

# La vision artificielle et ses applications industrielles

**Jonathan BONNET**

**Cora DONCHE**

**Lucie MOREL**

**Damien VIGNON**

Terminale STL Physique de Laboratoire et des  
Procédés Industriels

Option Mesures Physiques

Lycée Technique des Catalins

Montélimar

## INTRODUCTION

*Nous avons réalisé un objet lumineux animé d'un mouvement de rotation, commandé par un moteur. Cet objet peut prendre une infinité de positions.*

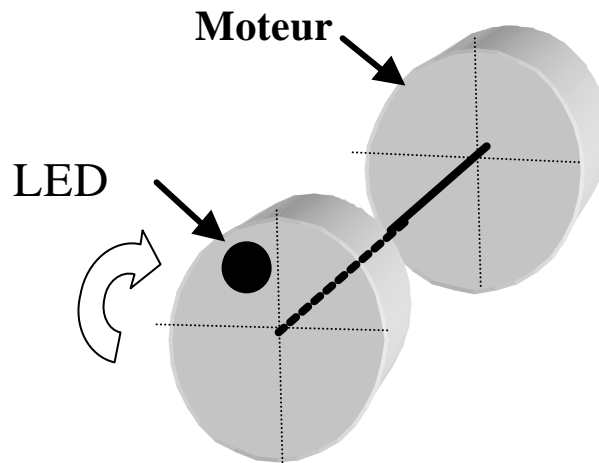
*Nous avons construit un appareil photo numérique très simple, permettant de mémoriser des positions particulières de cet objet.*

*Cette photographie est ensuite utilisée pour contrôler la position de l'objet : lorsque la position de l'objet correspond à la position photographiée, le moteur s'arrête (asservissement de position).*

*Dans un premier temps, toute l'application est réalisée avec des fonctions électroniques simples. Grâce à un système d'acquisition de données, les données constituant la photo numérique peuvent être transmises à un ordinateur. Le système peut alors être simplifié : certaines cartes « électroniques » peuvent être supprimées. Les fonctions correspondantes sont réalisées par un programme que nous avons écrit (langage : Agilent VEE).*

*L'instrument étant très simple, il est peu précis : nous avons évalué l'incertitude sur le repérage de la position de l'objet et compris comment l'on pouvait améliorer les performances de l'ensemble.*

L'objet lumineux animé :



Le moteur entraîne l'émetteur constitué d'une LED alimentée par des piles.

Lorsque le moteur tourne, la LED décrit un cercle et constitue un objet lumineux animé.

**Notre objectif est d'obtenir une photographie numérique de la position de l'objet à une date  $t$ .**

Pour simplifier la mise au point de notre projet, nous avons été amenés à réaliser un émetteur constitué de huit LEDs. Ces LEDs s'éclairent successivement et simulent, **sur huit positions**, la LED entraînée par le moteur (infinité de positions).

Dans un premier temps nous utiliserons ce simulateur pour expliquer comment est construit l'appareil photo numérique.

En conclusion, nous reviendrons à l'émetteur entraîné par le moteur pour évaluer les limites de notre instrument.

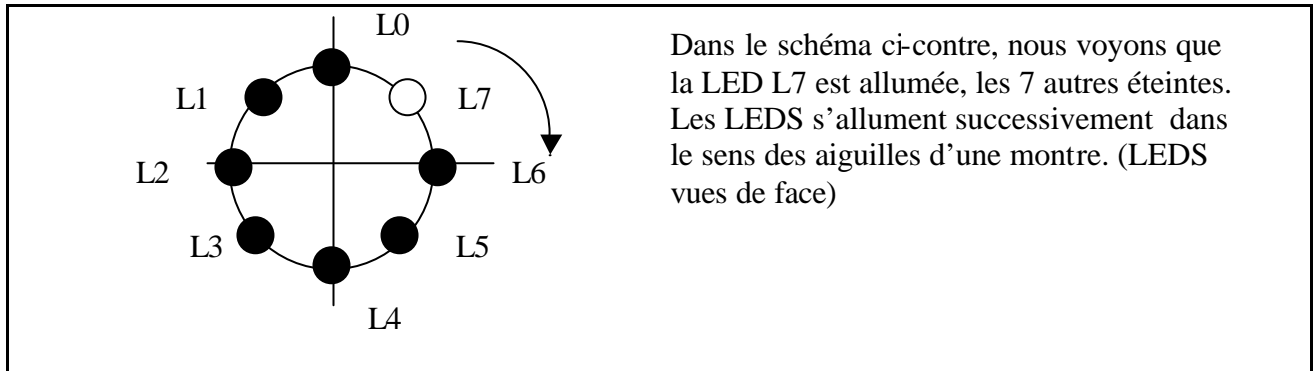
# Première partie : la photo numérique

## I L'objet lumineux :

Cet objet se compose de 8 LEDS en cercle simulant un moteur en rotation : une parmi les huit LEDS est allumée.

Pour pouvoir les reconnaître, on a mis en place un codage, en les numérotant de L0 à L7.

### Schéma du dispositif des LEDS :



Pour l'instant, nous n'avons qu'une image visuelle de l'état des LEDS.

Pour représenter simplement l'état des LEDS, nous avons mis en place un codage :

-Si la LED L0 est allumée, son état est 1 et  $L0 = 1$ .

-Si la LED L0 est éteinte, son état est 0 et  $L0 = 0$ .

Il en est de même pour les autres LEDS. L'état d'une LED (allumée ou éteinte) est donc représenté par 1 bit (0 ou 1).

Au lieu de décrire l'état de chacune des LEDS, (ce qui est long), on va mettre au point un autre code réunissant l'état de chaque LED. Nous avons appelé ce codage un MOT formé de huit bits, car il réunit les huit états des LEDS. La donnée MOT = L7 L6 ...L0.

Dans l'exemple précédent, seule la LED L7 est allumée donc, MOT s'écrira : MOT = 10000000 = 128.

### La donnée MOT est une information " optique" : on ne peut la lire qu'avec nos yeux.

A l'aide de MOT, nous aurons donc la possibilité de représenter l'état de chaque LED, donc l'état de l'objet. Les huit LEDS réunies vont donc simuler, en quelque sorte, le moteur en rotation.

Lorsque l'objet est animé, la donnée MOT change au cours du temps. La donnée MOT ne peut prendre que 8 valeurs : on ne peut repérer que 8 positions de l'objet. L'objet que nous voulons photographier pouvant prendre une infinité de positions, beaucoup d'informations sont perdues. Par contre, cette simplification nous a rendu service lors de la mise au point du matériel et du logiciel.

**Pour photographier l'objet il faut mémoriser l'état MOT de l'objet à une date t.**

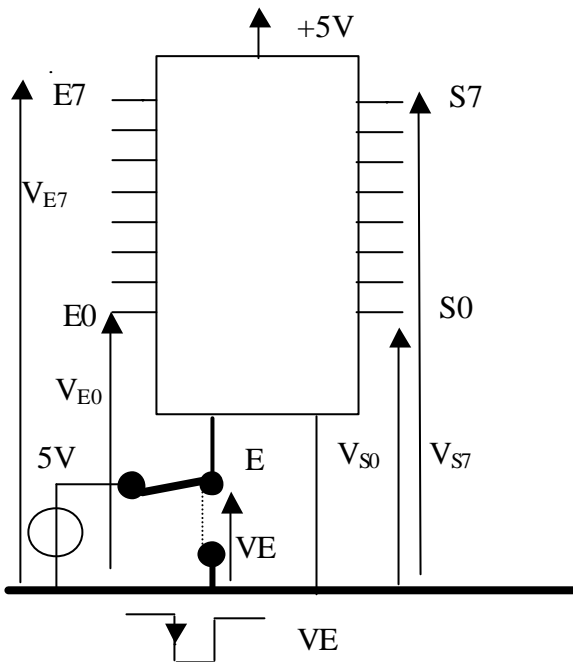
## II- Mémoriser un état « MOT » de l'objet :

### 1- Utilisation d'une maquette « mémoire »

La mémorisation de notre objet est capitale, mais elle est contraignante :

**ENTREE**

**SORTIE**



La maquette doit être alimentée par une tension de +5V.

