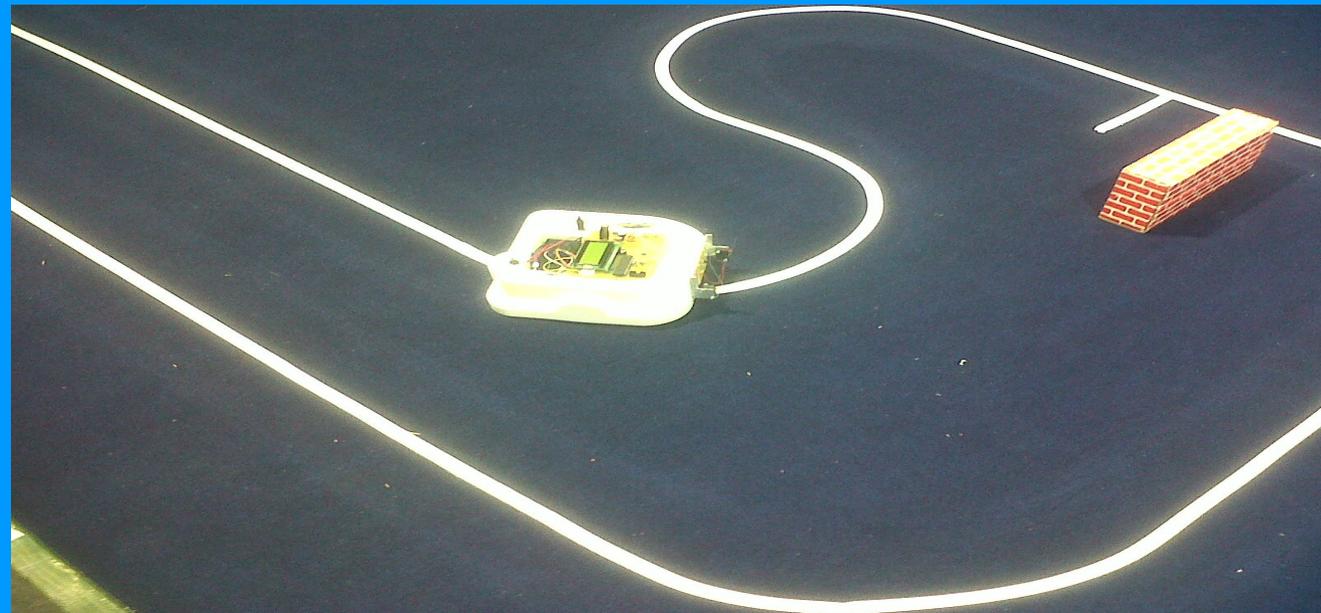




Projet d'étude et réalisation
Robot suiveur de ligne blanche



Vidal
Lallemand
2^{ème} année
Groupe K3b

Mathieu
Jérémy

Professeurs responsables: T LEQUEU
C GLIKSOHN

Robot suiveur de ligne blanche

Cahier des charges

Présentation du projet

L'électronique

La programmation

