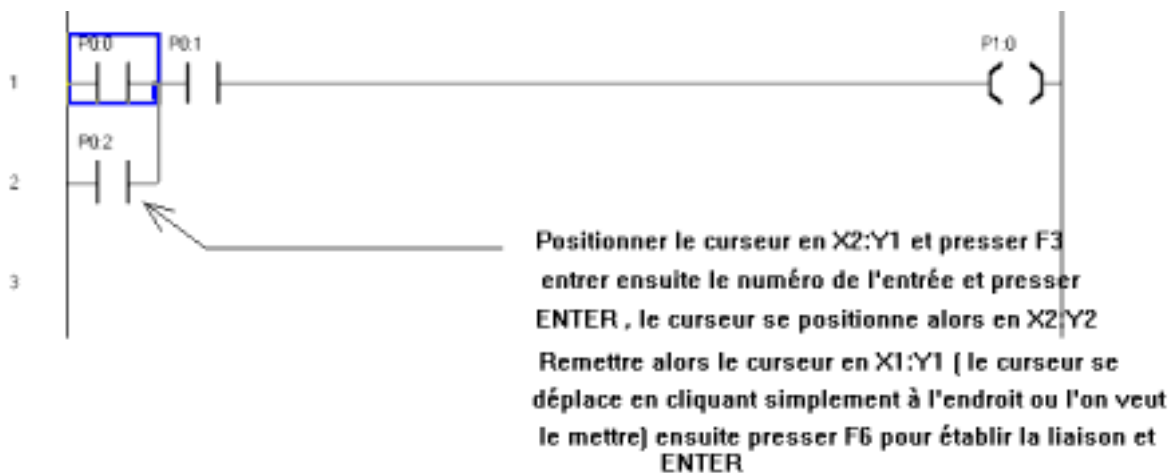
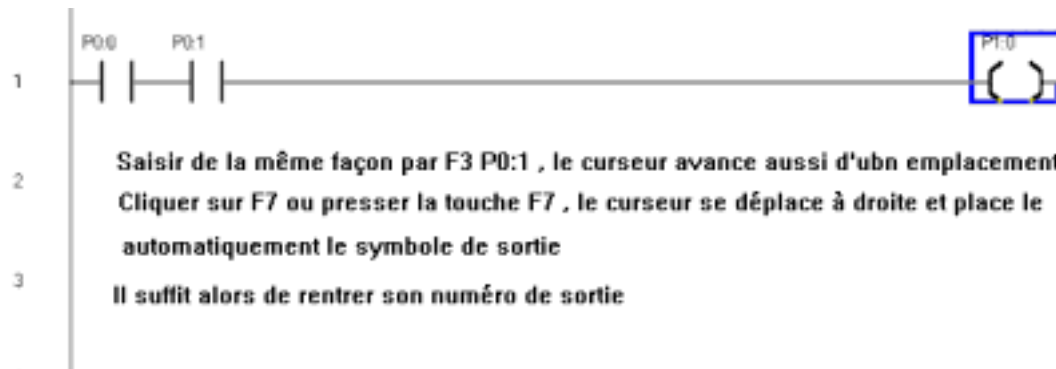
Touche	Explication
F1 Goto	Sélectionne le n° de ligne pour commencer la saisie
F2 Search	Recherche du texte sur une ligne , nom d'un relais ou commande
F3 	Entrée du symbole pour commencer une fonction
F4 	Entrée du symbole LOADN
F5 	Traçage d'un ligne (peut apparaitre ou disparaître utilise pour la fonction AND
F6 	Connexion pour les fonction OR
F7 	Symbole de résultat
F8 	Utilisé pour les symbols d'application et les temps

F9 [CT]	Symbole utilisé pour les compteurs
F11 	Symbole utilisé pour front montant (transition)
F12 	Front descendant (transition)
/ NOT	Symbole d'inversion de la commande
ESPACE	Efface le texte et le symbole au point du curseur
CTRL + I	Insère une ligne à l'emplacement du curseur , INS de la barre d'outils fait la meme fonction
CTRL + D	Supprime la ligne à l'emplacement du curseur , DEL de la barre d'outils fait la meme fonction
CTRL + U	Restaure une ligne supprimée , idem RES de la barre d'outils
ENTER	Sert à éditer le texte du symbole à l'emplacement du curseur , le texte apparait alors dans la fenêtre d'edition ou il peut être modifié , presser ENTER pour valider la ligne

EXEMPLE DE SAISIE

Exemple de saisie du LADDER expliqué pas à pas en copies d'écran





MPGL2 [untitled.mgl]

File Edit PLC Setup Help

Open Save Auto Run Stop Monitor On Ins Del Res

X1:Y3

1

2

3

4

5

6

7

8

Mettre le curseur en début de ligne 3 et appuyer
quelques fois pour mettre le curseur en X10 et presser F8
et entrer END dans la zone d'édition
(cliquer sur SHIFT F8 positionne directement le curseur à
droite)
Finalement sauvegardez
le programme en lui
donnant un nom

Save As

Nom de fichier :
essai.mgl
flicker.mgl
hatched.mgl

Dossiers :
c:\mpgl2

OK
Annuler
Réseau...

Lecture seule

Type :
MPGL File

Lecteurs :
c:

F1 Goto F2 Search F3 F4
F7 F8 F9 F10

Initialisation du PLC et des entrées et sorties

La direction des ports d'entrées et de sorties doivent être configurées par l'utilisateur

Attention certain modèle de TPC ont certains ports fixes en entrée et d'autres sont fixes en sorties , l'écran suivant correspond au TPC 37 , le port 1 est fixe en sortie le port 3 est exclusivement en entrée

Lorsque le signe X apparaît dans une case , il indique que le port est en entrée , les cases non cochées sont utilisées en sorties .

PLC setup

Sélection du type de PLC

PLC: **TPC37**

Data Communication

☐ Un Use

☒ Binary Mode

☐ RS232 ASCII Mode

☐ RS485 ASCII Mode

RS485 Node Address: 0~255

I/O Port Direction

Please check INPUT port only!

Port0	Port1	Port2	Port3	Port4
<input checked="" type="checkbox"/> P0:0	<input type="checkbox"/> P1:0	<input checked="" type="checkbox"/> P2:0	<input checked="" type="checkbox"/> P3:0	<input checked="" type="checkbox"/> P4:0
<input checked="" type="checkbox"/> P0:1	<input type="checkbox"/> P1:1	<input checked="" type="checkbox"/> P2:1	<input checked="" type="checkbox"/> P3:1	<input checked="" type="checkbox"/> P4:1
<input checked="" type="checkbox"/> P0:2	<input type="checkbox"/> P1:2	<input checked="" type="checkbox"/> P2:2	<input checked="" type="checkbox"/> P3:2	<input checked="" type="checkbox"/> P4:2
<input checked="" type="checkbox"/> P0:3	<input type="checkbox"/> P1:3	<input checked="" type="checkbox"/> P2:3	<input checked="" type="checkbox"/> P3:3	<input checked="" type="checkbox"/> P4:3
<input checked="" type="checkbox"/> P0:4	<input type="checkbox"/> P1:4	<input checked="" type="checkbox"/> P2:4	<input checked="" type="checkbox"/> P3:4	<input checked="" type="checkbox"/> P4:4
<input checked="" type="checkbox"/> P0:5	<input type="checkbox"/> P1:5	<input checked="" type="checkbox"/> P2:5	<input checked="" type="checkbox"/> P3:5	<input checked="" type="checkbox"/> P4:5
<input checked="" type="checkbox"/> P0:6	<input type="checkbox"/> P1:6	<input checked="" type="checkbox"/> P2:6	<input checked="" type="checkbox"/> P3:6	<input checked="" type="checkbox"/> P4:6
<input checked="" type="checkbox"/> P0:7	<input type="checkbox"/> P1:7	<input checked="" type="checkbox"/> P2:7	<input checked="" type="checkbox"/> P3:7	<input checked="" type="checkbox"/> P4:7

Sens des entrées et sorties - X = entrées

Option

☒ Source Upload (No check is Protect Mode)

☐ Auto Clear Keep Area when Download

☒ 2.5mS Scan Time **Si non coche = 5 ms**

Adresse de la liaison en RS485

Ok **Cancel**

Pour la sélection du type de PLC faire le choix dans la boîte de saisie .

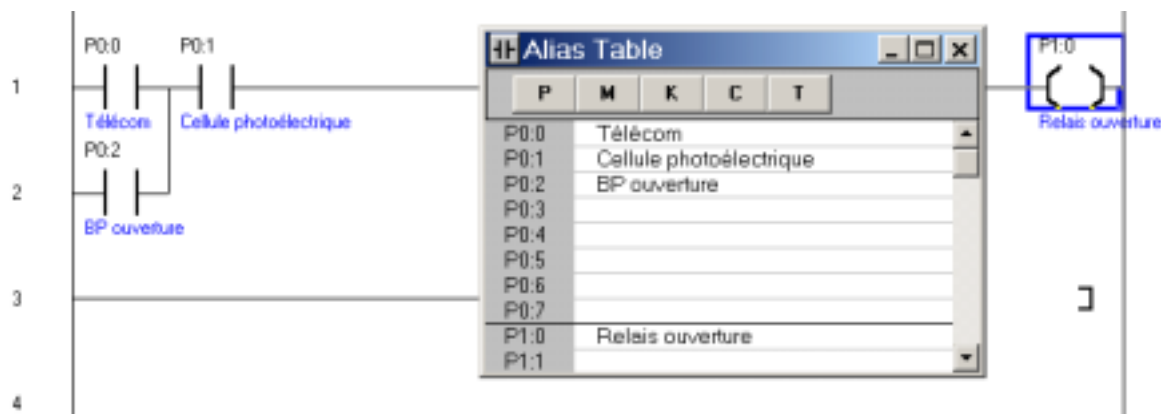
AutoClear permet de réinitialiser les valeurs sauvegardées lors du téléchargement

Source UPLOAD permet, s'il est coché, de télécharger un programme à partir d'un PLC déjà programmé .

Utilisation possible de RS232C (cas le plus courant par défaut ou RS 485, avec possibilité de spécifier de l'adresse)

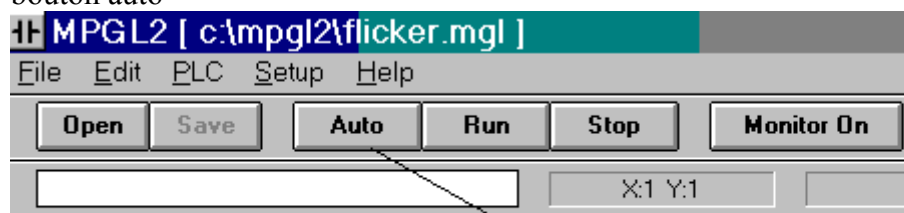
Entrée de commentaire pour les symboles

Pour faciliter la saisie et documenter le programme (très utile lorsque l'on reprend le programme plus tard), il est possible de saisir l'utilisation des différents symboles dans 'ALIAS TABLE' :



Compilation Téléchargement

Après avoir saisi le programme LADDER, défini le sens des ports entrées et sorties et sélectionné le type de PLC .Il faut le compiler et le télécharger dans le PLC . Cliquer sur le bouton auto



Compile et télécharge le programme

Si l'erreur TinyPLC NOT Found apparaît, vérifier l'alimentation et le câble comme expliqué avant

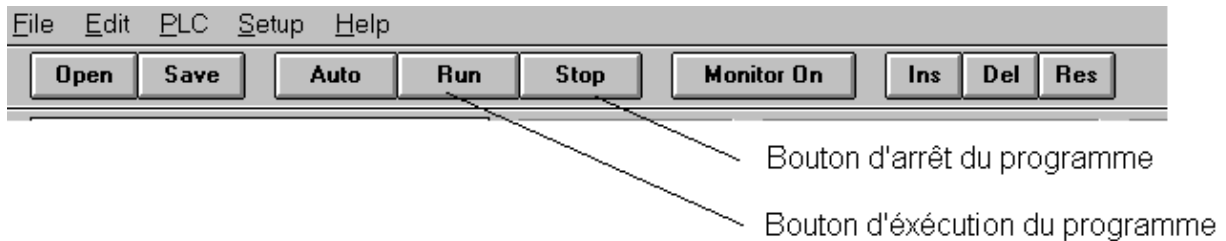
Pendant le téléchargement si l'erreur Writing Logic Fail apparaît . Arrêter MGPL2 et rebooter, couper l'alimentation (Une alimentation stable et un environnement pas trop perturbé sont nécessaire pendant le téléchargement)

Deux autres types d'erreur peuvent survenir pendant la compilation, les erreurs de « grammaire » et les erreurs de compilations, ces erreurs sont relativement faciles à trouver, elles sont affichées en rouge .

Pendant la compilation, une fenêtre DOS apparaît, si elle n'est pas fermée automatiquement, il est nécessaire de la fermer manuellement (il est possible de cocher la case fermeture automatique en fin de session)

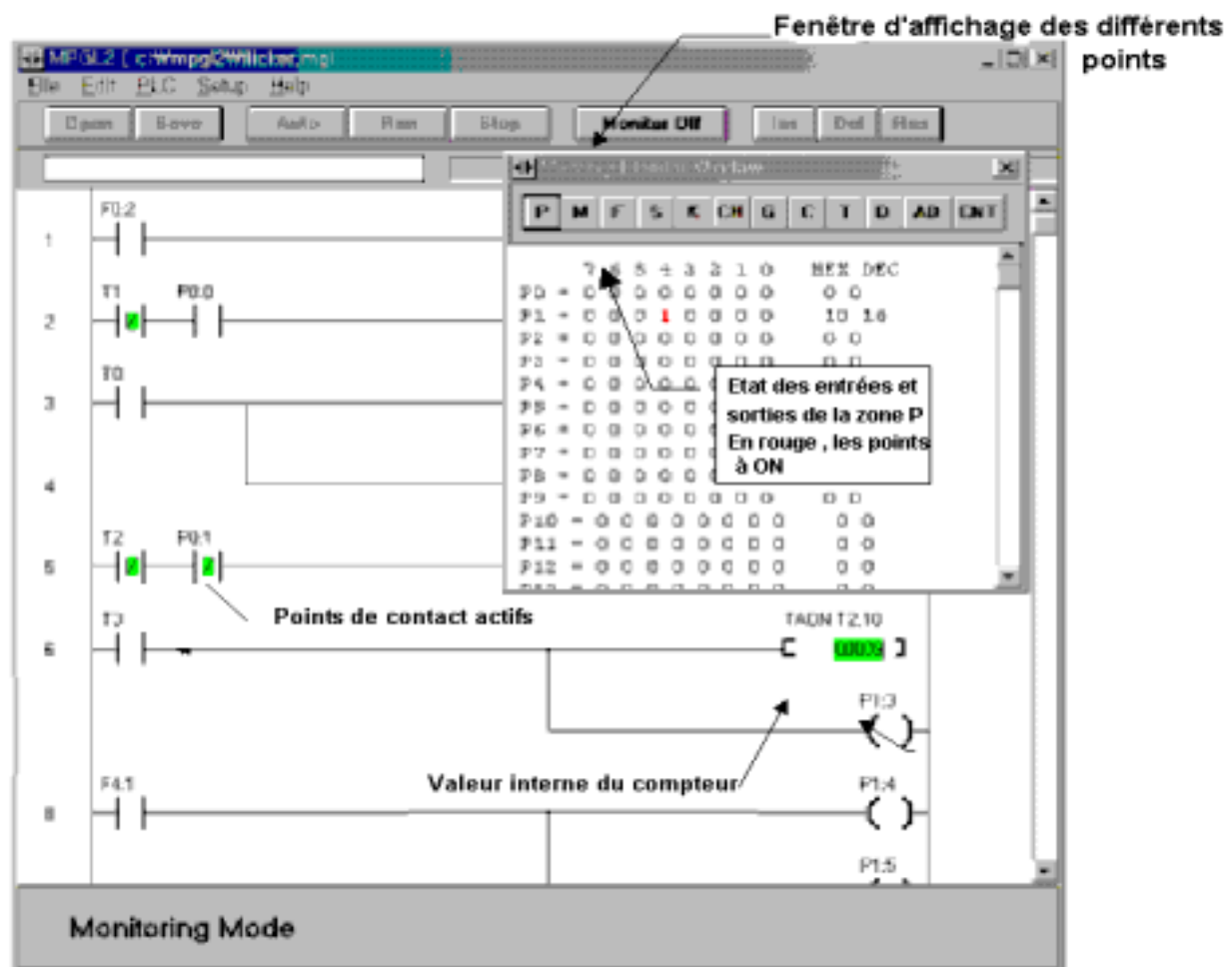
Exécution , état d'attente

Après téléchargement, TinyPLC reste inactif et attend une autre commande , soit une coupure d'alimentation pour un fonctionnement normal ou RUN pour un fonctionnement de mise au point.



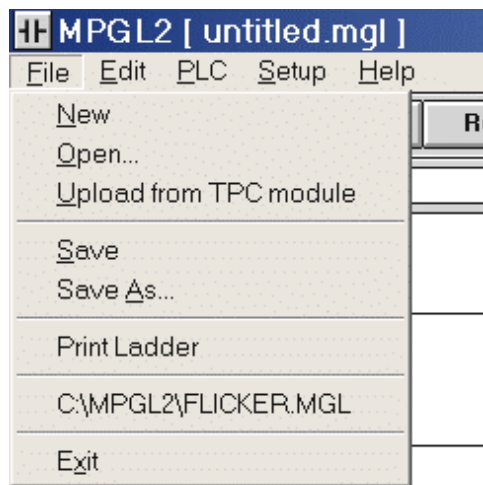
Moniteur mise au point

Pour faire la mise au point du programme , démarrer par le bouton Monitor , le suivi d'exécution . TinyPLC et le PC doivent être connectés ensemble par le câble . Dans cet état chaque entrée, sortie, compteur etc sont affichée en temps réel



Nota : L'état moniteur actif peut affecter le fonctionnement des autres programmes . Terminer les autres programmes si possible . Le moniteur n'est actif que connecté au PC.

