

Ingénierie des systèmes cyber-physiques

R1.10 - Electricité - Electrotechnique

Partie 3 : Bases d'automatisme

Introduction à la logique câblée

Support de cours

Thomas Fromentèze



Introduction

Dans ce premier cours sur l'automatisme, on étudie les éléments fondamentaux rencontrés dans les systèmes automatisés de production et de fabrication. Nos premiers objectifs sont les suivants :

- Découvrir l'architecture générale des systèmes automatisés.
- Apprendre à câbler de premiers éléments électriques servant au contrôle et à la sécurité.
- Maîtriser les bases de l'algèbre binaire pour faciliter l'étude de circuits logiques.

Ces efforts sont motivés par la nécessité de développer des systèmes automatisés permettant de réaliser des séries de tâches successives. Ces machines sont essentiellement développées pour le secteur industriel afin de remplacer des actions humaines pour des tâches mécanisables, permettant une production continue, plus fiable et souvent plus rapide. En contre-partie, l'investissement initial nécessaire au développement de telles machines est très important. Il faut de plus avoir recours à du personnel qualifié, capable d'assurer la maintenance de tels équipements.



FIGURE 1 – Exemple de ligne de montage automatisée permettant l'encartonnage de paquets.
Source : <https://www.legraphiste3d.com>

En première année de BUT, on simplifie l'étude de ces systèmes en ne considérant que des éléments fonctionnant selon la logique du **Tout ou Rien (TOR)**, c'est à dire que chaque élément ne peut être que dans deux états possibles. Par exemple :

- Un moteur est activé ou éteint, on ne contrôle pas sa vitesse.
- Un vérin est complètement rentré ou sorti, il ne prend pas de position intermédiaire.
- Un capteur est soit activé, soit désactivé.

Chaque élément est alors associé à un état : **1 (activé, vrai, haut)** ou **0 (éteint, faux, bas)**.

On s'intéressera dans ce domaine à l'interaction entre la mécanique et des systèmes automatisés fonctionnant à partir d'énergies **électrique, pneumatique et hydraulique**.

1 Architecture d'un système automatisé

On commence par une définition générale d'un système automatisé :

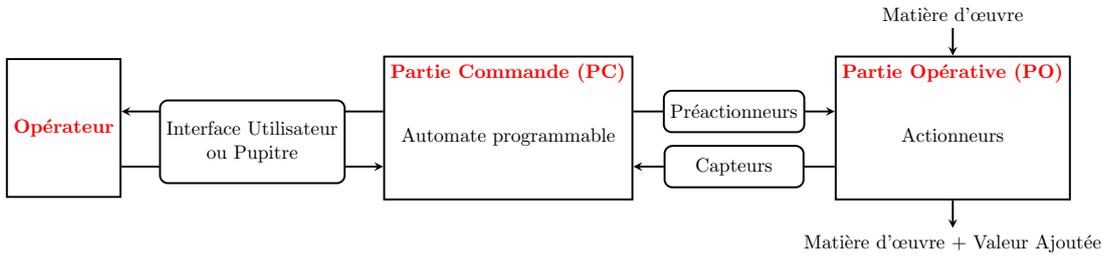


FIGURE 2 – Schéma d'un système automatisé

- **Partie Commande (PC)** : constituée d'un module électronique programmable (Ordinateur, Automate Programmable Industriel, Arduino, Nucléo, ESP32 ...) ou de circuits réalisant une logique câblée. Cette partie assure le pilotage du système automatisé par la coordination de tâches programmables.
- **Partie Opérative (PO)** : interagit directement avec les produits à transformer et/ou manipuler (Matière d'œuvre). On y retrouve souvent des systèmes mécaniques mis en mouvement par des actionneurs, tels que des moteurs et des vérins.

Il est généralement possible d'interagir avec un système automatisé par l'intermédiaire d'un pupitre ou d'une interface utilisateur pour les systèmes plus récents entièrement informatisés. Grâce à ces interfaces, un opérateur peut recevoir des informations sur l'état du système et envoyer des commandes.

Pour faciliter la compréhension de ce schéma et donner des exemples plus concrets de composants, un exemple basique est donné dans la figure suivante.

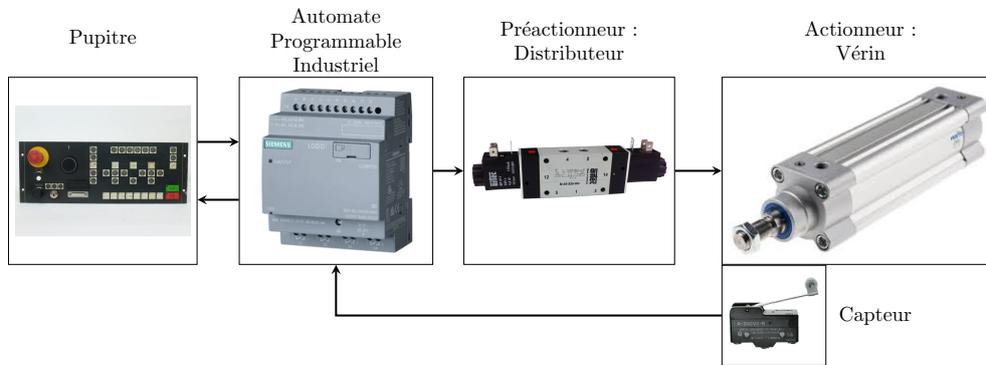


FIGURE 3 – Exemple de système automatisé permettant l'actionnement d'un vérin pneumatique

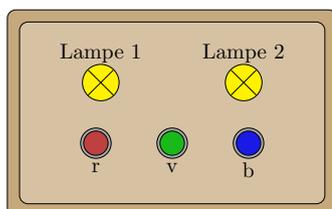
Sous la commande d'un **Automate Programmable Industriel (API)**, un vérin pneumatique peut être déployé ou rentré, l'information étant retournée à la Partie Commande par l'intermédiaire d'un capteur de fin de course. Ce schéma permet aussi de souligner l'utilité d'un préactionneur pour la conversion d'énergie dans certaines situations : le distributeur permet ici par une commande électrique de l'API d'acheminer ou non de l'air comprimé dans le vérin. Le câblage des systèmes électro-pneumatiques sera étudié plus en détail lors d'un prochain semestre.

