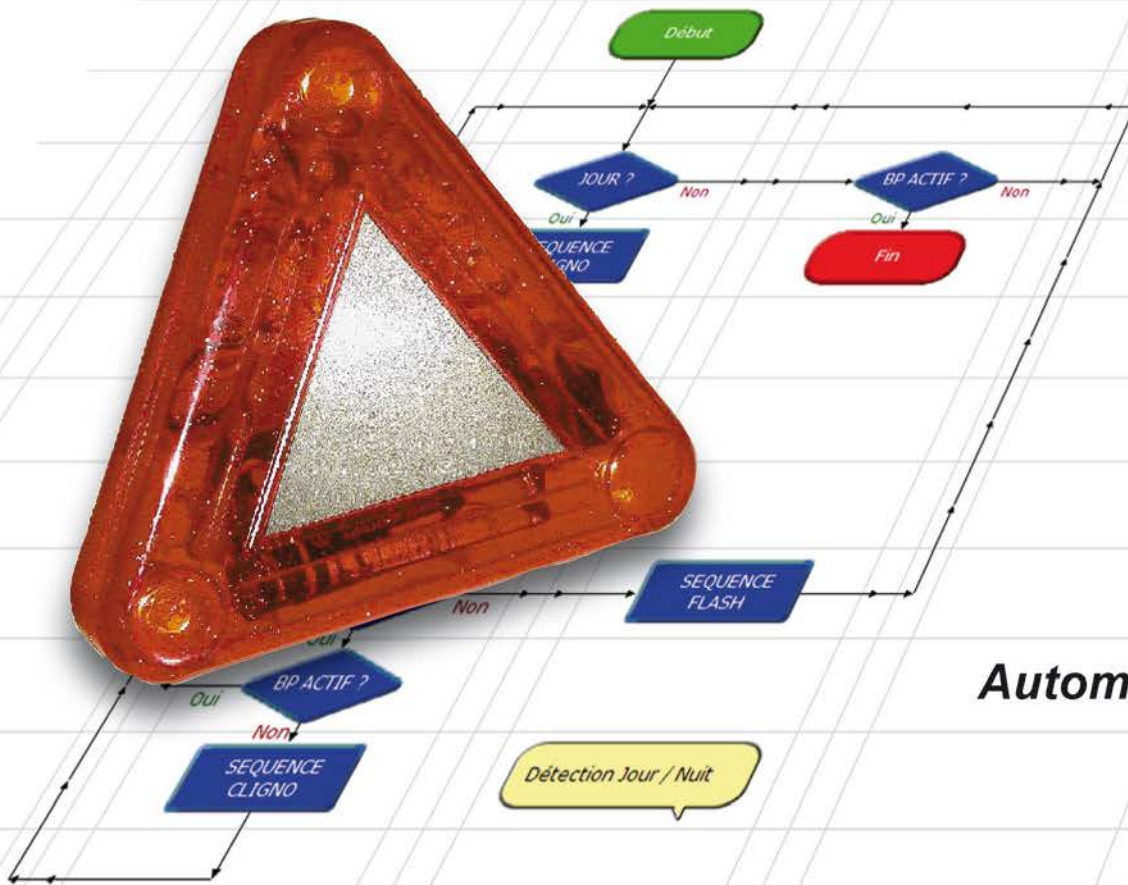
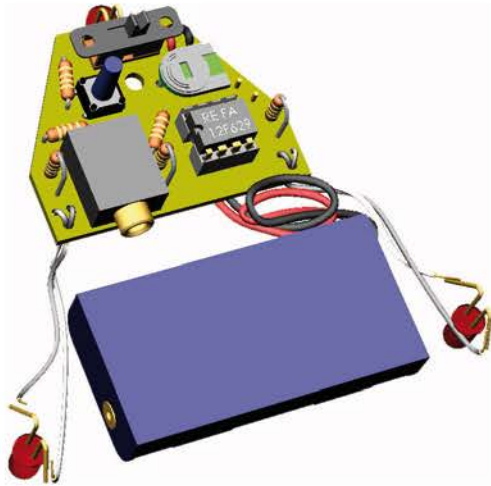


# FlashProg

## TRIANGLE DE SÉCURITÉ PROGRAMMABLE



**Automatisme**





Edité par la société A4  
5 avenue de l'Atlantique – 91940 Les Ulis  
Tél : 01 64 86 41 00 – Fax : 01 64 46 31 19  
[www.a4.fr](http://www.a4.fr)

## SOMMAIRE

<b>Présentation du projet</b> .....	<b>2</b>
Description du produit .....	2
Utilisation pédagogique.....	3
<b>Fabrication</b> .....	<b>5</b>
Plans d'ensemble.....	6
Plans du module électronique .....	8
Montage du module électronique dans le boîtier .....	10
Plan de perçage du circuit imprimé .....	13
Guide de fabrication.....	14
<b>Programmation</b> .....	<b>17</b>
Introduction .....	18
Automatisme niveau 1 - Choisir un programme dans une liste et le charger .....	19
Automatisme niveau 2 - Modifier un programme .....	35
Automatisme niveau 3 - Créer un programme.....	48

### Ressources numériques

Toutes les ressources numériques autour de ce projet sont disponibles gratuitement en téléchargement sur [www.a4.fr](http://www.a4.fr) :

- le dossier sous différents formats (word et PDF).
  - **la modélisation 3D complète** du produit dans ses différentes versions avec des **fichiers 3D** aux formats SolidWorks, Parasolid (.xb) et eDrawings.
  - les programmes d'automation sous *Logicator*.
  - des photos du produit, des images de synthèse, des perspectives au format DXF.
- Les ressources numériques sont également disponibles sur CDROM (réf "**CD-FP**").\*

### Ce dossier et les ressources numériques sont duplicables pour les élèves, en usage interne au collège\*

\*La duplication de ce dossier est autorisée sans limite de quantité au sein des établissements scolaires, à seules fins pédagogiques, à la condition que soit cité le nom de l'éditeur : Sté A4.

La copie ou la diffusion par quelque moyen que ce soit à des fins commerciales n'est pas autorisée sans l'accord de la Sté A4.  
La copie ou la diffusion par quelque moyen que ce soit en dehors d'un usage interne à l'établissement de tout ou partie du dossier ou des ressources numériques ne sont pas autorisées sans l'accord de la Sté A4.

# Présentation du projet

## Description du produit

FlashProg est un triangle de sécurité portatif programmable. Il est destiné à signaler la présence d'un piéton la nuit. Le module électronique fait flasher 3 LED haute luminosité visibles à plus de 100 mètres. Il dispose de :

- Trois sorties effecteurs indépendantes constitués par trois LED.
- Deux entrées constituées par un capteur photosensible (LDR) et un bouton poussoir



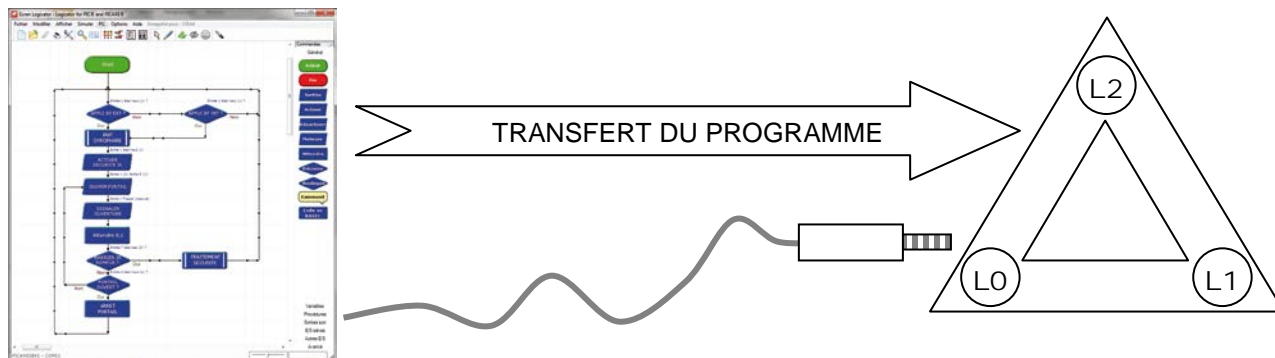
La programmation du module permet de créer une grande variété de séquences d'allumage des LED. Le module accueille un bouton-poussoir et une LDR qui permettent de sélectionner différentes séquences d'allumage des LED en fonction d'un évènement extérieur (appui sur le bouton poussoir, seuil de luminosité ambiante).

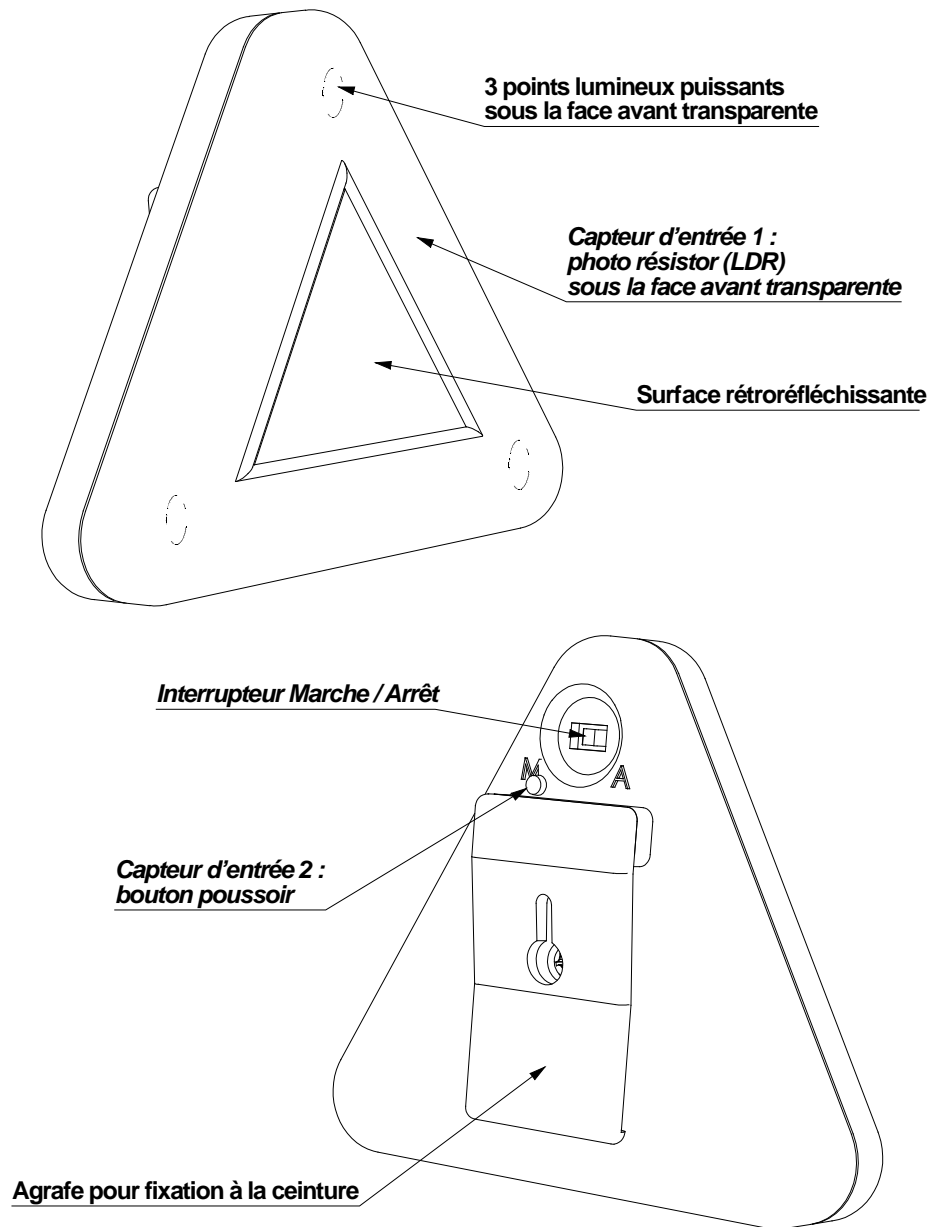
La variété du jeu d'instructions disponibles pour la programmation permet de créer des séquences d'allumage des LED des plus simples aux plus complexes comme par exemple :

- le clignotement simultané des LED,
- le clignotement alterné des LED (effet « chenillard »),
- le chaînage de séquences différentes,
- des tests conditionnels sur des actions extérieures (réaction au niveau lumineux ambiant, réaction à l'appui sur le bouton poussoir),
- le comptage d'un nombre de cycles (pour répéter l'exécution d'une séquence plusieurs fois),
- l'exécution d'un programme principal qui fait appel à des sous programmes
- . . . .

La programmation du module FlashProg est faite à l'aide du logiciel PICAXE *Logicator* qui permet de créer des diagrammes de programmation qui seront transférés dans le module en connectant l'ordinateur au FlashProg à l'aide d'un câble approprié.

Le module reçoit le programme dans sa mémoire (Flash Prog) et est reprogrammable à volonté.





## Utilisation pédagogique

Le but de ce dossier est de guider le professeur pour l'exploitation du FlashProg avec des élèves et d'aborder par l'exemple des notions simples d'automatisme.

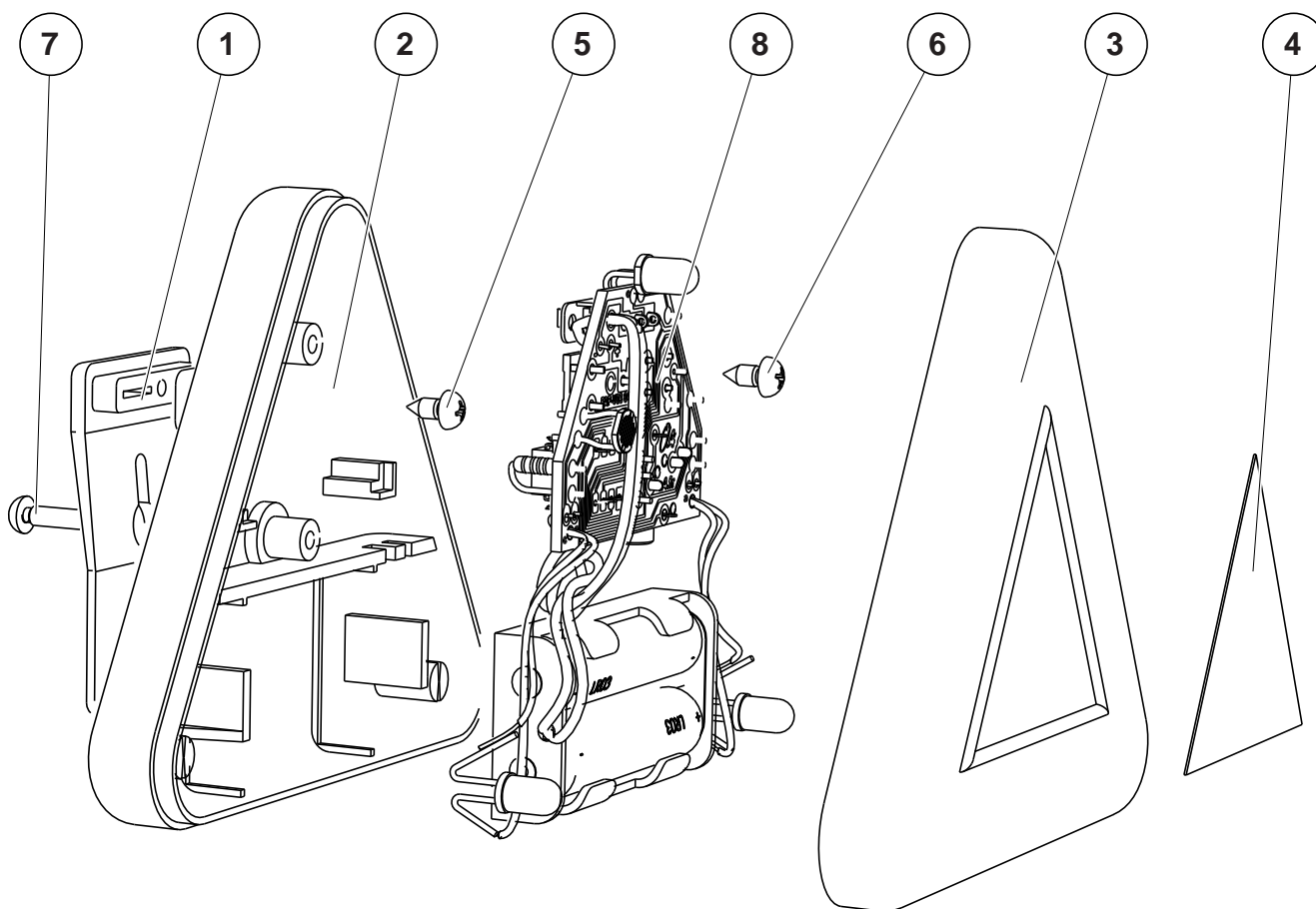
- Le chapitre « Fabrication » traite de la fabrication du produit et l'implantation des composants sur le circuit imprimé.
- Le chapitre « Programmation » traite de l'exploitation du FlashProg en automatisme.  
 Cette partie ne se substitue pas à la documentation de l'environnement de programmation qui décrit (en anglais) l'intégralité des possibilités offertes par ce système.  
 L'environnement de programmation « PICAXE Logicator » et l'interface logicielle préprogrammée du microcontrôleur du FlashProg sont des marques déposées de la société Revolution Education.  
 Le produit FlashProg est une conception de la société A4.



# FABRICATION



<b>Ensemble.....</b>	<b>6</b>
<b>Module électronique.....</b>	<b>10</b>
<b>Montage du module électronique.....</b>	<b>14</b>
<b>Circuit imprimé.....</b>	<b>15</b>
<b>Guide de fabrication.....</b>	<b>16</b>

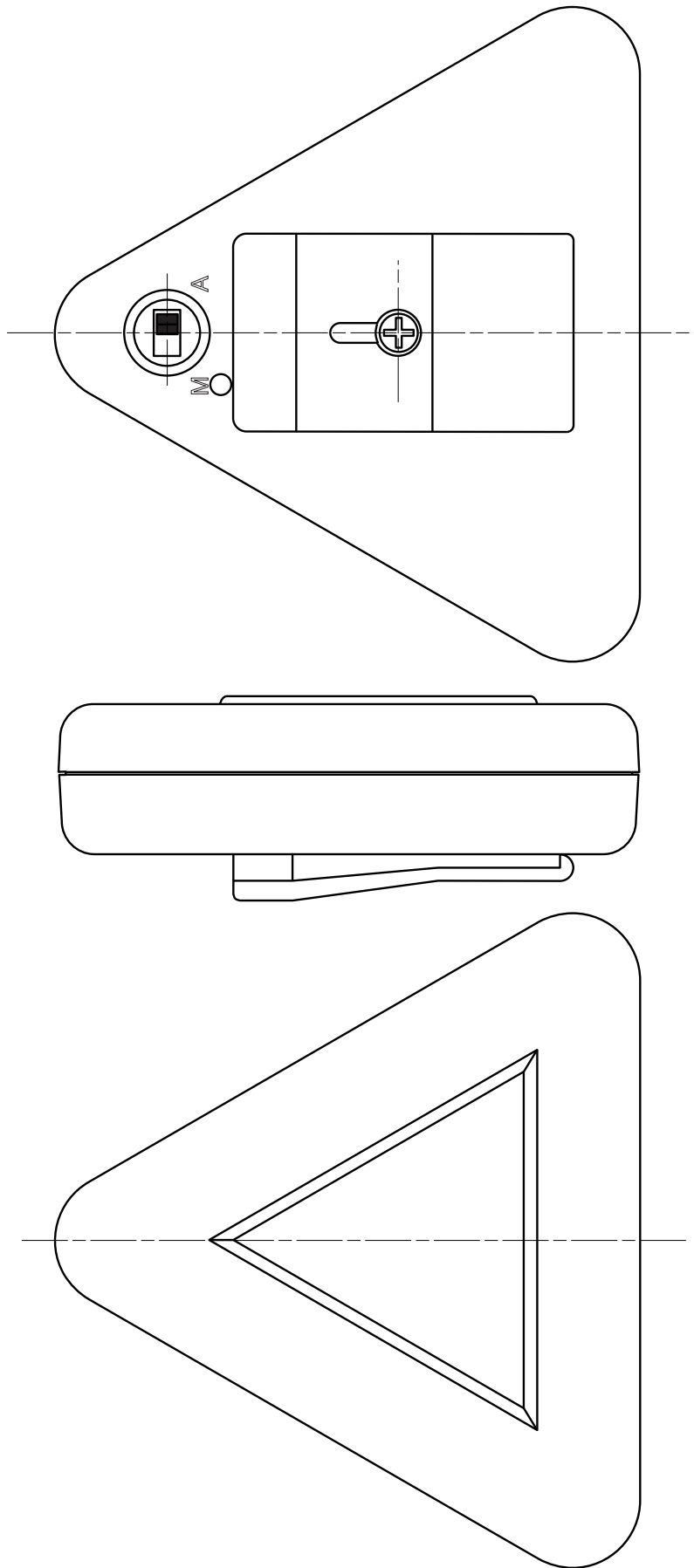



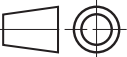
8	01	Module électronique complet	Voir Nomenclature du module électronique
7	01	Vis de fixation de la demi-coque avant	Type tôle - Tête cylindrique - Ø 3.2 x longueur 16
6	01	Vis de fixation du module	Type tôle - Tête cylindrique - Ø 3.2 x longueur 6.5
5	01	Vis de fixation de l'agrafe	Type tôle - Tête cylindrique - Ø 3.2 x longueur 6.5
4	01	Rétro réfléchissant	Rétro réfléchissant auto-adhésif - Triangle isocèle 50mm
3	01	Demi-coque avant	PS cristal transparent teinté injecté
2	01	Demi-coque arrière	PS cristal transparent teinté injecté
1	01	Agrafe	PS cristal transparent teinté injecté

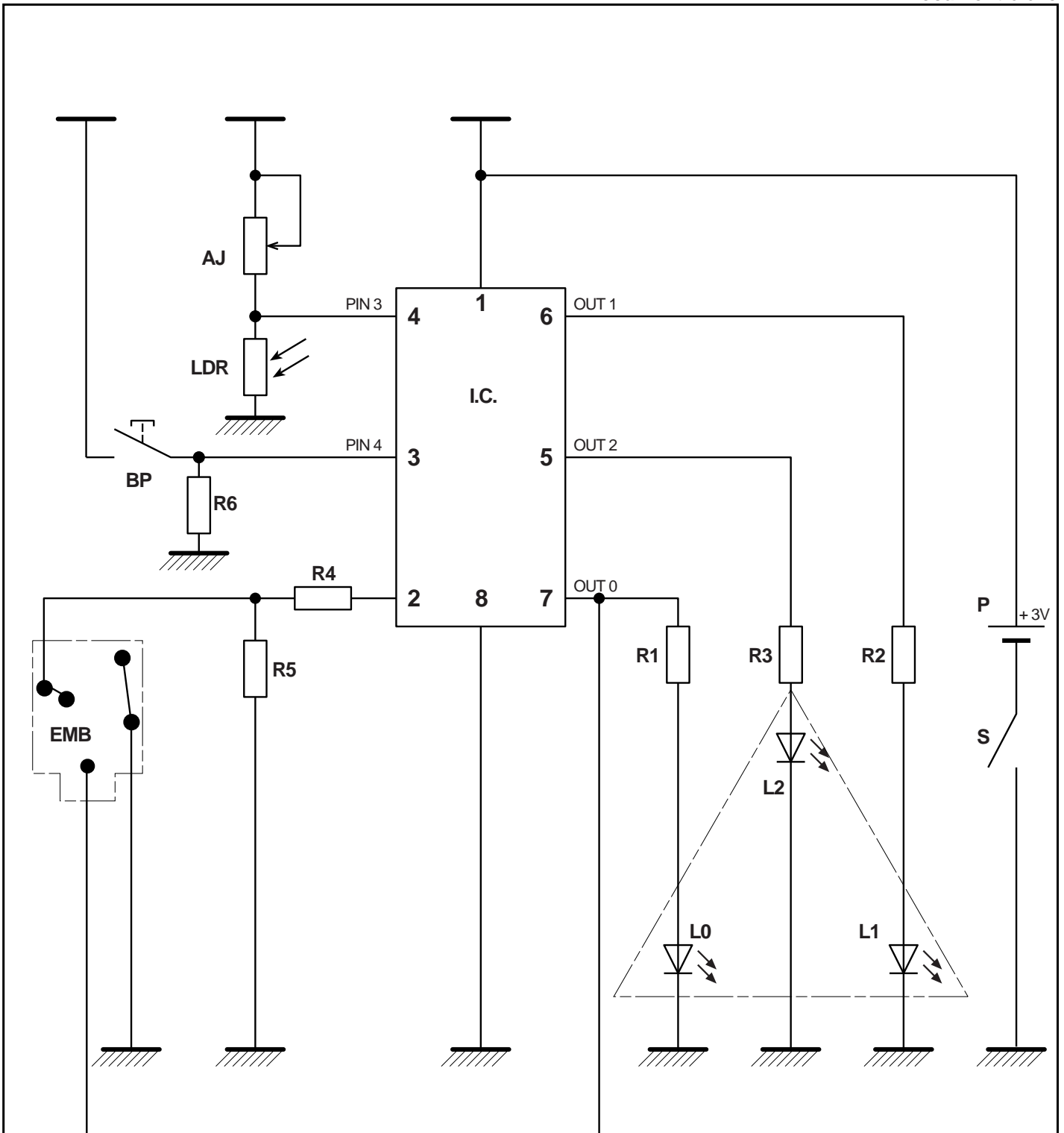
REPERE	NOMBRE	DESIGNATION	CARACTERISTIQUES
			PROJET <b>FlashProg</b> PARTIE <b>ENSEMBLE</b>
	Collège	Classe <b>A4</b>	TITRE DU DOCUMENT
	Nom	Date	<b>ECLATÉ ET NOMENCLATURE GENERALE</b>


