

TP2 Supervision - Mrabet		Pt	A	B	C	D	Note
I.	Création du process virtuel (1pt)						
1	Ajouter un bloc SIM sur votre programme, il simulera le fonctionnement d'un procédé réel. Donner lui un nom.	0,5	A				0,5
2	Procéder à son paramétrage en respectant les valeurs suivantes :	0,5	A				0,5
II.	Réglage de la boucle de régulation (7pts)						
1	Ajouter à votre programme un bloc PID afin de créer une régulation de votre procédé virtuel.	1	A				1
2	Régler la boucle de régulation utilisant la méthode par approches successives.	2	A				2
3	Enregistrer la réponse de la mesure X à un échelon de consigne W de 20%.	1	A				1
4	Mesurer le temps de réponse à $\pm 5\%$ , le premier dépassement, ainsi que l'erreur statique.	3	C				1,05
III.	Supervision	5	A				5
IV.	Alarme	5	A				5
V.	Boutons	2	A				2
		Note : 18,05/20					

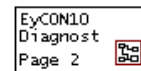
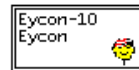
# TP2 Supervision

## I. Création du process virtuel

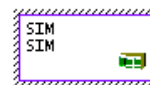
1/ Ajouter un bloc SIM sur votre programme, il simulera le fonctionnement d'un procédé réel. Donner lui un nom.

```
FILENAME:
DATE :
VERSION :

FUNCTION: Eycon-10 Standard Diagnostics
          With Database Header
```



```
!!!!!! IF NOT A LAYER DATABASE !!!!!
!!!!!! RENAME DIAGNOSTIC BLOCKS !!!!!
!!!!!! THEN DELETE THIS MESSAGE !!!!!
```



2/ Procéder à son paramétrage en respectant les valeurs suivantes

Block: SIM						
Comment		Connections				
<b>TagName</b>	<b>SIM</b>			<b>Link Name</b>	<b>SIM</b>	
<b>Type</b>	<b>SIM</b>			<b>DBase</b>	<b>&lt;local&gt;</b>	
				<b>Rate</b>	<b>0</b>	
<b>Mode</b>	<b>AUTO</b>			<b>Alarms</b>		
<b>Fallback</b>	<b>AUTO</b>			<b>NoiseMax</b>	<b>0.0</b>	<b>Eng2</b>
<b>PV</b>	<b>0.0</b>	<b>%</b>		<b>Lag1</b>	<b>10.00</b>	
				<b>Lag2</b>	<b>12.00</b>	
<b>Bias</b>	<b>0.0</b>	<b>%</b>		<b>TimeBase</b>	<b>Secs</b>	
<b>Track</b>	<b>0.0</b>	<b>%</b>				
				<b>Intgr</b>	<b>FALSE</b>	
<b>HR_PV</b>	<b>100.0</b>	<b>%</b>		<b>Invert</b>	<b>FALSE</b>	
<b>LR_PV</b>	<b>0.0</b>	<b>%</b>				
				<b>Init</b>	<b>TRUE</b>	
<b>OP</b>	<b>0.0</b>	<b>Eng2</b>		<b>SelfTrack</b>	<b>FALSE</b>	
<b>HR_OP</b>	<b>100.0</b>	<b>Eng2</b>				
<b>LR_OP</b>	<b>0.0</b>	<b>Eng2</b>				
<b>HL_OP</b>	<b>100.0</b>	<b>Eng2</b>				