On s'intéresse en premier lieu aux notions de logique câblée, laissant pour l'instant de côté les circuits logiques programmables qui sont plus récents. Ces notions seront abordées lors d'une prochaine unité d'enseignement avec l'utilisation des API.

2 Logique câblée

La logique câblée va nous permettre de résoudre des problèmes liés à l'exécution de tâches Tout Ou Rien avec des contacts électriques.

Prenons un exemple simple : une boîte comporte deux lampes et trois boutons-poussoir.

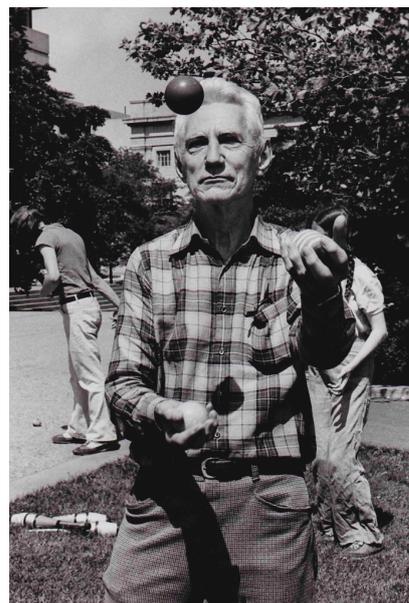


On vous demande de réaliser un câblage électrique afin de satisfaire le cahier des charges suivant :

- **Lampe 1** s'allume quand on appuie sur **r** ou **b**.
- **Lampe 2** s'allume quand on appuie sur **v** et **b**.
- **Lampe 1** et **Lampe 2** s'allument quand on appuie sur **v**.

Comment peut-on traduire efficacement des instructions logiques en câblage électrique ?

On parle de lampes pour cet exemple par simplicité, mais la même logique s'applique directement au contrôle des vérins et des moteurs.



Claude Shannon 1916 - 2001

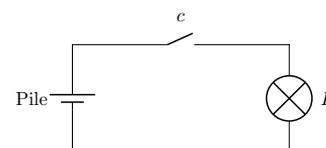
- Développe la logique câblée à 21 ans pendant son Master au MIT
- Père des télécommunications
- Film-documentaire *The bit player* pour les plus motivés et motivées

2.1 Principes fondamentaux

Le principe central introduit par Shannon est de considérer qu'un circuit électrique TOR peut être analysé avec de l'algèbre Booléenne, adaptée aux problèmes de logique.

Dans ce circuit, une lampe s'allume à condition qu'un contact soit fermé :

- Le contact c peut être ouvert ($c = 0$) ou fermé ($c = 1$).
- La lampe L est soit éteinte ($L = 0$), soit allumée ($L = 1$).



On remarque que dans un circuit TOR, chaque élément peut être défini par un état binaire, c'est à dire valant 0 ou 1.

On remarque aussi que la lampe s'allume quand le contact est fermé, soit $L = c$. On peut alors écrire une **table de vérité**, qui sera évidente dans un cas si simple :

Ce tableau permet de faire le lien entre tous les états possibles des capteurs (les variables) et ceux de l'actionneur du circuit.

| c | L |
|-----|-----|
| 0 | 0 |
| 1 | 1 |

Dans tous les schémas techniques vus en automatisme, tous les éléments sont représentés dans leur état de repos.

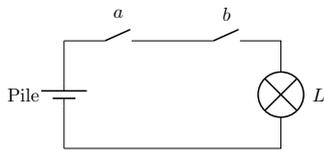
 contact normalement ouvert (NO) - *normally open* (NO)

 contact normalement fermé (NF) - *normally closed* (NC)

On s'intéresse maintenant aux circuits logiques élémentaires qui permettront ensuite d'en élaborer de plus complexes :

2.1.1 Association en série

Dans quelle(s) situation(s) est-ce que la lampe *L* s'allume ?



Avec deux contacts NO dans ce problèmes, on dénombre maintenant quatre situations à analyser.
La table de vérité suivante est à remplir.

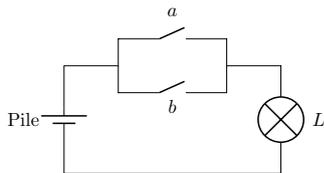
| <i>a</i> | <i>b</i> | <i>L</i> |
|----------|----------|----------|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

Ce montage de deux contacts en série s'appelle une **porte ET**. La relation entre l'état de la lampe *L* et ceux des contacts *a* et *b* s'écrit de la façon suivante :

$$L = a \cdot b$$

2.1.2 Association en parallèle

Dans quelle(s) situation(s) est-ce que la lampe *L* s'allume ?



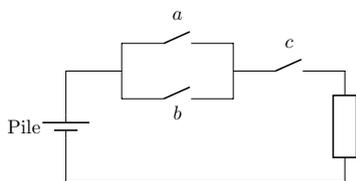
Avec deux contacts NO dans ce problèmes, on dénombre toujours quatre situations à analyser.
La table de vérité suivante est à remplir.

| <i>a</i> | <i>b</i> | <i>L</i> |
|----------|----------|----------|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

Ce montage de deux contacts en parallèle s'appelle une **porte OU**. La relation entre l'état de la lampe *L* et ceux des contacts *a* et *b* s'écrit de la façon suivante :

$$L = a + b$$

2.1.3 Montages plus complexes



Tous les cas plus compliqués vont pouvoir être décomposés à partir des portes logiques précédentes.

De façon générale, on utilise un symbole de résistance à la place de l'actionneur pour représenter une charge quelconque qui absorbe l'énergie électrique.