Sur la maquette, il y a 8 bornes d'entrée (E0 à E7) et 8 bornes de sorties (S0 à S7). A chaque borne d'entrée correspond une tension  $V_{Ei}$ .

Une tension  $V_{Ei}$  correspond à un état logique  $Ei$

$0 < V_{Ei} < 0,8V \rightarrow Ei = 0$

$3,5 < V_{Ei} < 5V \rightarrow Ei = 1$

Les tensions d'entrée  $V_{Ei}$  **DOIVENT**

correspondre à l'état logique 0 ou 1.

A chaque borne de sortie correspond une tension  $V_{Si}$ , codée de la même façon que  $V_{Ei}$ .

L'écriture se fait à partir de l'entrée d'écriture E.

A l'instant t, choisi pour prendre la photo, on fait basculer la tension  $V_E$  de 5 V à 0V en

manoeuvrant l'interrupteur. Les tensions

d'entrée sont alors recopiées à la sortie, donc  $V_{Ei} = V_{Si}$ .

Lorsque l'on fait remonter la tension  $V_E$  à 5 V,

les tensions  $V_{Ei}$  et  $V_{Si}$  deviennent indépendantes les unes des autres.

Donc si  $V_{Ei}$  change,  $V_{Si}$  ne change pas : à t, les tensions  $V_{Ei}$  ont été mémorisées.

#### Codages :

A chaque tension d'entrée  $V_{Ei}$ , on associe un état logique  $Ei$  (0 ou 1).

A l'entrée de la mémoire, ces 8 codes forment un mot binaire de 8 bit : **MOT1 = E7E6...E0**

De même, on associe à chaque tension de sortie  $V_{Si}$  un état logique  $Si$  (0 ou 1).

Les 8 états de sortie forment de la même façon un MOT binaire : **MOT2 = S7S6...S0**

**On présente la donnée MOT1 à l'entrée de la mémoire.**

**A une date t on déclenche le signal d'écriture.**

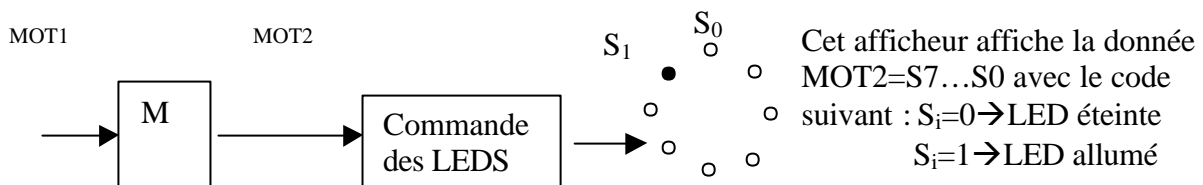
**La mémoire recopie MOT2 = (MOT1 à t) et conserve cette information jusqu'à l'écriture suivante.**

**Les données MOT1 et MOT2 sont des informations de nature «électrique» (8 tensions)**

**Telles qu'elles sont codées, les informations MOT1 et MOT2 peuvent être**

**représentées par un nombre fini de nombres entiers constitués de 0 et de 1 : ce sont des informations numériques.**

**L'afficheur : la donnée MOT2 mémorisée est constituée de 8 tensions (0 ou +5V). Ces tensions permettent de commander un afficheur constitué de LEDS vertes.**



## 2- La conversion lumière → tension

Nous avons vu, auparavant, que MOT est commandé par l'état des LED. MOT n'est donc qu'une information optique.

Pour que cette information puisse être enregistrée par la mémoire il faut la transformer en une donnée MOT1 de nature électrique (8 tensions  $V_e$ ).

### Conception d'une voie :

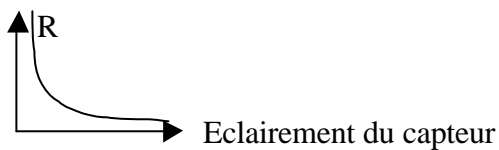
**Codage :** on veut obtenir le fonctionnement suivant

LED  $L_i$  allumée ( $L_i = 1$ )  $\rightarrow 3,5 < V_{Ei} < 5 \text{ V} \rightarrow E_i = L_i = 1$

LED  $L_i$  éteinte ( $L_i = 0$ )  $\rightarrow 0 < V_{Ei} < 0,8 \text{ V} \rightarrow E_i = L_i = 0$

### Première étape : la conversion lumière → Résistance.

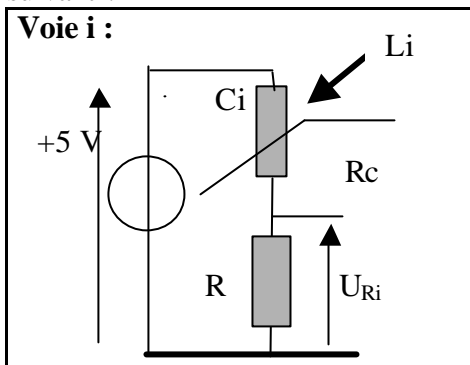
Pour transformer cette lumière en résistance nous avons utilisé des photo-résistances (CdS). On utilise une lentille pour que chaque capteur  $C_i$  soit éclairé par la LED ( $L_i$ ). La résistance du capteur diminue lorsqu'il est de plus en plus éclairé.



La réponse du capteur n'est pas linéaire, nous verrons plus loin que cela ne pose pas de problème pour réaliser notre projet.

### Deuxième étape : la conversion résistance → tension

Pour transformer la résistance du capteur en tension nous avons réalisé le diviseur de tension suivant :

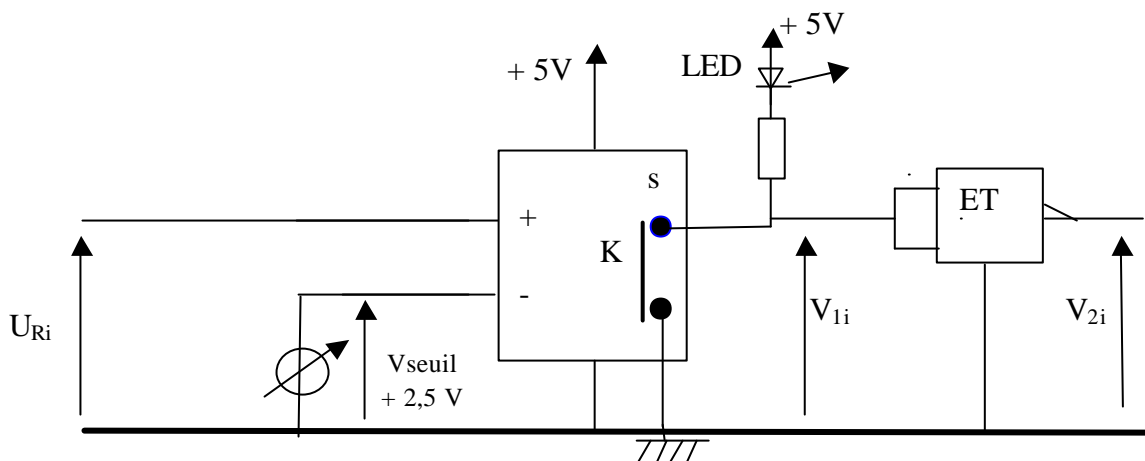


Lorsque le capteur  $C_i$  n'est pas éclairé, il se comporte comme une résistance infinie donc comme un interrupteur ouvert. Ce qui force un courant nul dans le circuit donc  $U_{Ri} = R_i = 0 \text{ V}$

Lorsque  $C_i$  est fortement éclairé, il se comporte comme une résistance presque nulle donc comme un interrupteur fermé.

$R_c = 0 \rightarrow U_{Ri} = 5 \text{ V}$

Nous avons constaté, lors des essais, que les tensions de sortie  $U_{Ri}$  n'étaient pas forcément comprises entre les marges  $0 \text{ V} \rightarrow 0,8 \text{ V}$  et  $3,5 \text{ V} \rightarrow 5 \text{ V}$ . Pour ne pas sortir des intervalles autorisés nous avons utilisé un comparateur à la sortie du diviseur de tension. Ce comparateur commande une LED qui est dans le même état que la LED  $L_i$ . Ce modèle est valable pour chaque voie.