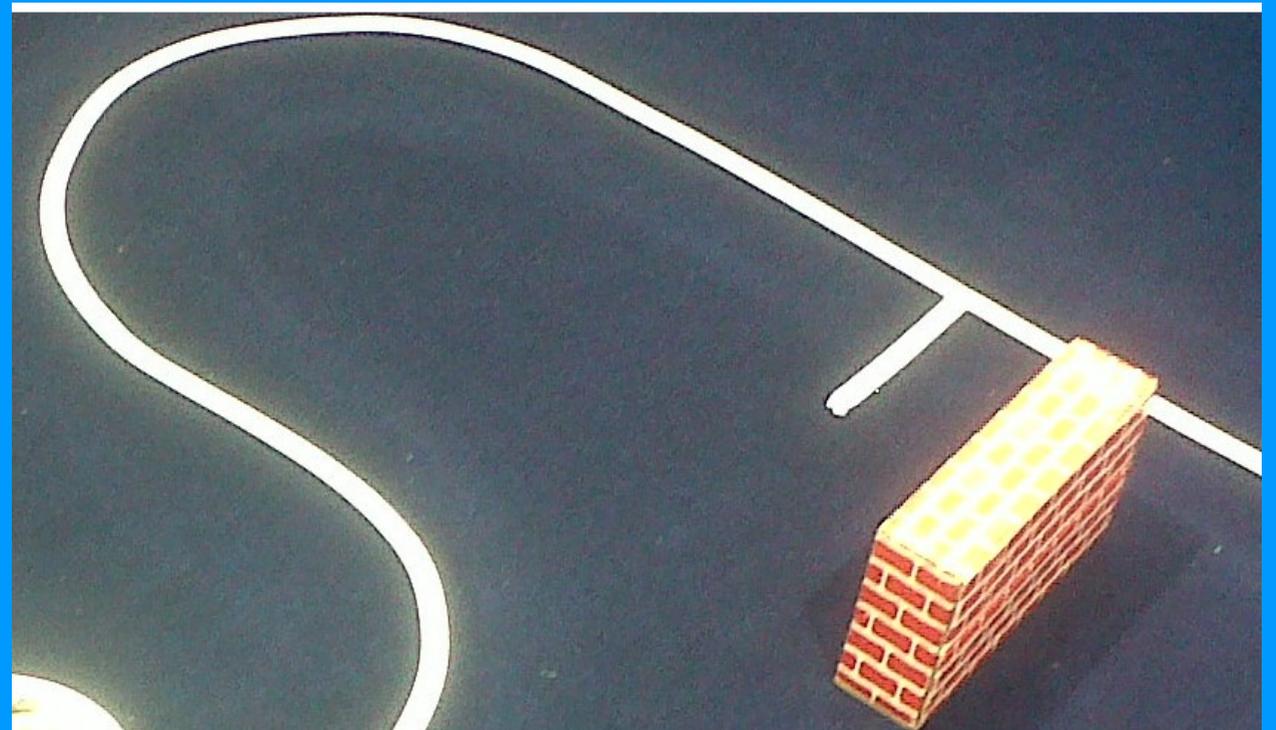
Le challenge robotique

Conclusion

Cahier des charges

Robot suiveur de ligne de ligne blanche

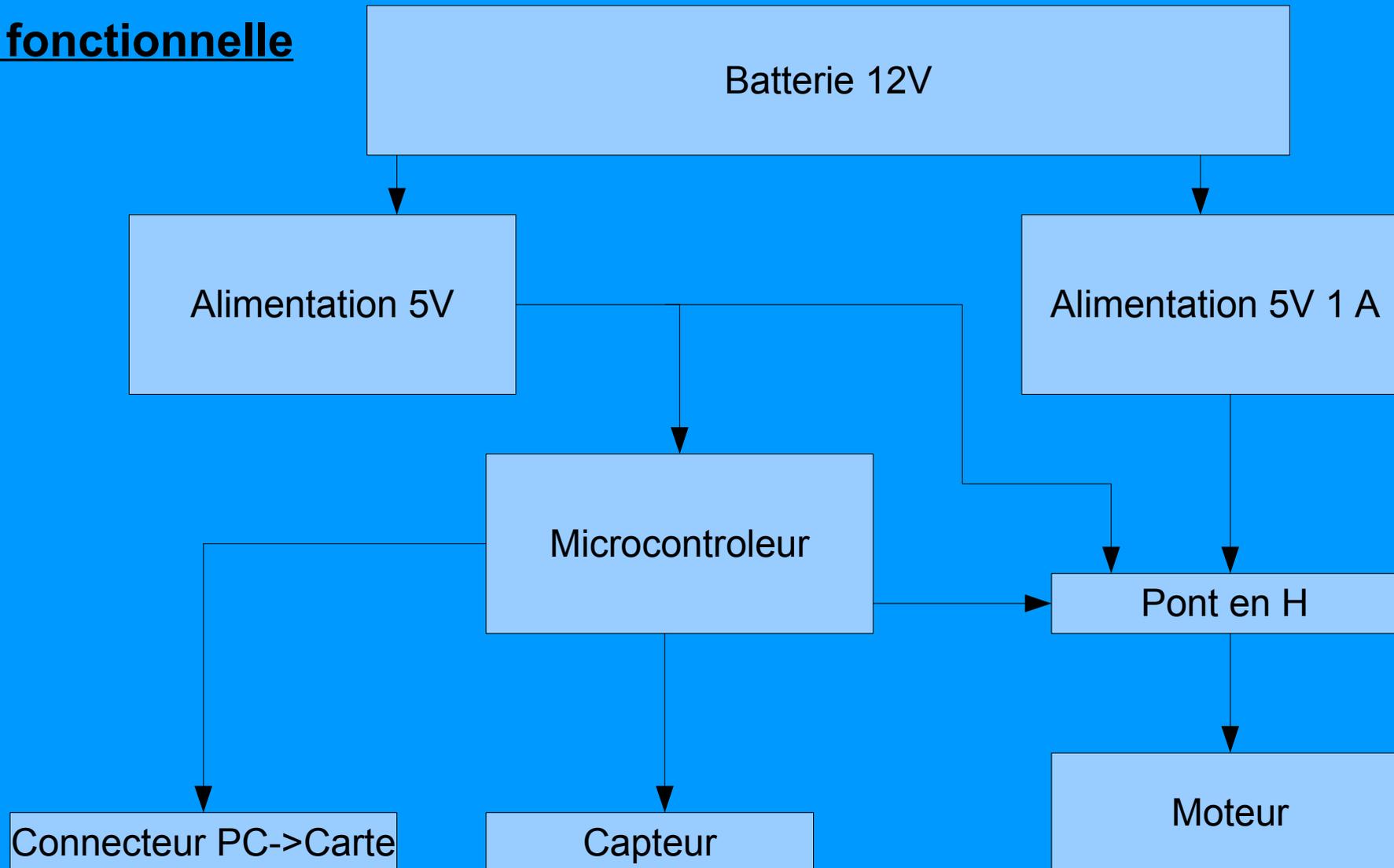
- Dimensions
- Respecter les obstacles du parcours
- Autonome
- Kit imposé
- Arrêt d'urgence obligatoire
- Prise jack pour le démarrage



