

FlashProg

2004

Triangle de sécurité programmable

Automatisme



CONTENU DES CDROM

Le dossier est accompagné de deux CDROM :

- CDROM « Programming Editor » :
Il permet d'installer l'environnement de programmation du FlashProg.
La procédure d'installation est décrite en annexe du dossier.
- CDROM CD-FP :
Il contient ce dossier (document Microsoft Word 2002 ®),
les fichiers de programmation décrits dans ce dossier,
les fichiers de modélisation en 3D du projet.

SOMMAIRE DU DOSSIER

A Présentation du projet

| | |
|-------------------------|--------|
| Description du produit | 3 et 4 |
| Utilisation pédagogique | 5 |

B Fabrication

| | |
|--|----------|
| Plans d'ensemble | 6 et 7 |
| Plans du module électronique | 8 à 11 |
| Montage du module électronique dans le boîtier | 12 |
| Plan de perçage du circuit imprimé | 13 |
| Guide de fabrication | 14 et 15 |

C Programmation

| | |
|--|---------|
| Notions de base | 17 |
| Automatisme niveau 1 - Choisir un programme dans une liste et le charger | 18 à 34 |
| Automatisme niveau 2 - Modifier un programme | 35 à 47 |
| Automatisme niveau 3 - Créer un programme | 48 à 59 |

D ANNEXES

| | |
|--|----------|
| A - Installation du logiciel « Programming Editor » | 61 |
| B - 1 ^{er} lancement du logiciel | 62 |
| C - Transfert d'un programme dans le FlashProg | 63 |
| D - Description des outils de programmation | 64 et 65 |
| E - Programme de test. | 66 |
| F - Fiche de suggestions et d'inscription aux mises à jour gratuites de ce dossier | 67 |

Droit de reproduction

- Le dossier et son CDROM 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 la seule condition que soit cité le nom de l'éditeur : 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 du CDROM ne sont pas autorisées sans l'accord de la Sté A4.

- Le logiciel Programming Editor qui peut accompagner l'acquisition de ce dossier est protégé par les lois du copyright international. Tout ou partie de ce logiciel ne peut être reproduit, copié, vendu, revendu ou exploité dans un but commercial qui n'ait été expressément autorisé par la société Revolution Education.

A – PRESENTATION DU PROJET

Description du produit 3

Utilisation pédagogique 4

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 DEL haute luminosité visibles à plus de 100 mètres. Il dispose de :

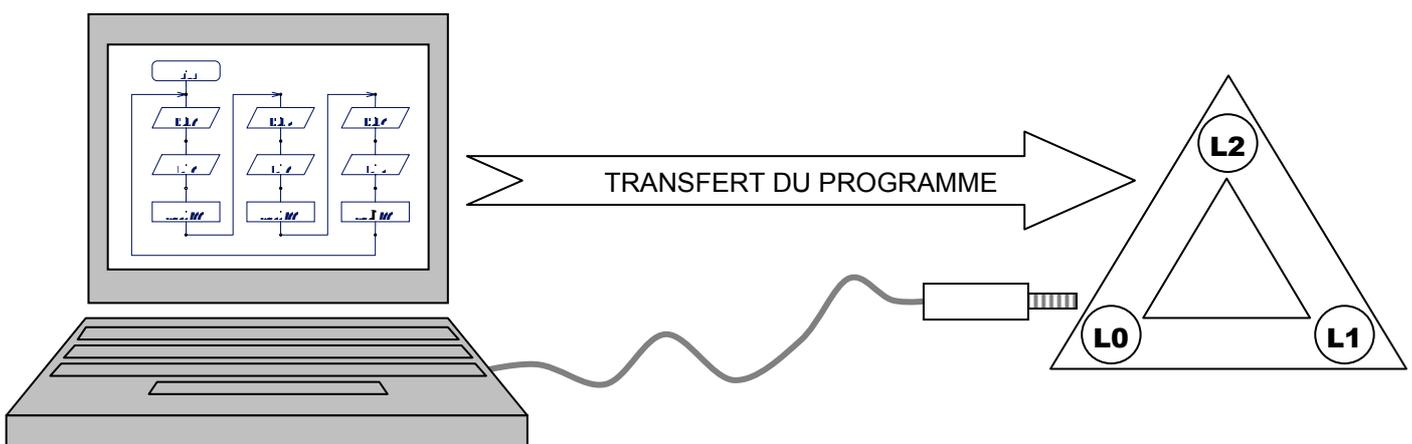
- Trois sorties effecteurs indépendantes constitués par trois DEL.
- Deux entrées constituées par un capteur photosensible (LDR) et un bouton poussoir

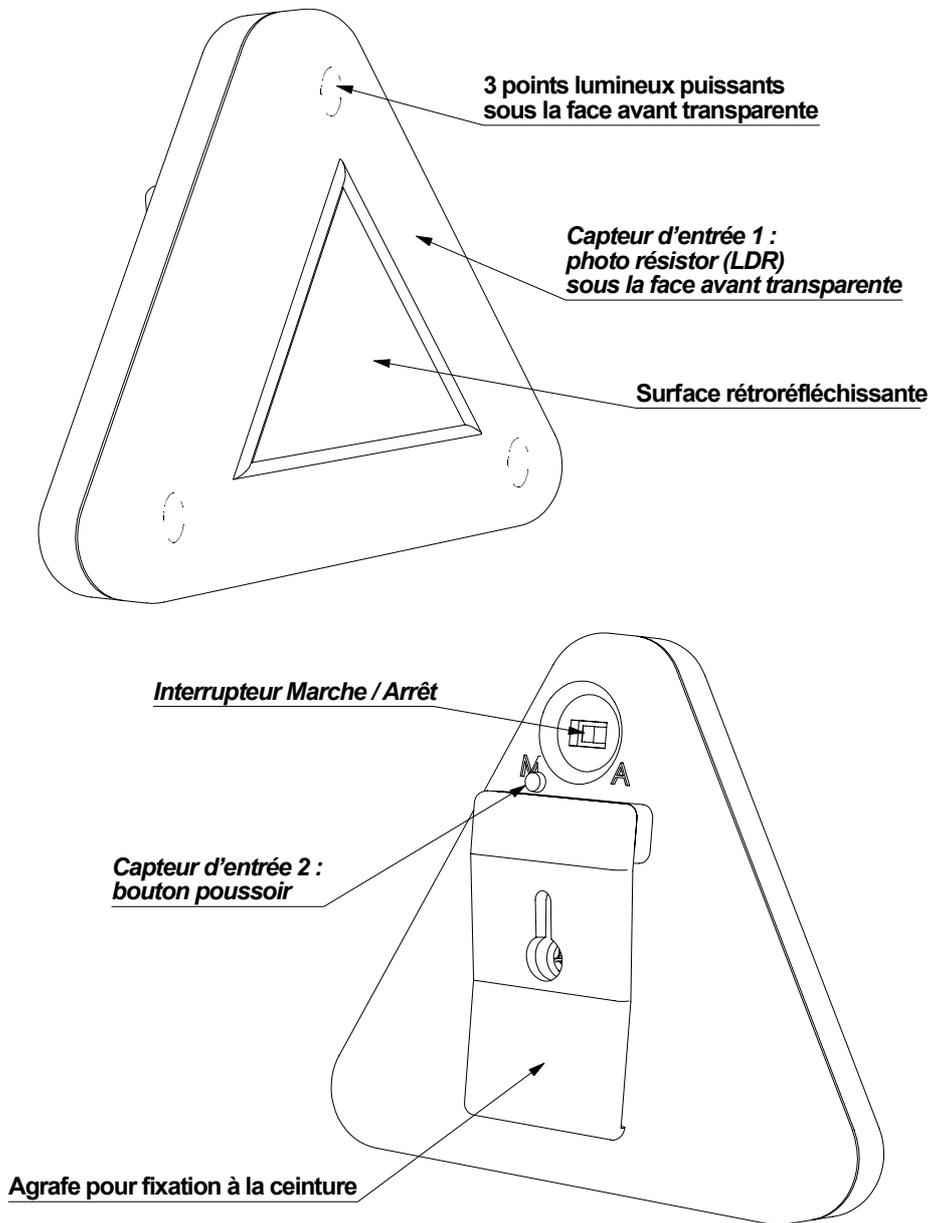
La programmation du module permet de créer une grande variété de séquences d'allumage des DELs. Le module accueille un bouton poussoir et une LDR qui permettent de sélectionner différentes séquences d'allumage des DELS en fonction d'un évènement extérieur (appui sur le bouton poussoir, seuil de luminosité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

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

- le clignotement simultané des DEL,
- le clignotement alterné des DEL (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 Programming Editor 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 Prom) et est reprogrammable à volonté.





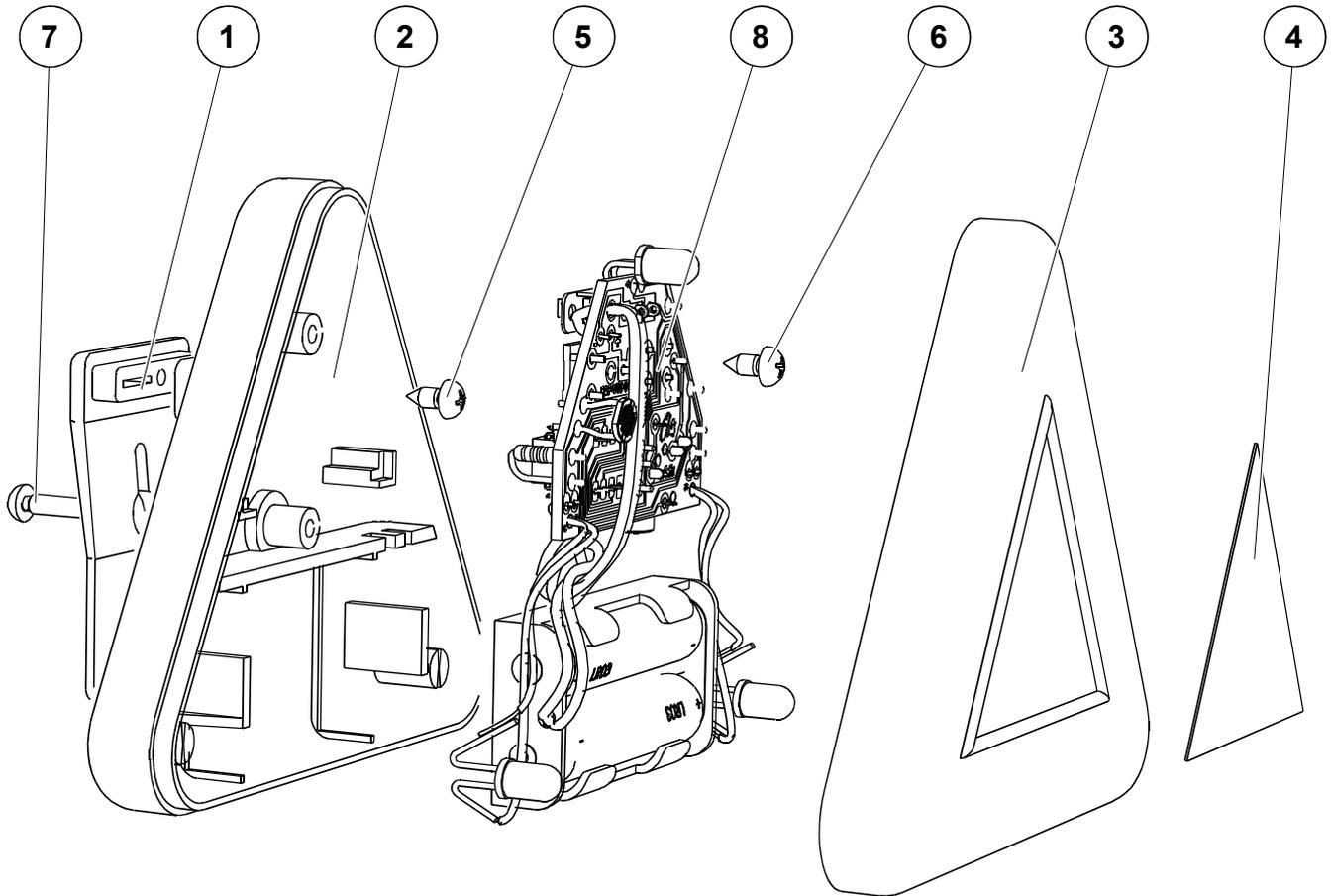
Utilisation pédagogique

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.

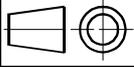
- Le chapitre « Fabrication » traite de la fabrication du produit et l'implantation des composants sur le circuit imprimé.
- Le chapitre « Programmation » traite de l'exploitation du FlashProg en automatisme. Cette partie ne se substitue pas à la documentation de l'environnement de programmation qui décrit (en anglais) l'intégralité des possibilités offertes par ce système. L'environnement de programmation « Programming Editor » 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.
- L'annexe présente des documents relatifs à la mise en œuvre du logiciel de programmation.

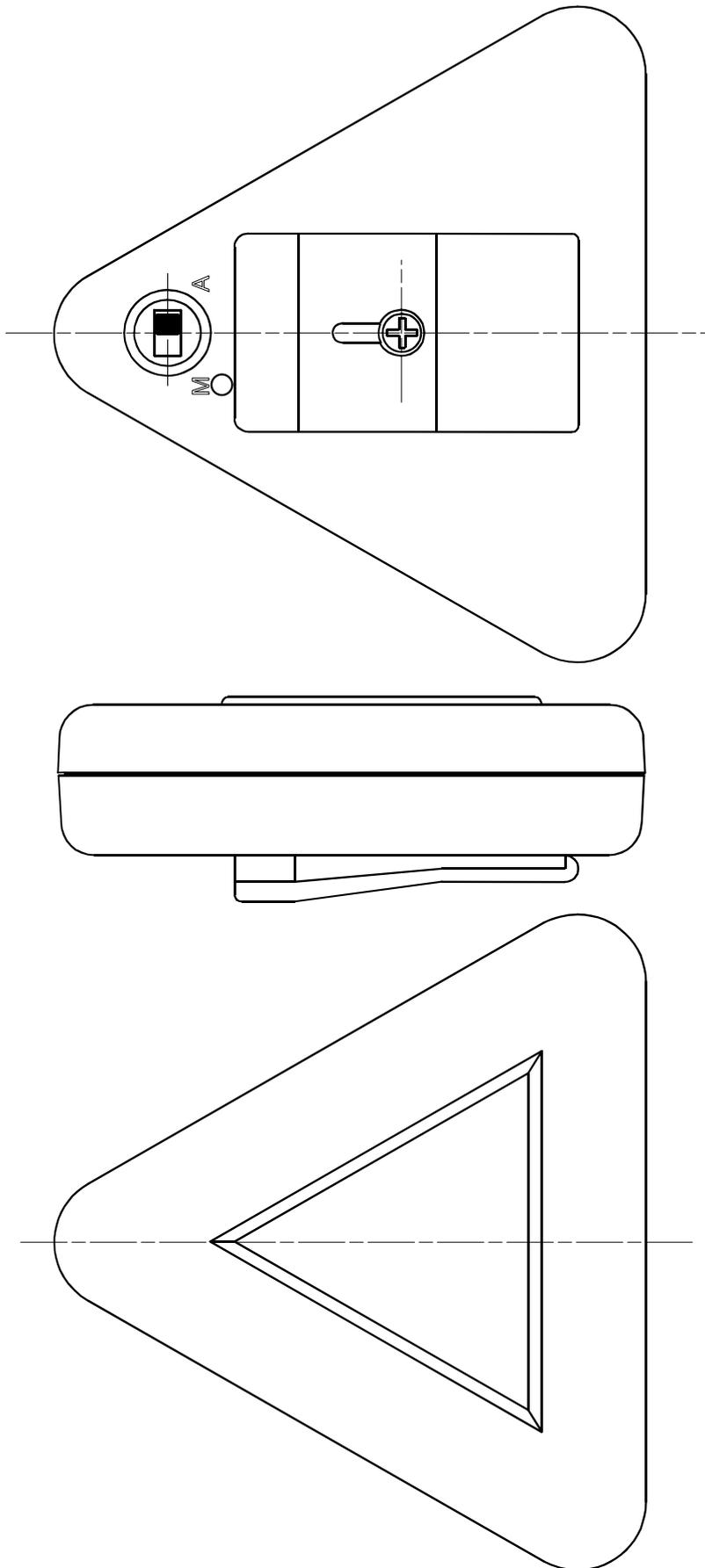
B – FABRICATION

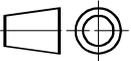
| | |
|--------------------------------|----------|
| Ensemble | 6 et 7 |
| Module électronique | 8 à 11 |
| Montage du module électronique | 12 |
| Circuit imprimé | 13 |
| Guide de fabrication | 14 et 15 |

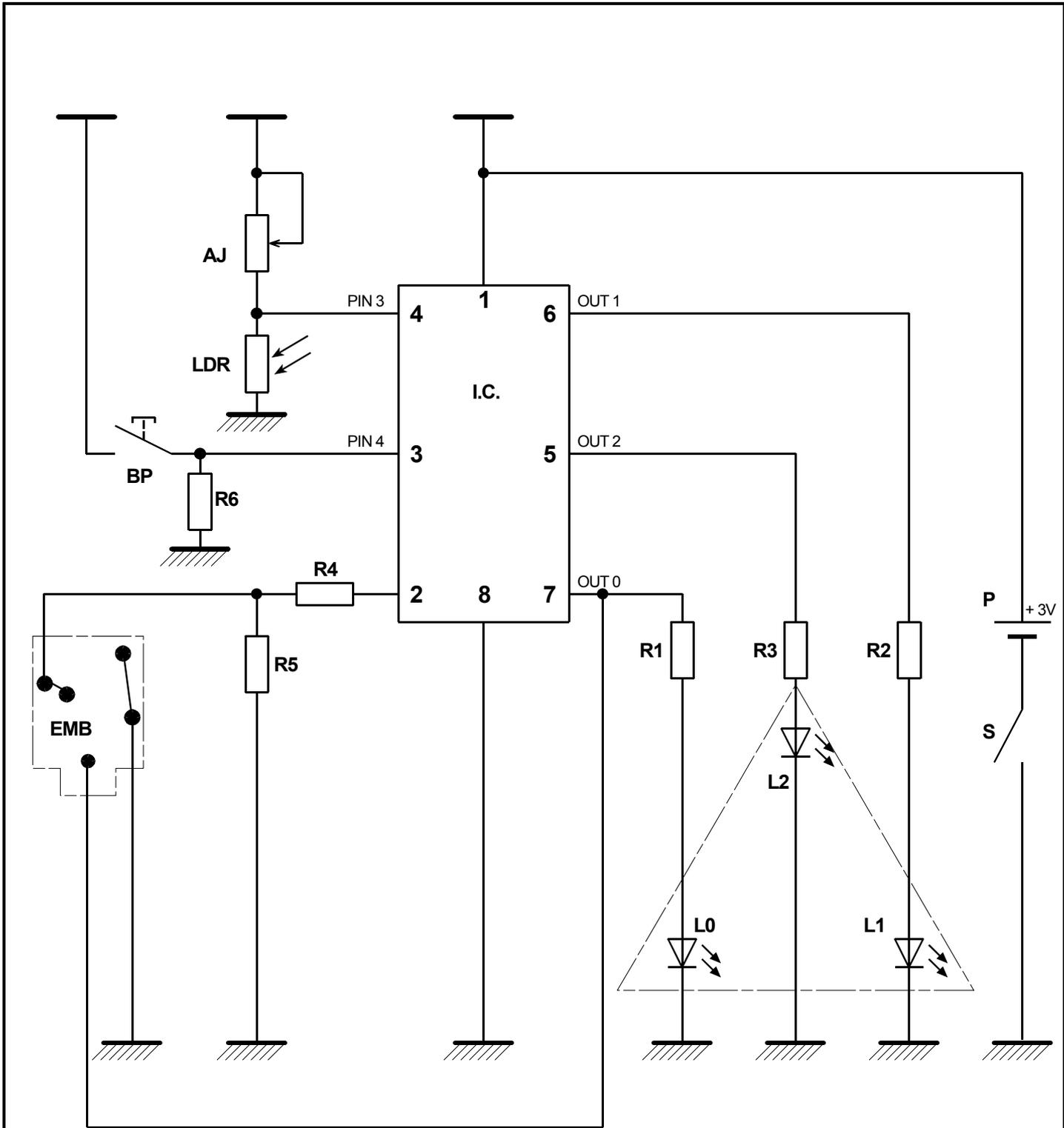


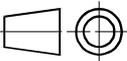
| | | | |
|---|----|--|--|
| 8 | 01 | Module électronique complet | Voir Nomenclature du module électronique |
| 7 | 01 | Vis de fixation de la demi-coque avant | Type tôle - Tête cylindrique - Ø 3.2 x longueur 16 |
| 6 | 01 | Vis de fixation du module | Type tôle - Tête cylindrique - Ø 3.2 x longueur 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 |

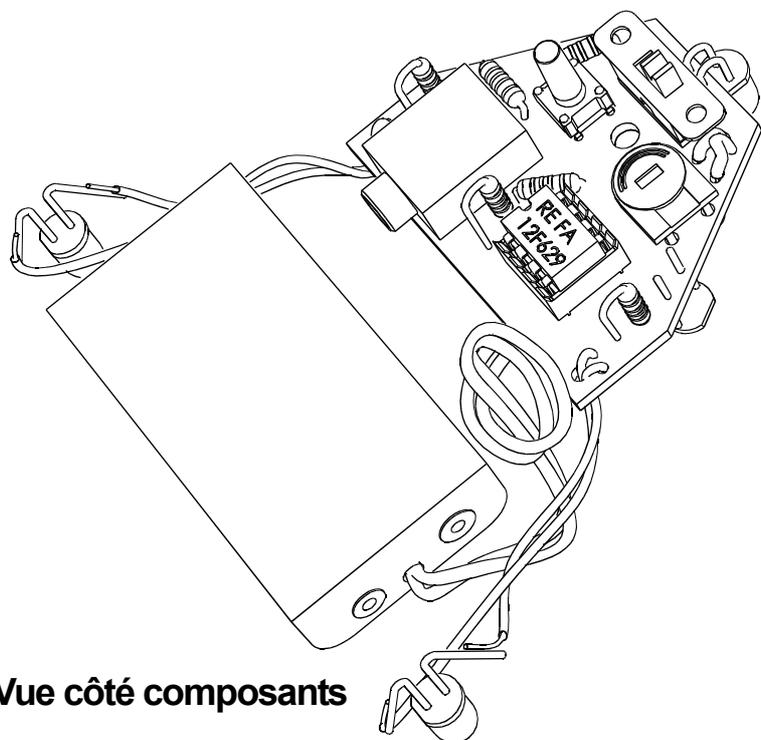
| REPERE | NOMBRE | DESIGNATION | CARACTERISTIQUES |
|--|---|-------------|--|
|  TECHNOLOGIE AU COLLEGE |  A4 | | PROJET FlashProg PARTIE ENSEMBLE |
| | Collège Classe | | TITRE DU DOCUMENT ECLATÉ ET NOMENCLATURE GENERALE |
| | Nom Date | | |



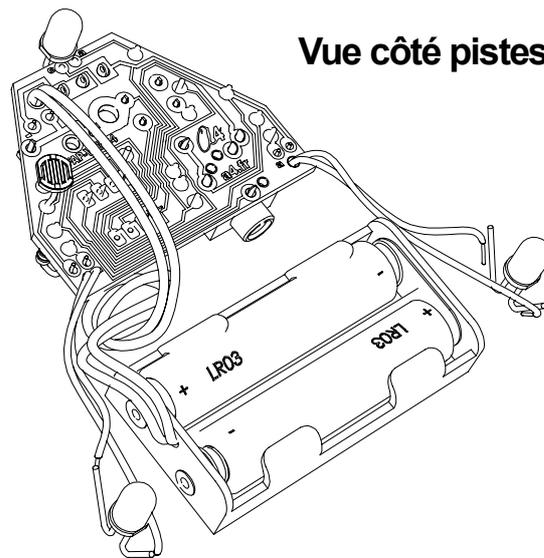
| | | | | | |
|--|--------------------|---|-----------|----------------------------|---------------------------|
|  TECHNOLOGIE AU COLLEGE | Echelle 1:1 |  | A4 | PROJET FlashProg | PARTIE ENSEMBLE |
| | Collège | | Classe | | TITRE DU DOCUMENT |
| | Nom | | Date | | DESSIN D'ENSEMBLE |



| | | | | | |
|---|---|---|-----------|----------------------------|--------------------------------------|
|  |  |  | A4 | PROJET FlashProg | PARTIE MODULE ELECTRONIQUE |
| | Collège | | Classe | | TITRE DU DOCUMENT |
| | Nom | | Date | | SCHEMA STRUCTUREL |