Chargement d'un programme source



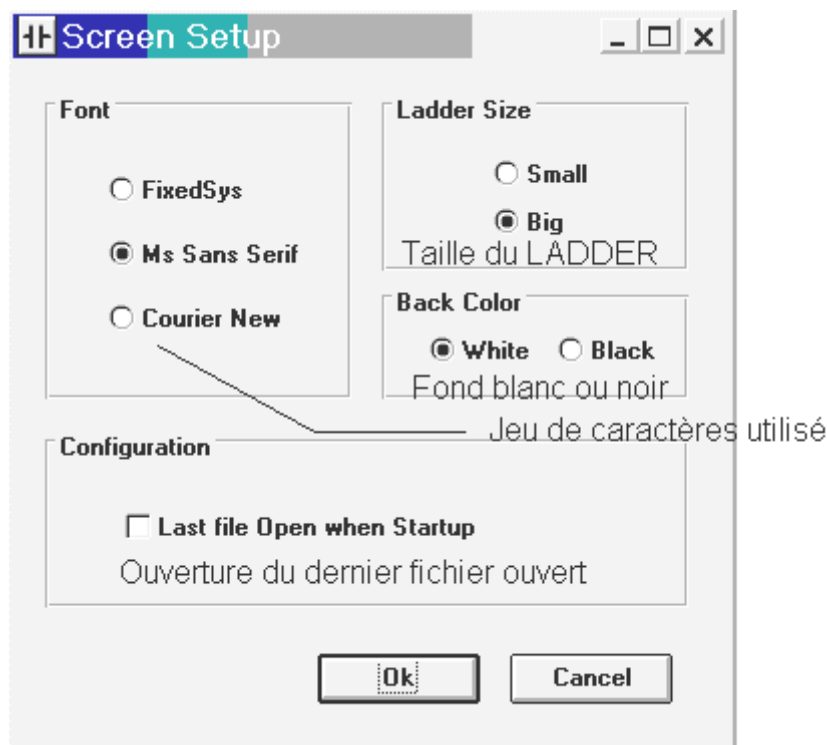
Sélectionner Upload from TPC module

Le programme source sauvegardé dans le module TPC est rechargé dans la mémoire du PC et apparaît dans la fenêtre normale d'édition .

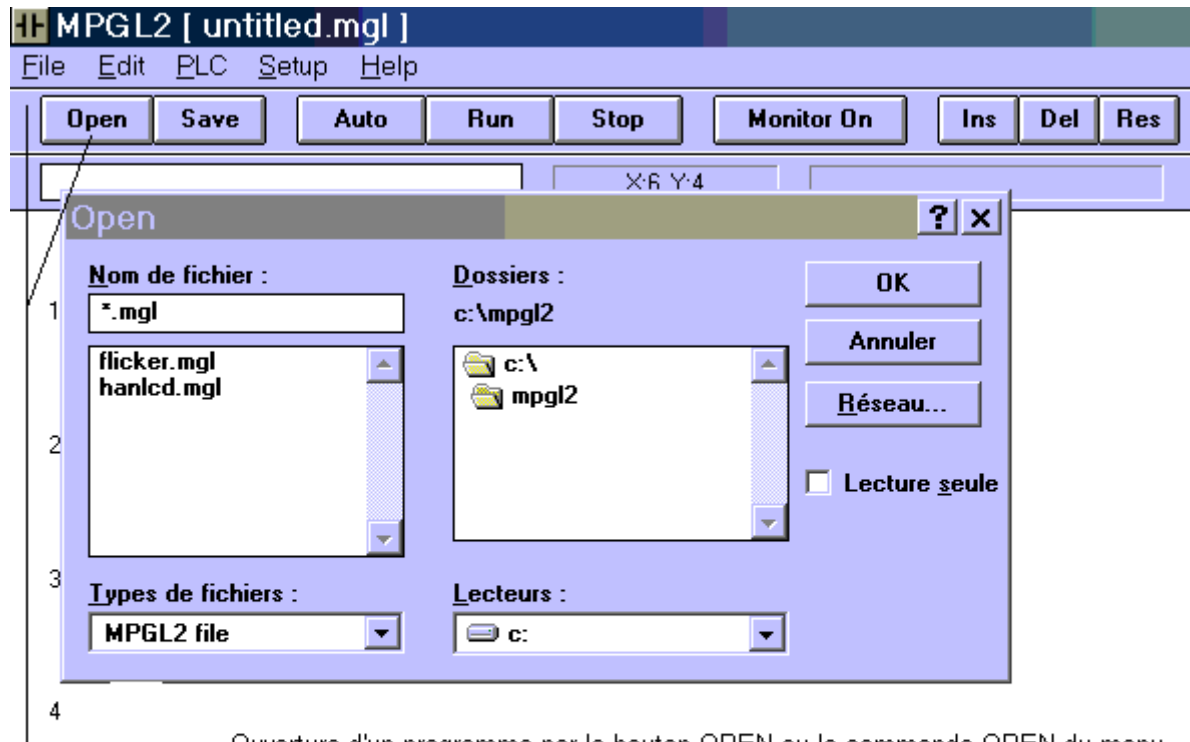
Cette fonction peut ne pas être possible , si une option a été validée dans le menu SETUP PLC

Si l'option a été validée, le programme ne peut être lu par une tierce personne, le programme ne peut être modifié
Le programme est automatiquement décompressé lors du chargement .

Environnement de MGPL2



Ouverture et sauvegarde programme



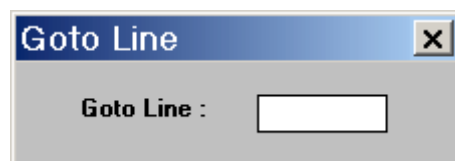
Ouverture d'un programme par le bouton OPEN ou la commande OPEN du menu fichier (les fichiers ont l'extension .mgl)

La sauvegarde est effectuée par la commande SAVE du menu fichier ou le bouton SAVE , il faut donner un nom au fichier sauvegardé

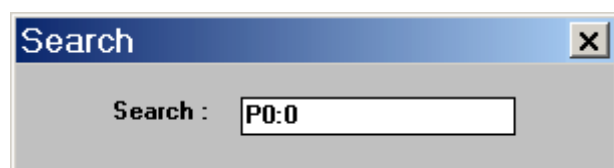
Lorsque le programme a été modifié , MGPL2 demande si l'on veut effectuer une sauvegarde

Passage à une ligne Recherche de symbole

Cliquer sur F1 pour aller directement sur une ligne et entrer son numéro dans la boîte qui s'affiche



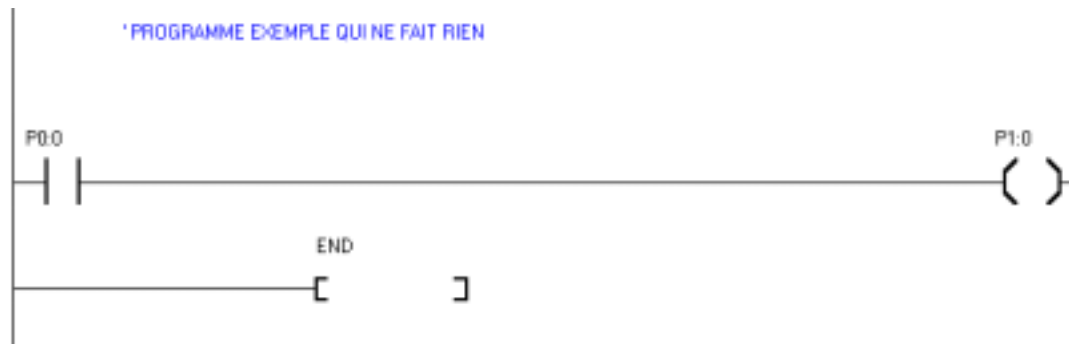
F2 permet l'affichage d'un écran popup entrer le texte à retrouver , et le programme se déplace vers l'endroit demandé



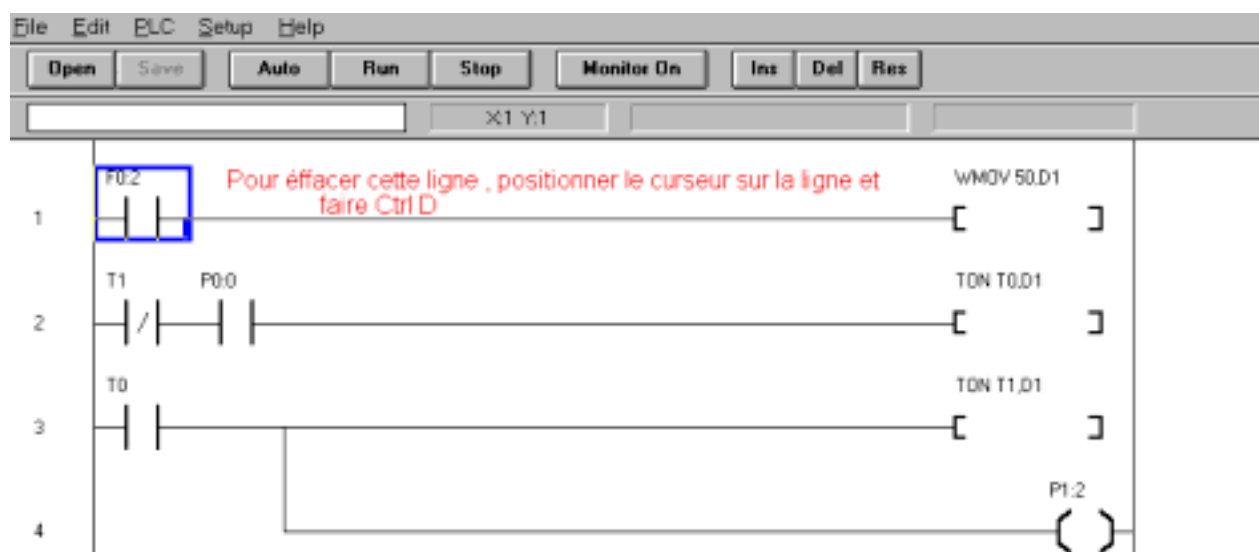
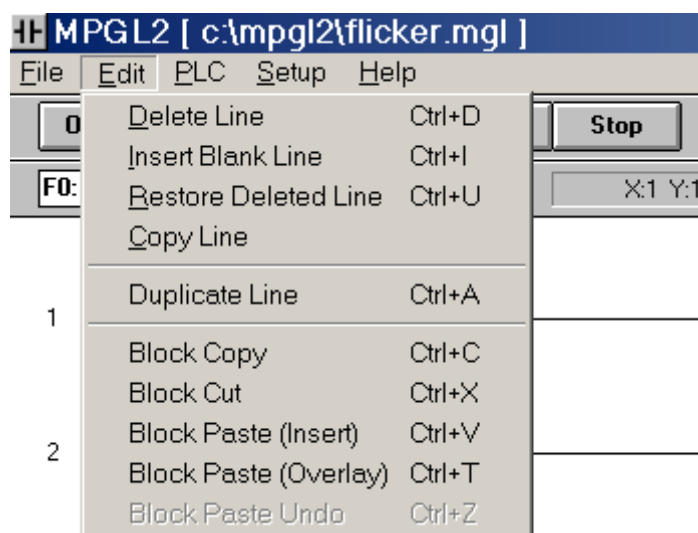
Saisie de commentaires

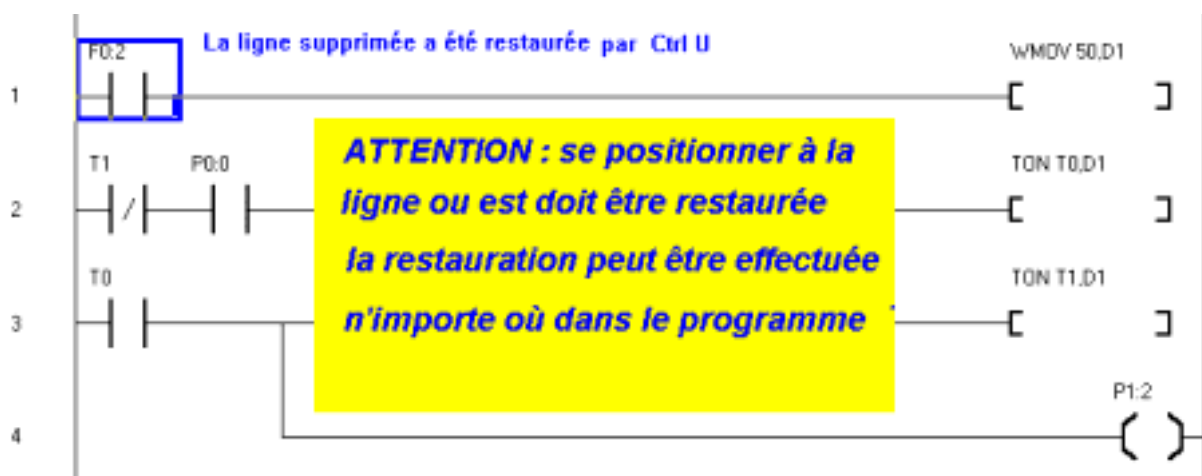
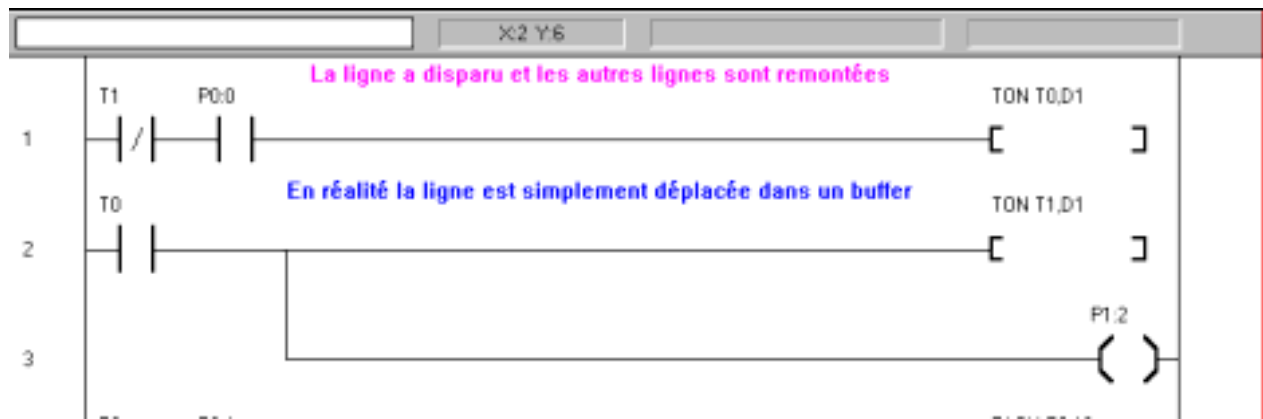
Des explications du programme peuvent être écrites dans le programme pour le documenter, très utile lorsque l'on veut le modifier plus tard.

Le commentaire commence par l'apostrophe, le texte s'affiche en bleu dans le programme, mais le compilateur l'ignore.

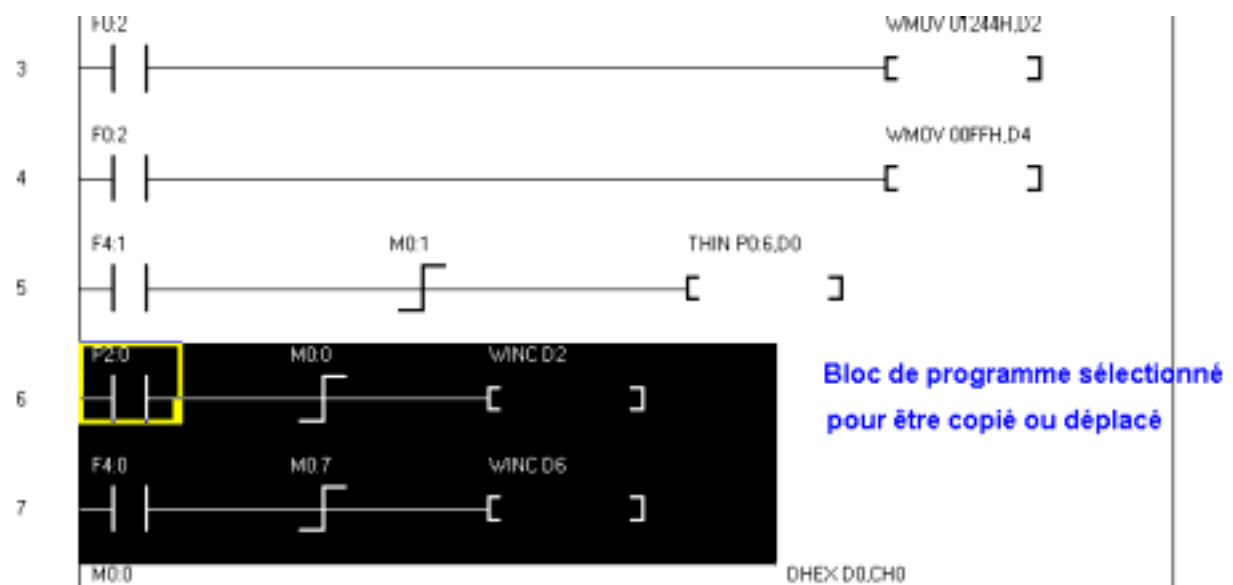


Remplacer, copier, déplacer des symboles LADDER





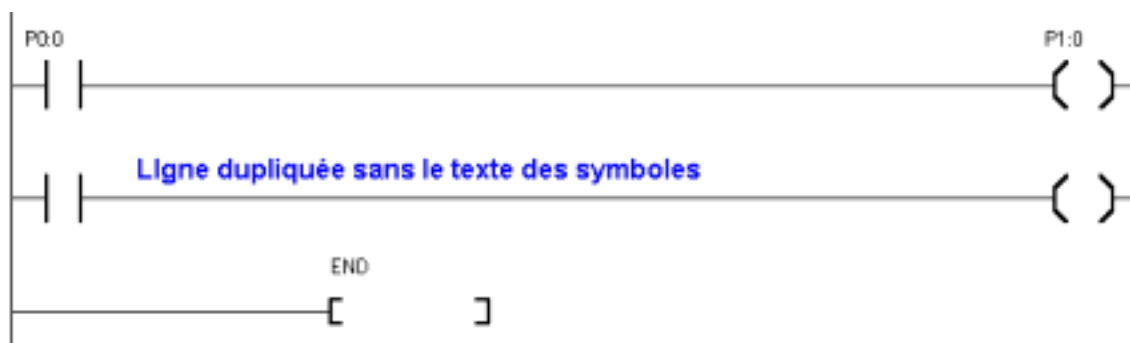
Il est possible d'effacer plusieurs lignes consécutives en pressant plusieurs fois de suite **Ctrl D**, (après avoir positionné le curseur sur les ligne à effacer). Ces lignes ne sont pas effacées de façon définitive mais sont sauvegardées dans un buffer.. les lignes peuvent être restaurées.



