

J.L.S. INFORMATIQUE

2, rue Clément ADER

B.P. 50065

57972 YUTZ CEDEX

Tel : 33 (0)3 82 86 00 16

Fax : 33 (0)3 82 86 00 12

URL : www.jls-info.com

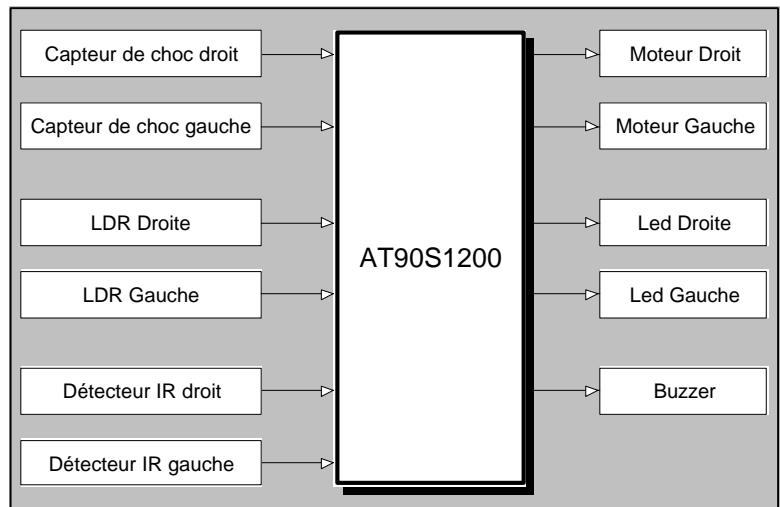
IPR1200

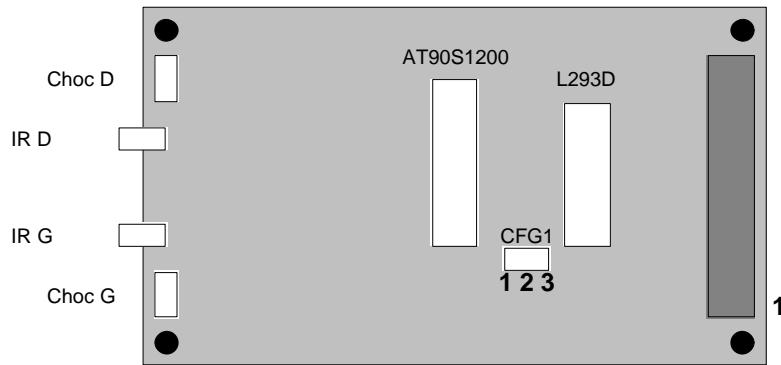
Interface Pour Robot à AT90S1200

Caractéristiques générales

La carte IPR1200 est une interface de pilotage complète pour la robotique mobile.

- Microcontrôleur AT90S1200, 4MHz
- 2 sorties de pilotage pour moteurs DC
- 2 capteurs IR de détection à distance
- 2 entrées pour capteurs de choc
- 2 Leds rouges
- 2 LDR
- 1 buzzer
- Fonctionnement de 0°C à 60°C
- Alimentation électronique >6.5Vdc
- Alimentation moteurs 3Vdc à 9Vdc





Brochage du connecteur 7 points

Broche	1	2	3	4	5	6	7
Nom	Vdd électronique	Vpp moteurs	GND	MG+	MG-	MD+	MD-
Description	> 6.5Vdc	3Vdc à 9Vdc	0V	Moteur gauche +	Moteur gauche -	Moteur droite +	Moteur droite -

