



Analyste Programmeur en Automatisation, Robotique et Informatique Industrielle TS ARII

Module MF 1.2

Analyser un traitement numérique

Le Langage TPE

Patrick MONASSIER – année 2019-2020

Module 1.2

Analyser un traitement numérique

Compétences

- **Analyser un traitement numérique.**

Objectifs

- *Appliquer les systèmes de codage aux automatismes analogiques et numériques.*
- **Appliquer les systèmes de codage à la robotique.**

Contenu

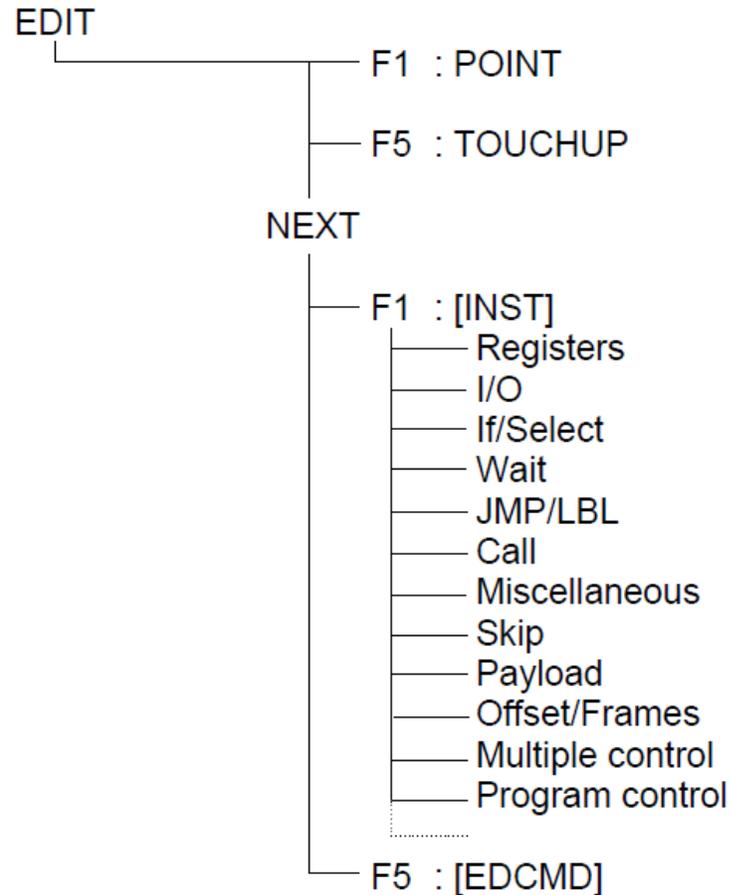
Codage des automatismes

- *Présentation des variables et objets utilisables en automatisation*
- *Elaboration d'un organigramme*
- *Construction d'un algorithme appliqué aux automatismes.*
- *Application sur les éditeurs dédiés (SCL pour Siemens et DFB pour Schneider)*
- *Développement des fonctions bibliothèques.*

Codage des robots

- *Introduction à la programmation d'un robot : architecture logicielle, contrôle des événements et des trajectoires*
- *Architecture d'une programmation appliquée aux robots, contrôle des trajectoires et repères*
- **Construction d'une programmation appliquée aux robots**
- **Elaboration d'une séquence robotique**
- **Exemples industriels**