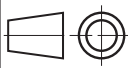


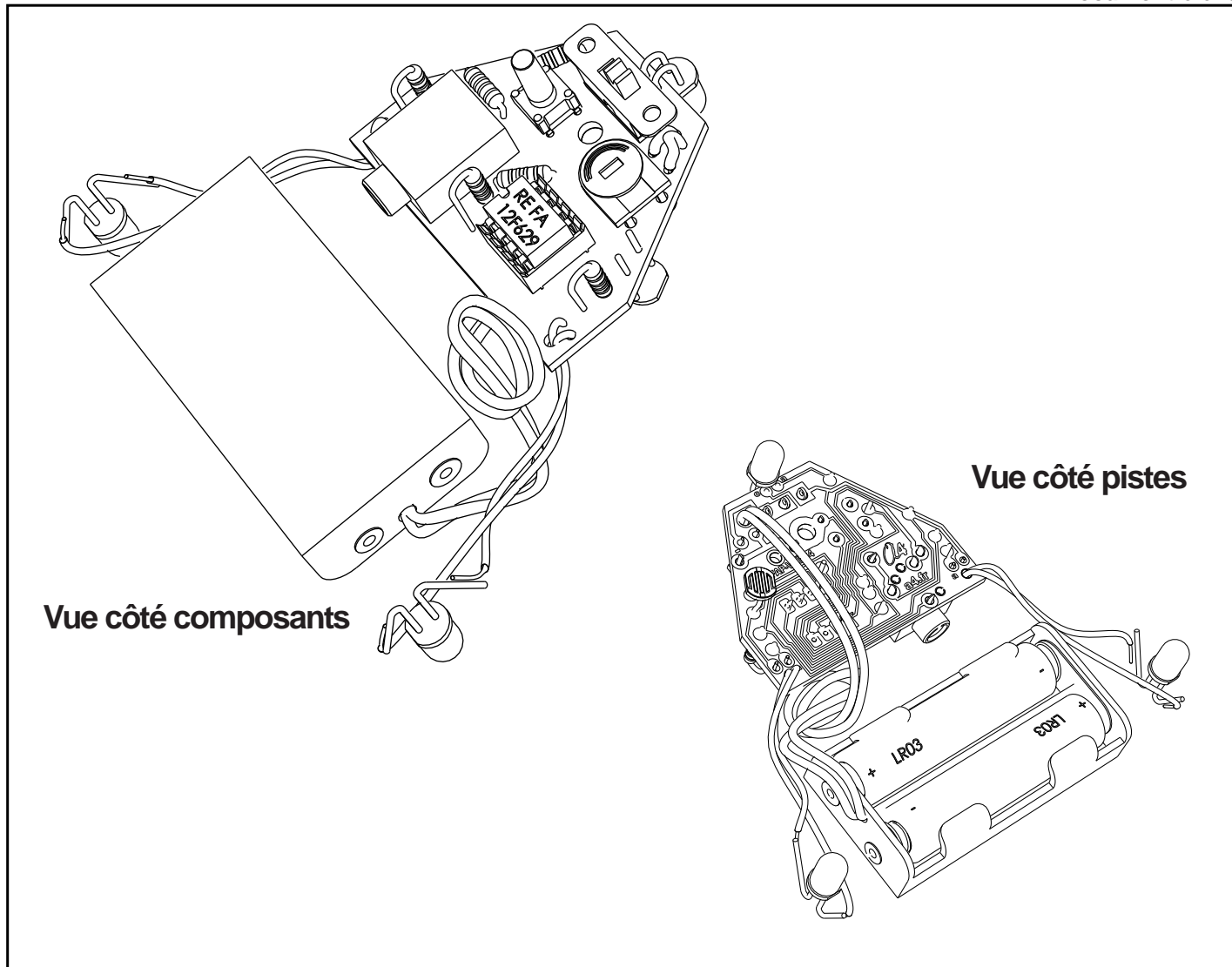





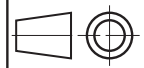

 TECHNOLOGIE AU COLLEGE	Echelle 1:1		<b>A4</b>	PROJET <b>FlashProg</b>	PARTIE <b>ENSEMBLE</b>
	Collège	Classe		TITRE DU DOCUMENT <b>DESSIN D'ENSEMBLE</b>	
	Nom	Date			

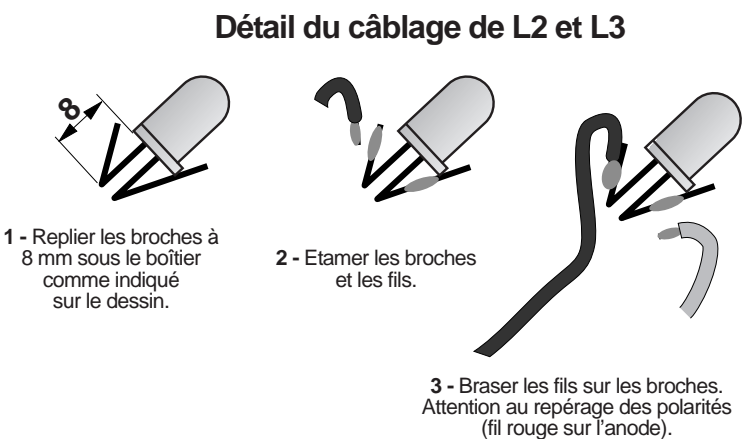
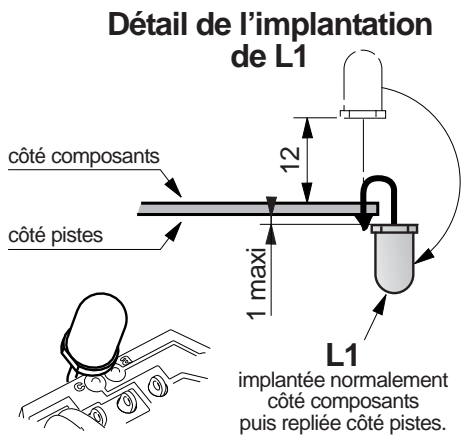
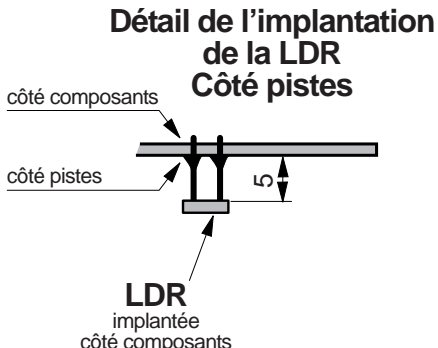
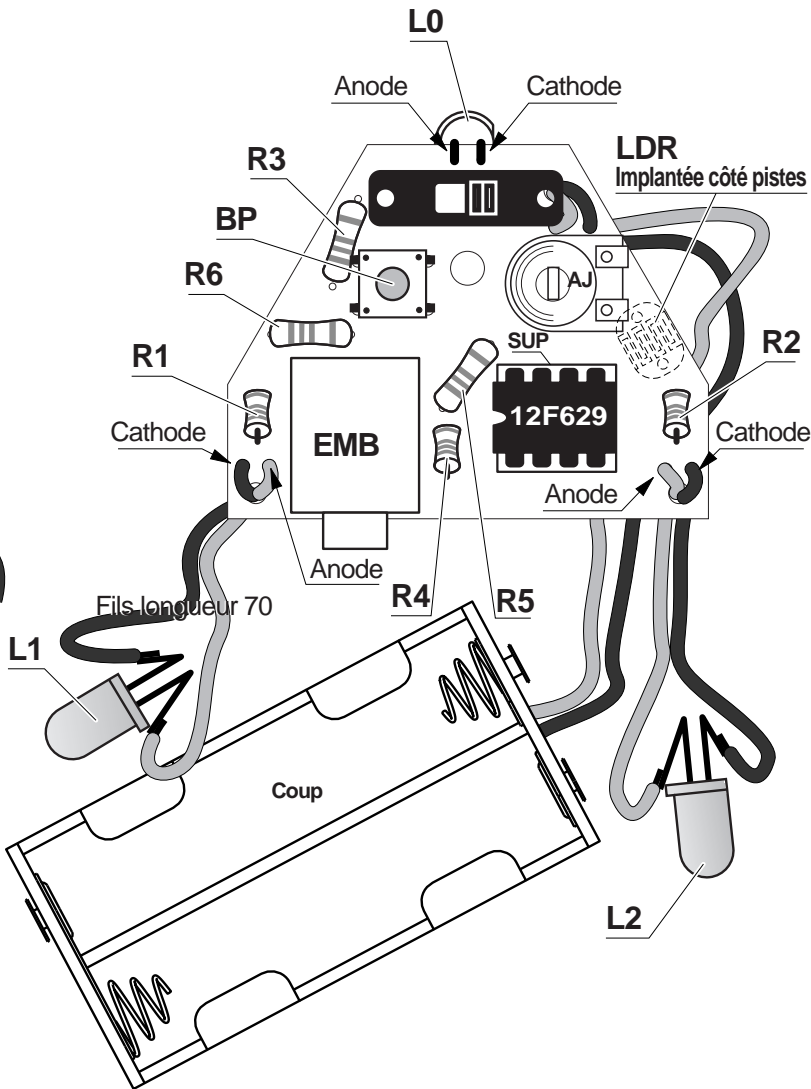
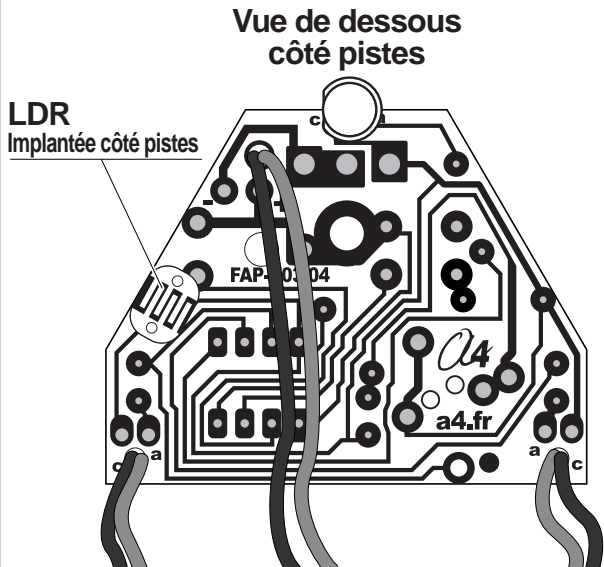


		<b>A4</b>	PROJET	PARTIE
	Collège	Classe	<b>FlashProg</b>	<b>MODULE ELECTRONIQUE</b>
	Nom	Date	TITRE DU DOCUMENT	<b>SCHEMA STRUCTUREL</b>

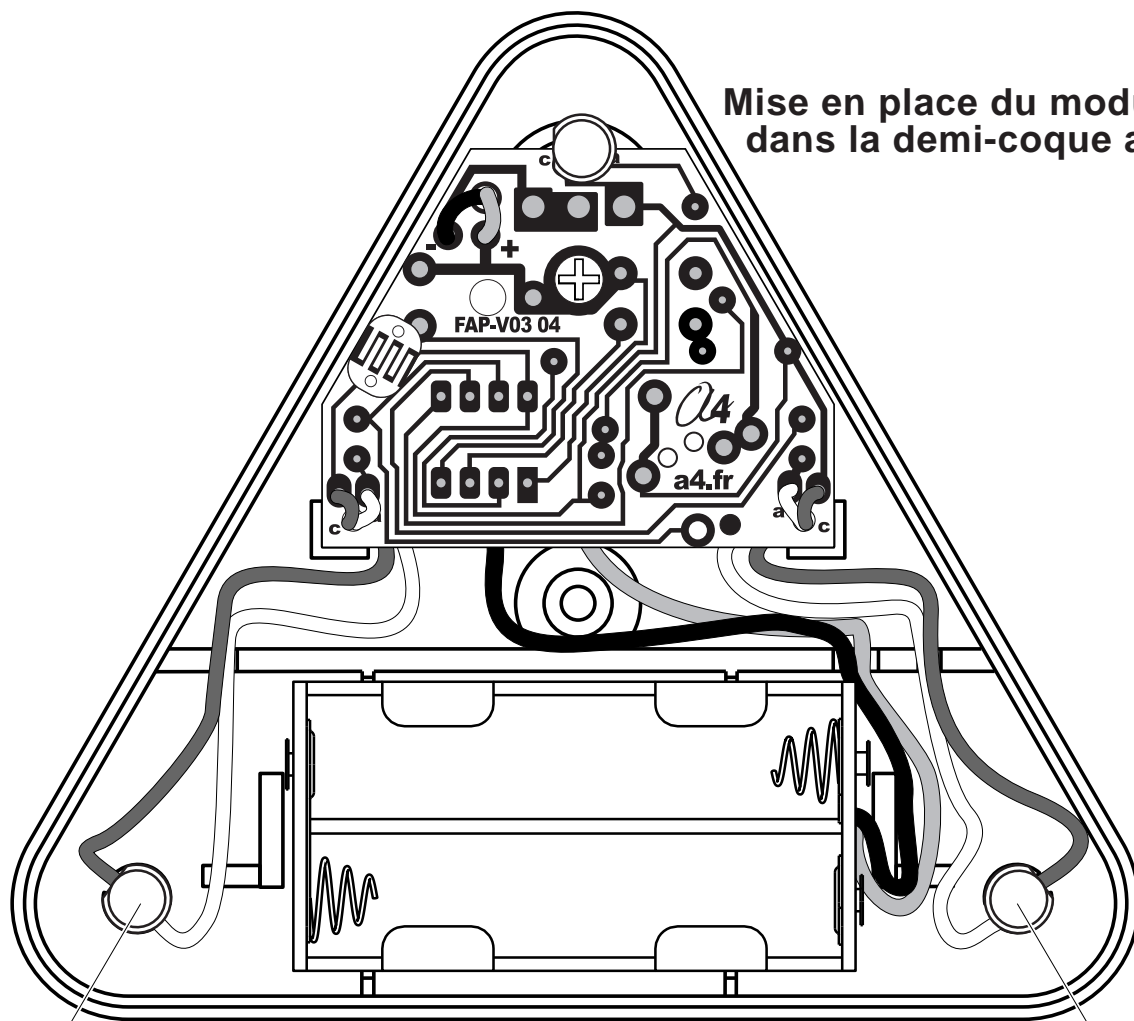


EMB	01	Embase jack	Pour jack stéréo Ø 3
S	01	Micro-inverseur	Micro-inverseur à glissière unipolaire
BP	01	Bouton poussoir	Bouton poussoir miniature de C.I. - 1 contact travail
LDR	01	Photo-résistor	LDR Ø 5
R6	01	Résistor 10 kOhm	1/4 Watt - 10 kOhm (Marron, Noir, Orange, Or)
R4 - R5	02	Résistor 22 kOhm	1/4 Watt - 22 kOhm (Rouge, Rouge, Orange, Or)
R1 à R3	03	Résistor 120 Ohm	1/4 Watt - 120 Ohm (Marron, Rouge, Marron, Or)
AJ	01	Résistor ajustable	Horizontal - 470 Ohm
L0 à L2	03	Diode Electroluminescente	Ø 5 - Rouge - Boîtier cristal - 1000 mcd
F	02	Fil deux conducteurs	Fil souple 2 conducteurs - Longueur 70 mm
P	02	Pile	1,5 Volt - Type LR03
Cou	01	Support de pile	Support pour 2 piles LR03 - Sortie fils
I.C.	01	Circuit intégré programmable	Référence "IC REFA 12F629" - Boîtier DIL 8
SUP	01	Support de circuit intégré	8 broches
C.I.	01	Circuit imprimé	Simple face - 1.6 x 35 x 45

REPERE	NOMBRE	DESIGNATION	CARACTERISTIQUES
			PROJET
			PARTIE
			TITRE DU DOCUMENT
		FlashProg	MODULE ELECTRONIQUE
		Nom	Date
		<b>NOMENCLATURE DU MODULE ELECTRONIQUE</b>	



	<b>Echelle 1.5:1</b>		<b>A4</b>	PROJET <b>FlashProg</b>	PARTIE <b>MODULE ELECTRONIQUE</b>
	Collège	Classe		TITRE DU DOCUMENT	
	Nom	Date		<b>PLAN D'IMPLANTATION</b>	

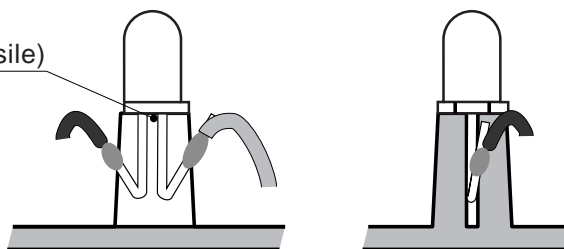


Mise en place du module dans la demi-coque arrière

L1

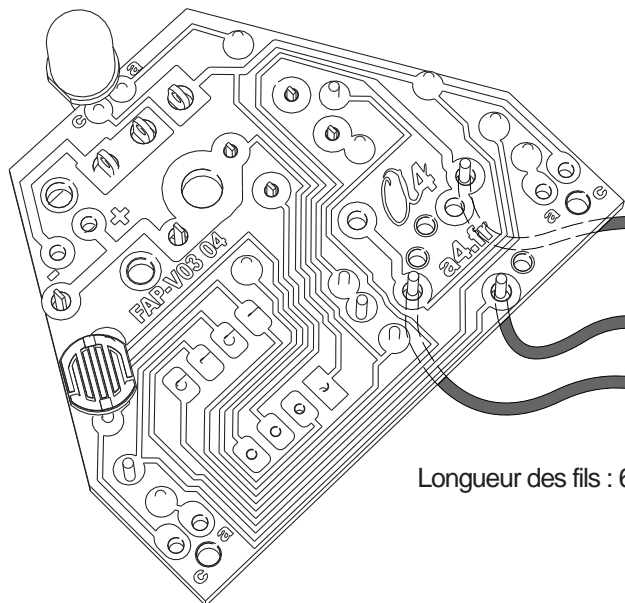
L2

Point de colle  
(colle PVC  
ou colle thermofusile)



Détail de la pose des DEL L1 et L2 sur leurs supports

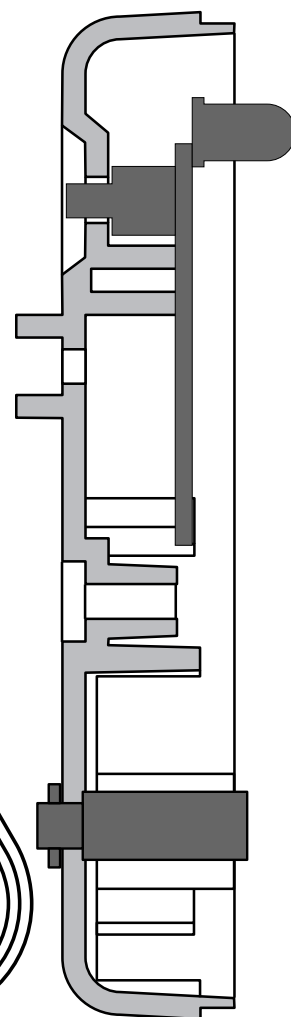
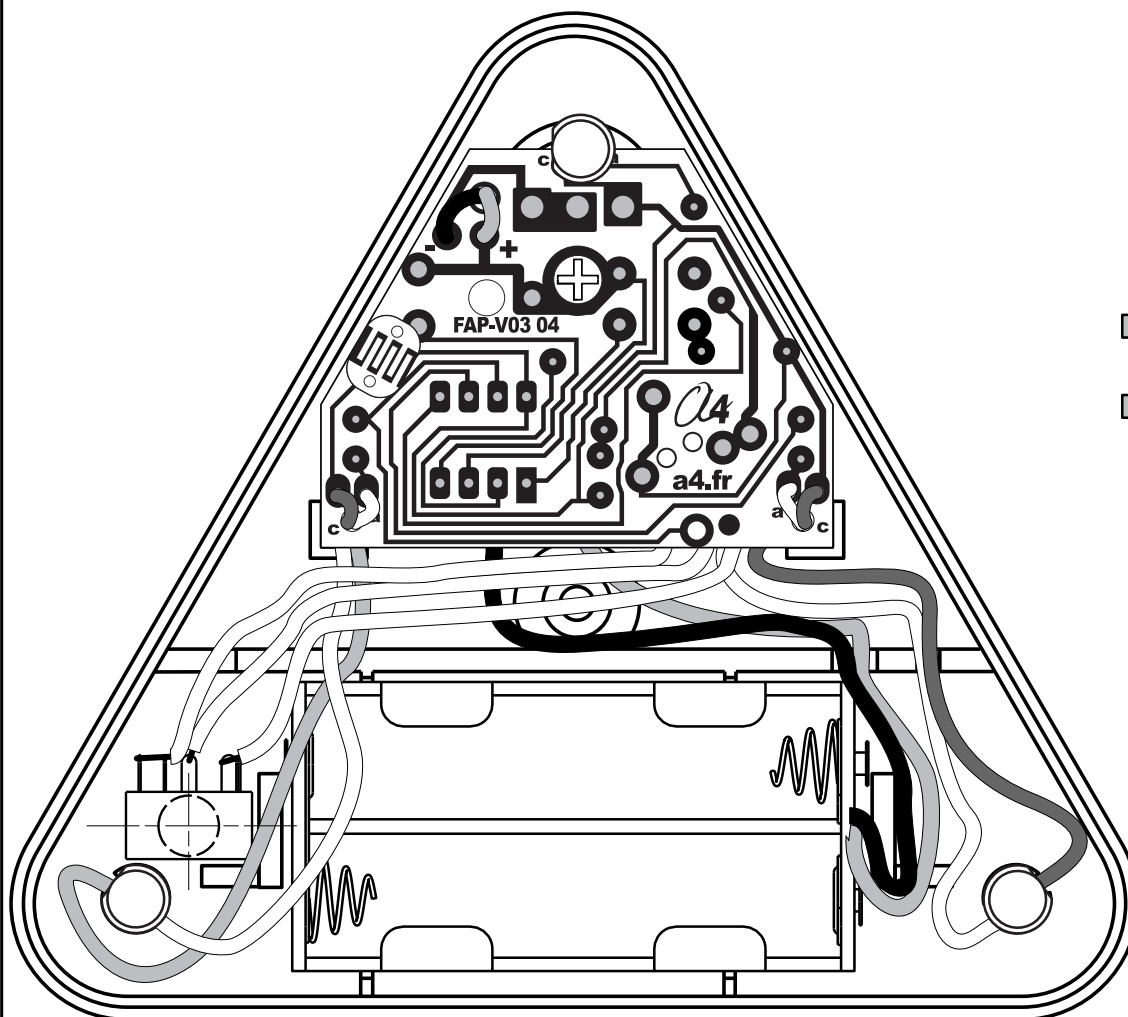
	Echelle 1.5:1		A4	PROJET	PARTIE
	Collège	Classe		FlashProg	MODULE ELECTRONIQUE
	Nom	Date		TITRE DU DOCUMENT PLACEMENT DU MODULE DANS LE BOÎTIER	


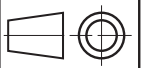


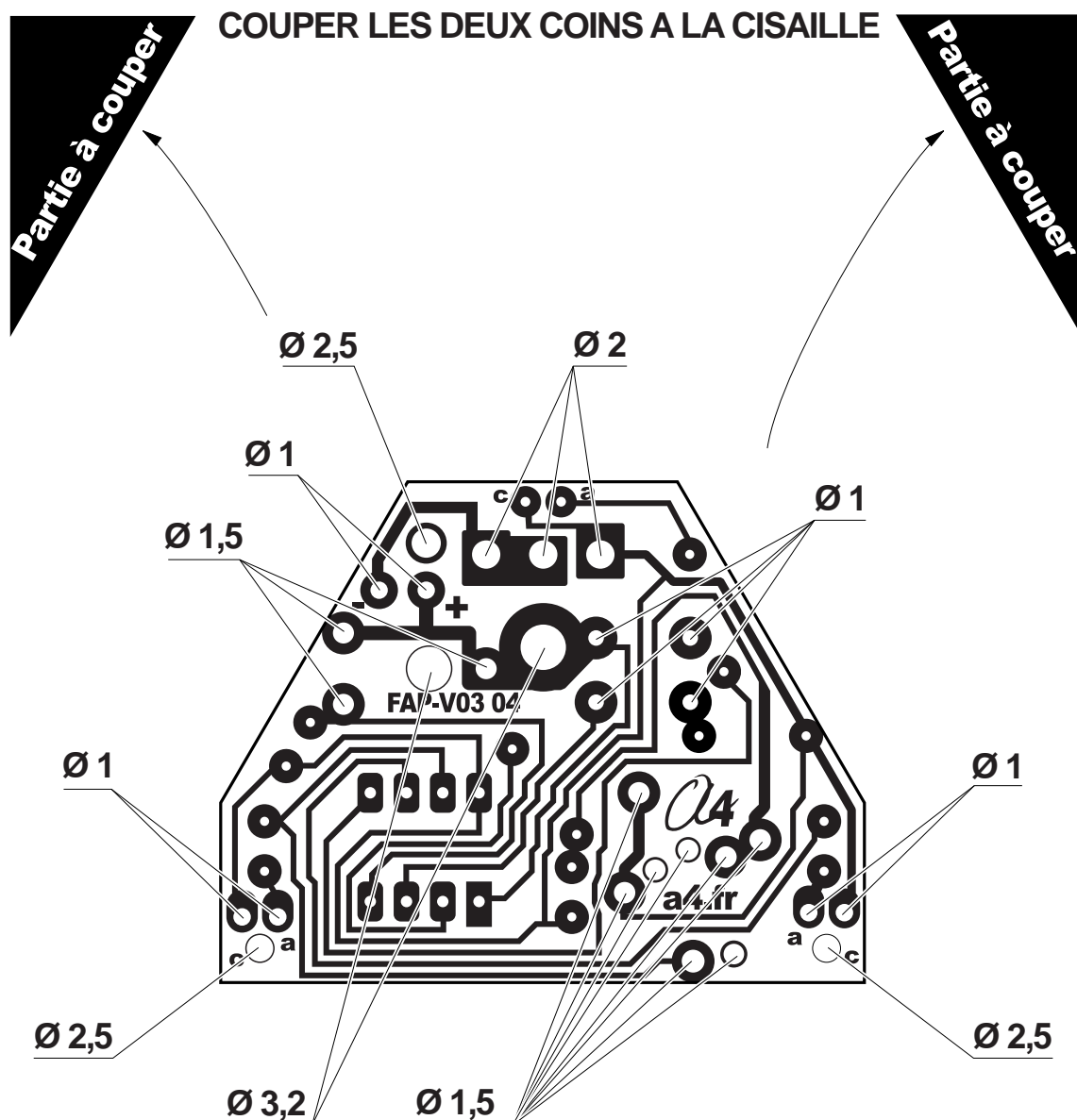
Afin de permettre la programmation plus facilement sans avoir à démonter le module, on peut déporter l'embase jack sur la demi-coque arrière du boîtier. Cela suppose un câblage du connecteur avec des fils et le perçage du boîtier au diamètre 6 comme indiqué sur ce document.

Longueur des fils : 60 mm.

Embase jack




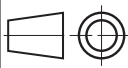
 TECHNOLOGIE AU COLLEGE	Echelle 1.5:1		A4	PROJET	PARTIE
	Collège	Classe		FlashProg	MONTAGE DU MODULE ELECTRONIQUE
	Nom	Date		TITRE DU DOCUMENT <b>MONTAGE DU CONNECTEUR                  DEPORTÉ SUR LE BOITIER</b>	



**PERCER TOUS LES TROUS AU Ø 0,8**

**Puis agrandir les trous de plus forts diamètres :**

- Ø 3,2 : 2 trous
- Ø 2,5 : 3 trous
- Ø 2 : 3 trous
- Ø 1,5 : 11 trous
- Ø 1 : 10 trous

 <b>TECHNOLOGIE AU COLLEGE</b>	Echelle 2:1		<b>A4</b>	PROJET	PARTIE
	Collège	Classe	FlashProg	MODULE ELECTRONIQUE	
	Nom	Date	TITRE DU DOCUMENT		
<b>PLAN DE PERCAGE ET DEÇOUPE DU CIRCUIT IMPRIMÉ</b>					

## A - Implantation et brasage des composants sur le circuit imprimé

Ordre d'implantation	Recommandations	Polarités
01 - Support de circuit intégré	En appui sur le circuit imprimé	Polarité : encoche d'un côté
02 - Inverseur à glissière	En appui sur le circuit imprimé	Polarité : aucune
03 - Résistor ajustable	En appui sur le circuit imprimé	Sans objet
04 - Embase jack	En appui sur le circuit imprimé (sauf montage déporté sur le boîtier : voir plan)	Sans objet
05 - Bouton poussoir	En appui sur le circuit imprimé	Polarité : aucune
06 - Résistors R1 à R6	En appui sur le circuit imprimé Les résistors R1, R2 et R4 sont implantés verticalement	Polarité : aucune
07 - LED L0	Hauteur 12 mm Replier après brasage (Voir plan)	Polarité : anode = broche longue
08 - LDR (photo résistor)	Hauteur 5 mm Implantation et brasage côté pistes	Polarité : aucune
09 - LED L1 et L2	Selon plan d'implantation - Préparer les LED (pliage et étamage des broches) - Préparer les fils (mise à longueur 70 + étamage) - Monter les LED sur les fils (brasage) - Planter et braser les fils sur le circuit imprimé	Polarité : anode = broche longue Fil repéré sur l'anode Fil repéré sur l'anode
10 - Coupleur de pile		Polarité : fil rouge = "+"
11 - Circuit intégré	Montage à la main sur son support	Polarité : encoche d'un côté

## B - Contrôle visuel du module électronique

Il est important, avant de placer pour la première fois des piles dans le support, de s'assurer qu'il n'y a pas de défauts flagrants qui pourraient conduire à endommager le module dès la première mise sous tension.