Vue côté composants



Vue côté pistes

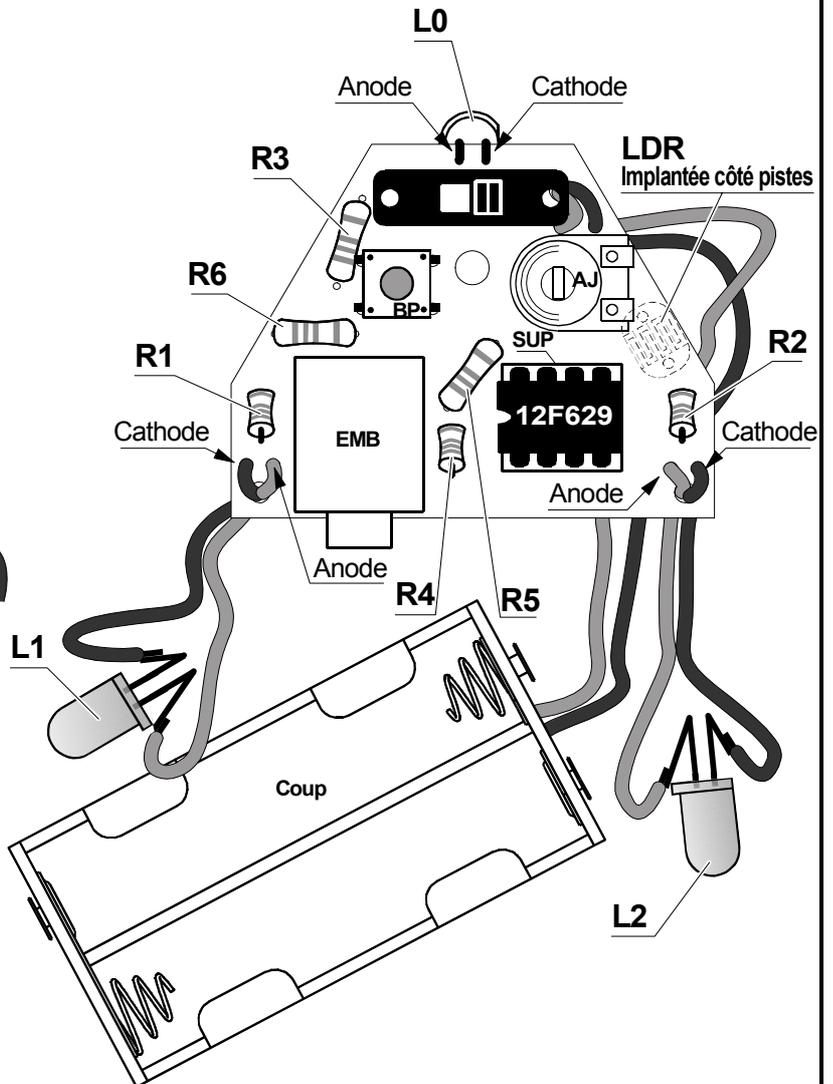
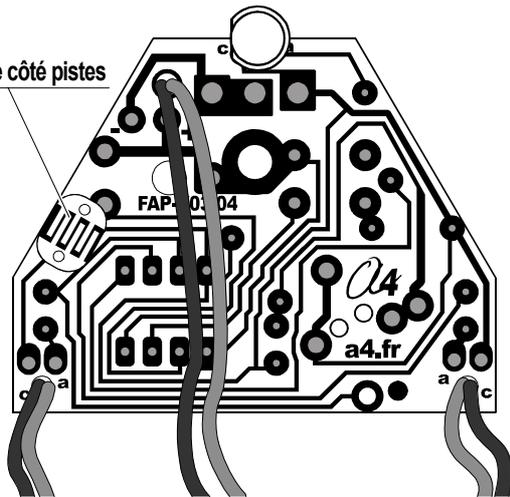
| | | | |
|---------|--------|------------------------------|---|
| EMB | 01 | Embase jack | Pour jack stéréo Ø 3 |
| S | 01 | Micro-inverseur | Micro-inverseur à glissière unipolaire |
| BP | 01 | Bouton poussoir | Bouton poussoir miniature de C.I. - 1 contact travail |
| LDR | 01 | Photo-résistor | 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 |
| | Collège Classe | | FlashProg | MODULE ELECTRONIQUE |
| | Nom Date | | TITRE DU DOCUMENT NOMENCLATURE DU MODULE ELECTRONIQUE | |

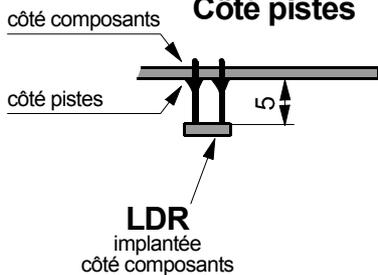


**Vue de dessous
côté pistes**

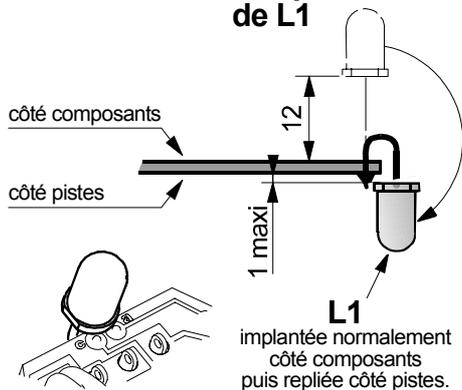
LDR
Implantée côté pistes



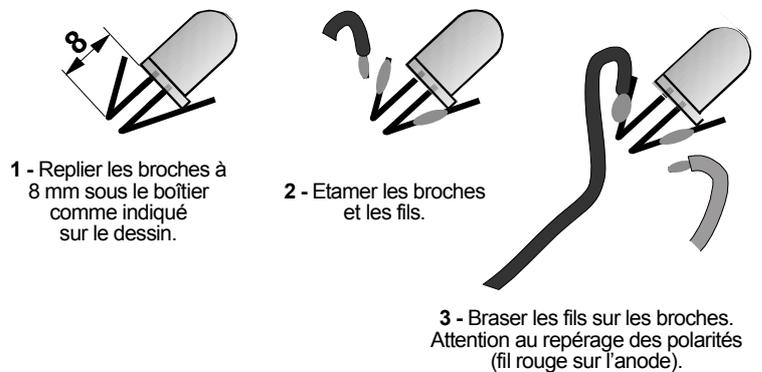
**Détail de l'implantation
de la LDR
Côté pistes**



**Détail de l'implantation
de L1**



Détail de l'implantation de L2 et L3

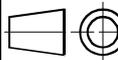


TECHNOLOGIE
AU COLLEGE

Echelle 1.5:1

Collège

Nom



Classe

Date

A4

PROJET

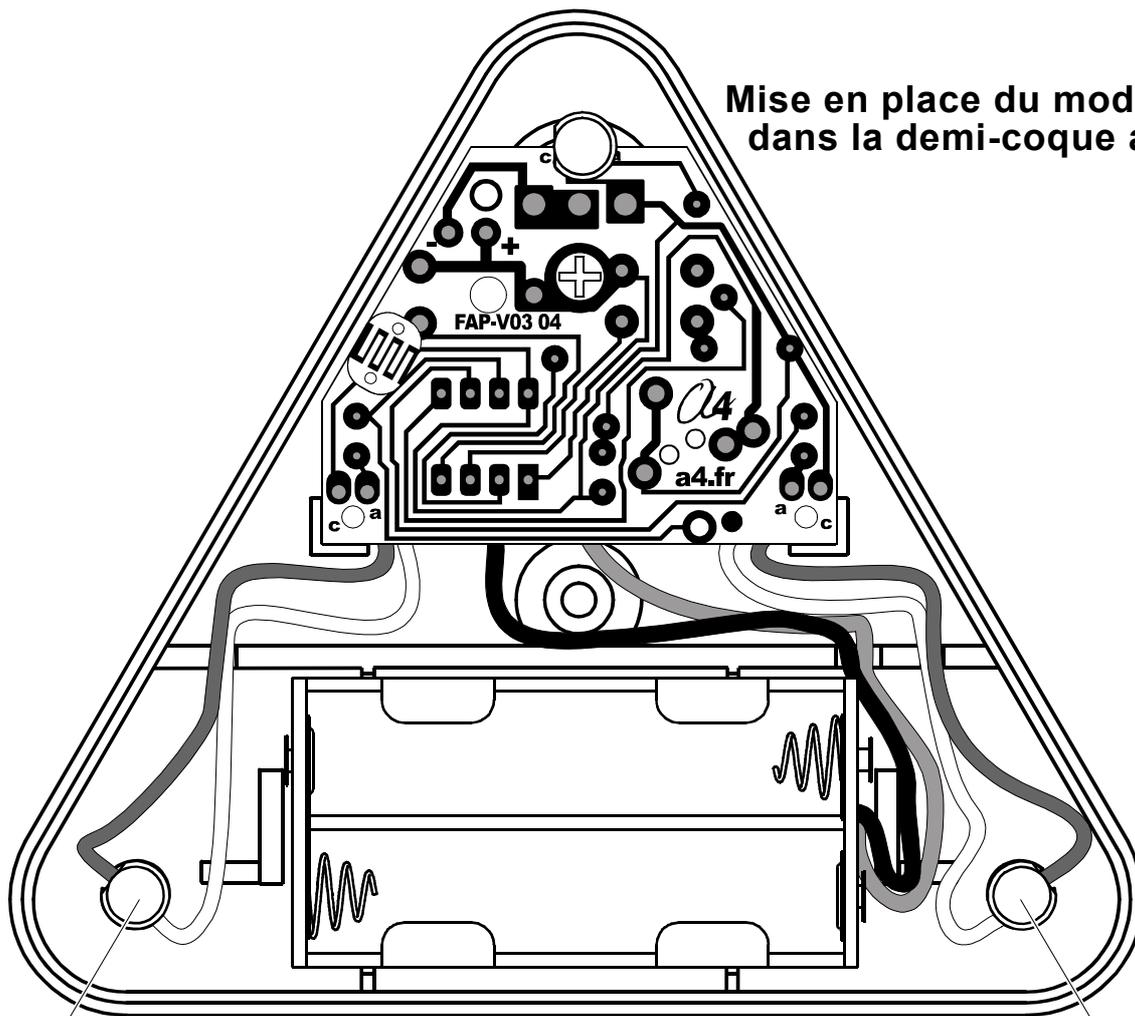
**FLASHADO
PROGRAMMABLE**

TITRE DU DOCUMENT

PLAN D'IMPLANTATION

PARTIE

**MODULE
ELECTRONIQUE**

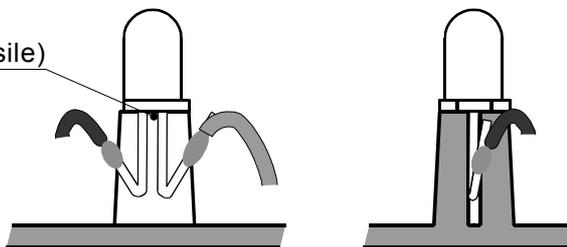


Mise en place du module dans la demi-coque arrière

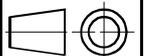
L1

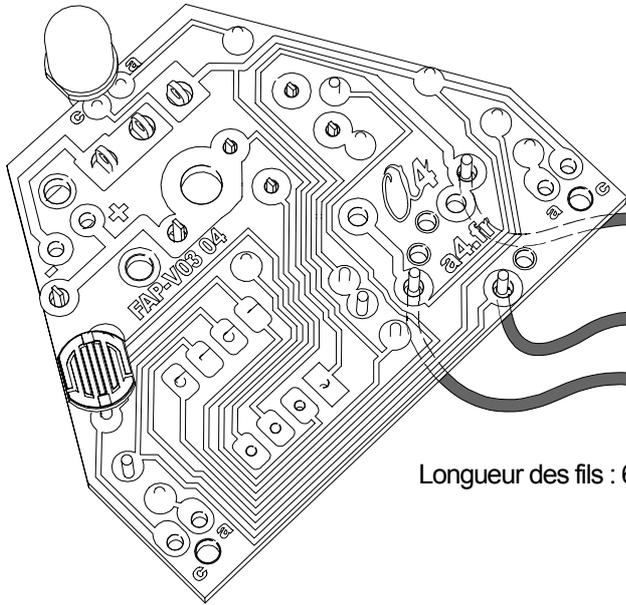
L2

Point de colle
(colle PVC
ou colle thermofusile)



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

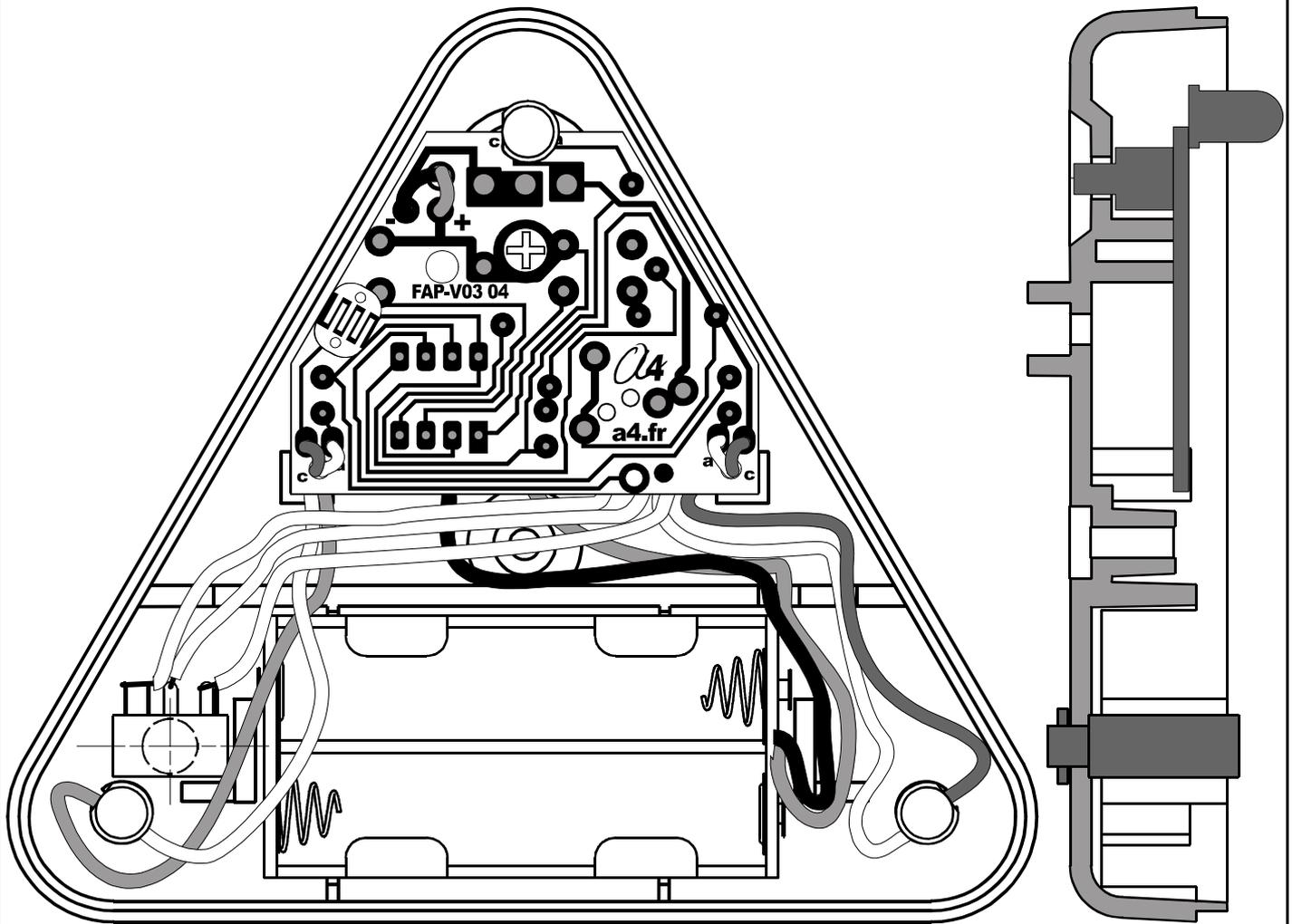
| | | | | | |
|--|---------------|---|-----------|---|---|
|  TECHNOLOGIE AU COLLEGE | Echelle 1.5:1 |  | A4 | PROJET FLASHADO PROGRAMMABLE | PARTIE MODULE ELECTRONIQUE |
| | Collège | Classe | | TITRE DU DOCUMENT | |
| | Nom | Date | | PLACEMENT DU MODULE DANS LE BOÎTIER | |

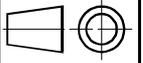


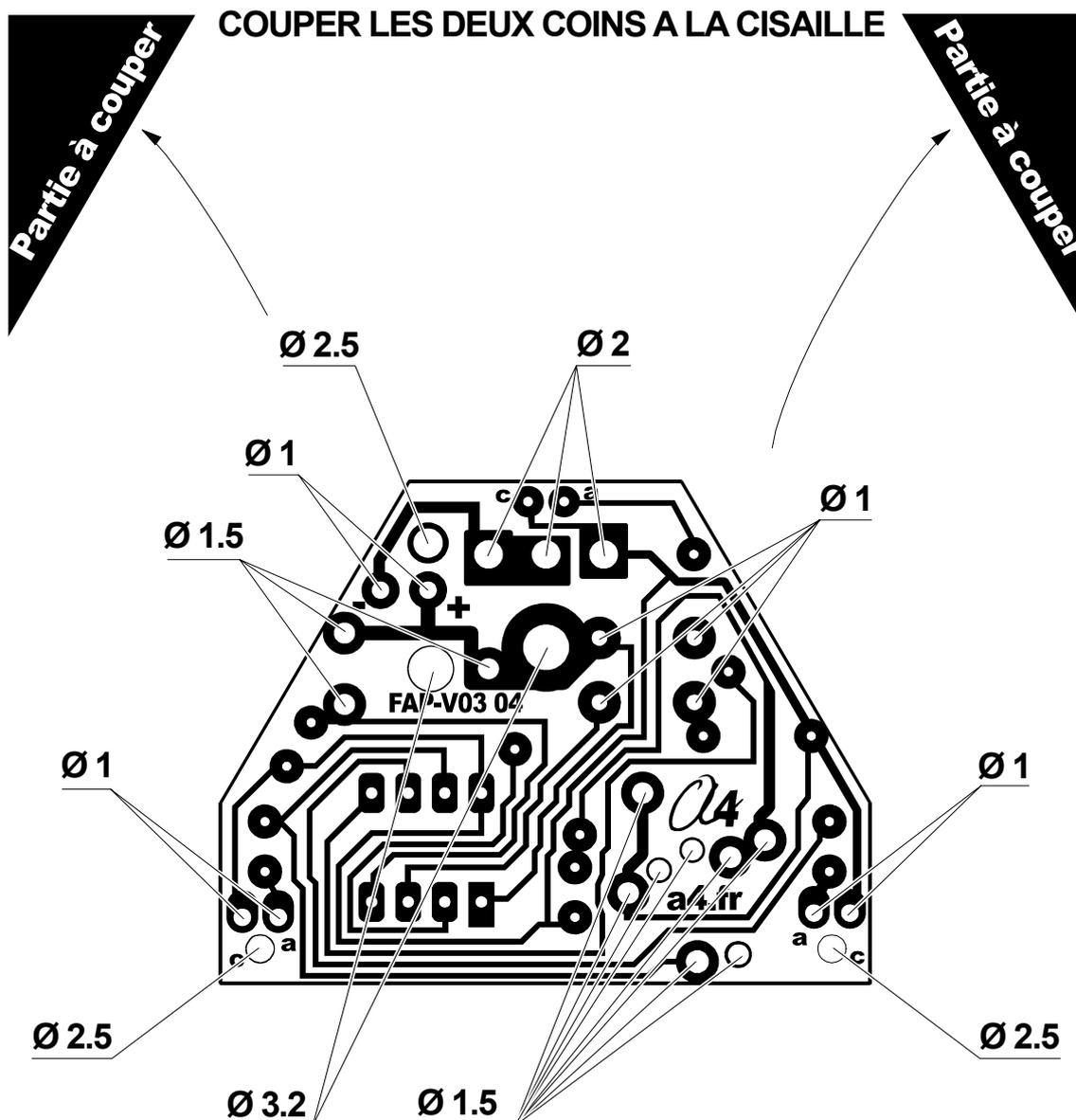
Afin de permettre la programmation plus facilement sans avoir à démonter le module, on peut déporter l'embase jack sur la demi-coque arrière du boîtier. Cela suppose un câblage du connecteur avec des fils et le perçage du boîtier au diamètre 6 comme indiqué sur ce document.

Longueur des fils : 60 mm.

Embbase jack

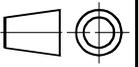


| | | | | | |
|--|----------------------|---|-------------------|--------|---|
|  TECHNOLOGIE AU COLLEGE | Echelle 1.5:1 |  | A4 | PROJET | PARTIE |
| | Collège | Classe | FlashProg | | MONTAGE DU MODULE ELECTRONIQUE |
| | Nom | Date | TITRE DU DOCUMENT | | MONTAGE DU CONNECTEUR DEPORTÉ SUR LE BOITIER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



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

| | | | | | |
|--|-------------|---|-----------|--|--|
|  TECHNOLOGIE AU COLLEGE | Echelle 2:1 |  | A4 | PROJET FLASHADO PROGRAMMABLE | PARTIE MODULE ELECTRONIQUE |
| | Collège | Classe | | TITRE DU DOCUMENT | |
| | Nom | Date | | PLAN DE PERCAGE ET DECOUPE DU CIRCUIT IMPRIME | |

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 - DEL 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 - DEL L1 et L2 | Selon plan d'implantation - Préparer les DEL (pliage et étamage des broches) - Préparer les fils (mise à longueur 70 + étamage) - Monter les DEL sur les fils (brasage) - Planter et brasier 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

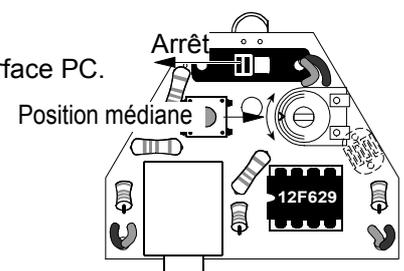
1 - Chargement d'un programme de test.

Vous pouvez vous reporter aux annexes du dossier pour plus de détails sur l'installation du logiciel Programming Editor et le chargement d'un programme.

Vous devez disposer :

- d'un ordinateur sur lequel vous avez installé l'application Programming Editor ;
- 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.

- Lancer l'application Programming Editor. A l'ouverture de l'application, paramétrer "PICAXE 8" (qui correspond au micro-contrôleur du FlashProg).
- Ouvrir sur le CDRom le programme test "TEST.cad" (C'est un fichier de type "diagramme").
- Déplacer l'interrupteur à glissière du module en position arrêt (vers la gauche).
- Régler l'ajustable sur la position médiane.
- Placer des piles dans le support du module.
- Connecter le module au port COM1 de l'ordinateur par l'intermédiaire du cordon interface PC.
- Déplacer l'interrupteur du module en position marche (vers la droite).



- Déplacer l'interrupteur du module en position marche (vers la droite).
- Presser la touche F5 du clavier de l'ordinateur (l'ordinateur convertit le diagramme en langage basic).
- 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".
- Eteindre le module FlashProg et le déconnecter de son câble interface PC.

2 - Test du bon fonctionnement du module électronique FlashProg

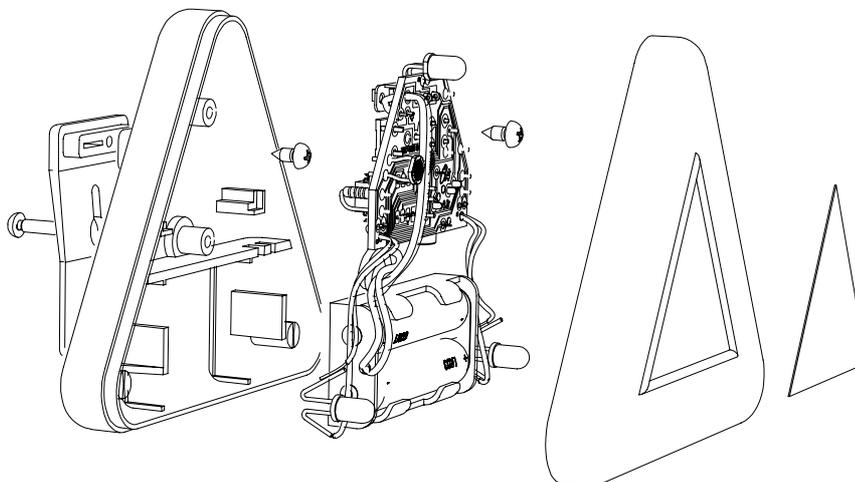
- A - Allumer le Module FlashProg en déplaçant le bouton de son interrupteur :
Les 3 DEL doivent s'allumer successivement (effet chenillard).
En cas d'échec, retirer les piles du module sans tarder et réparer la panne du module avant un nouvel essai.
- B - Appuyer une fois sur le bouton poussoir :
Le mode de clignotement doit changer : les 3 DEL 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.

- 01 - Dégrapper et ébavurer les 3 pièces du boîtier.
- 02 - Monter l'agrafe en place et la fixer par une vis 3x6,5.
- 03 - Monter le module en place et le fixer par une vis 3x6,5.
- 04 - Placer les DEL 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.
- 05 - Découper l'autocollant rétro réfléchissant en triangle équilatéral de côté 50 mm et le poser en place sur la demi-coque avant du boîtier.
- 06 - Assembler les deux demi-coques du boîtier et les maintenir par la vis 3x16.



C – PROGRAMMATION

| | |
|---|---------|
| Introduction | 17 |
| Automatisme niveau 1 Choisir un programme dans une liste et le charger | 18 à 34 |
| Automatisme niveau 2 Modifier un programme | 35 à 47 |
| Automatisme niveau 3 Créer un programme | 48 à 59 |

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 DEL).

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 «Programming Editor» 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 « Programming Editor »

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 «Programming Editor» 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 oeuvre du logiciel de programmation « Programming Editor »

Pour procéder à la programmation du module du FlashProg, vous devez disposer du CDROM «Programming Editor (Réf. : CD-FPPE), d'un ordinateur (Windows 95, 8Mb RAM mini) et du câble spécifique (Réf. : CABLE-FP) permettant de transférer vos programmes dans le module FlashProg. L'installation du logiciel est décrite en Annexe A de ce dossier.

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 CDROM CD-FP.

