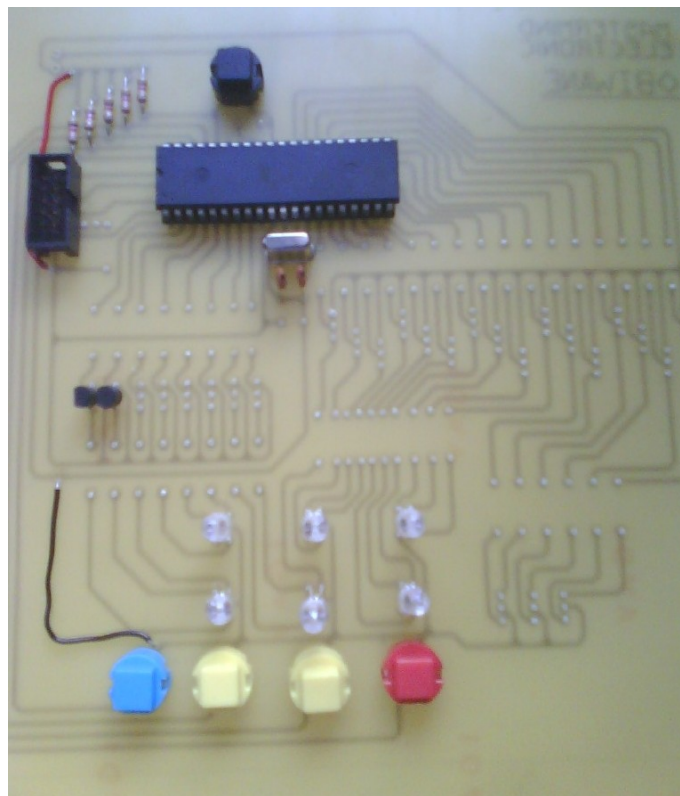


## Étude et Réalisation: *MasterMind Électronique*



Université François-Rabelais de Tours  
Institut Universitaire de Technologie de Tours  
Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS  
TOURS



Institut Universitaire de Technologie

Département  
GENIE ELECTRIQUE ET  
INFORMATIQUE INDUSTRIELLE

**Étude et Réalisation:**  
***MasterMind Électronique***

BESNARD Julien  
2 ème année - P2  
2009-2010

Enseignants  
M LEQUEU  
Mme AUGER

# Sommaire

Introduction.....	4
1. Présentation du projet.....	5
1.1. Règle du jeu de société.....	5
1.2. Une partie sur le MasterMind électronique .....	6
1.3. Cahier des charges.....	7
1.4. Planning de travail.....	7
2. Étude Générale de la solution technique proposée.....	8
2.1. But de l'étude.....	8
2.2. Schéma fonctionnel.....	8
2.3. Analyse fonctionnelle.....	8
2.4. Schéma électronique de principe.....	9
3. Étude de la partie électronique.....	10
3.1. Étude de la commande par bouton-poussoir FP1.....	10
3.2. Étude des commandes à transistors FP3.....	10
3.3. Étude des diodes multicolores FP4.....	12
4. Programmation de l'ATmega 8535, FP2.....	15
4.1. Présentation du programme.....	15
4.2. Génération du code couleurs.....	15
4.3. Choix des couleurs à tester.....	16
5. Poursuite du projet.....	17
5.1. Travail à faire.....	17
5.2. Piste pour l'avenir.....	17
Conclusion.....	18
Index des illustrations.....	19
Bibliographie.....	20

# Introduction

Dans le cadre du cours d'étude et réalisation du semestre 4, il était demandé de réaliser un projet en électronique associant une partie programmation au cahier des charges.

Mon choix s'est porté sur la réalisation d'un MasterMind électronique car il s'agit d'un jeu que j'apprécie et qui je pense pouvait apporter des perspectives de travail intéressantes en accord avec la formation GEII.

# 1. Présentation du projet

## 1.1. Règle du jeu de société

Avant de démarrer la partie, un premier joueur va placer ses pions de couleurs (blanc, noir, rouge, bleu, vert ou jaune) derrière un cache pour éviter que le second joueur ne voit les pions choisis.

Le but est de retrouver quels sont les 4 pions choisis par l'adversaire et d'en connaître les positions.