Exemple de parcours

Présentation du projet

Analyse fonctionnelle



Présentation du projet

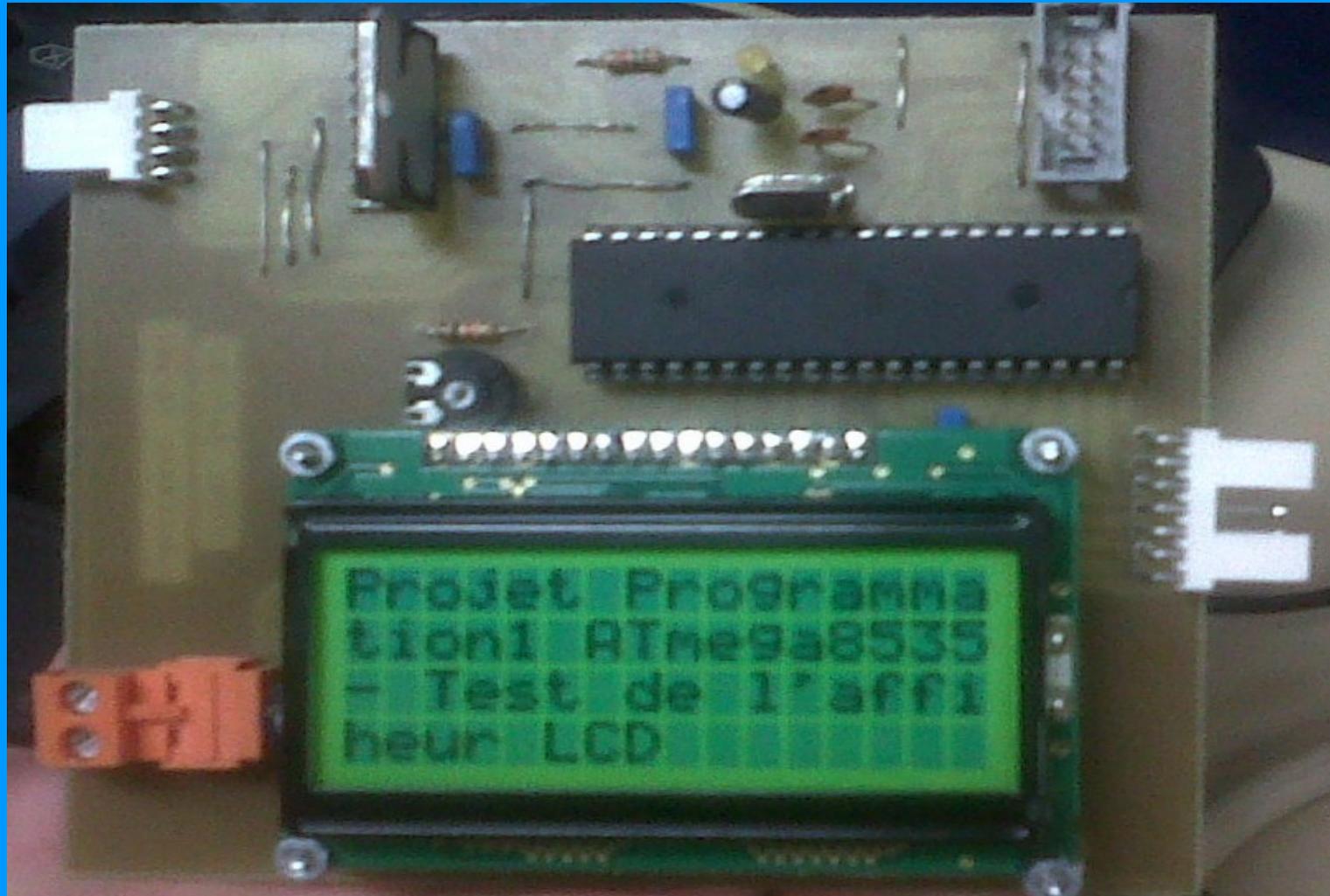
VIDAL Mathieu

LALLEMAND Jérémy

	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
Essais théorique		X		X					X								
Essais théorique		○	○	○						○							
Orcad			X		X												
Orcad					○						○						
Typon					X						○		○				
Typon					○	○											
Test						X							X		X	X	
Test						○							○				
Maintenance						X							X	X	X	X	
Maintenance									○	○							
Soudure					X		○	○					○				
Soudure							○	○					○				
Essais pratiques						X							X	X	X	X	
Essais pratiques									○				○				
Programmation										X	X						
Programmation										○	○		○				
Oral																	X
Oral																	○

Planning prévisionnel	X
Planning Réel	○
Coupe robotique	

L'électronique



Carte avec afficheur LCD

L'électronique

Composants utilisé:

- LM2575
- LM2574
- L298N
- Atmega 8535
- CNY70
- Écran LCD
- Batterie 12V_1,2A



L298N



Atmega 8535



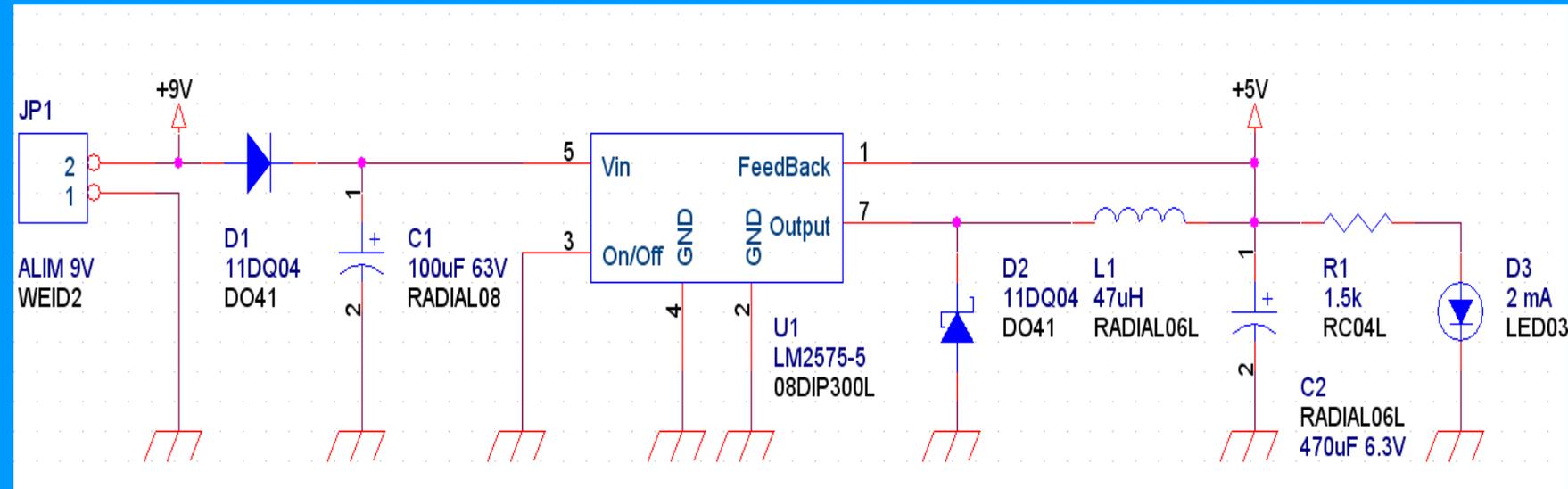
CNY70

L'électronique

LM2574



Schéma électrique



Caractéristiques:

- Tension max : 45V
- Tension de sortie : 5V
- Courant max : 500mA

L'électronique

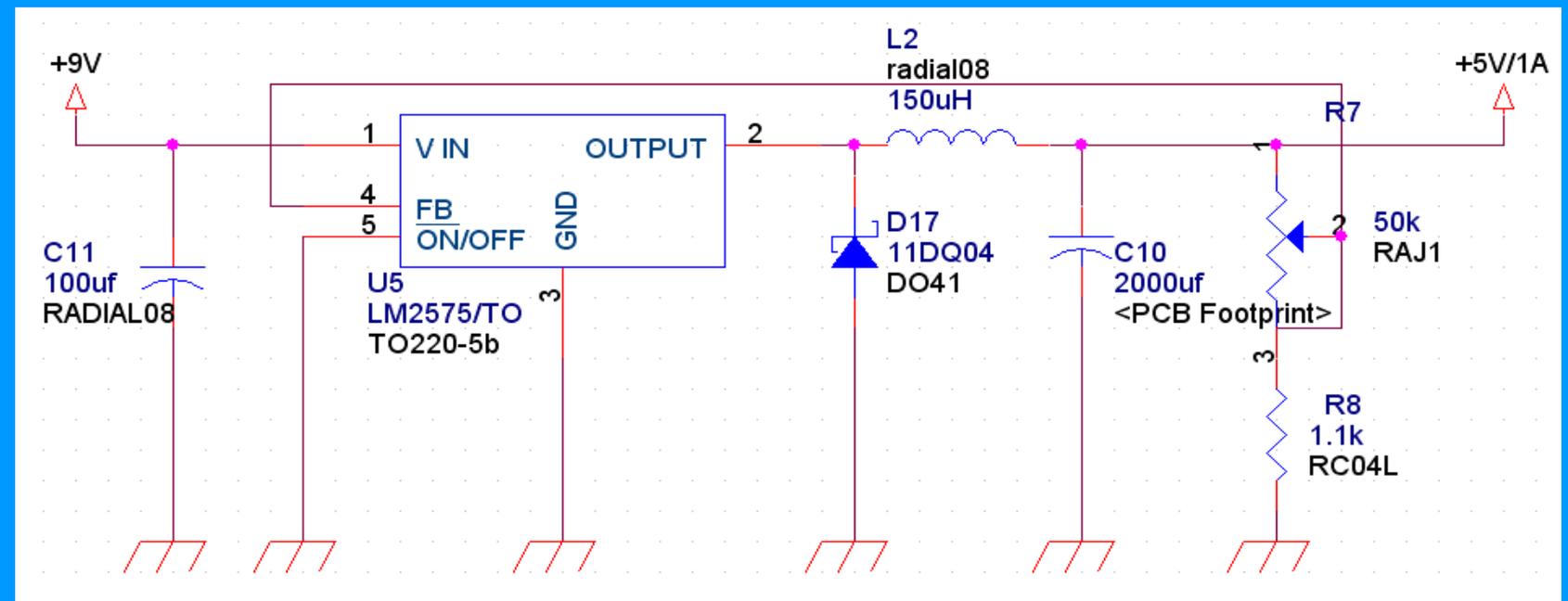
LM2575/1A



Caractéristiques:

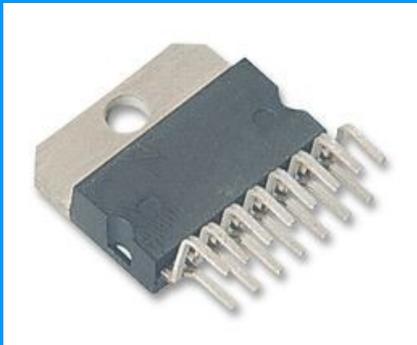
- Tension max : 18V
- Tension de sortie : 5V
- Courant max : 1A

Schéma électrique



L'électronique

L298N



Principales caractéristiques :

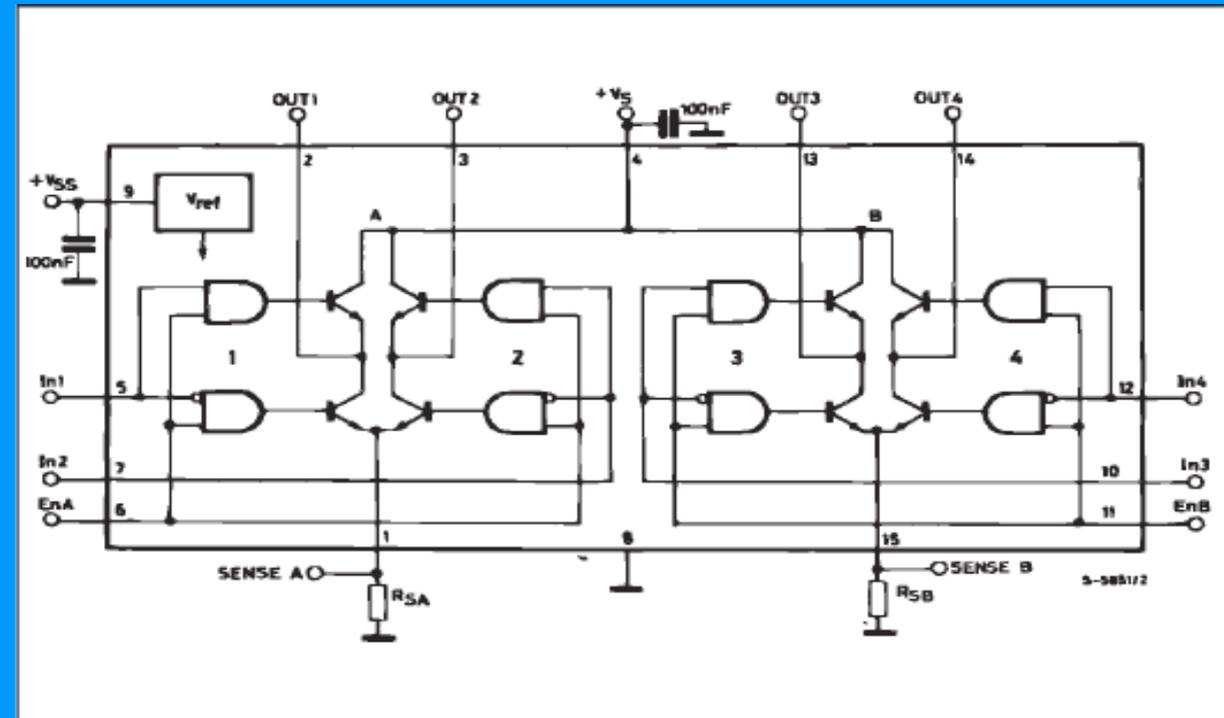
Tension d'alimentation du circuit logique : 5 V