Pour copier le bloc vers un autre endroit , faire **Ctrl C** et ensuite cliquer à l'endroit où l'on veut insérer ce bloc et presser **Ctrl V** (commandes copier coller standards de Windows)

Ctrl X supprime le bloc sélectionné (mais le garde dans un buffer) pour le déplacer par la commande **Ctrl V** pour l'insérer ou la commande **Ctrl T** pour remplacer

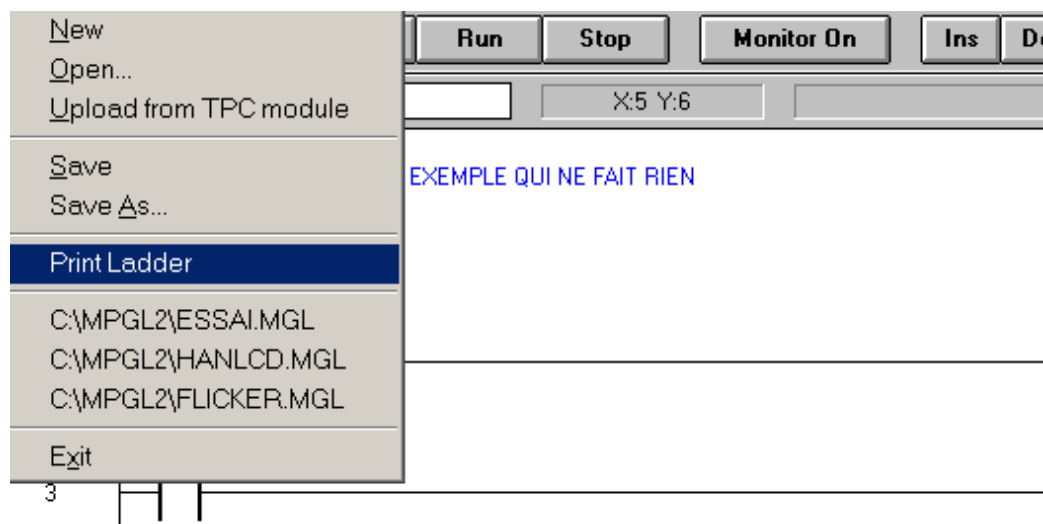
Pour dupliquer une ligne, il suffit de se positionner sur la ligne suivante , en début de ligne et de presser **Ctrl A** , la ligne insérée est identique à la ligne précédente mais le texte des symboles



ATTENTION Pour toutes les commandes copier , coller ,déplacer , FAIRE EXTREMEMENT ATTENTION, LE GRAPHIQUE AINSI MODIFIE PEUT REVELER DE GRANDES SURPRISES , à vérifier très soigneusement

IMPRESSION des LADDER

Sélectionner Print Ladder de l'onglet fichier de MGPL2 , le programme s'imprime sur l'imprimante par défaut de Windows .



NOTES

Chapitre 6

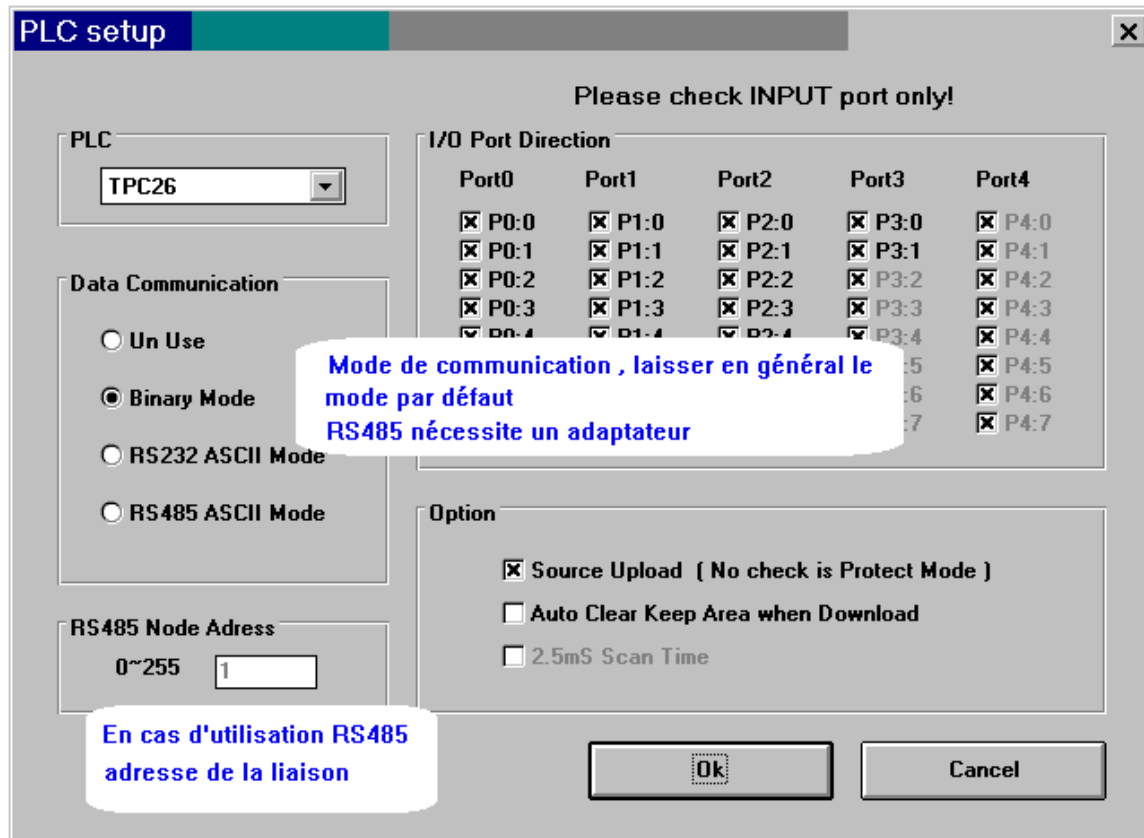
Transmissions

de

données

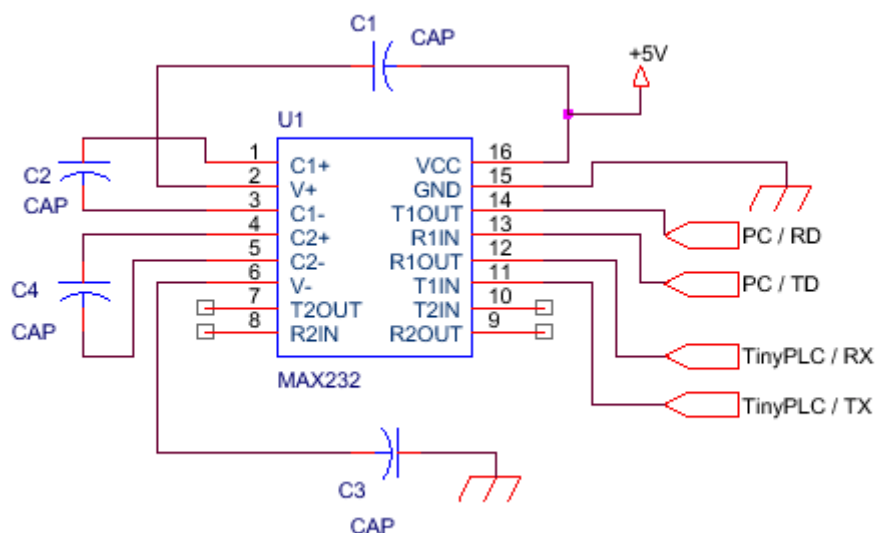
COMMUNICATION entre PC et TinyPLC

Les paramètres de communication sont définis dans PLC setup

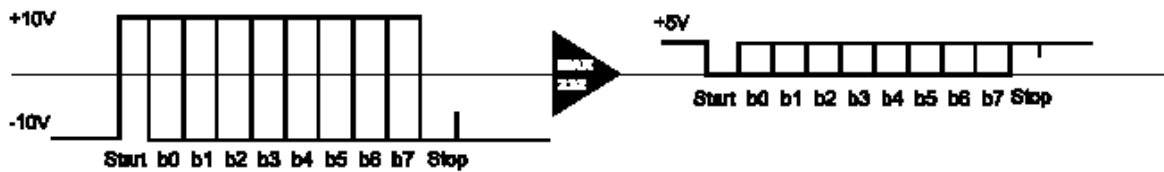


Les fonctions de communications utilisent les broches TX et RX des PLC, ces broches nécessitent un adaptateur de niveau pour communiquer avec l'extérieur (PC par exemple avec logiciel de communication). Ne pas confondre avec les broches DTX et DRX qui servent au téléchargement (sauf sur le TPC 26 où seules TX et RX existent et nécessitent aussi un adaptateur de niveau).

Exemple d'adaptateur RS232C



Les niveaux +10 volts et – 10 volts issus du PC sont convertis en 0 / 5 Volts pour le TinyPLC , le circuit utilisé est un MAX232 .(ou équivalent)

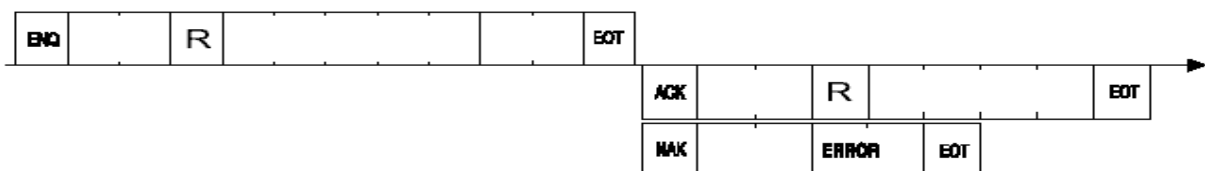


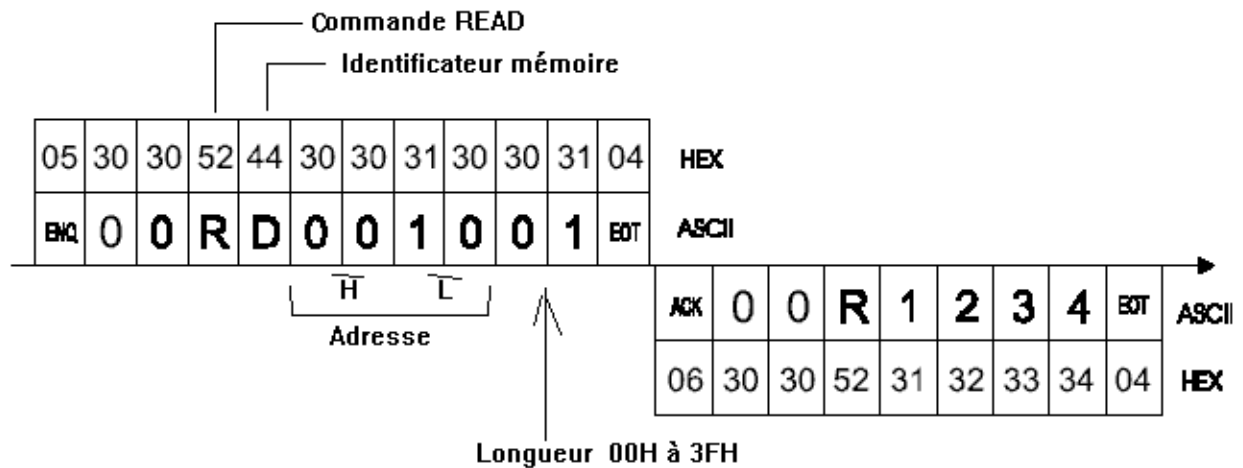
Communication standard

Paramètre	Spécification
Interface	RS232C ou RS485
Méthode de transmission	Half duplex asynchrone
Nombre de caractères	8 bits de données pas de parité 1 bit de stop
Contrôle	Code ASCII
Vitesse	9600 bauds
Distance	RS232C max 16 m RS485 maxi 800 m
Nombre de port	Uniquement un seul
RS232C	Pour PC port 9 ou 25 broches, utilisation des deux lignes RD et TD . Un convertisseur RS232 nécessaire pour le TinyPLC
RS485	Uniquement deux fils , convertisseur nécessaire pour TiniPLC (MAX485 ou LTC485 par exemple)

LIAISON ASCII RS232C

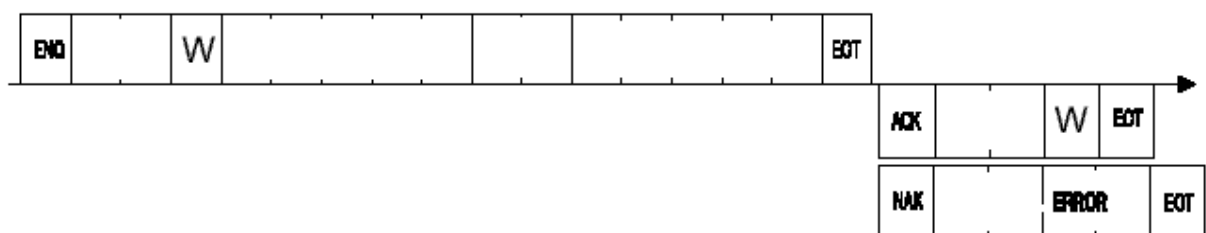
LECTURE de MOTS



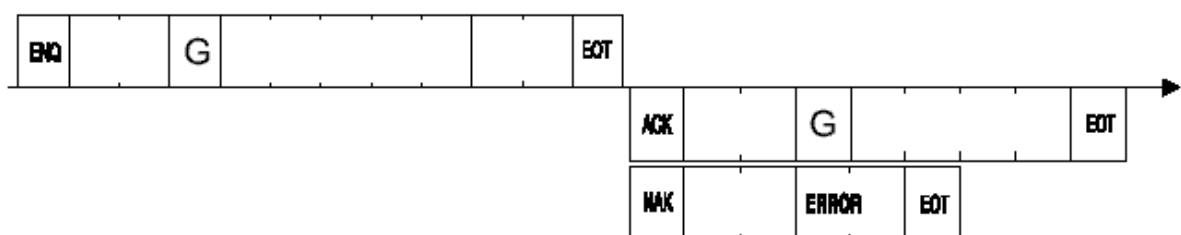


Domaine	Adresses	Longueur transm.
P	P0000~P015	P0000~P0015
M	M0000~M127	M0000~M0127
K	K0000~K031	K0000~K0031
F	F0000~F015	F0000~F0015
A		A0000~A0015
N		N0000
T		T0000~T0255
C		C0000~C0255
D		D0000~D1023
H		H0000~H0079
G		G0000~G0039

