



# **Atelier comment programmer un Grafcet sous Arduino.**

## Table des matières

1	Introduction.....	4
2	Le Grafcet.....	4
3	Construction.....	4
3.1	Étape/Action.....	4
3.2	Transition/Réceptivité.....	4
3.3	Liaison.....	4
4	Type d'action. EX.....	5
5	Structure.....	6
5.1	Type de séquence.....	6
5.2	Cas particuliers.....	7
6	Règles d'évolution.....	8
7	Règles de syntaxe.....	9
8	Règle de la programmation du GRAFCET.....	10
8.1	Règle générale.....	10
8.2	Règle pour séquences simultanées (ET).....	10
8.3	Règle pour séquences exclusives (OU).....	11
9	Situation problématiques.....	12
9.1	Evolution fugace.....	12
9.2	Problématique de la boucle à deux étapes.....	12
10	Aller plus loin, Le GEMMA.....	15
11	Conclusion.....	15
12	Références.....	16
13	Partie pratique.....	17
13.1	Avant de commencer.....	17
13.1.1	Variables temporelles.....	17
13.1.2	Étape initiale.....	17
13.1.3	Appel du Grafcet.....	17
13.1.4	Routines des Grafcets.....	17



13.1.5	Mise en oeuvre.....	17
13.2	Premier exercice, GRAFCET à séquence unique.....	18
13.2.1	Routine GRAFCET_01_Bool.....	18
13.2.2	Routine GRAFCET_01_Int.....	19
13.2.3	Routine GRAFCET_01_If.....	19
13.3	Deuxième exercice, GRAFCET à séquences multiples exclusives.....	20
13.4	Troisième exercice, GRAFCET à séquences multiples simultanées.....	21



# 1 Introduction.

Présentation

## 2 Le Grafcet.

GRAF CET (GRAph de Contrôle des Etapes et Transitions)  
(Graph de l'ACET- Association Française pour la Cybernétique Economique et Technique)

Permet de décrire le comportement logique d'un système indépendamment de la technologie utilisé. Décrit dans la norme NF EN 60848(1)

Donne une méthode pour discuter d'un automatisme avec une personne non technique.

Emmène à penser et mettre sur papier un automatisme avant de le programmer. Cela économise du temps en développement, déverminage et modification.

## 3 Construction.

Construit de trois éléments de base :

- Étape/Action.
- Transition/Réceptivité.
- Liaison. (2)

### 3.1 Étape/Action.

Étape initiale Double Carré. Autres étapes carré unique.



L'étape caractérise le comportement invariant à un instant donné. Les actions sont associées à une étape.(3)



### 3.2 Transition/Réceptivité.

**Transition** : Les différentes évolutions entre les étapes.

**Réceptivité** : A chaque transition est affectée une réceptivité présentant les conditions d'évolution. (3)

**Nature des réceptivités** :

- Des variables booléennes.
- Des comparaisons sur valeurs numériques. (5)
- Fronts.
- Temporelle.
- Toujours vraie, avec prudence.(3)
- Etc.

### 3.3 Liaison.

Les lignes qui relient les étapes entre elles.

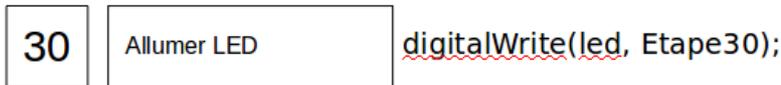
Forment l'évolution de l'automatisme d'une étape à l'autre. (3)



## 4 Type d'action. **EX**

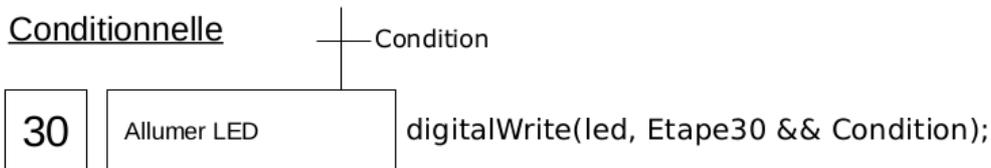
- **Continue.** S'exécute tout au long de l'étape.  
digitalWrite(led, Etape30);

### Continue



- **Conditionnelle.** S'exécute seulement si une condition associée est vraie.  
digitalWrite(led, Etape30 && **Condition**);

### Conditionnelle

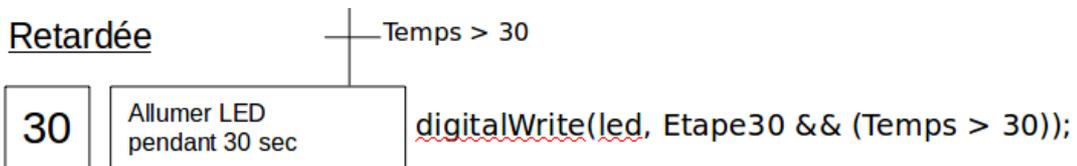


- **Action retardée ou limitée.** Action conditionnelle dont la condition associée est une variable temporelle.