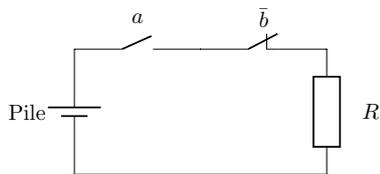
| <i>a</i> | <i>b</i> | <i>c</i> | <i>R</i> |
|----------|----------|----------|----------|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

Ce montage peut s'écrire avec l'expression booléenne suivante : $R = (a + b) \cdot c$.

Remarque : Quand on a *N* contacts indépendants dans un problème, il y a 2^N états possibles pour un circuit. Dans le dernier cas étudié, on a $2^3 = 8$ configurations différentes. On remplit les tables de vérités en comptant en binaire pour s'assurer qu'on les étudie bien toutes.

2.1.4 Le cas des contacts normalement fermés

On a vu précédemment qu'un contact peut être passant par défaut, on parle alors de contact normalement fermé (NF) ou *normally closed* (NO).



Dans ce cas, le symbole du contact change et on écrit son nom avec une barre, ici \bar{b} (on dit "b barre", ou "NON b"). Cette barre a pour effet d'inverser la valeur booléenne sur laquelle elle est appliquée :

- si $b = 0$, alors $\bar{b} = 1$.
- si $b = 1$, alors $\bar{b} = 0$.

| a | b | \bar{b} | R |
|-----|-----|-----------|-----|
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |

On souhaite remplir la table de vérité ci-contre. On remarque qu'il n'y a que 4 configurations étudiées car il n'y a que deux variables indépendantes (a et b , soit $2^2 = 4$ états).

Dans cet exemple, l'expression booléenne associée au circuit est :

$$R = a \cdot \bar{b}$$

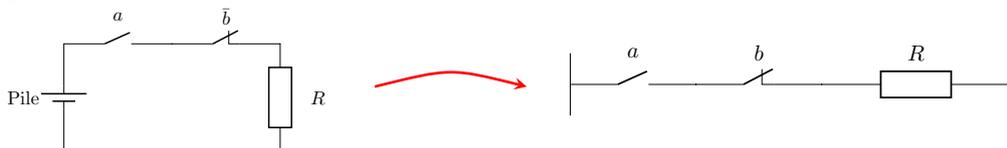
2.1.5 Théorème de De Morgan

Un opérateur d'inversion peut être appliqué non seulement sur une variable, mais aussi sur une expression entière. Il est nécessaire de connaître deux propriétés importantes :

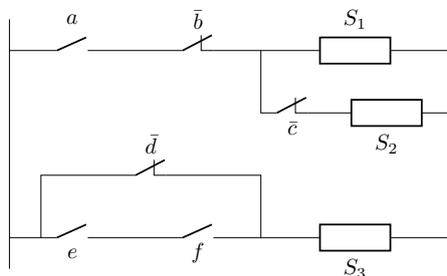
$$\overline{a \cdot b} = \bar{a} + \bar{b} \quad \overline{a + b} = \bar{a} \cdot \bar{b}$$

2.1.6 Représentation des circuits de logique câblée

Les schémas techniques et industriels utilisent une notation simplifiée des circuits de logique câblée. On ouvre les alimentations en tension de façon à éviter d'avoir à dessiner des boucles complètes :



On peut grâce à cette notation avoir une meilleure vue d'ensemble de toutes les relations entre capteurs et actionneurs.



- Les pôles + et - peuvent être étendus pour représenter plusieurs circuits de façon condensée.
- Les capteurs et les sorties peuvent être montés en parallèle et en série.

Trouver les expressions booléennes des sorties S_1 , S_2 et S_3 :

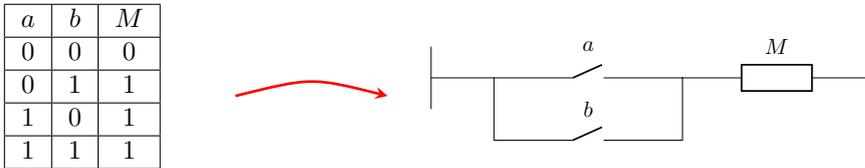
$$S_1 =$$

$$S_2 =$$

$$S_3 =$$

2.2 Simplification des expressions de logique câblée

Une des tâches les plus courantes à réaliser dans ce domaine est la conversion d'un cahier des charges en circuit équivalent. On peut parfois deviner directement une solution :



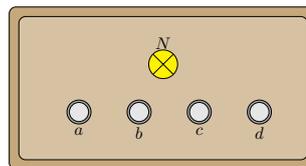
Dans ce tableau, on remarque que la sortie *M* est activée si l'on ferme le contact *a* **OU** le contact *b*, donc les contacts sont montés en parallèle. Il ne semble pas nécessaire d'avoir recours à des inversions de valeur pour que cette logique soit valable, donc les contacts sont ouverts au repos (NO).

Prenons maintenant un exemple un peu plus complexe et réaliste :

| <i>a</i> | <i>b</i> | <i>c</i> | <i>d</i> | <i>N</i> |
|----------|----------|----------|----------|----------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Il est parfois très difficile de déterminer directement un circuit logique à partir d'une table de vérité, notamment lorsque le nombre de variables devient important.

On vous demande de câbler le boîtier suivant pour qu'un moteur se mette en route lorsque les bonnes combinaisons de boutons sont pressées. Ces combinaisons sont données dans le tableau ci-contre. Pour la preuve de concept, le moteur est remplacé par une lampe *N*.



Dans un exemple avec 4 variables indépendantes, il y a $2^4 = 16$ configurations possibles pour ce circuit.

Il est nécessaire de maîtriser des outils permettant de transformer nos tables de vérité en circuits équivalents.

2.2.1 Binaire réfléchi

Nous allons d'abord avoir besoin de comprendre le principe de binaire réfléchi, aussi appelé code de Gray. Cette façon de compter en binaire a pour avantage de ne nécessiter qu'une seule modification par bit à chaque incrémentation.