Pour cela, à chaque tour, le second joueur doit se servir des pions de couleurs pour former une rangée et se rapprocher le plus possible de la solution (ou de la trouver) car le nombre de rangées pour trouver la solution est limité. Une fois les pions placés, le premier joueur compara le code proposé au code qu'il a dissimulé et devra indiquer :

- le nombre de pions de bonne couleur bien placés en utilisant les petits pions rouges;
- le nombre de pions de bonne couleur mais mal placés avec les petits pions blancs.

En aucun cas les pions bien placés sont indiqués.

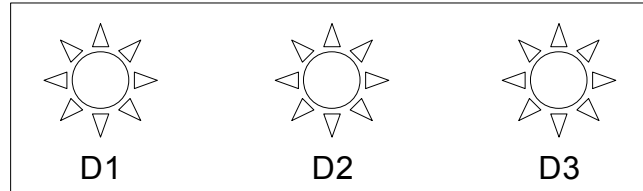


*Illustration 1: jeu de MasterMind original*

## 1.2. Une partie sur le MasterMind électronique

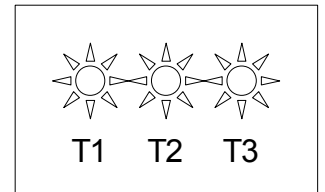
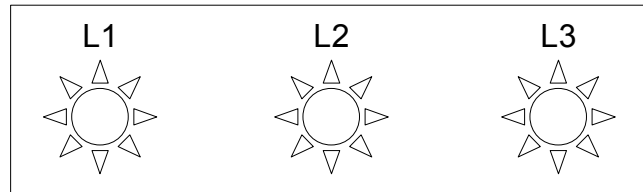
Voici ce à quoi ressemble le MasterMind électronique de façon schématique.

Diodes multicolores visualisant le code à deviner



Bouton poussoir  
génération de  
code

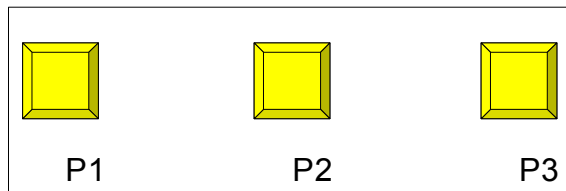
Diodes multicolores utilisées par le joueur pour proposer  
un code



Bouton  
poussoir  
validation  
de code



P4



Boutons poussoirs choix de couleur

Diodes tricolores qui  
montrent l'avancées de  
la partie

Illustration 2: schéma de principe du MasterMind électronique

La partie commence en appuyant sur le bouton « bp génération de code ». Un code de couleurs est alors généré et est envoyé sur la rangée de diodes multicolores, appelée D, qui est dissimulée sous un cache.

Après que le code est été généré, le joueur propose des codes en faisant défiler les couleurs sur les diodes multicolores non cachées à l'aide des boutons « bp choix couleurs ».

Une fois que le joueur est satisfait du code qu'il compte proposer, il appuie sur le bouton « bp validation de code » pour soumettre sa proposition à la machine.

Celle-ci le compare au code généré précédemment et indique au joueur, par le biais de diodes tricolores, les indications suivantes:

- rouge : la couleur n'est ni bonne ni au bon endroit;
- orange : la couleur est bonne mais pas au bon endroit;
- vert : la couleur est bonne et bien placée.

La partie s'achève lorsque les trois diodes tricolores sont vertes.

### 1.3. Cahier des charges

Le but étant de créer un MasterMind électronique utilisable n'importe où et accessible à tous, les contraintes à respecter sont les suivantes :

- créer un MasterMind électronique qui respecte le principe du jeu de société;
- programmation (ATmega 8535);
- réaliser une carte avec alimentation autonome;
- pouvoir programmer l'ATmega directement sur la carte;
- solution visuelle de communication homme/machine;
- coût et encombrement réduit : doit être abordable.

### 1.4. Planning de travail

Séance	3	4	5	6	7	8	9	10	11	12	13
prise de connaissance du sujet / récupération du travail effectuer	■			■	■						
étude de l'atmega		■		■	■						
Réalisation du programme			■	■	■	■	■	■			
Étude et réalisation de la carte				■	■		■	■		■	■
Test sur carte / finalisation du projet				■	■				■	■	■
présentation orale				■	■						■