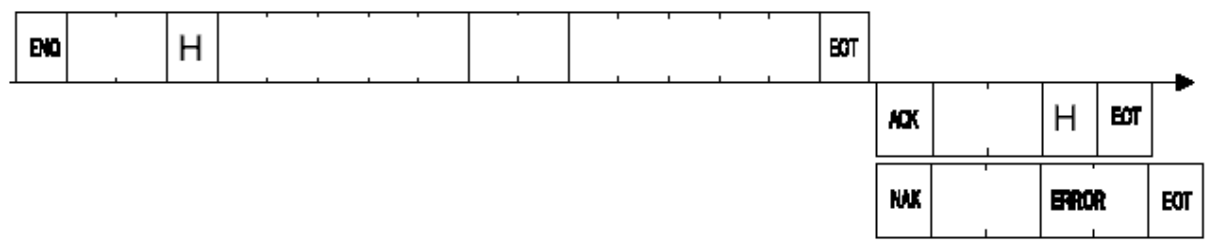
ECRITURE de MOTS



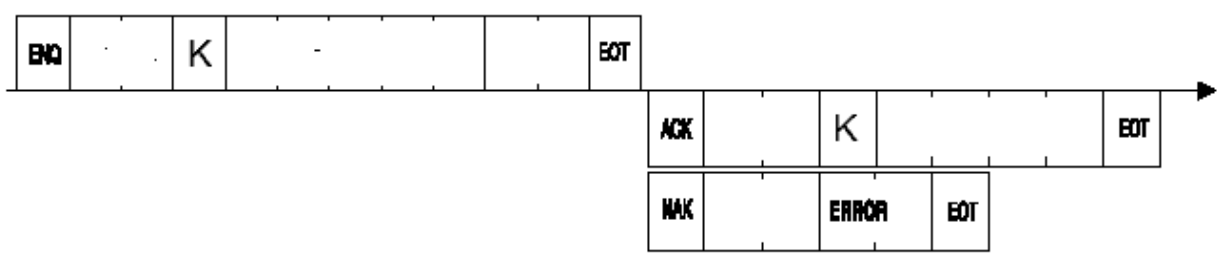
LECTURE de BITS



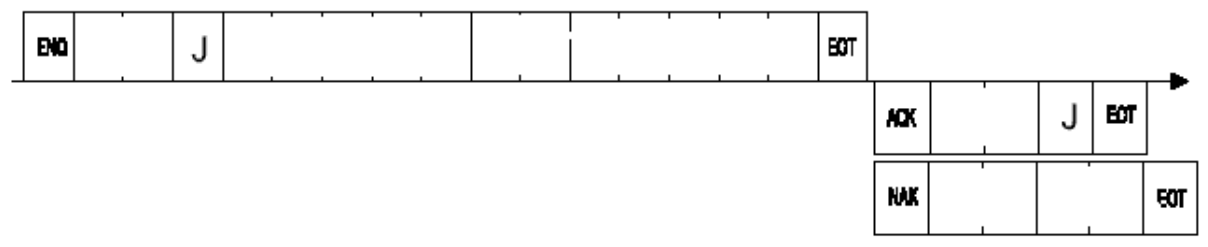
ECRITURE de BITS



LECTURE de BYTES

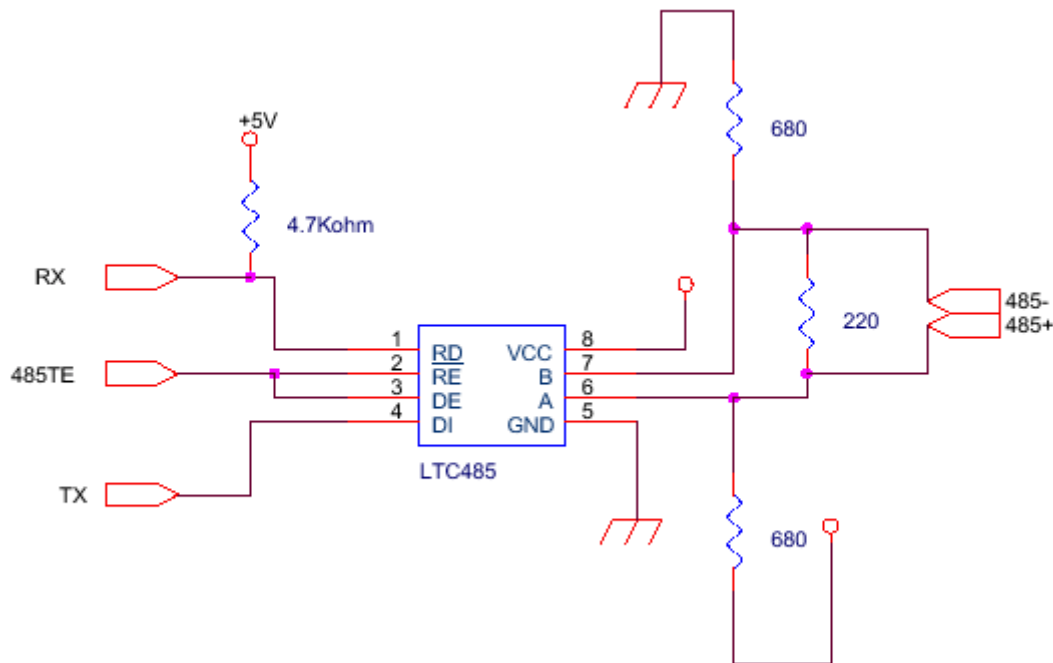


ECRITURE de BYTES

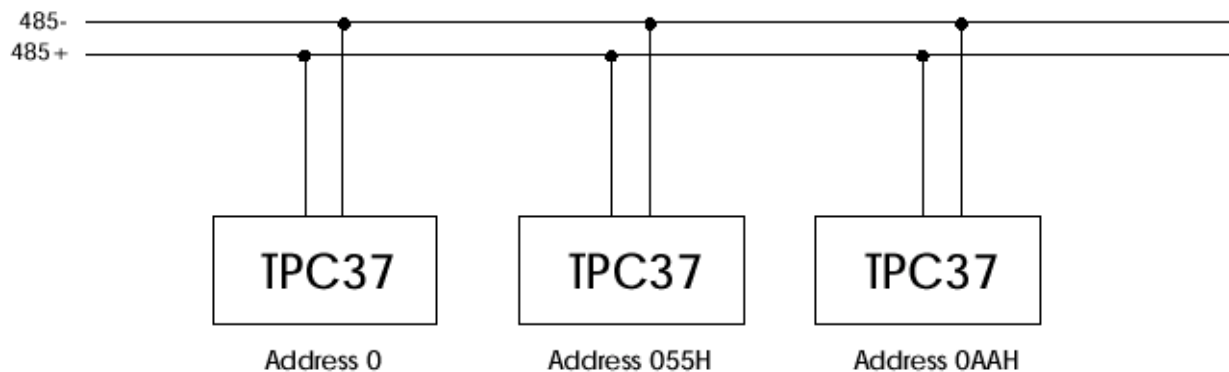


LIAISON ASCII RS485

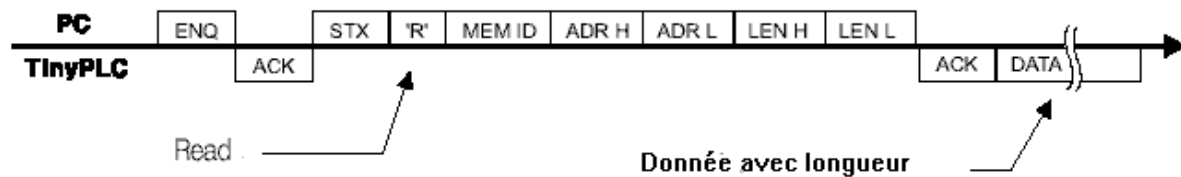
La longueur maximum d'une liaison RS232 est de l'ordre de 16 m , en cas de dépassement de cette longueur ,il est recommandé d'utiliser une liaison RS485 avec un adaptateur spécifique . La longueur de la RS485 peut atteindre 800 m.



Il est possible en utilisation RS485 de spécifier l'adresse lors de l'initialisation de la transmission, il est ainsi possible d'utiliser une même ligne plusieurs TinyPLC



LIAISON BINAIRE RS232C



Valeurs pour le TPC 37

Indication	Réel M	Adresse début	Longueur max	Remarques
P	Réel P	0~15	1~16	
M	Réel M	0~127	1~128	
K	Réel K	0~31	1~32	
S	Réel S	0~31	1~32	
F	Réel F	0~15	1~16	
A	Réel AD	0~15	1~16	Stocké dans l'ordre H,L
N	Réel CNT	0~1	1~2	Stocké dans l'ordre H,L
T	Réel T	0~511	1~512	Actuellement 256 mots , mais transmis par BYTE , dans l'ordre H,L
C	Réel C	0~511	1~512	Stocké dans l'ordre H,L
D	Réel D	0~2047	1~2048	Stocké dans l'ordre H,L
H	Réel CH	0~79	1~80	
G	Réel G	0~39	1~40	

Les valeurs réelles T C D sont des mots , mais sont transmis par Byte , elles sont mémorisées dans l'ordre H et L

Exemple de programme en Qbasic sur PC . Appuyer sur ESPACE pour lire 50 bytes des réels de T , et l'afficher sur l'écran . Finir par la touche Esc

```

'
'      Read from TinyPLC
'      RS232C COMMUNICATION PROGRAM
'
'
'      ouvrir com1      9600      N,8,1
'
OPEN "com1:9600,n,8,1,cs,ds" FOR RANDOM AS #1
'
'  Demarre une boucle infinie
'
DO
  LL = 50  'longueur de la transmission
  ad = 0   Adresse de lecture

```

```

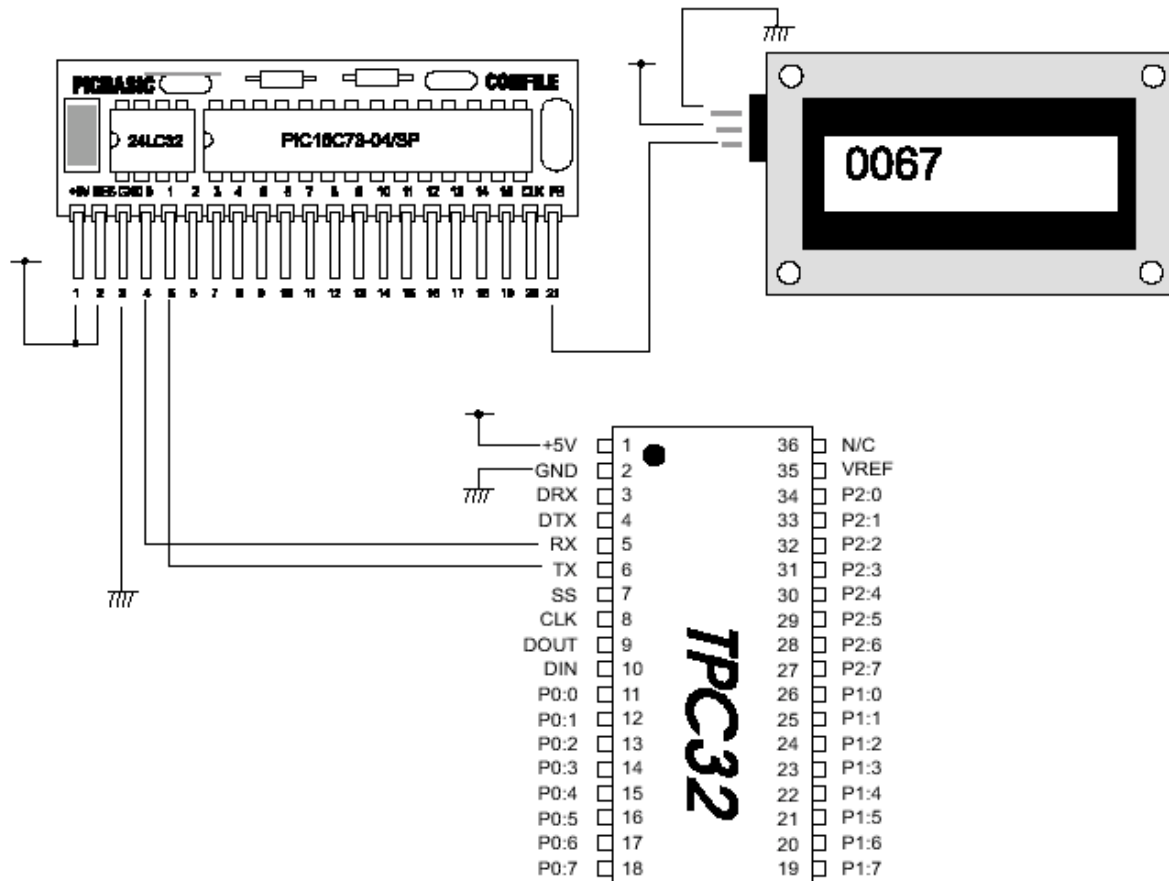
DO
  I$ = INKEY$
  LOOP UNTIL I$ = CHR$(32) OR I$ = CHR$(27)
  '
  PRINT #1, CHR$(&H5); Commande pour réception ACK
  '
  a$ = INPUT$(1, #1)
  '
  '   Commande de transmission , indication mémoire , adresse , donnée
  '   incluant ACK
  PRINT #1, CHR$(2); "R"; "T"; CHR$(ad \ 256); CHR$(ad MOD 256);
CHR$(LL \ 256); CHR$(LL MOD 256);
  '
  a$ = INPUT$(1, #1)
  PRINT ASC(a$)      ' Confirme réception et l'affiche sur la fenêtre
  '

  B$ = INPUT$(LL, #1)
  PRINT "READ DATA"
  '
  FOR I = 1 TO LL      'Affiche les données reçues
    PRINT HEX$(ASC(MID$(B$, I, 1))); " ";
  NEXT I
  PRINT
  PRINT
  '
LOOP UNTIL I$ = CHR$(27) ' Fin si ESC
END

```

Lecture avec PICBASIC

Ci-dessous schéma de raccordement entre un TinyPIC et un PICBASIC (Comfile), permettant de lire des données issues du PLC



```

PicBasic SERIN SEROUT TinyPLC

```

```

DIM D1 AS BYTE
DIM I AS BYTE
DIM ERR AS BYTE
DIM T AS BYTE
DIM CNT AS INTEGER
DIM RT(10) AS BYTE
SET PICBUS HIGH
LCDINIT
CNT=0
ERR=0
10 CNT=CNT+1
OUT 8,1
SEROUT 0,30,0,0,[5]
OUT 8,0

```

```

SERIN 1,30,0,10,ACK_ERR,[I]
IF I<>6 THEN GOTO ACK_ERR

OUT 8,1
SEROUT 0,30,0,0,[2,'R','P',0,0,0,1]
OUT 8,0
SERIN 1,30,0,500,ACK_ERR,[RT(0)~2]

LOCATE 0,0
PRINT HEX(RT(1))
LOCATE 0,1
PRINT "          "
GOTO 10

ACK_ERR:
LOCATE 0,1
PRINT "ACK ERROR"
GOTO 10

```

Ce programme est destiné aux PB 1S - 2B - 2S - 3B - 3H

Pour les PicBasic 2000 (PBM R1 et R5) , il convient d'utiliser le programme suivant qui utilise GET et PUT

```

'
' GET, PUT    TinyPLC <--> PicBasic 2000
'
DIM K AS BYTE
DIM D AS BYTE
DIM K1 AS BYTE

K = 0
K1 = 0
D = 0
LCDINIT
SET RS232 9600
20 BCLR          ' 수신 버퍼를 모두 클리어
PUT 5
GET K, 100, ACK_ERR
IF K<>6 THEN GOTO ACK_ERR
PUT 2,ASC("R"),ASC("P"),0,0,0,1
GET K1,100,ACK_ERR
IF K<>6 THEN GOTO ACK_ERR
GET D,100

```

```

LOCATE 0,0
PRINT HEX(D)
DELAY 100
GOTO 20

```

```

ACK_ERR:
LOCATE 0,1
PRINT "ACK ERR"
GOTO 20

```

Ecriture de BYTE

Les données sont stockées dans le format suivant , (si vous écrivez dans le domaine P , il est possible d'écrire directement sur le port)

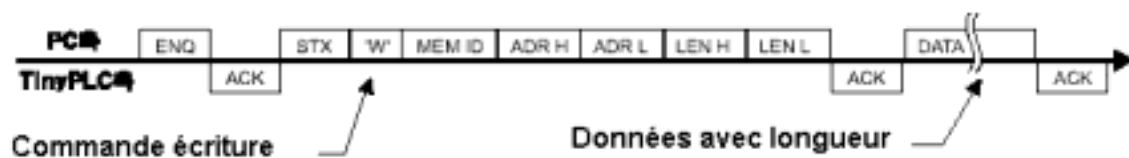


Tableau des adresses et longueurs possibles pour le TPC37

Indication	Réel M	Adresse début	Longueur max	Remarques
P	Réel P	0~15	1~16	
M	Réel M	0~127	1~128	
K	Réel K	0~31	1~32	
S	Réel S	0~31	1~32	
F	Réel F	0~15	1~16	
A	Réel AD	0~15	1~16	Stocké dans l'ordre H,L
N	Réel CNT	0~1	1~2	Stoché dans l'ordre H,L
T	Réel T	0~511	1~512	Actuellement 256 mots , mais transmis par BYTE , dans l'ordre H,L
C	Réel C	0~511	1~512	Stocké dans l'ordre H,L
D	Réel D	0~2047	1~2048	Stocké dans l'ordre H,L
H	Réel CH	0~79	1~80	
G	Réel G	0~39	1~40	

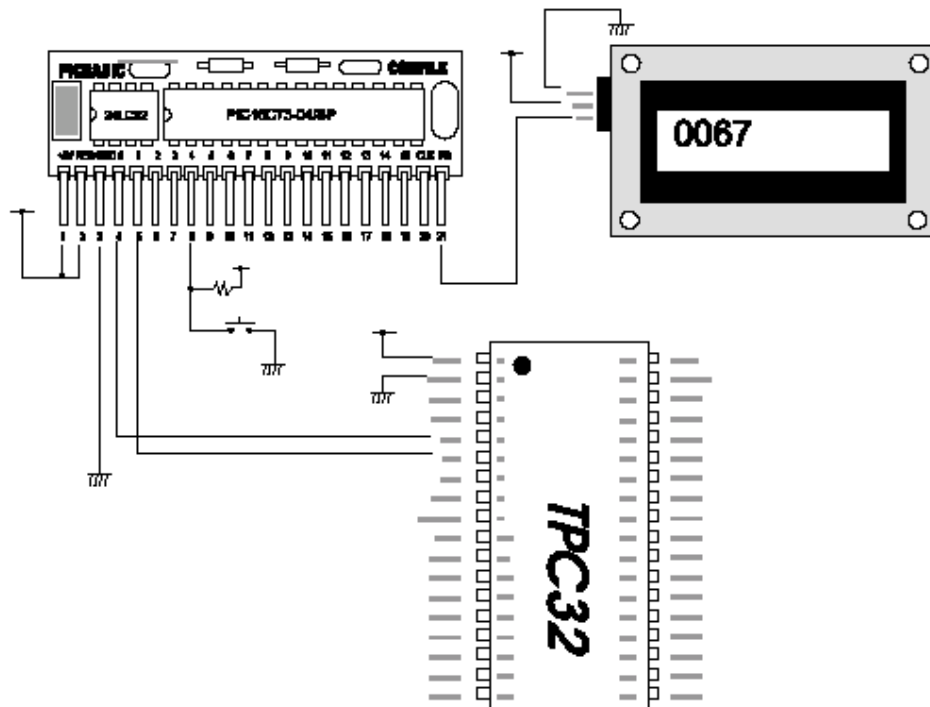
Programme en Qbasic permettant d'écrire des données dans TinyPLC

```

'
' Ecriture sur TinyPLC
' RS232C COMMUNICATION PROGRAM
'
OPEN "com1:9600,n,8,1,cs,ds" FOR RANDOM AS #1
'
DO      ' Demarre une boucle infinie
'
  LL = 16  ' Reset longueur
  ad = 0   ' Reset adresse
  '
  DO
    I$ = INKEY$  ' attente de la touche ESPACE ou ESC
    LOOP UNTIL I$ = CHR$(32) OR I$ = CHR$(27)
    '
    PRINT #1, CHR$(&H5);  ' Transmet le byte ENQ
    '
    A$ = INPUT$(1, #1)  ' Reception de ACK
    '
    ' transmission de STX , commande W , adresse mémoire et longueur
    '
    PRINT #1, CHR$(2); "W"; "T"; CHR$(ad \ 256); CHR$(ad MOD 256);
    CHR$(LL \ 256); CHR$(LL MOD 256);
    '
    '
    A$ = INPUT$(1, #1)  ' Attente de ACK
    PRINT ASC(A$)      ' affiche ACK sur l'écran
    '
    PRINT "WRITE DATA"  ' Transmet les données avec longueur
    PRINT #1, STRING$(LL, CHR$(0)); ' WRITE TO ALL 0
    '
    A$ = INPUT$(1, #1)  ' Reception ACK
    PRINT ASC(A$)      ' ack
    PRINT
    '
  LOOP UNTIL I$ = CHR$(27)  ' Si ESC sortie du programme
END
'