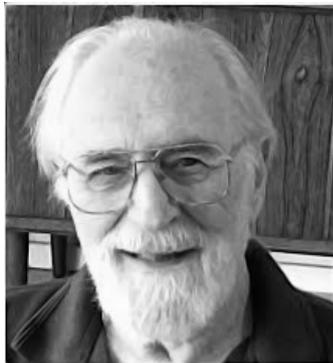
| Décimal | Binaire naturel | Binaire réfléchi |
|---------|-----------------|------------------|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |

→ - miroir
→ -)
→ -) - - - - - miroir

Pour passer d'un code à l'autre et contrairement au binaire naturel, il n'est toujours nécessaire de changer qu'un seul digit :

- de 3 à 4 : 0010 → 0110
- de 5 à 6 : 0111 → 0101

2.2.2 Tableaux de Karnaugh



Maurice Karnaugh
1924 - 2022

- Développe les tableaux de Karnaugh en 1953 (29 ans)

Les tableaux de Karnaugh permettent de simplifier les tables de vérités à l'aide d'une technique graphique relativement simple exploitant les propriétés du code de Gray. Cette technique est initialement prévue pour résoudre un problème contenant seulement deux variables. Il est cependant possible de grouper les variables de n'importe quel problème pour former deux ensembles.

Dans notre exemple, les 4 variables a, b, c, d sont converties en deux variables ab et cd , qui peuvent prendre les valeurs suivantes (en binaire réfléchi) : 00, 01, 11, 10.

On peut alors remplir notre premier tableau de Karnaugh pour trouver le câblage de notre boîte noire :

| a | b | c | d | N |
|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



| N | ab | | | |
|-----|------|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 0 | 1 | 1 | 1 |
| 01 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | 1 | 1 |
| 10 | 0 | 0 | 0 | 0 |

Nous ne sommes pas plus avancés pour l'instant parce que ce tableau nous permet seulement de présenter les mêmes informations sous une forme différente. La prochaine étape consiste à grouper les cas où notre sortie est activée ($N = 1$) en suivant les règles suivantes :

- On souhaite que l'ensemble des "1" soient compris dans **au moins** un groupe.
- Les groupes doivent être aussi grands que possible et de forme rectangulaire.
- Le nombre de "1" dans chaque groupe doit obligatoirement être une puissance de 2 (1,2,4,8,16).

Application des règles de sélection au cas étudié :

| | | | | | |
|----|----|----|----|----|----|
| N | | 00 | 01 | 11 | 10 |
| | 00 | 0 | 1 | 1 | 1 |
| cd | 01 | 0 | 1 | 1 | 1 |
| | 11 | 0 | 1 | 1 | 1 |
| | 10 | 0 | 0 | 0 | 0 |

↗ $ab \ \alpha$ ↗ β
↘ δ ↘ γ

On peut faire 4 groupes de 4 "1" pour sélectionner l'ensemble des cas où N est activée. On ne peut pas faire ici de groupes de 6 ou 9 éléments parce que ce ne sont pas des puissances de 2.

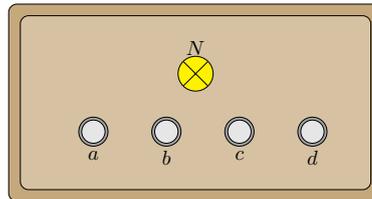
Chaque groupe représente une façon d'activer la sortie N : $\alpha = 1$ **OU** $\beta = 1$ **OU** $\gamma = 1$ **OU** $\delta = 1$ **OU** une combinaison des 4.

L'expression booléenne de la sortie N est donc la suivante : $N = \alpha + \beta + \gamma + \delta$.

Il faut maintenant exprimer chaque groupe en fonction de la valeur des variables les composant. On rappelle que l'objectif est de transformer un cahier des charges, exprimé sous la forme d'une table de vérité, en câblage de 4 boutons permettant l'activation d'un moteur. On étudie la démarche pour un des groupes et le même raisonnement sera appliqué aux autres.

- Groupe α :** on repère les variables qui valent toujours la même valeur dans le groupe rouge :
- a vaut 0 ou 1.
 - b vaut toujours 1.
 - c vaut toujours 0.
 - d vaut 0 ou 1.

Seules b et c ne changent pas de valeur dans ce groupe. En reprenant notre problème initial, cela signifie que l'on a identifié un groupe de combinaisons permettant d'activer la sortie : tant que l'on appuie sur b et que l'on n'appuie pas sur c , peu importe l'état des boutons a et d , la sortie N s'active.



L'expression de α s'écrit alors $\alpha = b \cdot \bar{c}$.

De cette façon, on s'assure que si les deux conditions $b = 1$ **ET** $c = 0$ (autrement dit $\bar{c} = 1$) sont remplies, alors la sortie N est activée. Les variables a et d n'apparaissent plus dans ce groupe parce qu'on a identifié que l'état de la sortie ne dépendait pas de leur activation.

En appliquant une logique similaire aux trois autres groupes, on trouve l'expression suivante (à redémontrer chez soi) :

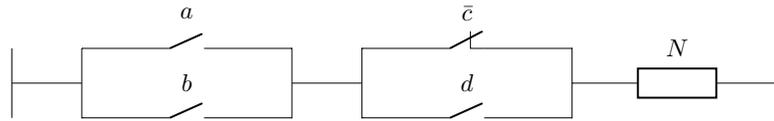
$$N = \underbrace{b \cdot \bar{c}}_{\alpha} + \underbrace{a \cdot \bar{c}}_{\beta} + \underbrace{b \cdot d}_{\gamma} + \underbrace{a \cdot d}_{\delta}$$