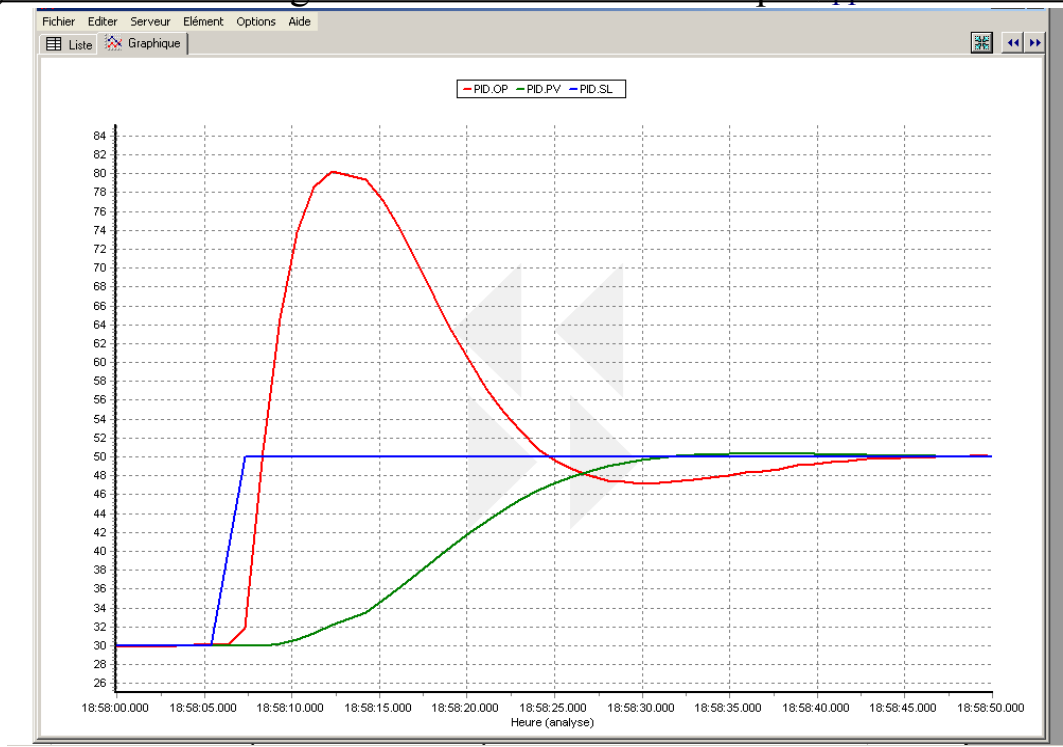
## II. Réglage de la boucle de régulation

1/Ajouter à votre programme un bloc PID afin de créer une régulation de votre procédé virtuel.

The screenshot displays the Eycon-10 software interface. The main window shows the 'Main (ROOT)' project with a file name 'Eycon-10', version '1.0', and function 'Standard Diagnostics With Database Header'. A message box indicates that if it's not a layer database, diagnostic blocks should be renamed or deleted. A diagram shows a 'PID' block connected to a 'SIM' block. The 'Properties' window on the right lists various parameters for the PID block, including 'Type' (Eycon-10), 'Version' (v5.5), and a list of variables like '3\_TERM', 'AN\_CONN', 'AN\_DATA', 'ANMS', 'DG\_CONN', 'DGMS', 'MAN\_STAT', 'MODE', 'PID', 'PID\_LINK', 'SETPOINT', and 'SIM'. The 'PID CONTROL BLOCK' description states: 'Generates a PID (Proportional/Integral/Derivative) control output OP, from a resultant setpoint SP & process variable input PV.'

