

Acquisition RedPitaya

Documentation generale

Cablage des detecteurs						
detecteur				numeriseur		
nom	pos.	capteur	reglage	pos.	nom	ressource
sphere	1	signal light	(mesure) HV1 led (mesure) HV2	1	AdcRp	ADC1 DAC-A E1-3 ADC2 DAC-B

Modeles de detecteurs			
nom	capteur	reglage	type
ball	signal	--	
rod	sensor	--	
	corrector	--	
1voie	signal	--	
2voies	signal	HV1 led	polar relais
	light	HV2	polar
4voies	near	--	
	far	--	
	single	--	
	light	--	
2beta	bille	--	
	sipm	--	

Modeles de numeriseurs

nom	ressource	categorie	valeurs	adrs	masque
AdcRp	E1-3	divers	key: off/on	00:00	1
	E1-4	divers	key: off/on	00:01	1
	E1-5	divers	key: off/on	00:02	1
	E1-6	divers	key: off/on	00:03	1
	E1-7	divers	key: off/on	00:04	1
	E1-8	divers	key: off/on	00:05	1
	E1-9	divers	key: off/on	00:06	1
	E1-10	divers	key: off/on	00:07	1
	E1-11	divers	key: off/on	00:08	1
	E1-12	divers	key: off/on	00:09	1
	E1-13	divers	key: off/on	00:10	1
	E1-14	divers	key: off/on	00:11	1
	E1-15	divers	key: off/on	00:12	1
	E1-16	divers	key: off/on	00:13	1
	DAC-A	auxiliaire	float: 1.06 .. -1.03 Volts	00:00	3FFF
	DAC-B	auxiliaire	float: 1.07 .. -1.04 Volts	00:16	3FFF
AdcCali	power-v1	divers	key: on/OFF	05:00	1
	power-v2	divers	key: on/OFF	05:04	1
	power-v3	divers	key: on/OFF	05:08	1
	power-v4	divers	key: on/OFF	05:12	1
	dth-v1	divers	key: non/active	05:01	1
	dth-v2	divers	key: non/active	05:01	5
	dth-v3	divers	key: non/active	05:01	9
	dth-v4	divers	key: non/active	05:01	13
	gain-v1	gain	key: 1/1.5	05:02	1
	gain-v2	gain	key: 1/1.5	05:06	1
	gain-v3	gain	key: 1/1.5	05:10	1
	gain-v4	gain	key: 1/1.5	05:14	1
	random-v1	divers	key: non/active	05:03	1
	random-v2	divers	key: non/active	05:07	1
	random-v3	divers	key: non/active	05:11	1
	random-v4	divers	key: non/active	05:15	1
	sortie	divers	int: 0 .. 65535	06	FFFF
AdcBO	--				
Sy127	HV1	polar	int: 0 .. 4000 V	01:50	0
	HV2	polar	int: -1000 .. 1000 V	02:10	0

Variables de l'automate				
nom	type	code	etat	valeur

Syntaxe des scripts SAMBA:

```

<instr_globale>
'Detecteur'      <detec>      { ':' <consigne_det> | '{' // (<consigne_det> //) '}' }
'Numeriseur'     <numer>     { ':' <consigne_num> | '{' // (<consigne_num> //) '}' }
'TypeDetecteur'  <type_detec> { ':' <consigne_det> | '{' // (<consigne_det> //) '}' }
'TypeNumeriseur' <type_numer> { ':' <consigne_num> | '{' // (<consigne_num> //) '}' }
'Media'          <media>     { ':' <message> | '{' // (<message> //) '}' }
'Var'            <nom>       ':' <texte>

<detec>      := { <nom> | '<' <type_detec> '>' | 'tous' } -- pour <type_detec>, 'TypeDetecteur' est utilise
<numer>      := { <nom> | '<' <type_numer> '>' | 'tous' } -- pour <type_numer>, 'TypeNumeriseur' est utilise
-- pour 'TypeDetecteur tous', 'Detecteur tous' est utilise
-- pour 'TypeNumeriseur tous', 'Numeriseur tous' est utilise

<consigne_det> := {
    <reglage>          <affect> <valeur_user>
    | <bb>'.'<ressource> <affect> { <valeur_user> | <hexa> }
    | <bb>'.'<hexa>      <affect> <entier>
    | <instr_globale>
}
-- <hexa>: sous-adresse ressource

<consigne_num> := {
    acces          <affect> { 'ouvert' | 'ferme' | 'oui' | 'non' }
    | <ressource>   <affect> { <valeur_user> | <hexa> }
    | <hexa>        <affect> <entier>
    | 'RAZ_DAC'
    | 'load_DAC'
    | <instr_globale>
}
-- <hexa>: sous-adresse ressource

<instr_globale> := {
    'exec'          [<affect>] <fichier_script>
    | 'boucle'       [<affect>] { <nb> | 'indefinie' }
    | 'recommence'
    | 'wait'         [<affect>] <secondes>
    | 'acquisition'  [<affect>] <secondes>
    | 'stream'       [<affect>] <secondes>
    | 'compensation'
    | 'partition'
    | 'sources'      [<affect>] { 'hs' | { 'actives' | 'oui' | 'on' | 'yes' } }
    | 'trigger'      [<affect>] { 'suspendu' | { 'actif' | 'oui' | 'on' | 'yes' } }
    | 'mode-regen'   [<affect>] { 'oui' | 'non' }
    | 'regul_clock'  [<affect>] <D3>
}

-- ou 'loop' et 'infinite'
-- ou 'endloop'

<bb>      := { <nom_numeriseur> | 'numeriseur('<nom_capteur>')' }
<ressource> := { <nom> | '<' <categorie> '>' }
<reglage>   := { <nom> | '<' <cmde> '>' }
<affect>     := {
    '=' -- memorise
    | ':' -- direct (non memorise par SAMBA si consigne det ou num)
    | '++' [ { '<' | '>' } { <limite> | 'standard' | 'consigne' } ] -- increment (memorise par SAMBA
    | '--' [ { '<' | '>' } { <limite> | 'standard' | 'consigne' } ] -- decrement (memorise par SAMBA

```

```

      | '=' [ { '<' | '>' } { <limite> | 'standard' | 'consigne' } ] -- facteur multiplicatif (memori
      | '/' [ { '<' | '>' } { <limite> | 'standard' | 'consigne' } ] -- division (memorise par SAMBA)
<valeur_user> := { <texte> | 'standard' | 'consigne' }
<entier>       := { <decimal> | <hexa> }
<hexa>        := '0x'<hexadecimal>

```