TALLER N° 1: Taller guía.

Considere el siguiente problema:

Se necesita un sistema que encienda un motor a través del uso de una llave. Cuando la llave se hace girar hacia algún sentido el motor empieza a funcionar, pero si la llave se deja en su posición inicial, el motor se apaga.

Solución:

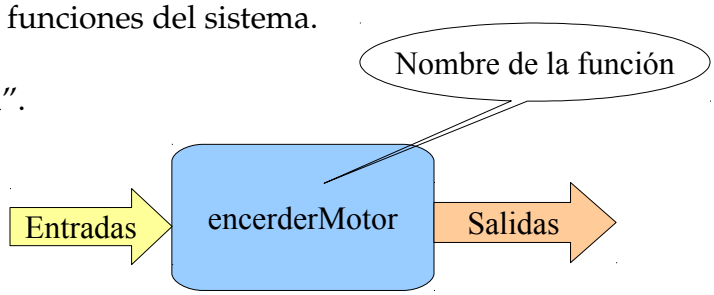
Siempre se hará de la siguiente manera.

1. ¿qué hace el sistema?.
2. Función "caja negra".
3. Variables de Entradas.
4. Variables de Salidas.
5. Tipo de variables tabla de estados.
6. Diagrama de flujo.
7. Función escrita en lenguaje de alto nivel.
8. Uso de las interfaces de el micro-controlador.

- Identificamos inicialmente **¿qué hace?** El sistema.

Este sistema lo que hace es **encender** un motor, es decir, que esta es la **función** principal. De esta manera se identifica las funciones del sistema.

- Función "**Caja Negra**".



- Identificación de **salidas** del sistema.
  En este caso se trata de encontrar los actuadores; los actuadores son aquellos que generan luz (como leds, bombillos, lámparas, pantallas), sonido (Altavoces, buzzer, alarmas, audífonos), movimiento (Motores, electro-imanes, bandas transportadoras), calor (hornos, estufas, resistencias, calentadores).

  Con ello identificamos que la salida del ejemplo es el motor.

- Identificación de **entradas** del sistema.
  Las entradas del sistema son aquellas que control inspecciona, analiza, observa, tiene

en cuenta para procesar y generar el estado de una salida.
En general las entradas son llaves, interruptores, pulsadores, teclados, sensores de temperatura o movimientos o de sonido.

Con ello identificamos que la entrada del sistema es la llave.

- **Tipo de variables.**
  Ya hemos hablado de las entradas y las salidas. Ahora tomaremos en cuenta que las entradas y las salidas son **variables.** Se llaman así debido a que estas cambian dependiendo de estados o elementos externos.

  Ya hemos hablado que tenemos por ahora dos tipos de variables
  - Bit : que solo tiene dos estados 1 o 0
  - Byte: el cual son 8 bits, lo que nos permite generar 8 entradas o salidas diferentes.

Ahora para conocer el tamaño de las variables se hacen una **tabla de estados**.

ENTRADAS	***SALIDAS***
Llave	***Motor***
Llave sin mover	apagado
Girando la llave	Encendido

La llave solo tiene dos estados, los cuales se pueden identificar con un solo bit.

Llave	bit
Sin mover	0
Girarndo	1

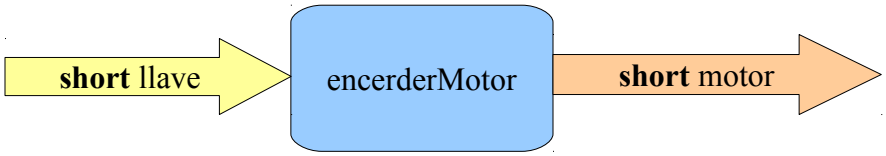
Lo mismo sucede con la variable de salida motor.