- Vérifier que de la brasure en excès ne provoque pas de court-circuit entre deux pistes. Utiliser le plan du circuit imprimé.
- Vérifier que des broches de composants qui n'auraient pas été coupées au ras des brasures ou des brins de cuivre des fils ne provoquent pas de court-circuit. Vérifier cela côté piste et aussi côté composants.

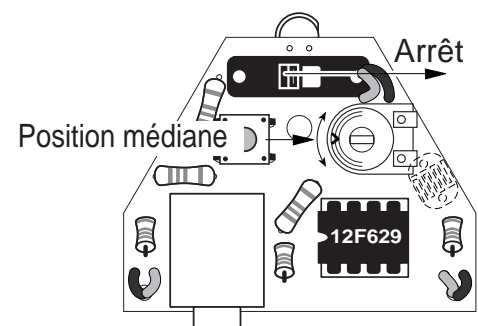
## C - Essais en marche du module électronique

### 1 - Chargement d'un programme de test.

Vous devez disposer :

- d'un ordinateur sur lequel vous avez installé le logiciel *Logicator*;
- du CDRom FlashProg qui contient le programme test ;
- d'un cordon interface PC (Réf CABLE FP) ;
- d'un module FlashProg à tester avec deux piles LR03.

1. Lancer l'application Logicator. A l'ouverture de l'application, paramétrer "PICAXE 8" (qui correspond au micro-contrôleur du FlashProg).
2. Ouvrir sur le CDRom le programme test "TEST.plf"
3. Déplacer l'interrupteur à glissière du module en position arrêt (vers la droite).
4. Régler l'ajustable sur la position médiane.
5. Placer des piles dans le support du module.
6. Connecter le module au port COM1 de l'ordinateur par l'intermédiaire du cordon interface PC.
7. Déplacer l'interrupteur du module en position marche (vers la gauche).





8. Presser la touche F5 du clavier de l'ordinateur (l'ordinateur convertit le diagramme en langage basic).
9. Presser à nouveau la touche F5 du clavier (l'ordinateur se connecte au module et lui transfère le programme). En cas d'échec du transfert, retirer les piles du module sans tarder et réparer la panne du module avant un nouvel essai. Le programme chargé avec succès, cliquez sur "OK".
10. Eteindre le module FlashProg et le déconnecter de son câble interface PC.

## 2 - Test du bon fonctionnement du module électronique FlashProg

- a. Allumer le Module FlashProg en déplaçant le bouton de son interrupteur :  
Les 3 LED doivent s'allumer successivement (effet chenillard).  
En cas d'échec, retirer les piles du module sans tarder et réparer la panne du module avant un nouvel essai.
- b - Appuyer une fois sur le bouton-poussoir :  
Le mode de clignotement doit changer : les 3 LED doivent clignoter simultanément.  
En cas d'échec, retirer les piles du module sans tarder et réparer la panne du module avant un nouvel essai.
- c - Occulter la LDR avec un capuchon noir (en plastique pour éviter les court-circuits) :  
Le clignotement doit s'accélérer nettement (quel que soit le mode choisi à l'aide du bouton-poussoir).  
En cas d'échec, retirer les piles du module sans tarder et réparer la panne du module avant un nouvel essai.

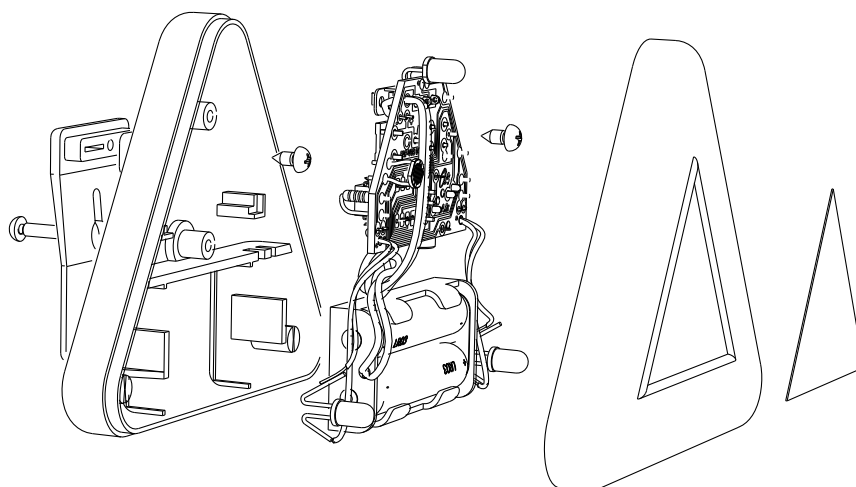
Dépannage : Comme pour tous les modules électroniques câblés artisanalement, 90% des pannes proviennent de défauts de brasage (court-circuits ou brasures sèches) et 10 % de pannes proviennent d'inversions de polarités. Très rares sont les pannes imputables à des défauts des composants mais il est possible bien que peu fréquent qu'un court-circuit prolongé puisse "griller" un circuit intégré.

La recherche d'une panne sur un module défectueux simple comme celui du FlashProg se fera plus efficacement avec une loupe et le plan d'implantation qu'avec un appareil de contrôle et le schéma structurel.

## D - Montage du module dans le boîtier

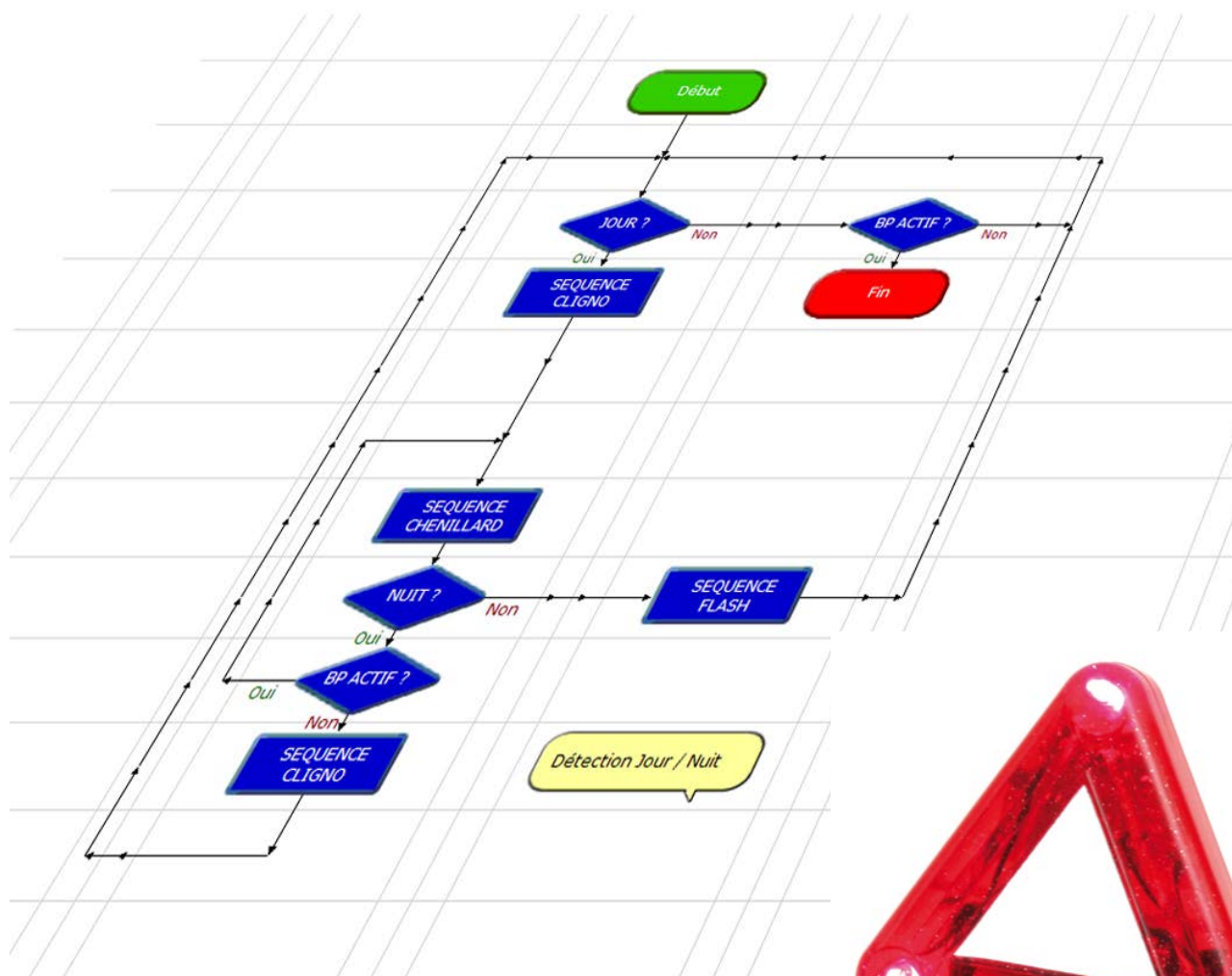
Vous devez utiliser la vue en éclaté et le plan de placement du module dans le boîtier.

- 1 - Dégrapper et ébavurer les 3 pièces du boîtier.
- 2 - Monter l'agrafe en place et la fixer par une vis 3x6,5.
- 3 - Monter le module en place et le fixer par une vis 3x6,5.
- 4 - Placer les LED L1 et L2 sur leur support (selon les détails visibles sur le plan de placement).  
On peut assurer leur maintien par un point de colle.
- 5 - Découper l'autocollant rétro réfléchissant en triangle équilatéral de côté 50 mm et le poser en place sur la demi-coque avant du boîtier.
- 6 - Assembler les deux demi-coques du boîtier et les maintenir par la vis 3x16.





# PROGRAMMATION



<b>Introduction.....</b>	<b>18</b>
<b>Automatisme niveau 1 .....</b>	<b>19</b>
Choisir un programme dans une liste et le charger	
<b>Automatisme niveau 2 .....</b>	<b>35</b>
Modifier un programme	
<b>Automatisme niveau 3 .....</b>	<b>48</b>
Créer un programme	

## Introduction

FlashProg est un petit système automatisé dont la simplicité de fonctionnement permet une approche à la fois ludique et sereine de l'automatisme. Ce chapitre est organisé en trois rubriques de difficulté croissante.

- Automatisme niveau 1 :  
l'élève dont le produit fonctionne va choisir un programme dans une liste et le charger dans son module. Les programmes proposés sont classés par ordre de difficulté croissante.
- Automatisme niveau 2 :  
l'élève va modifier quelques paramètres d'un programme existant et le tester sur son module.
- Automatisme niveau 3 :  
l'élève va créer un programme simple et le tester sur son module.

## Notions de base

### Qu'est-ce qu'un microcontrôleur

Un microcontrôleur est un circuit intégré qui contient :

- de la mémoire de programme (qui contient elle-même un programme) ;
- un microprocesseur (qui exécute les instructions du programme) ;
- des entrées pour connecter des capteurs (bouton-poussoir, LDR...) ;
- des sorties pour connecter des effecteurs (par exemple des LED).

### Qu'est-ce qu'un programme

Un programme est constitué d'une série d'instructions qui sont interprétées et exécutées les unes après les autres par le microcontrôleur.

Pour fonctionner, le programme doit être écrit dans un langage compris par le microcontrôleur.

Ce langage obéit à des règles strictes de syntaxe propres au microcontrôleur.

L'exécution du programme est cadencée par une horloge interne au microcontrôleur.

On peut retenir comme ordre de grandeur que le microcontrôleur utilisé dans le module FlashProg peut exécuter un million d'instructions de base par seconde.

Le langage de base du microcontrôleur est composé d'instructions rudimentaires (langage machine ou assembleur).

Afin de rendre la programmation plus conviviale, l'environnement «PICAXE Logicator» permet de définir un diagramme de programmation.

Ce diagramme est dans un premier temps converti en un langage évolué (basic), puis dans un deuxième temps en langage machine qui est chargé dans le microcontrôleur.

### Environnement de programmation « PICAXE Logicator »

Dans un souci de clarté et de prise en main rapide du système, nous limitons volontairement les explications qui concernent l'environnement de programmation aux éléments utiles à la programmation du FlashProg.

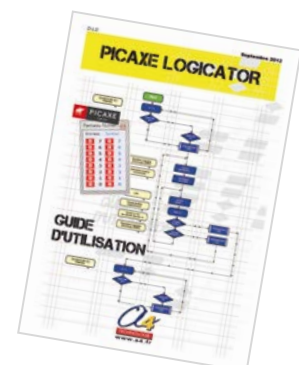
Le logiciel «PICAXE Logicator» permet d'écrire le programme qui va piloter le module FlashProg.

- Le mode Diagramme permet de créer des programmes à l'aide de blocs.
- Le contenu d'un diagramme est converti en programme basic.
- Le programme en basic est transféré dans le module pour être exécuté dès la mise sous tension du module électronique.

### Mise en œuvre du logiciel de programmation « PICAXE Logicator »

Pour procéder à la programmation du module du FlashProg, vous devez disposer du logiciel "PICAXE Logicator, téléchargeable gratuitement sur [www.a4.fr](http://www.a4.fr), d'un ordinateur et du câble spécifique (Réf. : **CABLE-FP**) permettant de transférer vos programmes dans le module FlashProg.

*Pour plus d'informations sur PICAXE Logicator, reportez-vous au manuel utilisateur Logicator (D-LG), téléchargeable gratuitement sur [www.a4.fr](http://www.a4.fr)*



# AUTOMATISME NIVEAU 1

## Choisir un programme dans une liste et le charger

L'élève charge un programme existant et vérifie que le FlashProg se comporte conformément à sa description.

Chaque programme est accessible à partir du dossier « Automatismes niveau 1 » du CD Rom **CD-FP**.

### Liste des fiches élèves :

	Description	Fichier programme
FICHE N°1	<b>Pilotage des sorties :</b> Clignotement simultané des trois LED.	F1-CLIGNO LED.plf
FICHE N°2	<b>Pilotage des sorties :</b> Clignotement alterné des trois LED (effet chenillard).	F2-CHENILLARD.plf
FICHE N°3	<b>Pilotage des sorties :</b> Bâton tournant. Les LED s'allument deux à deux en tournant.	F3-BATON TOURNANT.plf
FICHE N°4	<b>Enchaînement de séquences :</b> Alternance du clignotement simultané des trois LED et d'un effet chenillard.	F4-CLIGNO-CHENILLARD.plf
FICHE N°5	<b>Utilisation du capteur photosensible :</b> Mise en route automatique à la tombée de la nuit.	F5-DEMAR AUTO NUIT.plf
FICHE N°6	<b>Utilisation du capteur photosensible :</b> Détection d'un phare de voiture dans la nuit et changement automatique de mode de clignotement.	F6-DETEC PHARE.plf
FICHE N°7	<b>Utilisation du capteur Bouton-Poussoir :</b> Sélection alternée de 2 effets par le bouton-poussoir.	F7-SELECTION PAR BP.plf
FICHE N°8	<b>Utilisation du capteur photosensible et Bouton-poussoir :</b> Démarrage automatique dès qu'il fait nuit ou dès que l'on active le bouton-poussoir.	F8-DEMAR AUTO LDR-BP.plf
FICHE N°9	<b>Comptage de séquences :</b> Répétition de deux séquences différentes.	F9-REPET 2 SEQUENCES.plf
FICHE N°10	<b>Utilisation d'un temps de pause variable :</b> Effet d'accélération d'un chenillard suivi par un clignotement.	F10-ACCELERATION.plf
FICHE N°11	<b>Utilisation d'un temps d'attente long :</b> Exécution d'une séquence chenillard une fois toutes les 3 secondes le jour et en permanence la nuit.	F11-CHENILLARD JOUR-NUIT.plf
FICHE N°12	<b>Utilisation d'une variable aléatoire :</b> Clignotement aléatoire.	F12-HASARD.plf
FICHE N°13	<b>Utilisation d'un sous-programme :</b> Accélération d'un chenillard commandée par le bouton-poussoir.	F13-ACCELERATION PAR BP.plf

### Travail élève

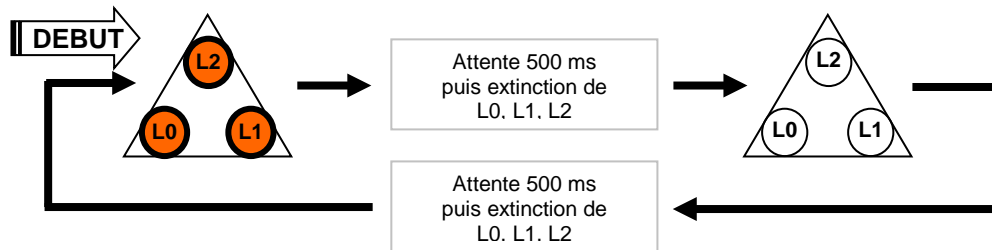
1. Consulter la description des programmes dans les fiches suivantes et choisir celui que tu souhaites essayer.
2. Lancer le logiciel PICAXE Logicator et ouvre ton programme (chaque programme est accessible à partir du dossier « Automatismes niveau 1 » du CD Rom **CD-FP**).  
Convertir le diagramme en Basic (F5).
3. Transférer le programme dans le FlashProg (F5) et vérifier que son exécution correspond à sa description.

## FICHE N°1 : programme F1-CLIGNO.plf

**But du programme** : faire clignoter simultanément L0, L1 et L2 (effet clignoteur).

**Notion de programmation abordée** : activation / désactivation de plusieurs sorties, utilisation d'un temps d'attente.

**Synoptique** :



**Commentaire** :

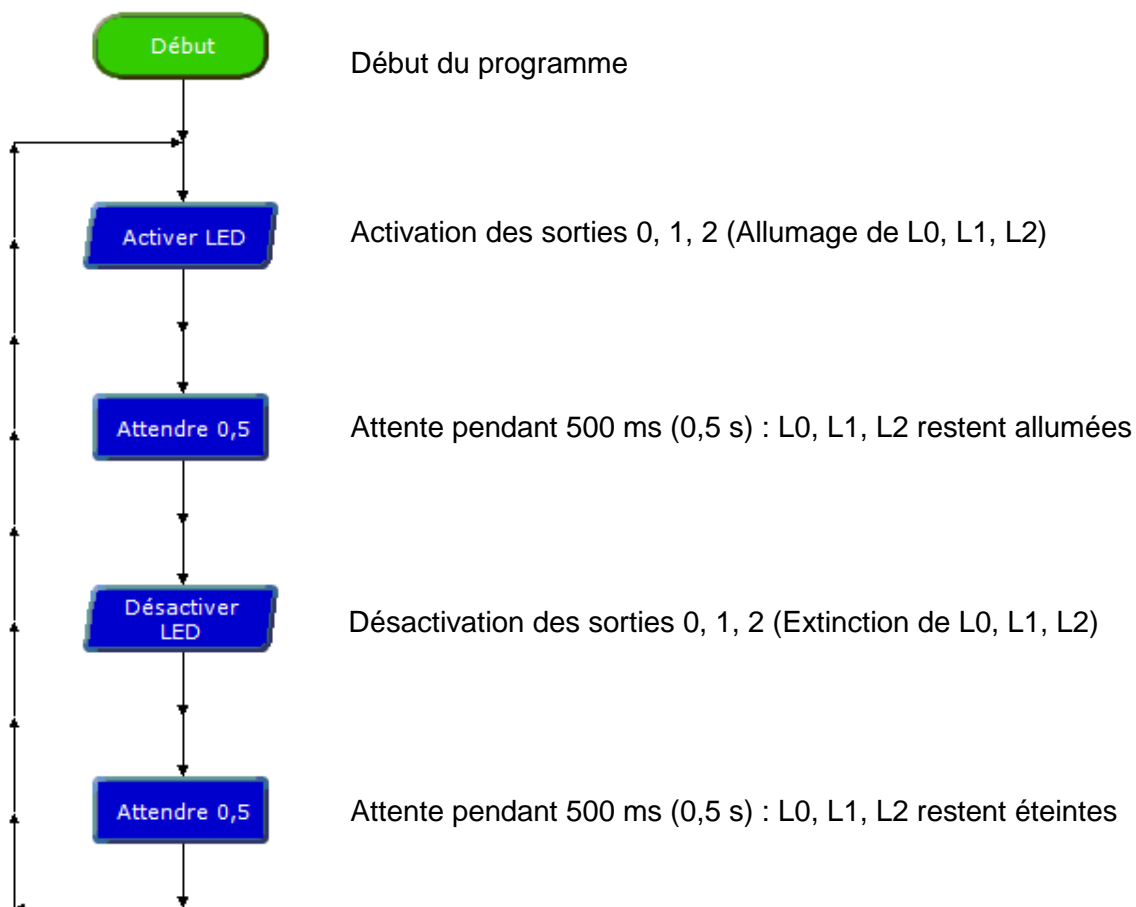
Dès la mise sous tension, L0, L1 et L2 sont allumées.

Au bout de 0,5 seconde, elles sont éteintes.

Un temps d'attente de 0,5 seconde est introduit avant de les allumer de nouveau.

Le programme reboucle indéfiniment sur lui-même.

**Diagramme de programmation** :

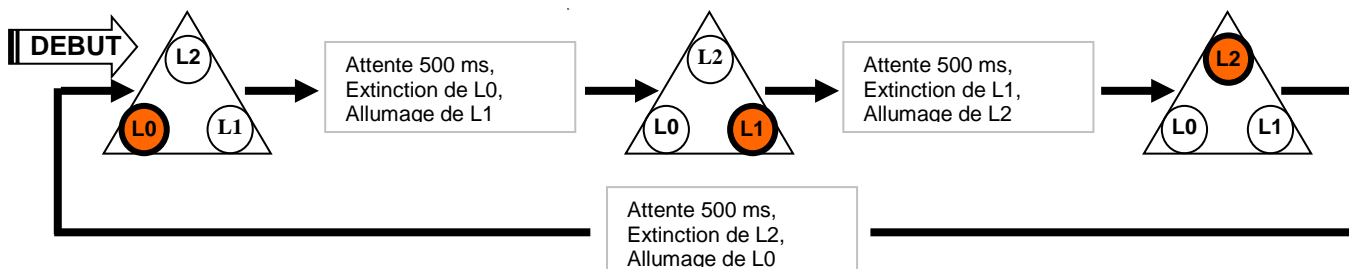


## FICHE N°2 : programme F2-CHENILLARD.plf

**But du programme :** faire clignoter successivement L0, L1 et L2 (effet chenillard).

**Notion de programmation abordée :** activation / désactivation de plusieurs sorties, utilisation d'un temps d'attente.

**Synoptique :**



**Commentaire :**

Dès la mise sous tension L0 est allumée, L1, L2 sont éteintes.

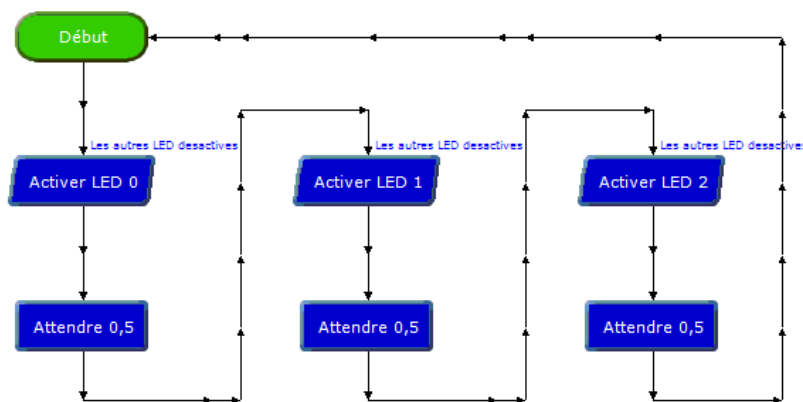
Au bout de 500 ms L0 est éteinte et L1 est allumée.

Au bout de 500 ms L1 est éteinte et L0 est allumée.

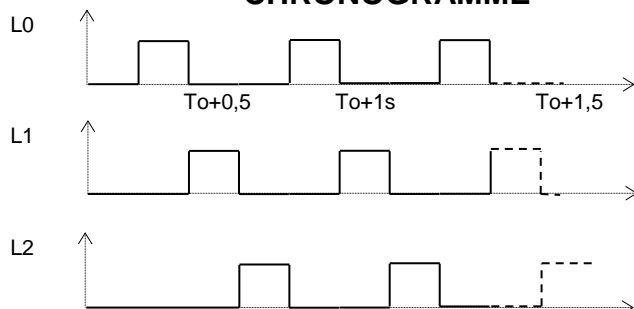
Un temps d'attente de 500ms est introduit avant d'allumer de nouveau L0.

Le programme reboucle indéfiniment sur lui-même ; cette séquence donne l'effet d'un chenillard.

**Diagramme de programmation :**



### CHRONOGRAMME



**Note :** A la mise sous tension du module (première exécution du programme), toutes les sorties sont initialisées à l'état bas (L0, L1, L2 sont éteintes).

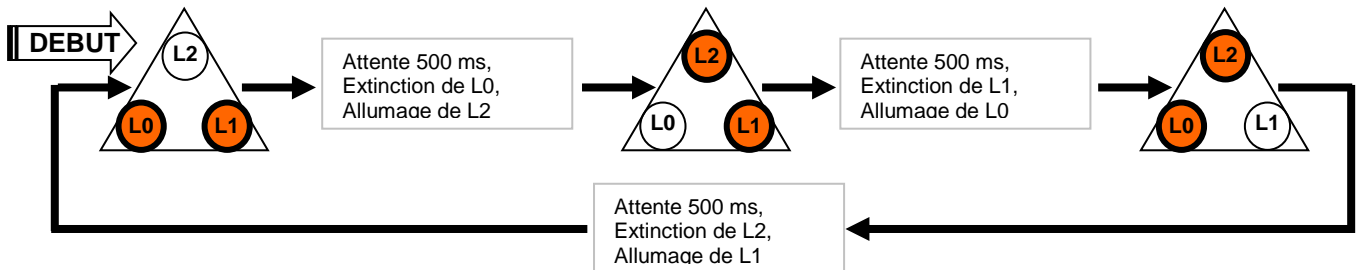
Une sortie qui est activée à l'état haut (ex. « high0 ») reste active jusqu'à ce que le programme rencontre une instruction qui la désactive (ex. low0).

### FICHE N°3 : programme F3-BATON TOURNANT.plf

**But du programme :** allumer momentanément une paire de LED et faire une rotation de cet effet en allumant momentanément la paire adjacente.

**Notion de programmation abordée :** activation / désactivation de plusieurs sorties, utilisation d'un temps d'attente.

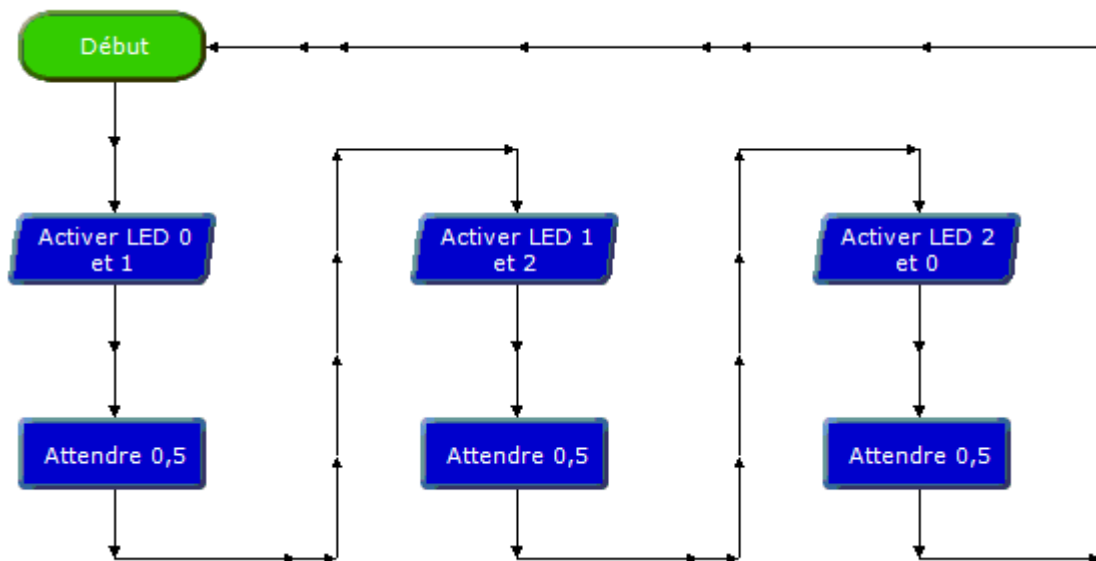
**Synoptique :**



**Commentaire :**

Dès la mise sous tension L0 et L1 sont allumées, L2 est éteinte.  
 Au bout de 500 ms L0 est éteinte et L2 est allumée.  
 Au bout de 500 ms L1 est éteinte et L0 est allumée.  
 Le programme reboucle indéfiniment sur lui-même.