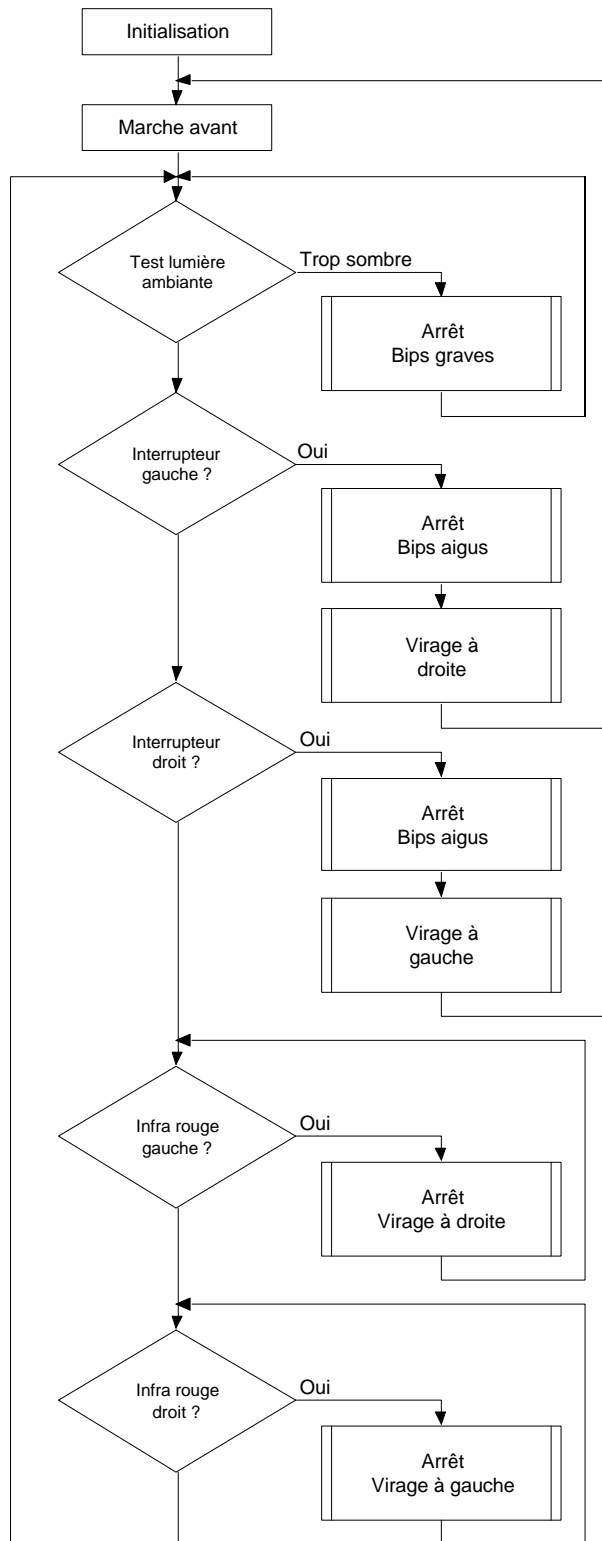
Configuration, sélection de tension moteurs

Moteur	3Vdc	3Vdc à 9Vdc	9Vdc
Alimentations	Alim. électronique > 6.5Vdc Alim. moteur 3Vdc	Alim. électronique > 6.5Vdc Alim. moteur	Alim. unique 9Vdc (pile 6F22, par ex.)
CFG1	1 - 2	1 - 2	2 - 3

Capteurs de choc avant

Il faut prendre des modèles miniatures avec un levier métallique, oblique de préférence. Ils sont appelés aussi mini rupteurs ou micro rupteurs. Ils disposent normalement de contacts C, NO, NF. Il faut câbler C et NO, ce qui produit le contact lors d'une collision.

Programmation



Code source

```
.include "1200def.inc"

.def    temp    = r16                ;registre general #1
.def    temp2   = r17                ;registre general #2
.def    tempo1  = r18                ;registres tempo
.def    tempo2  = r19
.def    buzz1   = r20                ;registres buzzer
.def    buzz2   = r21
.def    buzz3   = r22

.cseg                                     ;code segment
.org    $0000                            ;adresse vecteur reset
        rjmp    reset                    ;saut vecteur reset

.org    $000D                            ;adresse debut de code

init:
        ser     temp                    ;port b en sortie
        out    ddrb,temp
        clr    temp                    ;port d en entree
        out    ddrd,temp
        sbi    portb,pb5                ;desactive buzzer
        rcall  led_dg_off               ;led d +g eteintes
        ret

delay:
        ldi    tempo1,$ff
loop_t1:
        ldi    tempo2,$ff
loop_t2:
        dec    tempo2                  ;decremente boucle 1
        brne  loop_t2                  ;teste si termine
        dec    tempo1                  ;decremente boucle 2
        brne  loop_t1                  ;teste si termine
        ret

led_d_on:
        cbi    portb,pb6                ;led d allumee
        ret                               ;retour
led_g_on:
        cbi    portb,pb7                ;led g allumee
        ret                               ;retour
led_dg_on:
        in     temp,portb                ;lecture port b
        andi   temp,$3f                 ;led d + g allumees
        out    portb,temp                ;ecriture port b
        ret                               ;retour
led_d_off:
        sbi    portb,pb6                ;led d eteinte
        ret                               ;retour
led_g_off:
        sbi    portb,pb7                ;led g eteinte
        ret                               ;retour
led_dg_off:
        in     temp,portb                ;lecture port b
        ori    temp,$c0                 ;led d + g eteintes
        out    portb,temp                ;ecriture port b
        ret                               ;retour
```

```

toggle_leds:
    rcall    led_dg_on           ;allumage d + g
    rcall    delay              ;attente
    rcall    led_dg_off        ;extinction d + g
    rcall    delay              ;attente
    ret                                ;retour

.macro buzzer
    ldi      buzz3,@2          ;charge nbre boucles
loop_b3:
    ldi      buzz1,@0          ;charge temps bas
    cbi      portb,5           ;active buzzer
loop_b2:
    dec      buzz1             ;decremente boucle 1
    nop
    brne    loop_b2            ;teste si termine
    sbi      portb,5           ;desactive buzzer
    ldi      buzz2,@1          ;charge temps haut
loop_b1:
    dec      buzz2             ;decremente boucle 2
    nop
    brne    loop_b1            ;teste si termine
    dec      buzz3             ;decremente boucle 3
    brne    loop_b3            ;teste si termine
.endmacro

test_lum:
    in       temp,pind         ;charge port d
    andi     temp,$0c          ;isole bits ldr d + g
    brne    buz_low            ;teste si ldrs pas eclairees
    rcall    m_on              ;mise en route moteurs
    ret                                ;si non retour
buz_low:
    rcall    m_off              ;arret moteurs
    buzzer  $20,$ff,$7f        ;son grave
    rcall    delay              ;delai
    rjmp     test_lum          ;continue

m_d_av:
    cbi      portb,pb3          ;marche avant d
    sbi      portb,pb2
    ret

m_d_ar:
    sbi      portb,pb3          ;marche arriere d
    cbi      portb,pb2
    ret

m_g_av:
    cbi      portb,pb0          ;marche avant g
    sbi      portb,pb1
    ret

m_g_ar:
    sbi      portb,pb0          ;marche arriere g
    cbi      portb,pb1
    ret

m_on:
    sbi      portb,pb4          ;marche des moteurs
    ret

m_off:
    cbi      portb,pb4          ;arret des moteurs
    ret