Motor	bit
apagado	0
Encendido	1

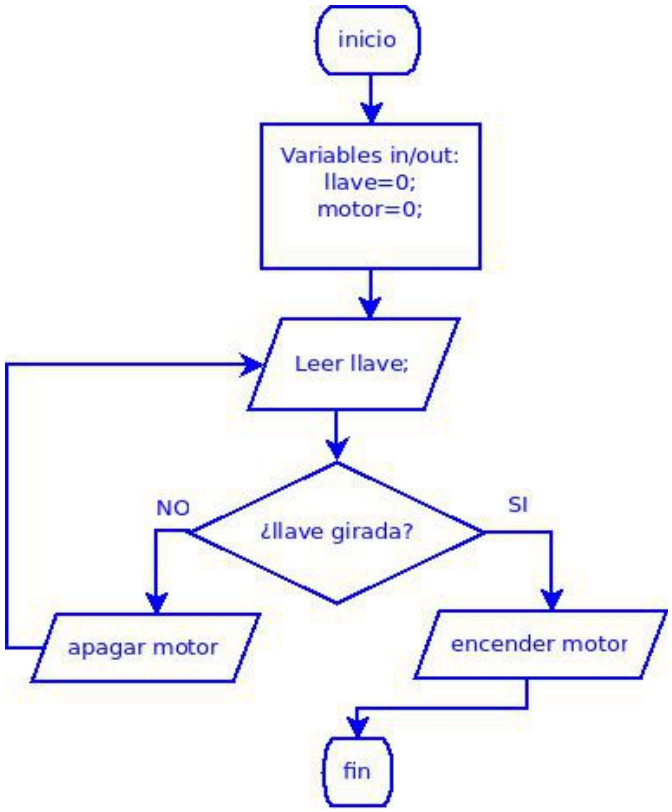
Ahora para completar la función encenderMotor “caja negra” escribimos el tipo de la siguiente manera:

- Cuando es un bit entonces la escribimos de tipo **short** nombreVariable.
- Cuando es un byte, entonces escribimos el tipo **int** nombreVariable.

  En este caso ambas variable de entrada y de salida son de tipo bit.

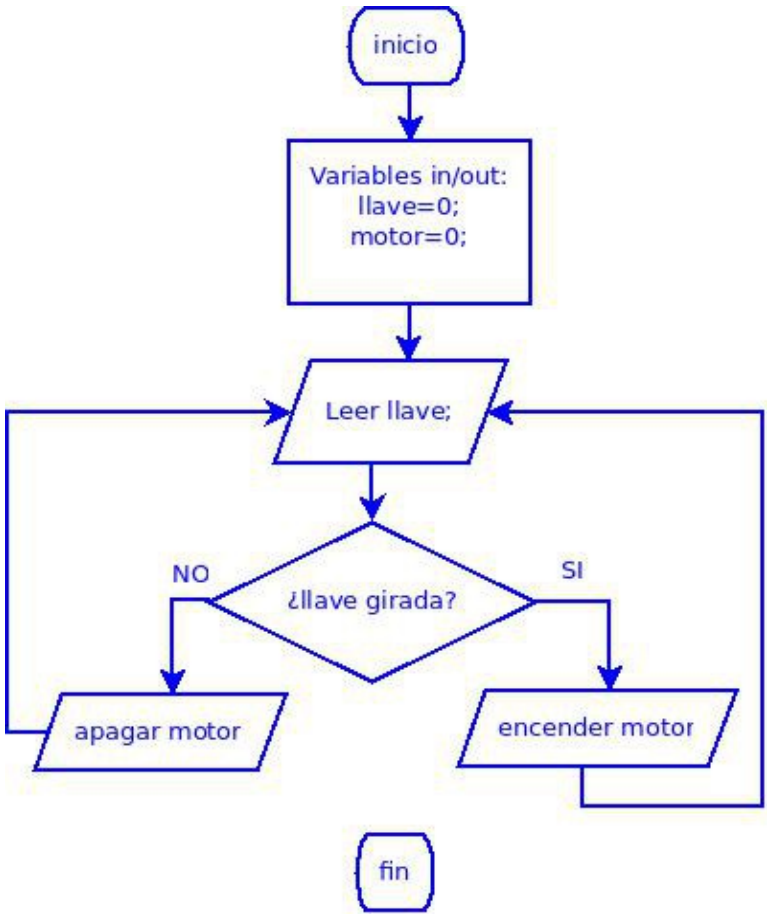


- **Diagrama de Flujo**: En el diagrama de flujo se identifica el algoritmo de la función.



