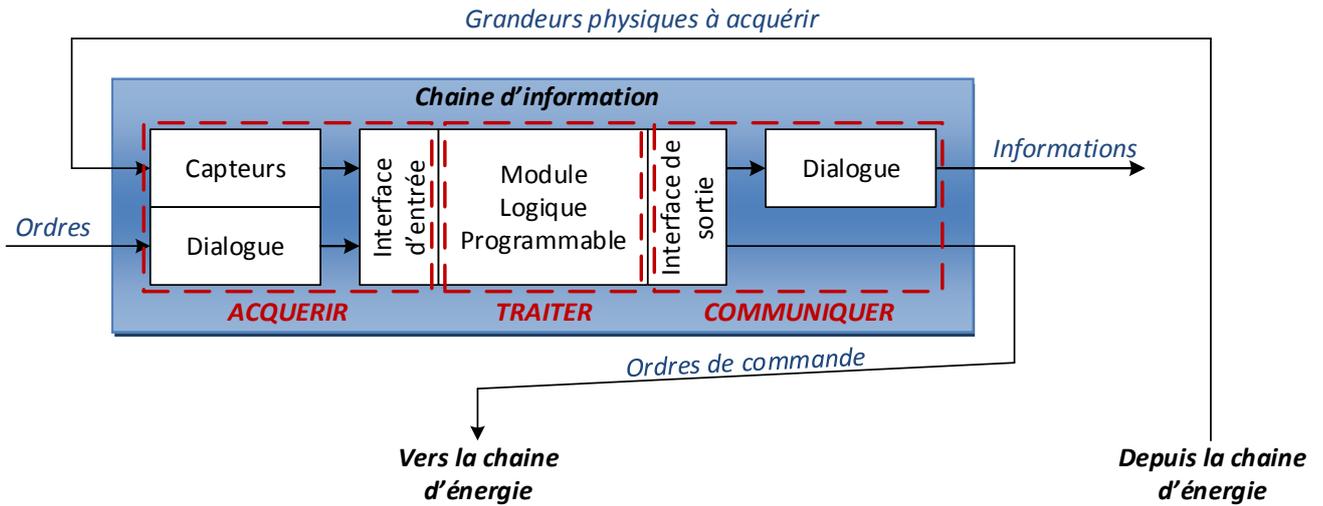


# MODULES LOGIQUES PROGRAMMABLES

## 1. Structure fonctionnelle de la chaine d'information

Les principales catégories de constituants assurant les différentes fonctions sont les suivantes :



**Fonction acquérir :**

- Ordres : .....
- Capteurs : .....
- .....



**Fonction traiter :**

- Logique câblée : .....
- .....
- Logique programmée : .....
- .....



**Fonction communiquer :**

- Ordres de commande : .....
- Informations : .....



**Comme sur toute chaine d'information, les signaux peuvent être :**

- .....
- .....
- .....

## 2. L'automate programmable ou PLC « Programmable Logic Controller »

### 2.1. Définition d'un automate programmable

C'est un appareil électronique qui comporte une mémoire, programmable par un utilisateur à l'aide d'un langage adapté, pour le stockage des instructions composant les fonctions d'automatisme comme par exemple :

- Logique séquentielle et combinatoire,
- Temporisation, comptage, décomptage, comparaison,
- Calcul arithmétique,
- Réglage, asservissement, régulation, etc...

.....

.....

.....



ZELIO de chez Schneider



EM4 de chez Crouzet



S7-200 de chez Siemens

### 2.2. Avantages et inconvénients

Les automates présentent de nombreux intérêts :

- .....  
.....  
..... (poussière environnante, perturbations électromagnétiques, vibrations des supports, variations de température...).
- .....  
..... Avec de plus une exécution séquentielle cyclique sans modification de mémoire, ils permettent d'assurer un temps d'exécution minimal, respectant un déterminisme temporel et logique, garantissant un temps réel.

Ils restent à l'heure actuelle les seules plateformes d'exécution considérées comme fiables en milieu industriel (avec les ordinateurs industriels) et pour certaines applications. De plus ils nécessitent la maîtrise de langages spécifiques qui reprennent dans leur forme la logique d'exécution interne de l'automate. Ces langages apparaissent toutefois à beaucoup d'utilisateurs plus accessibles et plus visuels que les langages informatiques classiques.