planning prévisionnel : ■

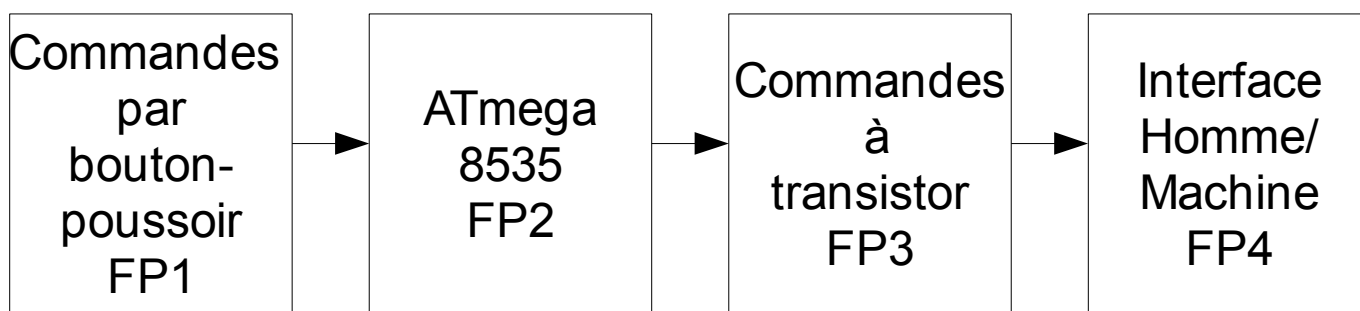
planning réel : ■

## 2. Étude Générale de la solution technique proposée

### 2.1. But de l'étude

Dans cette partie je vais présenter les différentes fonctions du système. Puis, dans les parties 3 et 4, je présenterai une étude de chacune des fonctions.

### 2.2. Schéma fonctionnel



Dessin 1: schéma fonctionnel de niveau 2

### 2.3. Analyse fonctionnelle

- Commandes par bouton-poussoir FP1

Il s'agit des boutons qui vont permettre de jouer. L'étude est faite dans la partie 3.

- ATmega8535 FP2

L'ATmega permettra de générer le code à deviner, de comparer le code proposé par le joueur au code généré ainsi qu'à informer le joueur sur l'avancée de la partie.

L'ATmega ne dispose pas d'assez de sorties pour commander le nombre de diodes nécessaire pour correspondre au jeu réel, c'est pourquoi le code à chercher n'utilisera que 3 diodes.

L'étude du programme est faite dans la partie 4.

- Commandes à transistor FP3

L'ATmega ne délivrant pas assez de tension en sortie pour commander les DEL, cette commande se fera par l'intermédiaire de transistors. L'étude est faite dans la partie 3.

- Interface homme/machine FP4.

L'interface homme/machine est géré par les diodes multicolores. L'étude est faite dans la partie 3.