Tension d'alimentation du circuit puissance :
0 V / 48 V (nous serons à 5 V)

Tension minimale de sensibilité MLI : 2.3 V

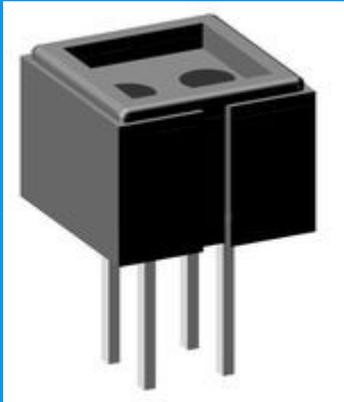
Courant maximal du circuit puissance : 3A

Schéma électrique



L'électronique

CNY70



Capteur infrarouge sensible à la couleur, niveau logique de 5V lorsqu'il perçoit une ligne blanche et au niveau bas pour du noir.

Caractéristiques:

- Tension max :1.6V
- Tension nominal : 1.25V
- Courant d'alimentation : 50mA

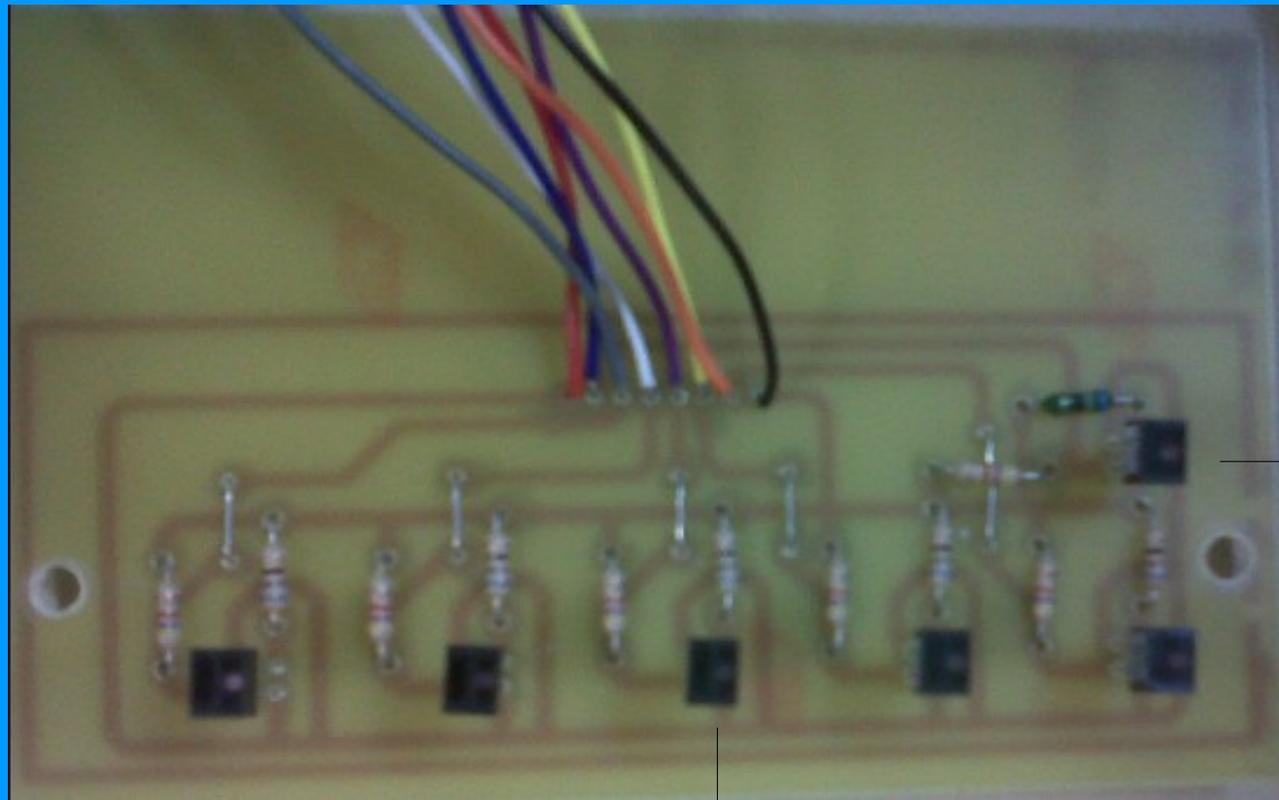
Détecteur:

- Tension de sortie nominal: 5V
- Tension de sortie maximal: 5V
- Courant de sortie nominal: 100uA
- Courant de sortie maximum 1mA

L'électronique

La carte capteurs

Réalisée de façon a capter le mieux possible la ligne blanche



Capteurs pour le stop figure

Capteur du milieux pour la ligne droite

L'électronique

L'écran LCD



L'électronique

Batterie 12V / 1,2Ah



Programmation

Programmation sur Code Vision AVR : Logiciel servant à créer une programmation en langage C, à les compiler, puis à envoyer le programme vers le micro-contrôleur rentré en paramètres

Les Variables :