Arborescence de l'éditeur



Instructions de variables

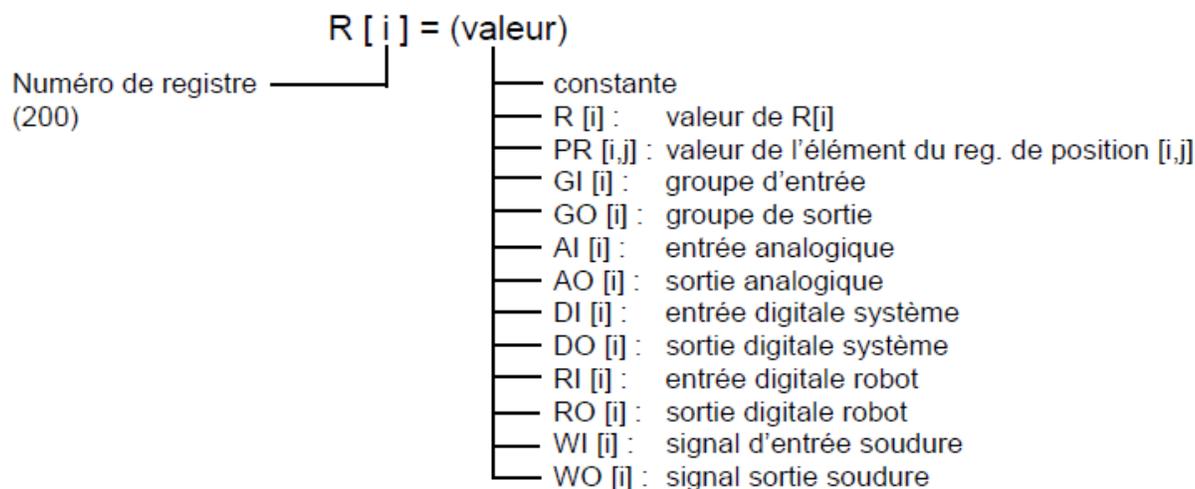
Les **variables disponibles** à l'utilisateur sont :

- Les registres : entier ou réel
- Les registres de position : points en coordonnées joint, points en coordonnées cartésiennes ou matrices.

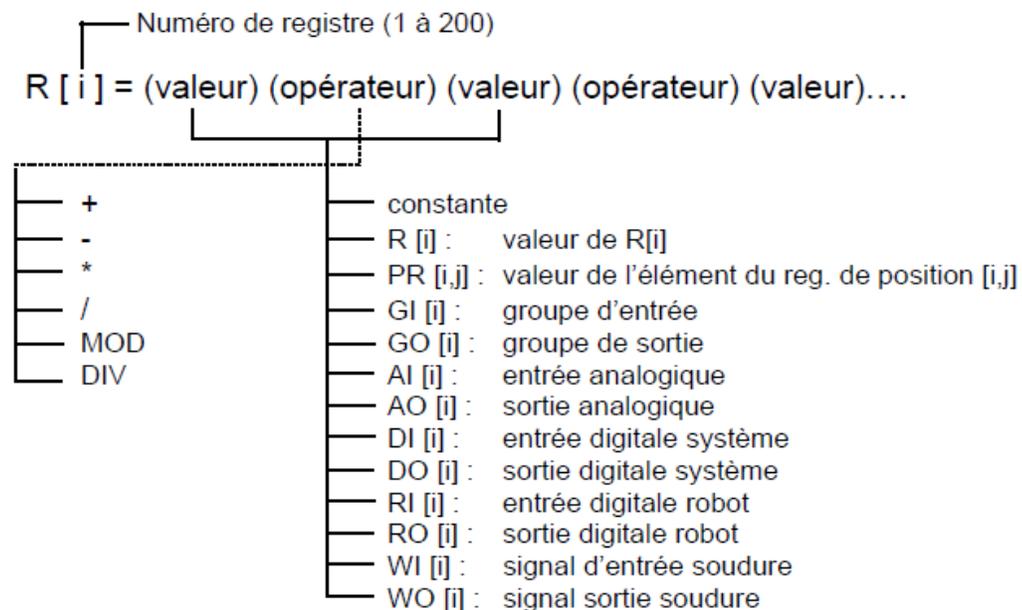
Ce sont des variables globales (tous les programmes ont accès à tous les registres et registres de position)

Les registres :

Une instruction de registre réalise une opération arithmétique sur les registres. Un registre est une variable contenant un entier ou un réel. 200 registres sont disponibles au maximum configurable à la demande.



