// Je lis mon capteur : PROT : uint8\_t detection\_presence(void) ;

// S'il y a quelqu'un => je dois ouvrir :

// Si la donnée capteur > valeur seuil et que le portail est fermé alors j'actionne mon moteur pour ouvrir et mes leds doivent clignoter et le buzzer doit sonner et l'afficheur doit afficher OPENING : PROT : void cycle\_OPEN(void) ;

// Si le portail atteint sa position ouverte on arrête les moteurs et on démarre la temporisation : PROT : void wait(uint8\_t tmp) ;

// Si fin tempo et on ne détecte pas d'obstacle on actionne le moteur pour fermer et mes leds doivent clignoter et le buzzer doit sonner et l'afficheur doit afficher CLOSING : PROT : void cycle\_CLOSE(void) ;

// Si le portail atteint sa position fermée on arrête les moteurs

// Si le portail n'a pas encore atteint sa position fermée et que l'on détecte un obstacle on réouvre la porte et mes leds clignotent : PROT : void cycle\_OPEN(void) ;

// Si pendant la réouverture on détecte un obstacle on arrête les moteurs et on démarre la temporisation : PROT : void wait(uint8\_t tmp) ;

// Si fin tempo et pas d'obstacle on ferme et mes leds clignotent : PROT : void cycle\_CLOSE(void) ;

// Si fin tempo et on détecte un obstacle on relance la tempo : PROT : void wait(uint8\_t tmp) ;

// Si fin tempo et on détecte un obstacle on relance la tempo : PROT : void wait(uint8\_t tmp) ;

// Si le portail n'a pas encore atteint sa position ouverte et que l'on détecte un obstacle on referme et mes leds clignotent : PROT : void cycle\_CLOSE(void) ;

// Si pendant la re fermeture on détecte un obstacle on arrête les moteurs et on démarre la temporisation :

PROT : void wait(uint8\_t tmp) ;

// Si fin tempo et pas d'obstacle on réouvre et mes leds clignotent : PROT : void cycle\_OPEN(void) ;

// Si fin tempo et on détecte un obstacle on relance la tempo : PROT : void wait(uint8\_t tmp) ;

Avec :

* uint8\_t detection\_presence(void) ;
* void cycle\_OPEN(void) ;
  + LED\_CLIGNOTEMENT() ;
  + afficheur(uint8\_t mot) ;
  + sonnerie() ;
  + MOT\_SENS1(uint8\_t sens1) ;
  + MOT\_SENS2(uint8\_t sens2) ;
* void cycle\_CLOSE(void) ;
  + LED\_CLIGNOTEMENT() ;
  + afficheur(uint8\_t mot) ;
  + sonnerie() ;
  + MOT\_SENS1(uint8\_t sens1) ;
  + MOT\_SENS2(uint8\_t sens2) ;
* void wait(uint8\_t tmp) ;