### Alumno: Miguel Azar

### Plataforma embebida: EDU-CIAA-NXP

### Aplicación:

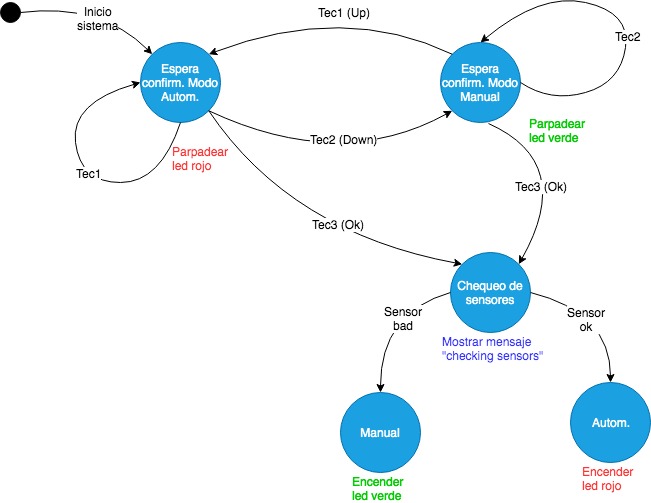
Configuración del sistema de control de temperatura y humedad para estufas de tabaco. En este módulo se solicita al usuario seleccionar entre **Modo Automático** y **Modo Manual**. Por defecto el sistema queda a la espera de confirmación del **Modo Automático**. El usuario dispone de 3 pulsadores: **Up, Down** y **Ok**. Una vez que acepta la opcion con **Ok** el sistema chequea el estado de los sensores. Si los sensores tienen fallas y se eligió **Modo Automático** entonces el sistema pasa a **Modo Manual**.

### 

### Periféricos (1 ó 2):

UART, GPIO

### Diagrama de estado de MEF con una breve descripción de cada estado.



### 

### Incluir una breve descripción de cada estado

### Espera confirmación Modo Autom.: El sistema espera a que el operador presione algunos de los 3 pulsadores: Ok, Up o Down. A su vez se mantiene un parpadeo del led verde. Este led indica que por defecto el sistema se ejecutará en Modo Automático.

### Espera confirmación Modo Manual: Si el usuario eligió Down entonces el sistema espera confirmacion de que el proceso se ejecutará en Modo Manual. Si estando en este modo el operador presiona Ok entonces el sistema pasa al estado Chequeo de sensores.

### Chequeo de sensores: En este estado el sistema verifica que los sensores esten enviando datos válidos. Esto se realiza debido a que los sensores se encontrarán instalados en un lugar propenso a daños y lógicamente el Modo Automático trabaja con el apoyo de los sensores.

### Manual: Este estado enciende el led rojo en forma permanente y desde este modo (en futuras versiones) correrán algoritmos que se encargarán de funciones básicas tales como indicar las mediciones de el resto de los sensores tales como Sensor de Humo y de Gas.

### Automático: Es el estado en donde el sistema inicia su rutina de control. Estas rutinas no seran desarrolladas por el momento.

### Definir los módulos de software (archivos) que va implementar para cada periférico.

controlDebounce.c y controlDebounce.h: Administración de teclas.

mensajesUART.c y mensajesUART.h: Comunicación por periférico UART de los mensajes al operador.

sensoresGPIO.c y sensoresGPIO.h: Lectura de los sensores integrados de temperatura y humedad. En esta primer versión se usará DHT11 pero se prevé emplear el SI7021.

**Definir los prototipos de las principales funciones públicas y privadas de cada módulo definido .**

void switchState(state\_t pulsador);

void updateMessage(message\_t texto);

int8\_t dataSensors(int8\_t idSensor);