### 2.3. Langages de programmation pour automate (Norme IEC 1131-3)

La norme définit cinq langages qui peuvent être utilisés pour la programmation des automates. Ces cinq langages sont :

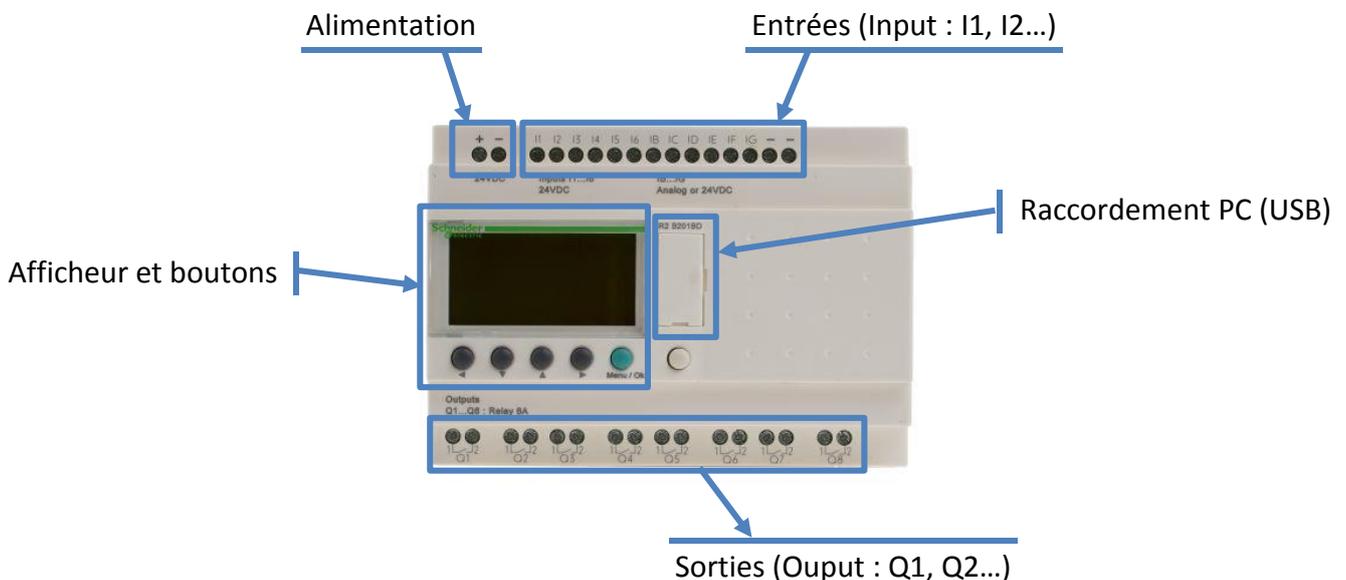
- ..... ce langage graphique est essentiellement dédié à la programmation d'équations logiques (vrai/faux).
- ..... ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.
- ..... ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions logiques. Il permet de manipuler tous les types de variables.
- ..... ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels. Il se rapproche du GRAFCET (Graphe Fonctionnel de Commande des Étapes et Transitions), mode de représentation et d'analyse d'un automatisme.
- ..... ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

### 3. Les modules Zelio Logic



Les modules Zelio Logic de chez Schneider regroupent les fonctions les plus courantes des modules logiques programmables. De plus, le logiciel de programmation est gratuit et regroupe les principaux langages de programmations.

#### 3.1. Présentation des modules



## 3.2. Raccordement du module

### 3.2.1. Généralités

Conformément à la structure de la chaîne d'information, sont raccordés au module :

**Sur les entrées**, tout élément transmettant :

- des ordres en provenance de l'opérateur (boutons poussoirs, claviers, interfaces tactiles...),
- des informations en provenance du système lui-même (capteurs de position, détecteurs...).