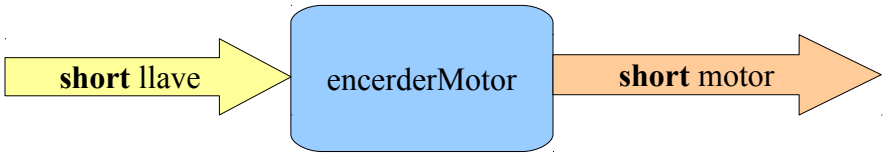
Como el algoritmo soluciona el problema y queremos que lo pueda hacer siempre entonces lo convertimos en un ciclo repetitivo, impidiendo que el flujo llegue al fin.

johnnycubides@yahoo.com



- Función lenguaje en alto nivel.

De la caja negra sacamos lo que necesitamos para escribir la función en lenguaje de alto nivel.




```
short encenderMotor(short llave){        //Función que enciende un motor.
        if(llave==1){                    //Pregunta si llave es igual a uno, devuelve falso o verdadero.
        return 1;                        //Si llave igual a uno(verdad) retorna 1, "encender motor".
        }
        else{                            //si llave igual a cero(falso) retornar 0, "apagar motor".
        retur 0;
        }
}
```

johnnycubides@yahoo.com

de esta manera se hizo la función que es capaz de encender o apagar un motor.

EJERCICIO N °1
Escribir en PIC C el programa guía del sistema que enciende un motor y compilarlo.

Programa escrito en lenguaje de alto nivel

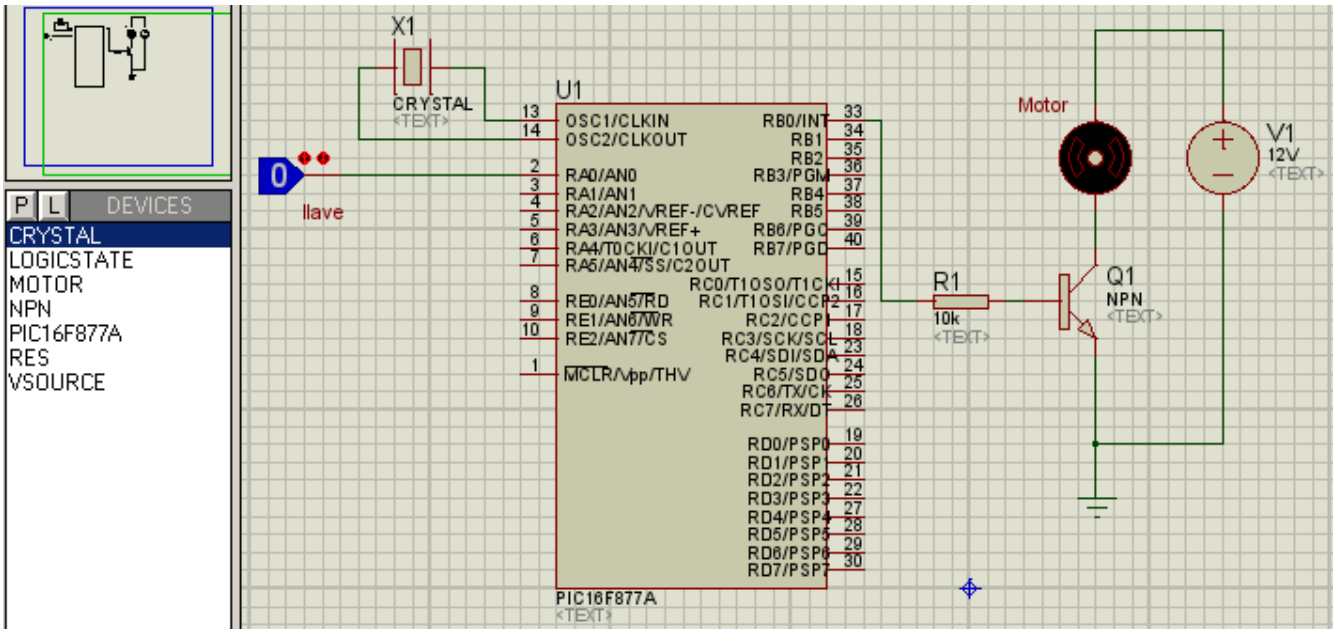
```
#include <16F877a.h> /*Se incluye la librería del dispositivo*/
#fuses XT,PUT,NOWDT /*Aqui se configura tales como el perro guardian y, el tipo de reloj*/
#use delay (clock=4M) /*Configuración de la velocidad del reloj*/

#byte port_a=0x05
#byte port_b=0x06
#byte port_c=0x07
#byte port_d=0x08
#byte port_e=0x09

#byte entrada=0xff
#byte salida=0x00
#byte apagado=0x00
#byte encendido=0xff

//Variables Glovales
short llave=0;//entrada
short motor=0; //salida

//Funciones Auxiliares
short encenderMotor(short llave){       //función que enciende un motor
        if(llave==1){   .       .       //La llave fue girada?
        return 1;       .       .       //si encianda
        }
        else{   .       .       .       //sin girar
        return 0;       .       .       //apague
        }
}
```