Si  $L_i = 1 \rightarrow U_{Ri} \# +5V > 2,5V \rightarrow K \text{ fermé} \rightarrow \text{LED allumée} \rightarrow V_{1i} = 0 V$

Si  $L_i = 0 \rightarrow U_{Ri} \# 0V < 2,5V \rightarrow K \text{ ouvert} \rightarrow \text{LED éteinte} \rightarrow V_{1i} = +5V$

**Problème !** LED allumée  $\rightarrow V_{1i} = 0V$  et LED éteinte  $\rightarrow V_{1i} = +5V$  alors que nous voulions l'inverse. C'est pour cela que nous avons utilisé une porte NON-ET montée en inverseur. Grâce à cet inverseur, nous pouvons rétablir la logique : LED allumée  $\rightarrow V_{2i} = 5V$  et LED éteinte  $\rightarrow V_{2i} = 0V$

### Choix de la valeur de R du diviseur de tension et du seuil du comparateur : (voir ANNEXE)

L'influence de la lumière ambiante a été évaluée en mesurant la résistance des capteurs. Cela nous a conduits à protéger les capteurs avec un capot.

La résistance de chaque capteur  $C_i$  a été mesurée (capteur éclairé ou non).

Le fonctionnement du diviseur de tension a été simulé avec un tableur. On a recherché la valeur de R permettant d'obtenir la plus grande variation possible de  $U_{Ri}$  lorsqu'un capteur est éclairé ou non éclairé. On a retenu  $R = 1,8 \text{ k}\Omega$ . Les essais ont continué en mesurant  $U_{Ri}$  dans tous les cas de figure : dans quelques cas cette tension sortait des fenêtres imposées par le fonctionnement de la mémoire. On a donc été contraints d'utiliser un comparateur. Le seuil a été ajusté expérimentalement à 2,5 V. Finalement toutes les tensions  $V_{2i}$  sont au voisinage de 0 V ou de 5 V.

Nous pouvons donc dire que chaque voie fonctionne de la façon suivante :

$$L_i = 0 \rightarrow C_i \rightarrow V_{2i} = 0 V$$

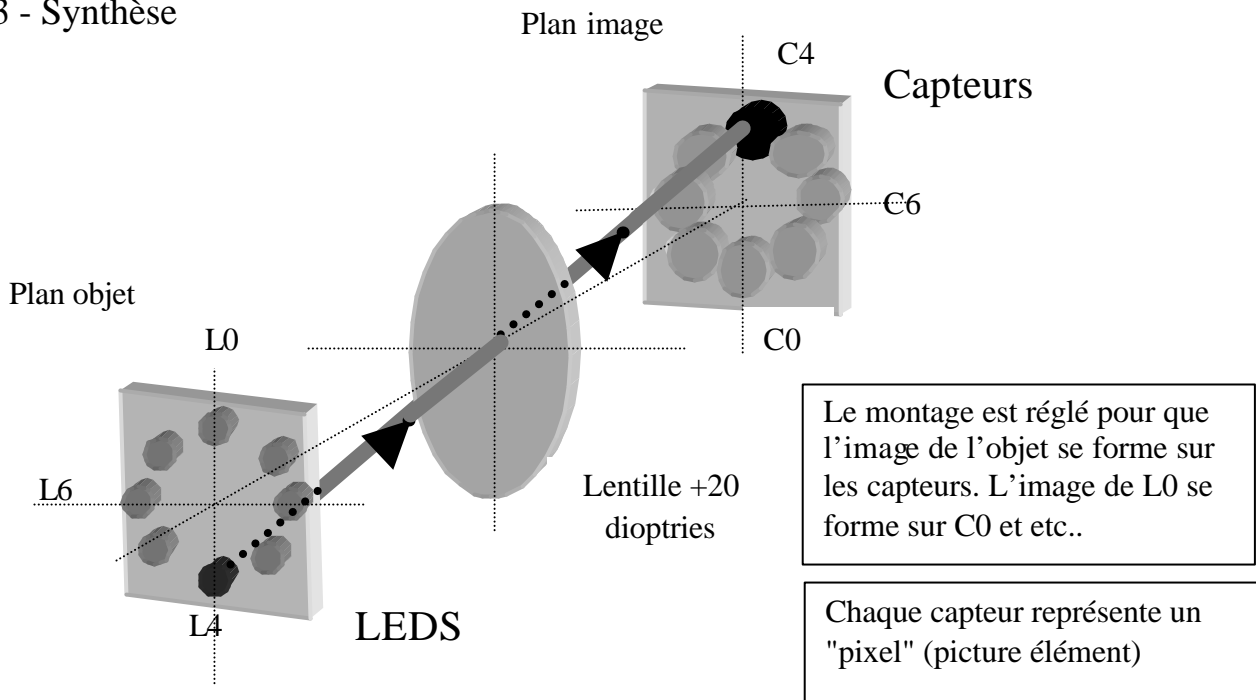
$$L_i = 1 \rightarrow C_i \rightarrow V_{2i} = 5 V$$

La tension  $u_{Ri}$ , à la sortie du diviseur de tension peut prendre une infinité de valeurs entre 0 et 5 V : c'est une tension **analogique**.

La tension  $V_{2i}$ , à la sortie du comparateur, ne peut prendre que les valeurs 0 V ou +5 V (nombre fini de valeurs) et elle peut être codée par deux états logiques 0 ou 1 : c'est une grandeur **numérique**.

Les comparateurs réalisent une Conversion Analogique-Numérique (CAN sur 1 bit).

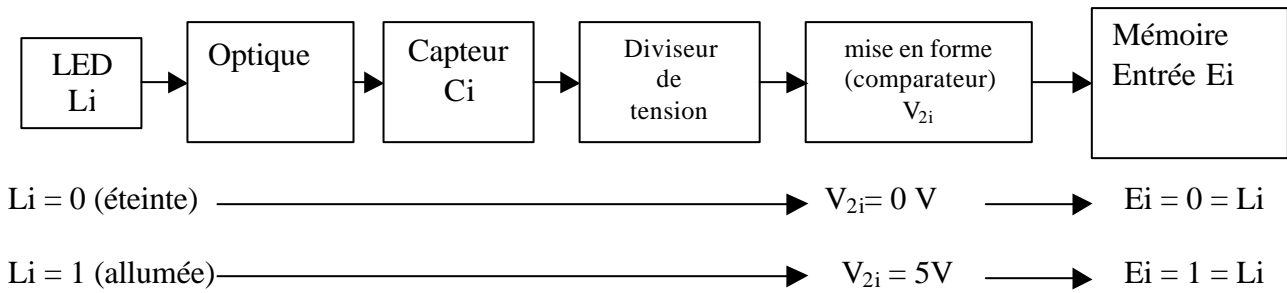
### 3 - Synthèse



Le plan « image » est donc découpé en 8 éléments (il est discrétisé) et chaque élément est occupé par un capteur. La «matrice» de capteurs correspond à 8 pixels (picture élément). L'optique est réglée pour que l'image d'une LED couvre la surface d'un capteur.

Le système est constitué de 8 voies identiques :

Les tensions  $V_{2i}$  (au format 0 V ou 5 V) sont reliées aux entrées  $E_i$  de la mémoire



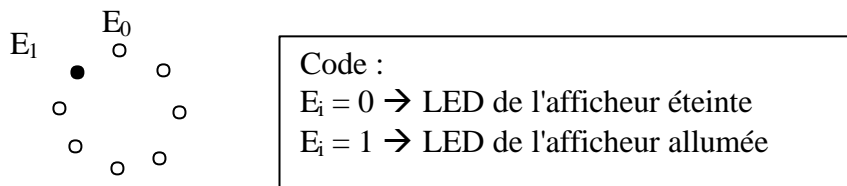
Les 8 données binaires  $E_i$  sont regroupées pour former la donnée  $MOT1 = E_7, E_6, \dots, E_0$

Nous avons MOT (une information optique qui représente la position de l'objet), nous l'avons convertie en MOT1 qui est une information électrique.

Ces codages nous conduisent à  $MOT1 = MOT$  : MOT1 est l'image de MOT donc une image de la position de l'objet.