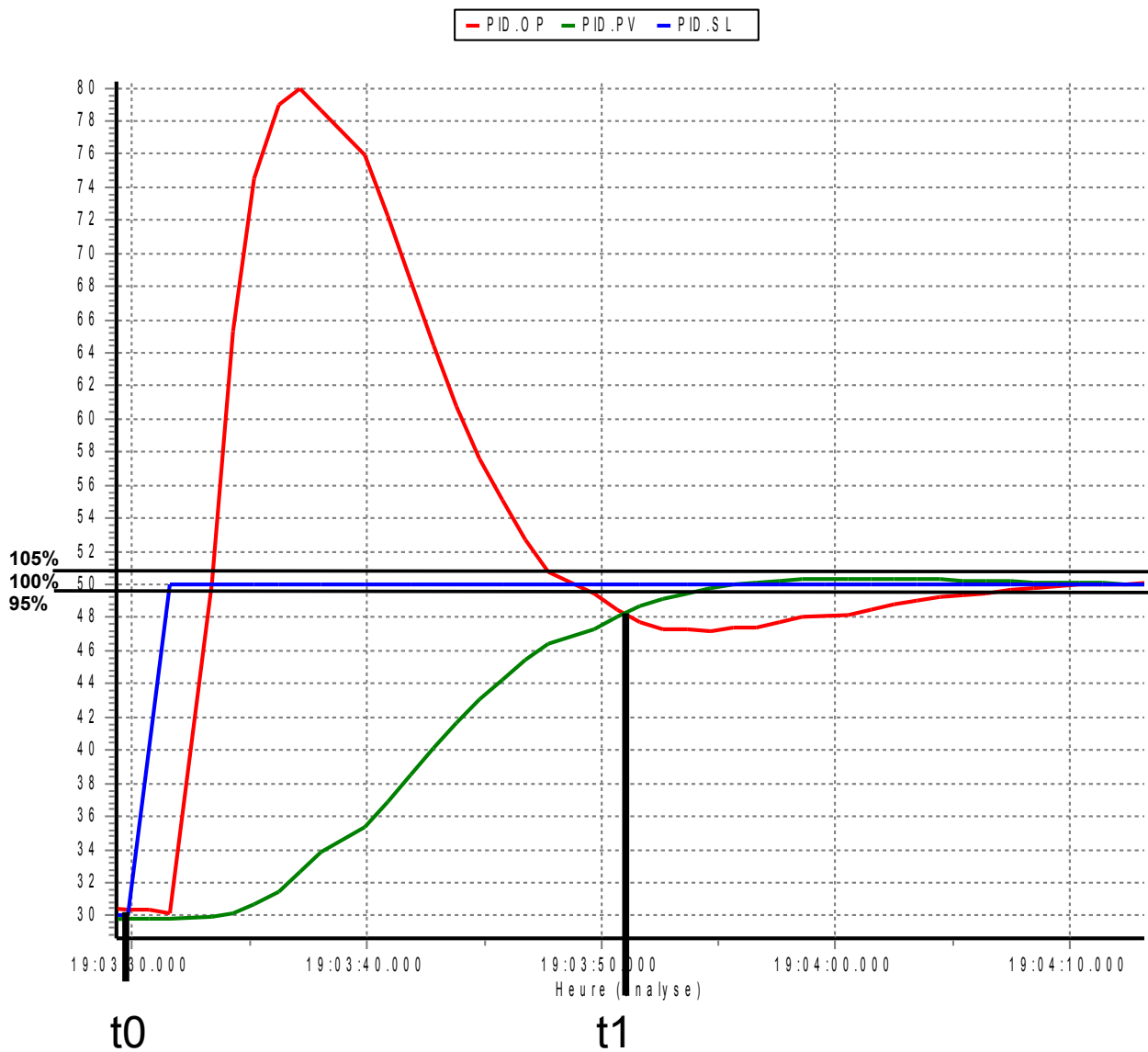
Tag/Name	PID	DBase	Rate	Alarms	TimeBase	Options	SelfMode	ModeSel	ModeAct	FF_PID	FB_OP
Type	PID										
Mode	AUTO										
FallBack	AUTO										
PV	0.0	%									
SP	0.0	%									
OP	0.0	%									
SL	0.0	%									
TrimSP	0.0	%									
RemoteSP	0.0	%									
Track	0.0	%									
HR_SP	100.0	%									
LR_SP	0.0	%									
HL_SP	100.0	%									
LL_SP	0.0	%									
HR_OP	100.0	%									
LR_OP	0.0	%									
HL_OP	100.0	%									
LL_OP	0.0	%									

2/ Régler la boucle de régulation utilisant la méthode par approches successives.