Sommaire des fiches élèves :

| Page | | Description | Fichier programme |
|----------|-------------|--|--------------------------|
| 19 | FICHE N° 1 | Pilotage des sorties : Clignotement simultané des trois DEL. | CLIGNO DEL.cad |
| 20 | FICHE N°2 | Pilotage des sorties : Clignotement alterné des trois DEL (effet chenillard). | CHENILLARD.cad |
| 21 | FICHE N°3 | Pilotage des sorties : Bâton tournant. Les DEL s'allument deux à deux en tournant. | BATON TOURNANT.cad |
| 22 | FICHE N°4 | Enchaînement de séquences : Alternance du clignotement simultané des trois DEL et d'un effet chenillard. | CLIGNO-CHENILLARD.cad |
| 23 | FICHE N°5 | Utilisation du capteur photosensible : Mise en route automatique à la tombée de la nuit. | DEMAR AUTO NUIT.cad |
| 24 | FICHE N° 6 | Utilisation du capteur photosensible : Détection d'un phare de voiture dans la nuit et changement automatique de mode de clignotement. | DETEC PHARE.cad |
| 25 | FICHE N° 7 | Utilisation du capteur Bouton Poussoir : Sélection alternée de 2 effets par le bouton poussoir. | SELECTION PAR BP.cad |
| 26 | FICHE N° 8 | Utilisation du capteur photosensible et Bouton Poussoir : Démarrage automatique dès qu'il fait nuit ou dès que l'on active le bouton poussoir. | DEMAR AUTO LDR-BP.cad |
| 27 28 | FICHE N° 9 | Comptage de séquences : Répétition de deux séquences différentes. | REPET 2 SEQUENCES.cad |
| 29 30 | FICHE N° 10 | Utilisation d'un temps de pause variable : Effet d'accélération d'un chenillard suivi par un clignotement. | ACCELERATION.cad |
| 31 | FICHE N° 11 | Utilisation d'un temps d'attente long : Exécution d'une séquence chenillard une fois toutes les 3 secondes le jour et en permanence la nuit. | CHENILLARD JOUR-NUIT.cad |
| 32 | FICHE N° 12 | Utilisation d'une variable aléatoire : Clignotement aléatoire. | HASARD.cad |
| 33 34 | FICHE N° 13 | Utilisation d'un sous programme : Accélération d'un chenillard commandée par le bouton poussoir. | ACCELERATION PAR BP.cad |

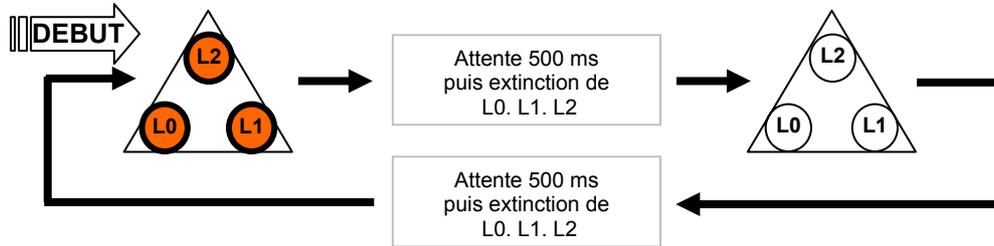
Travail élève

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

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

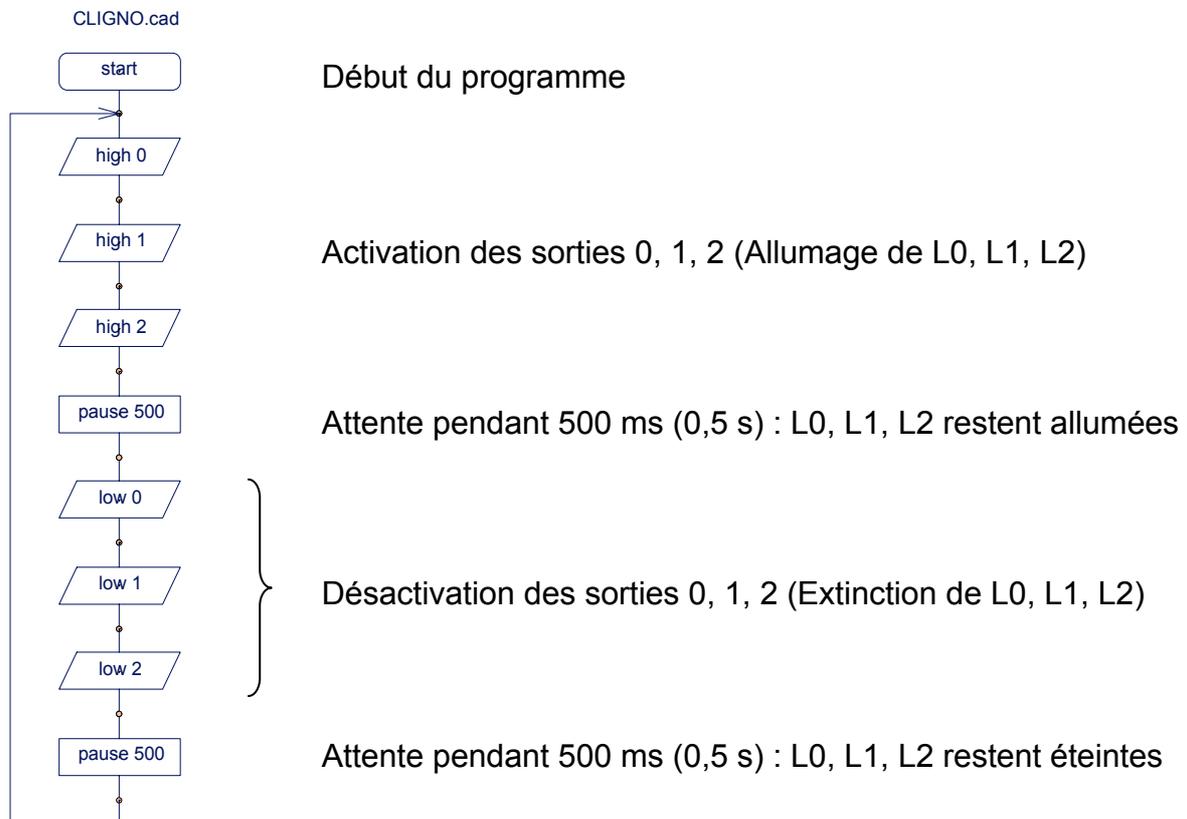
Synoptique :



Commentaire :

Dès la mise sous tension L0, L1, L2 sont allumées. Au bout de 500 ms elles sont éteintes. Un temps d'attente de 1500 ms est introduit avant de les allumer de nouveau. Le programme reboucle indéfiniment sur lui même.

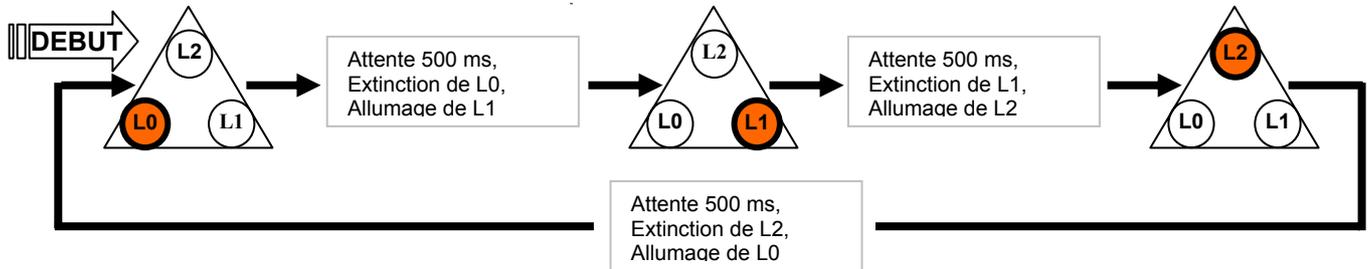
Diagramme de programmation :



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

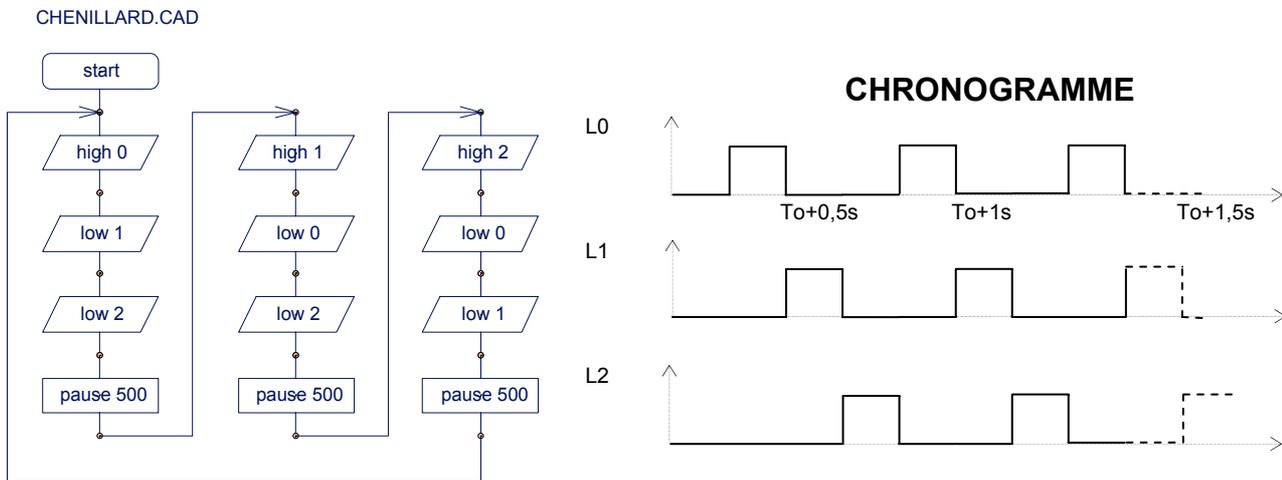
Synoptique :



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 :



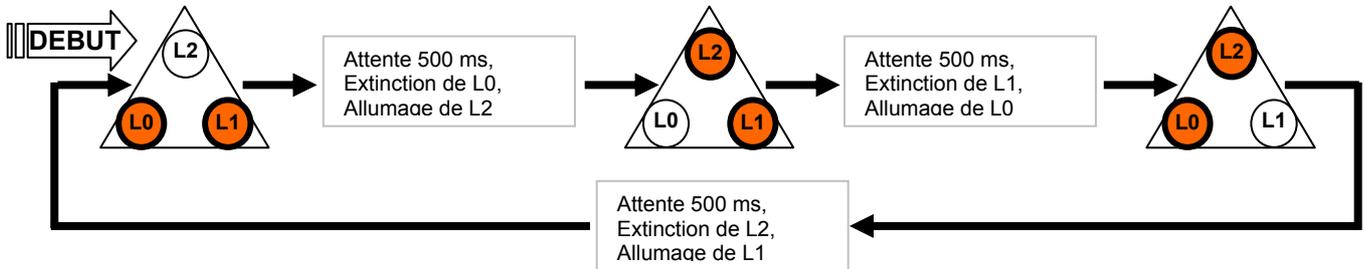
Note : A la mise sous tension du module (première exécution du programme), toutes les sorties sont initialisées à l'état bas (L0, L1, L2 sont éteintes). Une sortie qui est activée à l'état haut (ex. « high0 ») reste active jusqu'à ce que le programme rencontre une instruction qui la désactive (ex. low0).

FICHE N°3 : programme BATON TOURNANT.cad

But du programme : allumer momentanément une paire de DEL 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 :

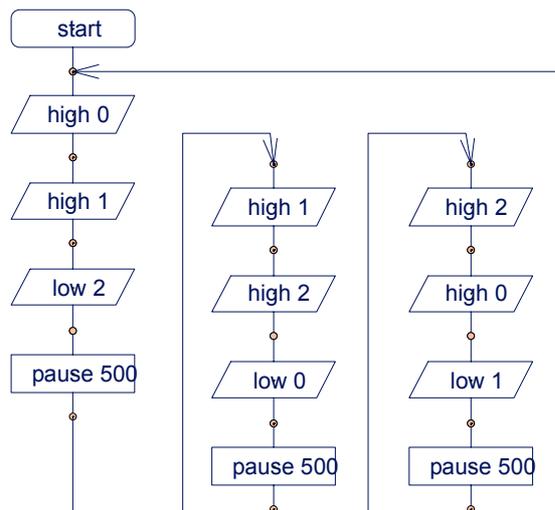


Commentaire :

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 :

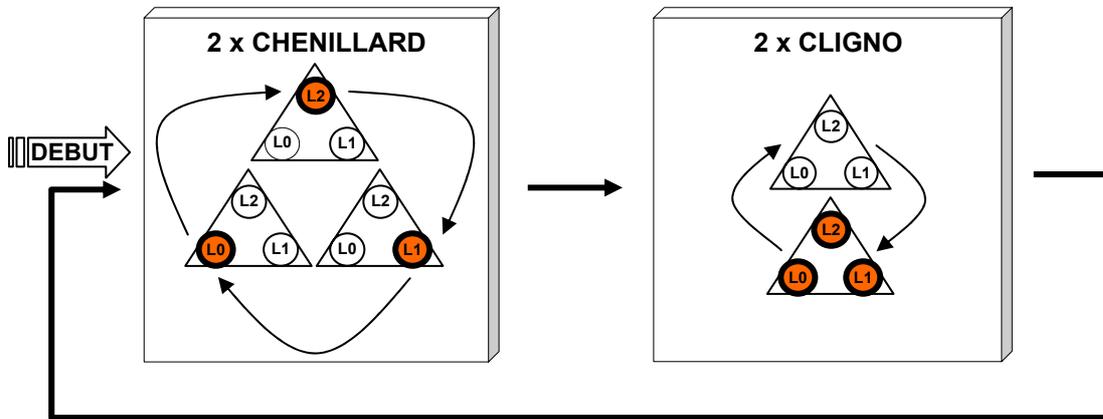
BATON TOURNANT.cad



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

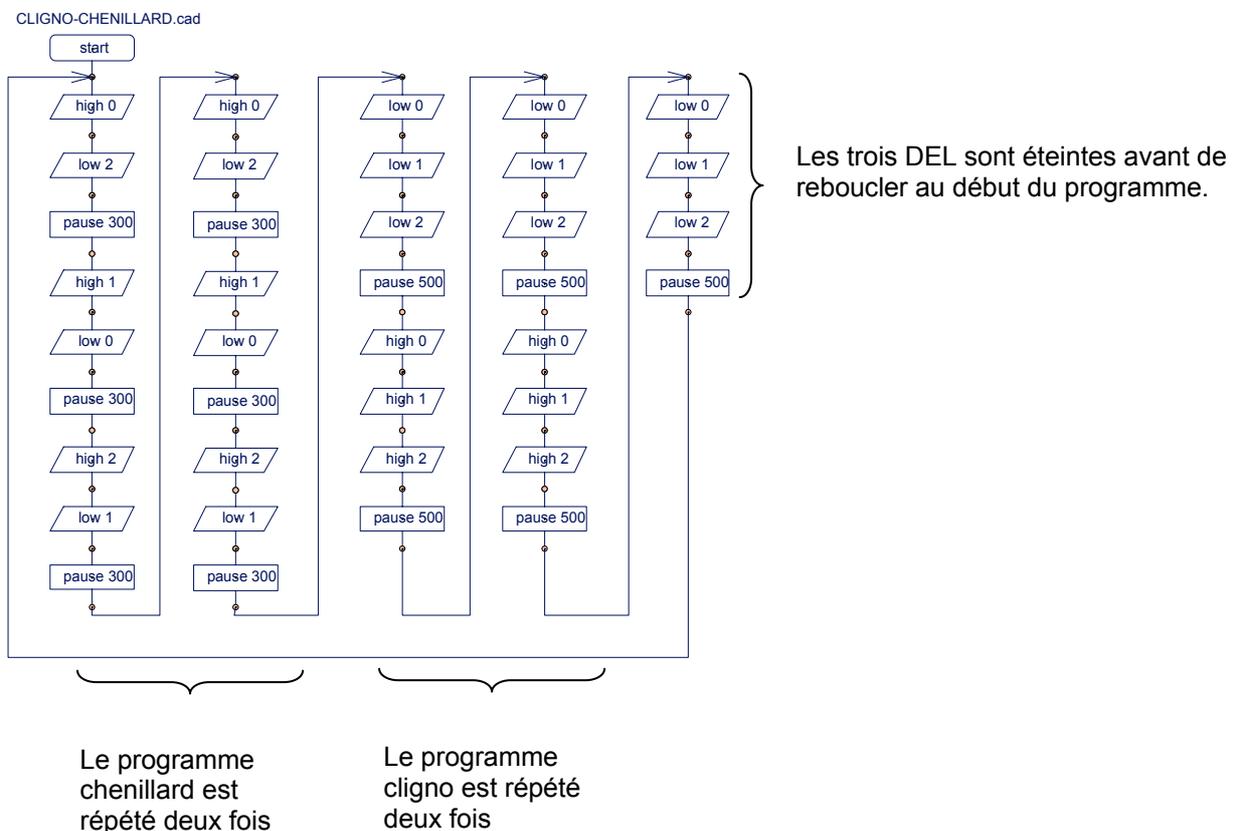
Notion de programmation abordée : activation / désactivation de plusieurs sorties, utilisation d'un temps d'attente, répétition de séquences.

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).

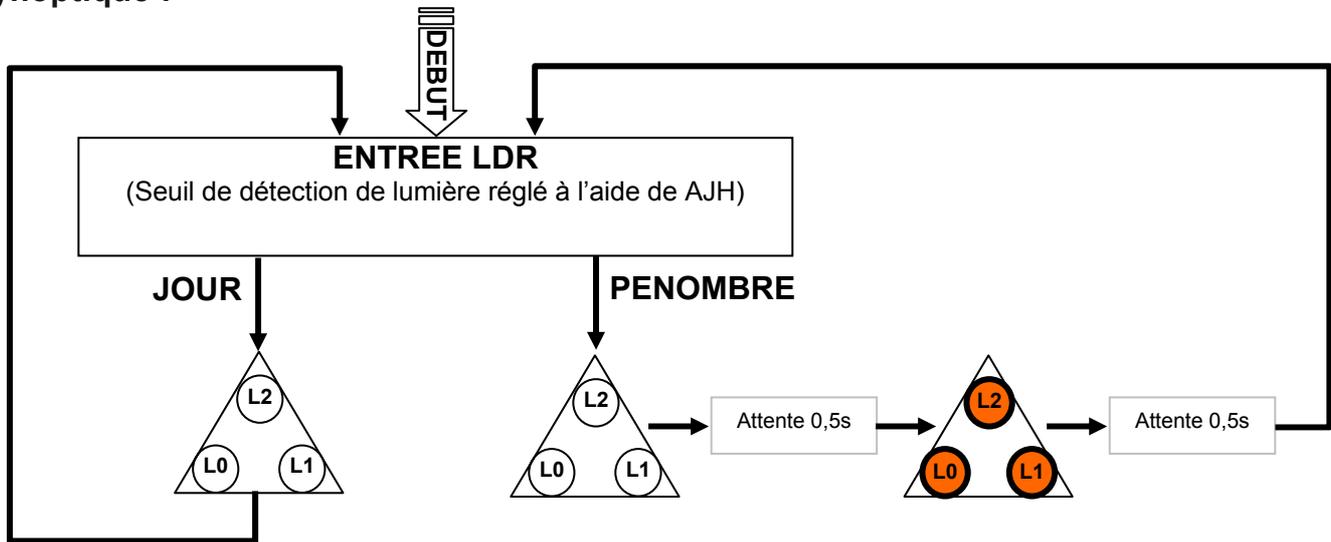
Diagramme de programmation :



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 :



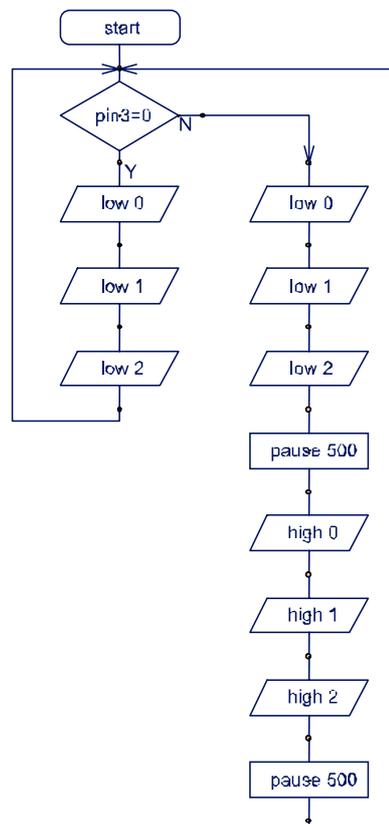
Commentaire :

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

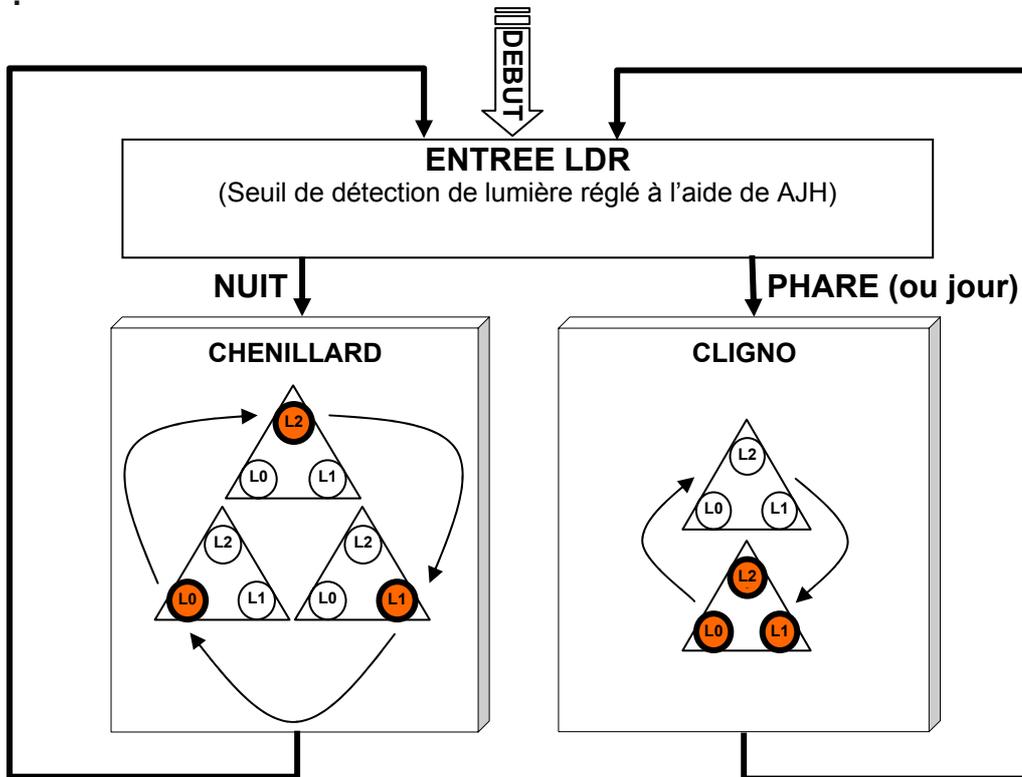
DEMAR AUTO NUIT.cad



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 :



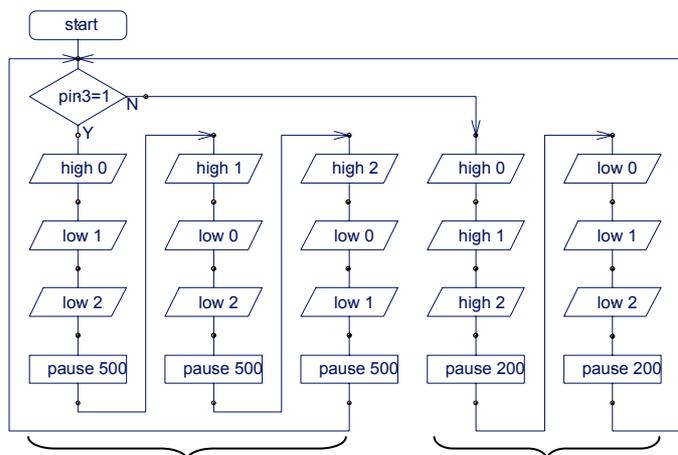
Commentaire :

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 :

DETEC PHARE.cad



La séquence « chenillard » est exécutée si le niveau lumineux ambiant est sombre (nuit).

La séquence « cligno » est exécutée si le niveau lumineux ambiant est lumineux (jour).

