

Punteros a funciones

Punteros a Funciones

En el enunciado de la [práctica 1](#) se indica, para varias librerías, la necesidad de

- instalar función manejadora ...

Se quiere es que cuando se produzca un determinado evento (*salte* una interrupción) se llame a una función proporcionada por el usuario.

No queremos que el usuario tenga que definir directamente la rutina de tratamiento de interrupción como figura en el apartado 2.5.2 del documento de [compilación en C](#). Debemos ser nosotros los que, en nuestra librería, definamos esa rutina de tratamiento, y desde ella llamemos a la función que el usuario nos indico (previamente).

Sabemos que de una función podemos obtener un puntero a la misma a través de su nombre. Este puntero debemos almacenarlo en una variable para poder, posteriormente, invocar a la función cada vez que salte la interrupción.

En [este artículo](#) se explica como hacer esto.

Ejemplo 1

Si tenemos una función con prototipo:

```
void funcion1(void);
```

La variable que pueda almacenar la dirección de esta función se declarará con:

```
void (*punteroFuncion1)(void);
```

Para darle valor al puntero podemos hacer:

```
punteroFuncion1 = funcion1;
```

aunque queda más claro claro si le damos valor con:

```
punteroFuncion1 = &funcion1;
```

Recibiremos el puntero como parámetro de una función con:

```
int instalaManejador(void (*ptFun)(void)) {  
    punteroFuncion1 = ptFun;  
}
```

Para invocar la función a partir del puntero podemos hacer:

```
punteroFuncion1();
```

aunque queda más claro si hacemos:

```
(*punteroFuncion1)();
```

Ejemplo 2

Si tenemos una función con prototipo:

```
void funcion2(uint8_t numDatos, uint16_t* buffer);
```

La variable que pueda almacenar la dirección de esta función se declarará e inicializará así:

```
void (*punteroFuncion2)(uint8_t, uint16_t*);
```

```
punteroFuncion2 = &funcion2;
```

y se usará con

```
(*punteroFuncion2)(7, buffer);
```

Última modificación: jueves, 19 de noviembre de 2020, 12:43