$X_p=10$     $T_i=10s$     $T_d=3s$

3/ Enregistrer la réponse de la mesure X à un échelon de consigne W de 20%



$T_0 = 19:03:29 = 0s$

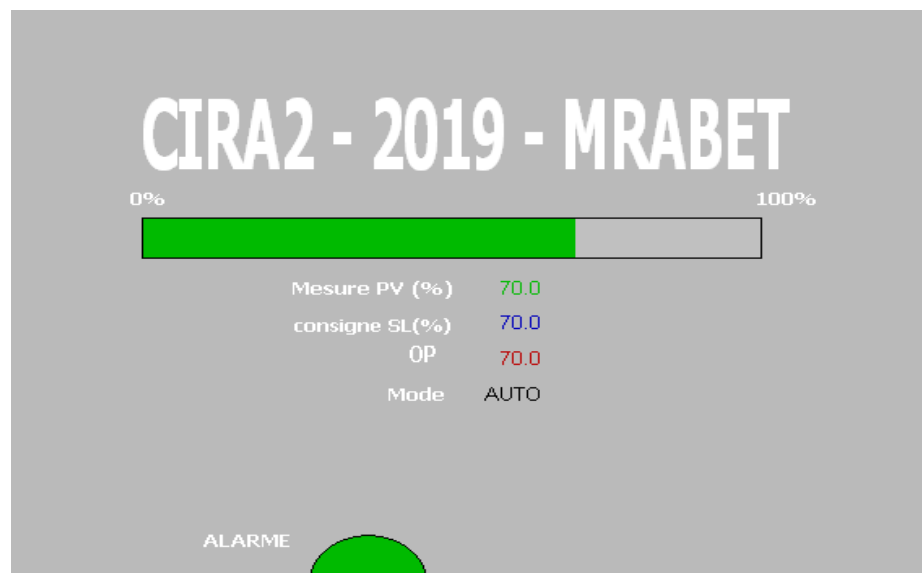
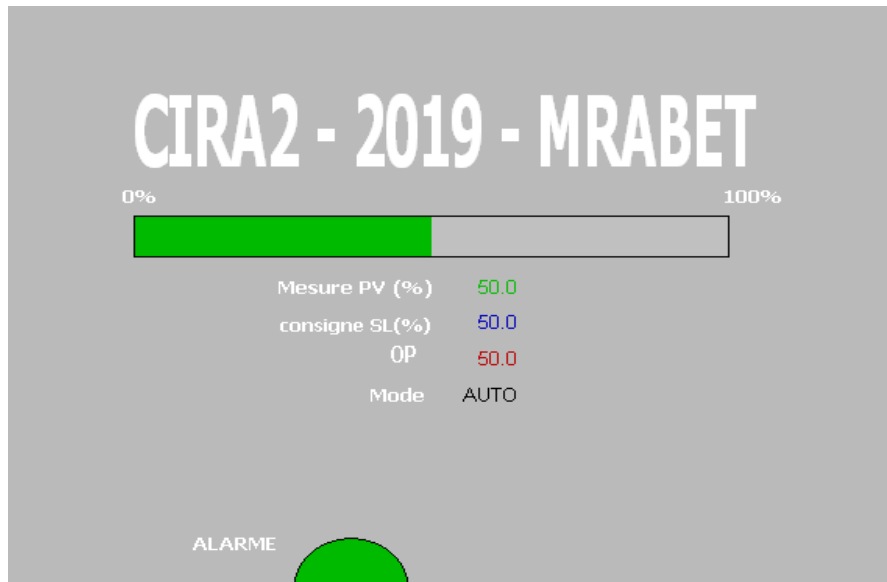
$t_1 = 19:03:50 = 21s$

4/ Mesurer le temps de réponse à  $\pm 5\%$ , le premier dépassement, ainsi que l'erreur statique.

$E_s = t_1 - t_0 = 21 - 0 = 0\%$  Donc il n'y a pas de dépassement

### III. Supervision

1/ Réaliser la programmation du superviseur en respectant le synopsis ci-dessous. On devra pouvoir contrôler la commande, la consigne et le mode de fonctionnement par l'intermédiaire d'Intouch. La mesure s'affichera en temps réel.