## 2.4. Schéma électronique de principe

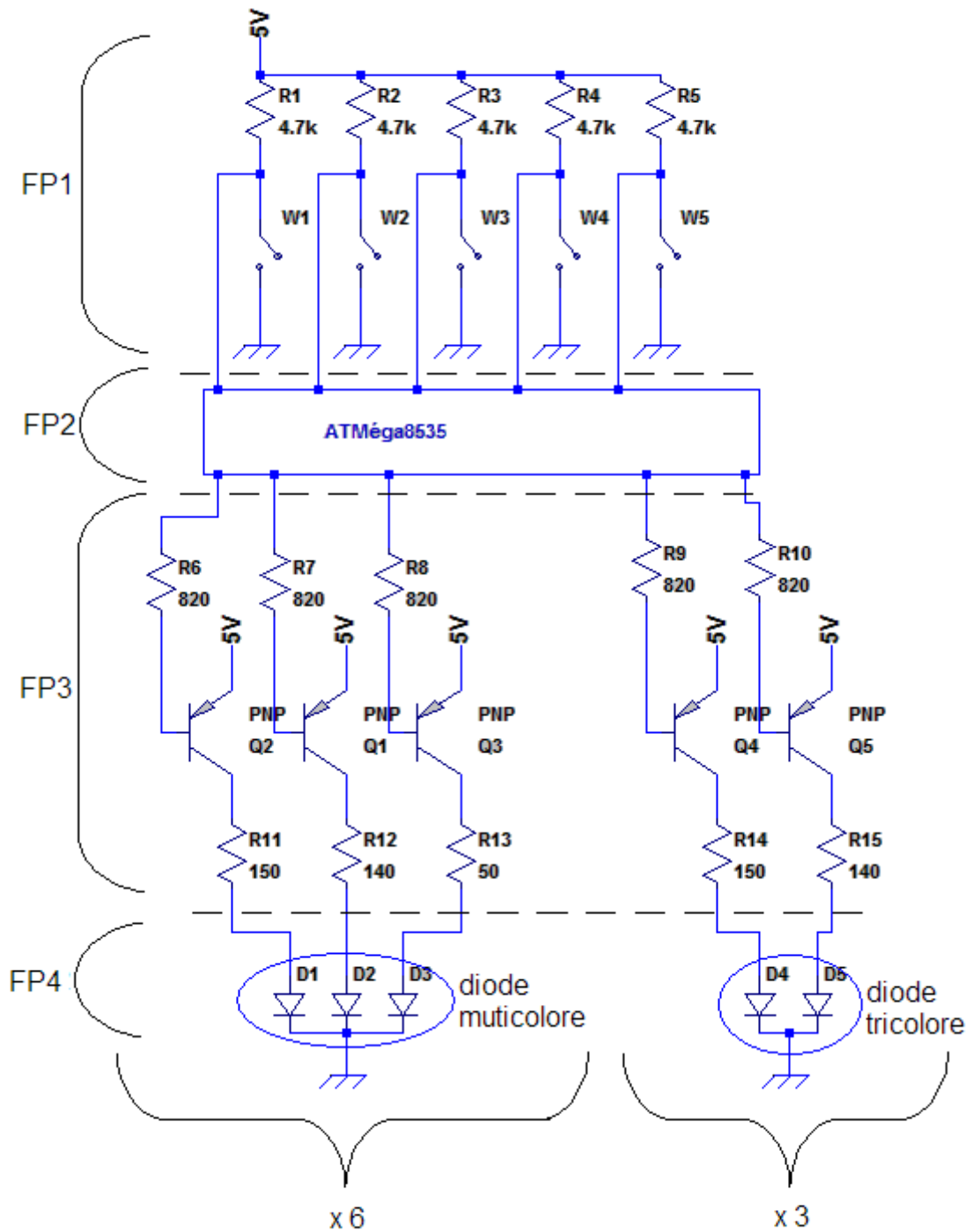


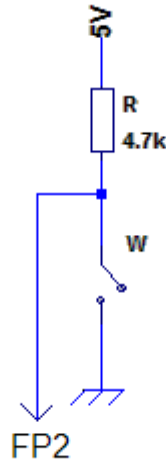
Illustration 3: schéma de principes

Sur ce schéma, W1 à 5 représentent les boutons poussoirs et Q1 à 5 représentent les transistors PNP.

## 3. Étude de la partie électronique

### 3.1. Étude de la commande par bouton-poussoir FP1

#### 3.1.1. Principe de fonctionnement



*Illustration 4: schéma structurel de FP1*

Le principe est simple, lorsque le bouton poussoir n'est pas actionné, l'entrée de l'ATmega est reliée au 5V. Et lorsque le bouton poussoir est actionné, le système est court-circuité et l'ATMÉga n'est plus alimenté.

#### 3.1.2. Conclusion de l'étude

L'appui sur le bouton poussoir correspond donc à un niveau logique 0 pour l'ATmega.

### 3.2. Étude des commandes à transistors FP3

#### 3.2.1. But de l'étude

Le but de cette étude est de rappeler le principe de fonctionnement d'une commande à transistor de type PNP.

### 3.2.2. Principe de fonctionnement

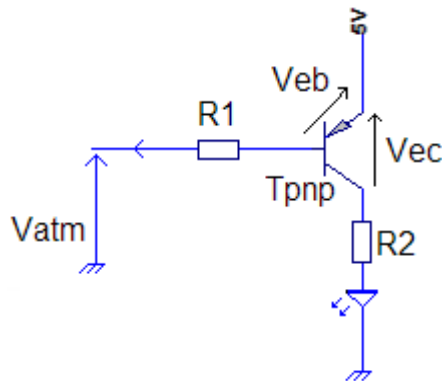


Illustration 5: montage à transistor PNP

Le transistor a deux modes de fonctionnements dans ce montage : bloqué et passant. Le passage de l'un à l'autre se fait en fonction de la tension entre l'émetteur et la base  $V_{eb}$ .

Lorsque  $V_{eb} = 0V$ , le transistor est bloqué, ce qui équivaut à un circuit ouvert entre l'émetteur et le collecteur. La DEL est donc éteinte.

Lorsque  $V_{eb} = 0,7V$ , le transistor est saturé, c'est à dire que la DEL vas s'allumer.