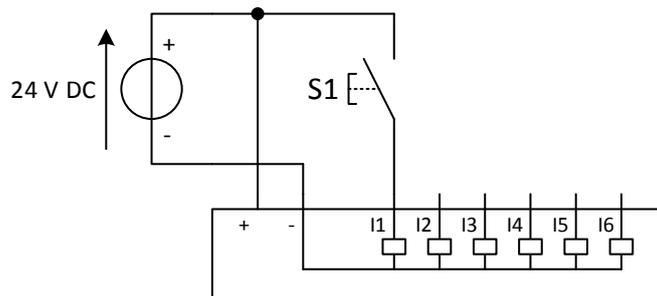
**Sur les sorties**, tout élément transmettant :

- des informations en direction de l'opérateur (voyants, texte sur un écran...),
- des ordres pour piloter la chaîne d'énergie (contacteurs, distributeurs...).

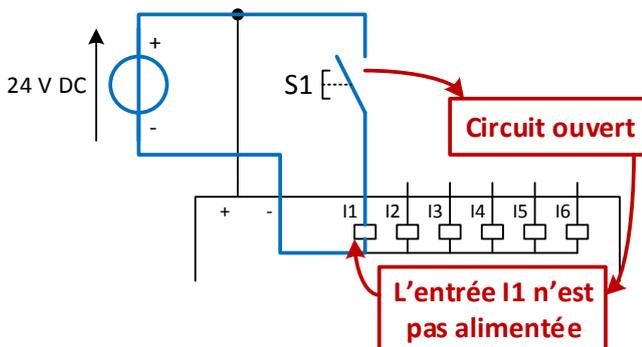
Ces raccordements sont recensés dans une table d'adressage qui donne la correspondance entre les éléments physique du schéma et les adresses de l'automate.

### 3.2.2. Raccordement des entrées TOR

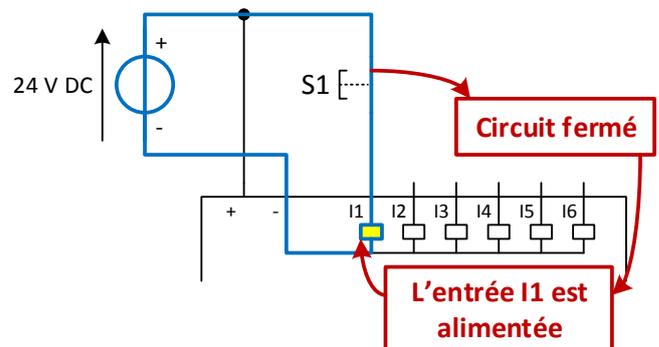
Soit l'exemple d'un bouton-poussoir S1 raccordé à l'entrée n°1 d'un module Zelio fonctionnant en 24 V DC :



- Si le bouton **S1** est au repos :



- Si le bouton **S1** est actionné :

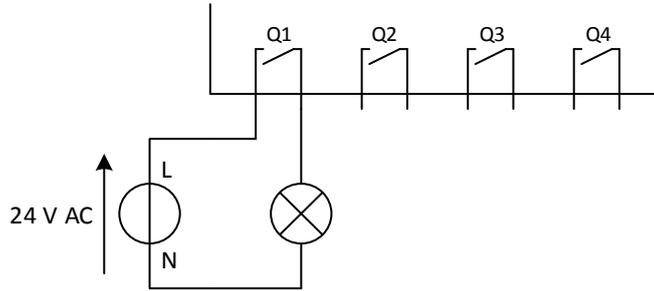


La table d'adressage de ce schéma peut être :

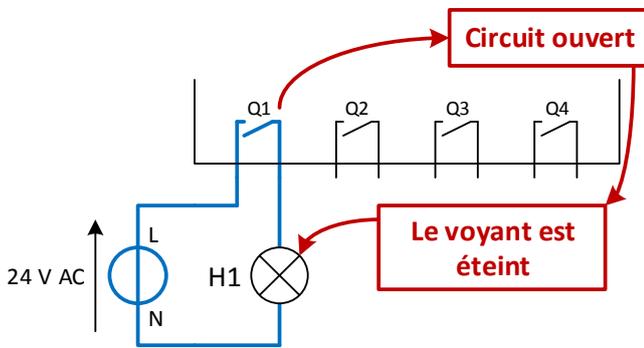
Repère	Description	Adresse
S1	Bouton-poussoir	I1