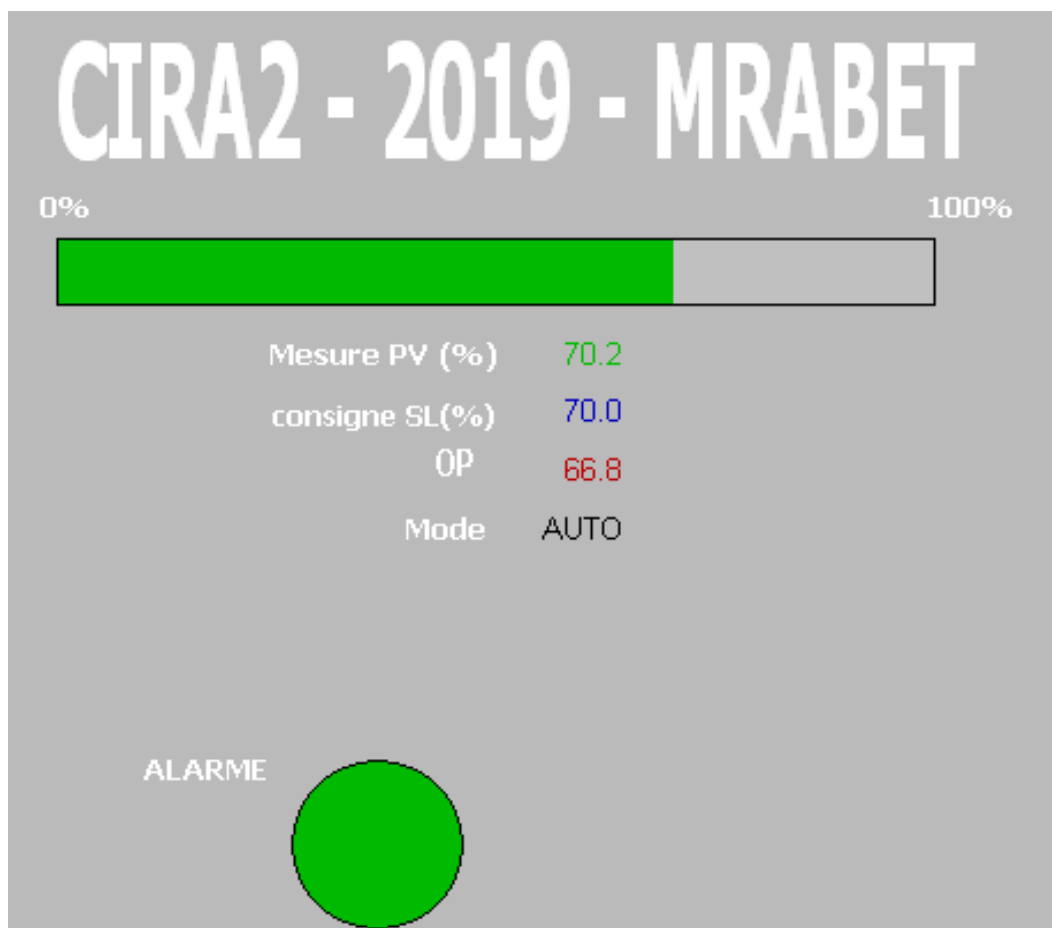


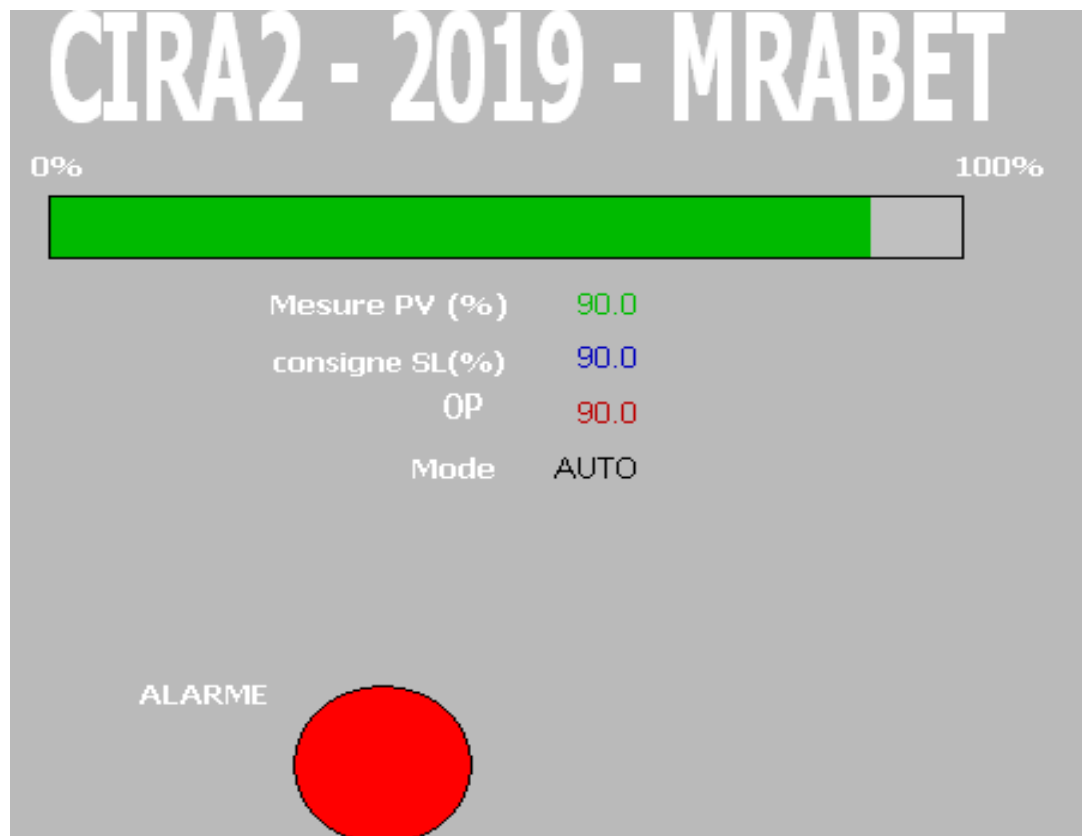
## IV. Alarme

1/ Modifier votre synopsis pour y ajouter un voyant d'alarme haute

Alarms		
HAA	80.0	%
LAA	0.0	%
HDA	100.0	%
LDA	100.0	%

Alarms	Value	Priority
Software		1
HighAbs	In Ack	2
LowAbs		0
HighDev		0
LowDev		0
Combined	In Ack	2





## V. Boutons

1/ Créer deux boutons, un pour passer le régulateur en mode AUTO, un autre pour passer le régulateur en mode MANU. Un voyant sur chaque bouton indiquera le mode sélectionné.

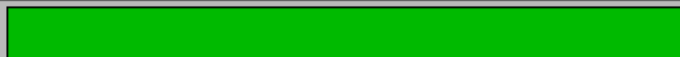
Bouton:Mode Auto

# CTRA2 - 2019 - MRABET

Object type: Button

Prev Link Next Link

OK Cancel



Mesure PV (%) @@@

consigne SL(%) @@@

OP @@@

Mode @@@



ALARME



Touch -> Action Script

File Edit Insert Help

Key equivalent

☐ Ctrl ☐ Shift Key... None

Condition Type: On Left Click/Key Down

Scripts used: 1

```
LINData.PID.SelMode.SelAuto = 1;  
LINData.PID.SelMode.SelAuto = 0;
```