Dans le cas présent on veut que le transistor soit saturé lorsque la sortie de l'ATMéga soit à 0. Lorsque la sortie est à 0, La tension sur la patte de l'ATMéga est de 0V. On se trouve donc dans la disposition suivante :

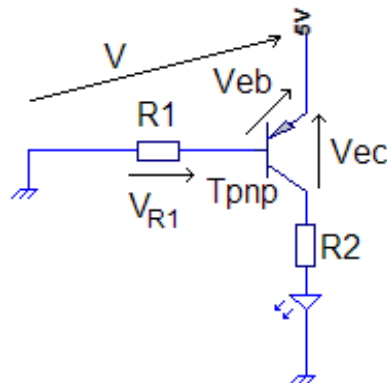


Illustration 6: cas où le transistor est saturé

La tension  $V$  est alors égale à 5V, il faut donc définir  $R1$  de façon à ce que  $V_{R1}$  soit égale à 4,3V et que  $V_{eb}$  soit égale à 0,7V. Grâce au test sur plaque d'essai j'ai pu définir  $R1 = 820\Omega$ .

Ainsi lorsque la sortie de l'ATMéga sera à 1, on se trouvera dans la disposition suivante :

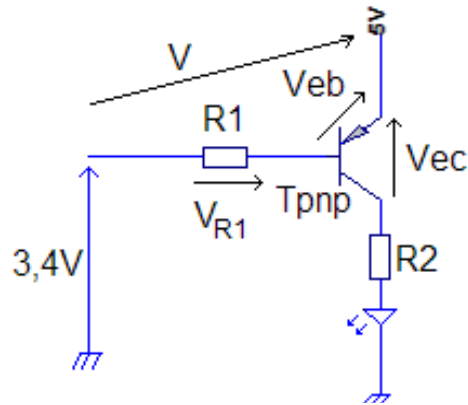


Illustration 7: cas où le transistor est bloqué

V ne vaudra plus que 1,6V, cette tension sera captée par la résistance R1 et Veb sera donc de 0V.

### 3.2.3. Conclusion de l'étude

Pour la réalisation nous utiliserons des transistors dont la référence est 2N2907A.

## 3.3. Étude des diodes multicolores FP4

### 3.3.1. But de l'étude

Le but de cette étude est de trouver comment obtenir 6 couleurs, différenciables par l'utilisateur du MasterMind électronique, à partir d'une diode multicolore. Le datasheet de ce composant se trouve en annexe.

### 3.3.2. Présentation de la diode multicolore

La diode multicolore est un composant qui permet d'obtenir n'importe quelles couleurs du spectre lumineux de la lumière blanche (de 400 à 750 nm) à partir des 3 couleurs primaires en optique : le bleu (425 nm), le vert (565 nm) et le rouge (625 nm).

Ce composant contient 4 diodes électro-luminescentes, deux bleus, une verte et une rouge et fonctionne avec 6 pattes, quatre anodes et deux cathodes, dont la répartition est la suivante :

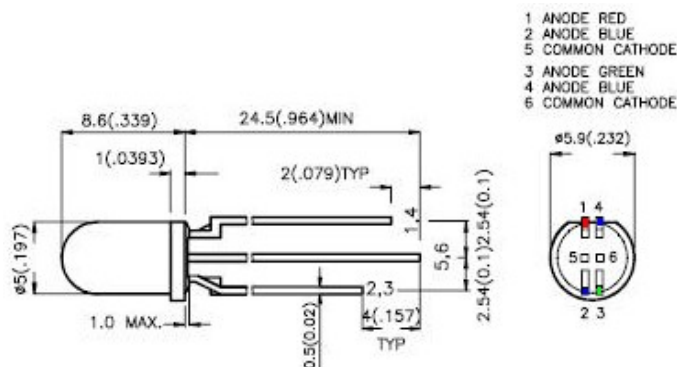


Illustration 8: diode multicolore

Pour câbler cette diode multicolore il suffit de mettre les cathodes à la masse et de mettre les anodes sur la tension d'alimentation par l'intermédiaire d'une résistance.

Remarque : ce genre de diode est aussi appelé par deux acronymes, DEL RVB(diode électroluminescente rouge vert bleu) ou en anglais LED RGB(light emitting diode red green blue).