**MOT1 pouvant être représenté par un nombre fini de valeurs (nombres entiers), MOT1 est une image numérique de l'état de l'objet.**

Sur la maquette "comparateurs", 8 LEDS rouges visualisent l'état de MOT1 et constituent un afficheur.



Il ne s'agit pas encore d'une photographie de l'objet : lorsque l'objet change de position,  $MOT1 = MOT$  varie.

#### 4- LA PHOTOGRAPHIE NUMERIQUE

**La position de l'objet (8 positions) a été transformée en son image MOT1(numérique).**

**Les 8 tensions (0 V ou 5 V) qui constituent MOT1 peuvent être présentées à l'entrée de la mémoire et être mémorisées lorsqu'on active le signal d'écriture.**

A l'instant  $t$ , on active le signal d'écriture de la mémoire.

On mémorise MOT1, on obtient  $MOT2 = (MOT1 \text{ à l'instant } t) = (MOT \text{ à l'instant } t)$ .

MOT2 est donc l'image de MOT1 à l'instant  $t$  qui est lui-même une image de MOT à l'instant  $t$ .

Donc si l'état de l'objet change, MOT change, MOT1 change mais MOT2 ne change pas.

On peut donc dire que **MOT2 est la photographie numérique de l'objet à l'instant  $t$ .**

## Deuxième partie : l'asservissement de position

Notre second objectif est d'utiliser la photographie numérique pour contrôler l'animation de l'objet. L'animation de l'objet doit être bloquée dès que la position de l'objet correspond à celle qui a été photographiée.

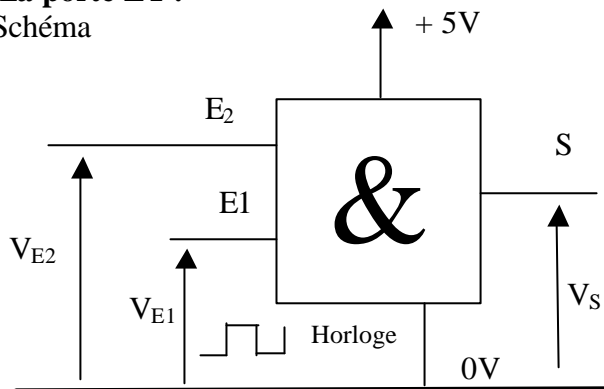
### I – LA COMMANDE DES LEDES

Les LEDES sont commandées par un boîtier de **commande des LEDES**.

Celui-ci comporte une porte ET.

#### La porte ET :

Schéma

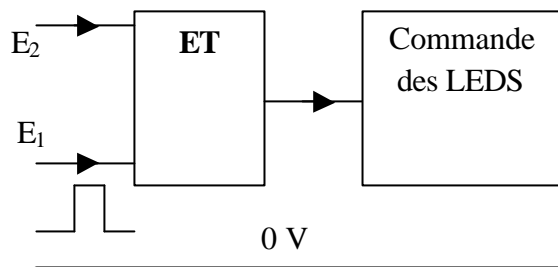


$E_1$  est relié a une horloge qui oscille entre 0 V ( $E_1= 1$  ) et +5V ( $E_1= 0$ ).

Si  $E_2 = 0$  ( $V_{E2} = 0$  V) alors  $S = 0$  quel que soit  $E_1$   
La porte est bloquée.

Si  $E_2 = 1$  ( $V_{E2} = +5$  V) alors  $S = E_1$   
La porte est "transparente" et l'on retrouve le signal d'horloge à la sortie S de la porte ET.

#### Schéma complet de la commande des LEDES :



Les LEDES sont commandées par le signal d'horloge.

Si la porte est ouverte ( $E_2 = 1$ ), la commande des LEDES reçoit le signal d'horloge et les LEDES s'allument dans le sens des aiguilles d'une montre. Si la porte est bloquée ( $E_2 = 0$ ), la commande des LEDES ne reçoit plus de signal d'horloge et l'objet lumineux reste dans l'état où il était. L'animation des LEDES est arrêtée.

L'entrée  $E_2$  ( $V_{E2} = 0$  ou 5 V) de la porte ET nous permet donc de contrôler la position de l'objet (entrée « asservissement »).

### 4- Le comparateur de MOTS

Nous souhaitons contrôler la position de l'objet lumineux. Nous pouvons photographier l'objet dans une position particulière : la valeur correspondante de MOT2 est mémorisée.

Lorsque l'objet est animé, la donnée MOT1, à la sortie des comparateurs varie et représente la position de l'objet.

On veut que l'objet soit **animé** si la position de l'objet est différente de celle qui correspond à la photographie :  $MOT1 \neq MOT2 \rightarrow E2 = 1$



On veut que l'animation soit bloquée si la position de l'objet correspond à la photographie :

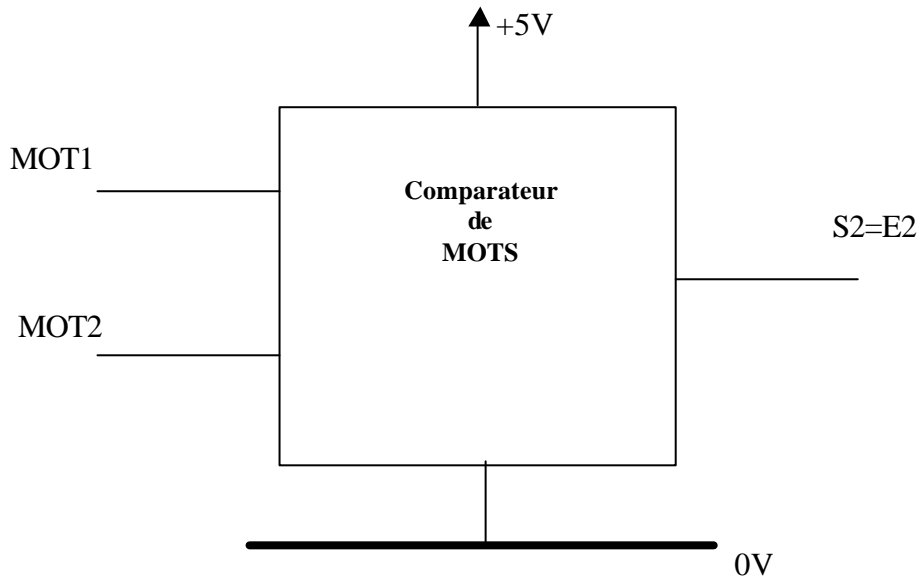
$$\text{MOT1} = \text{MOT2} \rightarrow \text{E2} = 0$$

Il faut donc réaliser la fonction suivante :