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

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

Synoptique :

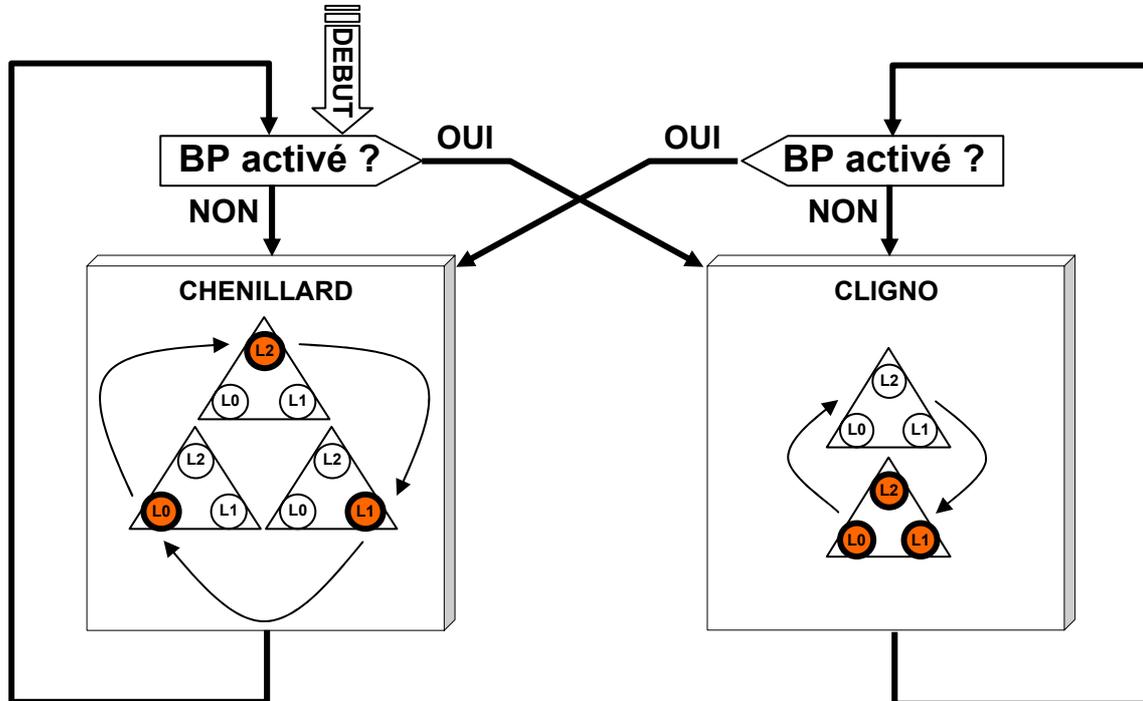
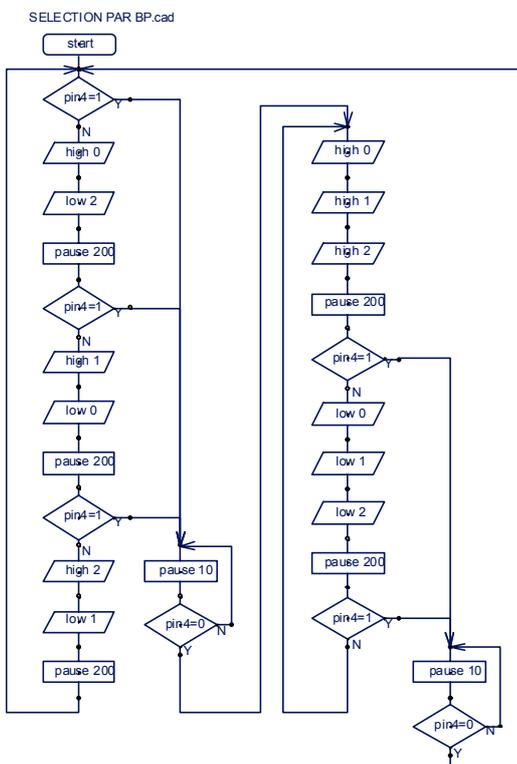


Diagramme de programmation :



Commentaire :

Il s'agit ici de tester l'état du bouton poussoir et de basculer soit vers la séquence CLIGNO soit 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 (pin4=1 est vrai), on attend 10 ms que son état se stabilise ; en effet, lorsqu'il est activé, celui-ci génère une multitude d'impulsions (rebonds) qui pourraient être interprétés 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.

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 gèrent une séquence d'allumage des DEL.

Synoptique :

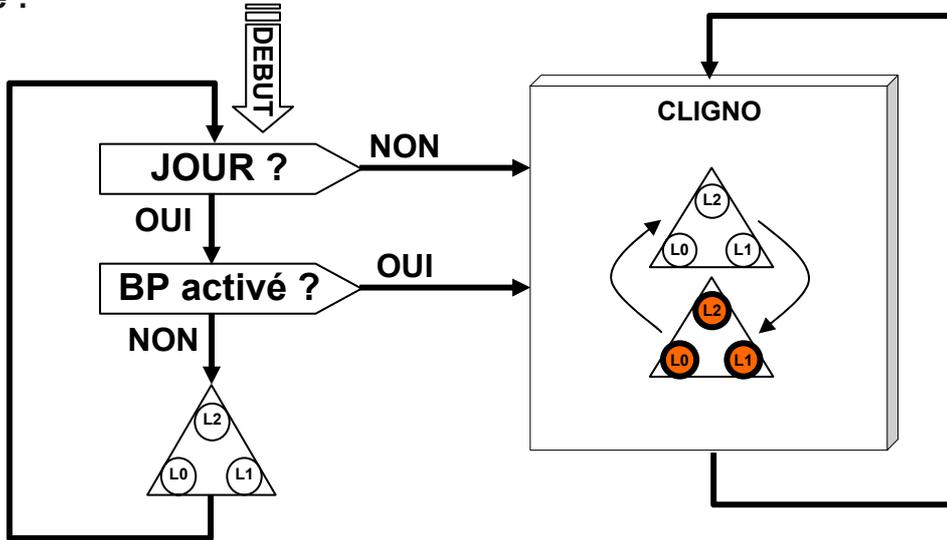
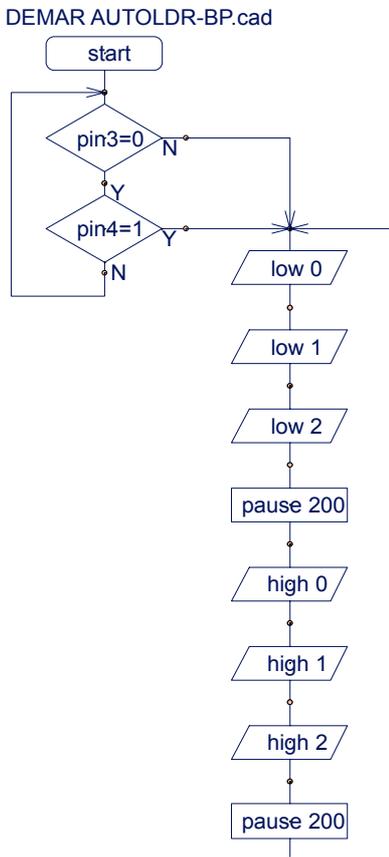


Diagramme de programmation :



Commentaire :

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

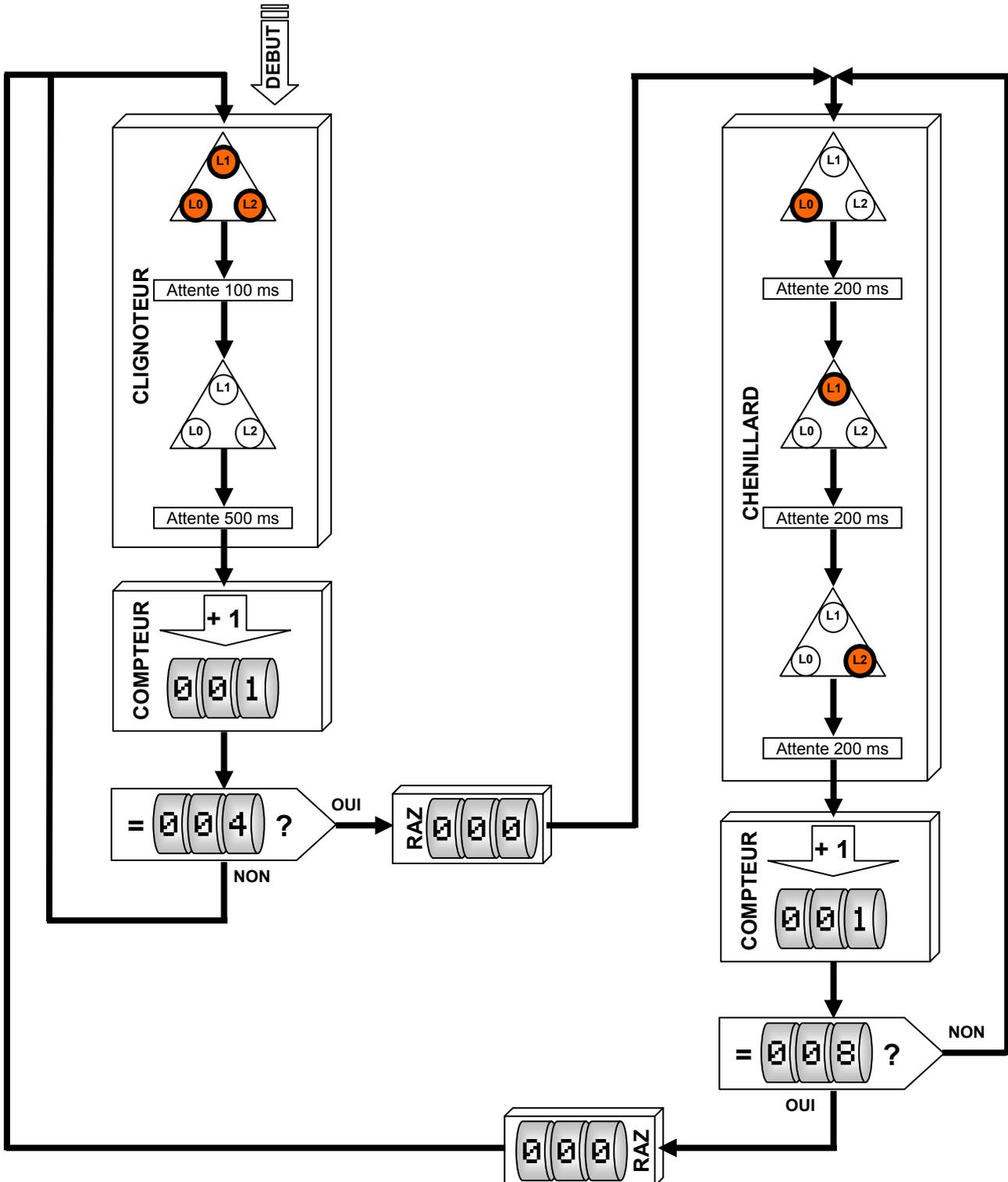
Lorsqu'il fait jour, l'entrée LDR voit un niveau logique bas (pin3=0 est vrai) ; le programme teste alors l'entrée bouton poussoir. Tant que le bouton poussoir n'est pas activé (pin4=1 est faux), le programme reboucle sur le test de la LDR.

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.

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.



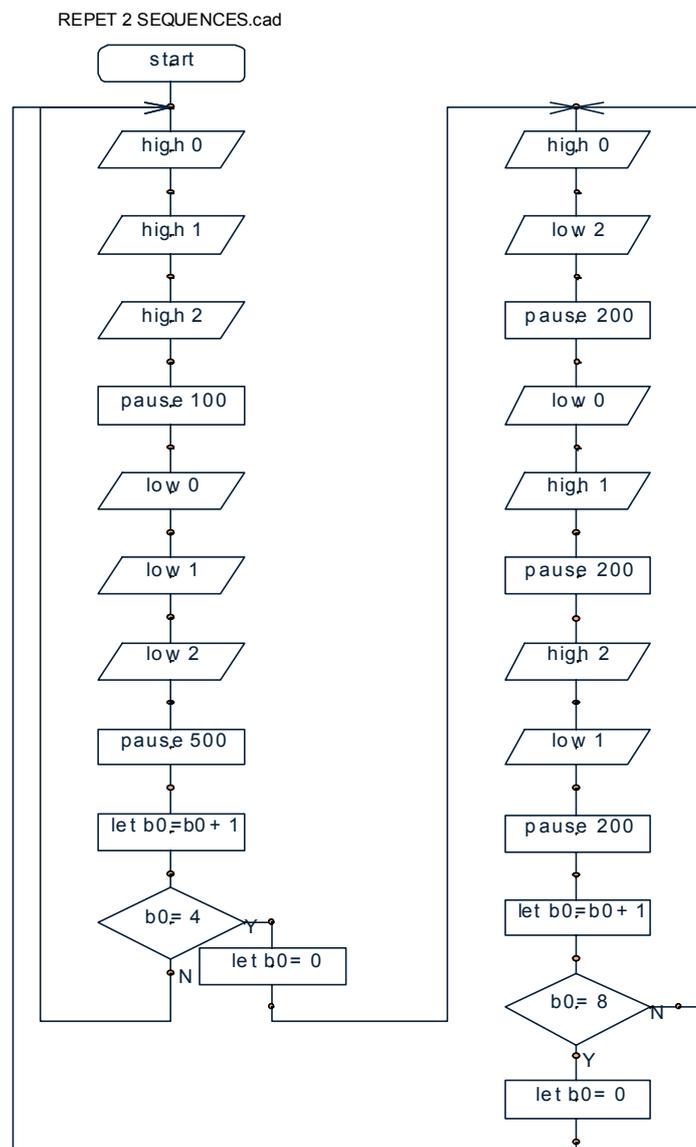
Commentaire :

Dès la mise sous tension le programme « CLIGNO » est exécuté. Après chaque exécution de ce programme, un compteur nommé b zéro (b0) 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 b0 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 du 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 DELs. Le test de la variable b0 permet de déterminer le nombre d'itérations de chaque séquence.

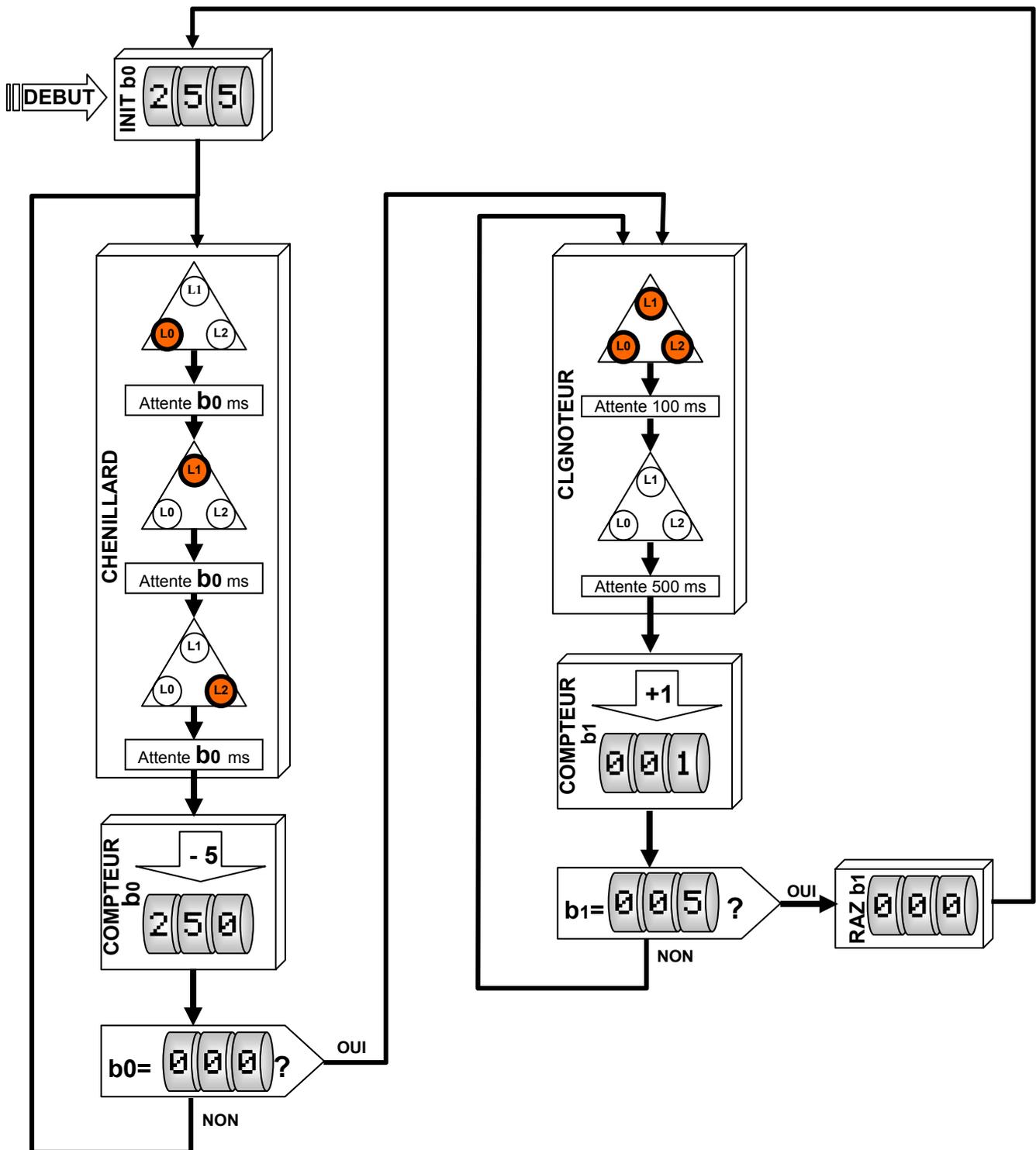
Diagramme de programmation (programme REPET 2 SEQUENCES.cad :



But du programme : provoquer un effet chenillard qui s'accélère progressivement et qui est suivi par un effet clignotement simultané des DEL 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.

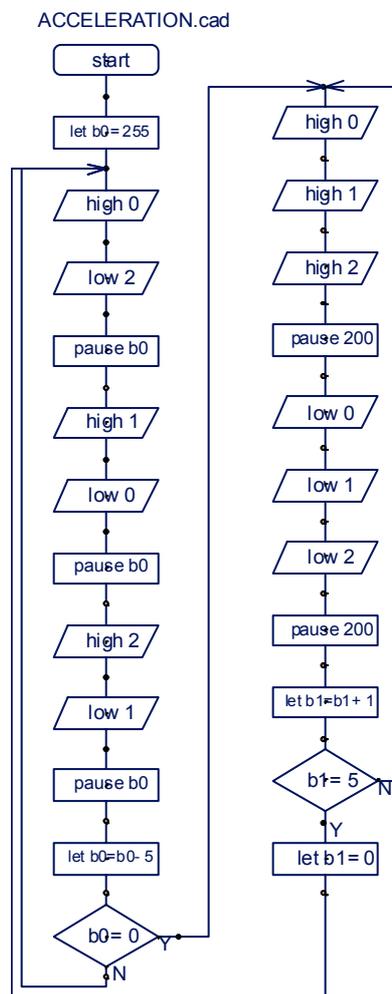
Synoptique :



Commentaire :

Dès la mise sous tension le programme initialise une première variable nommée b0 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 b0. Tant que b0 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 b1 est incrémentée. Tant que la variable b1 n'a pas atteint la valeur 5, le programme reboucle sur cette séquence. Sinon, le programme réinitialise la première variable b0 à 255 et enchaîne de nouveau la séquence « chenillard » . . .

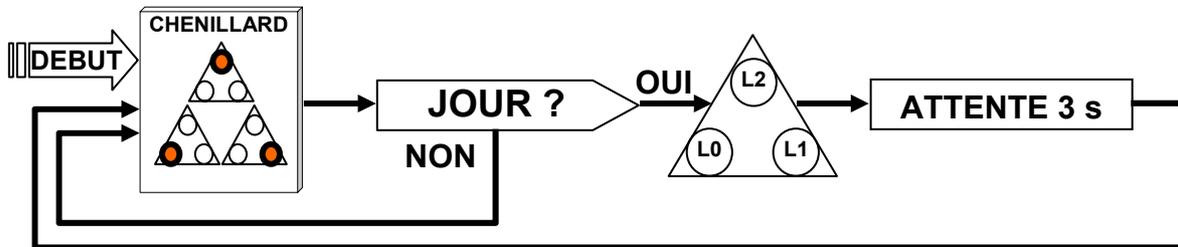
Diagramme de programmation :



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 :

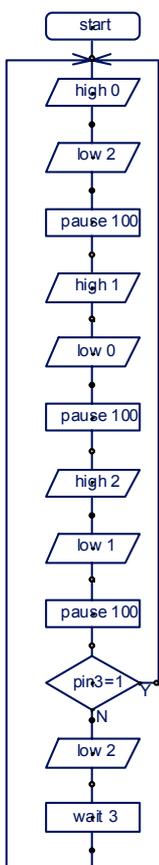


Commentaire :

Une séquence « chenillard » est exécutée. A la fin de cette séquence, le programme vérifie l'état de l'entrée LDR ; si 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 :

CHENILLARD JOUR-NUIT.cad



L'instruction « pause » permet d'introduire un temps d'attente « court » exprimé en millisecondes (il peut être compris entre 1 et 65 535 millisecondes).

Note importante : si l'on charge un programme qui fait appel à l'instruction « wait » 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 (F5) (l'ordinateur essaye de dialoguer avec le module), puis de mettre immédiatement le module sous tension.

L'instruction « wait » permet d'introduire un temps d'attente « long » exprimé en secondes (il peut être compris entre 1 et 65 secondes).

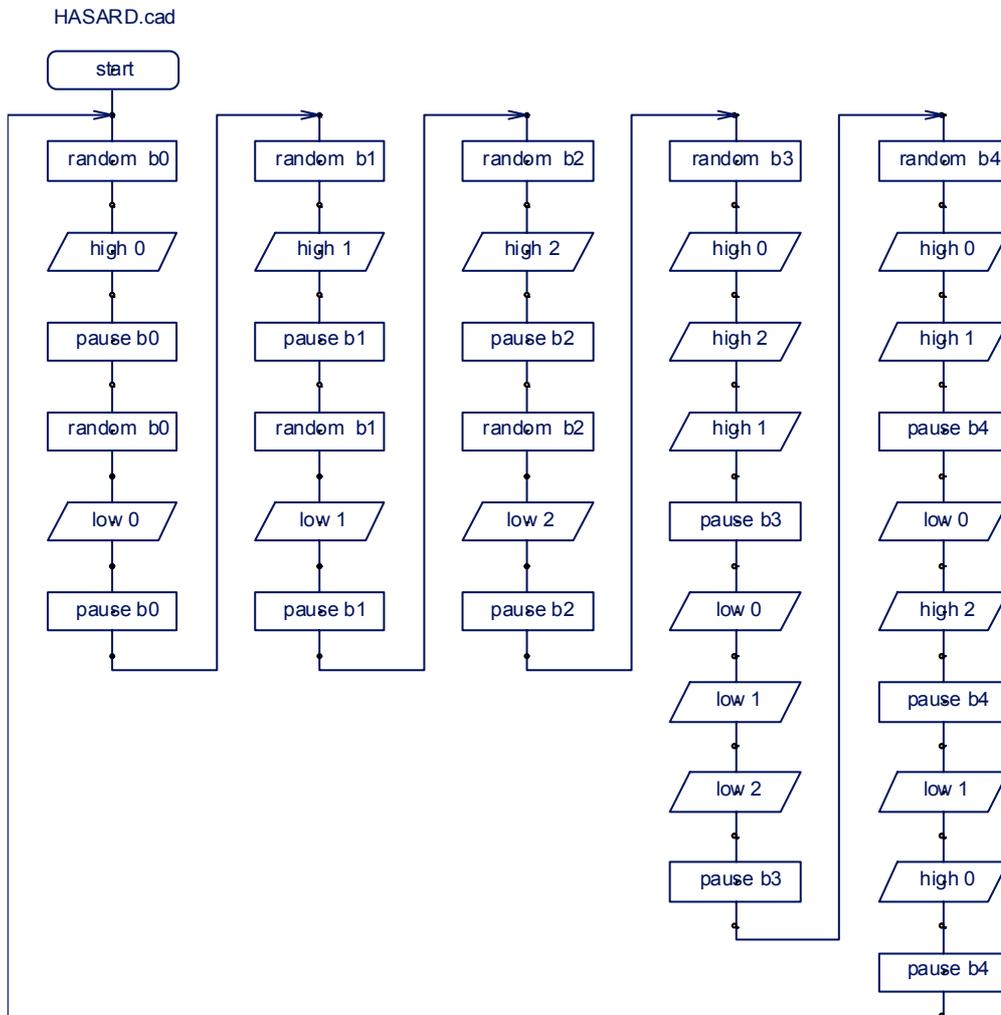
But du programme : enchaîner plusieurs séquences en introduisant des temps d'attente dont la valeur est aléatoire.

Notion de programmation abordée : activation / désactivation de plusieurs sorties, génération de variables pseudo aléatoires, utilisation de temps d'attente fonction de ces variables.

Commentaire :

L'instruction « random bx » permet de générer une valeur pseudo aléatoire (le microcontrôleur détermine lui-même une valeur et initialise la variable bx avec cette valeur). On génère ici cinq valeurs pseudo aléatoires (b0, b1, b2, b3, b4) qui sont alors utilisées pour introduire de temps d'attente aléatoire (pause b0, pause b1, . . .). On chaîne différentes séquences d'allumage des DEL qui sont ponctuées par des temps aléatoires.