**Diagramme de programmation :**



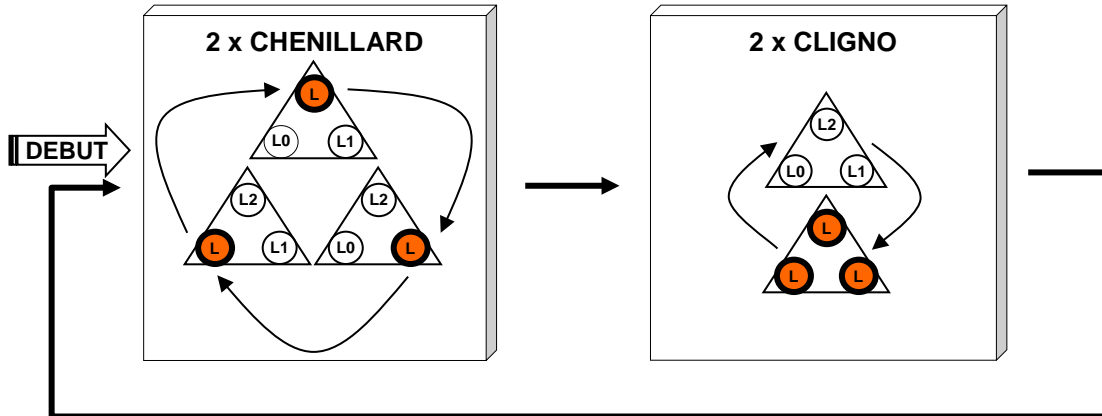


## FICHE N°4 : programme F4-CLIGNO-CHENILLARD.p1f

**But du programme** : alterner deux séquences que l'on répète deux fois chacune (chenillard suivi de clignotement simultané des LED).

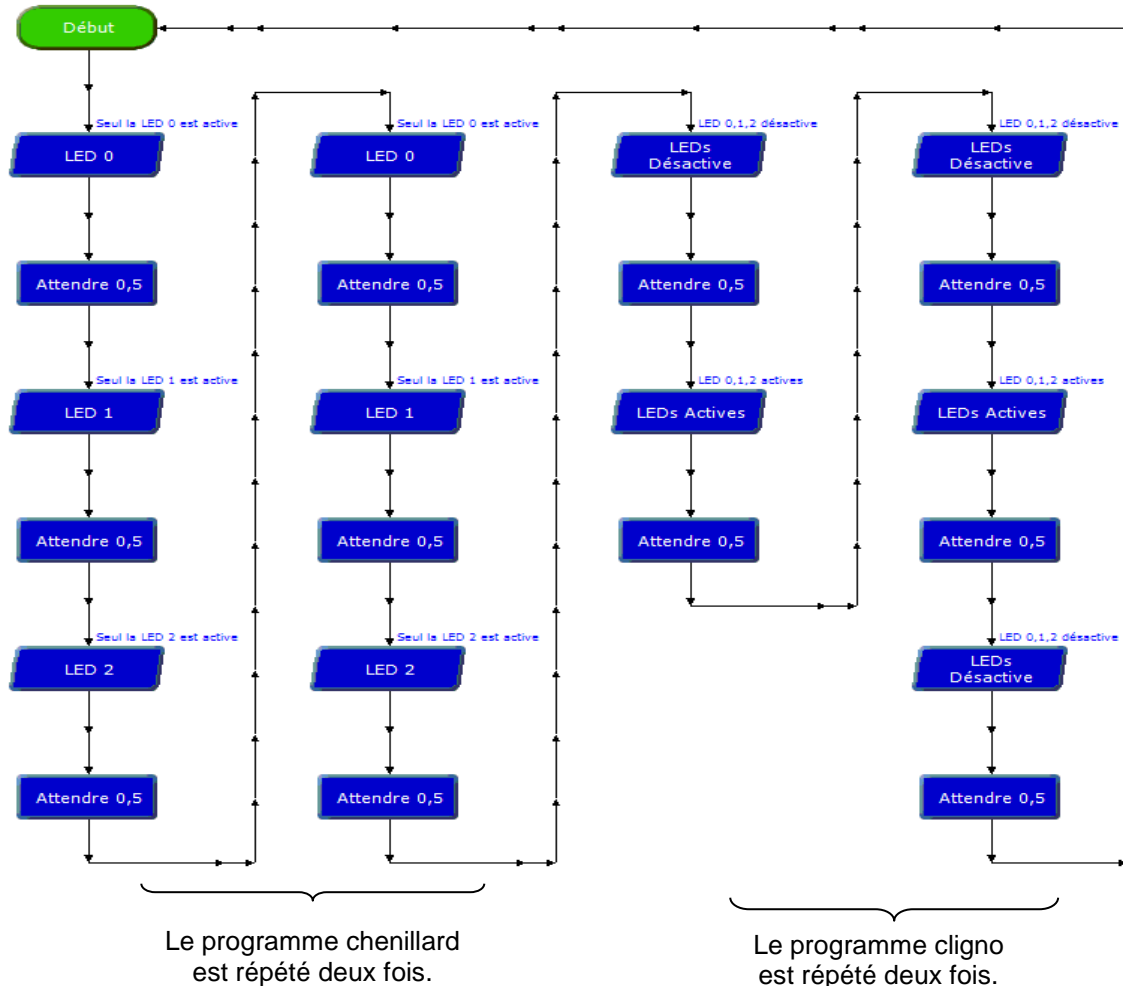
**Notion de programmation abordée** : activation / désactivation de plusieurs sorties, utilisation d'un temps d'attente, répétition de séquences.

**Synoptique** :



**Commentaire** : on remarque ici que l'on répète plusieurs fois l'exécution d'instructions identiques afin de répéter des séquences. On découvrira plus loin que l'on peut optimiser l'écriture d'un programme en utilisant des instructions plus appropriées (dans le but de simplifier et d'économiser de la mémoire de programme).

**Diagramme de programmation** :

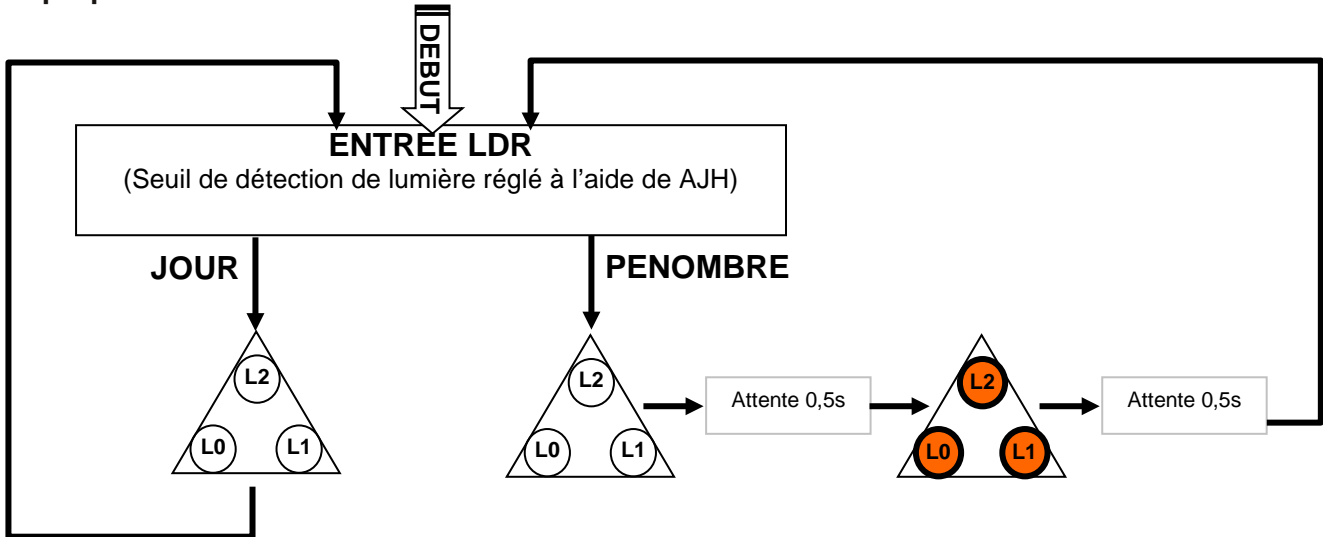


## FICHE N°5 : programme F5-DEMAR AUTO NUIT.plf

**But du programme :** faire clignoter simultanément L0, L1 et L2 dès lors qu'un seuil de pénombre est atteint.

**Notion de programmation abordée :** test d'une l'entrée (LDR).

**Synoptique :**

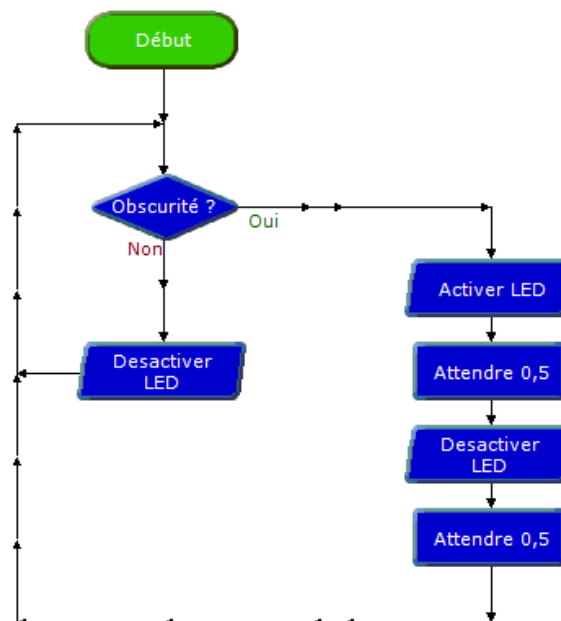


**Commentaire :**

Dès la mise sous tension l'état de l'entrée LDR est testé. Tant que le niveau lumineux ambiant est élevé (jour) L0, L1, L2 restent éteintes. Dès que le niveau lumineux ambiant est faible (pénombre) L0, L1, L2 clignotent.

**Note :** le réglage du seuil de détection de la pénombre est fait à l'aide du résistor ajustable AJH.

**Diagramme de programmation :**

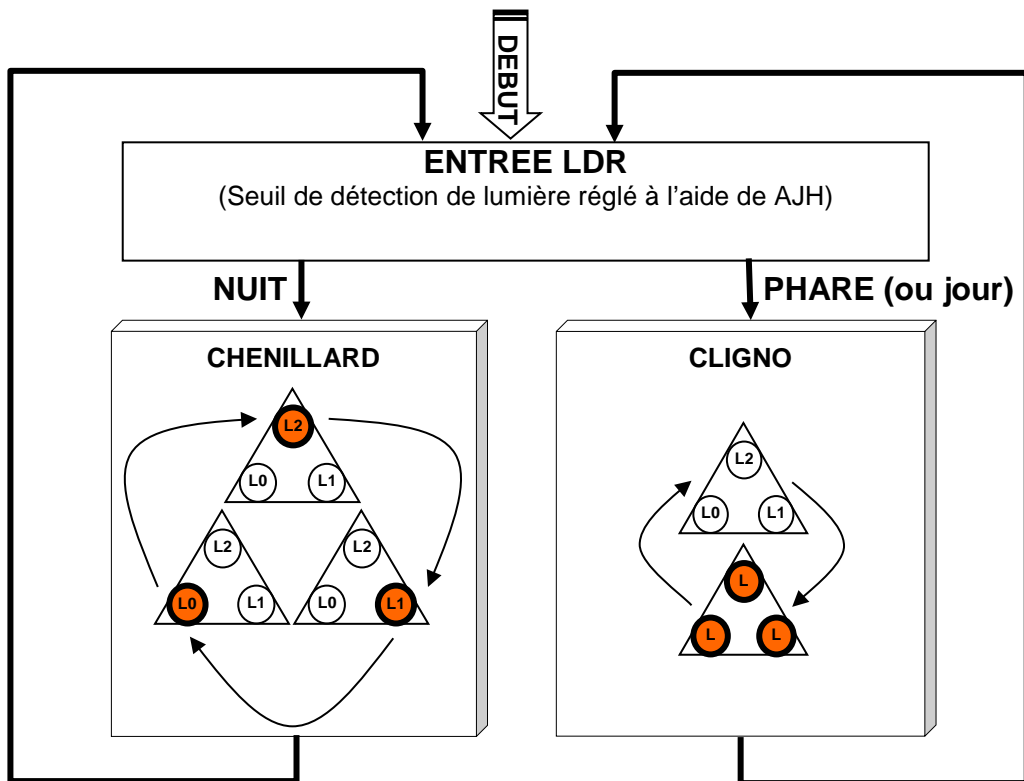


## FICHE N°6 : programme F6-DETEC PHARE.plf

**But du programme :** exécuter une séquence « chenillard » lorsqu'il fait nuit, basculer sur une séquence clignotement dès que la LDR détecte la présence de lumière.

**Notion de programmation abordée :** test d'une l'entrée (LDR).

**Synoptique :**

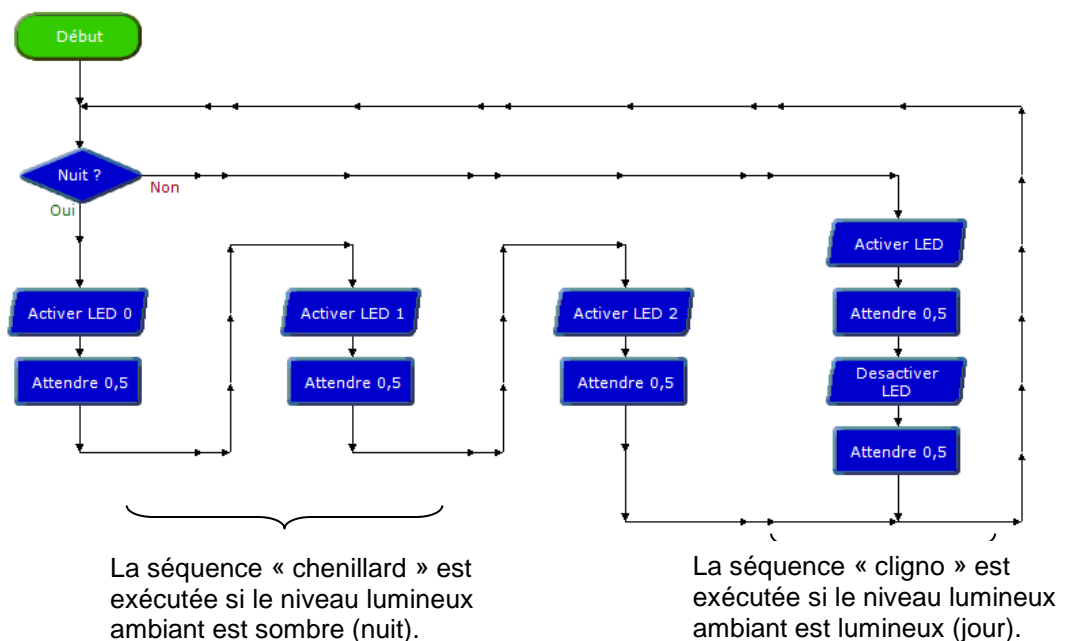


**Commentaire :**

Dès la mise sous tension l'état de l'entrée LDR est testé. Tant que le niveau lumineux ambiant est faible (nuit) le programme boucle sur l'effet chenillard. Dès que le niveau lumineux ambiant atteint un certain seuil (présence de lumière) le programme boucle sur l'effet clignotement.

Note : le réglage du seuil de détection de la lumière est fait à l'aide de AJH.

**Diagramme de programmation :**

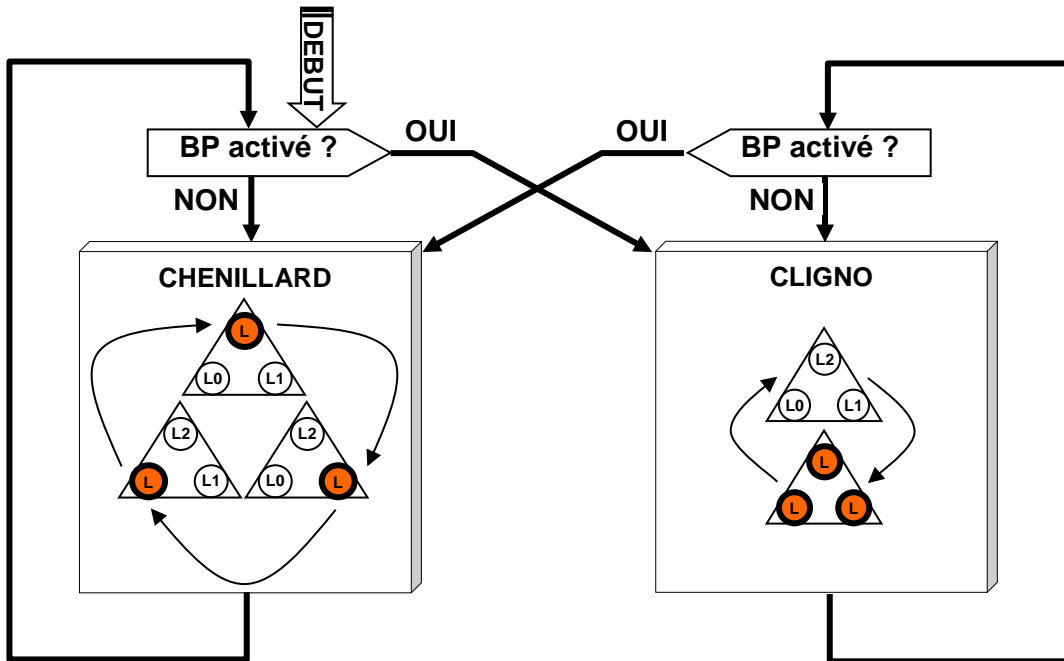


## FICHE N°7 : programme F7-SELECTION PAR BP.plf

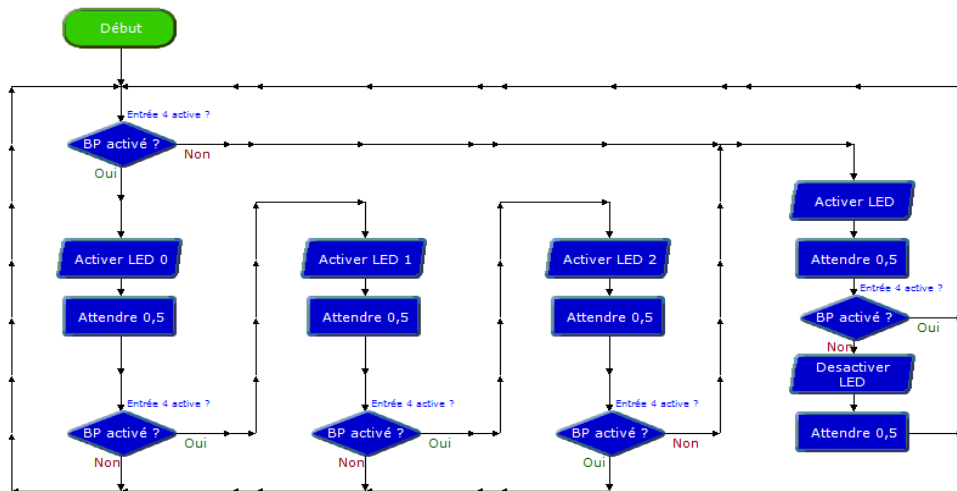
**But du programme :** alterner la séquence chenillard avec la séquence clignotement simultanée des LED dès que l'on active le bouton-poussoir.

**Notion de programmation abordée :** test d'une l'entrée (BP).

**Synoptique :**



**Diagramme de programmation :**



**Commentaire :**

Il s'agit ici de tester l'état du bouton-poussoir et de basculer vers la séquence CLIGNO ou vers la séquence CHENILLARD.

L'état du bouton-poussoir est testé aussi souvent que possible afin de ne pas rater l'instant où il sera activé (risque d'appui pendant l'exécution d'une instruction).

Lorsque qu'il est activé (BP activé est vrai), on attend 10 ms que son état se stabilise.

En effet, lorsqu'il est activé, il génère une multitude d'impulsions (rebonds) qui pourraient être interprétées comme des appuis multiples.

A partir de l'instant où il est activé, on attend qu'il soit relâché avant de basculer vers une séquence différente.

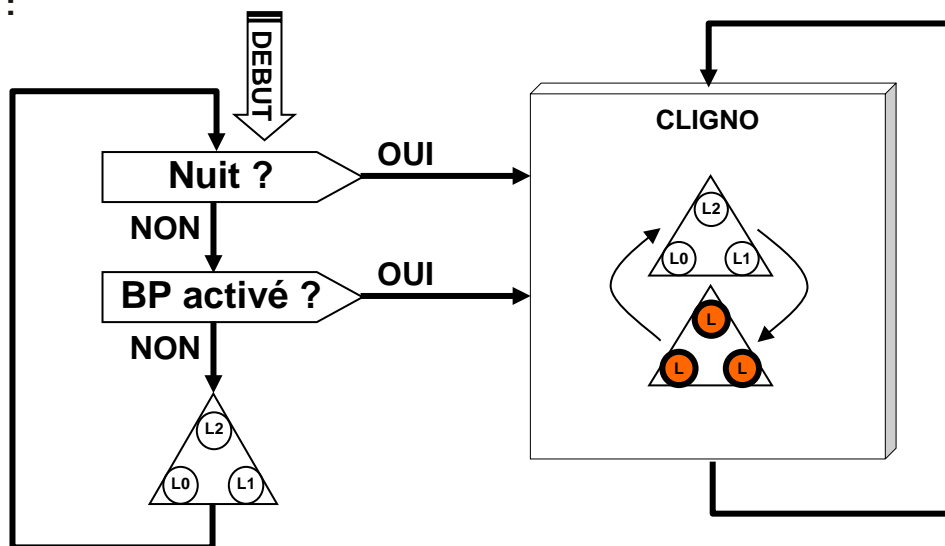
## FICHE N°8 : programme F8-DEMAR AUTO LDR-BP.plf

**But du programme :** faire démarrer une séquence de clignotement automatiquement à la tombée de la nuit (économiser l'énergie lorsqu'il fait jour), pouvoir forcer le démarrage du système lorsque l'on active le bouton-poussoir (forcer le démarrage lorsqu'il fait jour).

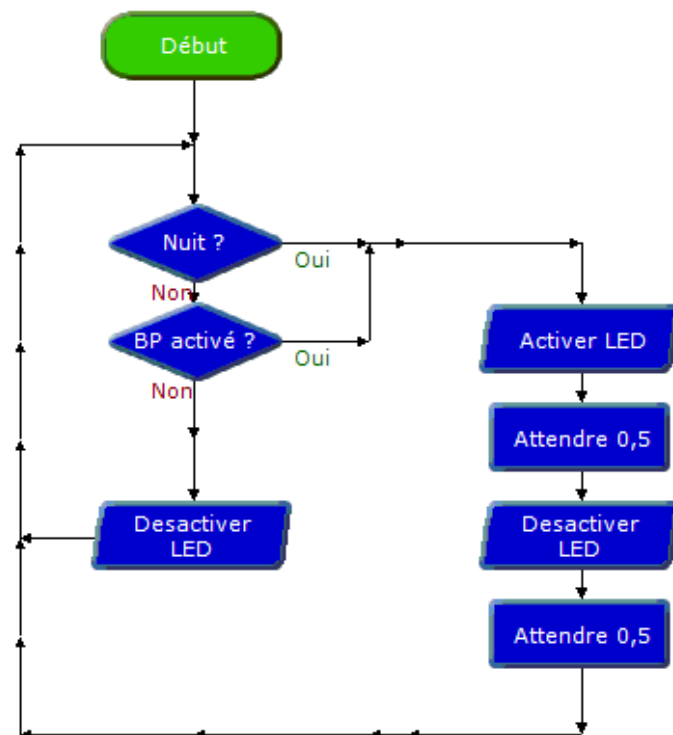
**Notion de programmation abordée :** mise en œuvre d'un programme principal qui teste l'état de l'entrée LDR et de l'entrée bouton-poussoir.

Appel à un programme qui gère une séquence d'allumage des LED.

**Synoptique :**



**Diagramme de programmation :**



**Commentaire :**

La séquence « CLIGNO » démarre lorsque le niveau lumineux ambiant est faible (nuit) ou lorsque l'on appuie sur le bouton-poussoir.

Lorsqu'il fait jour, l'entrée LDR voit un niveau logique bas (« Nuit » est faux) ; le programme teste alors l'entrée bouton-poussoir.

Tant que le bouton-poussoir n'est pas activé (« BP activé ? » est faux), le programme reboucle sur le test de la LDR.

Dans tous les autres cas, la séquence « CLIGNO » est exécutée.

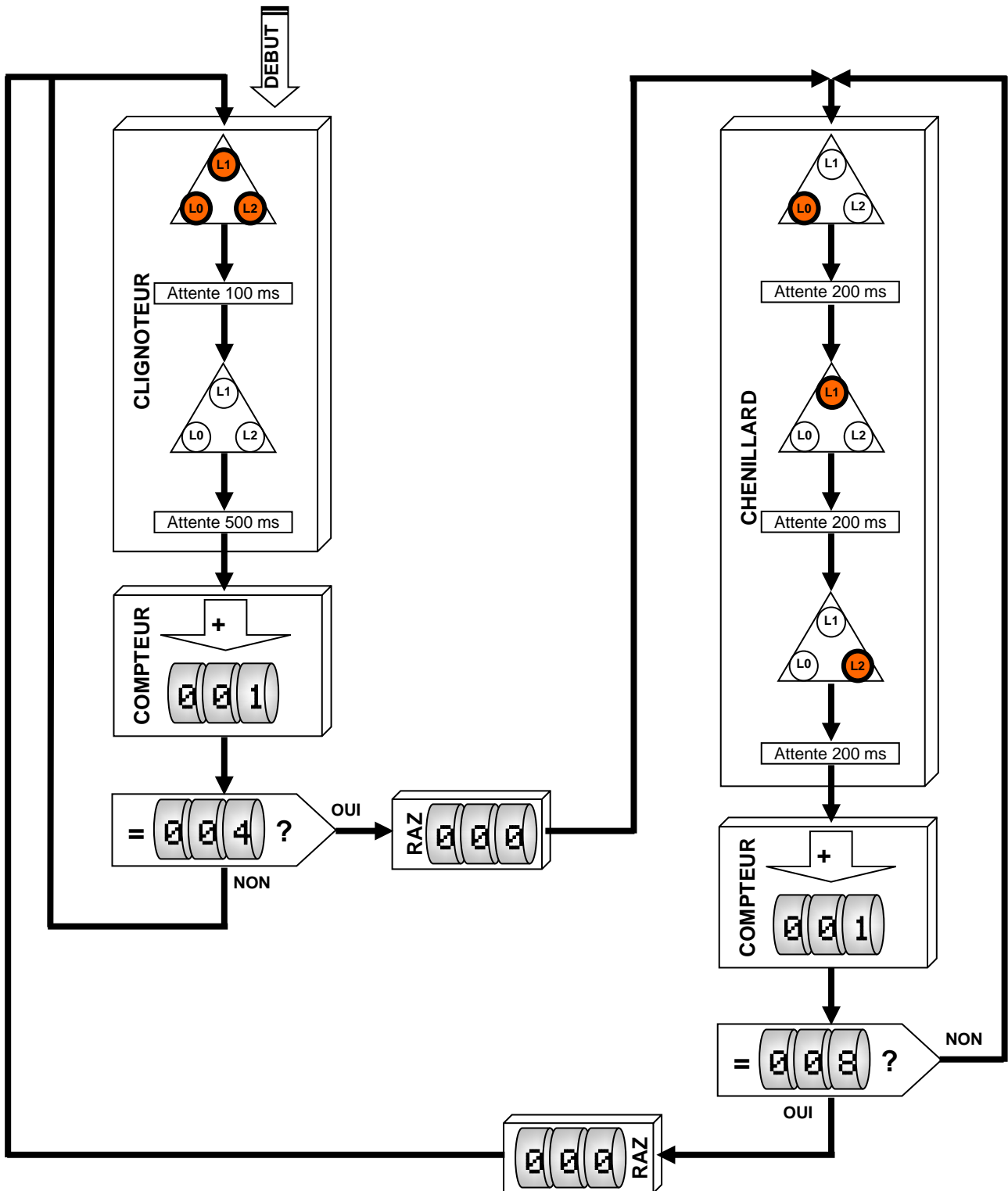
Dès lors, le programme ne teste plus ni l'état de la LDR ni celui du bouton-poussoir et reboucle indéfiniment sur l'exécution de cette séquence « CLIGNO ».