```


Utilisation de PicBasic



Programme PicBasic pour écrire des données

```

' SERIN, SEROUT      Pour PicBasic 1S 2B 2S 3B 3H
'
DIM D1 AS BYTE
DIM I AS BYTE
DIM ERR AS BYTE
DIM T AS BYTE
DIM CNT AS INTEGER
DIM RT(10) AS BYTE
SET PICBUS HIGH
LCDINIT
CNT=0
ERR=0
10 IF KEYIN(4) = 1 THEN GOTO 10

SEROUT 0,30,0,0,[5]
SERIN 1,30,0,10,ACK_ERR,[I]
IF I<>6 THEN GOTO ACK_ERR

SEROUT 0,30,0,0,[2,'W','P',0,0,0,2]
SERIN 1,30,0,500,ACK_ERR,[I]
SEROUT 0,30,0,0,[100,100]
SERIN 1,30,0,500,ACK_ERR,[I]

```

```

LOCATE 0,0
PRINT "WRITE DONE ";HEX
GOTO 10

```

```

ACK_ERR:
  LOCATE 0,1
  PRINT "ACK ERROR"
  GOTO 10

```

Pour le PicBasic 2000 PBM R1 et R5

```

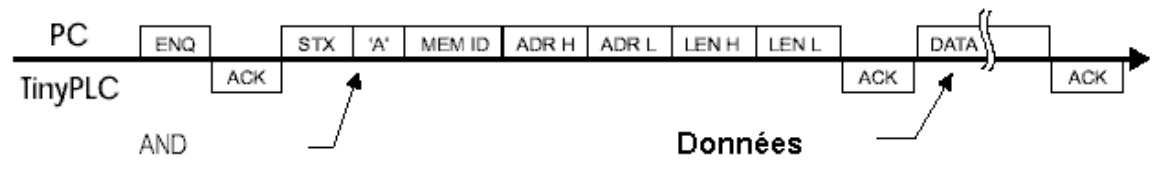
*
*   GET, PUT   Pour liaison entre TinyPLC et PicBasic 2000 R1 et R5
*
  DIM K AS BYTE
  DIM D AS BYTE
  DIM K1 AS BYTE

  K = 0
  K1 = 0
  D = 0
  LCDINIT
  SET RS232 9600
20 IF KEYIN(4) = 1 THEN GOTO 20
  BCLR                      ' 수신 버퍼를 모두 클리어
  PUT 5
  GET K, 100, ACK_ERR
  IF K<>6 THEN GOTO ACK_ERR
  PUT 2,ASC("W"),ASC("P"),0,1,0,1
  GET K1,100,ACK_ERR
  IF K<>6 THEN GOTO ACK_ERR
  PUT 255
  LOCATE 0,0
  PRINT "WRITE DONE"
  DELAY 100
  GOTO 20

ACK_ERR:
  LOCATE 0,1
  PRINT "ACK ERR"
  GOTO 20

```

Transmission Byte AND



Transmission Byte OR



Notes

Chapitre 7

Installation

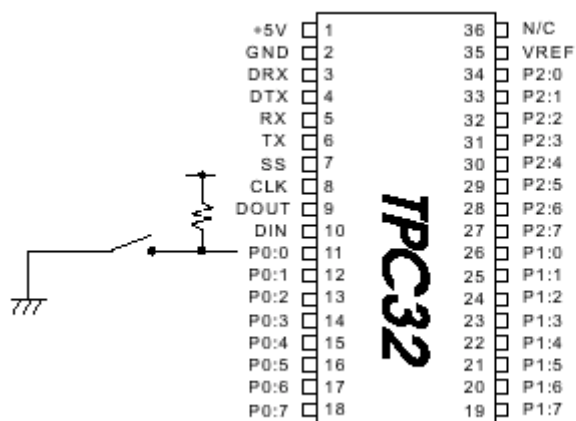
et

Organisation

Entrées / Sorties des TinyPLC

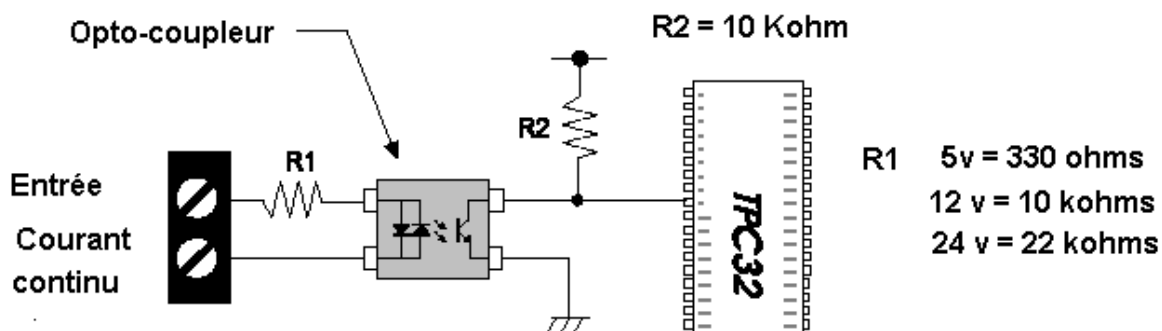
Les entrées sorties des modules TinyPLC ont les mêmes caractéristiques que les circuits intégrés TTL ou CMOS . Si elles sont utilisées en entrées , elles passent en haute impédance , sans consommation électrique. Le 5 Volts est reconnu comme un 1 (ON) et 0 volt est considéré comme un 0 (OFF). La plupart des broches d'E/S "TinyPLC" peuvent être indépendamment configurées pour être utilisées en entrée ou en sortie. Certaines peuvent également faire office d'entrée dans le cadre d'une conversion analogique/numérique. Dans ces conditions, il conviendra d'être extrêmement vigilant avec le type de signaux appliqués sur ces broches et le type de dispositifs pilotés par ces broches. Ceci est d'autant plus vrai lors des premières phases d'utilisation ou pour le besoin de vos tests, pendant lesquels vous serez amené à changer souvent le rôle de vos "broches". Correctement utilisé, votre "TinyPLC" vous permettra de réaliser d'innombrables quantités de montages et d'applications dont vous ne pouvez même pas imaginer la puissance. Toutefois il vous faut impérativement garder à l'esprit que les "TinyPLC" ne sont rien d'autre qu'un microcontrôleur et au même titre qu'avec tout autre microcontrôleur il vous faut respecter certaines règles de bases afin d'éviter qu'il ne rende l'âme !

- 1) Ne jamais alimenter les "TinyPLC" sous une tension supérieure à +5 Vcc.
- 2) Si vous appliquez des tensions issues de dispositifs extérieurs sur les "TinyPLC":
 - Vérifiez toujours que ces tensions soient égales ou inférieures à + 5 Vcc.
 - Coupez **en PRIORITÉ** l'alimentation des capteurs externes **AVANT** de couper celle du "TinyPLC" afin d'éviter qu'une tension soit toujours présente sur l'entrée du "TinyPLC" alors que ce dernier n'est plus alimenté (sans quoi le port du "TinyPLC" serait HS).
 - Selon la même recommandation que ci-dessus, vérifiez que vous ne disposez pas de condensateurs de forte valeur reliés sur les entrées des "TinyPLC", lesquels pourront stocker une tension qui viendra alors se décharger dans le "TinyPLC" lorsque vous couperez les alimentations.
- 3) Ne jamais inverser la polarité d'alimentation du "TinyPLC".
- 4) Lorsque vous utilisez les ports du "TinyPLC" en entrées, n'utilisez jamais de grand fils pour y raccorder des boutons-poussoirs et autres capteurs sans avoir recours à un circuit de mise en forme et de protection (circuit RC avec zener de protection ou préférence par opto-coupleur). Si pour vos tests vous n'utilisez pas de protection de ce type, limitez la longueur de vos fils à 3 ou 4 cm afin d'éviter les phénomènes de "latch-up" ou de destruction par électricité statique.



Il est possible de connecter directement un interrupteur , comme dans le schéma ci-contre , sous réserve de mettre une résistance de rappel au 5 volts . (valeur de 4,7 K ou 10 K) Quand le contact est établi ma borne P0 :0 passe au niveau 0. Le fil de raccordement ne devra pas dépasser 3 ou 4 cm.

Pour connecter des signaux dont la tension est supérieure à 5 volts , il est possible d'utiliser des opto-coupleurs



- 5) Utilisez impérativement des diodes de protection lorsque vous pilotez (via une interface de puissance) des charges inductives (moteurs par exemple) et évitez de placer le câble de téléchargement à côté de cette source.
- 6) Découplez rigoureusement l'alimentation du "TinyPLC" (au plus près de celui-ci).
- 7) Avant d'appliquer une quelconque tension (+ 5V ou masse) sur une des broches du "TinyPLC", vérifiez IMPÉRATIVEMENT que cette broche ai bien été configurée en ENTREE. Dès lors, ne reliez aucune tension (+ 5V ou masse) sur les ports du "TinyPLC" configurés en sorties (sous peine de court-circuit et de destruction de ces derniers).
- 8) Passez toujours par un montage à transistor ou à opto-coupleur pour alimenter et piloter un dispositif consommant plus d'une vingtaine de milli-ampère.
- 9) Si certaines broches du "TinyPLC" ne sont pas utilisées pour les besoins de votre application, configurez tout de même impérativement ces dernières en SORTIE et placez ces dernières au niveau logique « 0 ».
- 10) Comme TOUT microcontrôleur, les "TinyPLC" sont sensibles à l'électricité statique. Ces derniers devront donc être manipulés (et soudés) avec les précautions qui s'imposent afin d'éviter leurs destruction ou leur fragilisation.

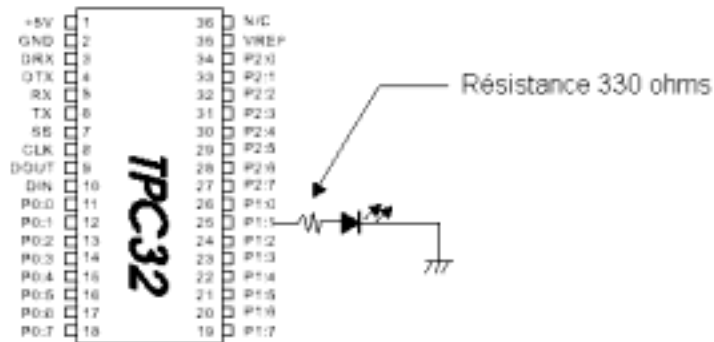


- 11) Ne coupez jamais l'alimentation des "TinyPLC" (de même, ne déconnectez jamais le câble de Download lorsque ce cernier est en phase de programmation ou de monitoring !).

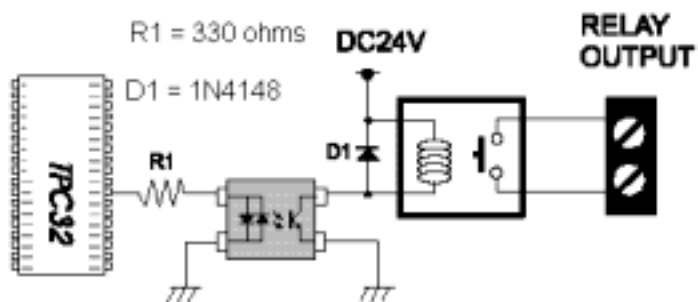
En cas de non respect des limites et des conditions d'utilisations indiquées dans ce manuel, la fiabilité et la durée de vie des "TinyPLC" sera remise en cause (et l'échange du "TinyPLC" ne pourra pas être pris en charge au titre de la garantie).

Organisation des ports de sorties

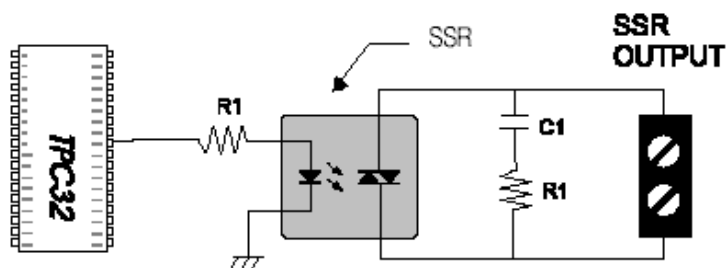
Les ports configurés en sorties sont toujours à l'état Haut ou l'état bas , et peuvent fournir un courant de l'ordre de 25 ma , suffisant pour allumer une LED



Vous pouvez connecter un photo-coupleur de la façon ci-dessous , ceci diminue les parasites extérieurs de ceux de l'alimentation du module lors des commutations



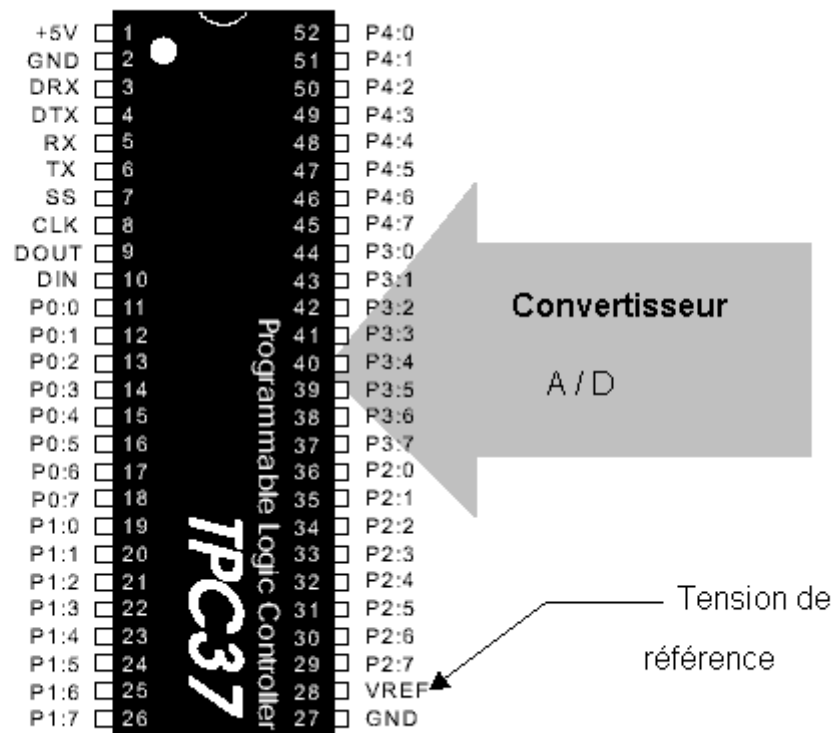
Il est possible de connecter directement un relais statique (SSR)



Convertisseur A/D

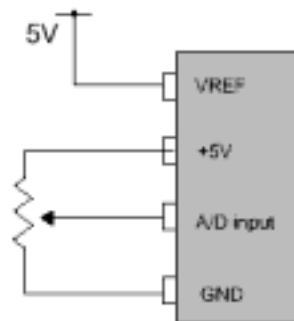
La série TPC 3X comporte un convertisseur A/D 8 voies 10 bits de précision . Elle ne nécessite pas de commande spéciale pour initialiser le convertisseur . La conversion est toujours exécutée pendant le fonctionnement du programme . Le résultat de la conversion est stocké dans les nombre réels A/D ,

Mot réel mémorisé	Port du TPC 32	Port du TPC 37
AD0	P2:0	P3:0
AD1	P2:1	P3:1
AD2	P2:2	P3:2
AD3	P2:3	P3:3
AD4	P2:4	P3:4
AD5	P2:5	P3:5
AD6	P2:6	P3:6
AD7	P2:7	P3:7



La tension de référence (Vref) détermine la tension maximale d'entrée du signal A/D , Le maximum est de 5 volts . Le convertisseur utilise $V_{ref} / 1024$ pour calculer la valeur , si vous mettez 3 volts en Vref , chaque division vaut $3 / 1024$. Sur les TPC26 et 32 les ports utilisés pour le convertisseur A/D doivent être configurés en entrée.