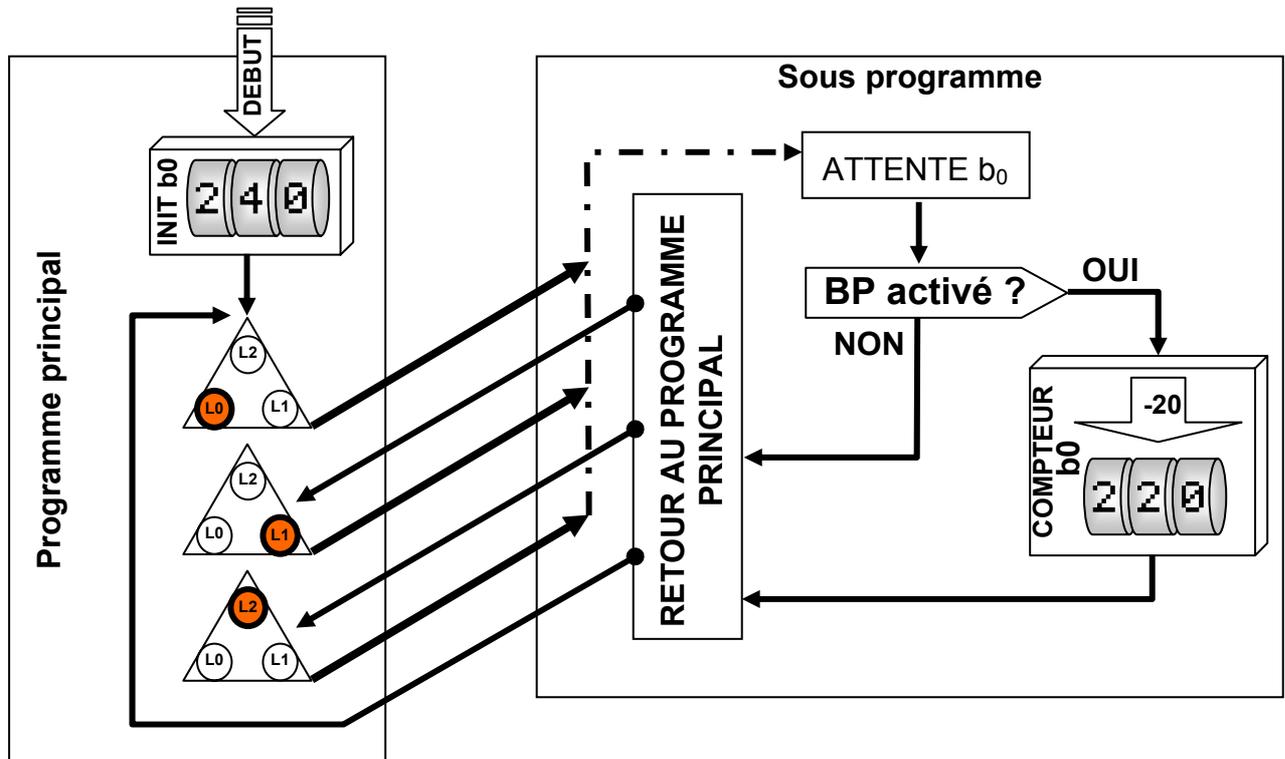
Diagramme de programmation :



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 d'une variable b_0 .

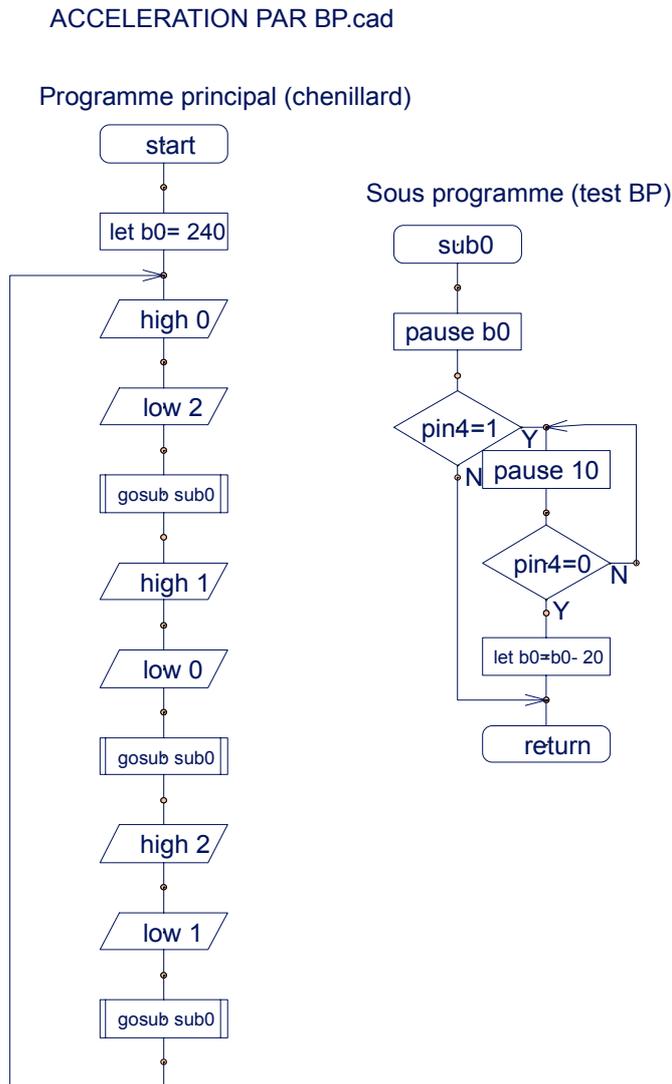
Synoptique :



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 DEL). 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.

Diagramme de programmation :



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

L'instruction « gosub sub0 » introduite dans le programme principal demande au programme d'exécuter le sous programme « sub0 ».

Le sous programme gère le temps d'attente (« pause b0 ») qui contrôle la durée d'allumage ou d'extinction des DEL. 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 « return » 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 « gosub sub0 ».

Note : « gosub sub0 » signifie « go to subroutine N°0 » c'est-à-dire aller au sous programme N°0.

« sub0 » signifie « subroutine N°0 ». Cette instruction marque le point de départ du sous programme N°0.

« return » 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 CDROM CD-FP).

Sommaire des fiches élèves :

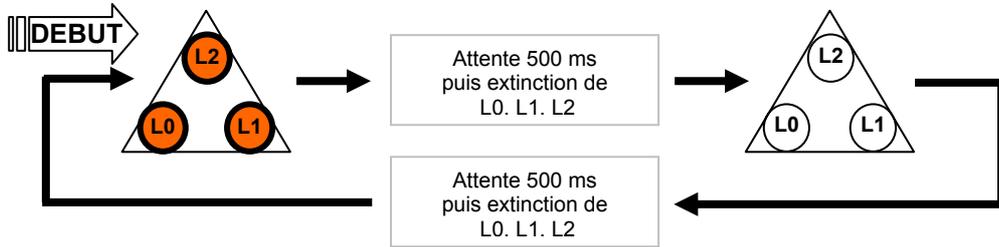
| Page | | Programme à charger | Consigne |
|----------|-------------|-----------------------------|--|
| 36 37 | FICHE N° 21 | FLASH COURTS.cad | Réduire la durée d'allumage des DEL. |
| 38 39 | FICHE N° 22 | ACCELERER CHENILLARD.cad | Accélérer la vitesse de défilement du chenillard |
| 40 41 | FICHE N° 23 | INVERSER SENS ROTATION.cad | Inverser le sens de rotation du chenillard |
| 42 43 | FICHE N° 24 | MODIFIER REPETITION.cad | Répéter l'exécution du programme clignoteur cinq fois et celle du programme chenillard vingt fois. |
| 44 45 | FICHE N° 25 | INVERSER MODE JOUR-NUIT.cad | Inverser le mode de fonctionnement jour / nuit. |
| 46 47 | FICHE N° 26 | DEMAR AUTO NUIT.cad | Démarrage automatique lorsqu'il fait nuit. |

Travail élève :

- 1** → Lancer le logiciel Programming Editor 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.

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

Synoptique :

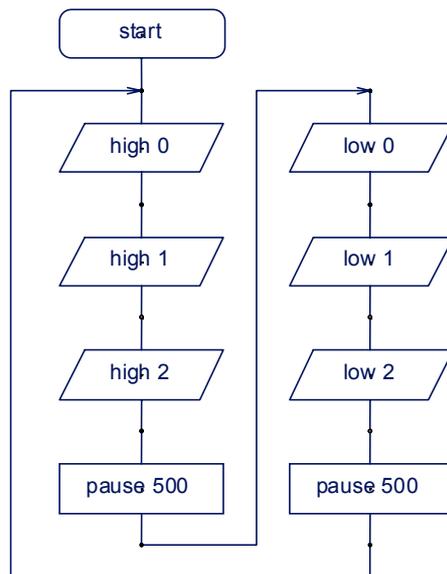


Consignes :

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

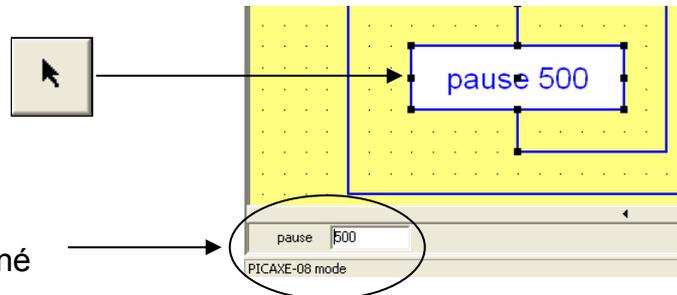
Diagramme de programmation :

FLASHS COURTS.cad



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

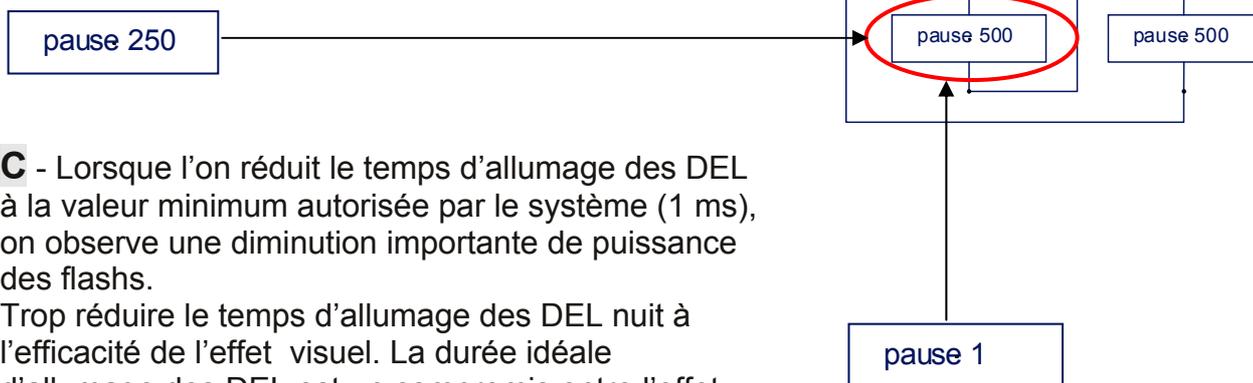
Pour modifier la valeur d'un bloc d'un diagramme de programmation, il faut d'abord sélectionner à l'aide de l'outil « flèche » le bloc que l'on souhaite modifier.



L'information du coin bas gauche de la fenêtre indique le type de bloc sélectionné et permet de saisir une nouvelle valeur.

A - L'instruction qui détermine le temps d'allumage des DEL est l'instruction « pause 500 » située immédiatement après les instructions qui provoquent l'allumage des DEL. Dès que les DEL sont allumés, le programme attend 500 ms avant d'exécuter les instructions qui vont éteindre les DEL.

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

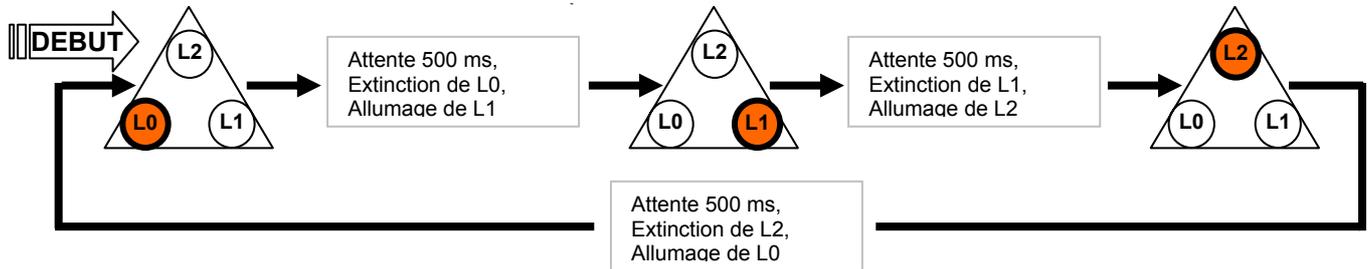


C - Lorsque l'on réduit le temps d'allumage des DEL à la valeur minimum autorisée par le système (1 ms), on observe une diminution importante de puissance des flashes.

Trop réduire le temps d'allumage des DEL nuit à l'efficacité de l'effet visuel. La durée idéale d'allumage des DEL est un compromis entre l'effet souhaité pour répondre aux exigences du FlashProg (être visible de loin) et conserver une autonomie de fonctionnement acceptable.

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

Synoptique :

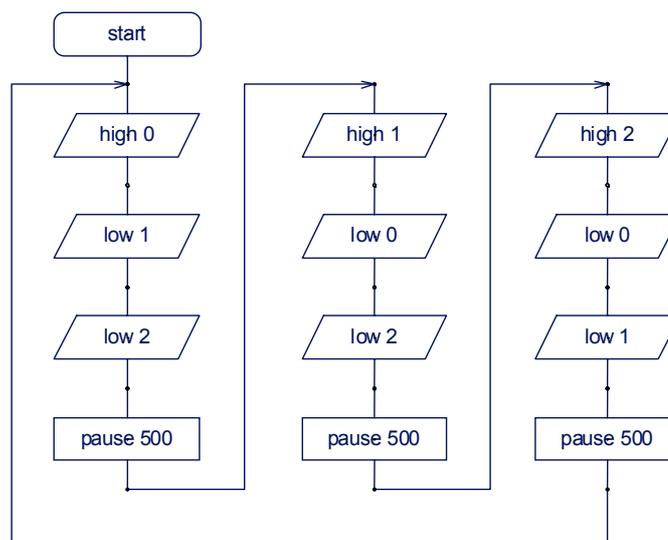


Consignes :

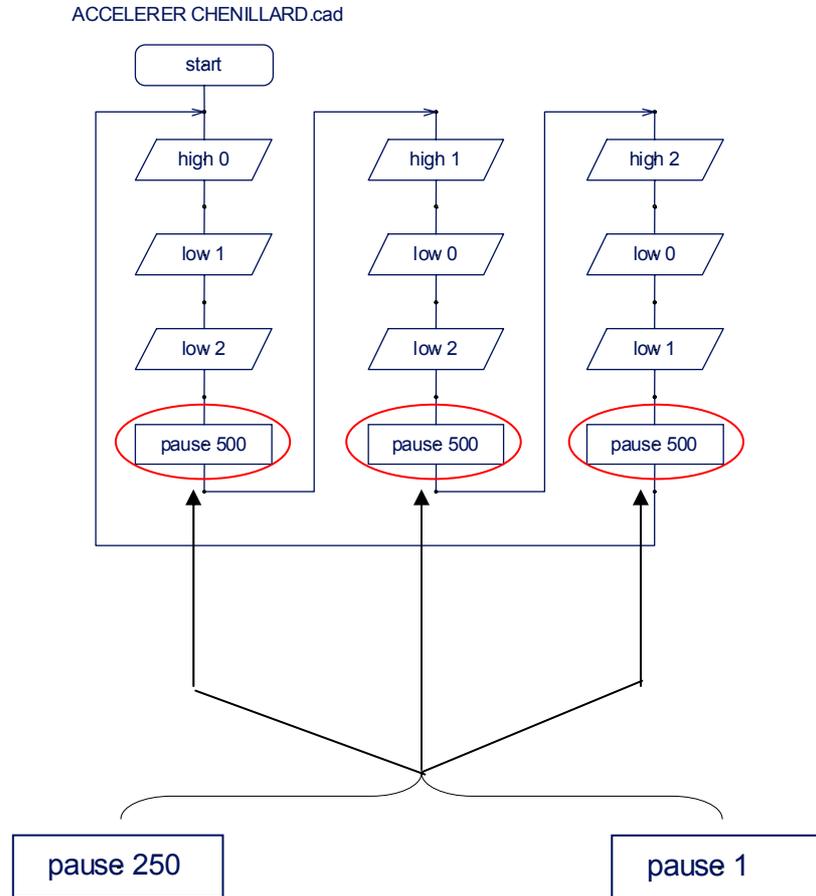
- A** - Entourer sur le diagramme ci-dessous la (les) instruction(s) qui détermine(nt) la vitesse de défilement du chenillard.
- B** - Modifier le diagramme à l'écran afin de doubler la vitesse du chenillard.
Charger le programme dans le FlashProg et vérifier que le chenillard tourne deux fois plus rapidement.
- 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 :

ACCELERER CHENILLARD.cad



A - La vitesse de défilement du chenillard est déterminée par les trois instructions « pause 500 ». Les DEL s'allument successivement pendant 0,5 s.
La période d'un cycle complet est de 1,5 s.

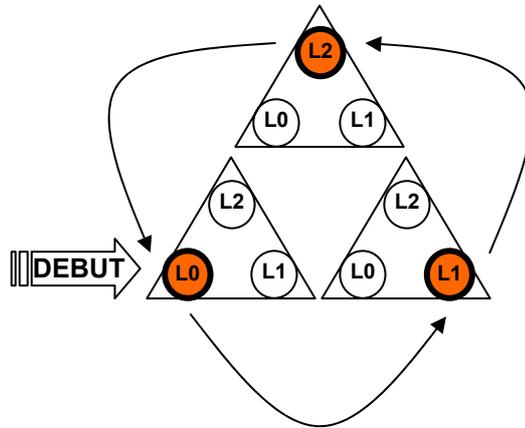


B - Pour doubler la vitesse de défilement du chenillard, il faut diviser par deux le temps d'attente introduit par les trois instructions d'origine « pause 500 ».
Les trois instructions « pause xxx » doivent avoir une valeur identique sans quoi l'effet visuel est déséquilibré. On divise par deux le temps d'attente introduit par les trois instructions « pause 500 » 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 « pause xxx » (soit 1ms), le défilement est si rapide que l'on a l'impression que les DEL sont allumées en permanence et l'effet chenillard disparaît.

Description du programme : chenillard tournant dans le sens anti-horaire.

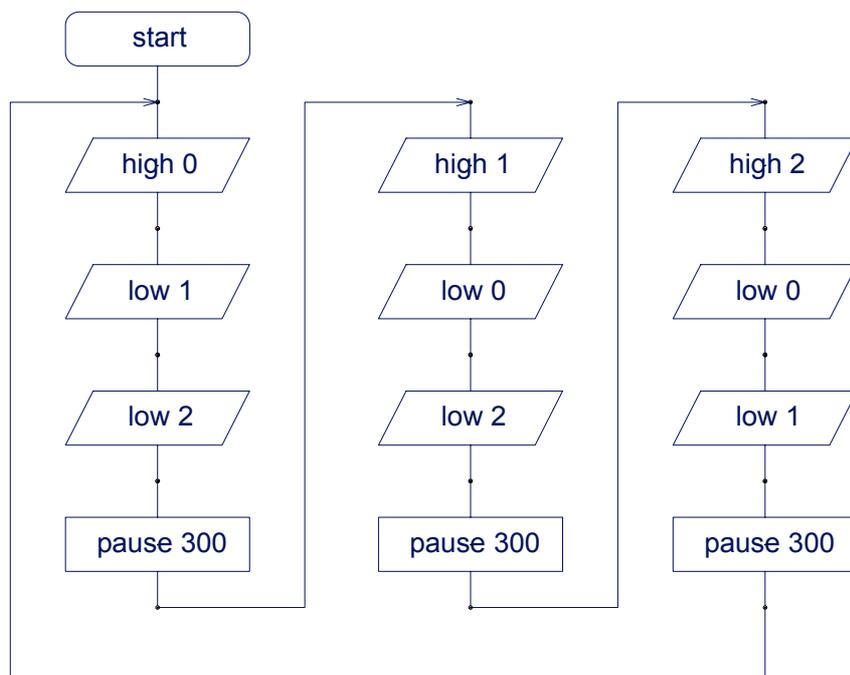
Synoptique :



Consignes :

- A** – Entourer sur le diagramme ci-dessous la (les) instruction(s) qui provoque(nt) l’allumage des DEL.
- B** - Surligner sur le diagramme ci-dessous la (les) liaisons(s) qui détermine(nt) 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.

INVERSER SENS ROTATION.cad



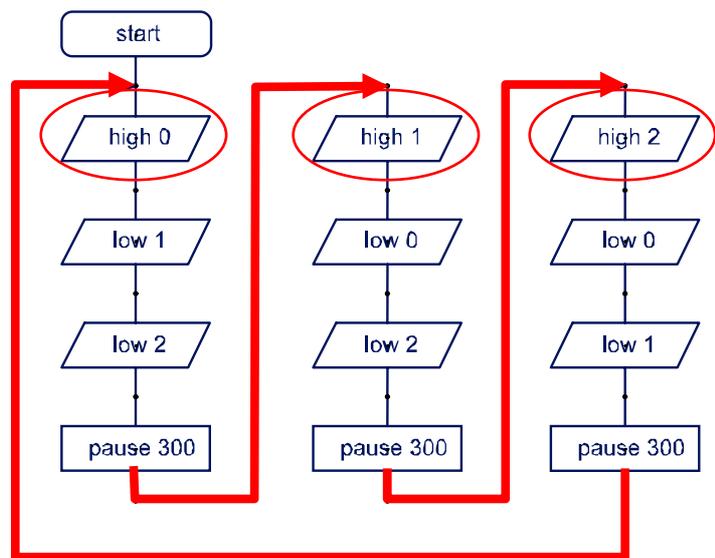
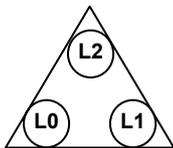
Le programme initial est un chenillard tournant dans le sens anti-horaire, c'est-à-dire dans le sens inverse des aiguilles d'une montre.

A – Se sont les instruction « high 0 », « high 1 », « high 2 » qui provoquent l'allumage respectif des DEL L 0, L1 et L2. Les instructions « low 0 », « low 1 », « low 2 » provoquent l'extinction des DEL.

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

INVERSER SENS ROTATION.cad

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

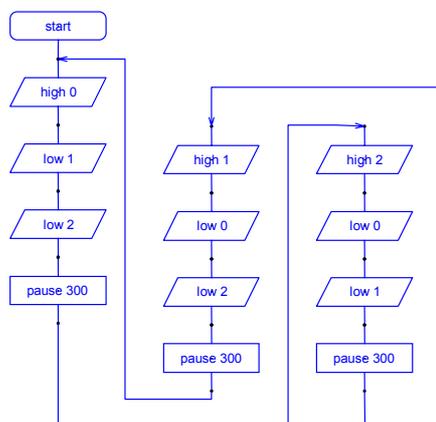


C – Pour inverser le sens de rotation (dans le sens horaire) on peut :

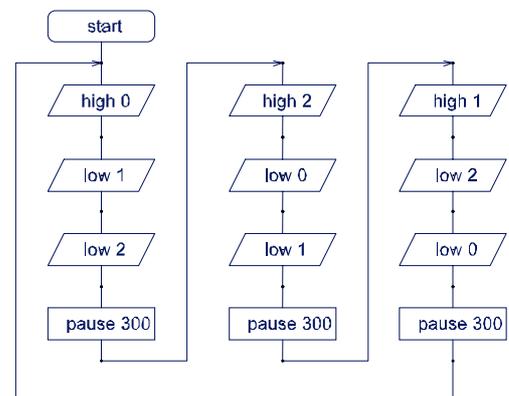
- soit modifier l'enchaînement des liaisons de telle sorte que les DEL s'allument selon la séquence L0, L2, L1 (sens horaire)

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

CORREC INVERSER SENS ROTATION (A).CAD

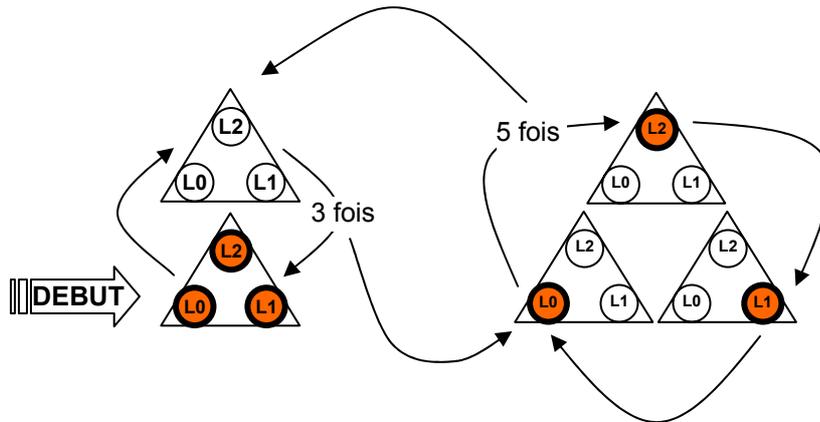


CORREC INVERSER SENS ROTATION (B).cad



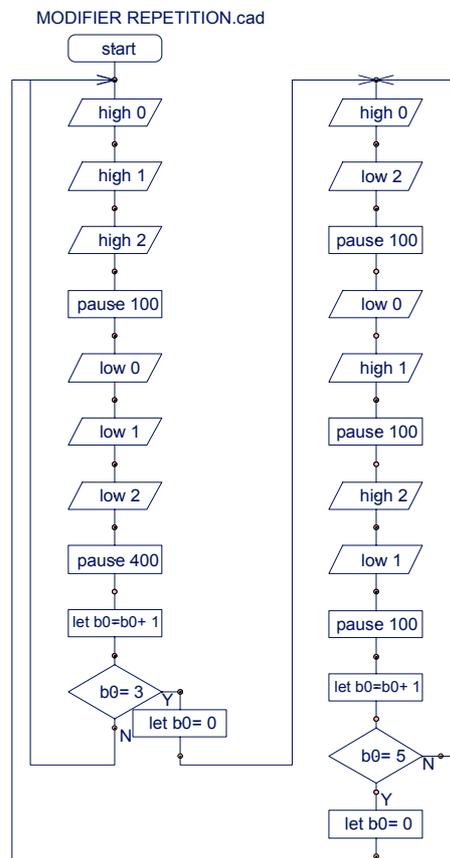
Description du programme : répétition trois fois d'une séquences clignotement suivie de la répétition cinq fois d'une séquence chenillard.