$$\text{MOT1} \neq \text{MOT2} \rightarrow \text{E2} = 1$$

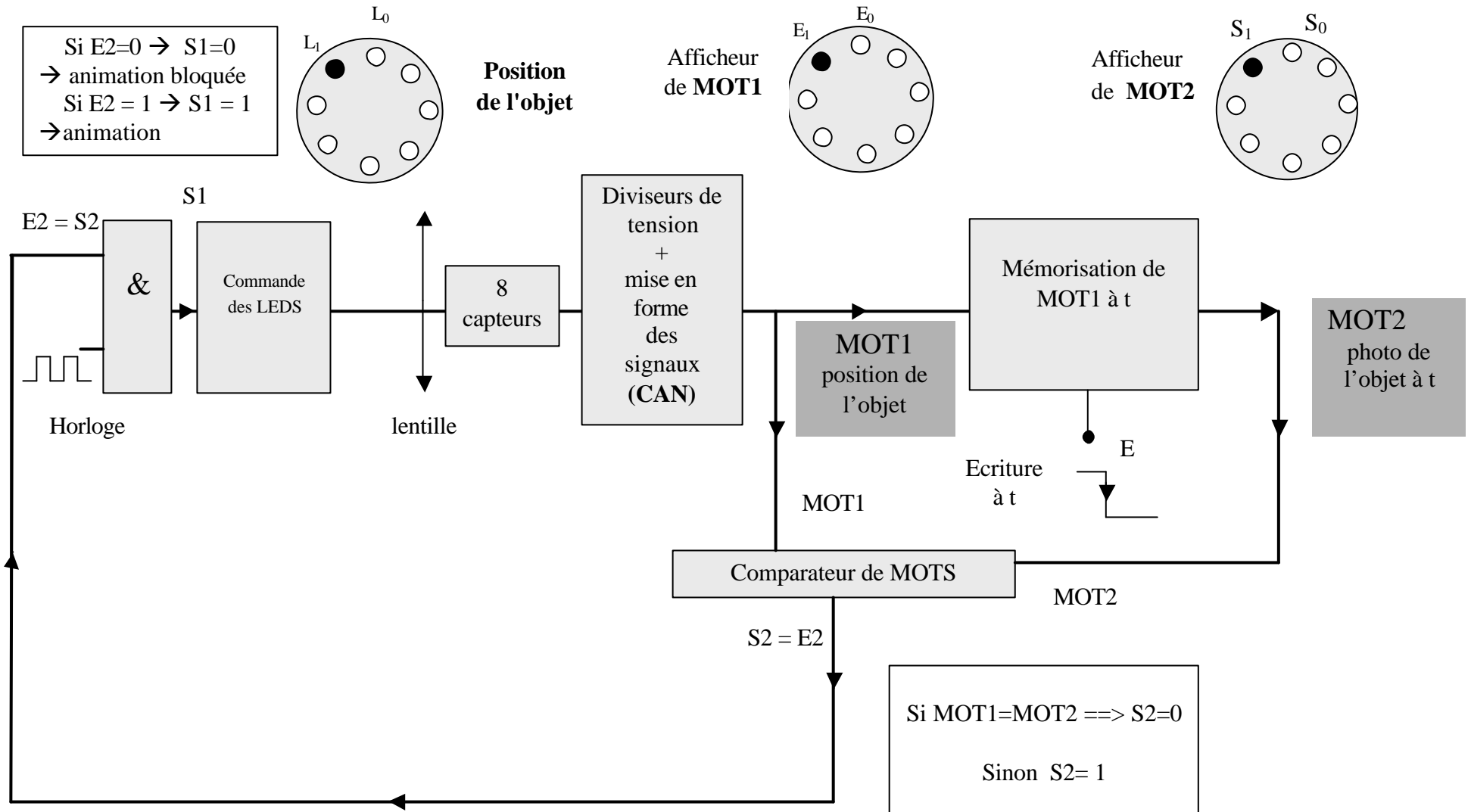
$$\text{MOT1} = \text{MOT2} \rightarrow \text{E2} = 0$$

Cette fonction est réalisée par un circuit intégré comparateur de MOTS :



La sortie S2 du comparateur de MOTS est reliée à l'entrée E2 de la porte ET.

## II- SCHEMA GLOBAL

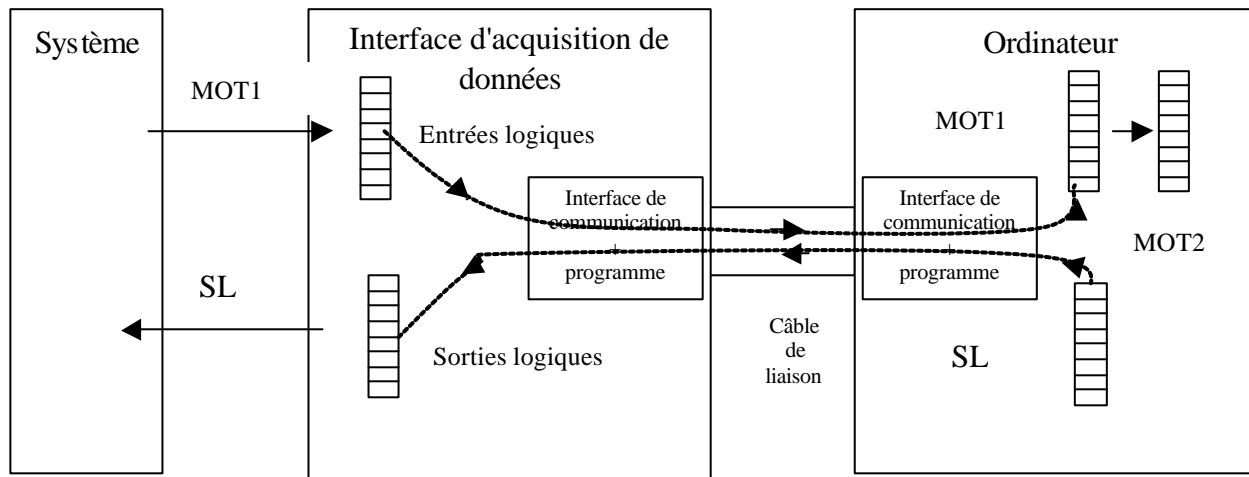


## Troisième partie : l'utilisation de l'ordinateur

*Pour l'instant toutes les fonctions utilisées ont été réalisées avec des composants électroniques. Nous allons voir comment certaines maquettes peuvent être supprimées, les fonctions correspondantes étant réalisées par un programme informatique et un interface de communication.*

**L'utilisation d'un interface de communication avec un ordinateur et le programme de photographie numérique.**

**I- L'interface de communication avec un ordinateur :**



L'interface de communication est équipée de deux mémoires de 8 bits. Ces mémoires fonctionnent avec les règles qui ont été définies plus haut : les grandeurs d'entrée ou de sortie sont 8 tensions "0 V ou + 5 V", c'est à dire des signaux "logiques" pouvant être représentés par les états 0 ou 1. La donnée mémorisée peut donc être représentée par un MOT de 8 bits.

Ces deux mémoires communiquent avec l'ordinateur grâce à une interface (RS232) et à un programme de communication.

Les "entrées logiques" fonctionnent de la façon suivante :

A leur entrée on présente la donnée MOT1 (8 tensions (0 V ou + 5 V) → 8 bits). Lorsque le programme utilisé par l'ordinateur exécute l'instruction "EL", la donnée MOT1 est écrite dans les entrées logiques puis transmise à l'ordinateur. Celui-ci écrit alors la donnée MOT1 dans sa mémoire.

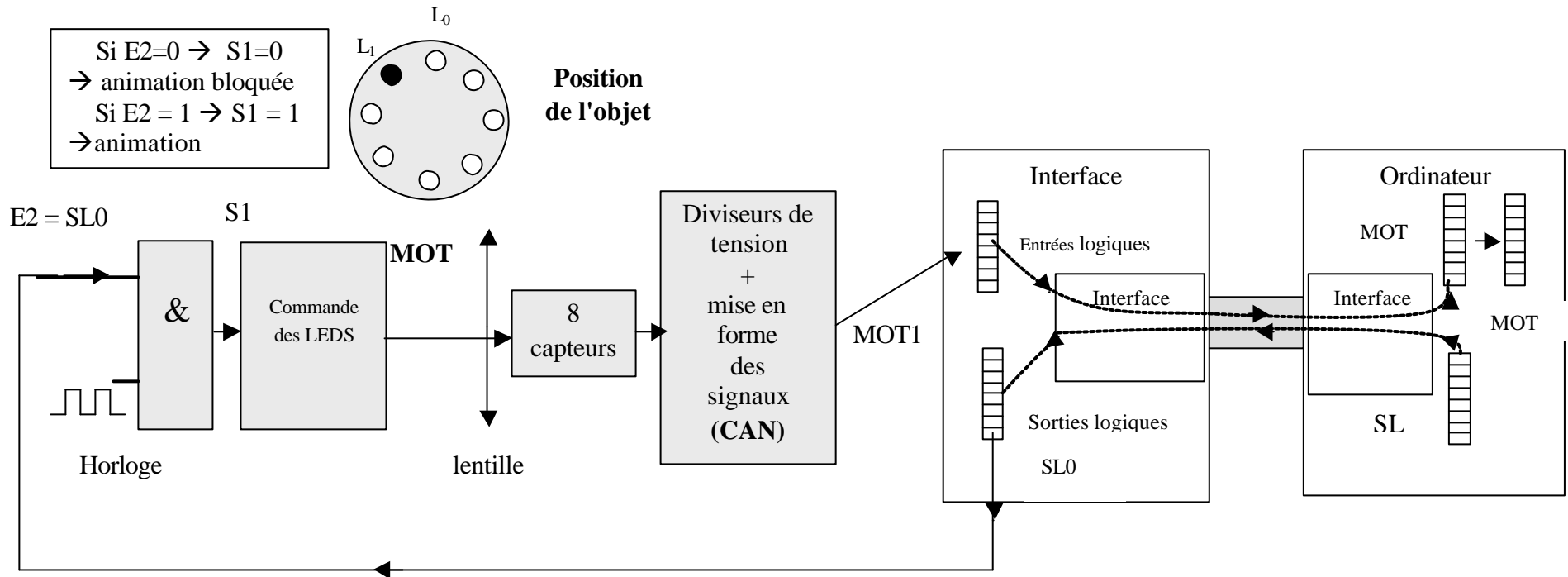
Les sorties logiques fonctionnent de la façon suivante :

Une donnée de 8 bits est écrite, par le programme, dans une ligne de mémoire de l'ordinateur. Lorsque le programme exécute l'instruction "SL", cette donnée est transmise à l'interface et écrite dans les "sorties logiques". Les 8 tensions "0 V ou + 5 V" correspondantes peuvent être utilisées par le système.