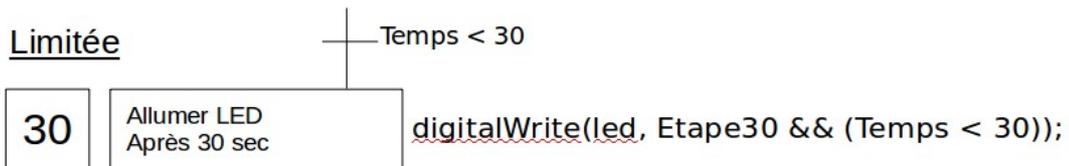
**Retardée :** digitalWrite(led, Etape30 && (**Temps > 30**));

**Limitée :** digitalWrite(led, Etape30 && (**Temps < 30**));

### Retardée

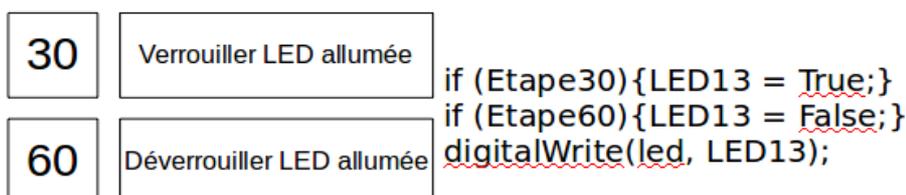


### Limitée



- **Action mémorisée.** Affecter une valeur qui doit être conservée à une sortie.  
(Set, Reset)(3)  
if (Etape30){LED13 = **True**;}  
if (Etape60){LED13 = **False**;}  
digitalWrite(led, LED13);

### Mémorisée



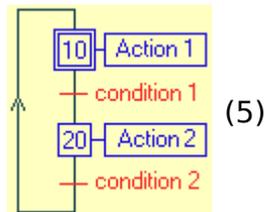
## 5 Structure.

La construction d'un Grafcet se structure selon les configurations suivantes :

### 5.1 Type de séquence.

- **Séquence unique :**

Une suite d'étapes dont le déroulement s'effectue toujours dans le même ordre.



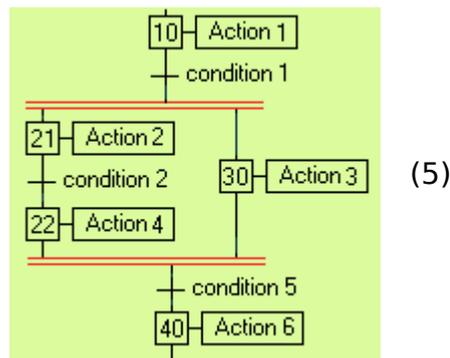
- **Séquences multiples simultanées (ET) :**

Lorsque le franchissement d'une transition conduit à activer plusieurs étapes. Débute toujours sur une réceptivité unique.

Se termine toujours sur une réceptivité unique.

Toutes les étapes finales des séquences doivent être actives pour franchir la transition finale.

Représenté par une double barre.



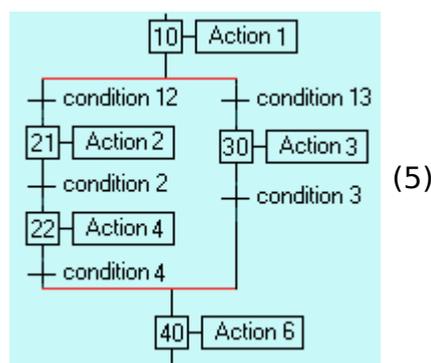
- **Séquences multiples exclusives(OU) :**

Lorsque, à partir d'une étape, on peut effectuer un choix entre plusieurs séquences possibles conditionnées par plusieurs réceptivités exclusives.

Au début, pour chaque aiguillage possible, une transition unique et exclusive.

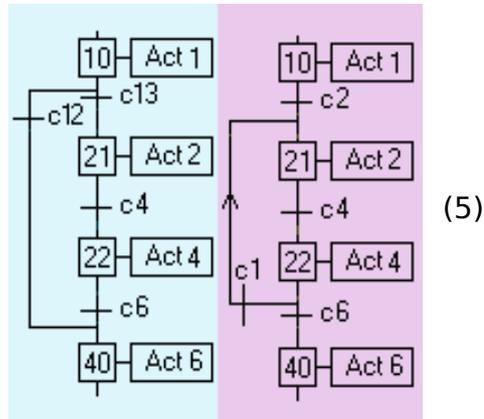
A la fin, autant de transitions que de séquences aiguillables.

Représentée par une barre simple.