### 3.3.3. Calcul des résistances pour chaque DEL

Pour connaître la résistance à placer sur chaque anode, il faut tout d'abord se référer au document fourni par le constructeur. On y trouve les courbes d'intensité lumineuse en fonction du courant d'entrée dans l'anode. La couleur de chaque courbes représente la couleur de l'anode à laquelle elle se réfère.

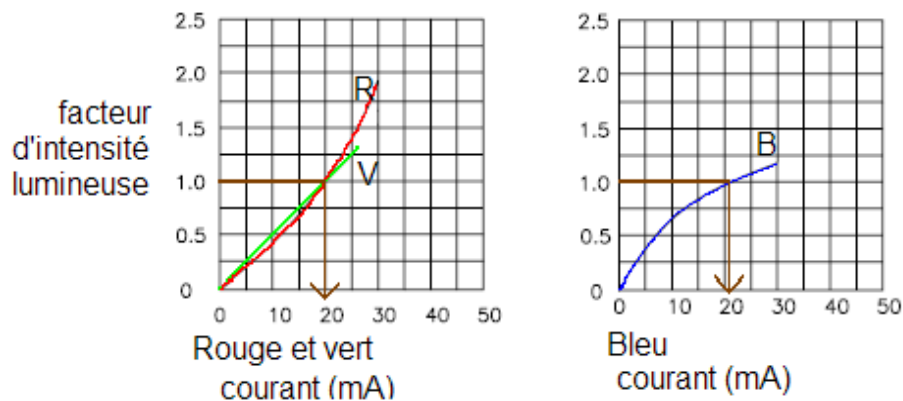


Illustration 9: intensité=fct(facteur d'intensité lumineuse)

Afin qu'aucune couleurs ne prédominent sur les autres, on regarde le courant équivalent à un même facteur d'intensité lumineuse, soit ici 1. On sélectionne donc un courant de 20mA.

Ensuite on se refaire aux schémas qui nous donnent, pour chaque anode, la tension nécessaire d'utilisation en fonction du courant.

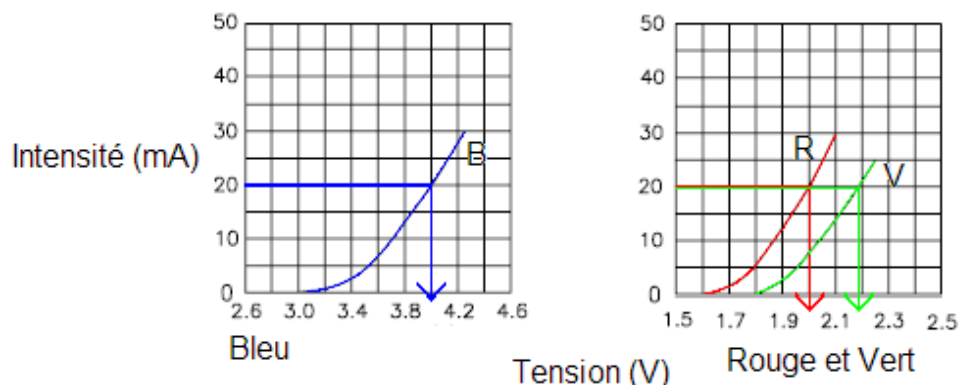


Illustration 10: tension=fct(intensité)

On obtient donc d'après les graphiques de la page précédente :