Exemple : 1 : $R [1] = R [3]$
 2 : $R [R [4]] = AI [R [1]]$



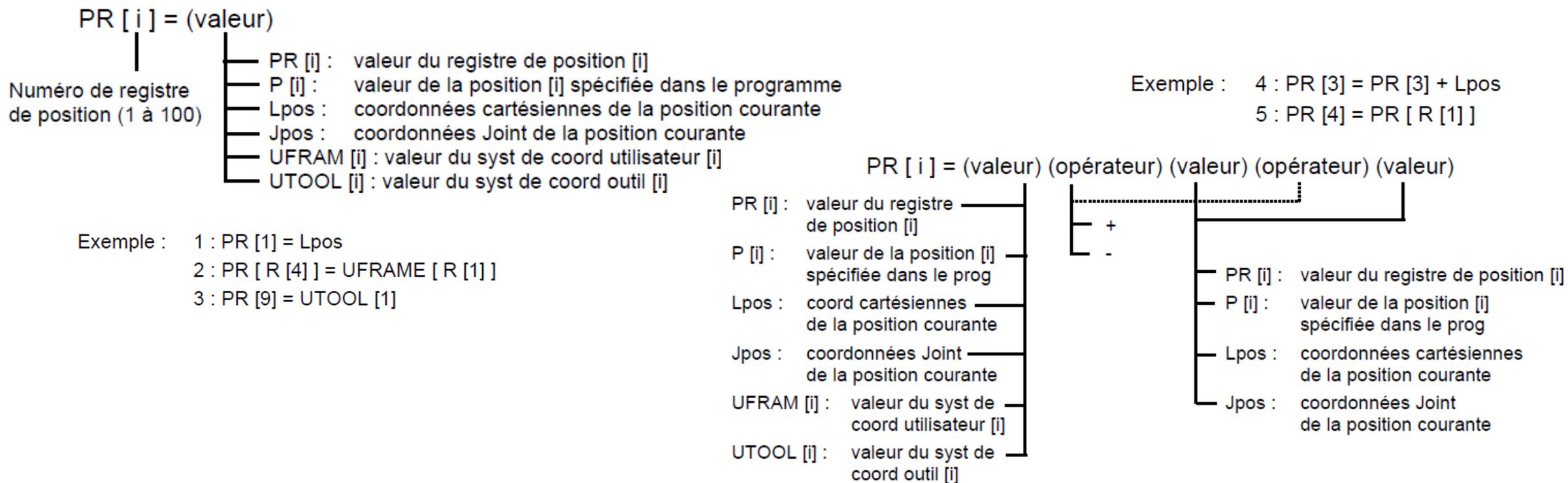
Exemple : 3 : $R [3 : \text{flag}] = DI [4] + PR [1 , 2]$
 4 : $R [R [4]] = R [1] + 1$

Opérations possibles : + - * / MOD DIV

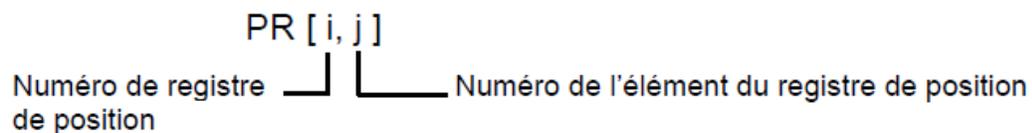
Les registres de position :

Un registre de position est une variable pour stocker des données de position (x,y,z,w,p,r).

Il en existe maximum 100 en R-J3 et R-J3i (configurable à la demande). Une instruction de registre de position exécute une opération arithmétique entre des registres de position. Une instruction registre de position peut charger des données de position, des sommes/différences de 2 valeurs, dans un registre de position spécifié.