Exemples de circuits A/D :



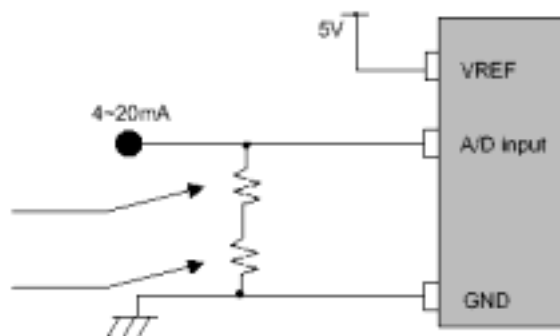
Ce circuit utilise un potentiomètre , il peut être utilisé pour sélectionner une valeur de tempo ou de compteur

Valeur de 1 à 5 Kohms

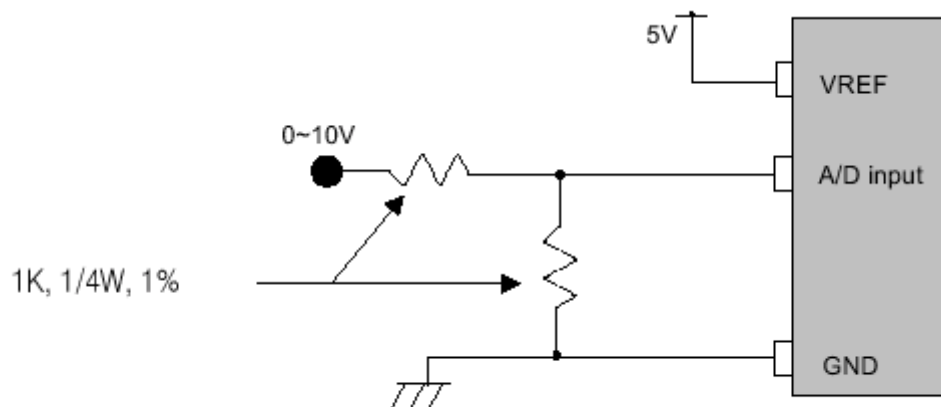
Circuit pour connecter des capteurs avec sortie 4 - 20 ma

220Ω 1/2W 1%

30Ω 1/2W 1%

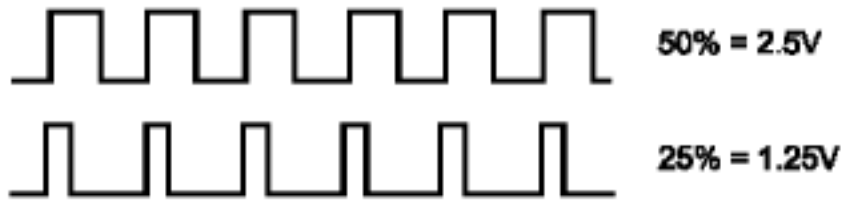


Circuit pour entrée 2 / 10 volts

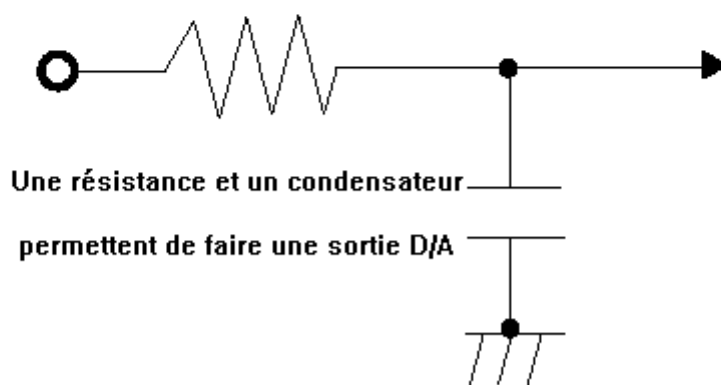
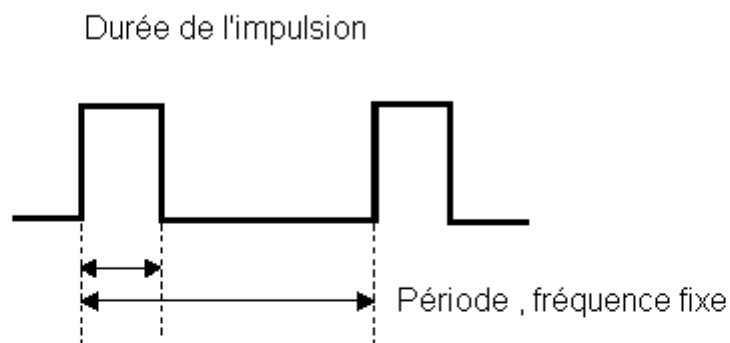


Convertisseur D / A

Le TPC 38 comporte deux convertisseurs D/A sur le principe PWM , DA0 (N° pin 43) et DA1 (n° pin 44) précision 10 bits .PWM = modulation de largeur d'impulsion .

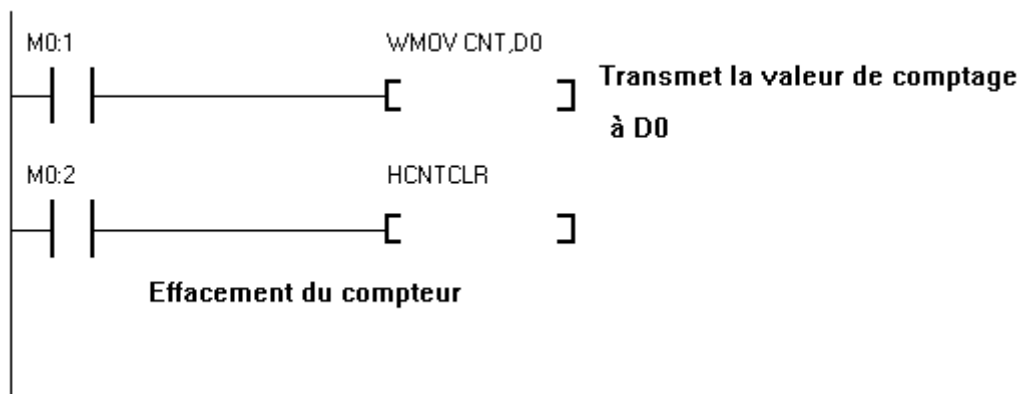


Plus la durée de l'impulsion est longue plus la tension de sortie est importante



Compteur rapide 16 bits

TinyPLC comporte un compteur rapide 16 bits (comptage de 0 à 65535). Il est positionné en P0 :6 . Ce port doit être en entrée s'il est utilisé . Il ne nécessite pas d'opération spéciale . il stocke les impulsions présentes sur P0 :6 dans CNT . Pour lire la valeur il suffit d'utiliser une instruction comme WMOV

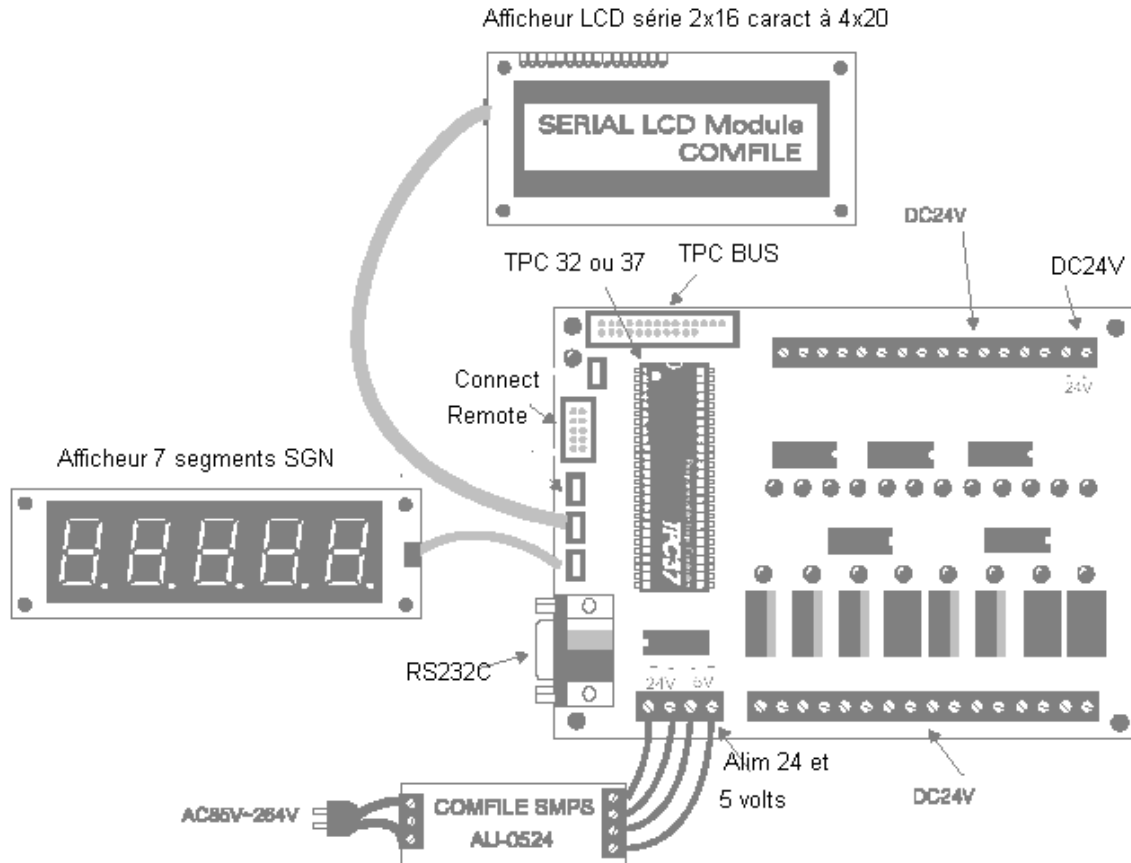


Quand M0 :1 passe à ON , il transmet la valeur de CNT à D0 , pendant que M0 :1 reste à ON , il transmet les impulsions comptées à D0 .

UN compteur rapide est nécessaire dans un PLC , car les impulsions de comptage sont vérifiées à cycle (2.5 ou 5 ms) ce qui limite la fréquence de comptage et les impulsions rapides ne sont donc pas comptées .La fréquence limite du CNT est de 25 à 30 khz , et n'a pas de relation avec le temps de cycle.

Platines de test

Nous disposons de platines test qui vous permettront de vous aider lors des premières phases de développement. Vous trouverez ci-dessous la platine C20R qui utilise un TPC37.

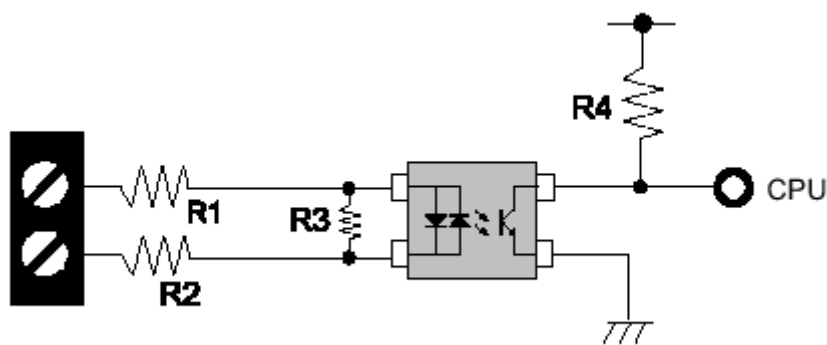
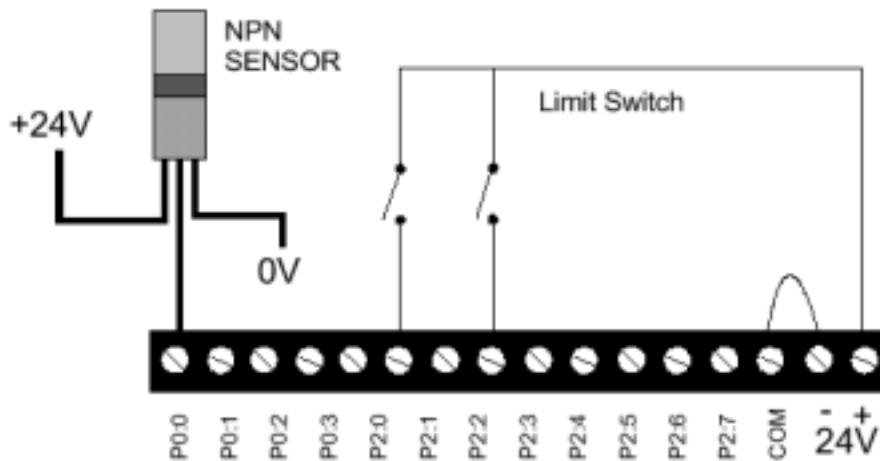


Type du port I/O	Nombre	Ports assignés
Entrées DC 24 volts	14	P0 :0 à P0 :3 P2 :0 à P2 :7
Sorties relais	8	P1 :0 à P0 :7
Module LCD	1	P0 :4
Module SGN	1	P0 :5
Compteur rapide connecté à	1	P0 :6 et P0 :7 P0 :7 led d'état
TPC-BUS module extension	1	P3 :0 à P0 :7 P4 :0 à P4 :0 à P4 :7

NOTE : Les platines « TPNP-BOARD » - « BASE-F24R » - « BASE-C20R » - « BASE-C24R » - « BASE-D24 » - « BASE-D24 » - « BASE-D20 » - « BASE-D32 » - « ADATPC » et « PLC Study-Board » sont uniquement destinées à des usages de prototypage ou d'expérimentation – Ces dernières ne sont pas conçues, ni destinées pour une intégration finale au sein d'applications.

Port entrée 24 volts DC

Tous les ports d'entrées ont une LED pour visualiser l'état de l'entrée. Il est possible de connecter des contacts ou capteurs NPN, ou entrées bi-directionnelles, suivant schémas suivants :

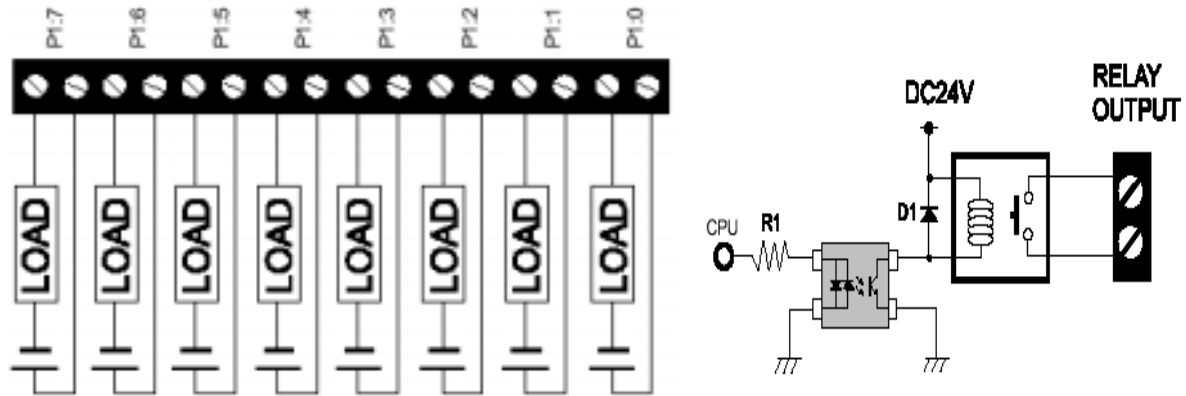


Caractéristiques des ports d'entrées :

Elément	Standard
Tension fournie pour l'utilis.	DC24V
Tension sur les entrées	DC10.2V ~26.4V Ondulation 5%
Courant d'entrée	10mA
ON tension courant	DC9.5V 4mA
OFF tension courant	DC 6V / 1mA
Impédance d'entrée	2.2KΩ
Temps de réponse	10mS
Courant interne	50mA
Indication LED	LED allumée quand le contact est ON

Port de sortie à Relais

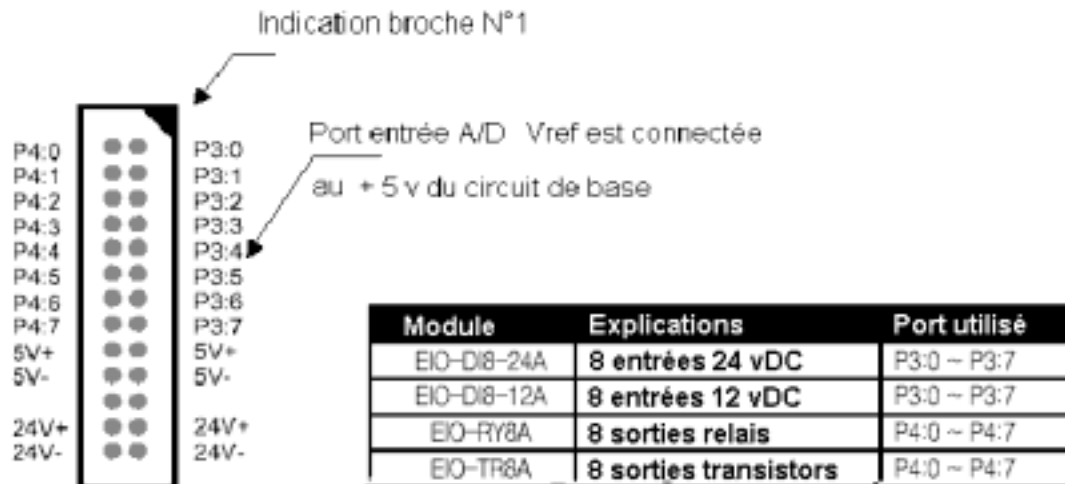
Tous les ports de sorties ont une LED , pour indiquer l'état de la sortie , deux bornes sont attribuées à chaque relais , il n'y a donc pas de commun . Ils peuvent être contrôlés de façon indépendantes , Voir schéma ci-dessous :



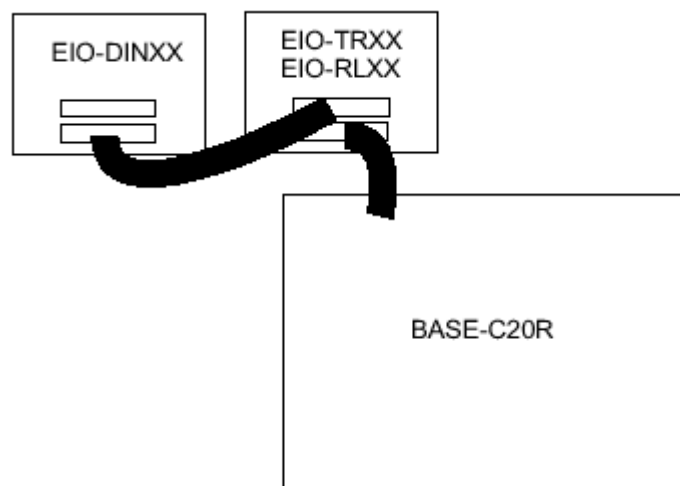
Ces relais ne sont pas prévus, ni autorisés pour piloter une tension supérieur à 24 Vcc.

BUS Standard TPC

TPC6BUS est un bus d'expansion que utilise les ports inutilisés du TPC 37 , vous pouvez connecter les modules d'expansion produits par Comfile . En montant un TPC 32 , le port d'expansion I/O ne peut pas être utilisé .

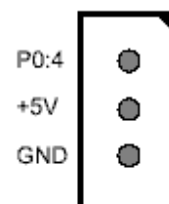


Quand P3 est utilisé en port d'entrée et P4 en port de sortie , il est possible de connecter deux ports (figure ci-après) , les modules ont deux ports de montés pour cette fonction.