## 5.2 Cas particuliers.

- **Saut d'étape :**  
Permet de sauter un certain nombre d'étapes.
- **Reprise d'étape :**  
Permet de répéter une même séquence jusqu'à ce que la réceptivité de fin de séquence soit vraie. (5)

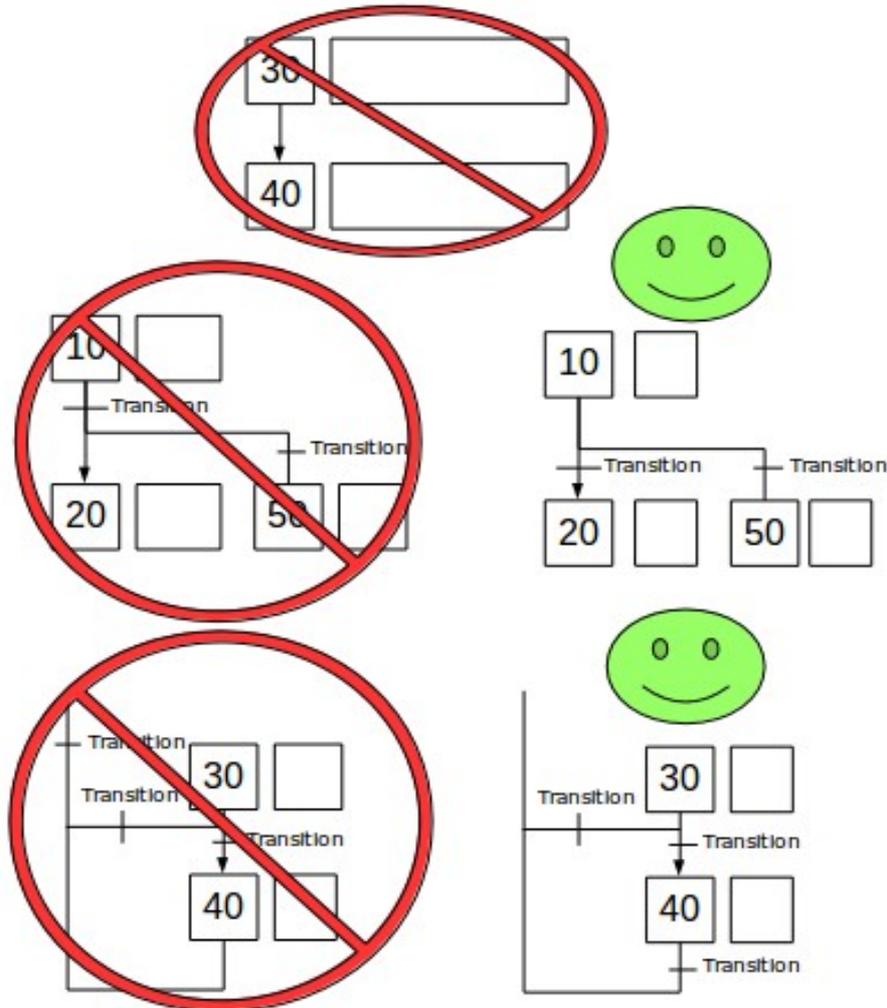


## 6 Règles d'évolution.

1. L'initialisation correspond aux activités à la mise sous tension. Par convention les étapes initiales sont représentées par un double-carée. Les étapes initiales sont activées sans autre condition.(3)
2. Une transition est soit validée soit non-validée.  
Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives.  
Une transition ne peut être franchie que lorsqu'elle est validée et que la réceptivité associée à la transition est vraie.
3. Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation des étapes immédiatement précédentes.
4. Plusieurs transitions simultanément franchissables sont simultanément franchies.
5. Si au cours du fonctionnement, une étape doit être désactivée et activée simultanément, elle reste activée.(4)

## 7 Règles de syntaxe.

1. L'alternance étape-transition doit être respectée.
2. Deux étapes ne doivent jamais être reliées directement.
3. Deux transitions ne doivent jamais être reliées directement.



Les flèches sont inutiles, elle ne participent pas à la syntaxe. Elle peuvent tout de même aider à la lecture.(5)

GRAF CET Niveau 1 : Spécification fonctionnelles. Description textuelle.

GRAF CET Niveau 2 : Spécification technologiques, opérationnelles.

GRAF CET Niveau 3 : Spécification de programmation.

Niveau 1	30	Allumer LED
Niveau 2	30	Pin 13 Haute
Niveau 3	30	<code>digitalWrite(led, HIGH);</code>

## 8 Règle de la programmation du GRAFCET.

*Jusqu'à présent si vous flottiez dans une torpeur semi-consciente, REVEILLEZ-VOUS !!! Voici l'essentiel de ce que vous devez retenir.*

La programmation d'un Grafcet obéit aux règles suivantes.

### 8.1 Règle générale.

Activation, auto-maintient, désactivation.

Activation : Toujours étape précédente et réceptivité.

EtapeX = EtapeX-1 . Réceptivité

Auto-maintient : Une fois activée, une étape se garde activée par elle même.

EtapeX = EtapeX

Désactivation : Toujours étape suivante.