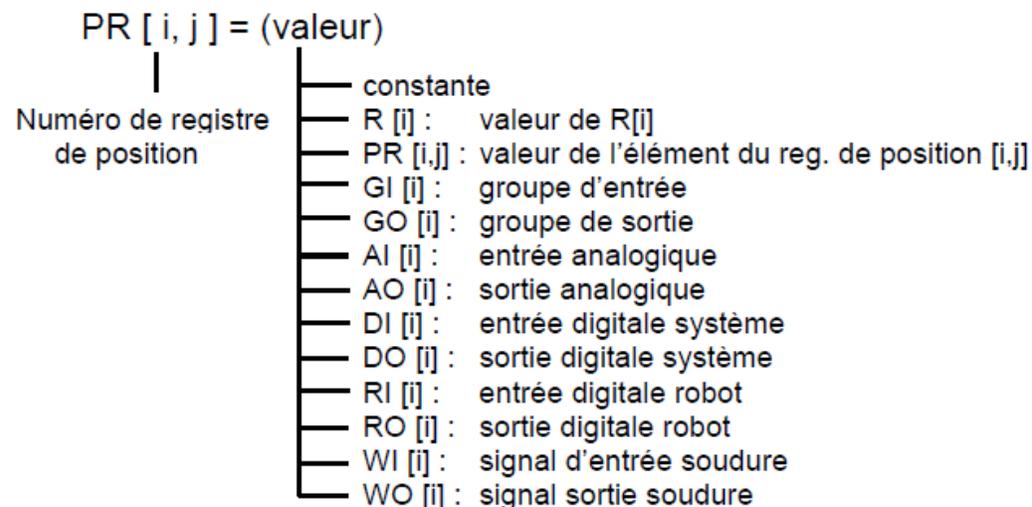
Une **instruction axes registre de position** exécute une opération arithmétique avec les éléments de registres de position. Le **i** de **PR[i,j]** représente un numéro de registre de position et le **j** représente le numéro de l'élément du registre de position. L'instruction axe registre de position peut charger la valeur d'un élément de donnée de position ou une somme, une différence, un produit ou un quotient de 2 valeurs dans un élément du registre de position spécifié. Cette instruction utilise le même format que l'instruction registre.



Système de coordonnées cartésien : Système de coordonnées Joint :