Synoptique :



Consignes :

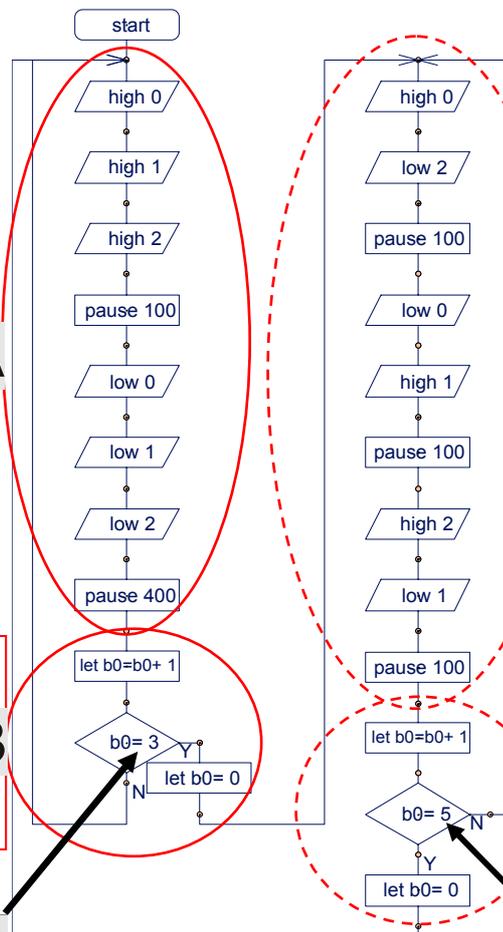
- A** - Entourer en trait plein sur le diagramme ci-dessous la séquence clignotement et en pointillé la séquence chenillard.
- B** - Entourer en trait plein sur le diagramme ci-dessous la (les) instruction(s) qui détermine(nt) le nombre de répétitions de la séquence clignotement et en pointillés dessous 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.



A / B – La première séquence exécutée fait clignoter les DEL trois fois de suite. Dès que la variable b0 qui est incrémentée de un en un atteint la valeur 3, la deuxième séquence chenillard est alors exécuté. 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 b0 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 b0 de 3 à 4 pour que la séquence clignotement soit exécutée 4 fois. On modifie la valeur de test de la variable b0 de 5 à 6 pour que la séquence clignotement soit exécutée 6 fois.

MODIFIER REPETITION.cad



A Séquence clignotement

A Séquence chenillard

B Incrément de la variable b0, test de sa valeur pour éventuellement exécuter la deuxième séquence

B Incrément de la variable b0, test de sa valeur pour éventuellement exécuter la première

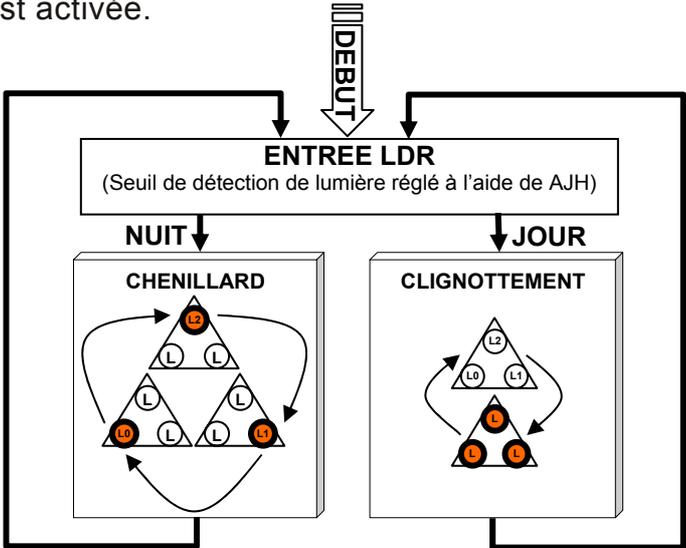
C On modifie la valeur de test de b0 à 4 pour que la première séquence soit exécutée quatre fois

C On modifie la valeur de test de b0 à 6 pour que la première séquence soit exécutée six fois

programme **INVERSER MODE JOUR-NUIT.cad**

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

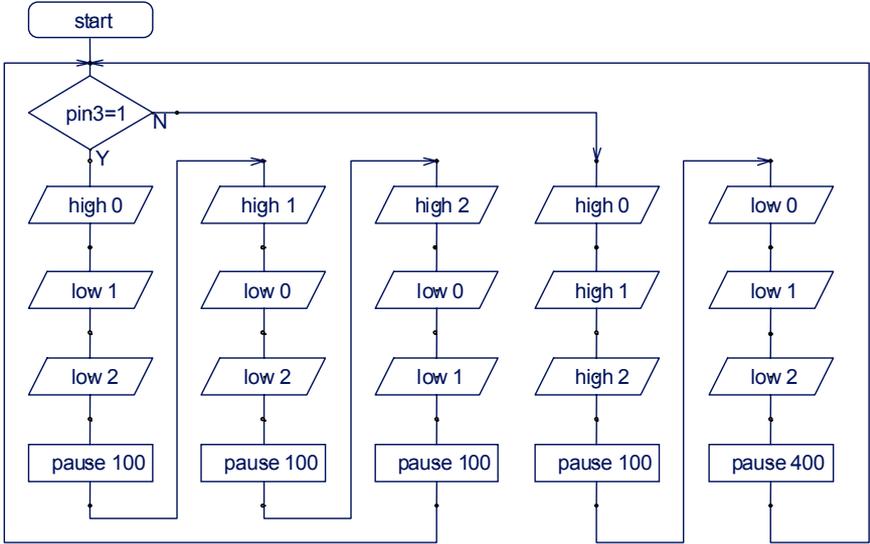
Synoptique :



Consignes :

- A** - Ouvrir le diagramme INVERSER MODE JOUR-NUIT.cad 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 DEL.
- 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 DEL.

INVERSER MODE JOUR-NUIT.cad



CORRECTION FICHE N°25 : INVERSER MODE JOUR-NUIT.cad

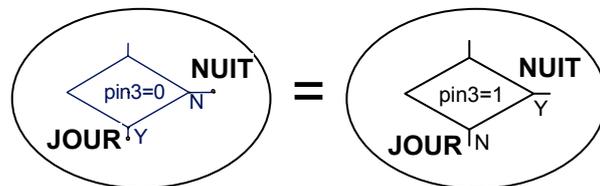
- A** – C'est le niveau de tension sur l'entrée « pin3 » qui détermine si il fait jour ou si il fait nuit.
- Si il fait jour, la résistance de la LDR est minimum et le niveau de tension sur l'entrée « pin 3 » est minimum : pin 3 = 0.
 - Si il fait nuit, la résistance de la LDR est maximum et le niveau de tension sur l'entrée « pin 3 » est maximum : pin 3 = 1.

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 DEL 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 DEL se mettent à clignoter alternativement (effet chenillard).

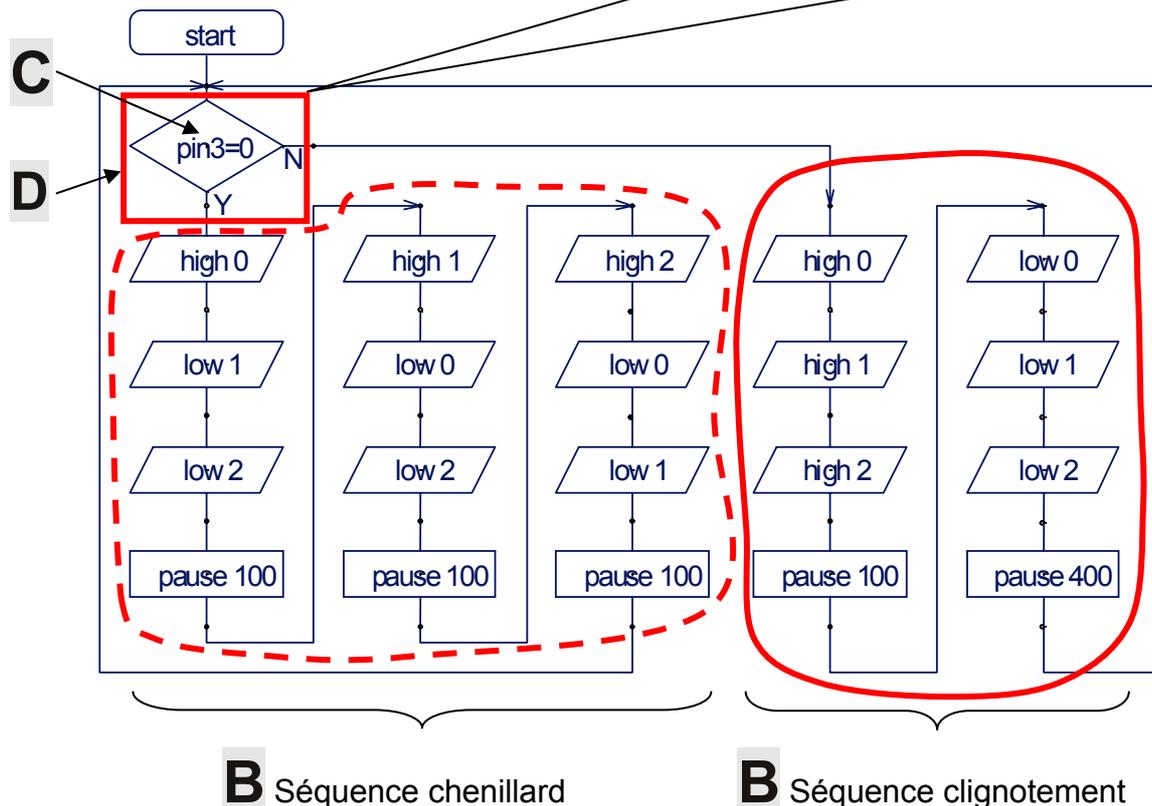
- B / C** – Si pin3=1 est vrai (« Yes ») il fait nuit ; c'est la séquence chenillard qui est exécutée ; sinon, c'est la séquence clignotement qui est exécutée.

D – Pour inverser le mode de fonctionnement du système (nuit = clignotement, jour = chenillard), on peut modifier la valeur de test de pin3 ou garder la valeur de test d'origine et remplacer le bloc de test d'origine avec un bloc dont les sorties sont organisées de manière opposée.

On peut aussi décider de garder le bloc de test d'origine et réorganiser le diagramme en connectant la séquence clignotement à la place de la séquence chenillard ; cependant cette modification entraîne des manipulations plus compliquées sur le diagramme.



CORREC INVERSER MODE JOUR-NUIT.cad



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.

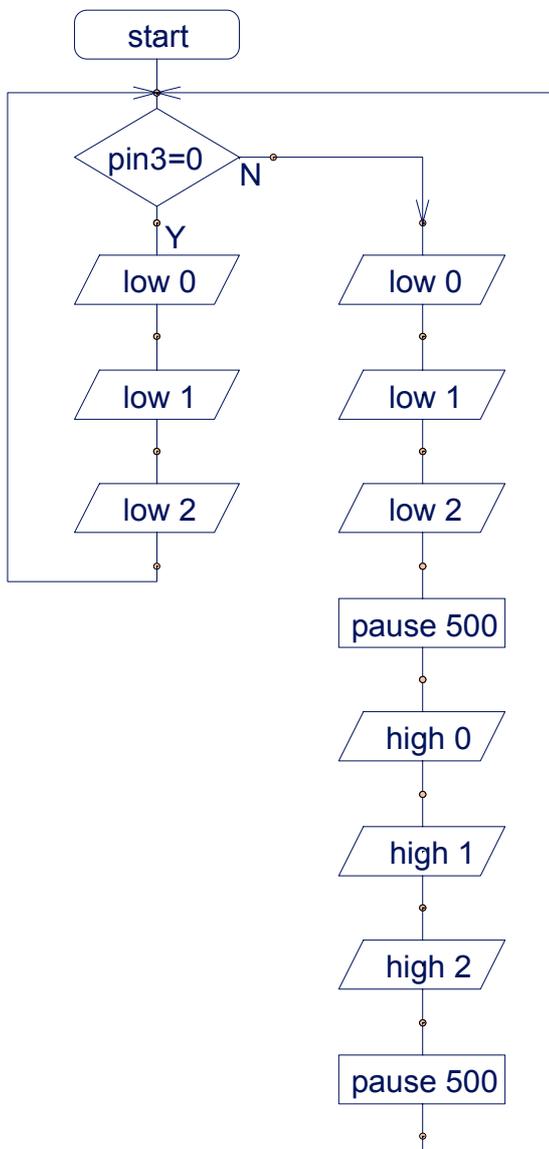
On observe sur le diagramme ci-dessous que l'instruction de test « pin3=0 » permet de déterminer l'état de l'entrée LDR :

- Si il fait jour (pin3=0 est vrai), le programme force l'extinction des trois DEL (low0, low1, low2) ; la consommation d'énergie est alors minimale.
- Si il fait nuit (pin3=0 est faux), le programme exécute alors une séquence chenillard et permet d'activer automatiquement un signal lumineux de sécurité.

Diagramme :

Consignes :

DEMAR AUTO NUIT.cad



A - Régler l'ajustable du FlashProg en position médiane.

Ouvrir le diagramme DEMAR AUTO NUIT.cad 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 DEL.

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 DEL 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.

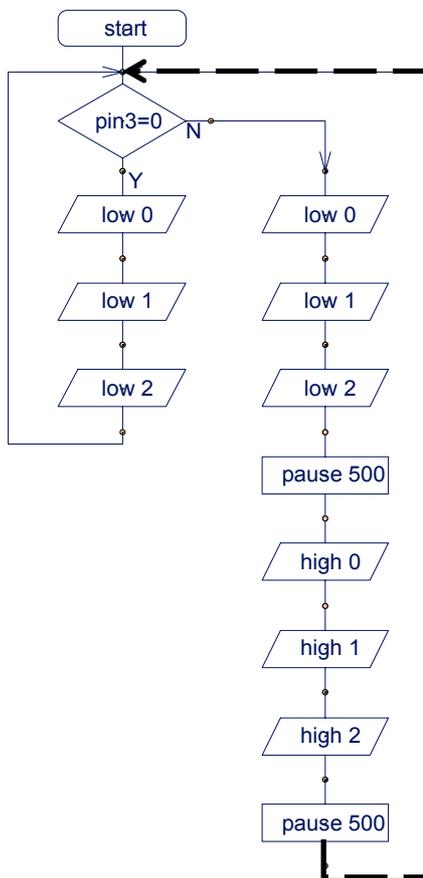
A – Après chargement du programme initial DEMAR AUTO NUIT, le FlashProg étant dans la lumière ambiante de la classe (le jour), si l’ajustable est réglé en position médiane, toute les DEL restent à priori éteintes. Si ce n’est pas le cas (on observe une séquence chenillard), il convient alors de modifier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

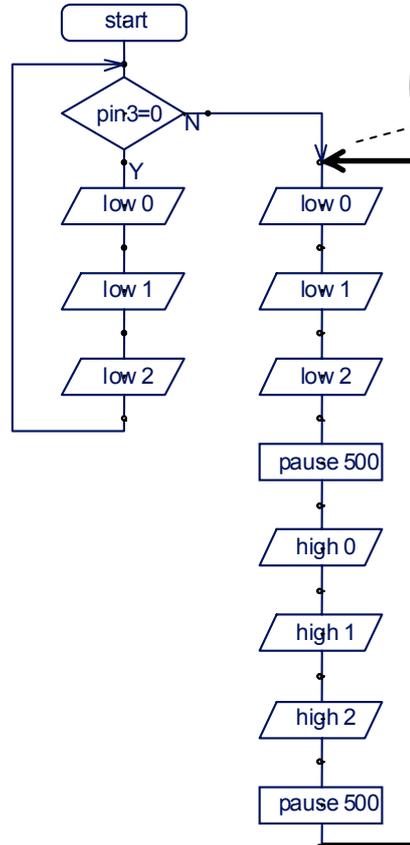
DEMAR AUTO NUIT.cad



On fait disparaître la connexion à modifier à l’aide de l’outil 

PROGRAMME AMELIORE

CORREC DEAMAR AUTO NUIT.cad



On réalise les nouvelles connexions à l’aide de l’outil 

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 CDROM 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 (« pause 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.

| Page | | Effet visuel souhaité |
|----------|-------------|---|
| 50 51 | FICHE N° 31 | Faire clignoter trois fois de suite la DEL 0, puis trois fois de suite la DEL 1, puis trois fois de suite la DEL 2 |
| 52 53 | 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 DEL sont éteintes et qui repart dans l'autre sens. |
| 54 55 | FICHE N° 33 | Allumage de L0, extinction de DEL0, allumage des trois DEL, extinction des trois DEL, allumage de DEL1 . . . |
| 56 57 | FICHE N° 34 | Train de cinq flashes rapides (clignotement simultané des DEL), ponctué par un temps d'arrêt ou toutes les DEL sont éteintes. |
| 58 59 | FICHE N° 35 | Lorsqu'il fait jour, clignotement lent de L2 (témoin de fonctionnement), sinon clignotement rapide et simultané des trois DEL. |

Travail élève :

- 1** → Lancer le logiciel Programming Editor 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 programmation (ici il s'agit des « blocs » utilisés dans les diagrammes). Une instruction a un rôle bien déterminé et peut être paramétrée. Par exemple « pause xxx » indique au microcontrôleur d'attendre pendant un temps de xxx ms ou xxx est une valeur paramétrable exprimée en millisecondes (de 1 à 65 535 ms). 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 « start ».

Une instruction du type « high 0 » qui pilote l'état de la sortie out0 et provoque l'allumage de la DEL L0 et maintient cette action tant que le programme ne rencontre pas une instruction qui annule pas cette action (en l'occurrence « low 0 »).

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 :

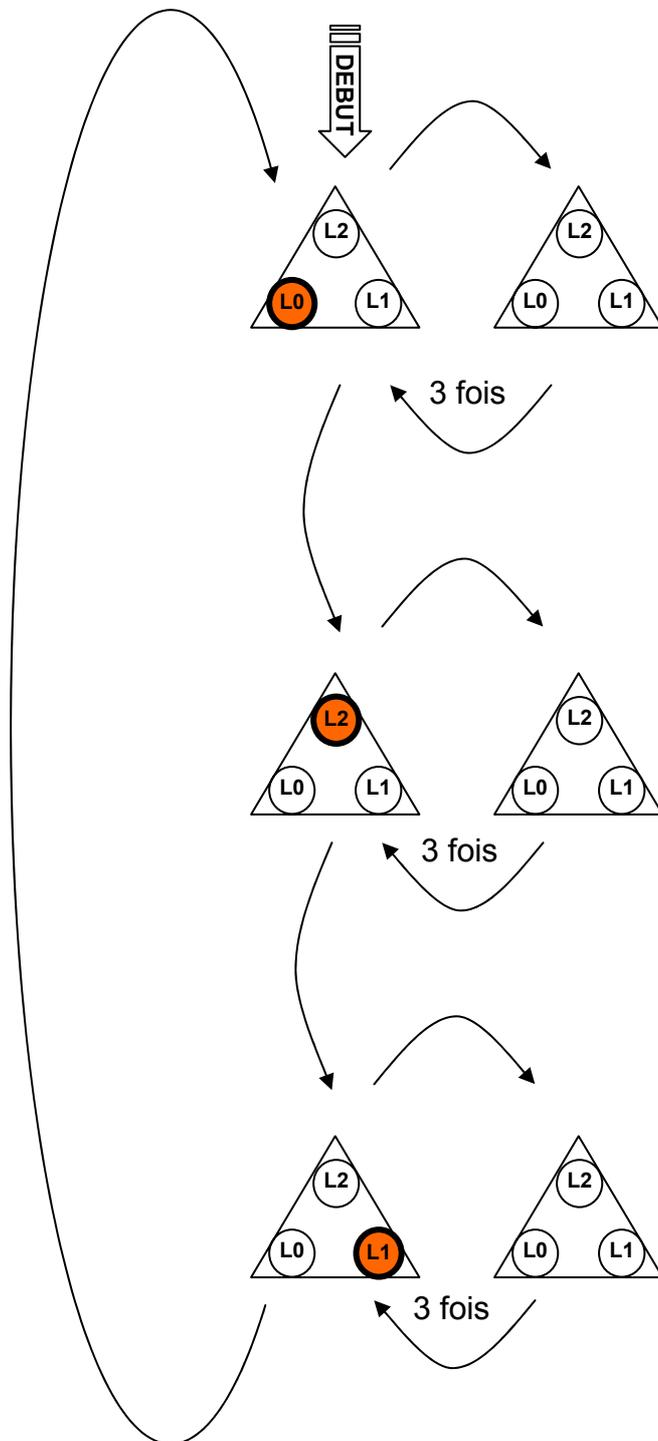
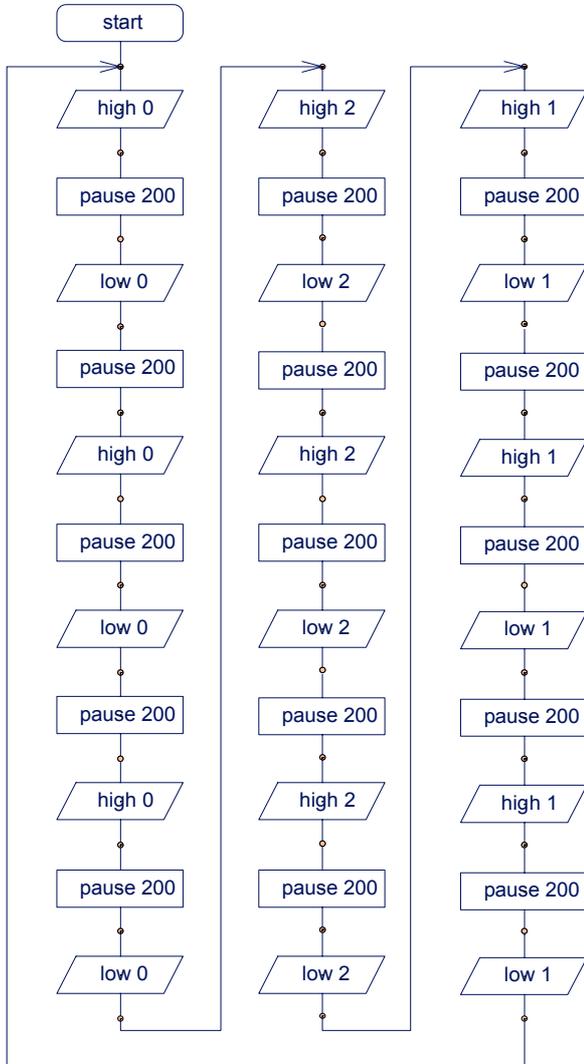


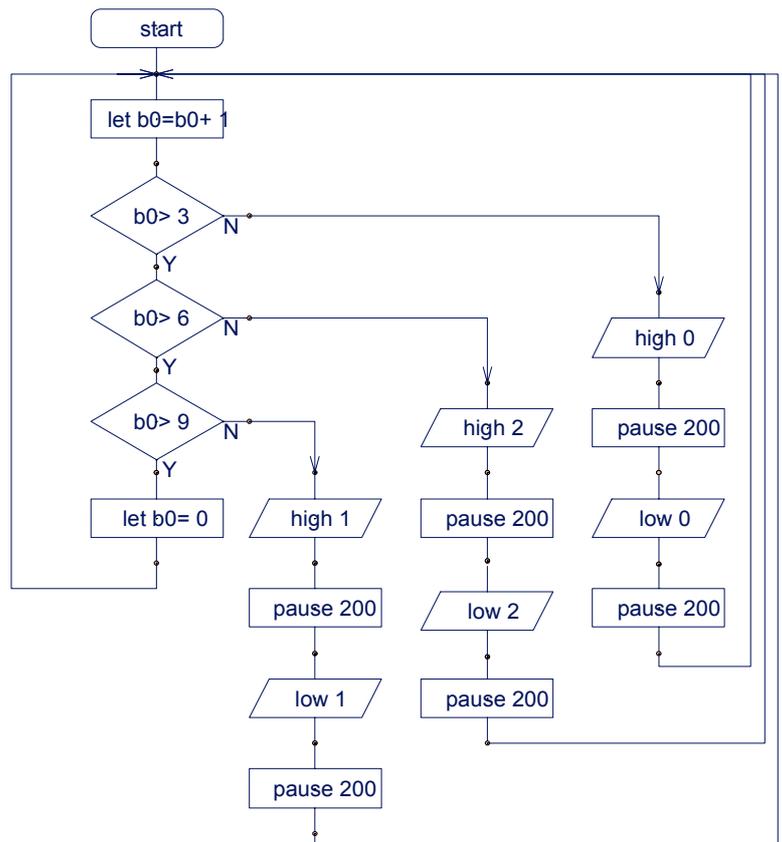
Diagramme :

F31-CORRECTION a.cad



Note : à la mise sous tension, toutes les sorties sont à zéro. Une sortie reste à zéro tant qu'une instruction « high 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 57 octets.

F31-CORRECTION b.cad



On peut simplifier le programme en utilisant une variable locale (b0).
Le programme occupe ici 49 octets.