Port LCD

Connecté à P0:4 qui doit être contrôlé par la commande LCDOUT . Connecté avec les modules " série " produits par Comfile



Port SGN

Connecté à P0:5 Controlé par la commande SGNOUT

Utilisable avec les modules d'affichage 7 segments ,

Comfile

P0:5
+5V
GND



Compteur rapide

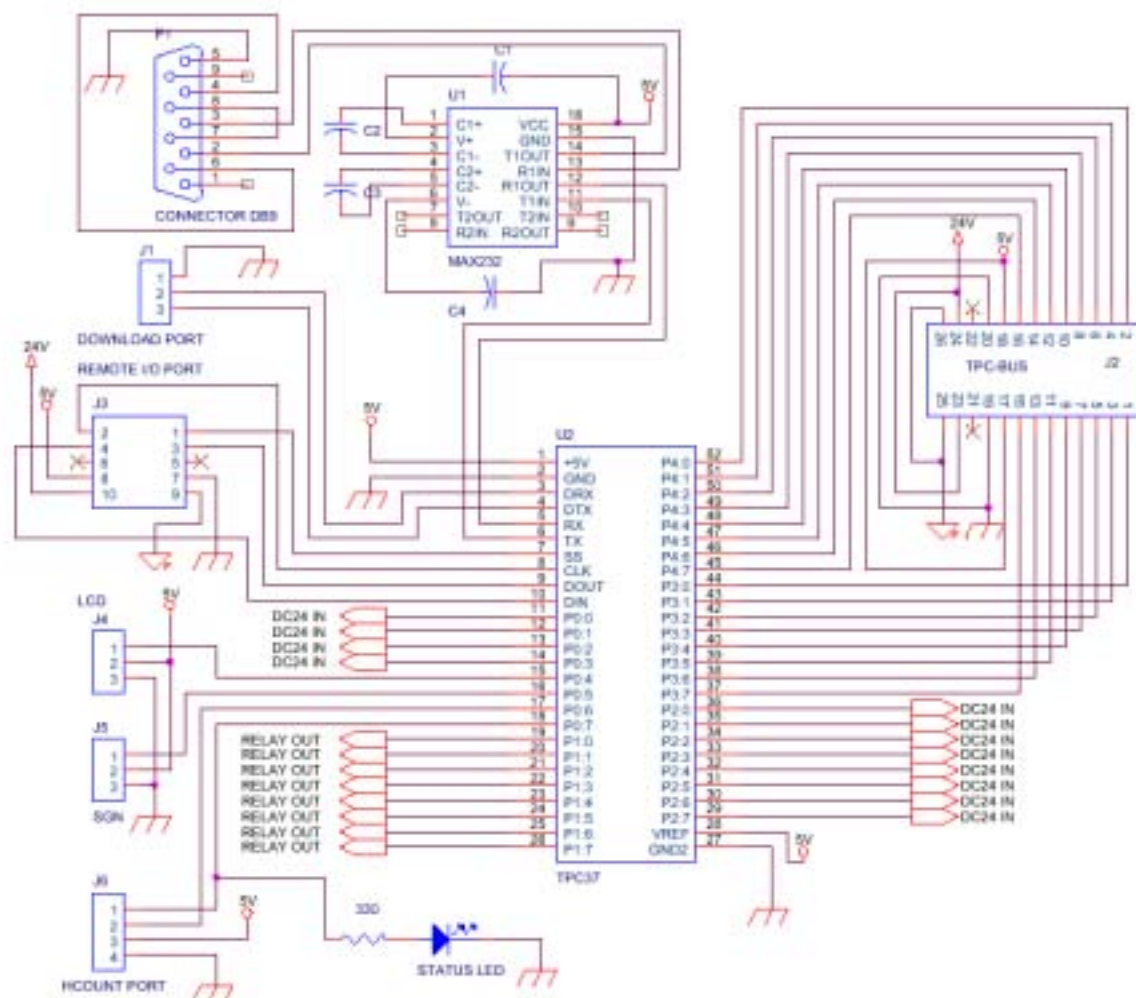
Les impulsion du compteur rapide sont comptées à P0:6

P0:7 est une broche de réserve qui est aussi connectée à la LED d'état du circuit de base

P0:7
P0:6
+5V
GND



Circuit de base C20R



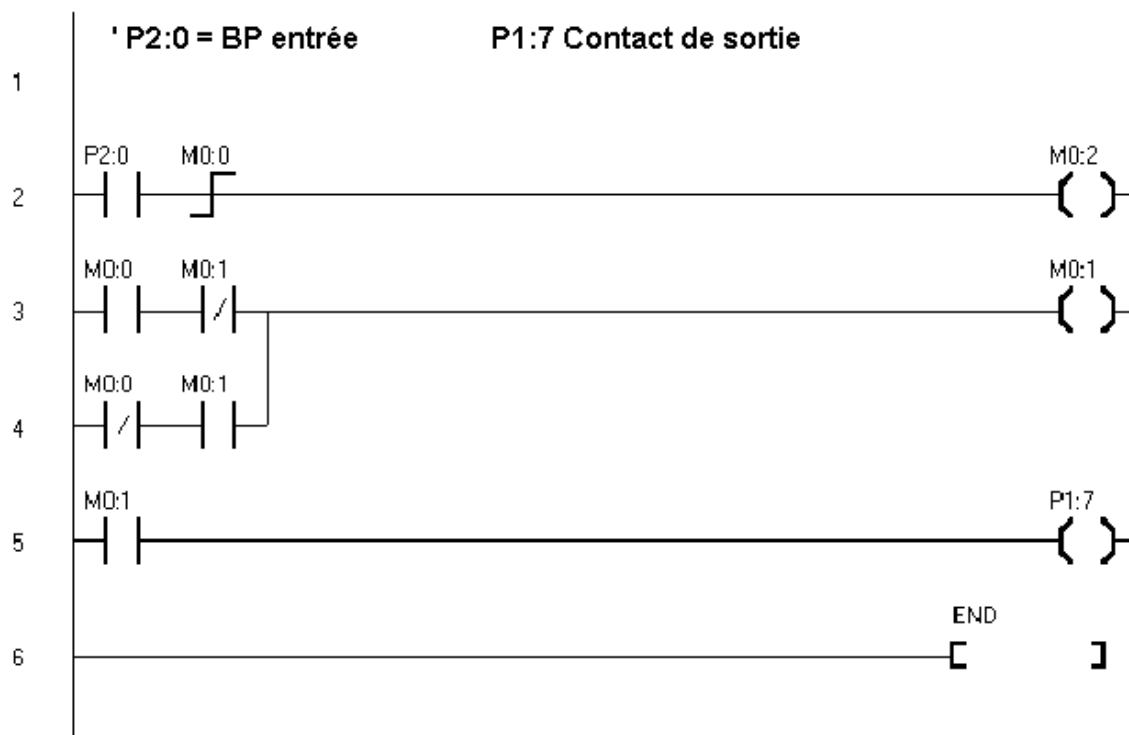
Les circuits entrées 24 vDC et sorties relais ont été omis (voir ports I/O)

Chapitre 8

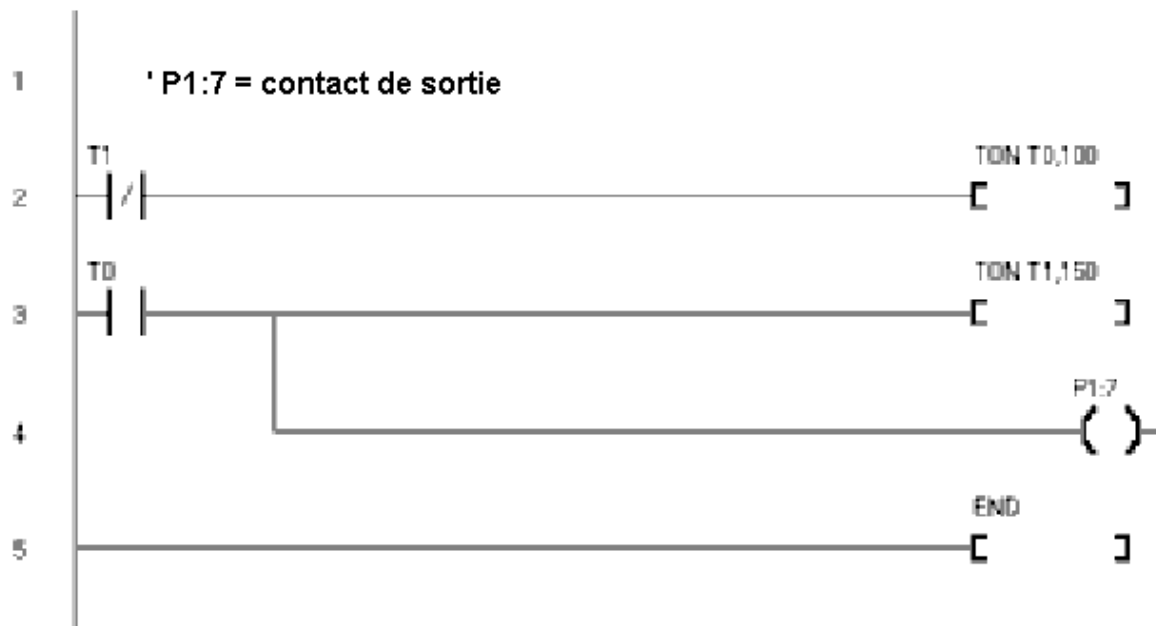
Applications

TinyPLC

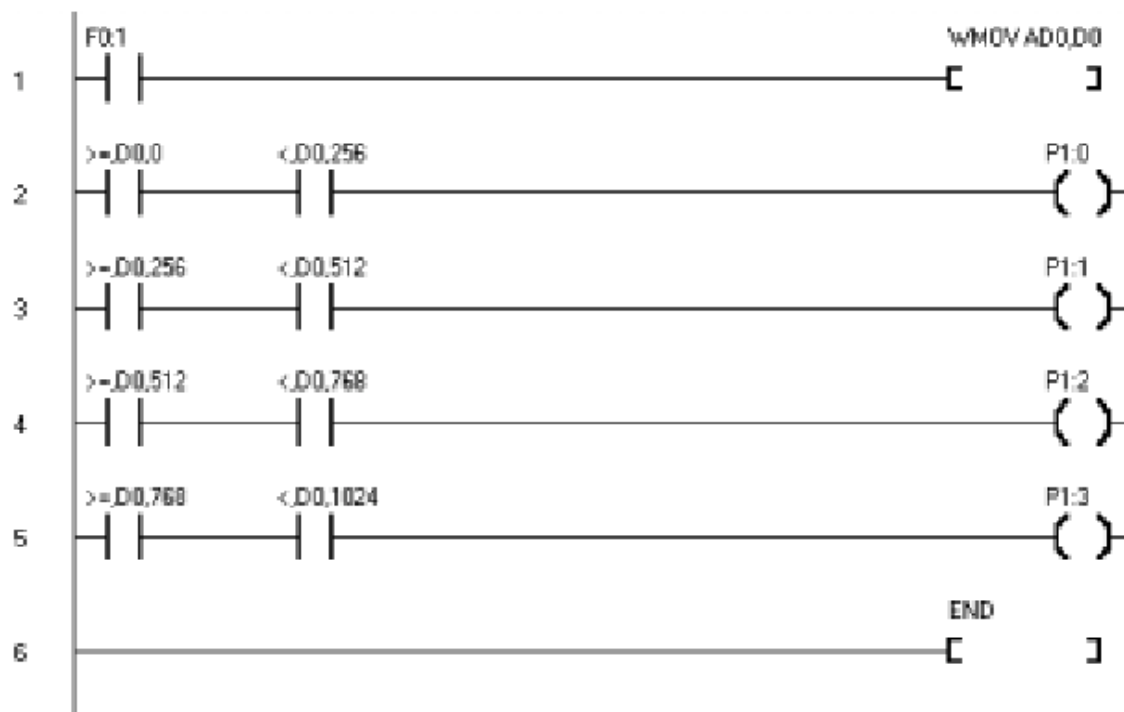
TOGGLE : inverse le sortie à chaque appui sur BP



FLICKER Clignotant inverse une sortie continuellement en utilisant un timer



Commutations de sorties en fonction d'un résultat A/D



Port utilisé	set-up
P1:0	OUTPUT
P1:1	OUTPUT
P1:2	OUTPUT
P1:3	OUTPUT
P3:0 (TPC37) P2:0 (TPC32)	INPUT

La sortie est cumulée en fonction du résultat du convertisseur A/D

0 à 255 sortie sur P1:0

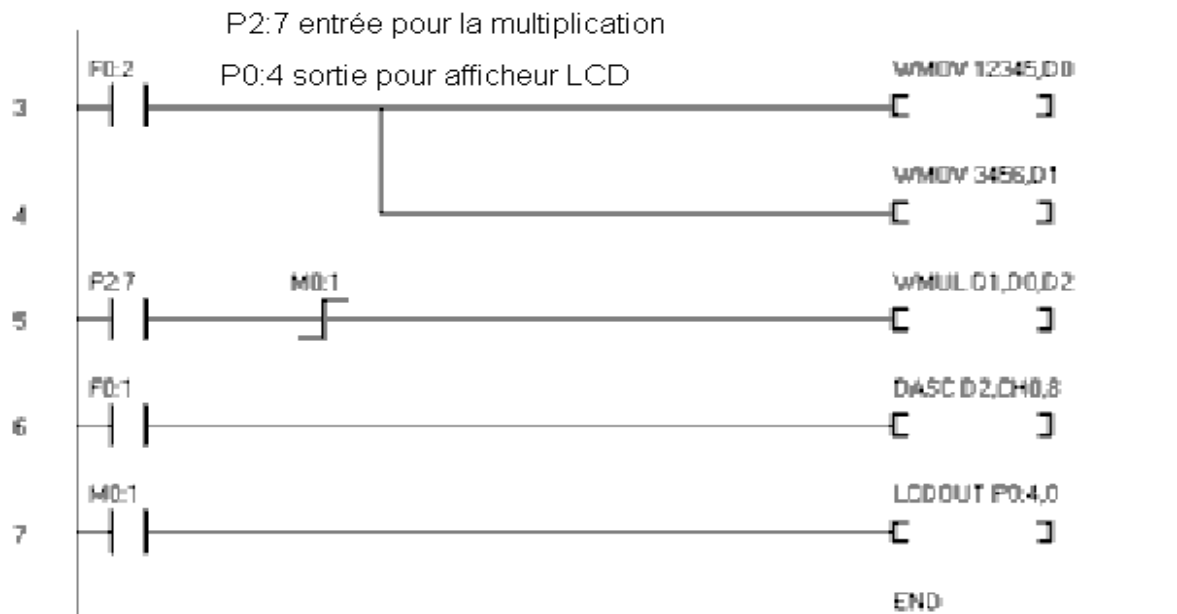
256 à 511 sortie sur P1:1

512 à 767 sortie sur P1:2

768 à 1024 sortie sur P1:3

WMUL2 : affiche le résultat d'une multiplication sur un LCD

Programme de multiplication de deux valeurs , $12345 \times 3456 = 42664320$ et le résultat est affiché sur un LCD . Au départ , D0 vaut 12345 et D1 vaut 3456 . La commande DASC est utilisée pour indiquer le résultat de l'opération sur l'afficheur LCD , stockée en D2 , M0 :1 de LCDOUT est à ON uniquement pour un cycle quand P2 :7 passe à ON .

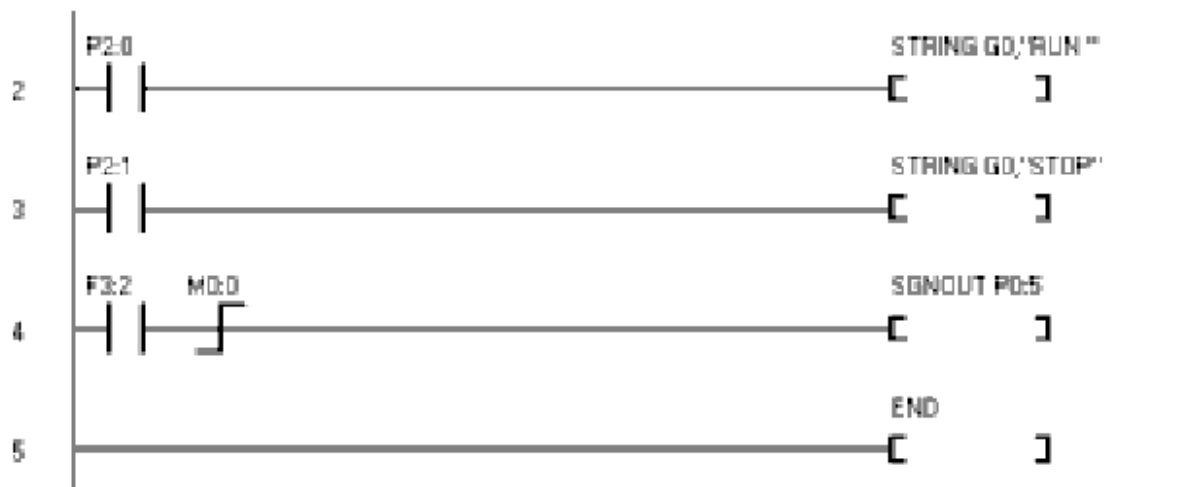


SGN-SW Affiche des caractères sur un afficheur 7 segments

Quand P2:0 est ON , l'afficheur SGN indique RUN

Quand P2:1 est ON , l'afficheur indique STOP

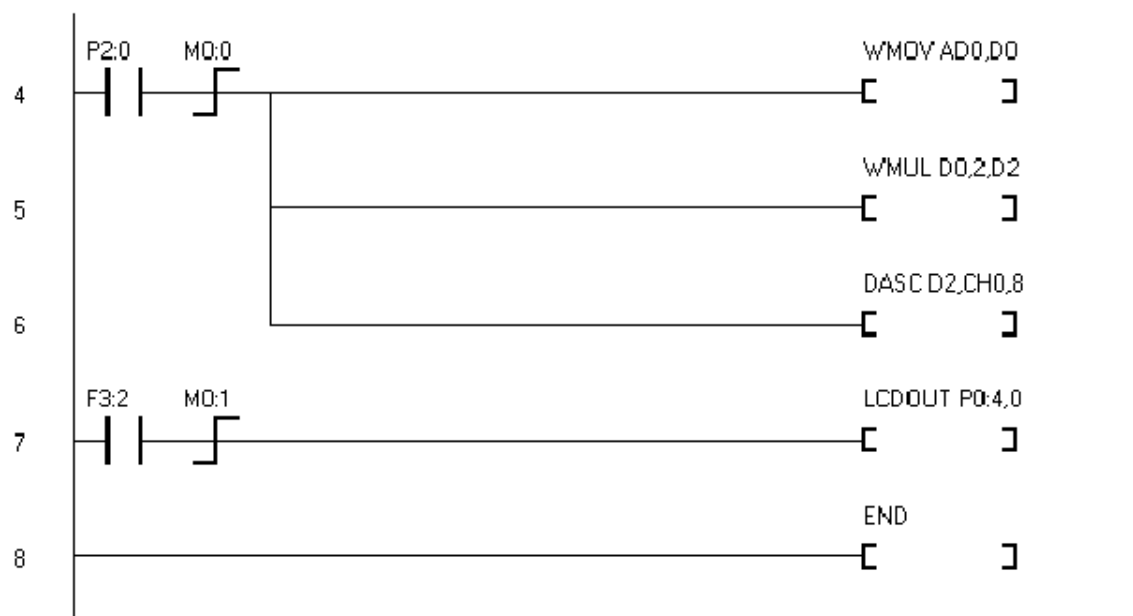
Les caractères sont stockés dans G en utilisant la commande STRING