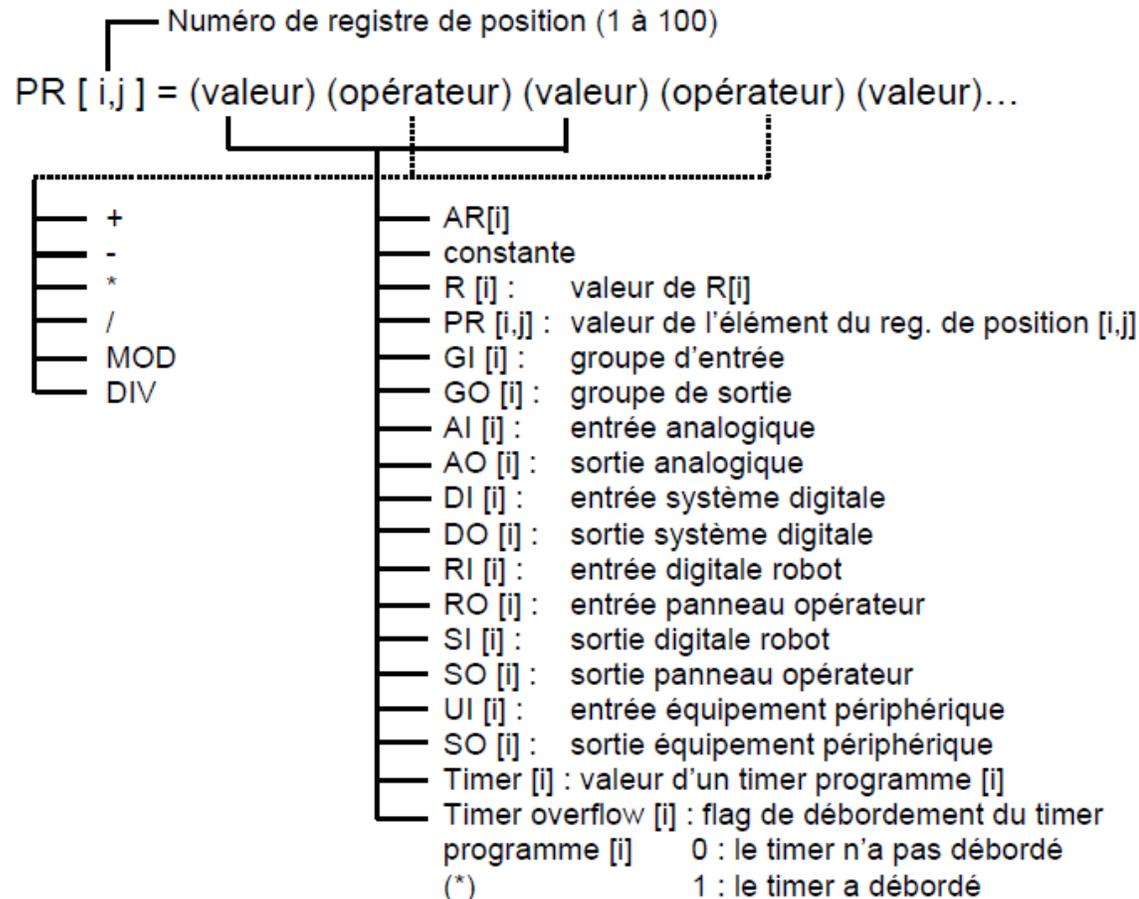
1 = X
2 = Y
3 = Z
4 = W
5 = P
6 = R

1 = J1
2 = J2
3 = J3
4 = J4
5 = J5
6 = J6
n = Jn



Exemple : 1 : PR [1, 2] = R [3]
2 : PR [4, 3] = 324,5

Instruction axes registre de position



* Le débordement du timer est effacé avec l'instruction timer [i] = reset

Exemple : 1 : PR [3, 5] = R [3] + DI [4]
 2 : PR [4, 3] = PR [1, 3] - 3.528

Opérations possibles : + - * / MOD DIV

Les sorties

Sorties digitales

DO[n] ou RO[n] = [valeur] → La [valeur] peut être : ON, OFF, une valeur 0 ou 1 d'un registre R[n].

DO[n] ou RO[n] = PULSE [valeur] → La [valeur] est un temps en seconde (de 0,11 à 25,01).

Sorties analogiques

AO[n] = [valeur] → La [valeur] est comprise entre 0 et 2047 par défaut.
0 correspond à 0 V
2047 correspond à 10V

Sorties groupées

GO[n] = [valeur] → La [valeur] est décimale et est limitée à 2^n pour n sorties groupées. La conversion en binaire est faite sur les sorties digitales groupées.

Les entrées

L'acquisition des entrées se fait par l'intermédiaire d'un registre

R[n] = DI[n] → R[n] contient 1 (pour ON) ou 0 (pour OFF).

$R [i] = RI [i]$
Numéro de registre (1 à 200) ———— Numéro de signal d'entrée robot

Exemple : 1 : R [1] = RI [1]
2 : R [R[3]] = RI [R[4]]

R[n] = RI[n] → R[n] contient 1 (pour ON) ou 0 (pour OFF).

R[n] = GI[n] → R[n] contient la valeur décimale correspondant au code binaire reçu sur les entrées digitales groupées.

$R [i] = GI [i]$
Numéro de registre (1 à 200) ———— Numéro de groupe de signaux d'entrée

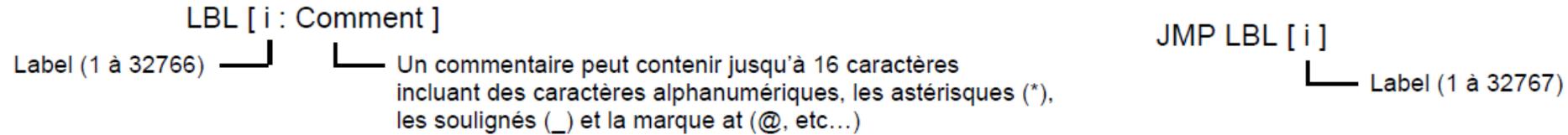
Exemple : 7 : R [1] = GI [1]
8 : R [R[3]] = GI [R[4]]

R[n] = AI[n] → R[n] contient une valeur entre 0 et 2000 ; correspondant à la valeur de la tension sur AI[n].