La séquence « CLIGNO » s'achève alors uniquement par la mise hors tension du module.

## FICHE N°9 : programme F9-REPET 2 SEQUENCES.plf

**But du programme :** exécuter une première séquence quatre fois de suite puis une autre séquence huit fois de suite, reboucler sur la première séquence.

**Notion de programmation abordée :** répéter une séquence plusieurs fois. Définir une variable de comptage interne au programme (variable locale), tester la valeur de cette variable.



## FICHE N°9 (Suite): programme F9-REPET 2 SEQUENCES.plf

### Commentaire :

Dès la mise sous tension, le programme « CLIGNO » est exécuté.  
Après chaque exécution de ce programme, un compteur nommé **A** est incrémenté de un en un.  
La valeur de ce compteur est testée après chaque itération du programme ; dès que celui-ci a été exécuté quatre fois, le programme « CHENILLARD » est alors exécuté.  
De la même manière, dès que celui-ci a été exécuté huit fois, le programme « CLIGNOTEUR » est de nouveau exécuté . . .

**Note** : la variable **A** est réinitialisé à zéro à chaque fois que sa valeur atteint quatre (ou huit) de telle sorte que le programme reboucle indéfiniment.

Une variable est une donnée qui est mémorisée dans une « case mémoire de 8 bits.

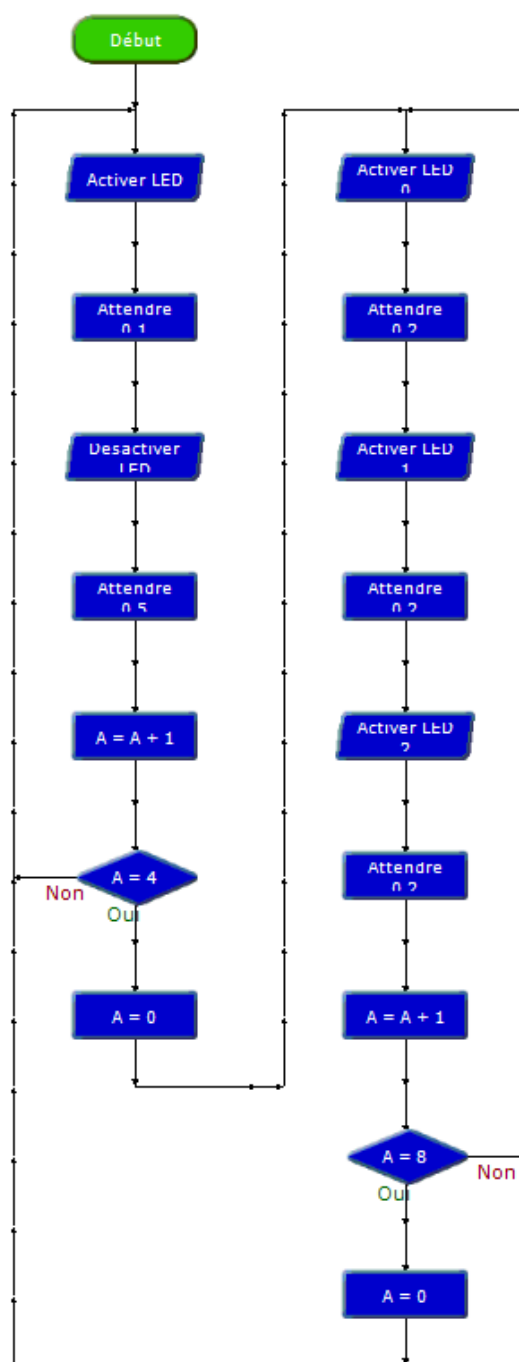
La valeur maximale d'une variable locale est 255 ( $2^8-1$ ).

Ici, la variable locale est incrémentée après chaque itération d'une séquence d'allumage des LED.

Le test de la variable **A** permet de déterminer le nombre d'itérations de chaque séquence.

### Diagramme de programmation

(programme REPET 2 SEQUENCES.plf :

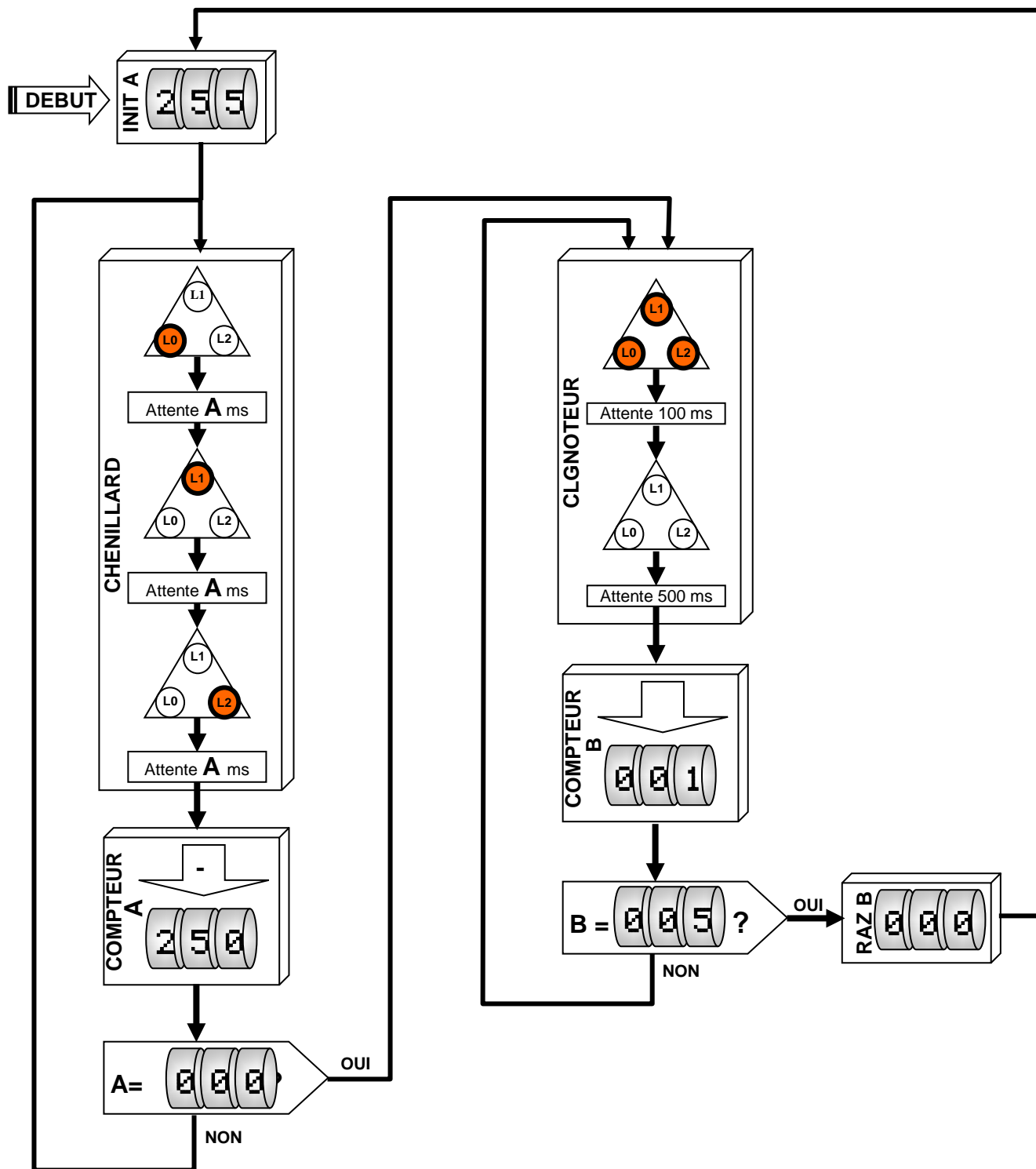


## FICHE N°10 : programme F10-ACCELERATION.plf

**But du programme** : provoquer un effet chenillard qui s'accélère progressivement et qui est suivi par un effet clignotement simultané des LED qui est lui-même répété cinq fois.

**Notion de programmation abordée** : intégrer un temps de pause qui est fonction d'une variable dont on fait évoluer la valeur. Utiliser une deuxième variable pour compter des cycles.

**Synoptique** :





## FICHE N°10 (suite) : programme F10-ACCELERATION.plf

### Commentaire :

Dès la mise sous tension, le programme initialise une première variable nommée **A** avec une valeur de 255 (valeur maximum d'une variable).

Suit une séquence « chenillard » à la fin de laquelle on décrémente de cinq la variable **A**.

Tant que **A** n'a pas atteint la valeur zéro, le programme reboucle sur la séquence chenillard.

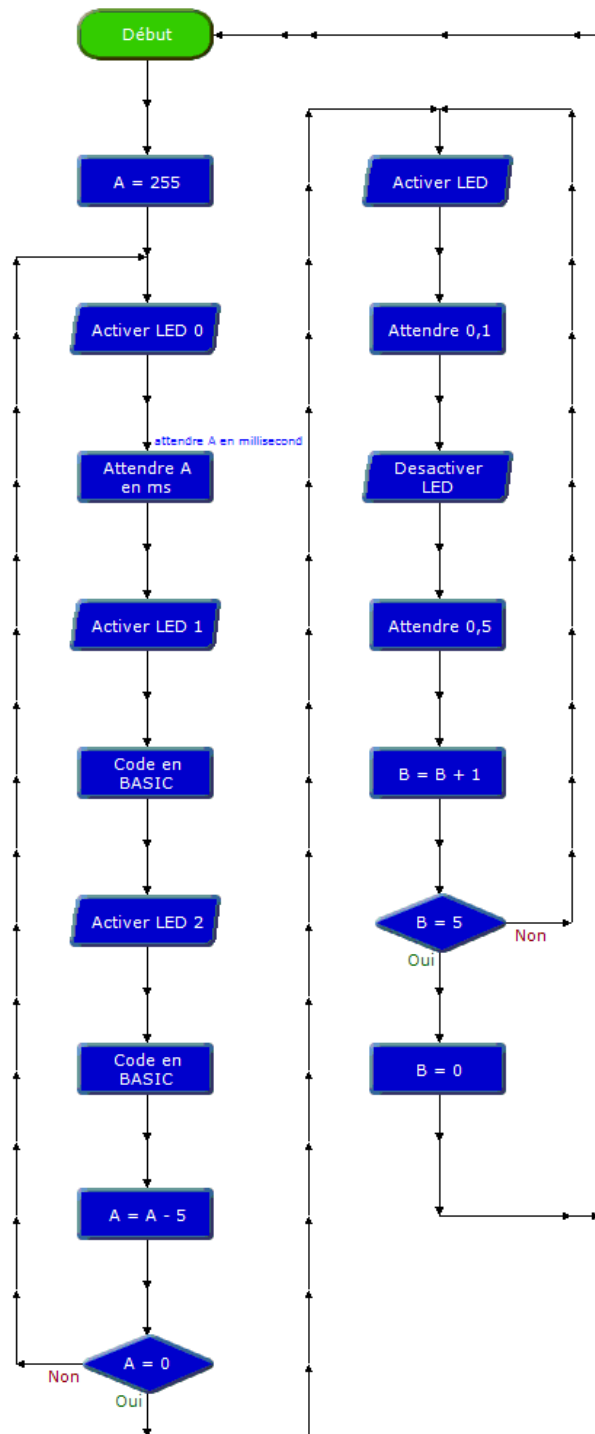
Sinon, le programme enchaîne une séquence « cligno ».

A l'issue de la séquence « cligno », une deuxième variable **B** est incrémentée.

Tant que la variable **B** n'a pas atteint la valeur 5, le programme reboucle sur cette séquence.

Sinon, le programme réinitialise la première variable **B** à 255 et enchaîne de nouveau la séquence « chenillard » . . .

### Diagramme de programmation :

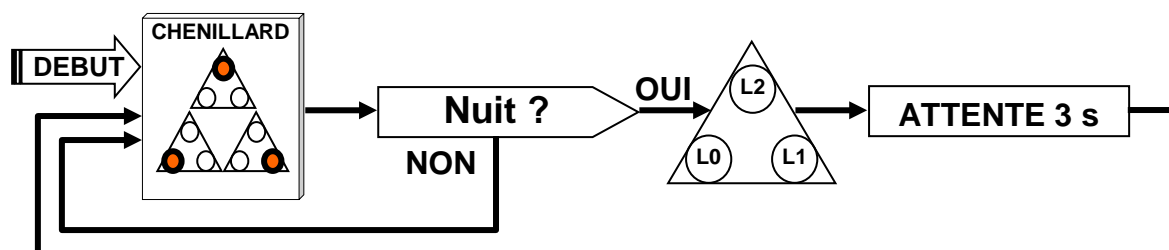


## FICHE N°11 : programme F11-CHENILLARD JOUR-NUIT.plf

**But du programme** : un effet « chenillard » est provoqué une fois toutes les trois secondes lorsqu'il fait jour. Si il fait nuit, l'effet « chenillard » est provoqué en permanence.

**Notion de programmation abordée** : activation / désactivation de plusieurs sorties, utilisation d'un temps d'attente court, utilisation d'un temps d'attente long, test d'une l'entrée (LDR).

**Synoptique** :



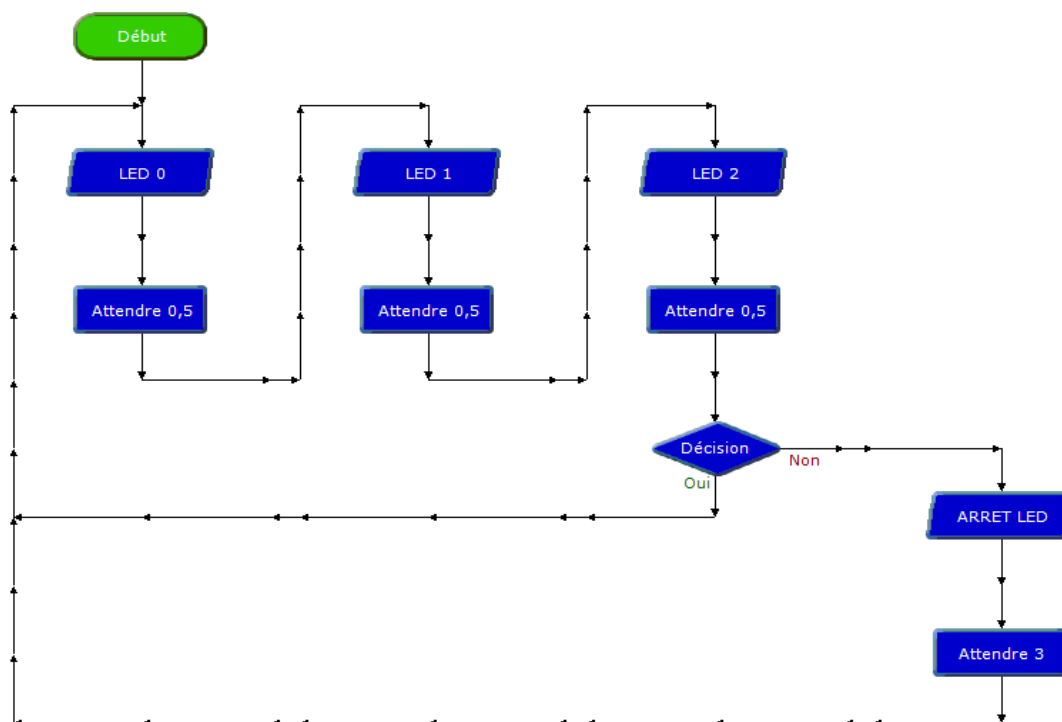
**Commentaire** :

Une séquence « chenillard » est exécutée.

A la fin de cette séquence, le programme vérifie l'état de l'entrée LDR ; s'il y a présence de lumière, le programme attend trois secondes avant de reboucler au début de la séquence.

Sinon, le programme reboucle immédiatement et l'effet chenillard est ininterrompu.

**Diagramme de programmation** :



**Note importante** : si l'on charge un programme qui fait appel à l'instruction « Attendre » dès le début d'un programme, le dialogue entre l'ordinateur et le module lors du chargement d'un nouveau programme risque d'être perturbé et le chargement peut échouer.

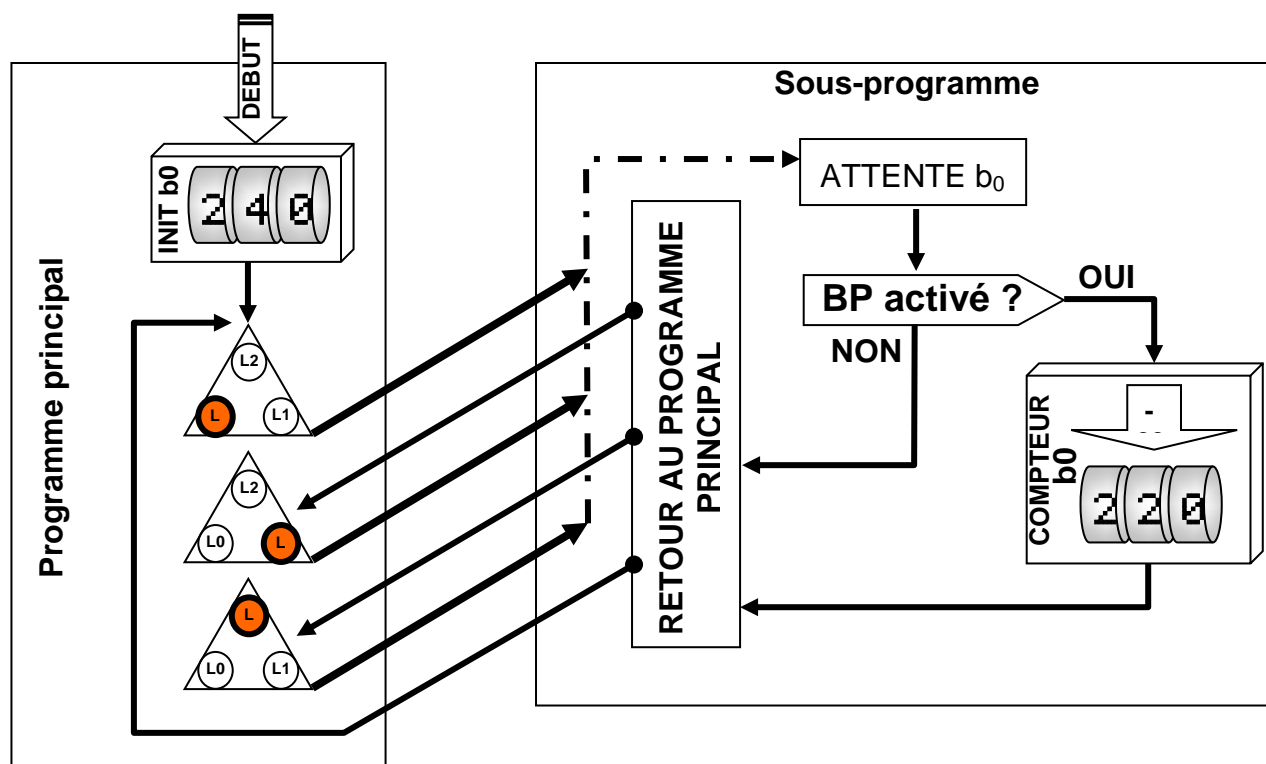
Il convient alors pour parvenir à charger son nouveau programme de mettre le module hors tension, de lancer le chargement (l'ordinateur essaye de dialoguer avec le module), puis de mettre immédiatement le module sous tension.

## FICHE N°12 : programme F12-ACCELERATION PAR BP.plf

**But du programme** : accélérer la vitesse d'une séquence chenillard lorsque l'on active le bouton-poussoir.

**Notion de programmation abordée** : mise en œuvre d'un programme principal et d'un sous-programme, utilisation de la variable A.

**Synoptique** :

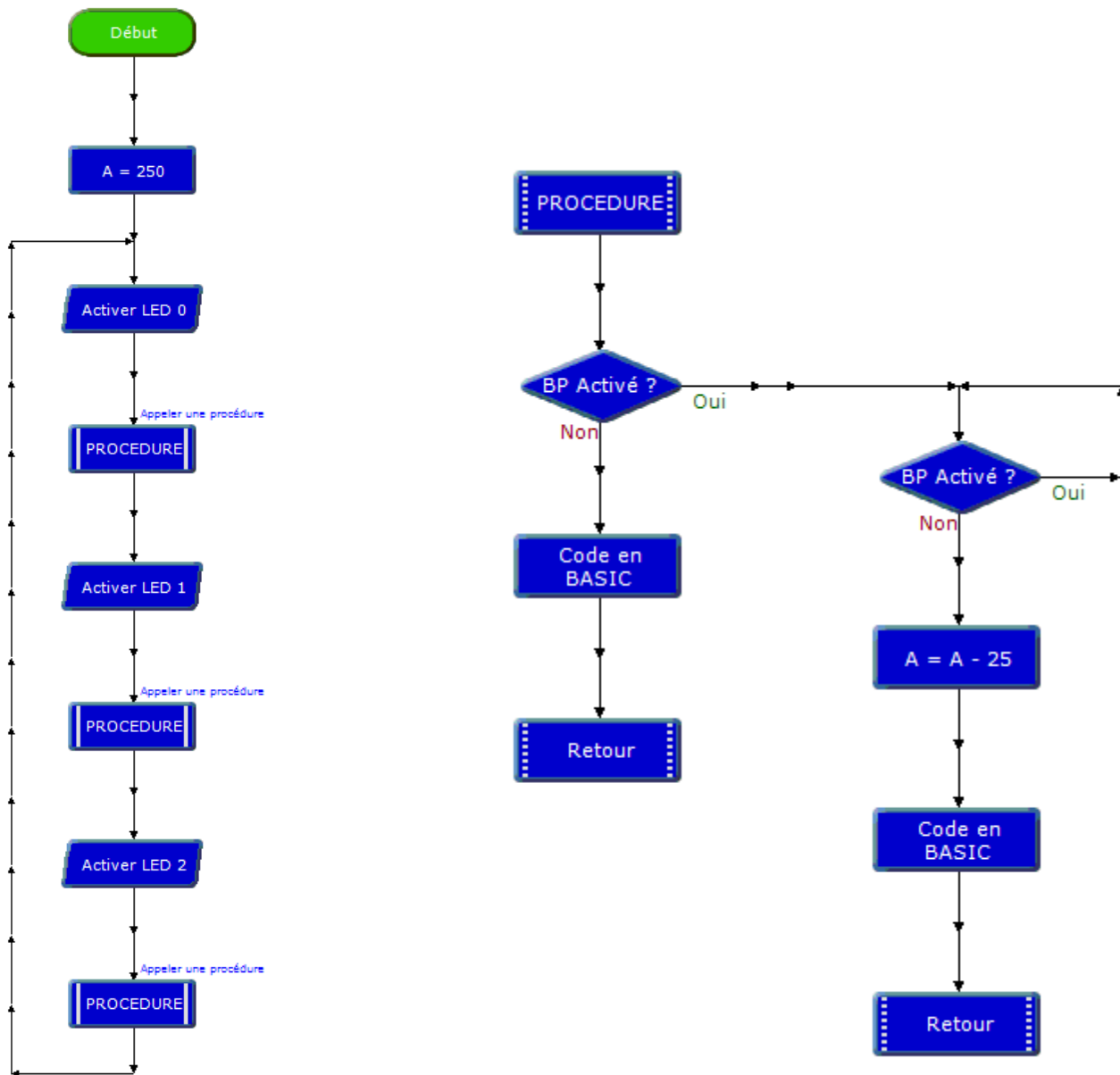


### Commentaire

Le programme principal exécute une séquence chenillard ponctuée par un appel à un sous-programme. Le sous-programme introduit un temps d'attente qui détermine le temps d'allumage (ou d'extinction des LED). Il teste par ailleurs l'état du bouton-poussoir et décrémente la valeur du temps d'attente à chaque fois que le bouton-poussoir est activé.

## FICHE N°12 (suite) : programme F12-ACCELERATION PAR BP.plf

Diagramme de programmation :



Le programme principal gère l'allumage ou l'extinction des LED.

L'instruction « PROCEDURE » introduite dans le programme principal demande au programme d'exécuter le sous-programme « PROCEDURE ».

Le sous-programme gère le temps d'attente (« pause b0 ») qui contrôle la durée d'allumage ou d'extinction des LED. Il teste par ailleurs l'état du bouton-poussoir et incrémente la variable b0 à chaque fois que le bouton poussoir est enfoncé puis relâché.

L'instruction « Retour » marque la fin du sous-programme et le retour au programme principal.

Le programme principal reprend son exécution à partir de l'instruction qui suit immédiatement l'instruction « PROCEDURE ».

**Note :** « Appeler Procédure » signifie aller au sous-programme nommé « Procédure ». « Retour » signifie retour de sous-programme.

## AUTOMATISME NIVEAU 2

### Modifier un programme

L'élève intervient sur un programme existant et modifie certains paramètres selon une consigne. (chaque programme est accessible à partir du dossier « Automatismes niveau 2 » du CD **CD-FP**).

#### Liste des fiches élèves :

	Programme à charger	Consigne
FICHE N°21	F21-FLASH COURTS.plf	Réduire la durée d'allumage des LED.
FICHE N°22	F22-ACCELERER CHENILLARD.plf	Accélérer la vitesse de défilement du chenillard
FICHE N°23	F23-INVERSER SENS ROTATION.plf	Inverser le sens de rotation du chenillard
FICHE N°24	F24-MODIFIER REPETITION.plf	Répéter l'exécution du programme clignoteur cinq fois et celle du programme chenillard vingt fois.
FICHE N°25	F25-INVERSER MODE JOUR-NUIT.plf	Inverser le mode de fonctionnement jour / nuit.
FICHE N°26	F26-DEMAR AUTO NUIT.plf	Démarrage automatique lorsqu'il fait nuit.

#### Travail élève :

1. Lancer le logiciel PICAXE Logicator et ouvrir le diagramme d'un des diagrammes de la liste ci-dessus.

Convertir le diagramme en Basic (F5).

Transférer le programme dans le FlashProg (F5) et vérifier que son fonctionnement correspond à la description du programme.

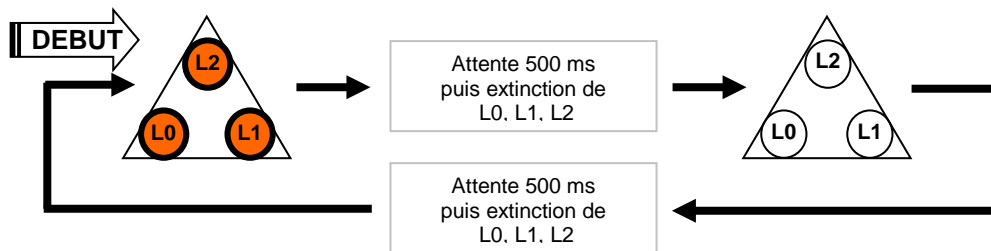
2. Modifier le diagramme selon les consignes (sauver éventuellement le diagramme sous un nouveau nom).
3. Convertir le nouveau diagramme en Basic (F5).