EtapeX = /EtapeX+1

Exemple :

EtapeX = ((EtapeX-1 . Réceptivité) + EtapeX) . /EtapeX+1

Ce qui donne en C++ :

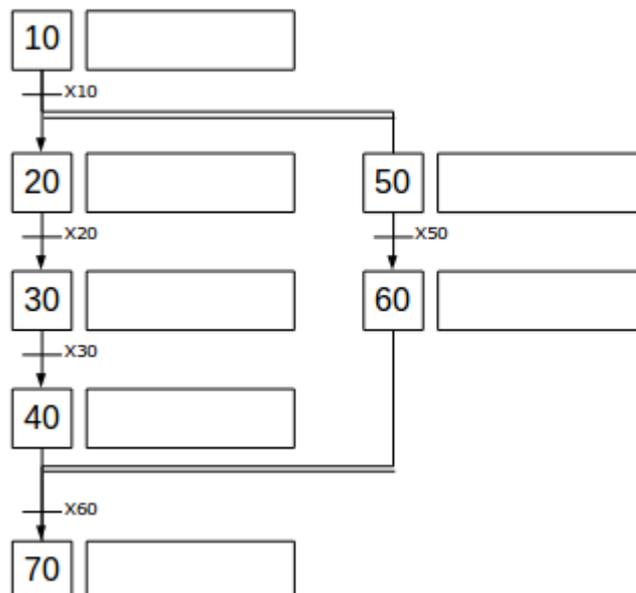
Etape2 = (((Etape1 && Receptivite) || Etape2) && !Etape3);

Cette règle s'applique à toutes les étapes d'une séquence unique.

### 8.2 Règle pour séquences simultanées (ET).

Lorsque le Grafcet contient des séquences simultanées, l'équation de l'étape précédent la divergence et celle de l'étape suivant la convergence doit être adapté.

E10 = ((E00 && X00) || E10) && !(E20 && E50);

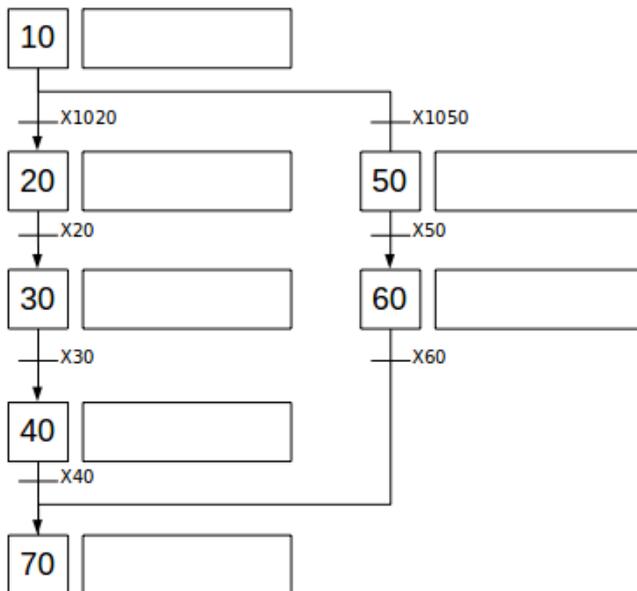


E70 = ((E40 && E60 && X60) || E70) && ! E80;

### 8.3 Règle pour séquences exclusives (OU).

Lorsque le Grafcet contient des séquences exclusives, l'équation de l'étape précédent la divergence et celle de l'étape suivant la convergence doit être adapté.

$$E10 = ((E00 \&\& X00) \parallel E10) \&\& !(E20 \parallel E50);$$



$$E70 = ((E40 \&\& X40) \parallel (E60 \&\& X60) \parallel E70) \&\& !E80;$$

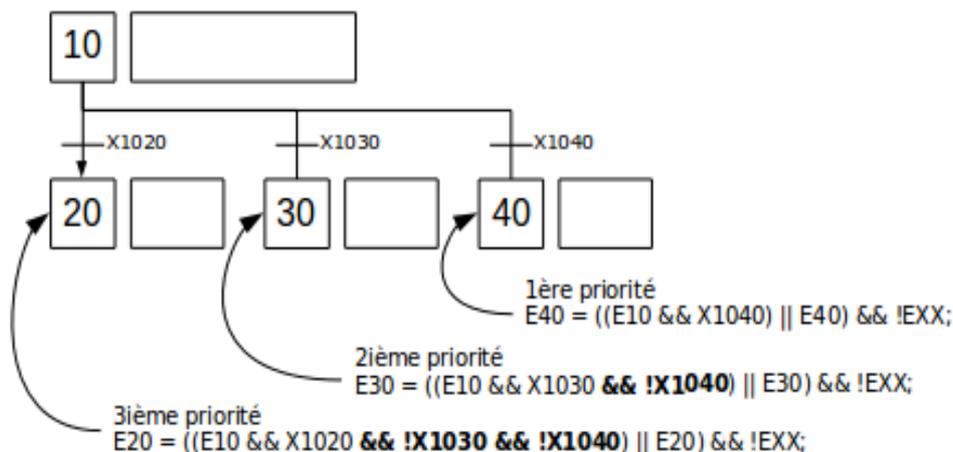
Quand vous faites face à des séquences exclusives, vous voudrez peut être donner la priorité à une séquence dans le cas où plusieurs transitions seraient vraies en même temps.

$$E20 = ((E10 \&\& X1020 \&\& !X1030 \&\& !X1040) \parallel E20) \&\& !EXX;$$

$$E30 = ((E10 \&\& X1030 \&\& !X1040) \parallel E30) \&\& !EXX;$$

$$E40 = ((E10 \&\& X1040) \parallel E40) \&\& !EXX;$$

Dans l'exemple précédent la première priorité ira à la transition E10 vers E40. Le deuxième priorité ira à la transition E10 vers E30.