Consigne :

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

Synoptique :

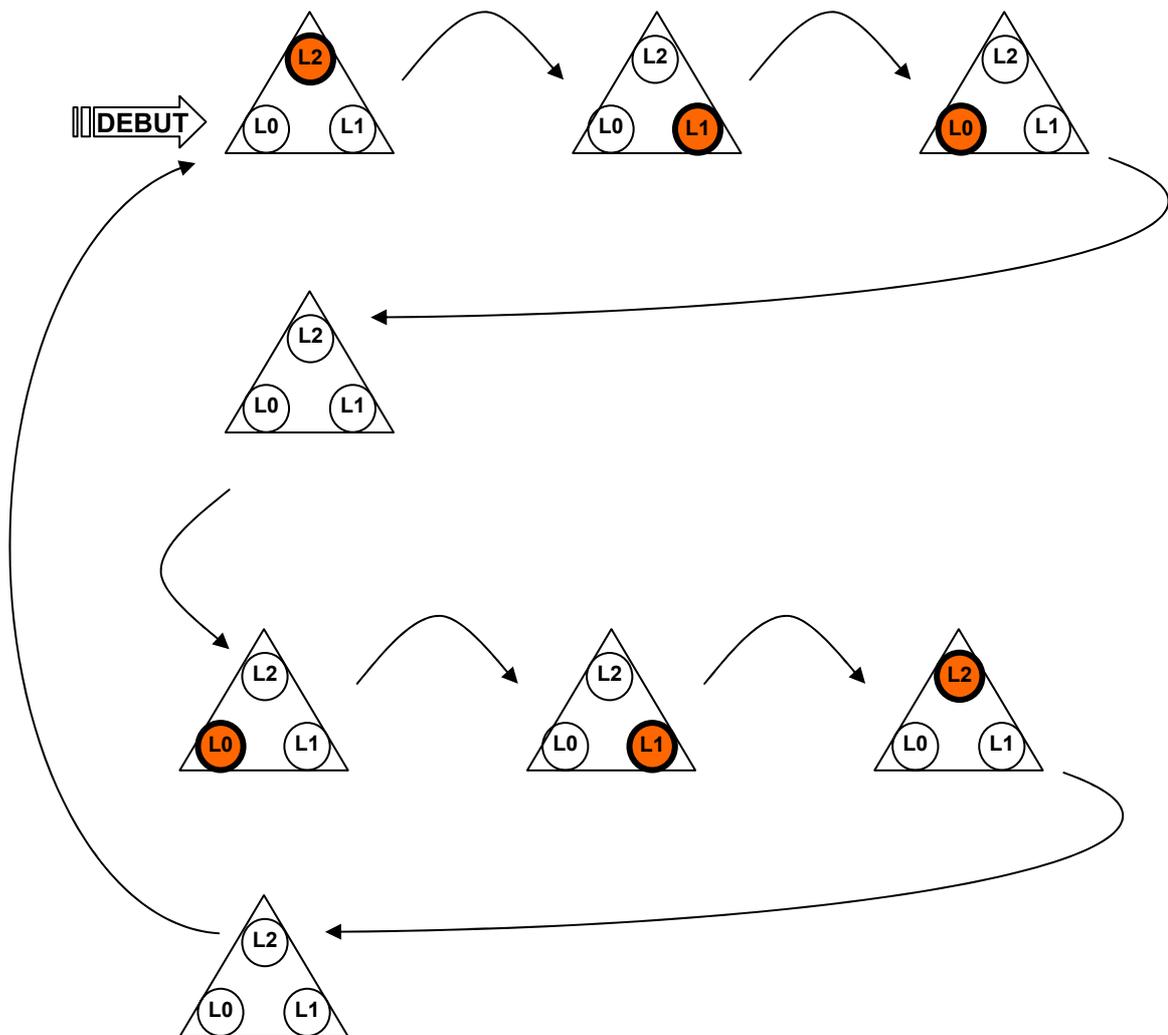
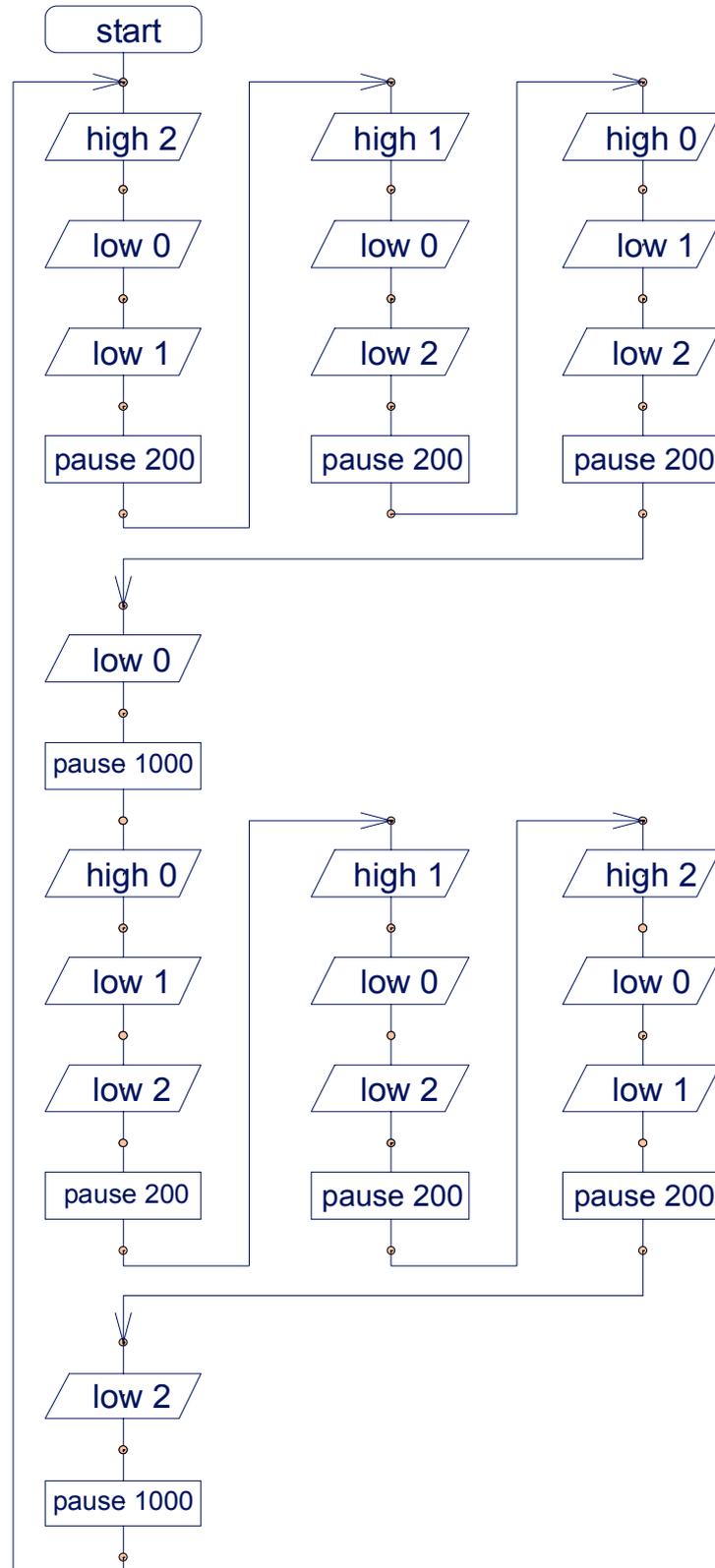


Diagramme :

F32-CORRECTION a.cad



Consigne :

Allumage de L0, extinction de DEL0, allumage des trois DEL, extinction des trois DEL, allumage de DEL1 . . .

Synoptique :

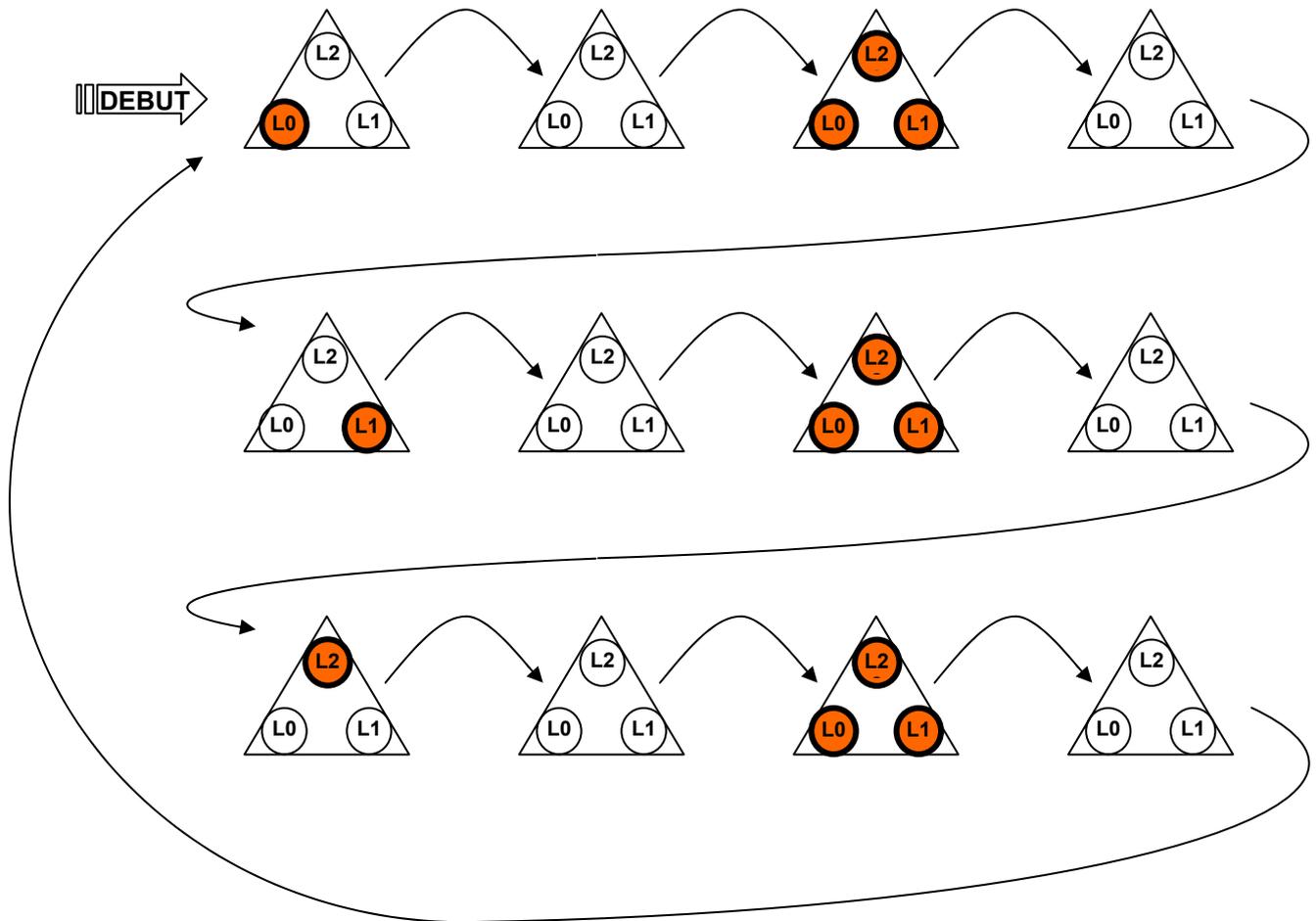
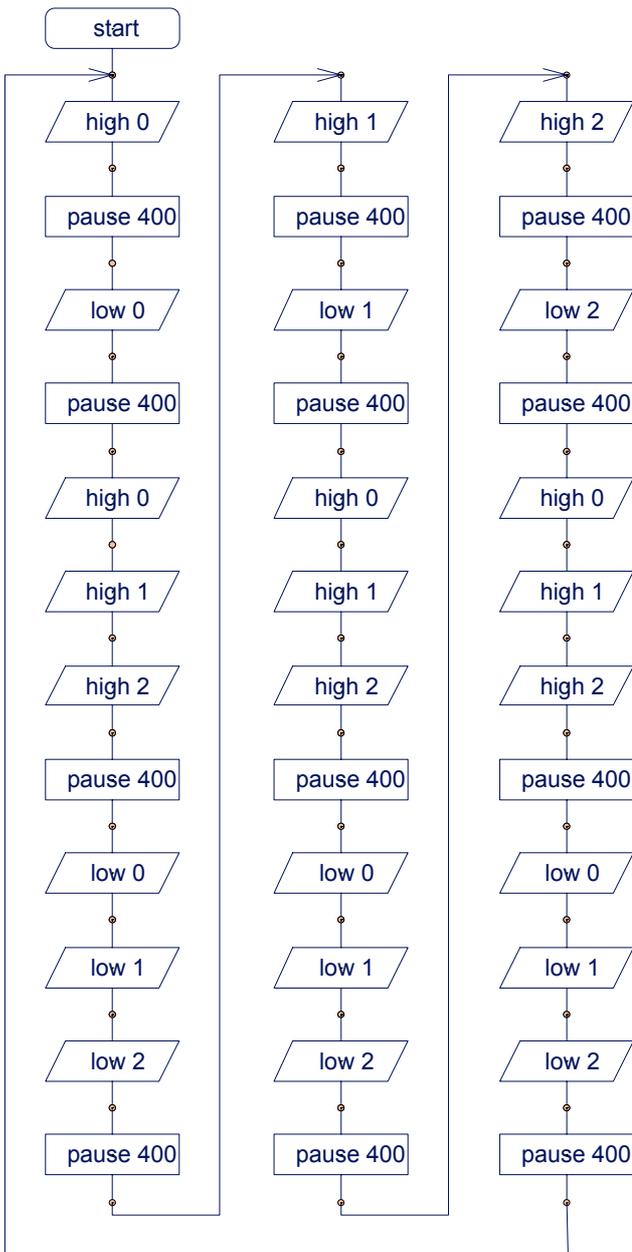
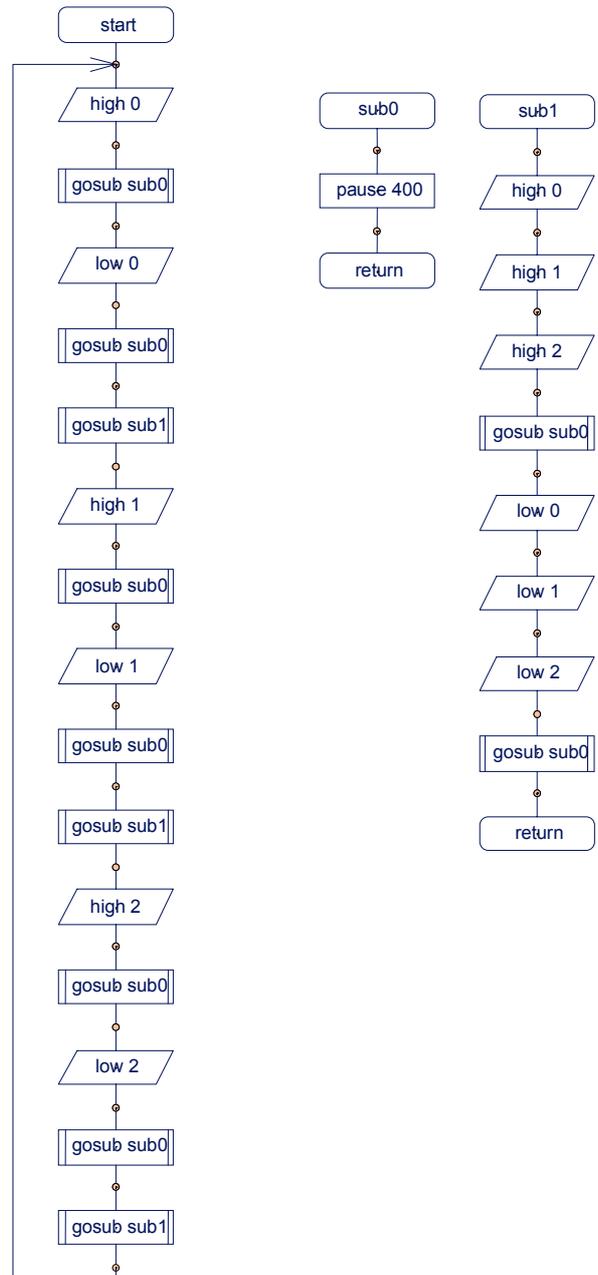


Diagramme :

F33-CORRECTION a.cad



F33-CORRECTION b.cad



1^{ère} possibilité de programmation :
 Utilisation de 37 blocs sur le diagramme.
 Consommation de 68 octets de mémoire de programme.

2^{ème} possibilité de programmation avec emploi de sous programmes :
 Utilisation de 29 blocs sur le diagramme.
 Consommation de 67 octets de mémoire de programme.

On notera qu'un sous programme peut lui-même appeler un autre sous programme.

Ici l'écriture est simplifiée mais plus élaborée. Le gain de mémoire de programme est minime (l'emploi de sous programmes est « gourmand » en mémoire).

Consigne :

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

Diagramme :

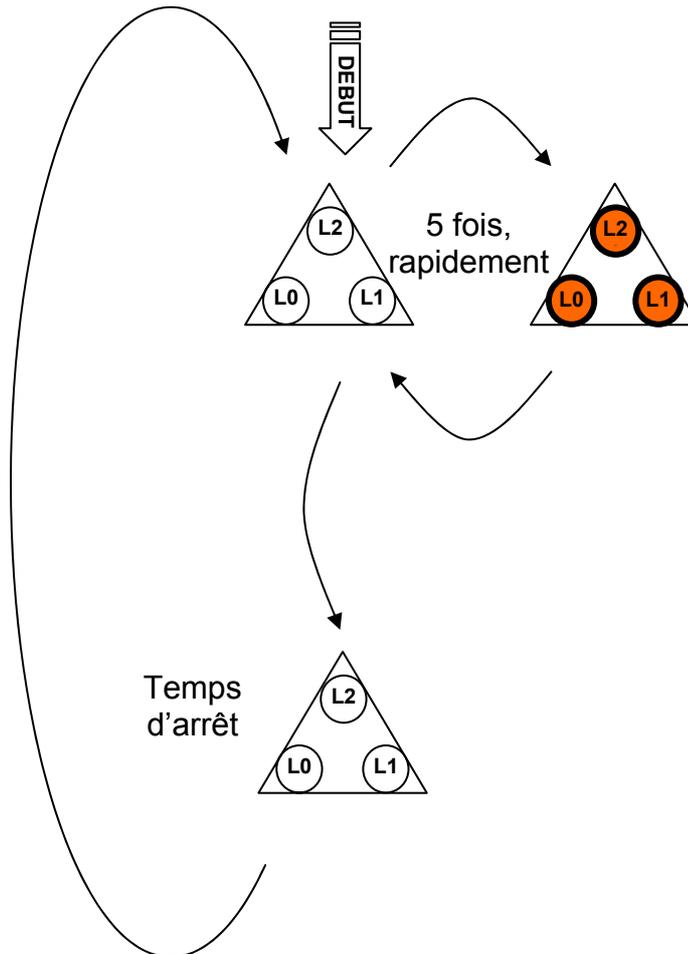
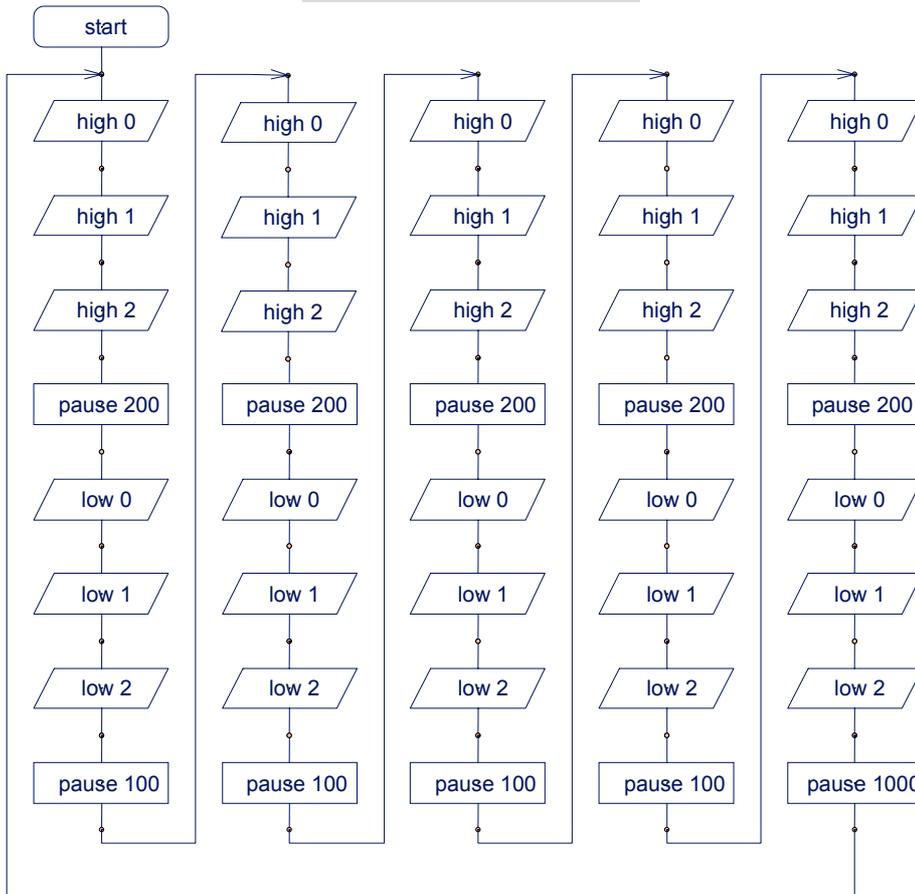


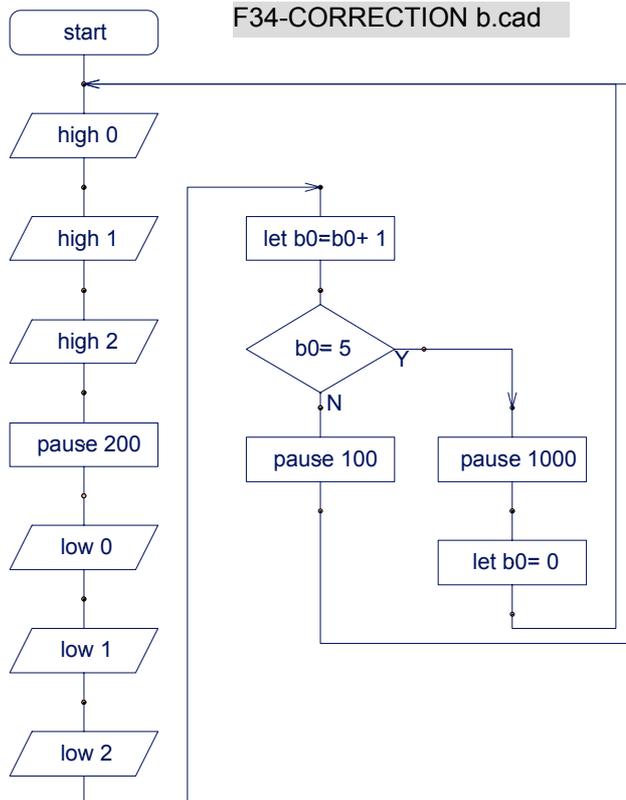
Diagramme :

F34-CORRECTION a.cad



1^{ère} possibilité de programmation :
 Utilisation de 41 blocs sur le diagramme.
 Consommation de 64 octets de mémoire de programme.

F34-CORRECTION b.cad



2^{ème} possibilité de programmation avec emploi d'une variable (compteur):
 Utilisation de 13 blocs sur le diagramme.
 Consommation de 31 octets de mémoire de programme.

Ici l'écriture est largement simplifiée mais plus élaborée. Le gain de mémoire de programme est indiscutable. 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 b0).