On peut appliquer les propriétés connues d'associativité, commutativité et distributivité des opérateurs \cdot et $+$ pour factoriser et réduire cette expression :

$$N = \bar{c} \cdot (b + a) + d \cdot (a + b)$$

$$N = (a + b) \cdot (\bar{c} + d)$$

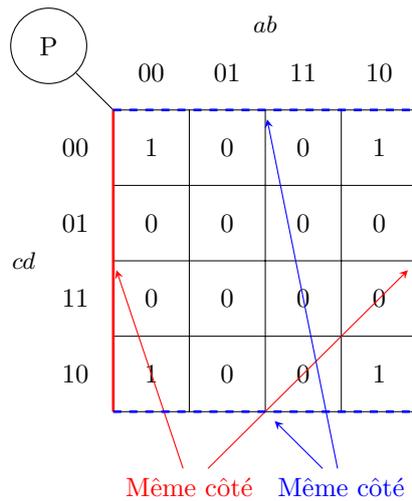
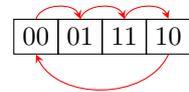
On arrive enfin à une expression compacte que l'on peut traduire sous forme de circuit équivalent :



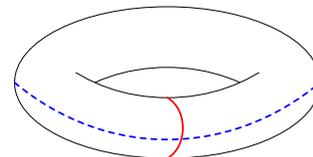
2.2.3 Propriétés cycliques des tableaux de Karnaugh

On a vu que les tableaux de Karnaugh utilisent le code de Gray pour passer d'un état à un autre. Les propriétés du binaire réfléchi permettent d'identifier les variables n'impactant pas l'activation des sorties sur chaque groupe, facilitant ainsi la réduction des expressions booléennes.

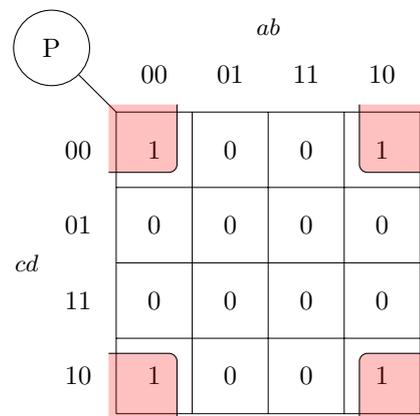
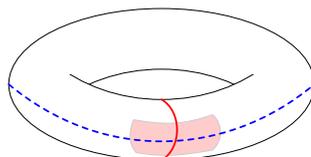
Grâce aux propriétés du code de Gray, pour chaque variable (par exemple ab) la transition d'un état au suivant se fait en changeant la valeur de seulement un digit. On remarque que cette propriété est vraie même lorsque l'on arrive au dernier code et que l'on revient au premier.



Cette continuité permet d'obtenir une propriété un peu surprenante : les côtés gauche et droite du tableau se retrouvent confondus, de même que les côtés haut et bas. En remplissant le tableau de façon à faire coïncider ces limites, on retrouve alors la forme d'un tore.



On peut donc réaliser des groupes qui dépassent les bords du tableau, tant qu'ils restent continus si l'on prend en compte ces nouvelles conditions aux limites.

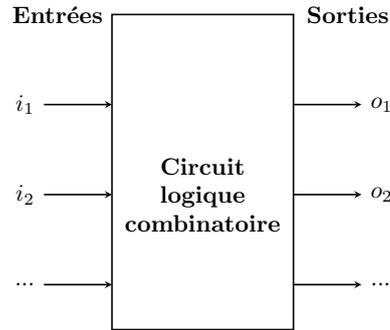


Cette expression booléenne peut donc être réduite à $P = \bar{b} \cdot \bar{d}$, b et d étant les seules variables qui ne changent pas dans ce groupe, valant tout le temps 0.

3 Du combinatoire au séquentiel

3.1 Logique combinatoire

Les circuits abordés jusqu'à maintenant dépendent uniquement de l'état des différentes variables d'entrée (contacts, capteurs). Il n'est pas nécessaire de connaître l'état précédent des variables du circuit pour déterminer la valeur des sorties.



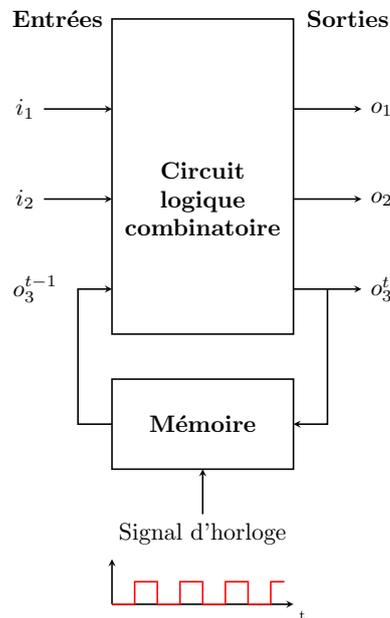
3.2 Logique séquentielle

Les circuits séquentiels sont un peu plus complexes car l'état de leurs variables de sortie dépend à la fois de la valeur des variables d'entrées et d'une ou plusieurs sorties à l'état précédent.

Le temps est découpé à intervalles réguliers définis par un signal d'horloge (habituellement de quelques kHz à quelques GHz en fonction des machines et des applications, soit un coup d'horloge tous les millièmes à tous les milliardièmes de seconde).

Certaines sorties sont ainsi mises en mémoire et réinjectées en entrée du système combinatoire considéré au coup d'horloge suivant.

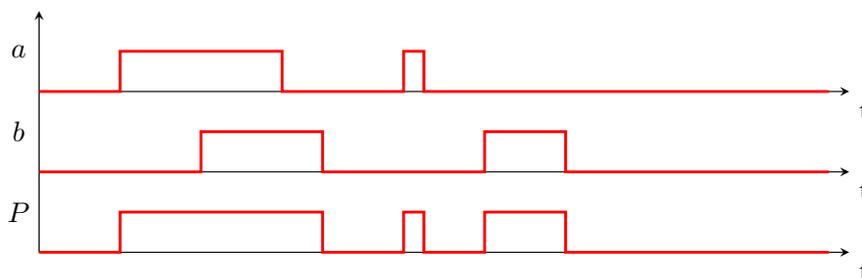
On différencie les systèmes séquentiels **synchrones** qui ne peuvent changer d'état qu'au rythme imposé par l'horloge, des systèmes **asynchrones** dont la valeur des sorties peut varier dès qu'un changement est réalisé en entrée.