### 3.2.3. Raccordement des sorties TOR

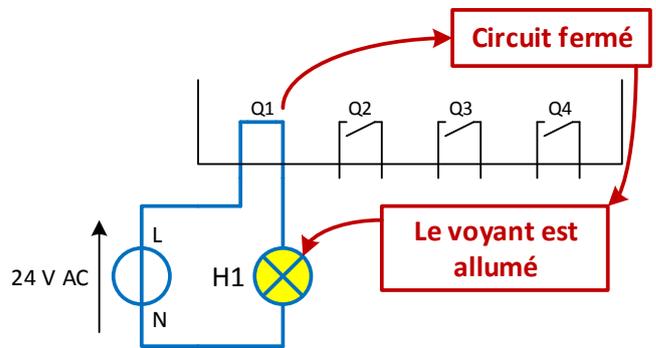
Soit l'exemple d'un voyant H1 fonctionnant en 24 V AC raccordé à la sortie n°1 d'un module Zelio :



▪ Si la sortie Q1 est au repos :



▪ Si la sortie Q1 est activée :



La table d'adressage de ce schéma peut être :

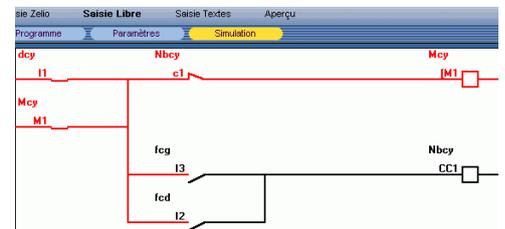
Repère	Description	Adresse
H1	Voyant	Q1

### 3.3. Le logiciel ZelioSoft

Les principaux langages de programmation de ce logiciel sont :

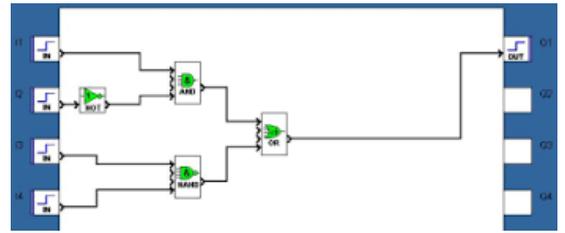
#### 3.3.1. Langage LD

Le langage LD (ladder diagram) est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données logiques, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.



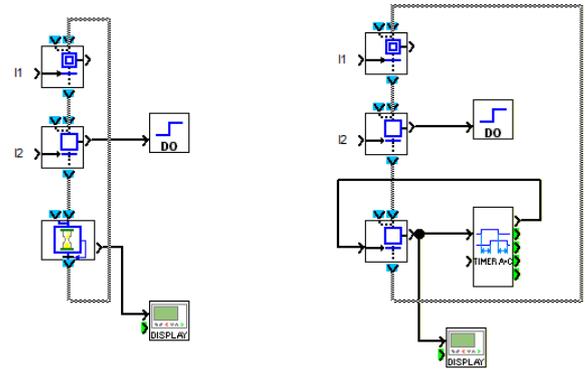
### 3.3.2. Langage FBD

Le langage FBD (Function Block Diagram) est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels issus de la logique combinatoire.



### 3.3.3. Langage SFC

Le langage SFC (Sequential Function Chart) est un langage graphique utilisé pour décrire des opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec le langage FBD.



