

Si hay constantes, definición estructuras, etc. necesarias en varios .c :

- Declarar en un .h
- Todos los .c que las necesitan incluyen a ese .h

Si hay variables globales necesarias en varios .c :

- Uno de ellos las declara “normal”, y si lo desea las inicializa:

```
uno.c
```

```
int _modo_control=MODO_CONTROL_REF;  
...
```

- El resto las declara “extern”, y no las puede inicializar:

```
otro.c
```

```
extern int _modo_control;  
...
```

Esquema solución ampliación 1:

La extracción de la tabla se realiza llamando de forma iterativa a la función strtod()

```
float tabla[TAM_MAX];
```

```
char texto="RZ=[.....]";
```

```
char *ptStart,*ptEnd;
```

```
ptStart=GetArgComando("RZ");
```

```
if (ptStart!=NULL)
```

```
{
```

```
    comprobar *ptStart == '['. En caso afirmativo, ptStart++
```

```
    Repetir mientras *ptStart no es ']', incrementando i en 1
```

```
        tabla[i]=strtod(ptStart,&ptEnd);
```

```
        ptEnd=Saltar espacios y comas tras ptEnd
```

```
        ptStart=ptEnd
```

```
}
```

Esquema solución ampliación 2: (ver documentación curses en página del trabajo)

2.1) Variables globales tipo WINDOW* para cada una de las ventanas (se usan en main y en la función de interrupción)

```
WINDOW *_ventana_cmd, *_ventana_estado, .... ;
```

2.2) Inicializar curses y ventanas en main()

```
main()
{
    ...
    Llamadas a funciones para iniciar curses

    // Creacion de una ventana
    _ventana_cmd=newwin(.....);
    wattro(_ventana_cmd, colores para la ventana);
    resto de inicialización de la ventana (borrar, poner marco, etc.)
    wrefresh(_ventana_cmd);

    ... Resto de inicializaciones
}
```

Esquema solución ampliación 2:

2.3) Escribir en una ventana cuando sea necesario

```
...  
wmove(id ventana, posición deseada del cursor en la ventana);  
wclrtoeol(id ventana); // Si se desea borrar hasta fin de linea  
wprintw(id ventana, resto como en printf);  
...  
wrefresh(id ventana);
```

2.4) Esperar por cadena en una ventana

```
...  
wmove(id ventana, posición deseada del cursor en la ventana);  
wclrtoeol(id ventana); // Si se desea borrar hasta fin de linea  
wprintw(id ventana, resto como en printf);  
wrefresh(id ventana);  
wgetstr(id ventana, cadena de caracteres);
```

Esquema solución ampliación 3:

3.1) Obtención de la posición en grados a partir del encóder

```
int enc_count=Simulator_ReadCounter(0); // Leer contador encóder 0  
posk = conversión de unidades enc_count a grados, teniendo en cuenta que:  
    enc = 64 pulsos x 4 cuentas/pulso  
    enc está asociado al eje motor. El eje de salida se mueve con relación 1:8
```

3.2) Derivación de la posición para obtener la velocidad

```
Velk=(posk-posk-1)/Tm (Ojo unidades deg/ms no son rpm)
```

Esquema solución ampliación 4:

*La solución planteada es sólo a modo de referencia.
No sigue perfectamente la especificación de la ampliación.*

4.1) Variables globales necesarias

```
int _modo_limpia -> TRUE/FALSE
int _time_last_activation_SW0_ms -> tiempo desde la última activación del pulsador 0
int _count_activation_SW0_3s -> número de activaciones del pulsador 0 en los últimos 3 segundos
```

4.2) En función de interrupción:

```
...
if (SW0 ha cambiado de 0 a 1) // SW0 recién pulsado → inicializar tiempo activación
{
    _time_last_activation_SW0_ms=0;
}
else if (SW0 se mantiene en 1) // SW0 estaba pulsado y continúa pulsado → incrementar tiempo activación
{
    _time_last_activation_SW0_ms+=Tm_ms;
}
else if (SW0 ha cambiado de 1 a 0) // SW0 recién liberado → comprobar comando según duración
{
    _count_activation_SW0_3s++;
    Según valores de _count_activation_SW0_3s Y _time_last_activation_SW0_ms:
    ACTIVAR LIMPIA → _modo_limpia=TRUE; control=MODO POS con REF teclado; _ref_teclado=70;
    DESACTIVAR LIMPIA → _modo_limpia=FALSE; _count_activation_SW0_3s=0;
}
...
```

Esquema solución ampliación 4:

4.2) Continuación

```
...
case MODO_CONTROL_POS:
  if (_modo_limpia)
  {
    if (posk[0] > 69.5)
      ref teclado = -70;

    if (posk[0] < -69.5)
      ref teclado = 70;
  }
  .... Resto del lazo de control, el control de posición hará el trabajo ....
}
```