## 9 Situation problématiques.

Voici la description de situations problématiques qui sont fréquemment rencontrées.

### 9.1 Evolution fugace.

On appelle **situation stable** d'un Grafcet une situation qui reste inchangée durant plusieurs cycles d'analyse d'évolution de Grafcet consécutive.

Dans le cas où plusieurs étapes sont actives pendant la durée d'un seul cycle d'analyse d'évolution le grafcet est **instable** et on qualifie donc son évolution de **fugace**. (1)

Les actions associées aux étapes instables risquent de ne pas être exécutées. Il y a lieu d'étudier le système et d'analyser les situations d'instabilité afin d'ajouter des conditions supplémentaires aux réceptivités.(3)

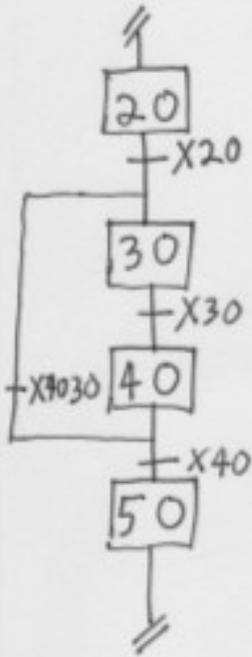
Méthodes possibles : Compteur, Temporisation.

### 9.2 Problématique de la boucle à deux étapes.

Une situation qui n'est pas abordée dans les ouvrages consultés. Pis encore, dans ces mêmes ouvrages on y montre une multitude d'exemples comportant des boucles à deux étapes.

Lorsqu'un grafcet comporte une boucle composée de deux étapes, l'équation booléenne des deux étapes est fautive en permanence dès lors que la première étape est activée. Voyons les explications suivantes.

## Problématique des deux étapes en boucle



$$E_{30} = ((E_{20} \cdot X_{20}) + (E_{40} \cdot X_{4030}) + E_{30}) \cdot / E_{40}$$

$$E_{40} = ((E_{30} \cdot X_{30}) + E_{40}) \cdot / (E_{50} + E_{30})$$

Si  $E_{30} \neq X_{30}$

$$E_{40} = ((E_{30} \cdot X_{30}) + E_{40}) \cdot / (E_{50} + E_{30})$$

$$((1 \cdot 1) + 0) \cdot / (0 + 1)$$

$$(1 + 0) \cdot / (1)$$

$$1 \cdot / 1$$

$$1 \cdot 0$$

$$0$$

Si  $E_{40} \neq X_{4030}$

$$E_{30} = ((E_{20} \cdot X_{20}) + (E_{40} \cdot X_{4030}) + E_{30}) \cdot / E_{40}$$

$$((0 \cdot 0) + (1 \cdot 1) + 0) \cdot / 1$$

$$(0 + 1 + 0) \cdot / 1$$

$$1 \cdot / 1$$

$$1 \cdot 0$$

$$0$$

## Problématique des deux étapes en boucle

\* solution possible

$$E30 = ((E20 \& \& X20) \parallel (Z40 > 3) \parallel E30) \& \& ! (Z30 > 3);$$

$$E40 = ((Z30 > 3) \parallel E40) \& \& ! ((Z40 > 3) \parallel E50);$$

if (E20) {

  Z30 = 0;

  Z40 = 0;

}

if (E30 & \& X30 & \& (Z30 < 5)) {

  Z30 ++;

  Z40 = 0;

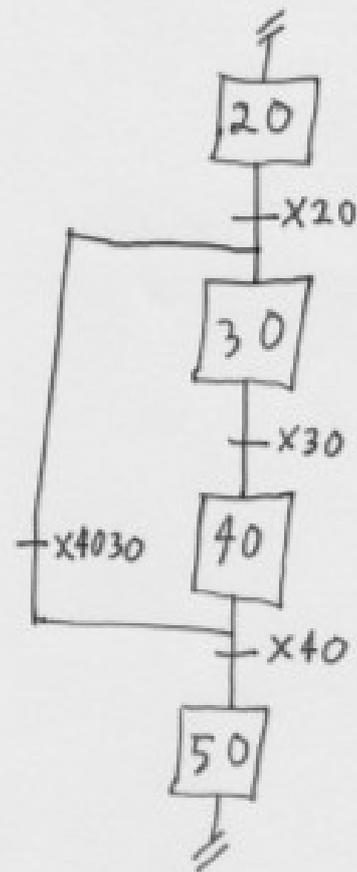
}

if (E40 & \& X40 & \& (Z40 < 5)) {

  Z40 ++;

  Z30 = 0;

}



## 10 Aller plus loin, Le GEMMA.

Le Grafcet en soit ne suffit pas à construire un automatisme au complet.