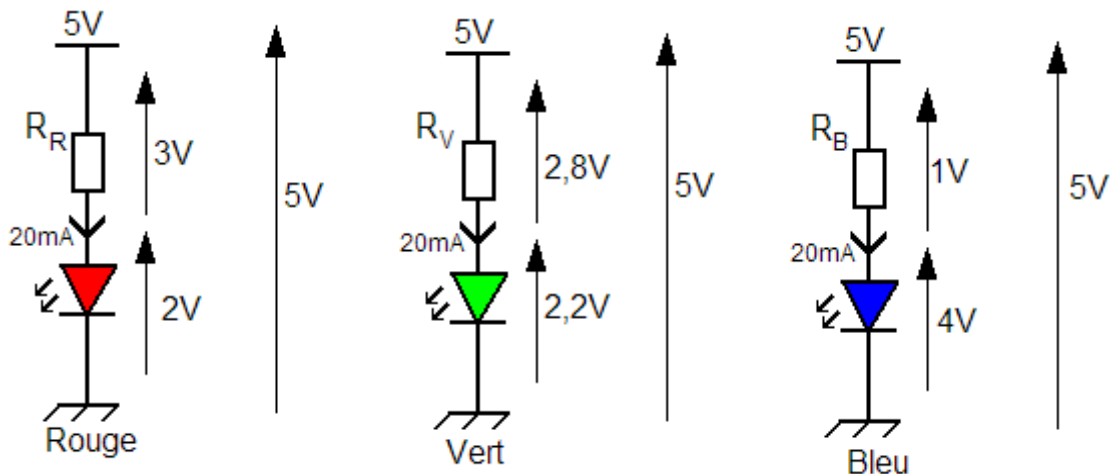


Illustration 11: calculs des résistances pour la diode multicolore

$$\text{Soit : } R_R = \frac{3V}{20mA} = 150\Omega ; \quad R_V = \frac{2,8V}{20mA} = 140\Omega ; \quad R_B = \frac{1V}{20mA} = 50\Omega$$

### 3.3.4. Résultat de l'étude

En s'assurant que chaque DEL éclaire à la même intensité on peut disposer des couleurs suivantes :

- Rouge
- Vert
- Bleu
- Orange (mélange de vert et de rouge)
- Violet (mélange de bleu et de rouge)
- Blanc (mélange de rouge, vert et de bleu)

Il s'agit donc de ces six couleurs ci que j'ai retenues pour la réalisation du MasterMind Électronique.

J'ai préféré choisir le violet et le orange plutôt que le noir et le jaune, normalement utilisées dans le jeu de société, car le jaune est peu différentiable du blanc et que le noir s'obtient en éteignant la diode.

L'étude de la partie électronique étant achevée, passons maintenant au programme.

## 4. Programmation de l'ATmega 8535, FP2

### 4.1. Présentation du programme

Le but du programme que va réaliser l'ATmega est, d'une part, de remplacer le joueur qui doit définir le code couleur à deviner et, d'autre part, de permettre au joueur de proposer le code qui lui semble juste.

Pour cela le programme se divise en 3 parties :

1. Génération du code couleur
2. Choix des couleurs à tester
3. Vérification du code proposée

Afin de simplifier l'attribution des couleurs aux diodes, j'ai associé un chiffre à chacune des couleurs. On aura donc le rouge représenté par 0, le vert par 1, le bleu par 2, le orange par 3, le violet par 4 et le blanc par 5. Ainsi les actions sur les modifications de couleurs seront gérées par des calculs et des comparaisons mathématiques.

### 4.2. Génération du code couleurs

Il s'agit de la partie du programme qui va générer le code de couleurs que le joueur devra deviner par la suite. Cette partie s'effectue en deux étapes, la première permet de définir une couleur aléatoire à associer à une des 3 diodes, et la seconde va gérer l'allumage de ces 3 diodes

Rappel : les trois diodes visualisant le code à deviner sont appelées D1, D2 et D3.

- Génération des couleurs aléatoires

On utilise pour cela la fonction appelée « rand() »<sup>1</sup>.

```
D1=rand()%6;
```

```
D2=rand()%6;
```

```
D3=rand()%6;
```

Grâce à cela chaque diode se verra attribuer un chiffre aléatoire allant de 0 à 5.

- Allumage des diodes

Cette étape est réalisée à l'aide d'un « switch »<sup>2</sup> qui va configurer les ports de sorties en fonction du chiffre associé à chaque diodes.

Dans l'exemple suivant, la diode Di (i prenant la valeur de 1, 2 ou 3) s'allumera en fonction de la valeur définie par le random. Il y aura donc 3 switch à effectuer.

---

1 Fonction qui génère un nombre aléatoire

2 Fonction de test dont la condition varie selon plusieurs cas de figure

```

switch(Di) //allumage de la LED Di
{
case 0: PORTx=PORTx || 0xYY;delay_ms(100);break; // ROUGE
case 1: PORTx=PORTx || 0xYY;delay_ms(100);break; // VERT
case 2: PORTx=PORTx || 0xYY;delay_ms(100);break; // BLEU
case 3: PORTx=PORTx || 0xYY;delay_ms(100);break; // ORANGE
case 4: PORTx=PORTx || 0xYY;delay_ms(100);break; // VIOLET
case 5: PORTx=PORTx || 0xYY;delay_ms(100);break; // BLANC
}

```

L'écriture sur un port de sortie se fait en hexadécimale.