**Le programme et l'interface, permettent d'échanger des informations "numériques" entre le système physique et l'ordinateur.**

## 2- Le programme de photographie numérique

Dans l'application précédente, on peut supprimer la carte "mémoire" et la carte comparateur de MOTS. L'interface et le programme de communication réaliseront ces fonctions.

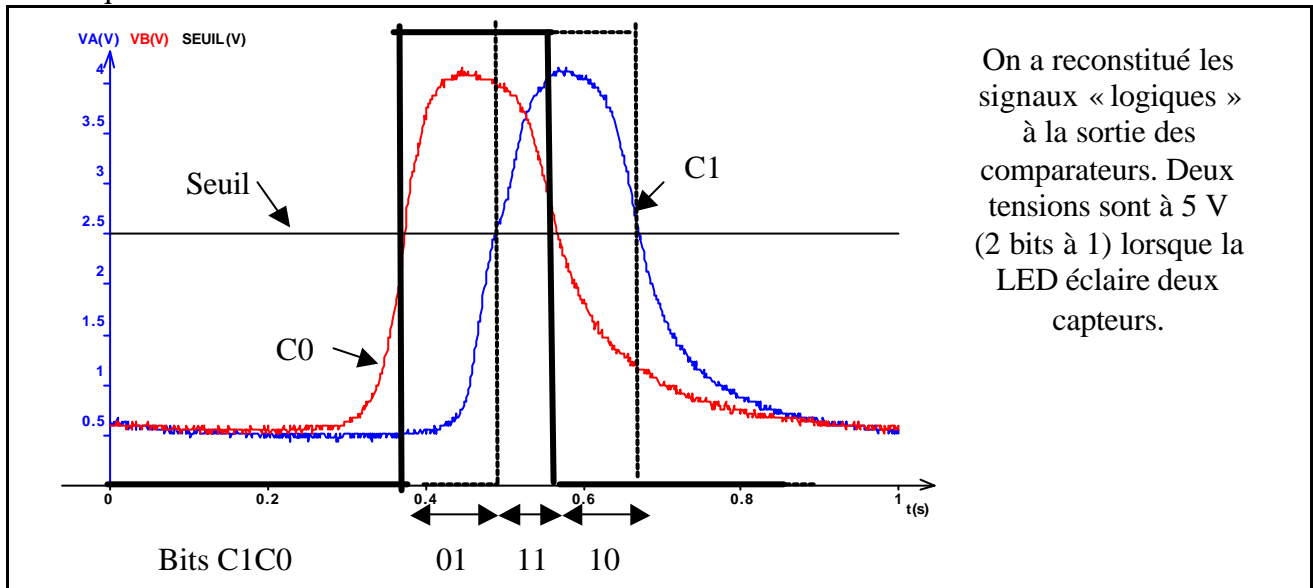


La donnée MOT1, à la sortie de la carte "comparateurs", est constituée de 8 tensions "0 ou + 5 V": elle peut donc être transmise aux entrées logiques. La sortie SL0 des sorties logiques est une tension « 0 ou + 5 V » (état logique de SL0 : 0 ou 1) : elle est reliée à l'entrée E2 de la porte ET. Le programme exécute l'instruction « SL » et écrit  $SL0 = 1$  ( $SL = 1$ ) : l'objet est donc animé. Le programme répète ensuite en permanence l'exécution de l'instruction « EL » : la donnée MOT1 est transmise à l'ordinateur et le programme la décode pour afficher une image identique à l'état de l'objet. En « cliquant » sur un bouton, l'utilisateur du programme peut commander la mémorisation de MOT1 à une date  $t$  (donnée MOT2). La donnée MOT2 est écrite dans une autre ligne de mémoire de l'ordinateur, elle est ensuite décodée et l'état de l'objet correspondant est affiché. Le programme compare alors les données MOT1 et MOT2 : si  $MOT1 = MOT2$ , alors il exécute l'instruction « SL » et écrit  $SL0 = 0$ , l'animation de l'objet est bloquée, sinon il écrit  $SL0 = 1$  et l'objet est animé.

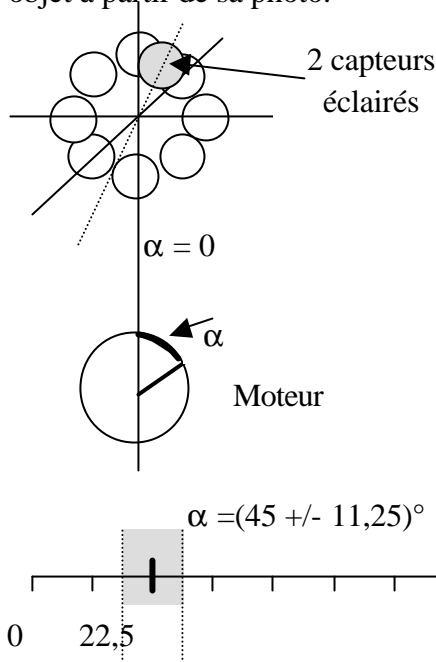
## Quatrième partie : la précision du repérage de la position de l'objet

Le système a été mis au point avec le simulateur. On utilise maintenant l'objet (une LED) animé par un moteur. Celui-ci peut prendre une infinité de positions et nous ne disposons que de huit capteurs.

On peut donc repérer 8 positions de l'objet. En fait, on constate que, quelquefois, deux bits de MOT1 sont à 1 : cela est dû au fait que deux capteurs sont éclairés en même temps et que deux tensions sont, simultanément, supérieures au seuil de 2,5 V des comparateurs. On a enregistré la variation de la tension à la sortie de deux diviseurs de tension. Elles correspondent aux capteurs C0 et C1 qui «se touchent».



On récupère donc des informations supplémentaires qui permettent de mieux définir la position de l'objet à partir de sa photo.



Lorsqu'un seul capteur est éclairé, 1 bit de MOT1 est à 1, les autres à 0 : on repère 8 positions de la LED.

Le pas entre deux positions est de  $360/8 = 45^\circ$ .

Lorsque 2 capteurs sont éclairés, 2 bits de MOT1 sont à 1, les autres à 0. On repère ainsi 8 positions supplémentaires de la LED, on peut donc repérer 16 positions avec un pas de  $360/16 = 22,5^\circ$ .

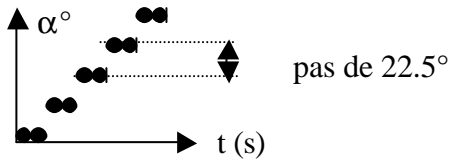
La position de l'axe du moteur peut être repérée par la valeur de l'angle  $\alpha$ . Cette grandeur peut prendre une infinité de valeurs : c'est une grandeur analogique.

Nous ne pouvons repérer que 16 valeurs de  $\alpha$  avec un pas de  $22,5^\circ$  : beaucoup d'informations ont été perdues.

Pour chaque valeur de MOT1, on peut calculer la valeur de la position  $\alpha$ . Cette position est définie avec une incertitude absolue de  $11,25^\circ$ .

La donnée MOT1 (image de la position de l'objet) est transmise périodiquement à l'ordinateur. Nous avons modifié le programme pour qu'il affiche la valeur de  $\alpha$  en fonction du temps.