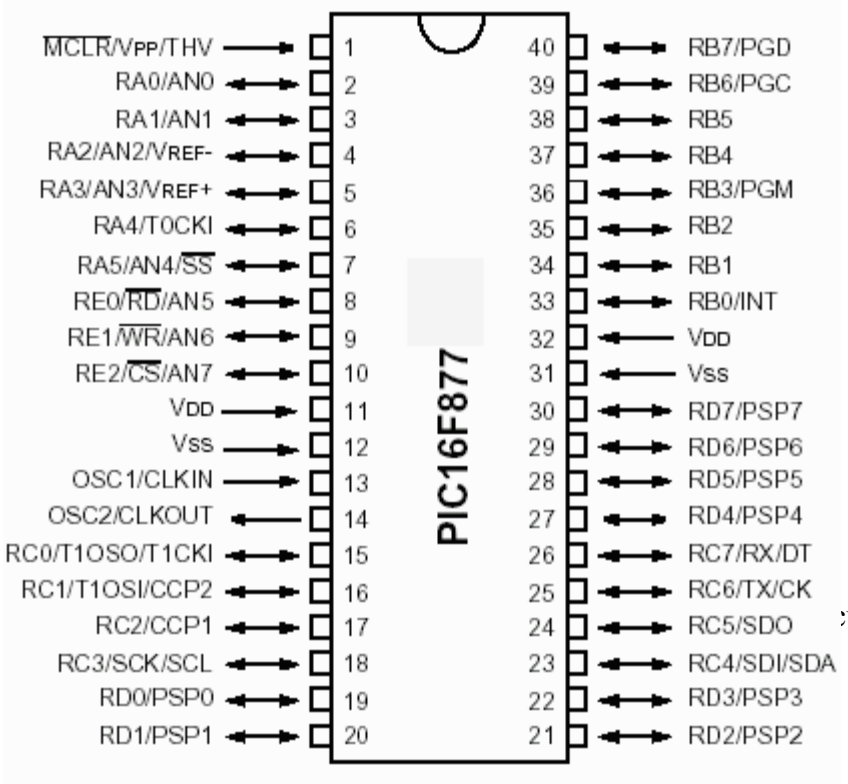
```
//      Inicio de función principal
void main(){
set_tris_a(entrada); .  //configura el puerto_a como entrada
set_tris_b(salida);     //Configura el puerto_b como salida

port_b=(apagado);

while(true){ //Unicio de ciclo repetitivo
        llave=input(pin_a0);    //Lee la llave
        motor=encenderMotor(llave);     //procesa el valor de la llave y lo guarda en motor
        output_bit(pin_b0, motor);      //sale el valor por el pin_bo segun sea motor
}
}
```

johnnycubides@yahoo.com

EJERCICIO N° 2

Hacer el siguiente esquema en ISIS PROTEUS y del ejercicio uno usar el archivo .hex generado para simular el sistema que enciende un motor como se muestra en el tutorial de internet.





cubides@yahoo.com