Remarque : l'ATmega à un fonctionnement pas à pas, c'est pourquoi il est indispensable d'utiliser un « ou » logique (« || ») car un même port sert à alimenter plusieurs diodes. Sans ce « ou » logique, le port de sortie prendrait la dernière valeur qui lui a été transmise sans tenir compte des précédents paramètres entrées.

### **4.3. Choix des couleurs à tester**

Une fois le code généré par l'ATmega, le joueur devra faire défiler les 6 couleurs sur les diodes L1 à 3 à l'aide bouton poussoir P1 à 3.

On définit L1, L2 et L3 comme étant égale à 0 (soit rouge par défaut). Ainsi lorsque le joueur appuie sur l'un des boutons P1 à 3, il suffit d'incrémenter la valeur de la diode qui y correspond. Une fois que la valeur 5 est dépassée, le programme reprend l'incréméntation à partir de 0.

Dans l'exemple suivant, le bouton P1 est relié à la patte 39 de l'ATmega. La patte 39 représente le bit 1 du port A défini ici en entré, d'où la variable de test PINA.1.

```

if(PINA.1==1 && L1<=5)  \\ si P1 appuyé et L1 inférieur ou égal à 5.
{ L1=L1+1; delay_ms(100); } \\ alors L1 est incrémenté.
if(PINA.1==1 && L1>>5)  \\ si P1 appuyé et L1 égal à 5.
{L1=0; delay_ms(100); } \\ alors L1 reprend la valeur 0.

```



# 5. Poursuite du projet

## 5.1. Travail à faire

Suite à des problèmes techniques, notamment vis à vis de la prise en main du logiciel de programmation et de la réalisation de la carte, le projet n'a pas pu aboutir à son terme, voici donc ce qu'il reste à effectuer puis quelques pistes envisageables pour la suite:

- rédaction de la partie « Vérification du code proposée » du programme;
- finalisation de la carte, soudure de composants;
- implantation et test du programme dans la carte;
- réalisation d'un boîtier de jeu pour la carte.

## 5.2. Piste pour l'avenir

- compenser le nombre limité de port de l'ATmega par l'utilisation de démultiplexeur afin de rajouter la 4ème colonne du jeu original;
- libérer un port (grâce au démultiplexage) afin d'ajouter un afficheur LCD dans le but de garder une trace des propositions précédemment effectuées.

# Conclusion

Ce projet ma beaucoup intéressé car il m'a permis d'étudier des composants que je n'avais pas eu l'occasion d'utiliser au cours de ma formation en GEII. De plus j'ai pu développer mes connaissances en matière de composants programmables.

Malheureusement je n'ai pas pu terminer la partie réalisation car j'ai passé trop de temps pour la compréhension du logiciel de programmation et réalisation de la carte. En revanche j'ai tout de même pu établir le schéma électrique de la carte et le typon de réalisation.

## Index des illustrations

Illustration 1: jeu de MasterMind original.....	5
Illustration 2: schéma de principe du MasterMind électronique.....	6
Illustration 3: schéma de principes.....	9
Illustration 4: schéma structurel de FP1.....	10
Illustration 5: montage à transistor PNP.....	11
Illustration 6: cas où le transistor est saturé.....	11
Illustration 7: cas où le transistor est bloqué.....	12
Illustration 8: diode multicolore.....	12
Illustration 9: intensité=fct(facteur d'intensité lumineuse.....	13
Illustration 10: tension=fct(intensité).....	13
Illustration 11: calculs des résistances pour la diode multicolore.....	14

## Bibliographie

- [1] **Thierry Lequeu**, "*Projet 02 - ROTALED / Gyrophare de signalisation à LEDs*", IUT GEII Tours , janvier 2010.
- [2] **Julien Besnard**, "*Mastermind Electronique*", IUT GEII Tours , mars 2009.
- [3] **Philips**, "*DATASHEET PH2907A*", [http://www.datasheetcatalog.com/datasheets\\_pdf/P/H/2/9/PH2907A.shtml](http://www.datasheetcatalog.com/datasheets_pdf/P/H/2/9/PH2907A.shtml) , 1999.
- [4] **Kingbright**, "*Full color rgb lamp*", <http://pdf1.alldatasheet.com/datasheet-pdf/view/92898/ETC/LF59EMBGMBW.html> , .
- [5] **Frank Sauret**, "*Brochage de l'ATmega8535*", <http://anyedit.free.fr/telechargement/atmega8535-francais.pdf> , 2005.

- [1][2][3][4][5]