IF ELSE AND < <= == <> >= >  
THEN ELSE IF OR = + - \* / ;  
ENDIF NOT

Clear All Clear OK Cancel

Convert Validate

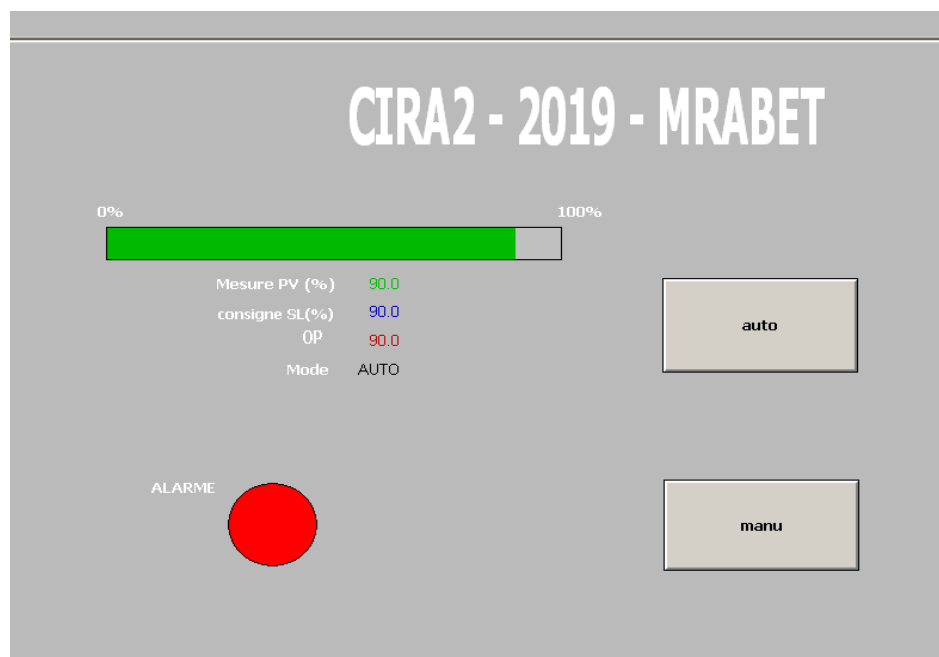
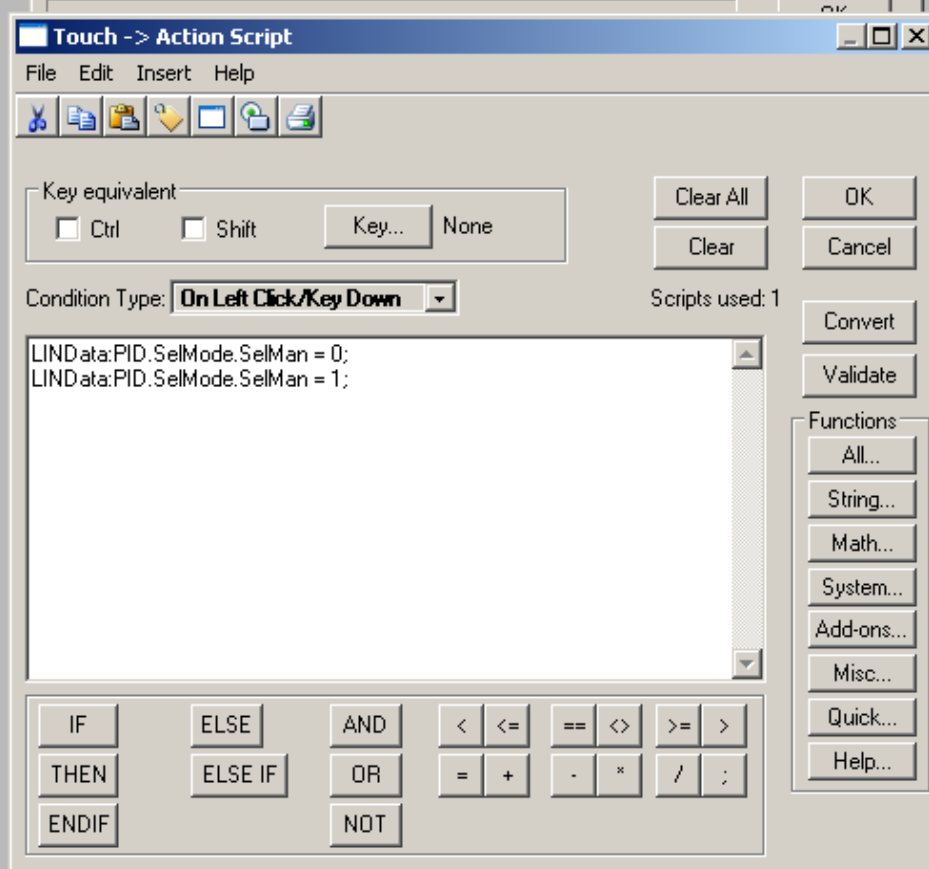
Functions

All... String... Math... System... Add-ons... Misc... Quick... Help...

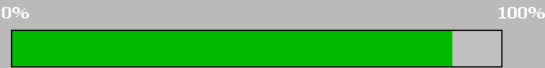
Bouton:Mode Manuel



# CTRA2 - 2019 - MRABET



# CIRA2 - 2019 - MRABET



Mesure PV (%) 90.0  
consigne SL(%) 90.0  
OP 90.0  
Mode MANUAL

auto

ALARME



manu