On obtient une courbe ayant l'allure suivante :



La valeur de  $\alpha$  peut être représentée par  $\alpha = q * n$  (nombre entier), ce nombre entier variant de 0 à 15 (sans unité). La grandeur  $\alpha$  est **quantifiée**, elle ne peut prendre qu'un **nombre fini** de valeurs (16).  $q = 22,5^\circ$  est le pas de la quantification (le quantum) : ce nombre permet de convertir le nombre entier sans unité en une grandeur physique  $\alpha$  (en degrés).

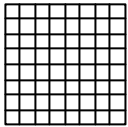
Le système effectue une mesure de  $\alpha$  tous les  $\Delta t$  (période d'échantillonnage) :  $\alpha$  est une grandeur **échantillonnée**.

$\alpha$  étant une grandeur **quantifiée** et **échantillonnée**, c'est une grandeur **numérique**.

Notre instrument permet de repérer la position du moteur à  $11,25^\circ$  près : sa **résolution** est médiocre. Cela est dû au faible nombre de capteurs utilisés. Si, sur un tour, on utilisait 16 capteurs, on aurait une résolution d'environ  $6^\circ$ .

Pour augmenter la résolution de notre instrument, il faudrait diminuer le pas  $\Delta\alpha$  entre deux capteurs, c'est à dire augmenter le nombre de capteurs sur un tour, donc augmenter le nombre de pixels. Chaque pixel étant représenté par 1 bit dans la mémoire, il faudra aussi augmenter le nombre de comparateurs et la capacité de la mémoire.

**Comment marchent les « vrais » appareils photo numériques ?**



L'écran sur lequel se forme l'image est constitué d'une matrice de capteurs (pixels). Dans les appareils actuels il y a au moins  $N = 4$  Mpixels.

Les capteurs utilisés sont des photodiodes qui débitent un courant proportionnel à la puissance lumineuse reçue par le capteur (capteurs linéaires).

Ces  $N$  courants sont convertis en  $N$  tensions. Chaque tension est ensuite convertie en un MOT binaire de 8 bits (1 octet) à l'aide d'un **Convertisseur Analogique Numérique (CAN)**. Chaque donnée MOT permet de coder l'éclairement d'un capteur sur 256 niveaux de gris. Lorsque l'on prend une photo, les  $N$  données MOT sont écrites dans la mémoire. Avec 4 Mpixels, il faut une mémoire de 4 Moctets.

Si on veut une photo en couleur chaque pixel est constitué de 3 capteurs équipés d'un filtre optique (rouge, vert, bleu). On prend donc 3 photos correspondant à ces couleurs. En superposant ces images on reconstitue l'état de l'objet en couleur. Avec 4 Mpixels, il faudra une mémoire de 12 Moctets. On comprend pourquoi les données numériques sont « comprimées » par des algorithmes. Les appareils photos numériques sont « bourrés » d'informatique.

## ANNEXE

### I- OBTENTION DE L'IMAGE D'UNE LED SUR UN CAPTEUR

Pour former l'image des LEDS sur les capteurs nous disposons d'une lentille de 20 dioptries et de son support. Les LEDS, soudées sur un circuit imprimé, forment un cercle de diamètre 1,7 cm (dimension de l'objet).

Les capteurs (photorésistances) soudés sur un circuit imprimé, forment un cercle de diamètre 3,6 cm (c'est la dimension de l'image, puisque la lumière émise par la LED Li doit converger sur le capteur  $C_i$ ).

Le grandissement du système est donc  $\gamma = |p'| / |p| = \text{image/objet} = 3,6/1,7 = 2,1$ .

D'autre part  $1/|p| + 1/|p'| = 1 / |f| = 20 \text{ m}^{-1}$

La résolution de ces deux équations conduit à  $|p'| = 15,5 \text{ cm}$  et  $|p| = 7,4 \text{ cm}$ .

On réalise alors un "banc optique" en bois sur lequel sont fixés tous les éléments (les positions de l'objet et de l'image sont réglables).

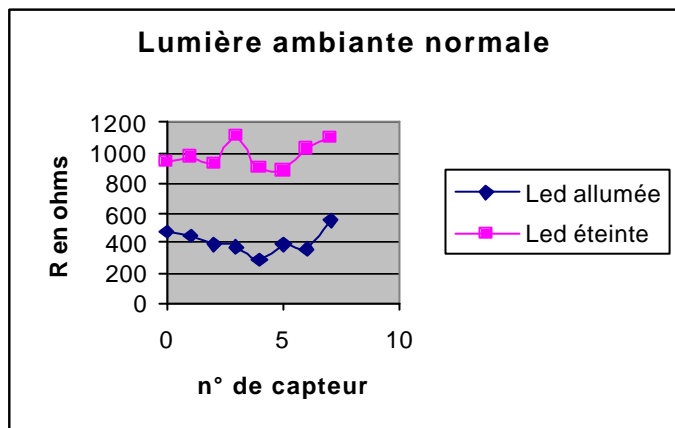
Finalement, les positions sont réglées pour que l'image de chaque LED se forme sur le capteur correspondant.

### II - INFLUENCE DE LA LUMIERE AMBIANTE SUR LA RESISTANCE DES CAPTEURS.

Chaque LED éclaire un capteur. Il faut transformer l'état (allumée, éteinte) d'une LED en une valeur de la résistance  $R_C$  du capteur. Afin de distinguer clairement l'état de la LED, la différence ( $R_{\text{cmax}} - R_{\text{cmin}}$ ) doit être aussi grande que possible. Au départ les capteurs ne sont pas protégés de la lumière ambiante.

#### Capteurs non protégés de la lumière ambiante

n° capteur	$R_C$ en ohms	
	Led Li allumée	Led Li éteinte
0	470	933
1	448	963
2	388	923
3	383	1119
4	290	895
5	393	885
6	356	1020
7	546	1098

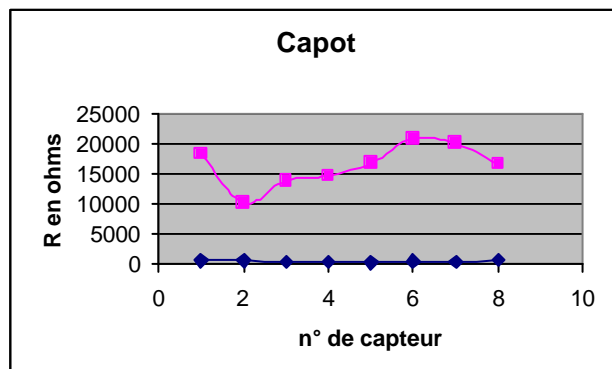


La différence ( $R_{\text{cmax}} - R_{\text{cmin}}$ ) pour les leds allumées ou éteintes est de l'ordre de 500 ohms.

Nous recommençons nos mesures après la confection d'un capot qui isole les capteurs de la lumière ambiante.

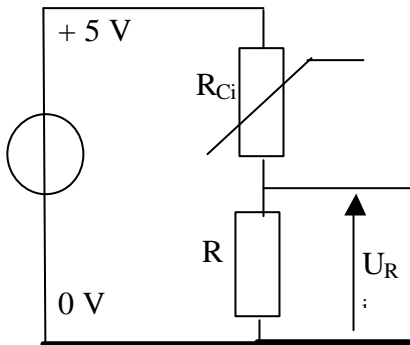
Avec capot

n° capteur	$R_C$ en ohms	
	Leds allumées	Leds Li éteintes
0	665	18200
1	626	10170
2	473	13890
3	443	14640
4	341	16710
5	517	20830
6	436	20050
7	748	16590



Avec le capot la différence entre les valeurs maximales et minimales de  $R_c$  est de l'ordre de 15000 ohms. L'utilisation d'un capot permet de gagner un facteur 30 sur la différence ( $R_{cmax} - R_{cmin}$ ). Les expériences continueront donc avec le capot de protection.

### III - CHOIX DE LA VALEUR DE R DU DIVISEUR DE TENSION



On veut que, selon que le capteur est éclairé ( $R_{Ci}$  faible) ou non ( $R_{Ci}$  forte), la différence entre  $U_{Ri\max}$  et  $U_{Ri\min}$  soit la plus grande possible.

