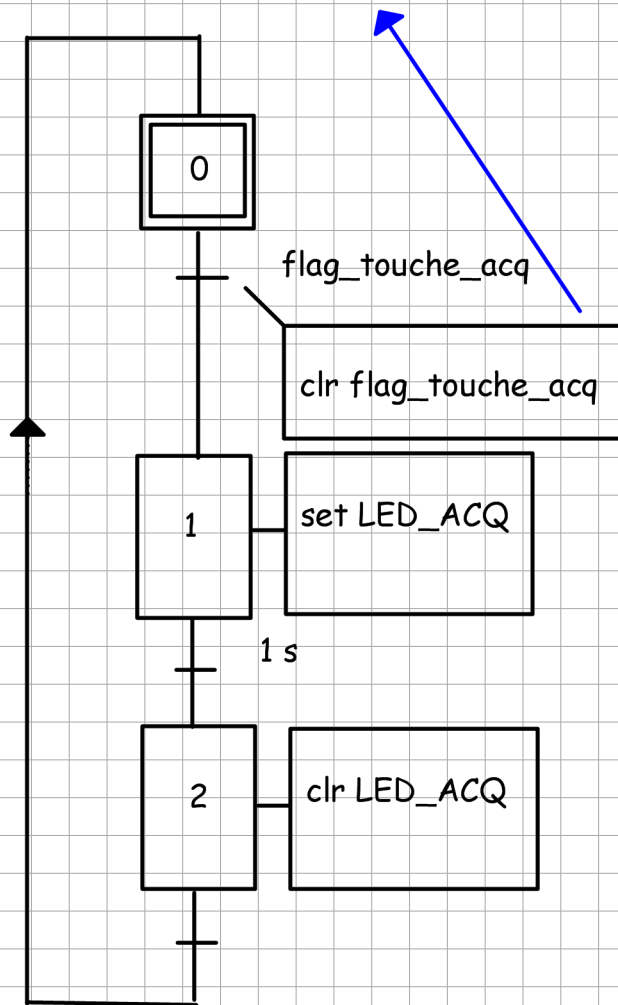


comment traduire en C une notion d'attente exprimée dans un grafcet ?

acquittement de la demande par
une action au franchissement



les instructions "delay" que l'on trouve dans le monde Arduino sont à proscrire car le processeur va être bloqué en attente, incapable d'exécuter d'autres tâches en parallèle (en pseudo parallèle)

On remplace la problématique d'attente bloquante par un concept d'état d'attente non bloquant.

On détecte la fin d'attente avec un simple test.

La durée d'attente est décomptée dans une IT qui donne la mesure du temps écoulé pour toutes les fonctions ayant un besoin de délais

Le programme change d'état à la fin de l'attente programmée.

```
volatile unsigned int attente = 0, attente2 = 0;
void it_timer(void) __irq //toutes les ms
{ if(attente) {attente --;}
  if(attente2) {attente2 --;}
  // pour une autre fonction
  TxIR=1; VicVecAddr = 0;
}
void gere_temoin_acquittement(void)
{ unsigned char etat_mae_temoin = 0;
  switch (etat_mae_temoin)
  { case 0:
    if(flag_touche_acqu != 0)
    { flag_touche_acq=0;
      attente = 1000;
      etat_mae_temoin++;
    }
    break;
    case 1:
    set_led_acq();
    if (attente == 0) {etat_mae_temoin++;}
    break ;
    case 2:
    clr_led_acq();
    etat_mae_temoin = 0;
    break;
  }
}
```

ce code n'est pas optimisé, c'est l'objet de la page suivante...

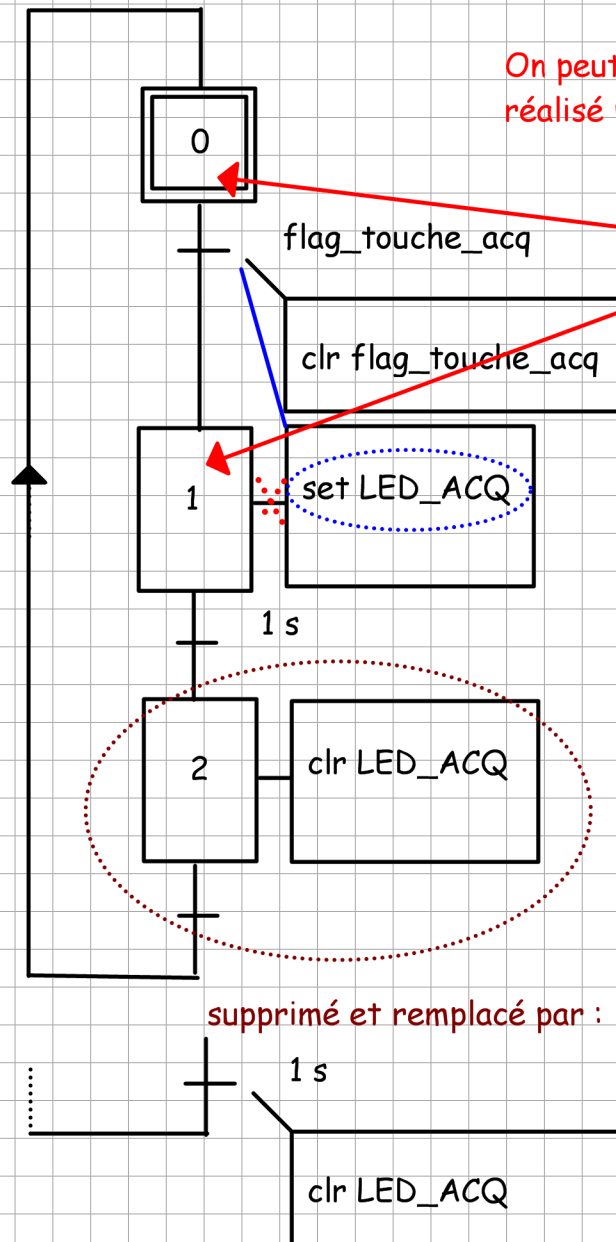
comment réduire un code de machine à état très simple?

comment économiser des cycles CPU ?

On peut remplacer systématiquement un état qui n'est exécuté qu'une fois, par une action au franchissement :

On peut aussi remplacer une action dans un état par une action au franchissement toute action qui n'a besoin d'être réalisée qu'une fois.

```
volatile unsigned int attente = 0;
void it_timer(void) __irq
{ if(attente) {attente --;}
  TxIR=1; VicVecAddr = 0;
}
void gere_temoin_acquittement(void)
{ unsigned char etat_mae_temoin = 0;
  if (! etat_mae_temoin) // etat 0
  { if(flag_touche_acqu != 0)
    { flag_touche_acq=0;
      attente = 1000;
      etat_mae_temoin++;
      set_led_acq();
    }
  }
  else // etat 1
  { if (attente == 0)
    { clr_led_acq();
      etat_mae_temoin = 0;
    }
  }
}
```



On peut se poser la question d'une machine à état implicite réalisée par la valeur d'une variable : variable attente ici

attente = 0
attente <> 0

```
volatile unsigned int attente = 0;
void it_timer(void) __irq
{ if(attente) {attente --;}
  TxIR=1; VicVecAddr = 0;
}
void gere_temoin_acquittement(void)
{ if(attente != 0) {return ;}
  clr_led_acq();
  // le clear ne peut plus être au franchissement
  if(flag_touche_acqu != 0)
  { flag_touche_acq=0;
    attente = 1000;
    set_led_acq();
  }
}
```