Int buf, i, val2 → 3 Variables entières

Unsigned char tampon[20], val[8] → Tableau de caractères

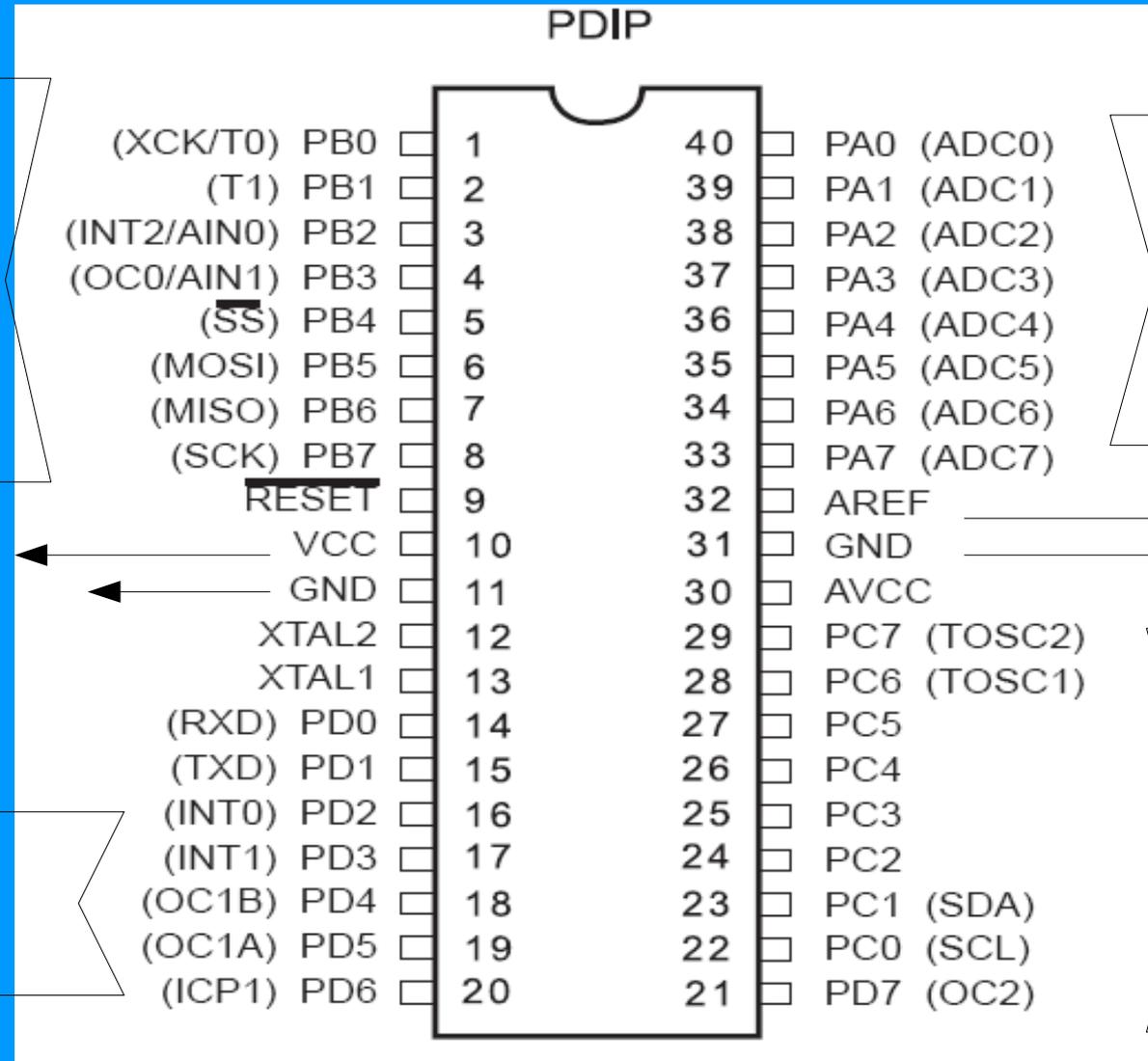
Programmation

Atmega 8535

Connections PC-CARTE

+5V
Masse

Moteur



Capteurs

+5v
Masse

LCD

Programmation

Configuration de l'ATmega 8535:

```
// Définition du port A en entrée
PORTA=0x00;
DDRA=0x00;

// Définition du port D en sortie
PORTD=0x00;
DDRD=0x30;

// Définition du timer 1 pour les MLI
TCCR1A=0xA1;
TCCR1B=0x01;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Définition du comparateur
ACSR=0x80;
SFIOR=0x00;

// Initialisation du convertisseur Analogique Numérique
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOR&=0xEF;
```

Programmation

Reconnaissance de la ligne:

Utilisation de l'analogique Numérique.

```
// Lecture du résultat de la conversion Analogique Numérique
unsigned int read_adc(unsigned char adc_input)
{
  ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
  delay_us(10);
  ADCSRA|=0x40;
  while ((ADCSRA & 0x10)==0);
  ADCSRA|=0x10;
  return ADCW;
}
```

Boucle effectuée autant de fois que le nombre de capteurs.