$R [i] = AI [i]$
Numéro de registre (1 à 200) ———— Numéro de signal d'entrée analogique

Exemple : 1 : R [1] = AI [1]
2 : R [R[3]] = AI [R[4]]

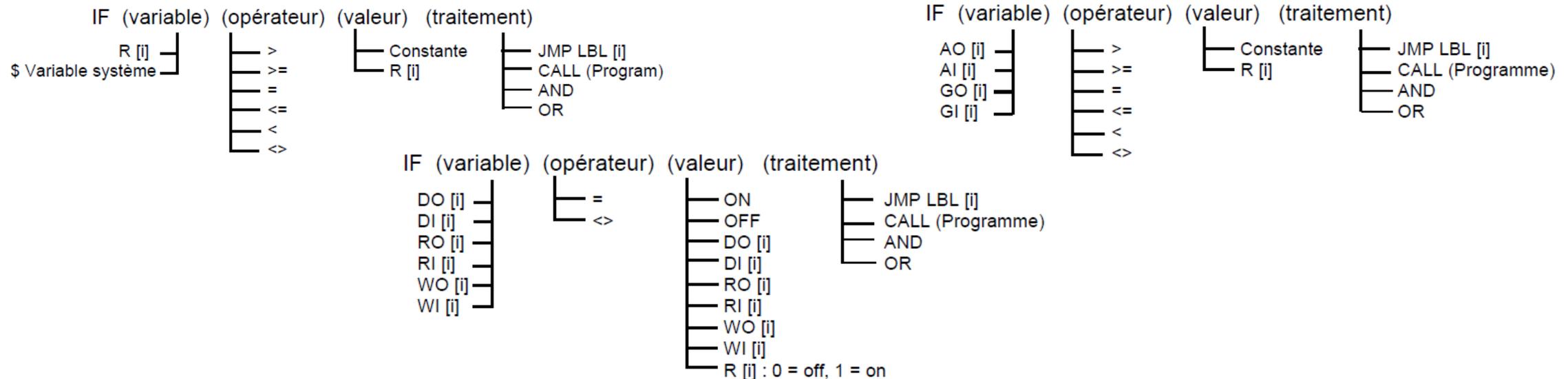
Label : **LBL[n : [commentaire]]**. Il peut être utilisé par les instructions de **saut** conditionnel ou inconditionnel (JMP LBL[n]).



Appel de programme : L'instruction « **CALL program** » permet de lancer un programme, avec ou sans paramètres

SKIP : L'instruction de saut conditionné spécifiée à l'avance une condition de saut

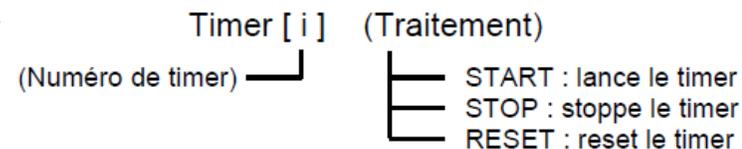
L'instruction IF :



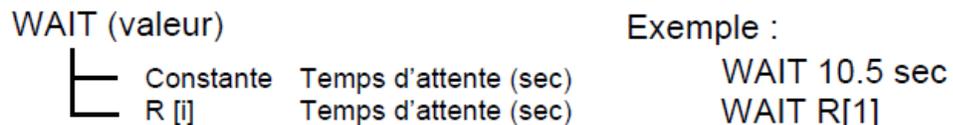
Instruction SELECT : Effectue un ou plusieurs sauts en fonction de la valeur d'un registre.

```
Exemple : 11 : SELECT R[1] = 1, JMP LBL [1]
          12 : = 2, JMP LBL [2]
          13 : = 3, JMP LBL [2]
          14 : = 4, JMP LBL [2]
          15 : ELSE, CALL SUB2
```