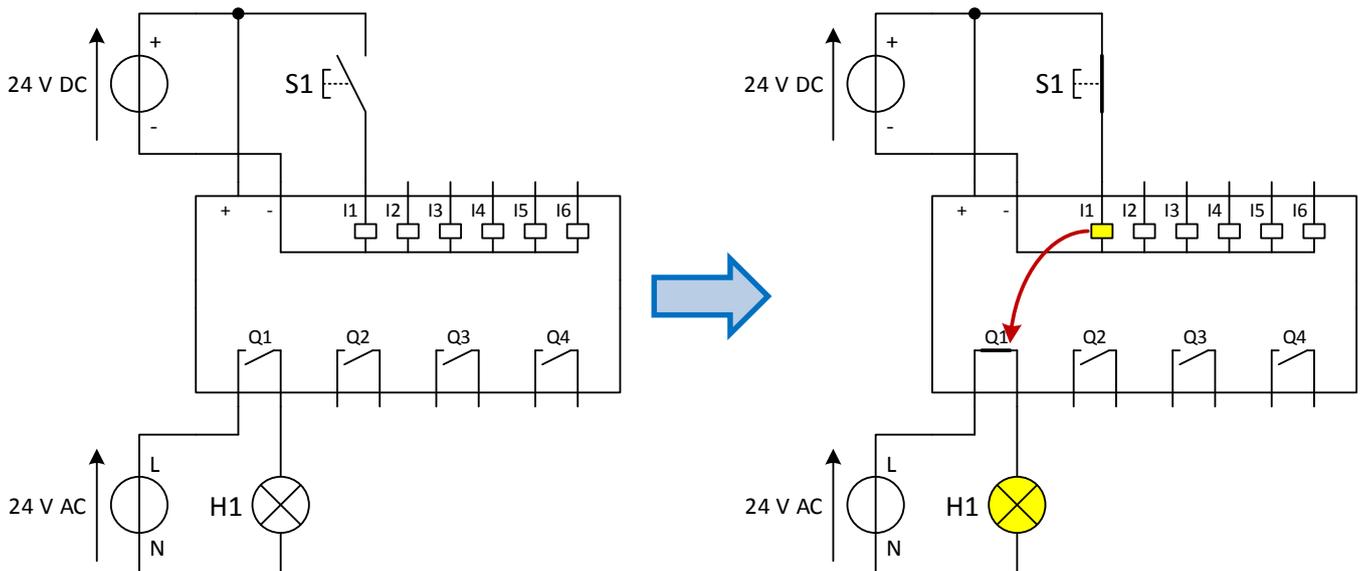
**Les principales règles graphiques sont :**

- un programme SFC doit contenir au moins une étape initiale.
- une étape ne peut pas être suivie d'une autre étape.
- une transition ne peut pas être suivie d'une autre transition.

### 3.3.4. Exemple de programmes

Soit l'exemple d'un bouton-poussoir S1 raccordé à l'entrée n°1 d'un module Zelio d'un voyant H1 raccordé à la sortie n°1.

On souhaite que le voyant soit allumé quand le bouton est activé. Autrement dit, la sortie Q1 doit être activée quand l'entrée I1 est alimentée :



▪ Programme en LD :

S1 est au repos → I1 n'est pas alimentée → Q1 est au repos → H1 est éteint :

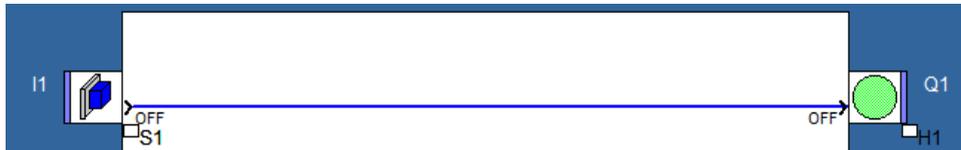


S1 est actionné → I1 est alimentée → Q1 est activé → le voyant est allumé :

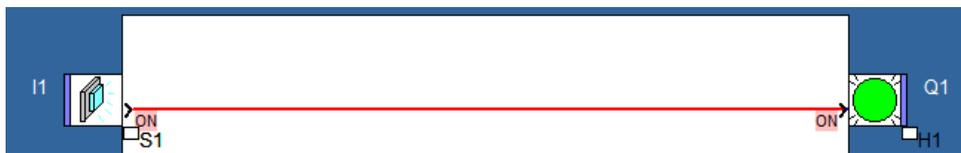


▪ Programme en FBD :

S1 est au repos → I1 n'est pas alimentée → Q1 est au repos → H1 est éteint :