```

```

reset:
    rcall  init                ;initialisation systeme
    rcall  delay               ;attente pour laisser
    rcall  delay               ;le robot libre

main:
    rcall  m_d_av              ;marche avant d
    rcall  m_g_av              ;marche avant g
    rcall  m_on                ;marche des moteurs

main2:
    rcall  test_lum            ;verifie lumiere ambiante
    sbis   pind,4              ;swg appuye ?
    rjmp   detour_sw          ;si oui detour
    sbis   pind,5              ;swd appuye ?
    rjmp   detour_sw          ;si oui detour
    sbis   pind,0              ;irg actif ?
    rjmp   detour_ir          ;si oui detour
    sbis   pind,1              ;ird actif ?
    rjmp   detour_ir          ;si oui detour
    rjmp   main                ;continue

detour_sw:
    rcall  m_off               ;arret moteurs
    ldi    temp2,$02          ;charge nombre de boucles

loop1:
    buzzer $10,$50,$7f       ;son aigu
    rcall  delay               ;delai
    dec    temp2               ;decremente nombre de boucles
    brne   loop1              ;teste si termine
    in     temp,pind           ;
    rcall  m_d_ar              ;marche arriere d
    rcall  m_g_ar              ;marche arriere g
    rcall  m_on                ;activation des moteurs
    ldi    temp2,$08          ;charge nombre de boucles

loop2:
    rcall  delay               ;delai
    dec    temp2               ;decremente nombre de boucles
    brne   loop2              ;teste si termine
    rcall  m_off               ;arret moteurs
    sbrs   temp,4              ;swg appuye ?
    rjmp   virage_sw_d        ;si oui virage droite
    sbrs   temp,5              ;swd appuye ?
    rjmp   virage_sw_g        ;si oui virage gauche

virage_sw_d:
    rcall  m_d_ar              ;moteur d arriere
    rcall  m_g_av              ;moteur g avant
    rjmp   fin_virage_sw      ;termine

virage_sw_g:
    rcall  m_d_av              ;moteur d avant
    rcall  m_g_ar              ;moteur g arriere

fin_virage_sw:
    rcall  m_on                ;marche des moteurs
    ldi    temp2,$07          ;charge nombre de boucles

loop3:
    rcall  delay               ;delai
    dec    temp2               ;decremente nombre de boucles
    brne   loop3              ;teste si termine
    rcall  m_off               ;arret des moteurs
    rjmp   main                ;continue

```

```

detour_ir:
    rcall  m_off           ;arret moteurs
    in     temp,pind       ;sauve etat precedent
    rcall  m_d_ar          ;marche arriere d
    rcall  m_g_ar          ;marche arriere g
    rcall  m_on            ;activation des moteurs
    ldi    temp2,$06       ;charge nombre de boucles
loop4:
    rcall  toggle_leds    ;leds clignotantes
    rcall  delay           ;delai
    dec    temp2           ;decremente nombre de boucles
    brne   loop4           ;teste si termine
    rcall  m_off           ;arret moteurs
    sbrs   temp,0          ;irg actif ?
    rjmp   virage_ir_d     ;si oui virage droite
    sbrs   temp,1          ;ird actif ?
    rjmp   virage_ir_g     ;si oui virage gauche
virage_ir_d:
    rcall  m_d_ar          ;moteur d arriere
    rcall  m_g_av          ;moteur g avant
    rjmp   fin_virage_ir   ;termine
virage_ir_g:
    rcall  m_d_av          ;moteur d avant
    rcall  m_g_ar          ;moteur g arriere
fin_virage_ir:
    rcall  m_on            ;charge nombre de boucles
    ldi    temp2,$08
loop5:
    rcall  delay           ;delai
    dec    temp2           ;decremente nombre de boucles
    brne   loop5           ;teste si termine
    rcall  m_off           ;arret des moteurs
    rjmp   main            ;continue

```

J.L.S. INFORMATIQUE

2, rue Clément Ader

B.P. 50065

57972 Yutz, Cedex

France

Tél.: 03 82 86 00 16

Fax : 03 82 86 00 12

www.jls-info.com