3.3 Chronogrammes

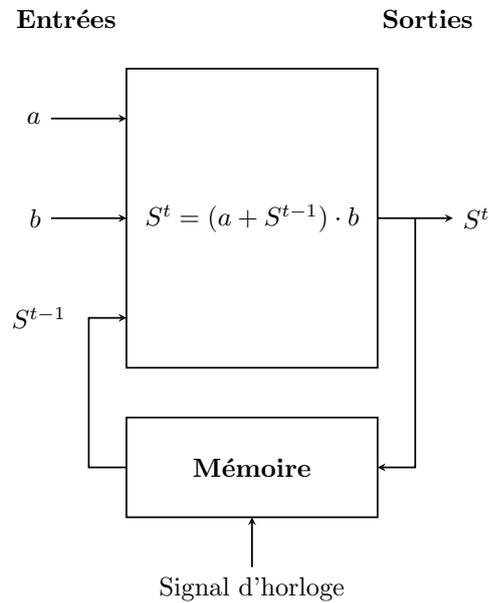
On conclue ce cours avec la notion de chronogrammes, qui permet d'avoir une représentation graphique de l'évolution des états d'une machine. Pour faciliter la compréhension, on analyse un circuit logique connu formant une porte OU de sortie P définie telle que $P = a + b$. a et b sont ici définis par l'état de boutons poussoirs normalement ouverts.

Un utilisateur appuie de façon arbitraire sur différentes combinaisons de boutons a et b :



Suivant la logique de la porte OU, la sortie P est activée lorsque qu'au moins une des entrées est vraie.

Dans le cas d'une machine séquentielle, au moins une des entrées doit correspondre à une sortie retardée. On représente maintenant la vitesse d'horloge dans les chronogrammes à l'aide de pointillés.

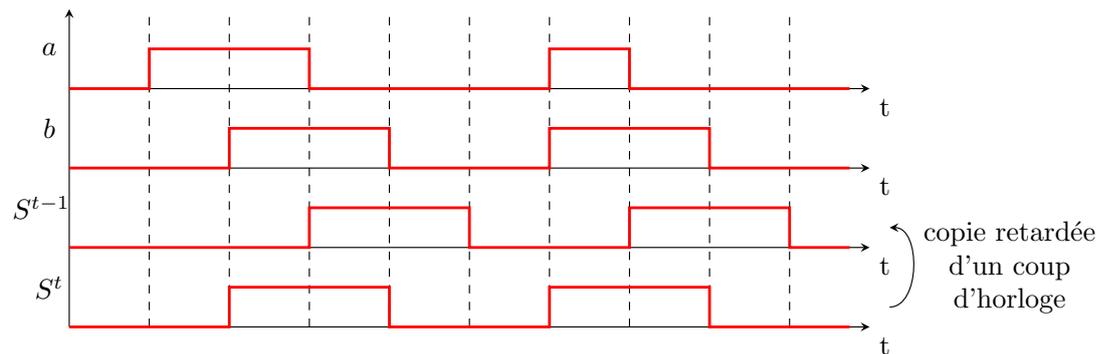


On se limite ici à l'étude de machines séquentielles synchrones. Dans cet exemple illustratif, la sortie est définie telle que :

$$S^t = (a + S^{t-1}) \cdot b$$

On choisit arbitrairement des combinaisons de valeurs sur les entrées a et b en fonction du temps. La valeur de la variable de sortie S^t ne peut être déterminée que si l'on connaît la valeur de cette dernière à l'instant précédent S^{t-1} .

Pour prendre en compte ce décalage temporel, on retarde la sortie S^t d'un coup d'horloge, dont le rythme est représenté par les pointillés dans la figure suivante :

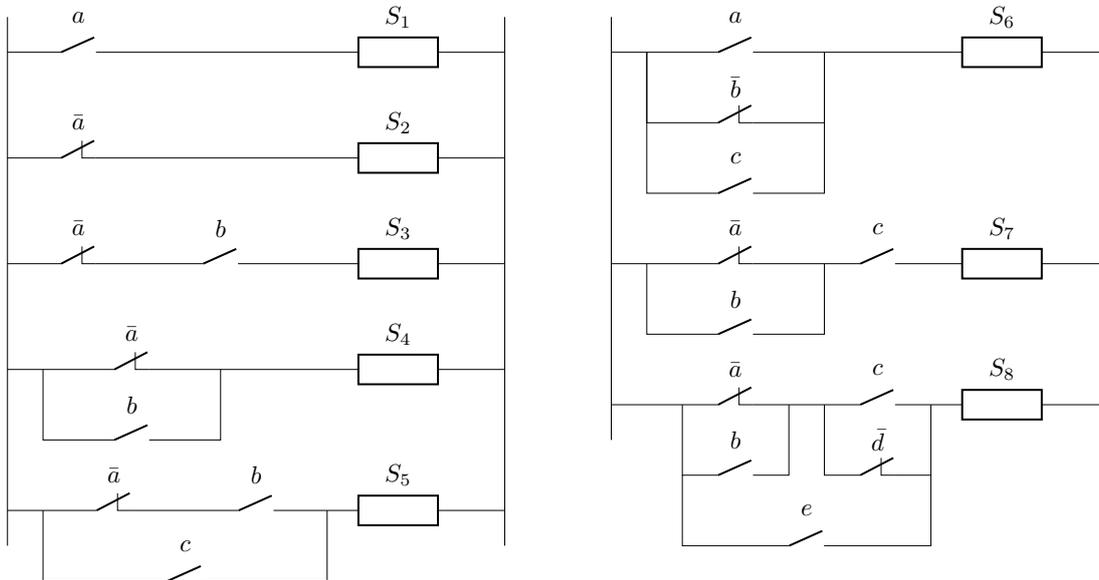


Pour cet exemple, la sortie S^t est activée si b est activée à l'instant t ET si a OU la sortie à l'instant précédent S^{t-1} est activée.