Transférer le programme modifié dans le FlashProg (F5) et vérifier que son comportement est conforme à la consigne.

## FICHE N°21 : programme F21-FLASHS COURTS.plf

**Description du programme :** ce programme fait clignoter simultanément L0, L1 et L2 (effet clignoteur).

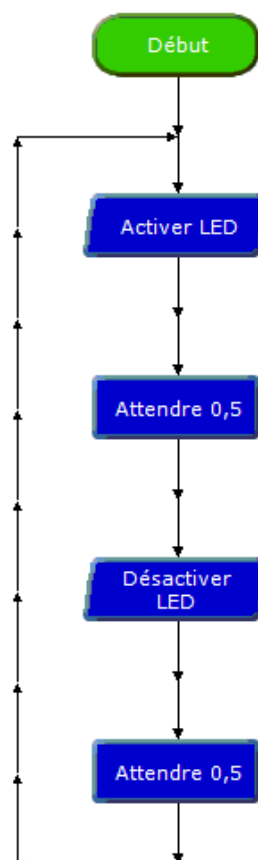
**Synoptique :**



**Consignes :**

- A** - Entourer sur le diagramme ci-dessous l'instruction qui détermine la durée pendant laquelle les LED sont allumées.
- B** - Modifier le diagramme à l'écran en réduisant de moitié le temps d'allumage des LED de telle sorte que les « flashs » soient plus courts. Charger le programme dans le FlashProg et vérifier que le temps d'allumage des LED est plus court. Quel peut être l'intérêt de réduire le temps d'allumage des LED ?
- C** - Modifier le diagramme à l'écran en réduisant le temps d'allumage des LED à la valeur la plus faible possible (soit 1ms). Charger le programme dans le FlashProg ; qu'observe-t-on alors ?

**Diagramme de programmation :**



## FICHE N°21 : FLASHS COURTS.p1f

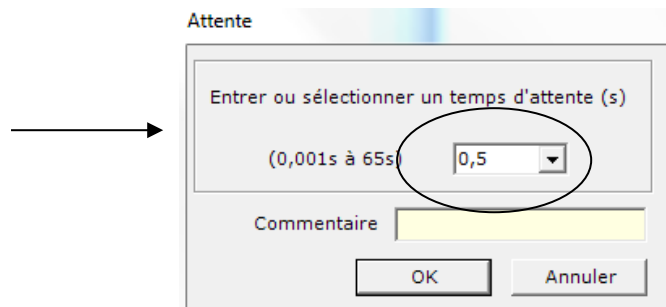
CORRIGÉ

On constate d'après le diagramme que les LED s'allument pendant  $\frac{1}{2}$  seconde chaque seconde. Les temps d'allumage des LED est déterminé par l'instruction « pause 500 ». A chaque seconde les LED s'allument pendant  $\frac{1}{2}$  seconde. La période du cycle est de 1s.

Pour modifier la valeur d'un bloc d'un diagramme de programmation, il faut double cliquer sur le bloc en question.



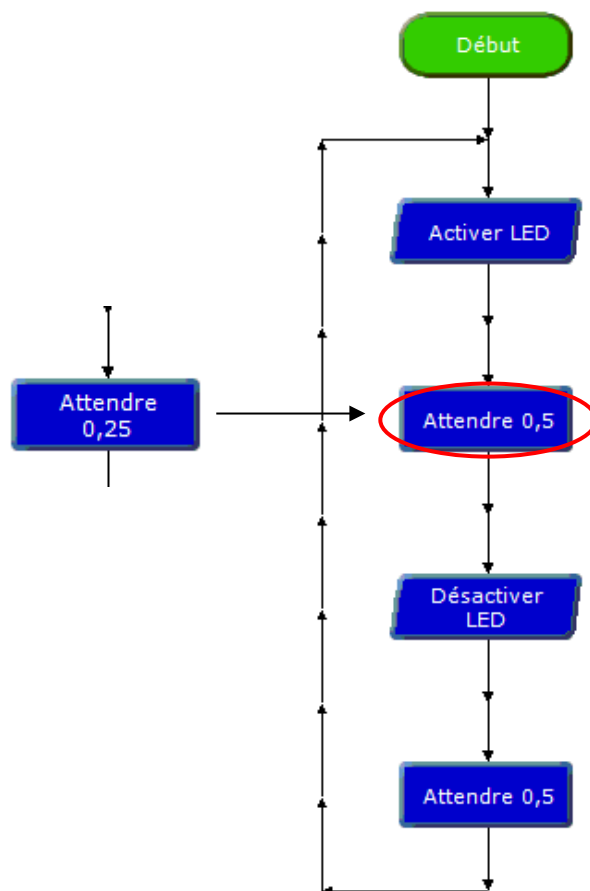
L'information de la fenêtre indique le temps à attendre et permet de saisir une nouvelle valeur.



**A** - L'instruction qui détermine le temps d'allumage des LED est l'instruction « Attendre 0,5 » située immédiatement après les instructions qui provoquent l'allumage des LED. Dès que les LED sont allumés, le programme attend 0,5 s avant d'exécuter les instructions qui vont éteindre les LED.

**B** - Réduire de moitié le temps d'allumage des LED permet de diminuer la consommation du module et améliore donc l'autonomie du FlashProg.

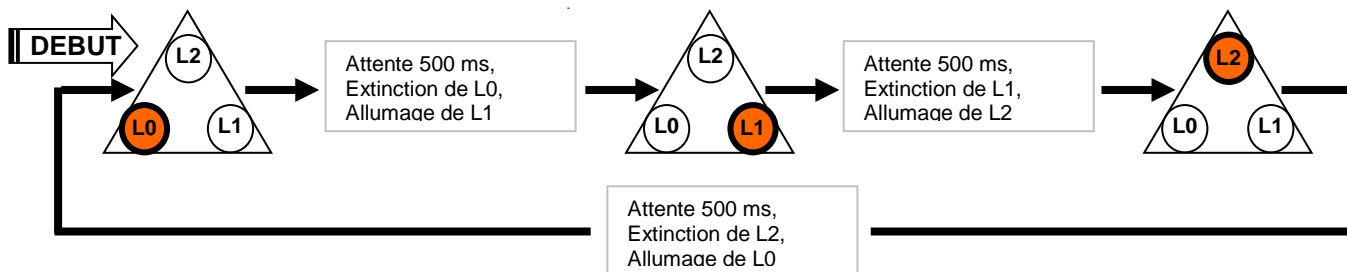
**C** - Lorsque l'on réduit le temps d'allumage des LED à la valeur minimum autorisée par le système (1 ms), on observe une diminution importante de puissance des flashes. Trop réduire le temps d'allumage des LED nuit à l'efficacité de l'effet visuel. La durée idéale d'allumage des LED est un compromis entre l'effet souhaité pour répondre aux exigences du FlashProg (être visible de loin) et conserver une autonomie de fonctionnement acceptable.



## FICHE N°22 : programme F22-ACCELERER CHENILLARD.p1f

**Description du programme :** faire clignoter successivement L0, L1 et L2 en provoquant un effet chenillard.

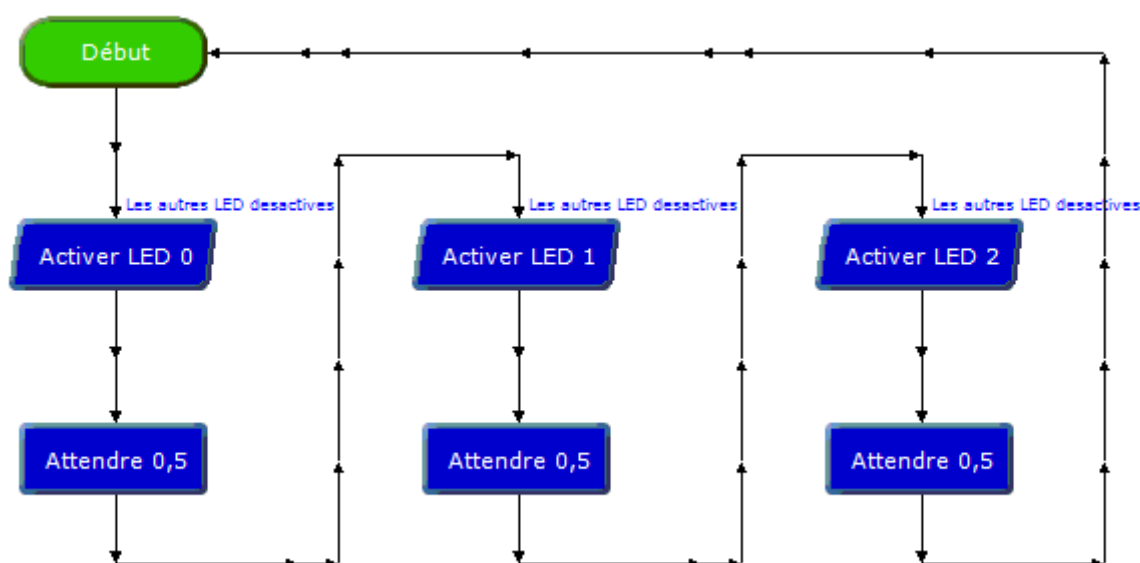
**Synoptique :**



**Consignes :**

- A** - Entourer sur le diagramme ci-dessous la (les) instruction(s) qui détermine(nt) la vitesse de défilement du chenillard.
- B** - Modifier le diagramme à l'écran afin de doubler la vitesse du chenillard.  
Charger le programme dans le FlashProg et vérifier que le chenillard tourne deux fois plus rapidement.
- C** - Modifier le diagramme à l'écran afin d'accélérer la vitesse du chenillard à la valeur maximum autorisée par le système (soit 1ms).  
Charger le programme dans le FlashProg ; qu'observe-t-on alors ?

**Diagramme de programmation :**

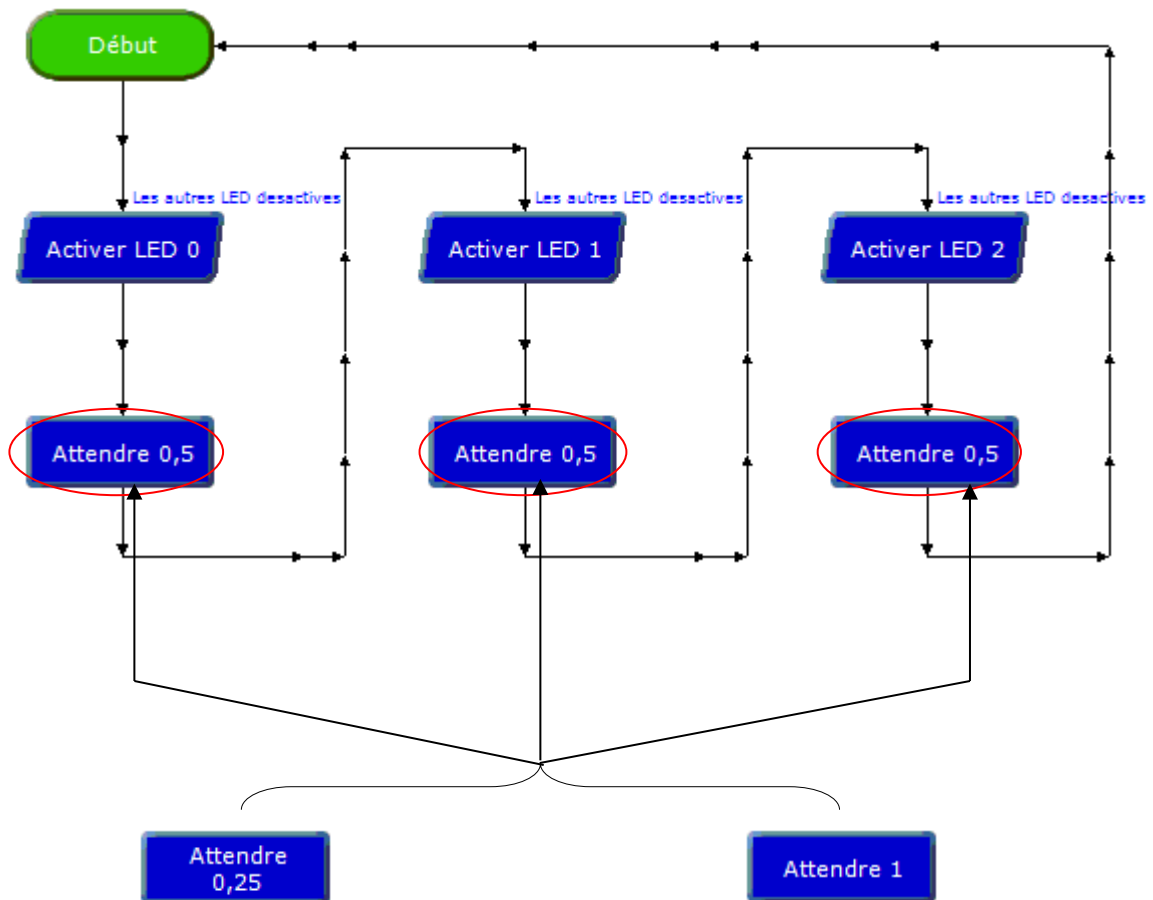




## FICHE N°22 : ACCELERER CHENILLARD.plf

CORRIGÉ

**A** - La vitesse de défilement du chenillard est déterminée par les trois instructions « Attendre 0,5 ». Les LED s'allument successivement pendant 0,5 s. La période d'un cycle complet est de 1,5 s.



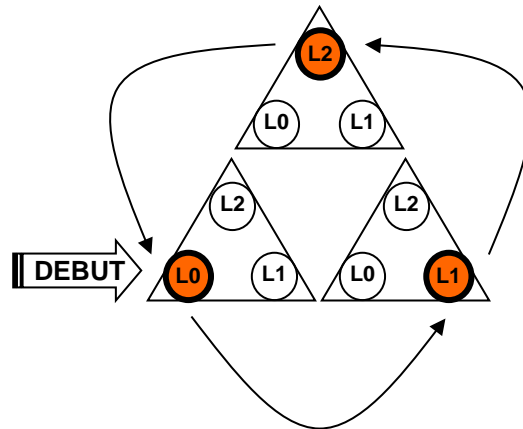
**B** - Pour doubler la vitesse de défilement du chenillard, il faut diviser par deux le temps d'attente introduit par les trois instructions d'origine « Attendre ».  
Les trois instructions « Attendre xxx » doivent avoir une valeur identique sans quoi l'effet visuel est déséquilibré.  
On divise par deux le temps d'attente introduit par les trois instructions « Attendre xxx » pour doubler la vitesse de défilement du chenillard.

**C** - Si l'on réduit au minimum autorisé le temps d'attente introduit par les trois instructions « Attendre xxx » (soit 1ms), le défilement est si rapide que l'on a l'impression que les LED sont allumées en permanence et l'effet chenillard disparaît.

## FICHE N°23 : programme F23-INVERSER SENS ROTATION.plf

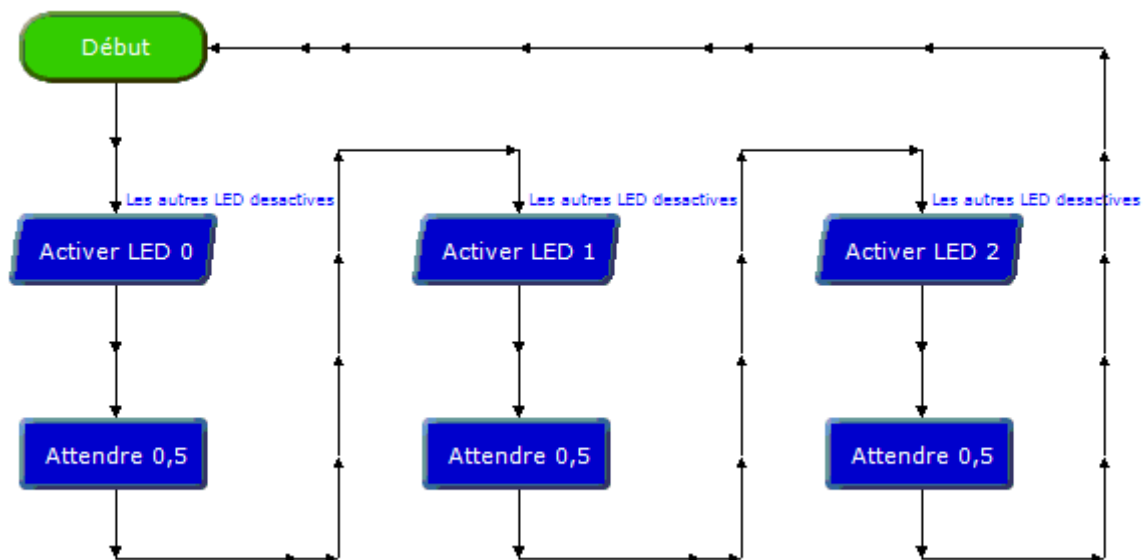
**Description du programme :** chenillard tournant dans le sens antihoraire.

**Synoptique :**



**Consignes :**

- A** – Entourer sur le diagramme ci-dessous la (les) instruction(s) qui provoquent l'allumage des LED.
- B** - Surligner sur le diagramme ci-dessous la (les) liaisons(s) qui déterminent le sens de rotation du chenillard.
- C** - Modifier le diagramme à l'écran afin d'inverser le sens de rotation du chenillard.  
Charger le programme dans le FlashProg et s'assurer que le sens de rotation est inversé.



## FICHE N°23 : INVERSER SENS ROTATION.p1f

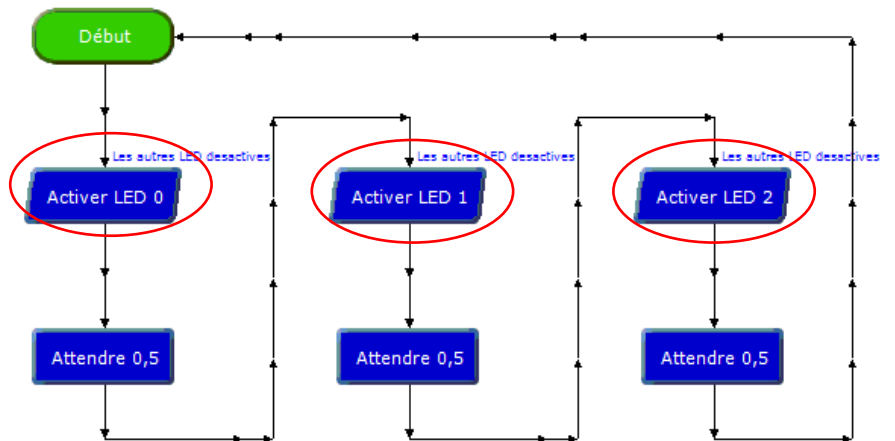
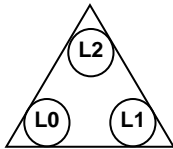
CORRIGÉ

Le programme initial est un chenillard tournant dans le sens antihoraire, c'est-à-dire dans le sens inverse des aiguilles d'une montre.

**A** – Ce sont les instructions « Activer 0 », « Activer 1 », « Activer 2 » qui provoquent l'allumage respectif des LED L0, L1 et L2. L'instruction d'activation d'une LED désactive les deux autres

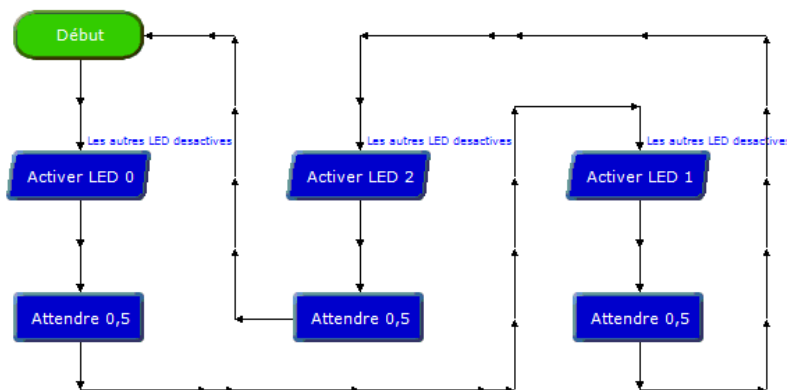
**B** – Les liaisons sont faites de telle sorte que la LED L0 s'allume, puis L1 et enfin L2. Cette séquence reboucle sur elle-même et on observe une rotation dans le sens antihoraire.

Rappel de la position des LED  
En face avant du FlashProg :

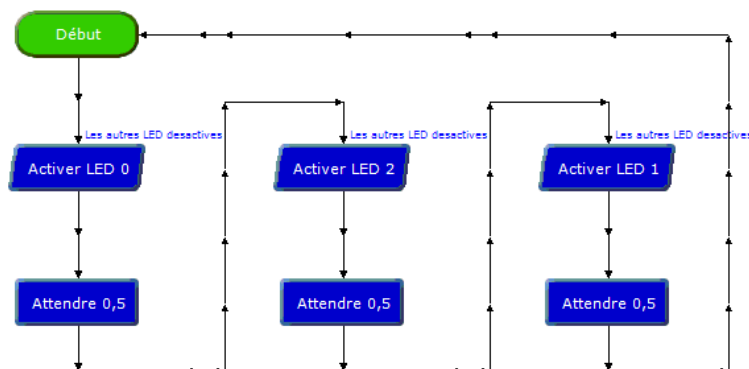


**C** – Pour inverser le sens de rotation (dans le sens horaire) on peut :

- modifier l'enchaînement des liaisons de telle sorte que les LED s'allument selon la séquence L0, L2, L1 (sens horaire).



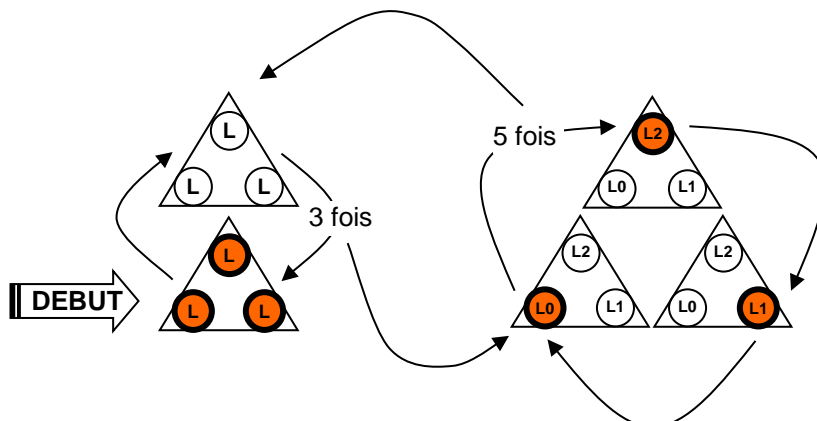
- garder les liaisons initiales et intervenir sur les instructions qui pilotent l'allumage et l'extinction des LED.



## FICHE N°24 : programme F24-MODIFIER REPETITION.plf

**Description du programme :** répétition trois fois d'une séquence clignotement puis répétition de cinq fois d'une séquence chenillard.

**Synoptique :**



**Consigne :**

- A** - Entourer en trait plein sur le diagramme ci-dessous la séquence clignotement et en pointillé la séquence chenillard.
- B** - Entourer en trait plein sur le diagramme ci-dessous la (les) instruction(s) qui détermine(nt) le nombre de répétitions de la séquence clignotement et en pointillés dessous la (les) instruction(s) qui détermine(nt) le nombre de répétitions de la séquence chenillard.
- C** - Modifier le diagramme à l'écran de telle sorte que la séquence clignotement soit répétée quatre fois et que la séquence chenillard soit répétée six fois.  
Charger le programme dans le FlashProg et s'assurer que le sens de rotation est inversé.

## FICHE N°24 : MODIFIER REPETITION.plf

CORRIGÉ

**A / B** – La première séquence exécutée fait clignoter les LED trois fois de suite.

Dès que la variable **A** qui est incrémentée de un en un atteint la valeur 3, la deuxième séquence chenillard est alors exécutée.

Selon le même principe, celle-ci est exécutée cinq fois de suite.

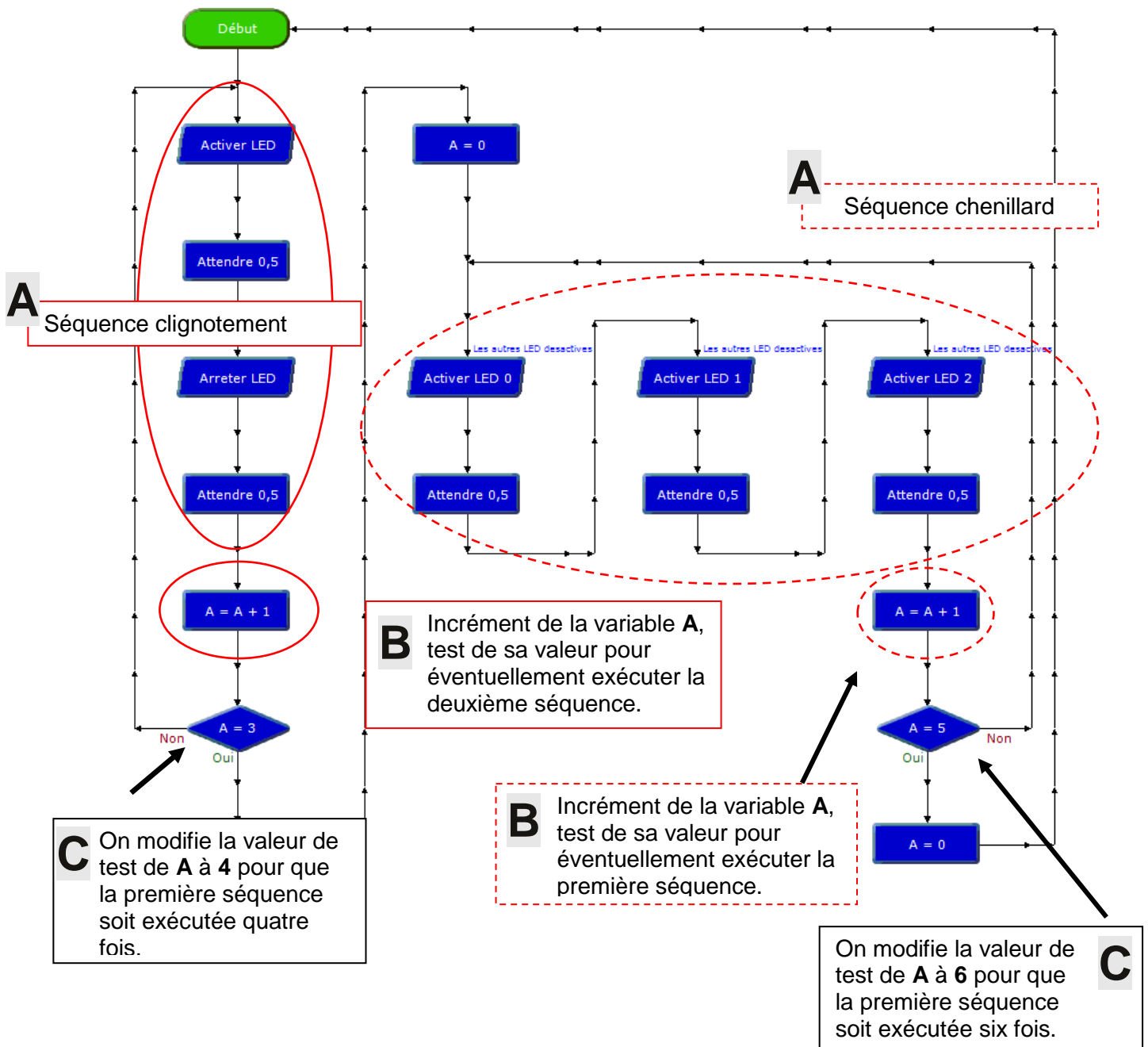
Le programme reboucle alors sur la première séquence.

