

```

1: /*
2:    GEN.H
3: */
4:
5: #define VITROBOT    20        // en cm/s de 5 à 30cm/sec
6:
7: #define EMPATTEMENT 21
8: #define REDUCTEUR   25
9: #define DIARQUE     52
10:
11:
12: #define NBSECTEUR   32        // nb secteurs disque optique
13: #define NBT1        (NBSECTEUR/2) // nb periodes disque optique
14:
15: #define Rmini    15        // ATTENTION à la taille des variables ...
16: #define Rmoyen   40        // ici T est unsigned int donc pas de
17: #define Rmaxi    100       // vitesse < 5 cm/sec ni de R < 15
18:
19: #include "mypic.h"
20:
21: //Defines
22:
23: #define OPTOMOTGauche RB4        //optos acquisition vitesse moteurs
24: #define OPTOMOTDroit  RB6
25:
26:                                //optos Ligne
27: #define OPTOLigneGauche RA6
28: #define OPTOLigneCentre RA5
29: #define OPTOLigneDroit  RA4
30:
31: #define LEDDA          RB0        // led mode demarage/arret
32:
33:
34: #define BASE_CLK      (2000000) // quartz 8MHz divisée par 4
35: #define PRESCALER     (1)
36: #define PRESCALERT1   (2)
37: #define NB_TICKS      ((BASE_CLK/PRESCALER)/256)
38: #define TICK          ((PRESCALER*1000000)/BASE_CLK) // en 1/10µsec
39: #define TICKT1        ((PRESCALERT1*1000000)/BASE_CLK) // en 1/10µsec
40:
41:
42: //MOTEURS
43: #define MOTGauche      RA3        // pwms moteurs
44: #define MOTDroit       RA1
45: #define sensMOTGauche  RA2        // sens moteurs
46: #define sensMOTDroit   RA0
47:
48: #define TRANCHE_MOT    0x10        // T du PWM moteur(2^x)
49: #define MASK_MOT       (TRANCHE_MOT - 1) // masque pwm pour 64 tranches
50: #define MAX_MOT        (TRANCHE_MOT*4) // pourquoi pas 4 ?!
51:
52: // SRF04
53: #define DIST_STOP      10        // distance d'arrêt
54: #define DIST_DANGER    50        // distance danger
55: #define DIST_OK        100       // pour init
56: #define LEDCOL         RB1        // led colision (flash=danger, on=stop)
57:
58: #define TRIGGER         RB3        // telemetre ultra-sons
59: #define PULSE          RB7
60:
61: #define SONAR_SEC       (3)        // nombre demandes mesures sonar par sec
62: #define NB_SONAR        (NB_TICKS/SONAR_SEC) // pour nb mesures par sec
63: #define FLASH_SEC       (2)        // nb de flashes par sec
64: #define NB_FLASH        (NB_TICKS/(FLASH_SEC*2)) // pour FLASH_SEC flashes par sec
65:
66:
67: // vitesse optos
68: #define NB_VIT         (NB_TICKS/10) // nb de mesures par sec
69:
70: //defines pour interpretation lecture ligne
71: #define NBCASMOT        7
72:
73: // NEW defines
74:
75: #define LAST            0xff       // garder la valeur actuelle
76: #define LLAST           0x0fff     // garder la valeur actuelle
77:
78:
79: // definition des quotients Vext/Vint f(rayon, empattement)
80: // QRx est applique pour la correction de direction f(ligne)
81: // le resultat est multiplie par 10 !!
82: // Q= (Rayon + (empattement/2)) / (Rayon - (empattement/2))
83: // car Vext= 2*PI*(Rayon + (empattement/2)) * nb_tour_robot/sec
84: // et Vint= 2*PI*(Rayon - (empattement/2)) * nb_tour_robot/sec
85:
86: #define QRmini ( ((Rmini*100)+((EMPATTEMENT*100)/2)) / ((Rmini*10)-((EMPATTEMENT*10)/2)) )
87: #define QRmoyen ( ((Rmoyen*100)+((EMPATTEMENT*100)/2)) / ((Rmoyen*10)-((EMPATTEMENT*10)/2)) )
88: #define QRmaxi ( ((Rmaxi*100)+((EMPATTEMENT*100)/2)) / ((Rmaxi*10)-((EMPATTEMENT*10)/2)) )
89: #define QRinfini (1)
90:
91:
92: // Vext = (2*vitesse*Q) / (1+Q)

```

```

93: // Vint = (2*vitesse) / (1+Q)
94: // car Vext/Vint=Q et Vext+Vint=2*vitesse
95:
96: // Vitesses int et ext f(quotient)
97:
98: #define VeRmini      ( (2*VITROBOT*QRmini)/(10+QRmini) )
99: #define ViRmini      ( (2*VITROBOT*10)/(10+QRmini) )
100: #define VeRmoyen     ( (2*VITROBOT*QRmoyen)/(10+QRmoyen) )
101: #define ViRmoyen     ( (2*VITROBOT*10)/(10+QRmoyen) )
102: #define VeRmaxi      ( (2*VITROBOT*QRmaxi)/(10+QRmaxi) )
103: #define ViRmaxi      ( (2*VITROBOT*10)/(10+QRmaxi) )
104: #define VRinfini     (VITROBOT)
105:
106: // T de T1 associees
107: #define TeRmini      ( (314159*DIARQUE/REDUCTEUR)/(NBT1*VeRmini) )
108: #define TiRmini      ( (314159*DIARQUE/REDUCTEUR)/(NBT1*ViRmini) )
109: #define TeRmoyen     ( (314159*DIARQUE/REDUCTEUR)/(NBT1*VeRmoyen) )
110: #define TiRmoyen     ( (314159*DIARQUE/REDUCTEUR)/(NBT1*ViRmoyen) )
111: #define TeRmaxi      ( (314159*DIARQUE/REDUCTEUR)/(NBT1*VeRmaxi) )
112: #define TiRmaxi      ( (314159*DIARQUE/REDUCTEUR)/(NBT1*ViRmaxi) )
113: #define TRinfini     ( (314159*DIARQUE/REDUCTEUR)/(NBT1*VRinfini) )
114:
115:
116: #define Vinverse      10000
117:
118: // Usefull Defines
119:
120: #define SET           1
121: #define RESET         0
122: #define CLEAR         0
123: #define TRUE          1
124: #define FALSE         0
125: #define ON            1
126: #define OFF           0
127: #define LEDON         1
128: #define LEDOFF        0
129: #define FOREVER       1
130: #define NEVER         0
131: #define OUI           1
132: #define NON           0
133: #define AVANT         0
134: #define ARRIERE       1
135:
136:
137: // Prototypes f()
138:
139: //r2006
140: extern void Init(void);
141: static void interrupt isr(void);
142: //srf04
143: void Init0_Sonar(void);
144: void Init_Sonar(void);
145: void Dem_Mes_Sonar(void);
146: void Conv_Mes_Sonar(void);
147: void TMR0_Sonar(void);
148: void int_Sonar(void);
149: void Gere_Led_Sonar(void);
150:

```