Le Grafcet permet de gérer le fonctionnement normal de l'automatisme. (3 )

Pour avoir une vision globale d'un automatisme, il faut appliquer le GEMMA.

Guide d'Étude des Modes de Marche et d'Arrêt.

Le GEMMA donne une méthode pour gérer les différentes situations et modes de marche possibles. C'est avec cette méthode qu'on traite les fonctionnements anormaux (Arrêt d'urgence, arrêt de production, marche de test, Marche préparatoire, etc). (3)

La partie que nous avons faite représente les modes A1 Arrêt dans l'état initiale et F1 Production Normale.

La loi suprême du GEMMA est l'unicité des modes.(6)

## 11 Conclusion.

L'utilisation du Grafcet permet de créer des séquences d'automatisme grâce à une méthode normalisée. Cette méthode permet de décrire l'automatisme avant même d'avoir fait le choix technologique. Toutefois, cette méthode n'est pas une méthode absolue. Elle se prête très mal au contrôle analogique. Dans des cas très simples il peut être plus rapide d'utiliser des équations logiques. Toutefois, le Grafcet saura vous simplifier la vie particulièrement lors de la création de contrôles séquentiels.

## 12 Références.

1. Sciences industrielles pour l'ingénieur tout-en-un 2e année MP, PSI, PT ...  
Par Jean-Dominique Mosser, Jacques Tanoh, Pascal Leclercq  
Dunod, Paris 2010  
ISBN 978-2-10-054636-7
2. Informatique et automation : automatismes programmables contrôlés par ordinateur  
Par Jean Mbihi  
Publibook 2006  
ISBN 2748311361
3. Le Grand Guide des systèmes de contrôle-commande industriels  
Cédric Sindjui  
Lexitis Edition 2014  
ISBN 978-2-36233-135-0
4. Commande des systèmes dynamiques : Introduction à la modélisation et au contrôle des systèmes automatiques.  
Arnaud Hubert  
Presses universitaires de Franche-Comté 2008  
ISBN 2848672358
5. <http://fr.wikipedia.org/wiki/Grafcet>
6. Le Gemma  
S. Moreno, E. Peulot  
Casteilla 1997  
ISBN 2 7135 1752 4

J'ai cité au mieux les références prises dans ces ouvrages. Il reste possible que certains passages se réfèrent à ces ouvrages sans qu'ils ne soient explicitement identifiés comme tel.

Autre référence (J'ai consulté mais je ne sais plus pourquoi au juste)

Automatique industrielle en 20 fiches

Gérard Boujat, Patrick Anaya

Dunod 2013

EAN13 : 9782100502042



## 13 Partie pratique

Vous trouverez dans cette section les informations relatives à la partie pratique de l'atelier.

### 13.1 Avant de commencer

Avant de commencer la partie pratique de l'atelier, voici quelques informations utiles.

#### 13.1.1 Variables temporelles

Les variables suivantes peuvent être utilisées pour faire clignoter la LED sur la sortie #13 de la carte Arduino à différentes fréquences selon l'étape active.

lent : variable qui bagotte à 1Hz.

normal : variable qui bagotte à 2Hz.

rapide : variable qui bagotte à 4Hz.

#### 13.1.2 Étape initiale

Activation de l'étape initiale du GRAFCET au démarrage de l'Arduino.

```
int GRAFCET_Etape = 0;  
boolean GRAFCET_Bool[] = {true, false, false,.....};
```

#### 13.1.3 Appel du Grafcet

Pour appeler la routine contenant le Grafcet désiré, il faut enlever les commentaires devant son appel et mettre les commentaires devant les appels des autres Grafcets.

```
// GRAFCET_Test(Serie_Lu, &Etape_Active);           Ce Grafcet n'est pas exécuté  
GRAFCET_01_Bool(Serie_Lu, &Etape_Active, GRAFCET_Bool);   Ce Grafcet est exécuté  
// GRAFCET_01_Int(Serie_Lu, &Etape_Active, &GRAFCET_Etape); Ce Grafcet n'est pas exécuté
```

#### 13.1.4 Routines des Grafcets

Voici le nom des différentes routines de Grafcet avec leur description.

GRAFCET_Test	Routine pour tester le lien série
GRAFCET_01_Bool	Grafcet à séquence unique programmé en équation booléenne.
GRAFCET_01_Int	Grafcet à séquence unique programmé avec switch case.
GRAFCET_01_If	Grafcet à séquence unique programmé avec if then.
GRAFCET_OU	Grafcet à séquences exclusives.
GRAFCET_ET	Grafcet à séquences simultanées.