D'après les mesures, on sait que  $R_{Ci}$  varie en moyenne de  $R_{Cmin} = 500\text{ohms}$  à  $R_{cmax} = 15000\text{ohms}$ .

Une valeur de  $R$  convenable est déterminée en effectuant une simulation avec un tableur.

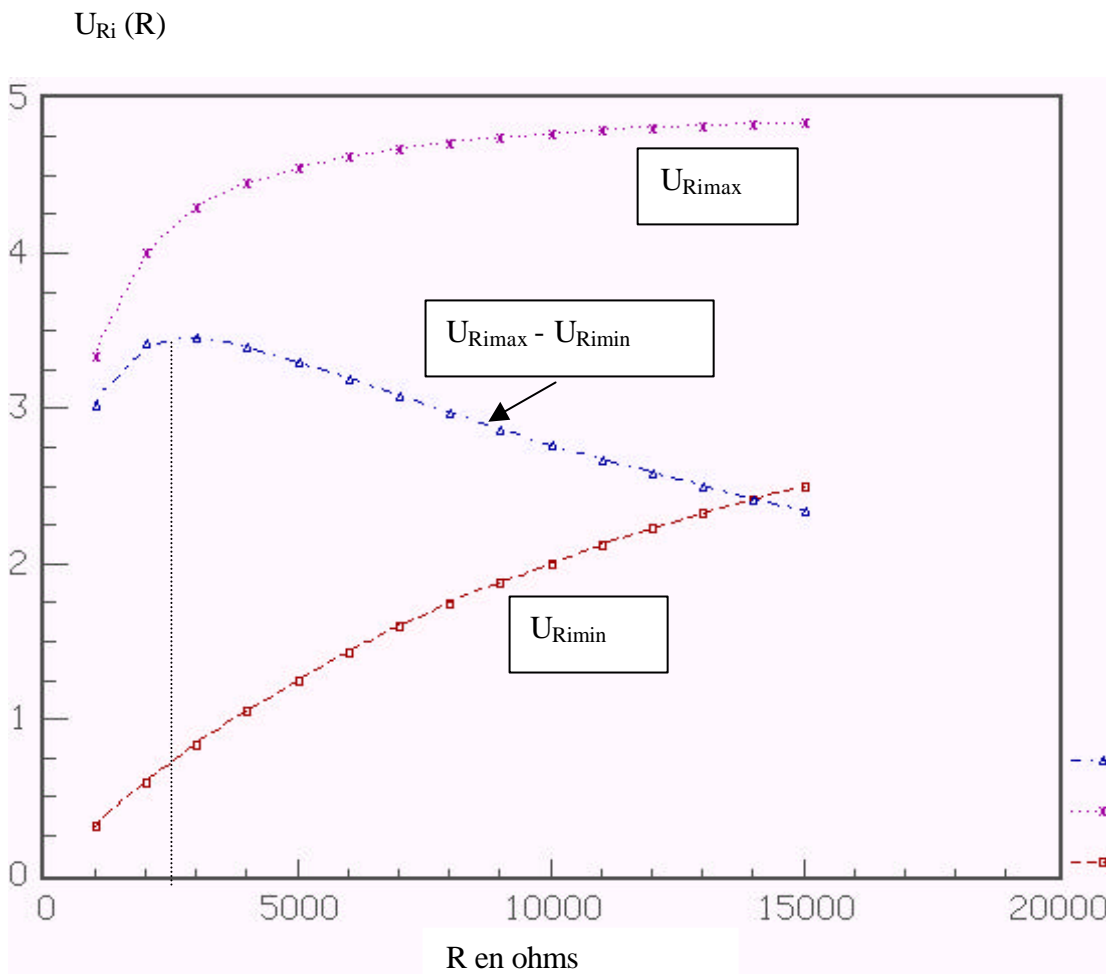
On fera les calculs de la tension  $U_{Ri}$  avec  $R_{Cmin} = 500\text{ohms}$  et  $R_{Cmax} = 15000\text{ohms}$ .

Calculs :

$$U_{Ri} = \frac{R \cdot 5}{R + R_{Ci}} = \frac{(R \cdot 5)/R}{(R + R_{Ci})/R} = \frac{5}{1 + (R_{Ci}/R)}$$

Donc  $U_{Ri\max} = 5/[1 + (R_{Cmin}/R)]$  et  $U_{Ri\min} = 5/[1 + (R_{Cmax}/R)]$

On trace  $U_{Ri\max} = f(R)$   $U_{Ri\min} = f(R)$  et  $(U_{Ri\max} - U_{Ri\min}) = f(R)$



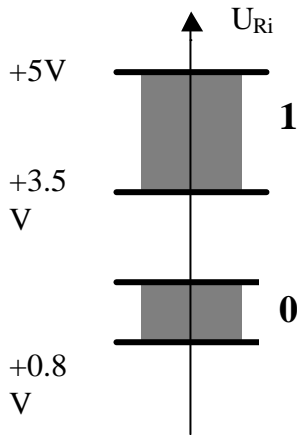


Conclusion :

La différence ( $U_{Ri,max} - U_{Ri,min}$ ) est **maximum** lorsque R est aux environs de  $2000 \Omega$ .

Après les essais sur plusieurs maquettes, on retient une valeur de  $R = 1,8 k\Omega$ .

**Contrôle du fonctionnement du diviseur de tension :**



Les tensions  $U_{Ri}$  doivent être mémorisées. Pour cela elles doivent appartenir aux intervalles définis sur le schéma.

On éclaire successivement chaque capteur et l'on mesure les 8 tensions  $U_{Ri}$  afin de vérifier si les tensions appartiennent aux intervalles imposés.

N° C	$U_{R0}$ en V	$U_{R1}$ en V	$U_{R2}$ en V	$U_{R3}$ en V	$U_{R4}$ en V	$U_{R5}$ en V	$U_{R6}$ en V	$U_{R7}$ en V
éclairé								
c0	3,68	0,66	0,19	0,16	0,25	0,33	0,4	0,4
c1	0,34	3,75	0,55	0,23	0,3	0,34	0,12	0,13
c2	0,08	0,62	4,02	0,57	0,35	0,22	0,13	0,13
c3	0,07	0,16	1,27	4,04	0,5	0,2	0,14	0,13
c4	0,07	0,14	0,17	1,9	4,23	0,5	0,24	0,15
c5	0,08	0,16	0,14	0,25	1,44	3,9	0,47	0,16
c6	0,09	0,17	0,11	0,21	0,2	0,87	4,06	0,4
c7	0,32	0,18	0,13	0,18	0,16	0,22	0,7	3,53

Suite à notre tableau de mesure, nous observons que 4 valeurs sortent des intervalles imposés.

Nous avons donc été contraints d'utiliser un comparateur qui nous permettra de séparer clairement  $U_{Ri,min}$  de  $U_{Ri,max}$ .

Nous faisons un essai des 8 comparateurs avec un seuil de 2,5V.

On mesure les tensions  $V_{2i}$  à la sortie des inverseurs :

Dans tous les cas LED  $L_i$  allumée  $\rightarrow V_{2i} \neq +5 V$  et LED  $L_i$  éteinte  $\rightarrow V_{2i} \neq 0 V$

**Le seuil des comparateurs sera donc fixé à 2,5V.**

Les tensions  $V_{2i}$  peuvent donc être appliquées à l'entrée de la mémoire.

**Dans ces conditions  $L_i = 0$  (LED éteinte)  $\rightarrow E_i = 0$  et  $L_i = 1$  (LED allumée)  $\rightarrow E_i = 1$ .**

## REMERCIEMENTS

Nous tenons à remercier Monsieur François WENDLING, Directeur Général de l'ARATEM de VALENCE (Agence Rhône Alpes pour la maîtrise des TEchnologies de Mesure ) qui nous a mis en contact avec la société Alliance Vision (Montélimar) spécialisée dans les applications industrielles de la vision artificielle.

Nous remercions aussi très chaleureusement Monsieur Eric AZIBI, de la société Alliance Vision.

Il nous a reçus dans son entreprise pour nous expliquer comment sont conçues les applications industrielles de la vision artificielle et nous a donné de précieux conseils.