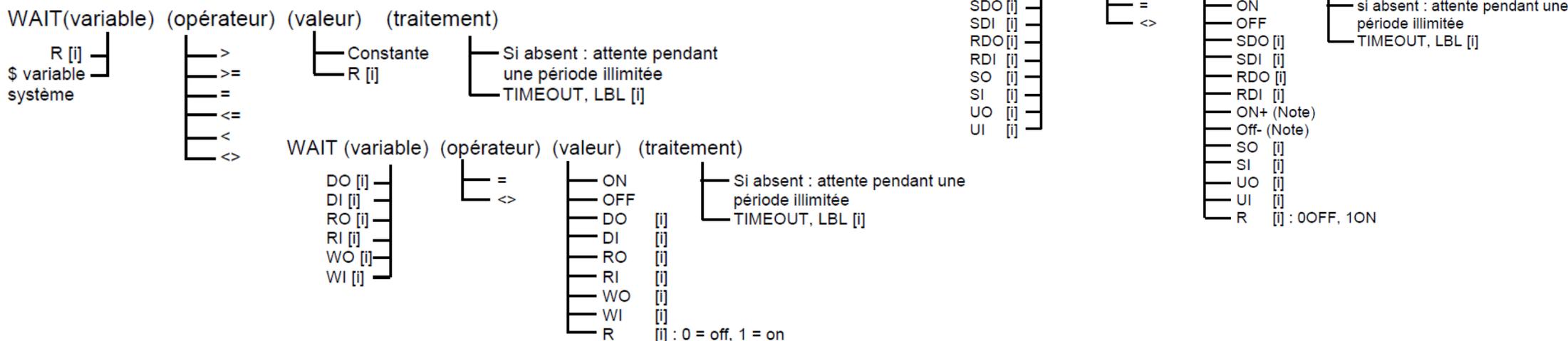
Instruction TIMER : L'instruction Timer démarre/arrête un timer programme. L'état de ce timer peut être visualisé dans l'écran des timers



Instructions d'attente WAIT : Retarde l'exécution du programme pendant un temps spécifié. La durée s'exprime en seconde.



WAIT [valeur 1] [opérateur] [valeur 2] [temps] : Retarde l'exécution du programme pendant un temps spécifié, jusqu'à ce que la condition soit vraie.



Instructions de repère : Un point est dépendant d'un (et **uniquement un**) repère utilisateur, et d'un (et **uniquement un**) repère outil !

Sélectionner les repères Utilisateur et Outil utilisés :

UFRAME_NUM = (valeur)
 ┌ R [i]
 └ Constante Numéro de repère utilisateur (0 à 9)

Exemple : 1 : UFRAME_NUM = 1
 2 : J P[1] 100% FINE
 3 : L P[2] 500mm/sec FINE
 4 : UFRAME_NUM = 2
 5 : L P[3] 500mm/sec FINE
 6 : L P[4] 500mm/sec FINE

UTOOL_NUM = (valeur)
 ┌ R [i]
 └ Constante Numéro de repère outil (0 à 9)

Sauvegarder un repère dans un registre de position

UFRAME [i] = (valeur)
 Numéro de repère utilisateur ┌ └ PR [i]
 (1 à 9)

UTOOL [i] = (valeur)
 Numéro de repère outil ┌ └ PR [i]
 (1 à 9)

OFFSET Instruction : L'instruction OFFSET CONDITION spécifie une condition de décalage définie à l'avance.

OFFSET CONDITION PR [i] (UFRAME [j])
Numéro de registre de position (1 à 100) Numéro de repère utilisateur (1 à 9)

Exemple :
1 : OFFSET CONDITION PR [R[1]]
2 : J P[1] 100% FINE
3 : L P[2] 500mm/sec FINE Offset

OFFSET Tools Instruction: L'instruction OFFSET TOOLS (décalage de l'outil) spécifie la condition de décalage à utiliser dans l'instruction offset de l'outil.

TOOL_OFFSET CONDITION PR [i] (UTOOL [j])
Numéro de registre de position (1 à 100) Numéro de repère outil (1 à 9)

Exemple :
1 : TOOL_OFFSET PR [1]
2 : J P[1] 100% FINE
3 : L P[2] 500mm/sec FINE Tool_Offset

Analyste Programmeur en Automatismes, Robotique et
Informatique Industrielle
TS ARII

Module MF 1.2

Analyser un traitement numérique

Le Langage TPE

Fin de Présentation