On note que la variable **A** est réinitialisée à zéro dès que le test de sa valeur est valide.

Le programme reboucle alors indéfiniment sur lui-même.

**C** – On modifie la valeur de test de la variable **A** de 3 à 4 pour que la séquence clignotement soit exécutée 4 fois.

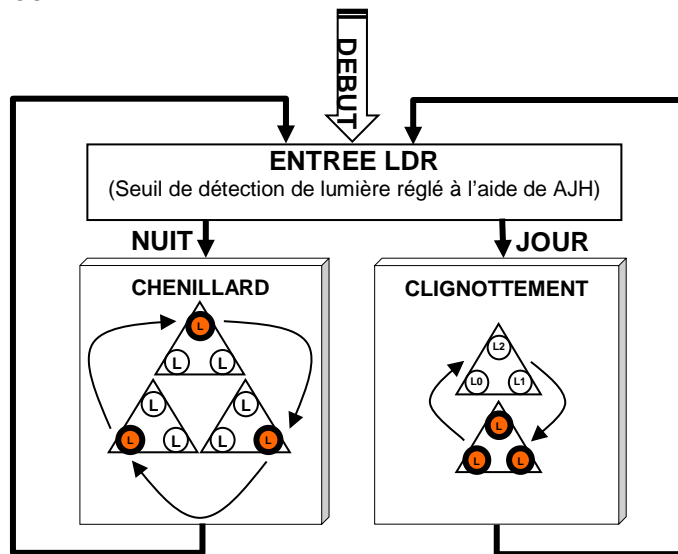
On modifie la valeur de test de la variable **A** de 5 à 6 pour que la séquence clignotement soit exécutée 6 fois.



## FICHE N°25 : programme F25-INVERSER MODE JOUR-NUIT.plf

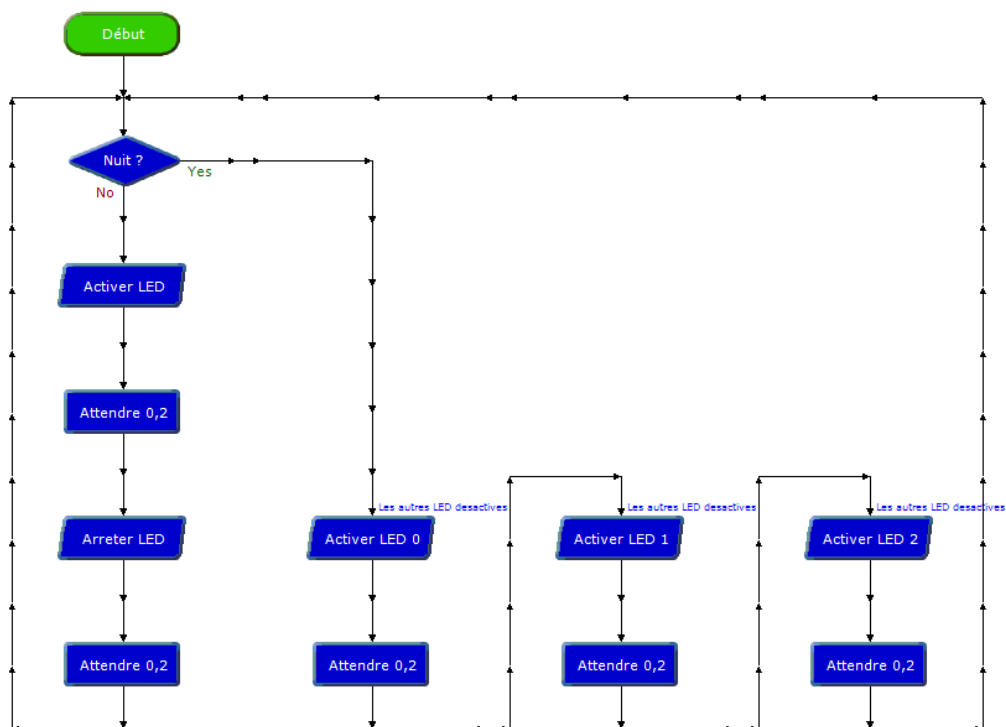
**Description du programme :** si il fait jour, une séquence clignotement est activée. Si il fait nuit une séquence chenillard est activée.

**Synoptique :**



**Consignes :**

- A** - Ouvrir le diagramme INVERSER MODE JOUR-NUIT.plf et le charger dans le FlashProg. Régler l'ajustable de telle sorte que si l'on obture la LDR (nuit), on observe une séquence chenillard. Vérifier que dès que la LDR n'est plus obturé (jour), on observe un clignotement simultané des trois LED.
- B** - Entourer en trait plein sur le diagramme ci-dessous la séquence clignotement et en pointillé la séquence chenillard.
- C** - Encadrer d'un rectangle l'instruction qui permet de déterminer si il fait jour ou si il fait nuit.
- D** - Modifier le diagramme l'instruction qui détecte le jour ou la nuit afin d'observer une séquence chenillard lorsqu'il fait jour et une séquence clignotement lorsqu'il fait nuit. Charger le programme dans le FlashProg et s'assurer que lorsque l'on obture la LDR on observe un clignotement simultané des trois LED.



## FICHE N°25 : INVERSER MODE JOUR-NUIT.plf

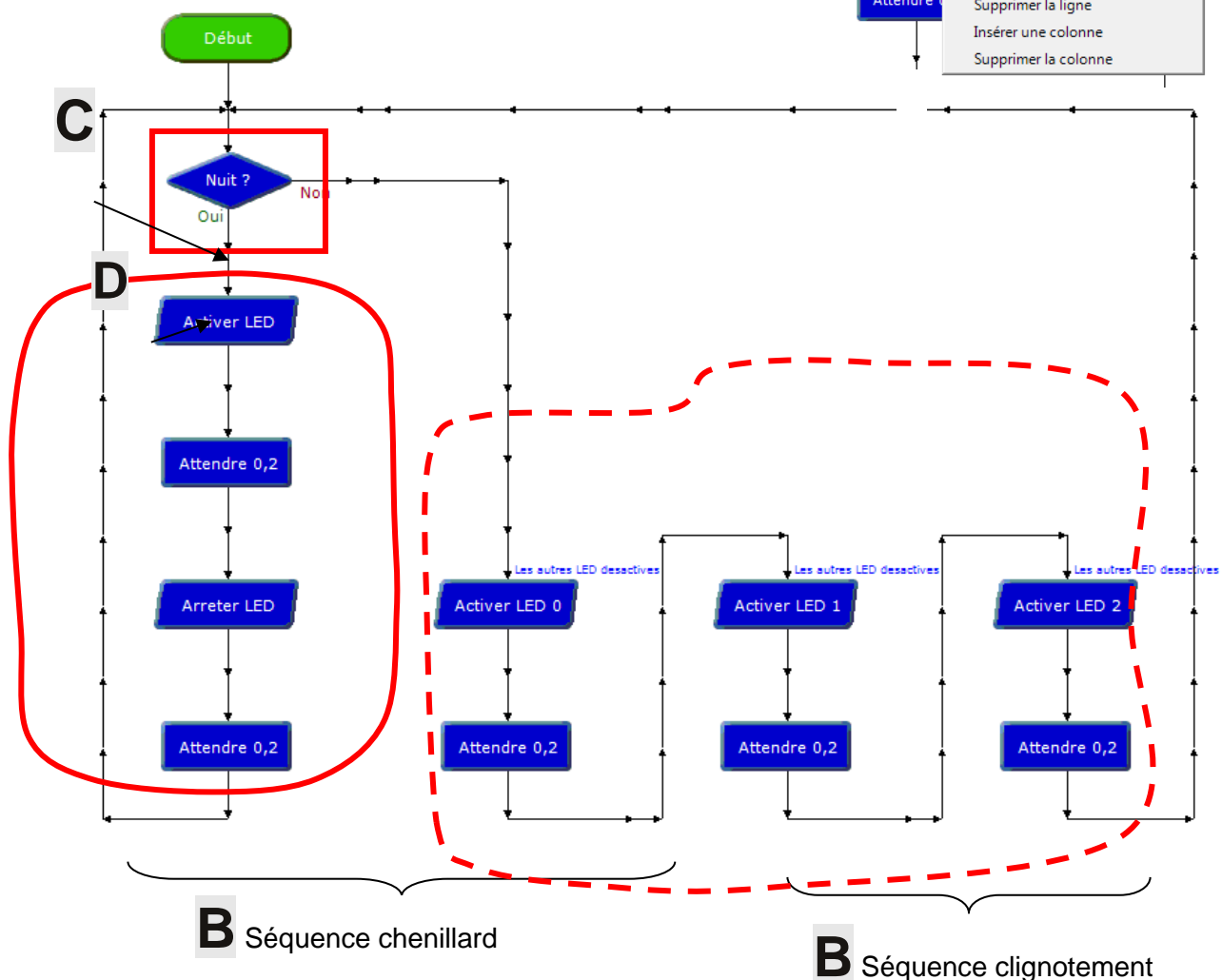
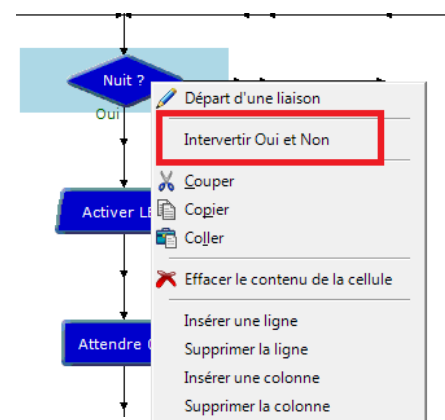
CORRIGÉ

- A** – C'est le niveau de tension sur l'entrée « 3 » qui détermine si il fait jour ou si il fait nuit.
- Si il fait nuit, la résistance de la LDR est minimum et le niveau de tension sur l'entrée « 3 » est minimum : Nuit = 1 = Oui.
  - Si il fait jour, la résistance de la LDR est maximum et le niveau de tension sur l'entrée « 3 » est maximum : Nuit = 0 = Non.

Pour vérifier le fonctionnement du programme, on positionne le curseur du résistor ajustable sur sa position médiane ; en considérant que le niveau de lumière ambiante de la classe est suffisamment élevé (lumière du jour), les LED L0, L1 et L3 doivent clignoter simultanément. Si l'on masque suffisamment le boîtier du FlashProg en plaçant sa main contre le boîtier au-dessus de la LDR, les LED se mettent à clignoter alternativement (effet chenillard).

- B / C** – Si Nuit=1 est vrai («Oui ») il fait nuit ; c'est la séquence chenillard qui est exécutée ; sinon, c'est la séquence clignotement qui est exécutée.

- D** – Pour inverser le mode de fonctionnement du système (nuit = clignotement, jour = chenillard), il suffit d'inverser les conditions :



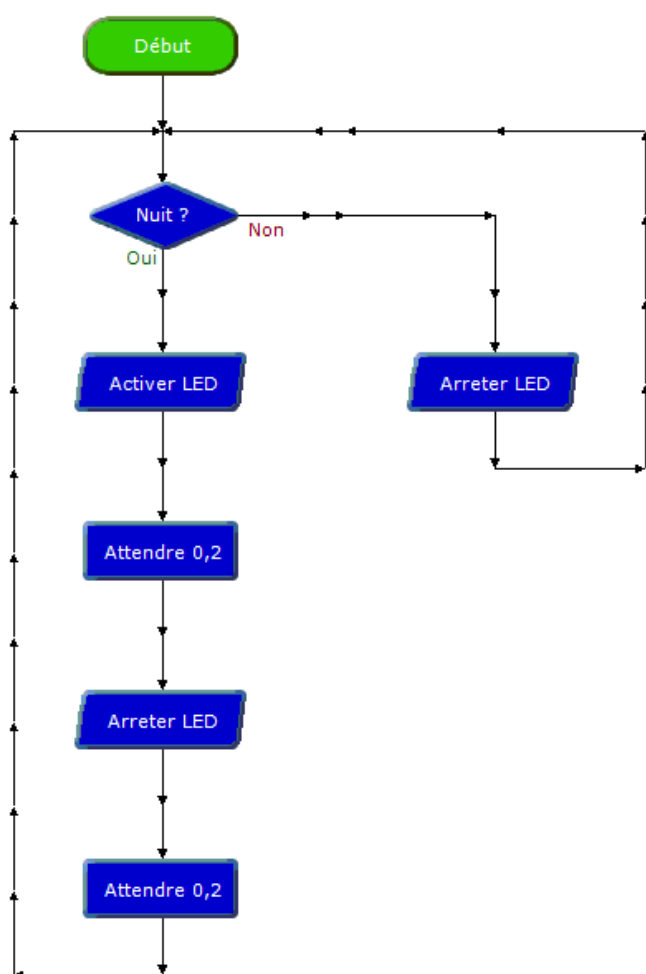
## FICHE N°26 : programme F26-DEMAR AUTO NUIT.plf

**Description du programme :** Ce programme permet de démarrer automatiquement le FlashProg lorsqu'il fait sombre. Le but de ce programme est de disposer d'un élément de sécurité qui s'active automatiquement à la tombée de la nuit et qui économise l'énergie lorsqu'il fait jour.

On observe sur le diagramme ci-dessous que l'instruction de test « pin3=0 » permet de déterminer l'état de l'entrée LDR :

- s'il fait jour (Décision est faux), le programme force l'extinction des trois LED (Arrêter LED) ; la consommation d'énergie est alors minimale ;
- s'il fait nuit (Décision est vrai), le programme exécute alors une séquence chenillard et permet d'activer automatiquement un signal lumineux de sécurité.

**Diagramme :**



**Consigne :**

**A** - Régler l'ajustable du FlashProg en position médiane.  
Ouvrir le diagramme **DEMAR AUTO NUIT.plf** et le charger dans le FlashProg.  
S'assurer que lorsqu'on obture la LDR (nuit), on observe une séquence chenillard.  
S'assurer que dès que la LDR n'est plus obturé (jour), on constate l'extinction complète des trois LED.

**B** – Dans la nuit, que risque-t-il de se passer si un phare de voiture éclaire le FlashProg ?

**C** - Modifier le diagramme de telle sorte que la fonction économie d'énergie soit préservée le jour et que la fonction chenillard soit activée en toutes circonstances la nuit.

**D** - Charger le programme dans le FlashProg.  
S'assurer que si le FlashProg est démarré le jour toutes les LED sont éteintes.  
S'assurer que si la LDR est obturée (simulation de la nuit), la séquence chenillard démarre automatiquement.  
S'assurer que si la LDR n'est plus obturée (simulation d'un phare de voiture qui éclaire le FlashProg), la séquence chenillard persiste.



## FICHE N°26 : DEMAR AUTO NUIT.plf

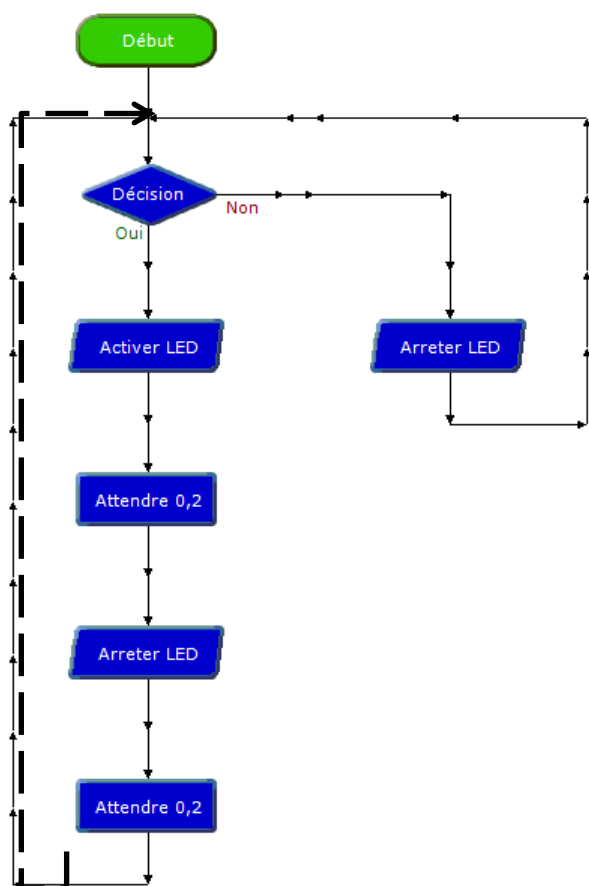
CORRIGÉ

**A** – Après chargement du programme initial DEMAR AUTO NUIT, le FlashProg étant dans la lumière ambiante de la classe (le jour), si l'ajustable est réglé en position médiane, toutes les LED restent à priori éteintes. Si ce n'est pas le cas (on observe une séquence chenillard), il convient alors de modifier le seuil de détection de la lumière en modifiant le réglage de l'ajustable jusqu'à ce que toutes les LED s'éteignent.

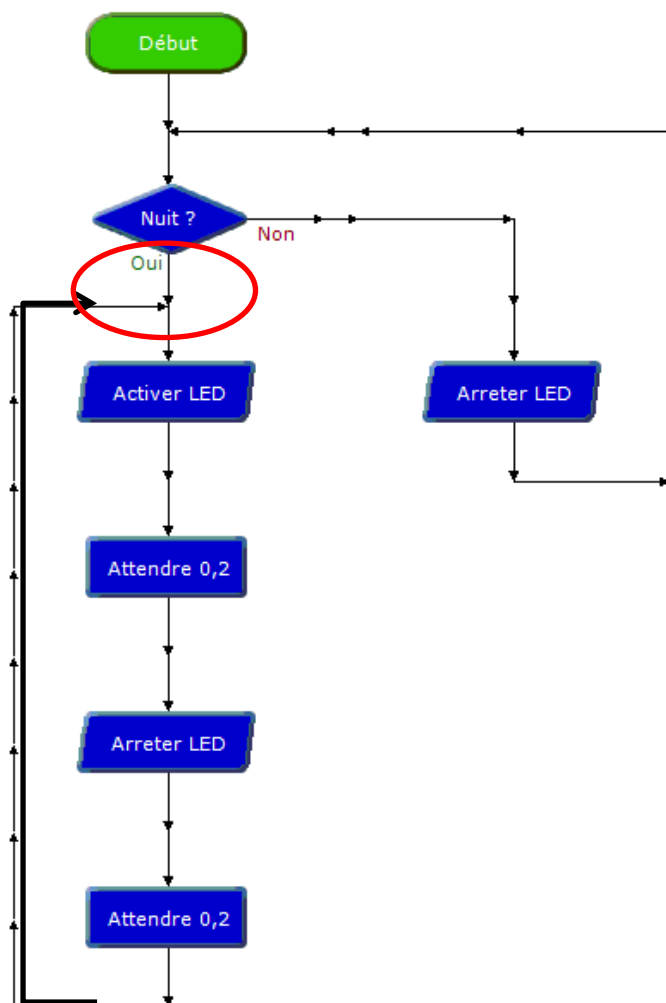
**B** - On constate que le programme revient systématiquement à l'instruction de test de l'entrée LDR, si bien que si il fait nuit et qu'un phare de voiture éclaire le FlashProg toutes les LED risquent de s'éteindre. Ce cas particulier n'est pas pris en compte par le programme et constitue une faille qui peut s'avérer dangereuse pour l'utilisateur.

**C** - La façon la plus simple de remédier à cette faille est sans doute de reboucler indéfiniment sur la séquence chenillard dès qu'il fait nuit. Pour cela on modifie la liaison qui renvoie systématiquement au test de la LDR et on la réoriente vers le début de la séquence chenillard.

## PROGRAMME INITIAL



## PROGRAMME AMÉLIORÉ



## AUTOMATISME NIVEAU 3

### Créer un programme

L'élève crée un programme selon les consignes énoncées.

Des fichiers de correction sont proposés dans le dossier CORRECTIONS Automatismes niveau 3 du CD Rom CD-FP.

Les fichiers de corrections proposés ne sont pas exhaustifs ; en effet le langage de programmation permet d'obtenir le même résultat de différentes manières. Par ailleurs le choix des temps d'attente (« attendre xxx ») qui ponctuent les programmes est arbitraire.

Les caractéristiques principales des outils de programmation sont décrites en Annexe D de ce document. Ces caractéristiques se limitent aux outils utilisés dans les chapitres précédents de ce document.

	<b>Effet visuel souhaité</b>
FICHE N°31	Faire clignoter trois fois de suite la LED 0, puis trois fois de suite la LED 1, puis trois de suite la LED 2
FICHE N°32	Créer une séquence qui allume fugitivement et successivement L2 puis L1 puis L0, qui marque un temps d'arrêt ou toutes les LED sont éteintes et qui repart dans l'autre sens.
FICHE N°33	Allumage de L0, extinction de LED0, allumage des trois LED, extinction des trois LED, allumage de LED1 . . .
FICHE N°34	Train de cinq flashes rapides (clignotement simultané des LED), ponctué par un temps d'arrêt ou toute les LED sont éteintes.
FICHE N°35	Lorsqu'il fait jour, clignotement lent de L2 (témoin de fonctionnement), sinon clignotement rapide et simultané des trois LED.

#### Travail élève :

1. Lancer le logiciel PICAXE Logicator et lancer la création d'un nouveau diagramme (Menu **Fichier \ Nouveau \ Nouveau diagramme**).
2. A partir de sa fiche, créer le diagramme selon la consigne énoncée.
3. Convertir le diagramme en Basic (F5). Transférer le programme dans le FlashProg (F5) et vérifier que son exécution correspond à la consigne énoncée.

**Rappels utiles :** Les instructions d'un programme sont exécutées les unes après les autres (il n'exécute pas plusieurs instructions à la fois).

Une instruction est constituée par un mot clé du langage de la programmation (ici il s'agit des « blocs » utilisés dans les diagrammes). Une instruction a un rôle bien déterminé et peut être paramétrée. Par exemple « attendre xxx » indique au microcontrôleur d'attendre pendant un temps de xxx secondes où xxx est une valeur paramétrable.

Tant que le microcontrôleur exécute cette instruction, il ne fait rien d'autre. Ainsi, si l'on veut réagir immédiatement à un stimulus extérieur (test de l'entrée LDR ou bouton-poussoir), il faut s'arranger pour que le programme ne soit pas « coincé » trop longtemps sur l'exécution de l'instruction en cours, sans quoi on risque de rater l'évènement extérieur (changement de luminosité, appui sur le bouton-poussoir).

Le microcontrôleur du FlashProg enchaîne l'exécution des instructions au rythme de 1 million d'instructions par secondes (ordre de grandeur).

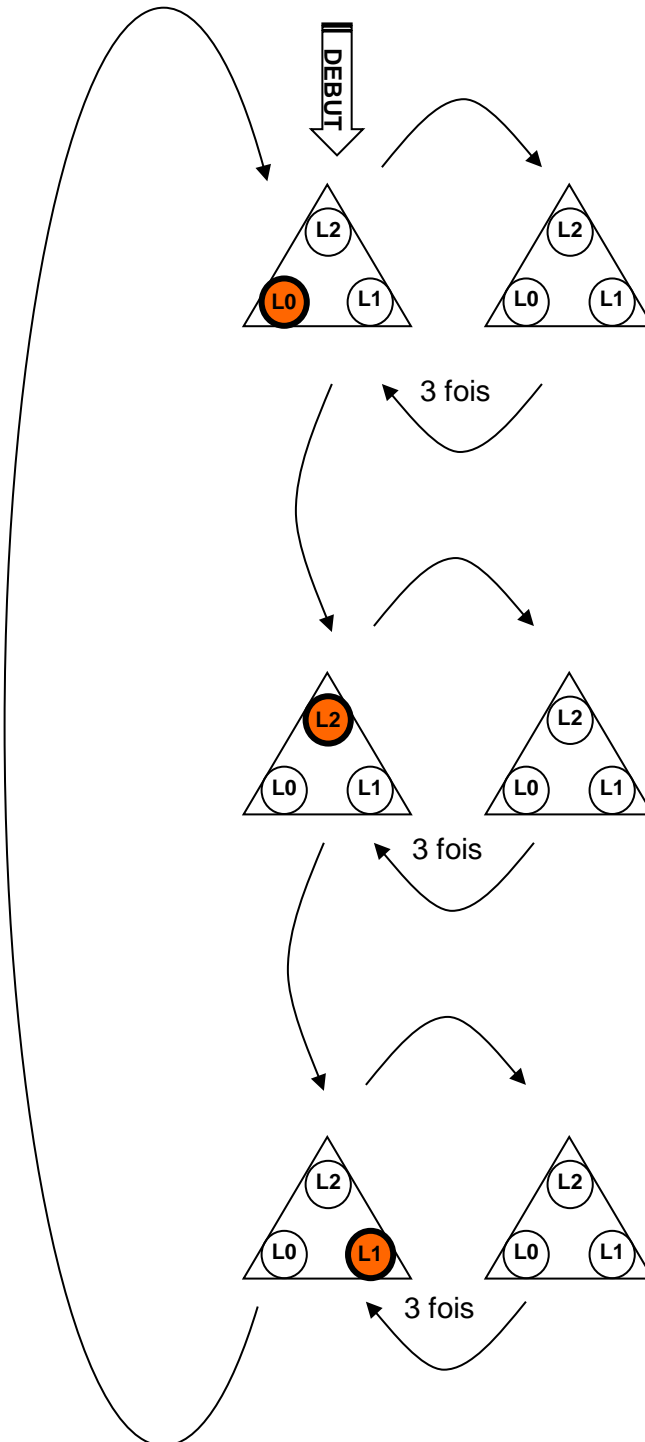
A la mise sous tension du FlashProg, le microcontrôleur est initialisé et toutes ses sorties sont remises à zéro ; le microcontrôleur commence alors à exécuter son programme à partir de l'instruction « Début ». Une instruction du type « Activer 0 » qui pilote l'état de la sortie 0 et provoque l'allumage de la LED L0 et maintient cette action tant que le programme ne rencontre pas une instruction qui annule pas cette action (en l'occurrence « Désactiver 0 »).

# FICHE N°31

## Consigne :

Créer un programme qui reboucle sur elle-même et fait clignoter trois fois de suite L0, puis trois fois de suite L2, puis trois fois de suite L1.