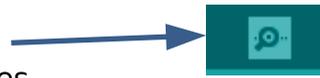
#### 13.1.5 Mise en oeuvre

Téléverser le programme dans l'Arduino.

Ouvrir le moniteur série en appuyant sur l'icône.

Le moniteur retourne le numéro des étapes actives.

Écrire une valeur de 1 à 9 dans le fenêtre puis cliquez sur envoyer pour transmettre une réceptivité au Grafcet.



Entrer ici le nombre de 1 à 9 à transmettre à l'Arduino.

# d'étapes retournées par l'Arduino



## 13.2 Premier exercice, GRAFCET à séquence unique.

La première manipulation se fera avec un Grafcet à séquence unique. Trois méthodes de programmation seront utilisées.

- L'équation logique.
- Le if.
- Le switch case.

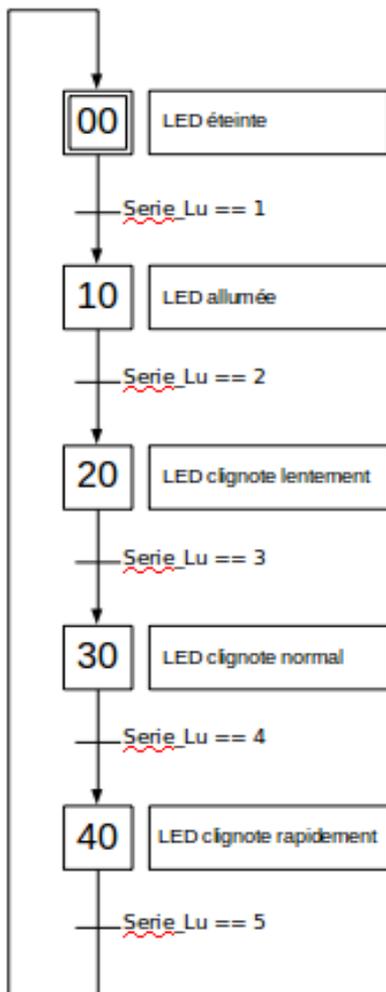
Les deux dernières méthodes sont particulièrement bien adaptées pour un Grafcet à séquence unique. Elles sont plus difficiles à mettre en œuvre avec une séquence multiple, particulièrement avec la méthode switch case à cause de l'utilisation de la commande break. Elles permettent aussi des actions sur transition. Cela ne fait pas parti des règles du GRAFCET mais est une possibilité intéressante.

### 13.2.1 Routine GRAFCET\_01\_Bool

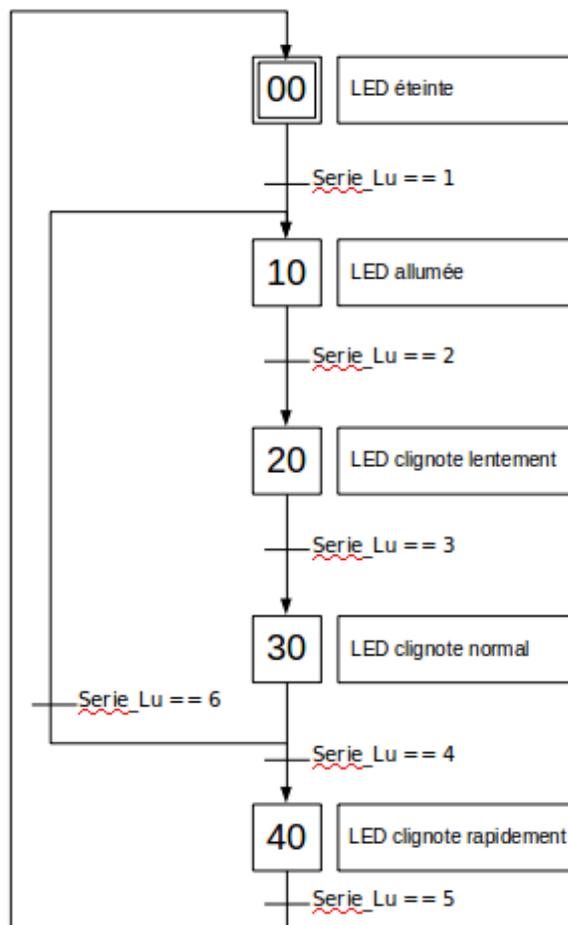
Dans cette routine les étapes sont activées et désactivées par des équations logiques tel que vu dans la partie théorique. Les étapes sont mémorisées dans un tableau de booléens.

Cette routine vous est donnée complètement programmée. Chargez la et faite évoluer le Grafcet. Observez la LED 13.

Exercice : programmez un rebouclage de l'étape 30 vers l'étape 10 comme dans le Grafcet ci-dessous.



Grafcet à séquence unique



Grafcet à séquence unique avec rebouclage

### 13.2.2 Routine GRAFCET\_01\_Int

Dans cette routine, les étapes sont activées et désactivées par une structure switch case. Les étapes sont mémorisées dans un entier.

Complétez la programmation incluant les actions dans les étapes.