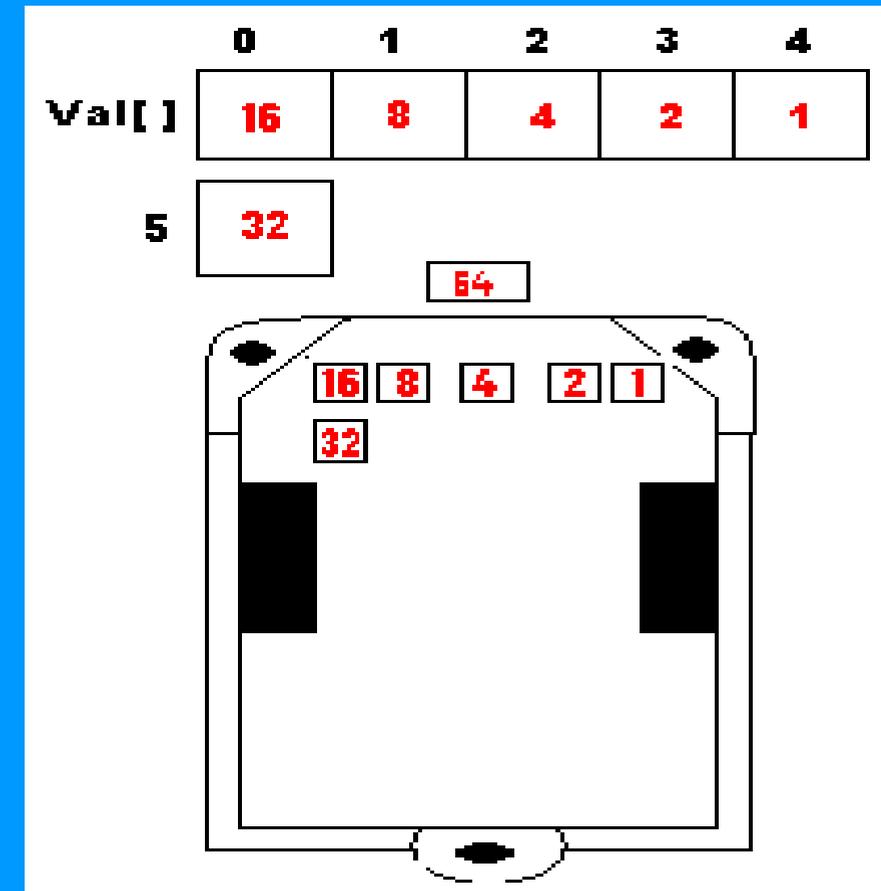
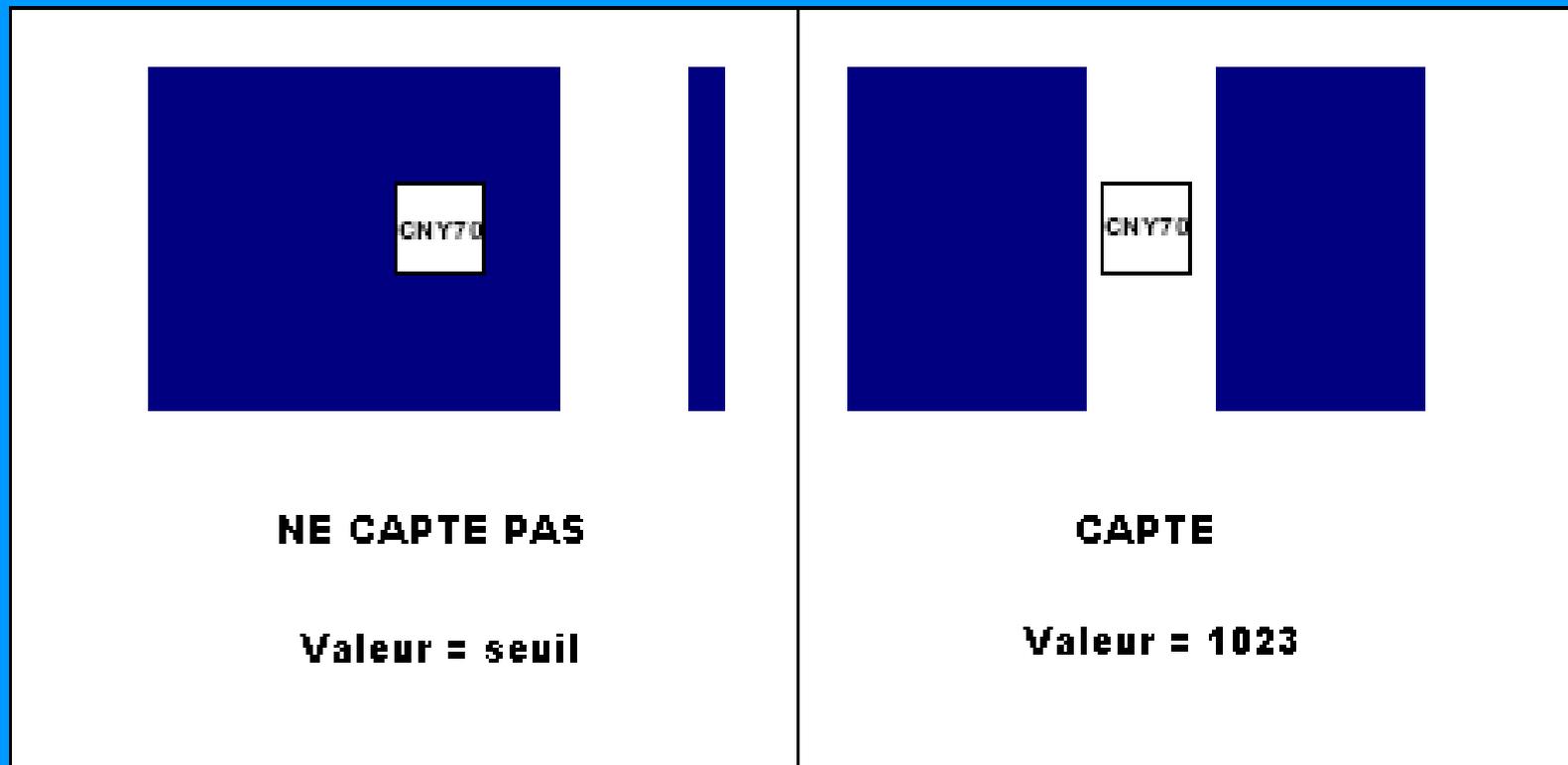
```
for (i=0; i<8; i++)
{
  |
  buf = read_adc(i);

  if(buf>250)
    val[i]=1;
  else
    val[i]=0;
}
val2= val[0]+val[1]*2+val[2]*4+val[3]*8+val[4]*16+val[5]*32+val[6]*64;
```

Programmation

La valeur « Val » nous renvoie une suite de 1 ou de 0 donc du binaire qui est ensuite convertie en une image décimale par un calcul:

$$\text{Val2} = \text{val}[0] + \text{val}[1] \times 2 + \text{val}[2] \times 4 + \text{val}[3] \times 8 + \text{val}[4] \times 16 + \text{val}[5] \times 32 + \text{val}[6] \times 64$$



Programmation

La gestion des cas possibles →

Etat des capteurs							Valeur Hexadécimale
Capteurs							
5	6	0	1	2	3	4	
0	0	0	0	1	0	0	0x04
0	0	0	0	1	1	0	0x06
0	0	0	0	0	1	0	0x02
0	0	0	0	0	1	1	0x03
0	0	0	0	0	0	1	0x01
0	0	0	1	1	0	0	0x0C
0	0	0	1	0	0	0	0x08
0	0	1	0	0	0	0	0x10
0	0	1	1	0	0	0	0x18
0	0	0	0	0	0	0	0x00
0	0	0	0	1	1	1	0x07
0	1	0	0	0	0	1	0x21
0	1	1	1	1	1	1	0x3F
1	0	0	0	1	0	0	0x44
1	0	0	0	0	0	0	0x40