Consigne :

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

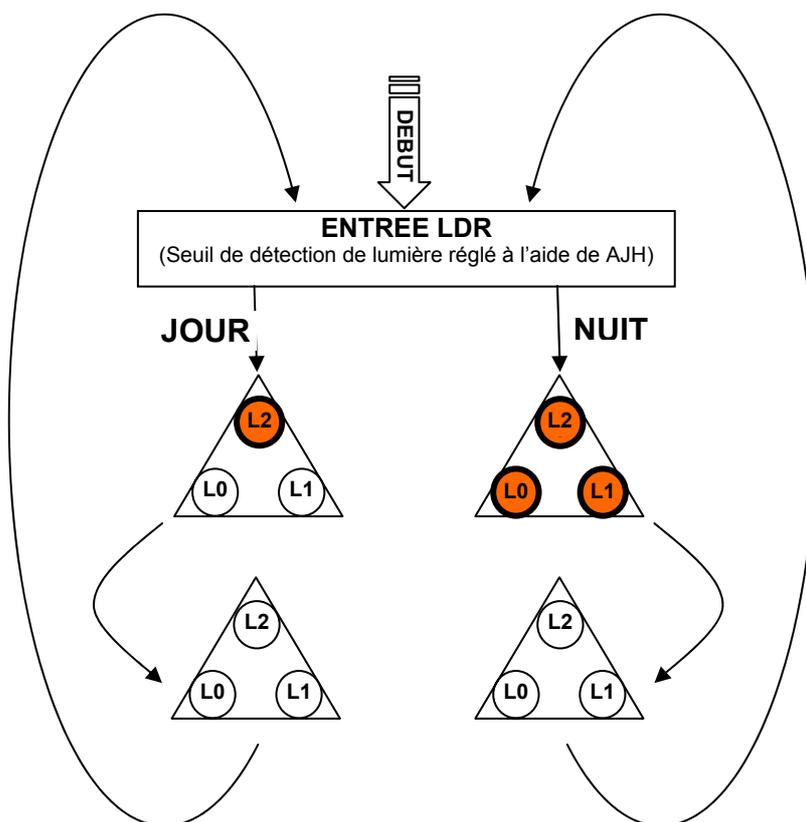
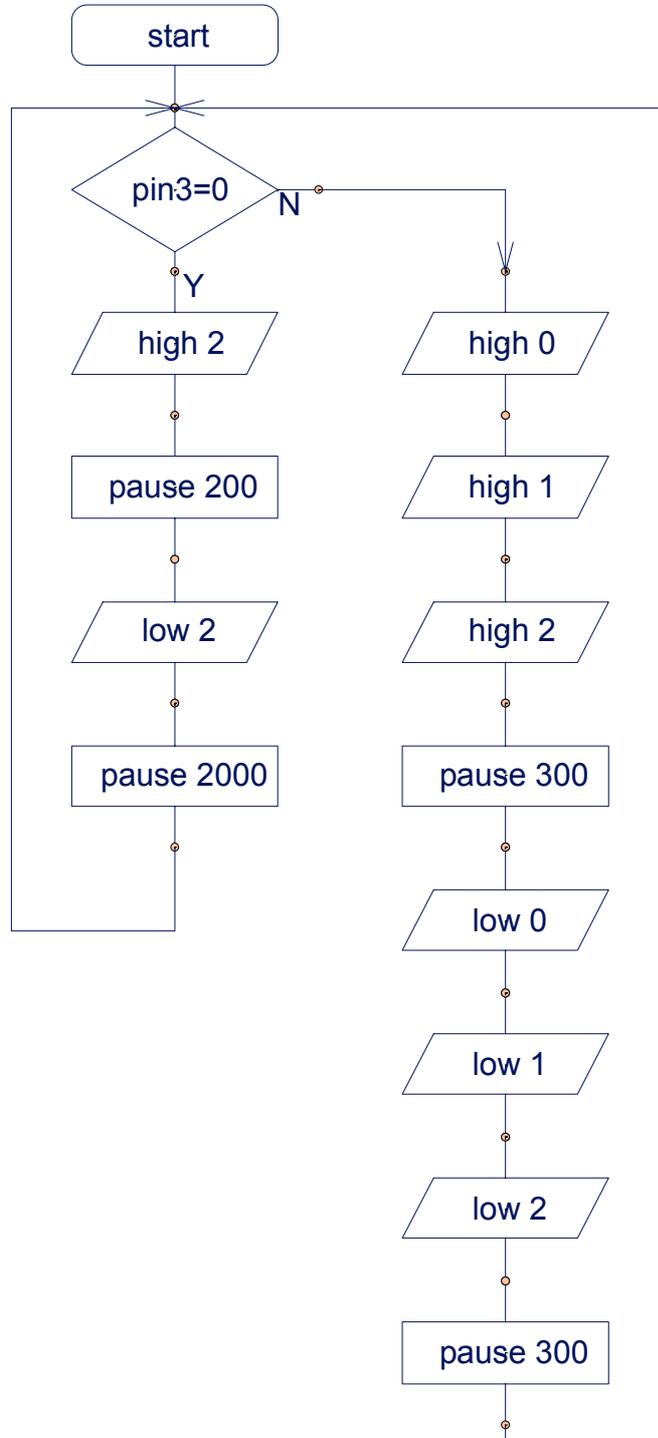


Diagramme :

F35-CORRECTION a.cad



D – ANNEXE

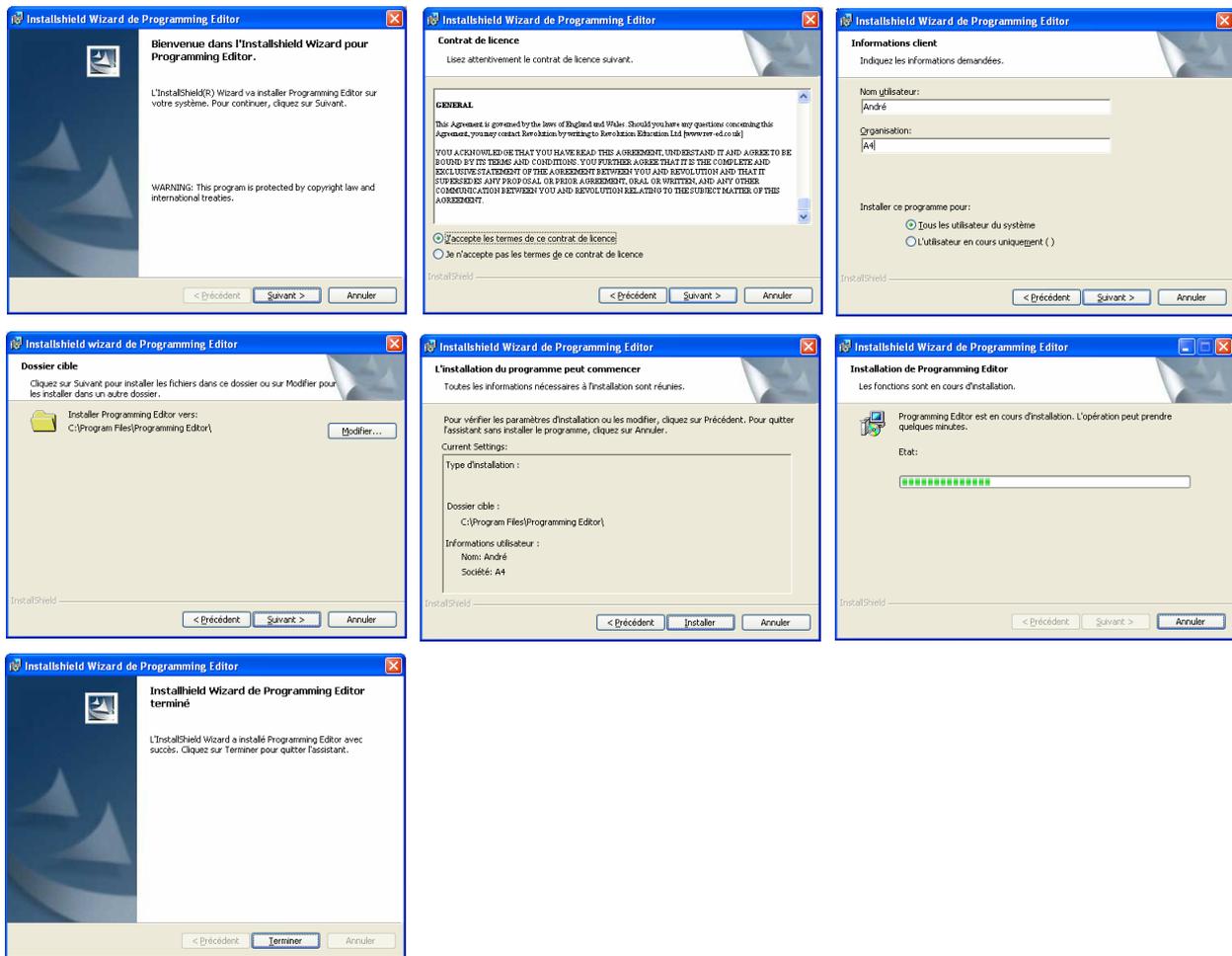
| | |
|---|----------|
| A - Installation du logiciel « Programming Editor » | 61 |
| B - 1 ^{er} lancement du logiciel | 62 |
| C - Transfert d'un programme dans le FlashProg | 63 |
| D - Description des outils de programmation | 64 et 65 |
| E - Programme de test. | 66 |
| F - Fiche de suggestions et d'inscription aux mises à jour gratuites de ce dossier | 67 |

A – Installation du logiciel Programming Editor

Configuration requise : PC installé avec Microsoft Windows 95 ® mini, 8Mo RAM mini, Port de communication 9 points.

1 Fermer toutes les applications en cours. Insérer le CDROM « Programming Editor » dans le lecteur de CDROM. L'interface d'installation démarre automatiquement (si ce n'est pas le cas, à partir du menu Démarrer, lancer la commande Exécuter, et ouvrir le fichier Setup.exe situé sur le CDROM).

2 Suivre pas à pas la procédure d'installation.



3 Vérifier que le logiciel se lance en ouvrant l'icône PICAXE Programming Editor placé sur le bureau.

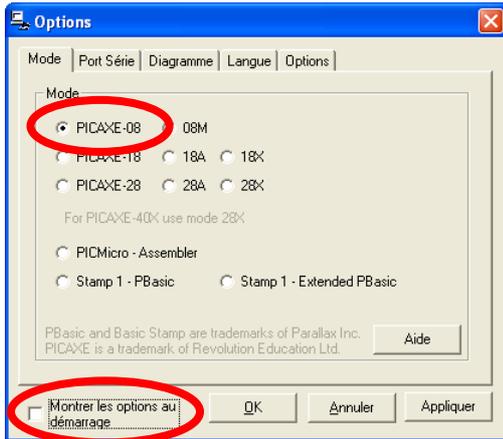


B – Premier lancement du logiciel Programming Editor

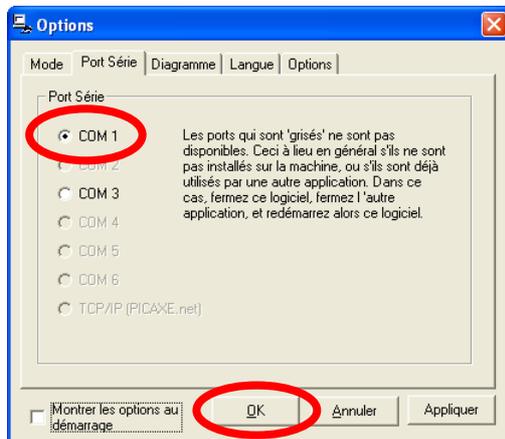
- 1 Lancer le logiciel Programming Editor en ouvrant l'icône PICAXE Programming Editor placé sur le bureau.



- 2 Sélectionner le mode « PICAXE-08 », décocher la case « Montrer les options au démarrage ».



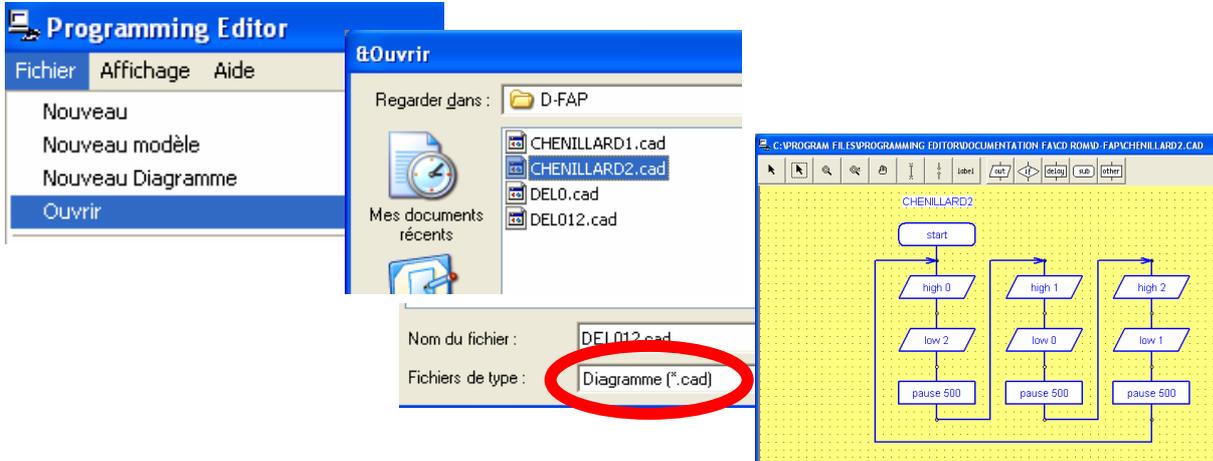
- 3 Ouvrir l'onglet « Port Série » et sélectionner le port sur lequel vous allez connecter le cordon de transfert des programmes vers le FlashProg. Fermez la fenêtre en cliquant



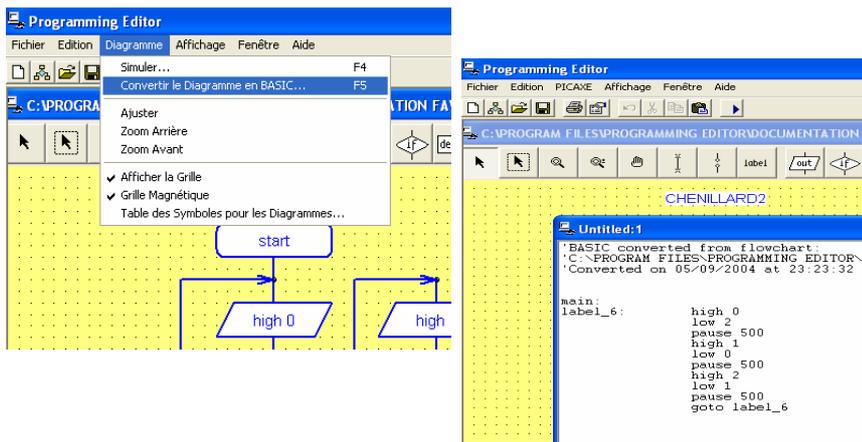
Note : Vous pouvez à tout moment modifier ces options à partir du menu Affichage – Options...

C - Transfert d'un programme de l'ordinateur dans le FlashProg

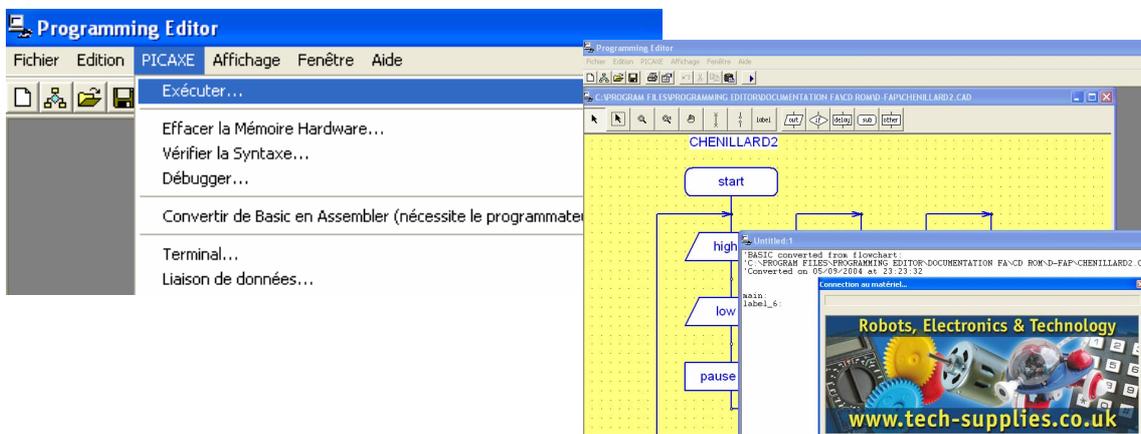
- 1 Lancer le logiciel Programming Editor et ouvrir un diagramme à partir du CDROM A4. [Attention, sélectionner les fichiers de type Diagramme (*.cad)]



- 2 Appuyer sur **F5** pour convertir le diagramme en Basic.



- 3 Connecter le FlashProg à l'ordinateur, mettre le FlashProg sous tension, appuyer sur **F5** puis sur Entrée pour transférer le programme dans le FlashProg.



D – Description des outils de programmation

| MANIPULATION DES BLOCS, LIAISONS ENTRE BLOCS | |
|---|--|
|  | Sélection d'un bloc |
|  | Sélection d'une zone -presser le bouton gauche de la souris, traîner le pointeur pour délimiter la zone, relâcher le bouton de la souris pour activer la sélection. |
|  | Agrandissement d'une zone - presser le bouton gauche de la souris, traîner le pointeur pour délimiter la zone, relâcher le bouton de la souris pour agrandir la zone. |
|  | Agrandir ou rétrécir le contenu de la fenêtre - presser le bouton gauche de la souris, traîner le pointeur du haut vers le bas pour rétrécir ou du bas vers le haut pour agrandir, relâcher le bouton de la souris. |
|  | Déplacer le contenu de la fenêtre - presser le bouton gauche de la souris, traîner le pointeur pour déplacer le contenu de la fenêtre. |
|  | Etablir une connexion entre deux blocs - placer le pointeur sur le point de départ de la liaison, presser le bouton gauche de la souris, placer le pointeur sur le point de destination, presser le bouton gauche de la souris. |
|  | Définir des points de connexion entre deux blocs (<i>Non utilisé dans ce document</i>) |
|  | Insérer un commentaire - placer le pointeur à l'endroit où l'on souhaite insérer un commentaire, presser le bouton gauche de la souris, saisir le texte. |
| PILOTAGE DES SORTIES | |
|  | Activation d'une sortie à l'état haut - placer le pointeur à l'endroit où l'on souhaite placer le bloc, presser le bouton gauche de la souris, placer le pointeur sur le point de destination, relâcher le bouton de la souris. Affecter le n° de sortie à activer dans la zone bas gauche de l'écran. |
|  | Activation d'une sortie à l'état bas - placer le pointeur à l'endroit où l'on souhaite placer le bloc, presser le bouton gauche de la souris, placer le pointeur sur le point de destination, relâcher le bouton de la souris. Affecter le n° de sortie à désactiver dans la zone en bas à gauche de l'écran. |
|  | <i>Non utilisé dans ce document</i> |
|  | <i>Non utilisé dans ce document</i> |
|  | <i>Non utilisé dans ce document</i> |
|  | <i>Non utilisé dans ce document</i> |
|  | <i>Non utilisé dans ce document</i> |
| TESTS CONDITIONNELS | |
|  | Test conditionnel sur une entrée – Sélectionner le type de bloc de test souhaité, positionner le pointeur à l'endroit souhaité, presser le bouton gauche de la souris. Affecter le n° d'entrée ou de sortie souhaitée dans la zone bas gauche de l'écran. |
|  | Test conditionnel sur une variable – Sélectionner le type de bloc de test souhaité, positionner le pointeur à l'endroit souhaité, presser le bouton gauche de la souris. Affecter le nom de la variable utilisée dans la zone en bas à gauche de l'écran. |

| | | |
|---|---|---|
|  | INTRODUIRE UN TEMPS D'ATTENTE | |
|  | Attendre entre 1 et 65535 milli secondes | - placer le pointeur à l'endroit où l'on souhaite placer le bloc, presser le bouton gauche de la souris. Affecter le temps d'attente souhaité (exprimé en millisecondes) dans la zone en bas à gauche de l'écran. |
|  | Attendre entre 1 et 65 secondes | - placer le pointeur à l'endroit où l'on souhaite placer le bloc, presser le bouton gauche de la souris. Affecter le temps d'attente souhaité (exprimé en secondes) dans la zone en bas à gauche de l'écran. |
|  | <i>Non utilisé dans ce document</i> | |
|  | <i>Non utilisé dans ce document</i> | |
|  | UTILISER UN SOUS PROGRAMME | |
|  | <i>Non utilisé dans ce document</i> | |
|  | Déclarer le point de départ d'un sous programme | - placer le pointeur à l'endroit où l'on souhaite placer le bloc de départ du sous programme, presser le bouton gauche de la souris. La fin d'un sous programme est obligatoirement marquée par un bloc « return » décrit ci-dessous. |
|  | Retourner au programme principal | - placer le pointeur à l'endroit où l'on souhaite placer le bloc de fin du sous programme, presser le bouton gauche de la souris. |
|  | Appeler un sous programme | - placer le pointeur à l'endroit où l'on souhaite placer le bloc d'appel du sous programme, presser le bouton gauche de la souris. Affecter le nom du sous programme appelé dans la zone en bas à gauche de l'écran. |
|  | AUTRES COMMANDES | |
|  | Définir la valeur prise par une variable | - placer le pointeur à l'endroit où l'on souhaite placer le bloc, presser le bouton gauche de la souris. Affecter le nom de la variable souhaitée dans la zone en bas à gauche de l'écran. |
|  | <i>Non utilisé dans ce document</i> | |
|  | <i>Non utilisé dans ce document</i> | |
|  | Génération d'une valeur pseudo aléatoire comprise entre 0 et 65535 | - placer le pointeur à l'endroit où l'on souhaite placer le bloc, presser le bouton gauche de la souris. Affecter le nom de la variable de destination du nombre aléatoire qui sera généré dans la zone en bas à gauche de l'écran. |
|  | <i>Non utilisé dans ce document</i> | |
|  | <i>Non utilisé dans ce document</i> | |

Comment copier un bloc :

- A** - Pointer avec la souris le bloc à copier.
- B** - Appuyer sur la touche Ctrl.
- C** - Presser une fois le bouton gauche de la souris (le bloc copié est alors superposé sur le bloc de départ).
- D** - Lâcher la touche Ctrl.
- E** - Presser une deuxième fois en maintenant appuyé le bouton gauche de la souris et la souris et faire glisser le bloc copié jusqu'à l'endroit souhaité et en déplaçant la souris. Relâcher le bouton.

Mise en garde : il est important de relâcher le bouton Ctrl après avoir pressé la première fois sur le bouton gauche de la souris sans quoi on risque d'effectuer des copies multiples qui perturbent l'opération de copie et génère des diagrammes au comportement incontrôlé.

E – Programme de test

Le programme de test permet de vérifier le bon fonctionnement des entrées (LDR et Bouton poussoir) et de ses sorties (DEL).

Test des sorties :

L0, L1, L2

A la mise sous tension du FlashProg on doit observer une séquence chenillard.

Test des entrées :

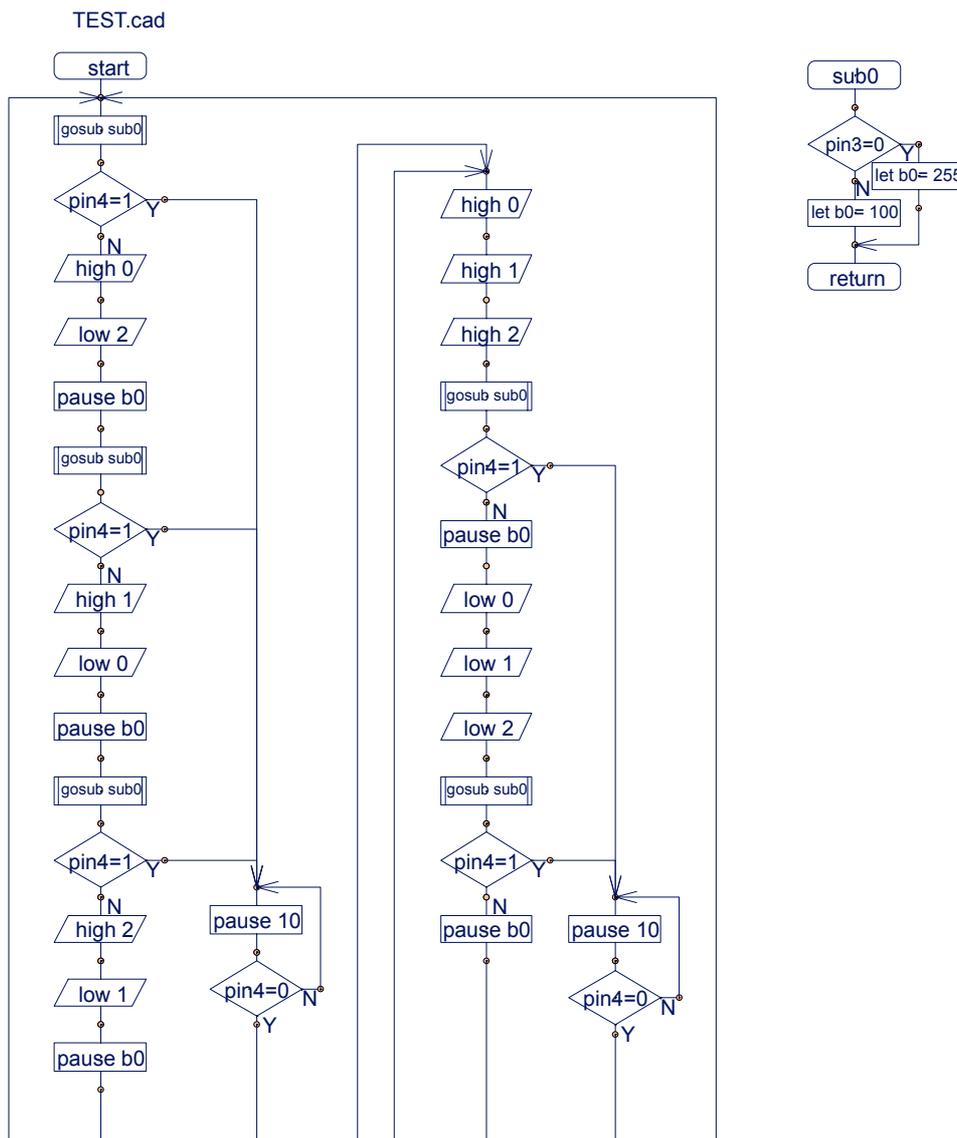
Bouton poussoir

Si l'on appuie sur le bouton poussoir, la séquence passe en mode clignotement simultané des trois DEL. Chaque appui sur le bouton poussoir provoque le basculement d'une séquence à l'autre.

LDR

L'ajustable étant préalablement réglé sur sa position médiane, la séquence en cours (chenillard ou clignotement) est exécutée à une fréquence moyenne le jour.

Si l'on occulte la LDR (obscurité), la fréquence de la séquence en cours est alors accélérée.



F – Fiche de suggestions et d'inscription aux mises à jour du dossier FlashProg

Afin de faire évoluer ce dossier nous vous invitons à nous faire part de vos remarques :

SUGGESTIONS

- j'autorise la société A4 à publier ces remarques.
 je n'autorise pas la société A4 à publier ces remarques.

Pour bénéficier gratuitement des mises à jour de ce dossier nous vous invitons à nous communiquer vos coordonnées.

Les mises à jour de ce dossier s'adressent exclusivement aux établissements ou personnes pouvant justifier de l'achat de ce dossier (N° de facture ou adresse de facturation). Les mises à jour sont communiquées exclusivement par Internet (veillez à communiquer une adresse email valide). Les mises à jour proposées ne concernent pas le logiciel Programming Editor.

INSCRIPTION MISES A JOUR DU DOSSIER FLASHPROG

Adresse email :

Nom :

Prénom :

Adresse de votre établissement :

Adresse personnelle :

(Nous vous remercions de nous communiquer le N° de facture du dossier ou bien votre adresse de facturation afin de bénéficier des mises à jour gratuites ; veillez à nous communiquer une adresse email valide)

La société A4 s'engage à ne pas utiliser votre adresse email indiquée dans cette fiche dans un autre but que de vous transmettre des mises à jour de ce dossier ou de vous communiquer des informations concernant ce produit.

**Fiche à retourner à la société A4 par télécopie au 01 64 46 31 19
ou par email à info@a4.fr**

Date :

Signature :