TD1 - Introduction à la logique câblée

1 Conversion de circuits en expressions booléennes

On s'intéresse premièrement à la conversion des circuits électriques suivants en expressions mathématiques équivalentes. Les contacts considérés ne peuvent être qu'ouverts ou fermés, il n'existe pas d'état intermédiaire.



2 Conversion d'expressions booléennes en circuits équivalents

On souhaite maintenant convertir des expressions booléennes en logique câblée. Cet exercice permet de déterminer le câblage d'actionneurs ou de pré-actionneurs en fonction de règles de fonctionnement et de contacts associés.

1. $R_1 = a + b + c \cdot \bar{d}$
2. $R_2 = a \cdot b \cdot c + a \cdot b \cdot \bar{c}$
3. $R_3 = (a + \bar{b}) \cdot (c + \bar{d})$
4. $R_4 = a \cdot b \cdot c + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + \bar{a} \cdot \bar{b} \cdot c$
5. $R_5 = \overline{a \cdot b} \cdot c$
6. $R_6 = \bar{a} \cdot b \cdot (c + d) \cdot (\bar{e} + f)$

3 Tables de vérité

On souhaite enfin déterminer des tables de vérité pour différentes expressions booléennes.

1. Un circuit logique est composé de contacts normalement ouverts a, b, c et d , permettant d'alimenter une charge R . Déterminer le nombre d'états possibles de ce circuit.
2. On souhaite alors généraliser cette étude. Quel est le nombre d'états d'un circuit composé de N contacts ?
3. Déterminer la table de vérité de l'expression précédente R_2 .
4. Déterminer enfin la table de vérité de l'expression R_1 .

TD2 - Tableaux de Karnaugh

1 Analyse d'un cahier des charges

Un client vous demande de réaliser une machine motorisée présentant trois contacts a, b et c permettant d'activer un moteur M .

Le cahier des charges transmis par le client est le suivant :

1. Lister premièrement les combinaisons qui permettent d'assurer l'activation du moteur
2. Représenter alors un premier circuit logique permettant de satisfaire ce cahier des charges
3. Après une simplification de l'expression booléenne, donner une version plus compacte du câblage logique de ce moteur.

| a | b | c | M |
|-----|-----|-----|-----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

2 Un outil pour les cas plus complexes

Le précédent exercice a permis de mettre en évidence qu'il était possible de déterminer le câblage logique associé à un cahier des charges relativement simple. Dans l'éventualité où le nombre de contacts impliqués est plus grand, les techniques précédemment utilisées peuvent être assez limitées.

| a | b | c | d | S |
|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Pour illustrer ces limitations, on analyse maintenant une table de vérité plus complexe. On souhaite premièrement remplir le tableau de Karnaugh suivant :

| | | | | |
|------------|------|----|----|----|
| S | ab | | | |
| | 00 | 01 | 11 | 10 |
| 00 | | | | |
| 01 | | | | |
| cd 11 | | | | |
| 10 | | | | |

A l'aide des méthodes vues en cours, déterminer l'expression booléenne simplifiée de S .

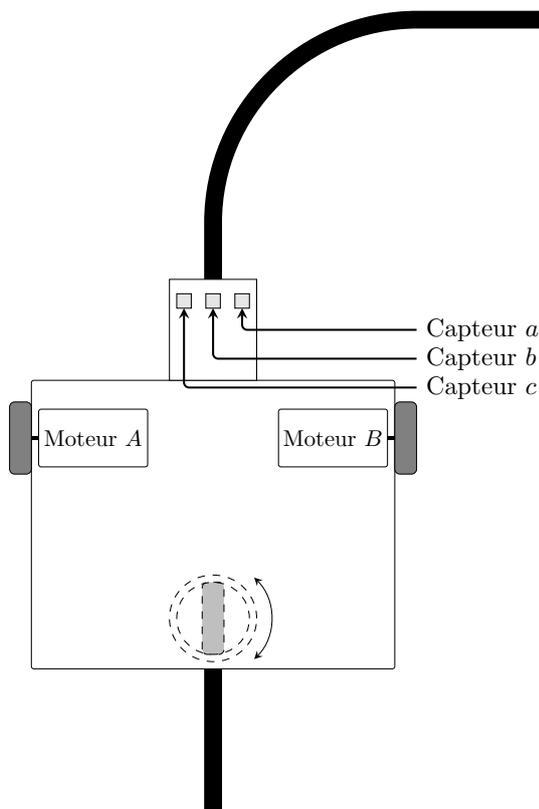
3 Approfondissements

Simplifier les expressions booléennes suivantes à l'aide de tableaux de Karnaugh, en remplissant des tables de vérités intermédiaires si nécessaire.

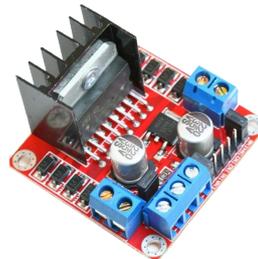
1. $R_1 = a \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c$
2. $R_2 = \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot b \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot c \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$
3. $R_3 = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot d + a \cdot \bar{b} \cdot c \cdot \bar{d}$
4. $R_4 = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot b \cdot \bar{c} \cdot \bar{d} + a \cdot b \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + a \cdot \bar{b} \cdot c \cdot \bar{d}$
5. $R_5 = (\bar{a} \cdot b + a \cdot \bar{b} + a \cdot b) \cdot (c \cdot \bar{d} + \bar{c} \cdot d) + (\bar{a} \cdot b + a \cdot \bar{b}) \cdot (c \cdot \bar{d})$

TD3 - Robot suiveur de ligne

On souhaite concevoir un robot capable de suivre une ligne noire au sol. Deux moteurs à courant continu A et B sont reliés aux roues avant du robot. Une troisième roue non-motrice, dite roue folle, est libre sur un axe de rotation. Trois capteurs optiques TCRT5000 notés a , b et c permettent de détecter la réflexion d'un signal lumineux au sol. Ces derniers sont utilisés en mode tout ou rien. La ligne noire est suffisamment large pour qu'au moins un des trois capteurs soit toujours activé. La mise en marche indépendante des moteurs est permise par un double pont en H L298N. Un micro-contrôleur arduino doit être programmé pour interroger continuellement l'état des capteurs et activer la rotation des moteurs par l'intermédiaire des ponts en H.