→ Seule le capteur du milieu capte

← Les 3 Capteurs de droites captent

Programmation

La programmation

```
lcd_gotoxy(0,0); // emplacement de l'écriture sur le LCD
sprintf(tampon, "%d", buf); // entre le buf dans tampon
lcd_puts(tampon); // restitue le tampon sur le LCD
lcd_gotoxy(0,1);
sprintf(tampon, "%2d", val2);
lcd_puts(tampon);
lcd_gotoxy(0,2);
sprintf(tampon, "%3d", OCR1AL);
lcd_puts(tampon);
lcd_gotoxy(0,3);
sprintf(tampon, "%3d", OCR1BL);
lcd_puts(tampon);
lcd_gotoxy(7,0);
lcd_putsf("mathieu"); // écriture du mots « mathieu » sur le LCD
lcd_gotoxy(7,1);
cd_putsf("jeremy");
```

Programmation

Suite du programme

```
switch (val2)
{
    case 0x04: //ligne droite
        OCR1AL=190;
        OCR1BL=190;
        break;
    case 0x06: // très petit virage droite
        OCR1AL=90;
        OCR1BL=190;
        break;
    case 0x02: //virage droite
        OCR1AL=90;
        OCR1BL=190;
        break;
    case 0x03: // gros virage droite
        OCR1AL=90;
        OCR1BL=190;
        break;
```

```
    case 0x01: //très gros virage droite
        OCR1AL=90;
        OCR1BL=190;
        Break;
    case 0x0C: //très petit virage gauche
        OCR1AL=190;
        OCR1BL=90;
        break;
    case 0x08: //virage gauche
        OCR1AL=190;
        OCR1BL=90;
        break;
    case 0x10: //très gros virage gauche
        OCR1AL=190;
        OCR1BL=90;
        break;
```

Programmation

```
case 0x18: // gros virage gauche
```

```
    OCR1AL=190;
```

```
    OCR1BL=90;
```

```
break;
```

```
case 0x00: // hors piste
```

```
    OCR1AL=190;
```

```
    OCR1BL=190;
```

```
break;
```

```
case 0x07: //priorité a droite
```

```
    OCR1AL=190;
```

```
    OCR1BL=190;
```

```
break;
```

```
case 0x21: //figure
```

```
    OCR1AL=190;
```

```
    OCR1BL=190;
```

```
break;
```

```
case 0x3F: // intersection
```

```
    OCR1AL=190;
```

```
    OCR1BL=190;
```

```
break;
```

```
case 0x1C: // raccourcis à gauche
```

```
    delay_ms(10);
```

```
    OCR1AL=40;
```

```
    OCR1BL=150;
```

```
break;
```

```
case 0x44:
```

```
    OCR1AL=0; //arrêt final
```

```
    OCR1BL=0;
```

```
    delay_ms(10000);
```

```
break;
```

```
case 0x40:
```

```
    OCR1AL=0;
```

```
    OCR1BL=0;
```

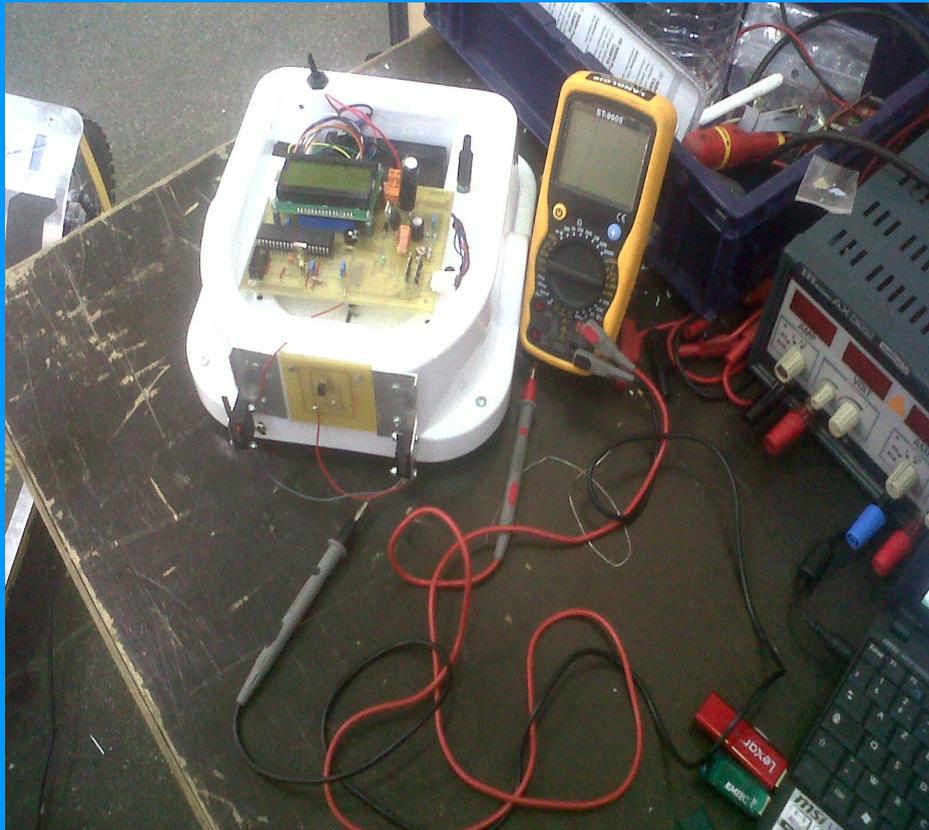
```
    delay_ms(10000);
```

```
break;
```

```
}
```

Le challenge robotique

Phase d'homologation



Réglage des capteurs afin de pouvoir continuer le concours et ce qualifier .

Élimination lors de la première phase