Exercice : programmez un rebouclage de l'étape 30 vers l'étape 10.

### 13.2.3 Routine GRAFCET\_01\_If

Dans cette routine, les étapes sont activées et désactivées par une suite de if(){ }. Les étapes sont mémorisées dans un entier.

Complétez la programmation incluant les actions dans les étapes.

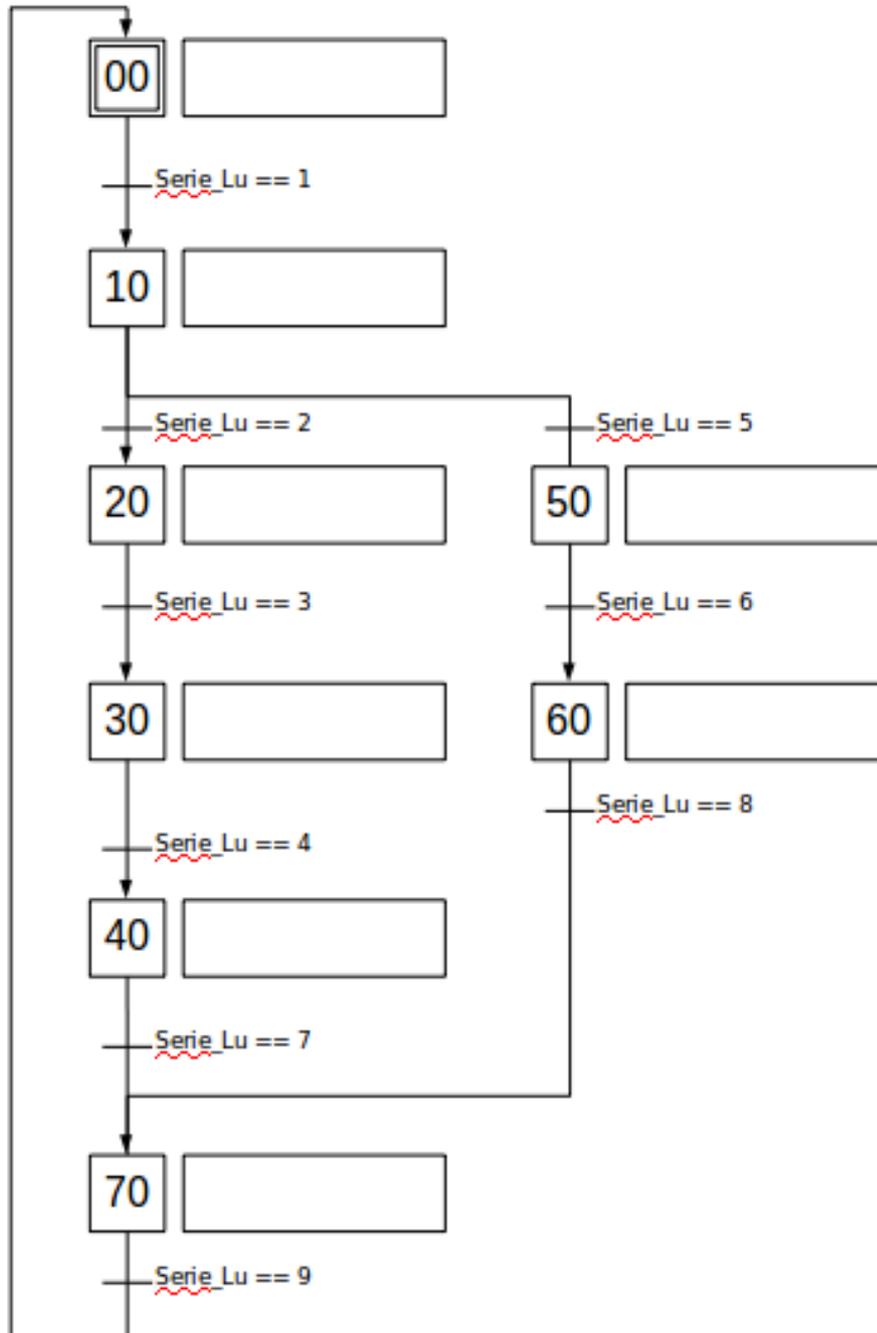
Exercice : programmez un rebouclage de l'étape 30 vers l'étape 10.

### 13.3 Deuxième exercice, GRAFCET à séquences multiples exclusives.

La routine à utiliser est GRAFCET\_OU.

Dans cette routine les étapes sont activées et désactivées par des équations logiques tel que vu dans la partie théorique. Les étapes sont mémorisées dans un tableau de booléens.

Complétez la programmation pour obtenir le Grafcet ci-dessous.



### 13.4 Troisième exercice, GRAFCET à séquences multiples simultanées.

La routine à utiliser est GRAFCET\_ET.

Dans cette routine les étapes sont activées et désactivées par des équations logiques tel que vu dans la partie théorique. Les étapes sont mémorisées dans un tableau de booléens.

Complétez la programmation pour obtenir le Grafcet ci-dessous.