Capteur optique TCRT5000



Double pont en H L298N



Microcontrôleur arduino

1. Déterminer les tables de vérité associées aux moteurs A et B en fonction de l'état des capteurs a , b et c . On considère que les capteurs renvoient la valeur 0 en présence de la ligne noire, 1 sinon. Un moteur à l'état 1 tourne et n'est pas alimenté à l'état 0. On utilisera la notation \times pour l'état d'un moteur dans les combinaisons des tables de vérité qui semblent impossibles.
2. En déduire les expressions booléennes simplifiées de A et B . Les états \times n'étant jamais rencontrés en pratique, on peut les utiliser pour former de plus grands groupes dans les tableaux de Karnaugh.
3. Proposer un schéma de câblage du système, en repérant les capteurs, le pré-actionneur, les actionneurs et la partie programmable.

Ingénierie des systèmes cyber-physiques

R1.10 - Electricité - Electrotechnique

Partie 3 : Bases d'automatisme

TP1 - Logique câblée

Thomas Fromentèze

Le TP dure 4h et un compte-rendu par groupe est à rendre sur feuille en fin de séance.

Objectifs

Vous devez dans le cadre de ce TP prouver que vous avez acquis les compétences relatives aux câblages de circuits logiques. Vous allez ainsi être amené(e)s à convertir des expressions logiques en circuits équivalents. Réciproquement, vous devez aussi être en mesure de convertir un cahier des charges transmis par un client en expression booléenne aussi simple que possible que vous transformerez ensuite en circuit logique. Vous serez enfin amenés à réaliser des vérifications expérimentales des solutions proposées.

L'ensemble des montages réalisés pendant le TP devront faire l'objet sur vos rapports de schémas électriques et de commentaires (des plus simples aux plus complexes).

1 Vérifications

On réalise une première série de câblages pour vérifier le bon fonctionnement des mallettes de TP. Ces dernières ayant été réalisées partiellement à partir d'éléments recyclés, l'identification et la maintenance des fonctions de cette platine font parties des compétences évaluées.

1. Vérifier le bon fonctionnement des LEDs R et S.
2. Tester l'ensemble des contacts normalement ouverts et fermés des quatre boutons.
3. Réaliser successivement un montage des expressions $R_1 = c \cdot d$ et $R_2 = c + d$ puis déterminer expérimentalement les tables de vérité dans ces deux cas.

2 Exercices sur feuille

La suite de ce TP est constituée d'exercices qui seront utilisés pour les validations expérimentales de la section suivante.

Déterminer dans chaque cas la table de vérité, le tableau de Karnaugh et le câblage logique associé à ces expressions booléennes :

1. $S_1 = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot c \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot d + a \cdot b \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot d$
2. $S_2 = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot c \cdot d + \bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot c \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot c \cdot d + a \cdot b \cdot \bar{c} \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot \bar{d} + a \cdot b \cdot c \cdot d$
3. $S_3 = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot d + \bar{a} \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot c \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot c \cdot d + a \cdot b \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot d$

3 Validation expérimentale

On souhaite maintenant réaliser les câblages des expressions précédentes S_1 , S_2 et S_3 . Dans chaque cas, l'ensemble des combinaisons de la table de vérité doivent être vérifiées.

Comme précédemment, l'ensemble des câblages seront donnés préalablement sur feuille et les résultats obtenus devront être synthétisés à l'écrit.

4 Cahiers des charges inconnus

Un microcontrôleur arduino est programmé pour générer une table de vérité aléatoire dépendant de 4 variables booléennes nommées i_1, i_2, i_3 et i_4 . Dans la version actuelle du code, seule la sortie O_1 passe à l'état 1 (3.3V) lorsqu'une combinaison adéquate est donnée sur ses entrées et reste à zéro sinon.

- Pour simplifier les câblages internes, les entrées sont activées par défaut et passent à zéro lorsqu'elles sont reliées à un état bas (0V). On parle alors de montage pull-up que nous étudierons plus tard de votre formation. La logique de câblage est donc inversée par rapport à ce que vous avez rencontré jusqu'alors.
- Pour vous simplifier la tâche, le code inverse l'état des entrées mesurées. Tout se passe alors sur le papier comme si vous aviez alimenté les boutons puis les entrées par un état haut (3.3V) permettant d'activer ces dernières.
- Pour harmoniser les notations, on impose de brancher les boutons tels que $a \rightarrow i_1, b \rightarrow i_2, c \rightarrow i_3$ et $d \rightarrow i_4$.
- En cas d'erreur de câblage entraînant un court-circuit, l'arduino n'est plus alimenté et l'exercice est perdu. Suivant un premier malus sur la note de TP, il faudra tout reprendre à zéro avec un nouvel exercice généré aléatoirement au redémarrage de la mallette.

Je dispose d'un moyen de vérifier sur mon téléphone la table de vérité communiquée par votre mallette. Pendant que je consulte cette dernière, le microcontrôleur se met en pause et il n'est plus possible de tester des combinaisons.

1. Déterminer le nombre de combinaisons indépendantes réalisables à partir de 4 variables booléennes.
2. Déterminer ensuite le nombre d'exercices qu'il est possible de rencontrer dans cette partie du TP.
3. Proposer un câblage sur feuille puis sur la mallette permettant d'interroger la table de vérité générée à la mise sous tension de la mallette.
4. Déterminer la table de vérité expérimentalement.
5. Déterminer ensuite le câblage simplifié équivalent.
6. Après avoir décâblé le précédent montage, réaliser un circuit permettant de reproduire la table de vérité précédente.