## Synoptique :

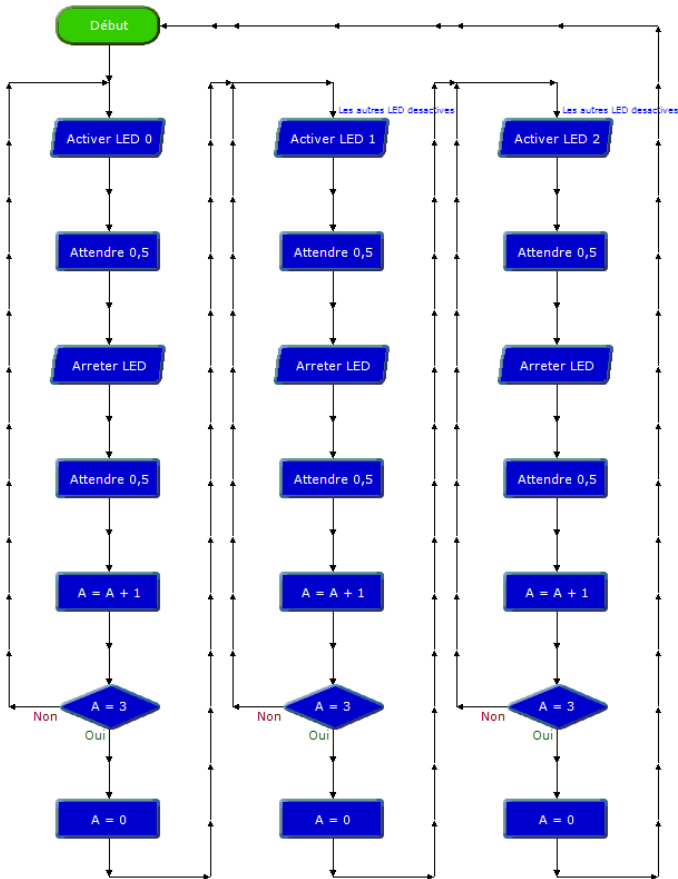


## FICHE N°31

Diagramme :

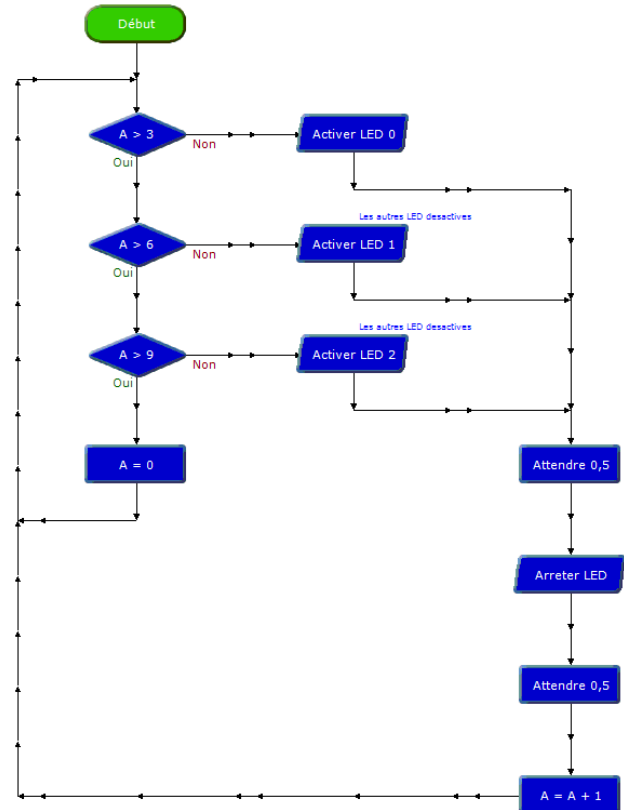
CORRIGÉ

F31-CORRECTIONa.plf



**Note** : à la mise sous tension, toutes les sorties sont à zéro.  
 Une sortie reste à zéro tant qu'une instruction « Activer x » n'active pas cette sortie.  
 Sachant cela, on évite d'écrire des instructions inutiles qui mettent à zéro des sorties qui n'ont jamais été activées.  
 Le programme occupe ici 58 octets.

F31-CORRECTIONb.plf



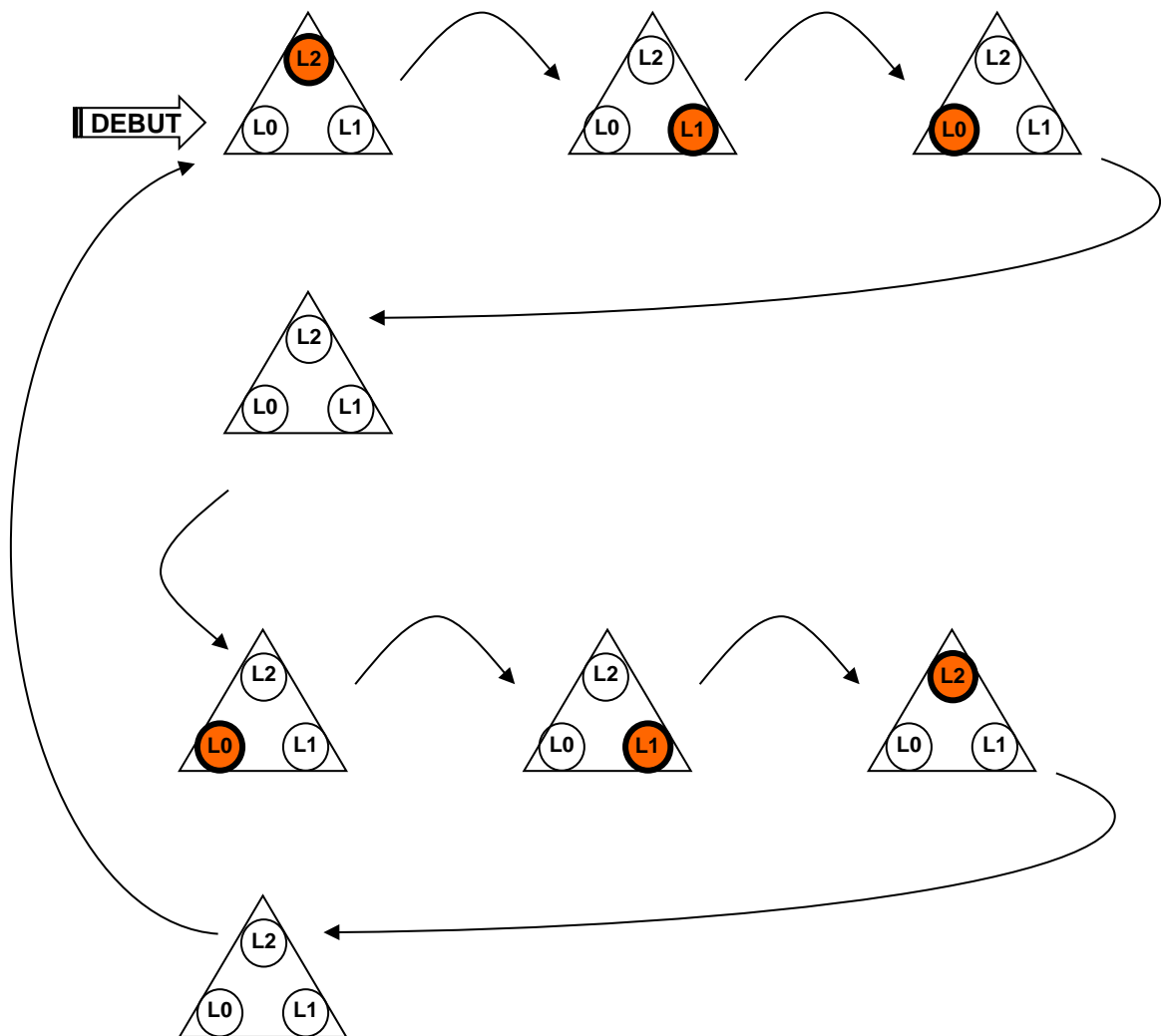
On peut simplifier le programme en utilisant une variable locale **A**.  
 Le programme occupe ici 32 octets.

## FICHE N°32

## Consigne :

Créer une séquence qui allume fugitivement et successivement L2 puis L1 puis L0, qui marque un temps d'arrêt ou toutes les LED sont éteintes et qui repart dans l'autre sens.

## Synoptique :

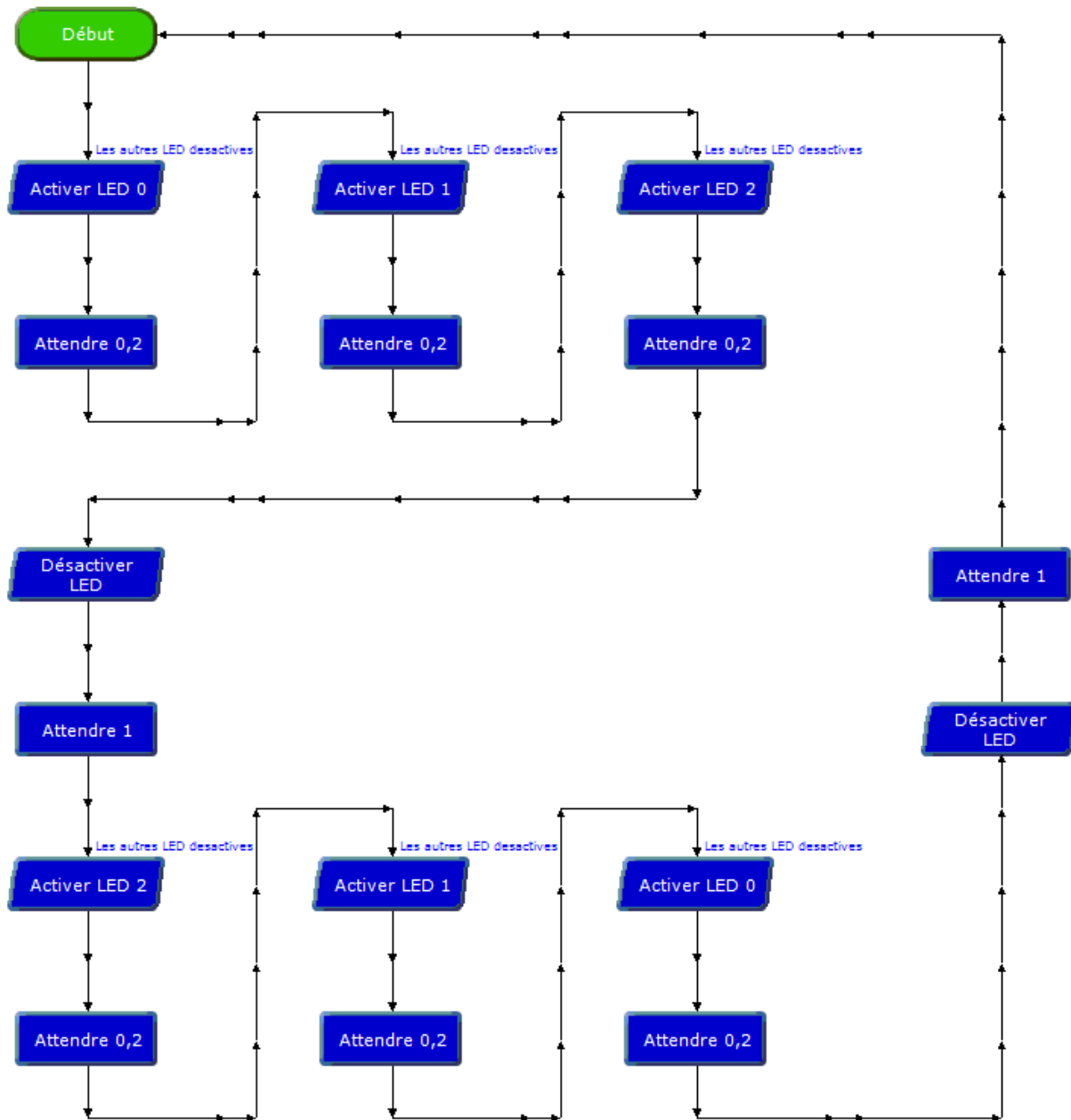


## FICHE N°32

CORRIGÉ

Diagramme :

F32-CORRECTIONa.plf

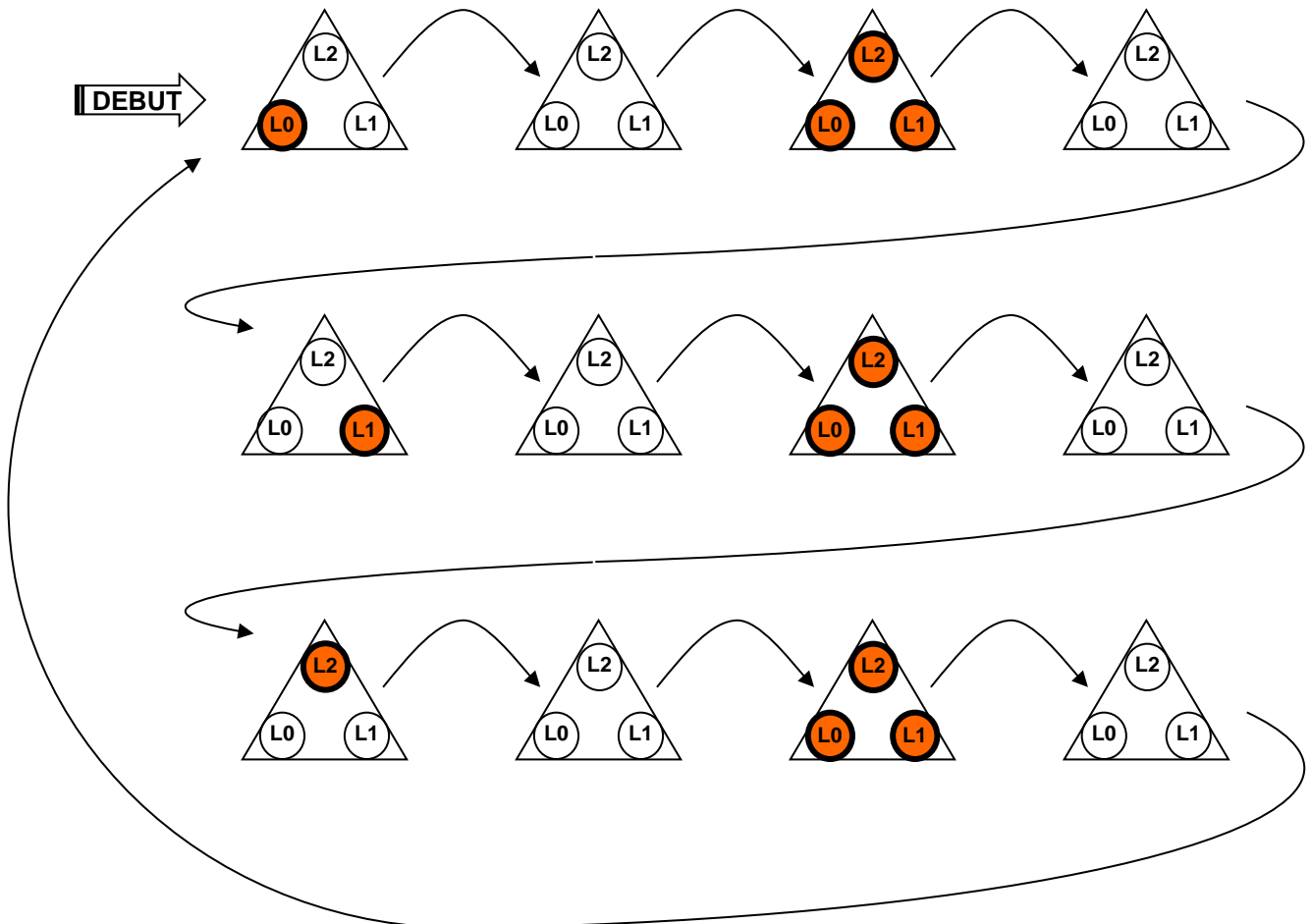


## FICHE N°33

## Consigne :

Allumage de L0, extinction de LED0, allumage des trois LED, extinction des trois LED, allumage de LED1.

## Synoptique :



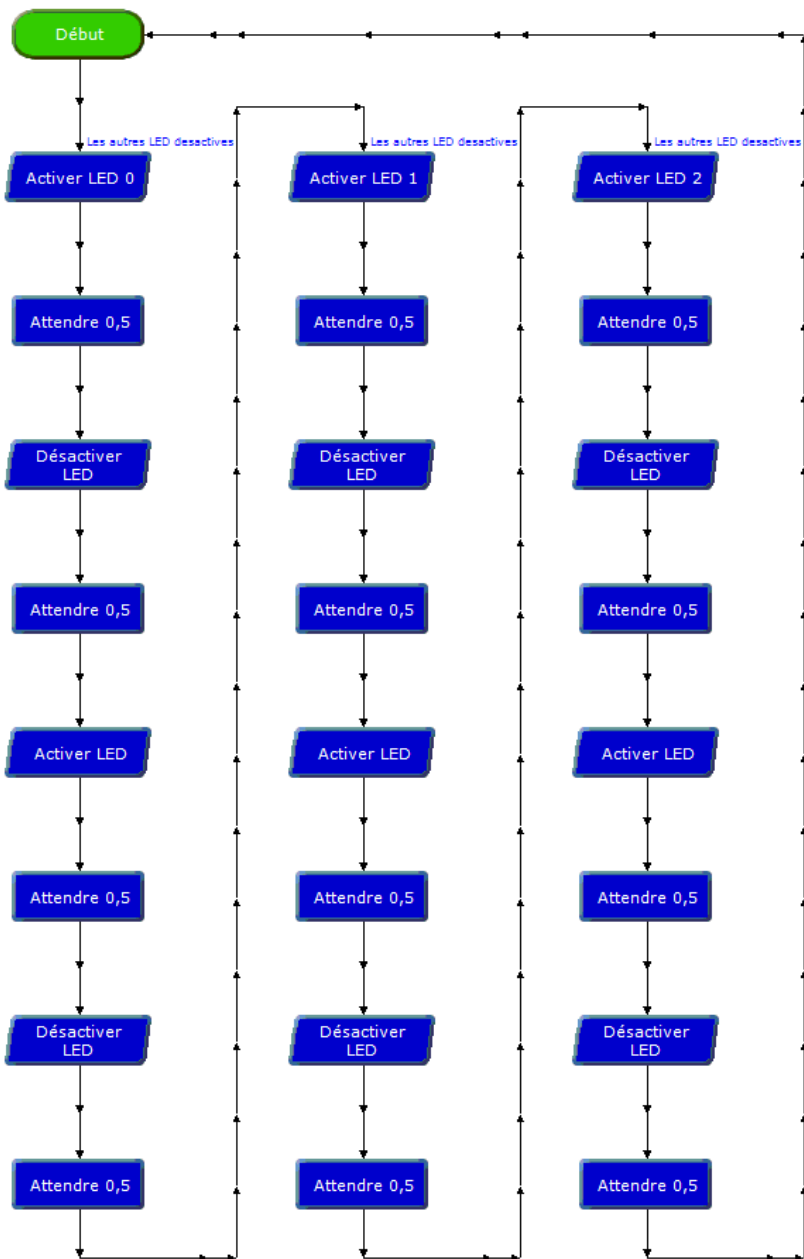


FICHE N°33

**CORRIGÉ**

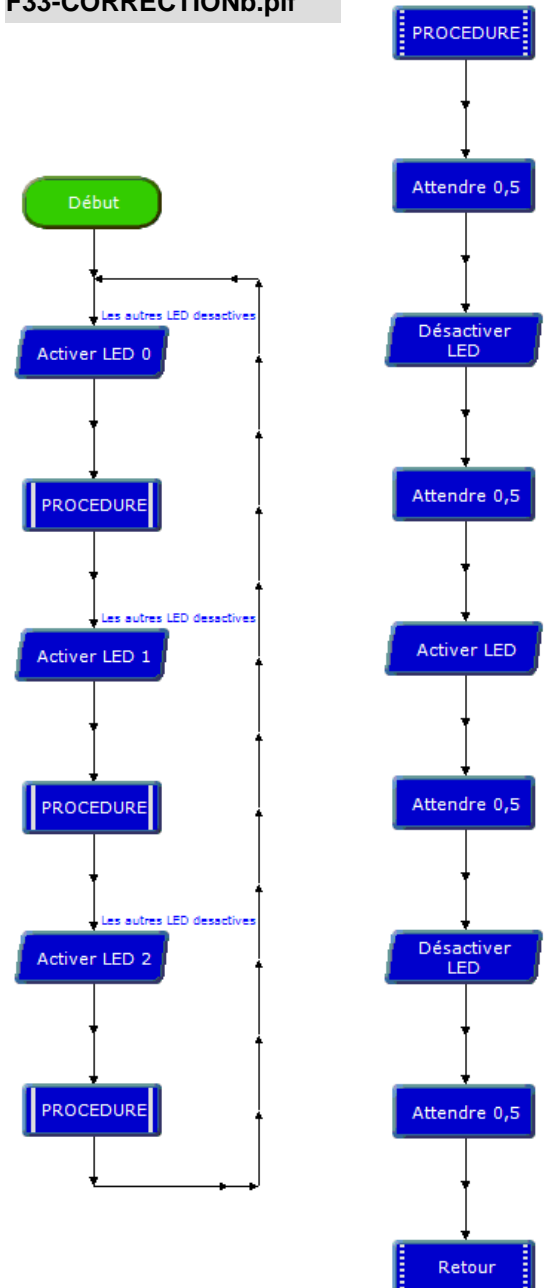
Diagramme :

F33-CORRECTIONa.plf



1<sup>ère</sup> possibilité de programmation :  
 Utilisation de 25 blocs sur le diagramme.  
 Consommation de 59 octets de mémoire de programme.

F33-CORRECTIONb.plf



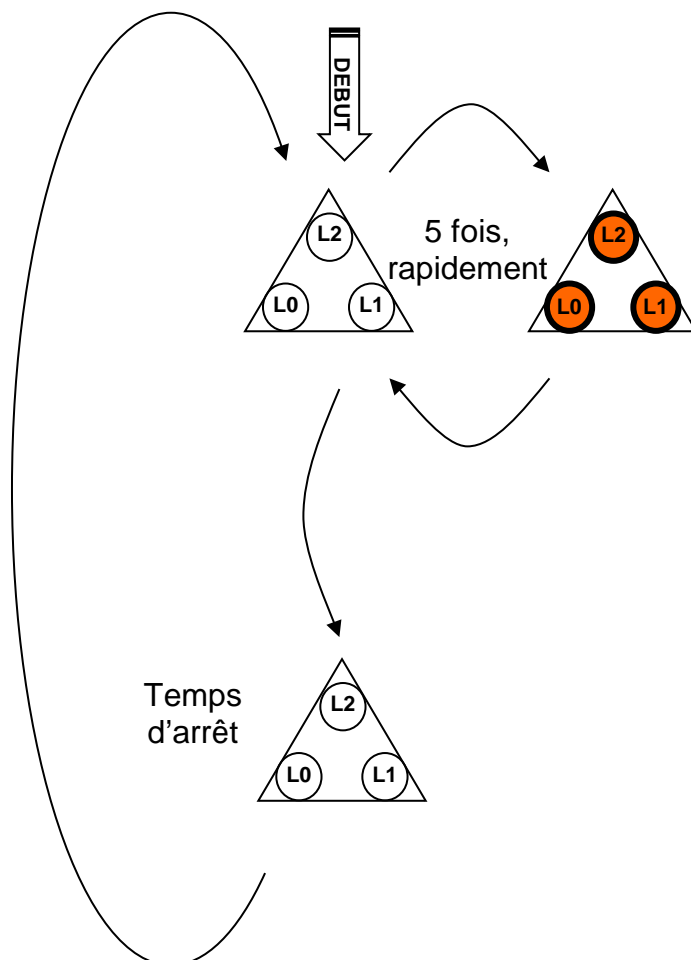
2<sup>ème</sup> possibilité de programmation avec  
 emploi de sous-programmes :  
 Utilisation de 16 blocs sur le diagramme.  
 Consommation de 31 octets de mémoire  
 de programme.

## FICHE N°34

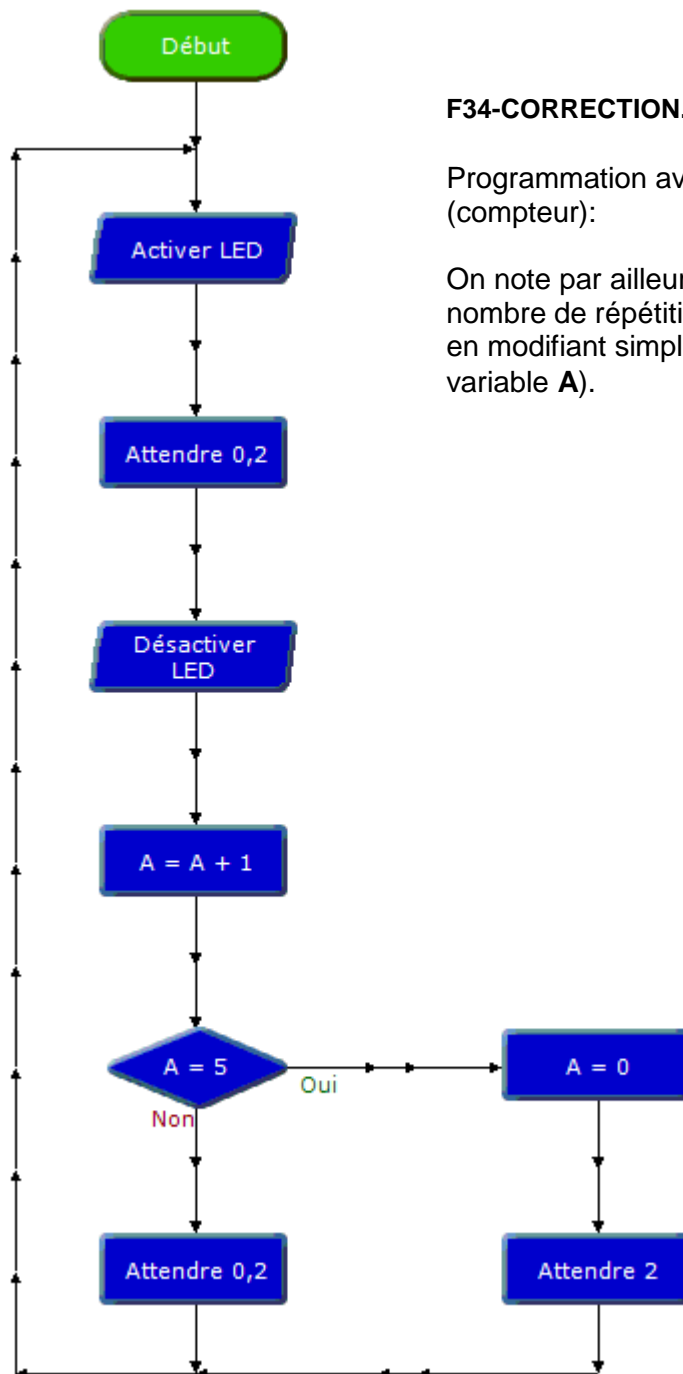
## Consigne :

Train de cinq flashes rapides (clignotement simultané des LED), ponctué par un temps d'arrêt ou toute les LED sont éteintes.

## Diagramme :



CORRIGÉ



## F34-CORRECTION.plf

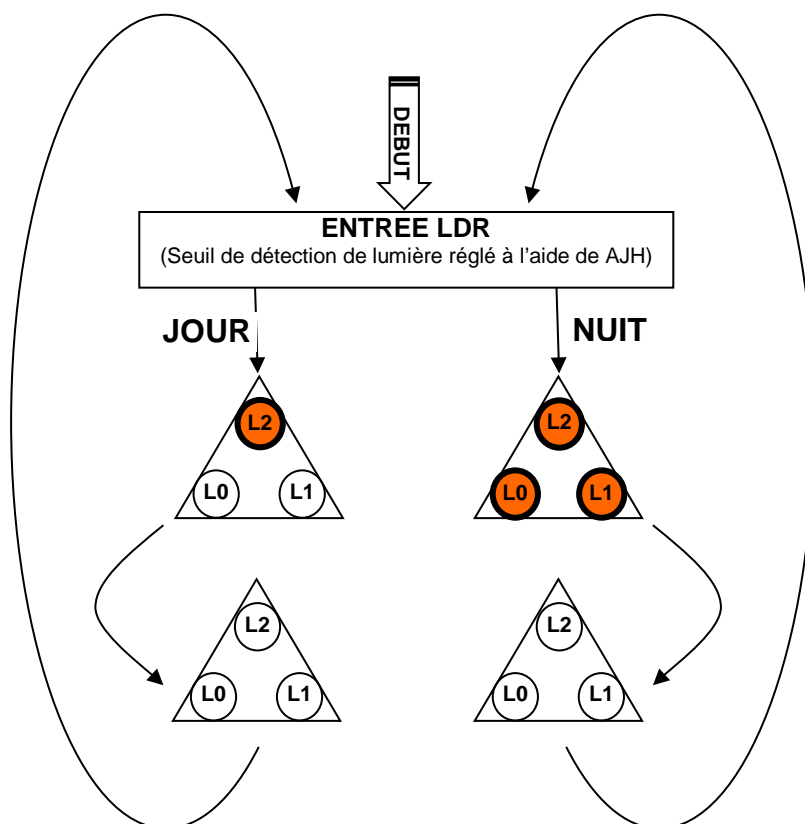
Programmation avec emploi d'une variable (compteur):

On note par ailleurs qu'il est très facile de modifier le nombre de répétitions de la séquence de clignotement en modifiant simplement un bloc (bloc de test de la variable **A**).

## FICHE N°35

## Consigne :

Lorsqu'il fait jour, clignotement lent de L2 (témoin de fonctionnement), sinon clignotement rapide et simultané des trois LED.



## FICHE N°35

CORRIGÉ

Diagramme :

F35-CORRECTIONa.plf