LCD – AD indique le résultat d'une conversion sur un LCD

Programme pour lire la valeur de AD0 , et l'indiquer sur un afficheur LCD , après multiplication par 2 quand P2 :0 est à ON .WMUL est la commande de multiplication et le résultat est stocké en D2 au format 32 bits (mot long) . Pour transformer une valeur 32 bits en nombre 10 digits , la commande DASC est utilisée . L'afficheur utilisé est ELCD162

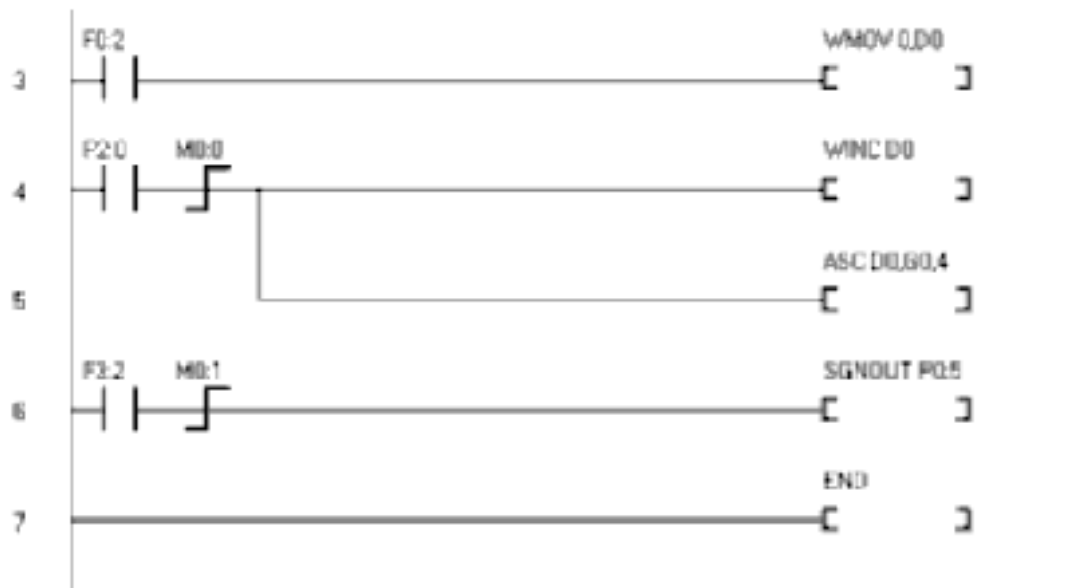
Port utilisé	I/O set-up
P2:0	INPUT
P0:4	OUTPUT
P3:0 (TPC37) P2:0 (TPC32)	INPUT



SGN-CNT Affiche le résultat d'un comptage sur SGN

D0 est une variable 16 bits qui s'incrémente de 1 à chaque fois que l'entrée P2 :0 passe à ON . Ce programme convertit la valeur D0 en code ASCII 10 digits et l'indique sur l'afficheur 7 segments

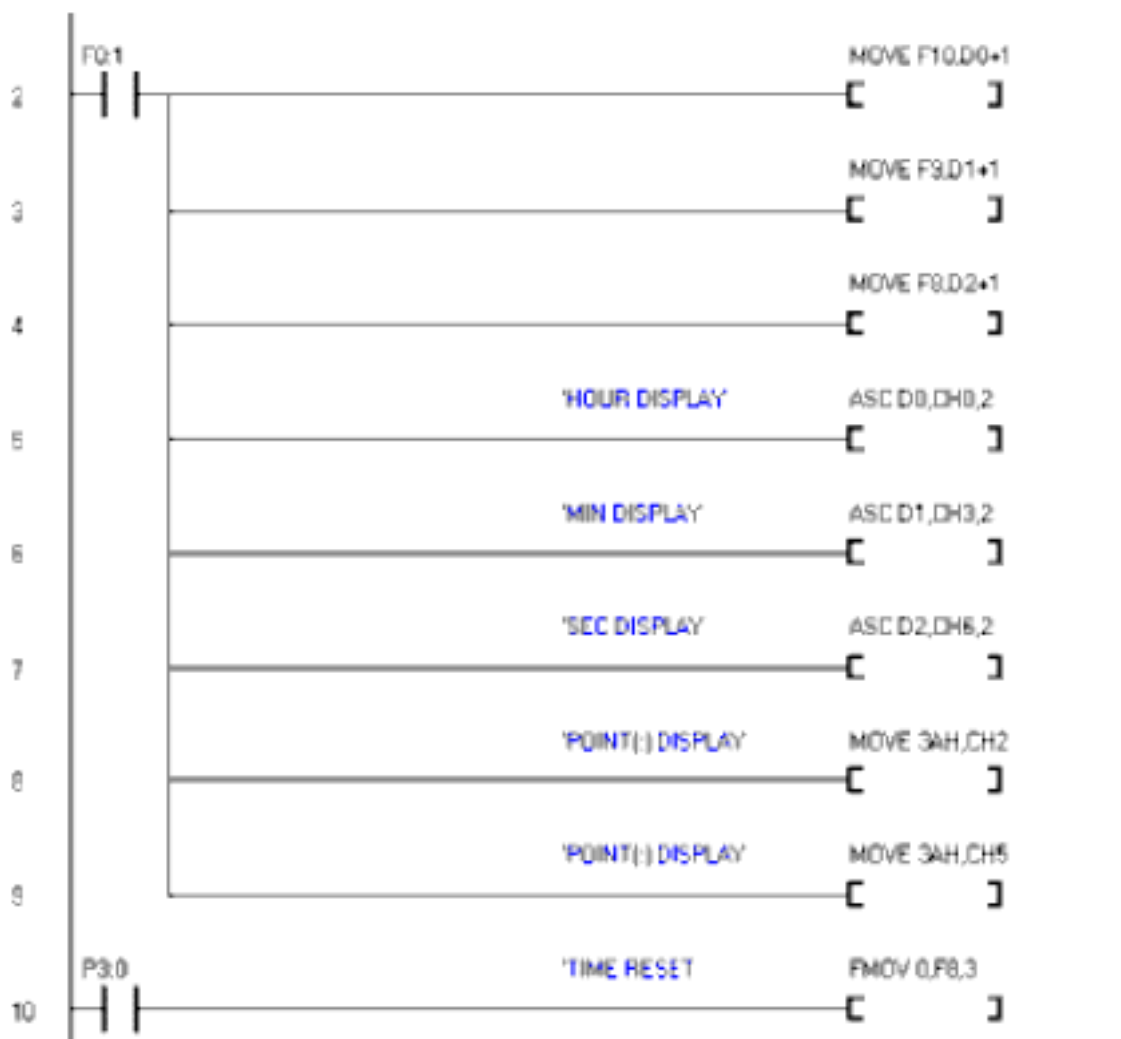
Port utilisé	Set-up I/O
P2:0	INPUT
P0:5	OUTPUT



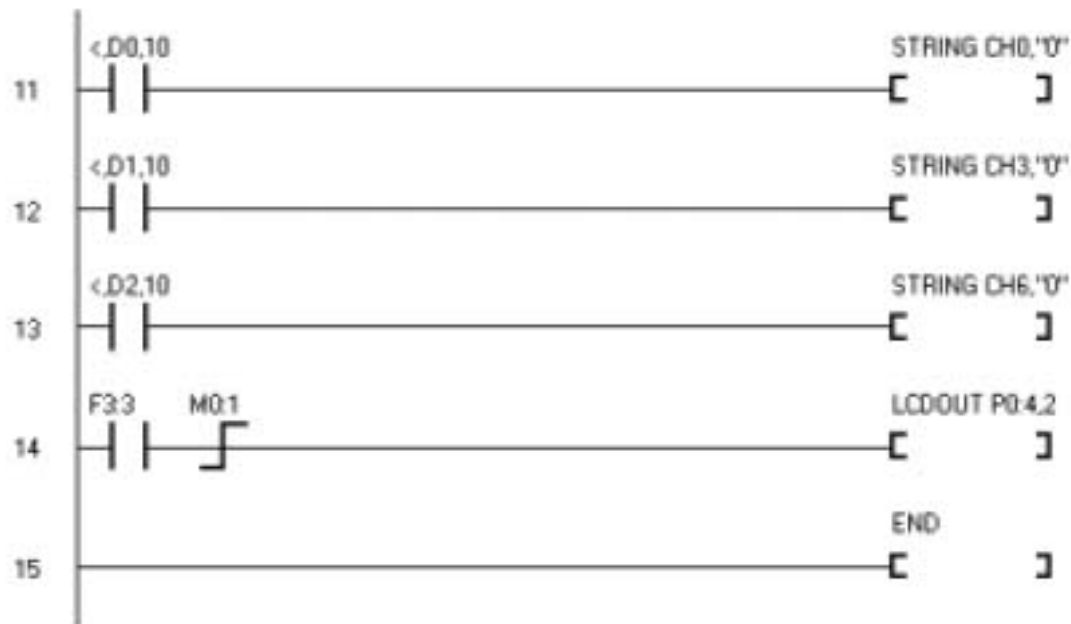
CLOCK Horloge affichage sur LCD

P1 : 0 fait un reset de l'horloge , utilisable avec TPC37 et 38 , Utilise les relais spéciaux F8 , F9, F10

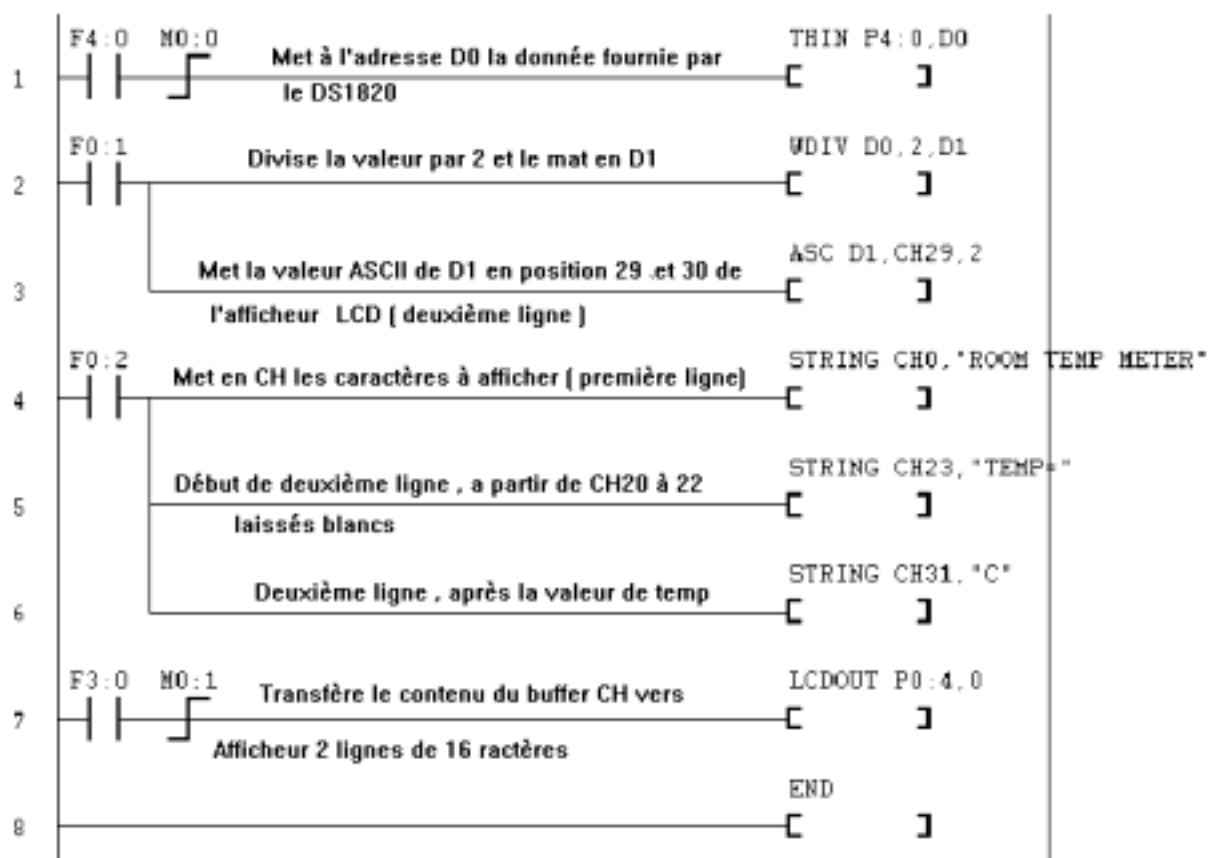
Port utilisé	Set-up I/O
P1:0	INPUT
P0:4	OUTPUT



Suite du programme : page suivante



Lecture de température avec un DS1820



Sommaire

Introduction -----	4
--------------------	---

Chapitre 1 Présentation des TinyPLC

Introduction à TinyPLC -----	6
Caractéristiques des TPC 3X -----	7
Brochage des TPC 3X -----	8
Connexion avec un PC -----	12
Liaison avec un ordinateur -----	13
Organisation mémoire de données -----	15
Relais et relais spéciaux -----	16

Chapitre 2 Bases des PLC

Un PLC c'est quoi ,structure générale-----	20
Types de relais -----	22
Chargement et exécution d'un programme LADDER -----	23

Chapitre 3 Instructions de base de TinyPLC

Tableau des instructions-----	28
Commandes de comparaisons -----	30
Précautions pour les entrées LADDER-----	31
LOAD , LOADN , AND , ANDN -----	32
OR , ORN , OUT , NOT , END -----	33
ANDS , ORS , SETOUT , RSTOUT -----	35
DF , DFN-----	35
MCS , MCSCLR -----	36
STEPSET -----	38
STEPOUT -----	39
TON , TAON , TOFF , TAOFF -----	41
CTU , CTD -----	42
Commandes de comparaisons de mots -----	44
Commandes de comparaisons de mots longs (double) -----	46

Chapitre 4 Commandes d'applications TinyPLC

Résumé des commandes d'applications -----	48
Explication des nombres traités par TinyPLC -----	52
Code ASCII -----	53
MOVE , WMOV -----	55
DWMOV -----	55

CMOVE , WCMOV , DWCMOV -----	56
WNEG , DWNEG -----	57
WXCHG , DWXCHG -----	58
FMOV , GMOV -----	59
WBCD , DWBCD -----	60
WBIN , DWBIN -----	61
WINC , DWINC , WDEC , DWDEC -----	62
WADD , DWADD -----	63
WSUB , DWSUB -----	64
WMUL , DWMUL -----	65
WDIV , DWDIV -----	66
BAND -----	67
WAND , DWAND -----	68
BOR -----	69
WOR , DWOR -----	70
BXOR -----	71
WXOR , DWXOR -----	72
WROL , DWROL , WROR , DWROR -----	73
WRCL , DWRCL -----	74
WRCR , DWRCR -----	75
BSHL , BSHR -----	76
WSHL , DWSHL -----	77
SEG -----	78
LCDCLS -----	79
LCDOUT -----	80
SGNOUT -----	82
STRING -----	84
HEX , DHEX -----	85
ASC , DASC -----	86
GOTO , LABEL -----	87
CALL , SBRT , RET -----	88
LOOP -----	89
KEYSCAN -----	90
OUTOFF -----	91
DIST -----	92
UNIT -----	93
ENCO , DECO -----	95
THIN -----	95

Chapitre 5 Comment utiliser MPGL2

Présentation de MPGL2 -----	98
Installation de MPGL2 -----	99
Messages d'erreurs de MPGL2 -----	101
Câble de téléchargement -----	102
Fenêtre de MPGL2 -----	103
Exemple de saisie -----	105
Initialisation du PLC et des entrées et sorties -----	106
Entrée de commentaire pour les symboles -----	107

Compilation , téléchargement -----	107
Exécution , moniteur mise au point -----	108
Chargement d'un programme source -----	109
Environnement de MGPL2 -----	109
Ouverture sauvegarde programme -----	110
Recherche ligne , symbole -----	110
Saisie de commentaires -----	111
Editer des symboles -----	111
Impression des LADDERS -----	113

Chapitre 6 Transmissions de données

Communication entre TinyPLV et PC -----	116
Communication standard -----	117
Liaison ASCII RS232C -----	117
Liaison ASCII RS485 -----	120
Liaison binaire RS232C (programmes Qbasic™ et Picbasic) -----	121

Chapitre 7 Installation et organisation

Entrées et sorties des TinyPLC -----	132
Convertisseur A/D -----	134
Convertisseur D/ZA -----	136
Compteur rapide 16 bits -----	136
Circuit de base C20R -----	138
Port entrée 24 v -----	139
Port de sortie à relais -----	140
Bus standard TPC -----	141
Port LCD -----	141
Port SGN -----	142
Schéma du C20R -----	142

Chapitre 8 Applications TinyPLC

Tooggle (inverse une sortie à chaque appui BP -----	144
Flicker , clignotant -----	144
Commutation de sorties en fonction résultat A/D -----	145
WMUL2 Résultat d'une multiplication sur LCD -----	146
SGN-SW Caractère sur afficheur 7 segments -----	146
LCD – AD Résultat d'une conversion sur LCD -----	147
SGN – CNT Résultat d'un compteur sur SGN -----	148
CLOCK Horloge sur LCD -----	149
Lecture de température d'un DS1820 sur LCD -----	150