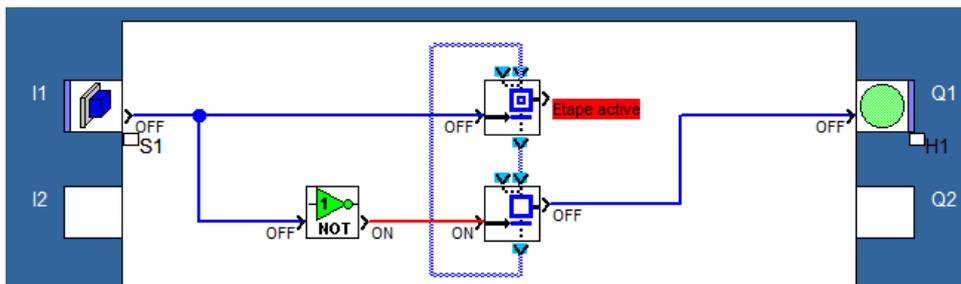


S1 est actionné → I1 est alimentée → Q1 est activé → le voyant est allumé :

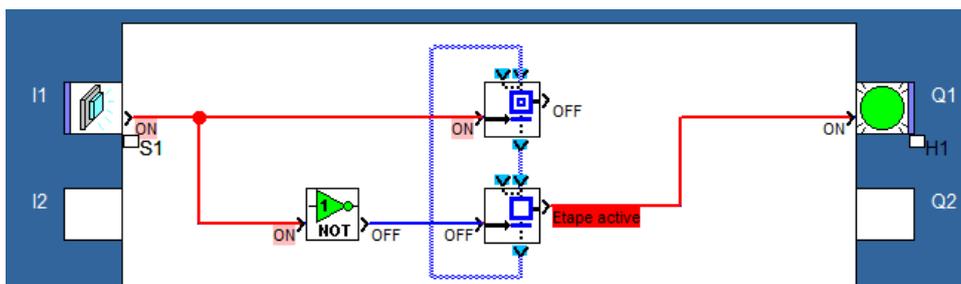


▪ Programme en SFC :

S1 est au repos → I1 n'est pas alimentée → Etape initiale active → Q1 est au repos → H1 est éteint :



S1 est actionné → I1 est alimentée → Etape suivante active → Q1 est activé → le voyant est allumé :



### 3.4. Critères de choix

Comme indiqué sur la documentation ci-dessous, les principaux critères de choix sont :

- La tension d'alimentation du module (utilisée également pour alimenter les entrées),
- Le nombre et le type d'entrée (TOR et analogique),
- Le nombre et le type de sortie (à relais ou à transistors),
- La présence ou non d'horloge (pour une programmation horaire, hebdomadaire et annuelle).

De plus, selon la gamme choisie, les modules peuvent être avec ou sans afficheur. Des extensions peuvent même être ajoutées à certains modules (entrées/sorties supplémentaires, port RJ45...).

## Références



SR2 A201BD

## Modules logiques Zelio Logic

### Modules logiques compacts

#### Modules logiques compacts avec afficheur

Nombre d'E/S	Entrées TOR	Dont entrées analogiques $\approx$ 0-10 V	Sorties à relais	Sorties à transistors	Horloge	Référence	Masse kg
<b>Alimentation <math>\sim</math> 24 V</b>							
12	8	0	4	0	Oui	SR2 B121B	0,250
20	12	0	8	0	Oui	SR2 B201B	0,380
<b>Alimentation <math>\sim</math> 100...240 V</b>							
10	6	0	4	0	Non	SR2 A101FU (1)	0,250
12	8	0	4	0	Oui	SR2 B121FU	0,250
20	12	0	8	0	Non	SR2 A201FU (1)	0,380
					Oui	SR2 B201FU	0,380
<b>Alimentation <math>\approx</math> 12 V</b>							
12	8	4	4	0	Oui	SR2 B121JD	0,250
20	12	6	8	0	Oui	SR2 B201JD	0,380
<b>Alimentation <math>\approx</math> 24 V</b>							
10	6	0	4	0	Non	SR2 A101BD (1)	0,250
12	8	4	4	0	Oui	SR2 B121BD	0,250
			0	4	Oui	SR2 B122BD	0,220
20	12	2	8	0	Non	SR2 A201BD (1)	0,380
		6	8	0	Oui	SR2 B201BD	0,380
			0	8	Oui	SR2 B202BD	0,280