### Alumno: Luis David Díaz Charris

### Plataforma embebida: STM32F401VD

### Aplicación:

Leer un teclado matricial 4x4. Cada vez que se presione una tecla, ésta se debe transmitir a la UART indicado tecla y tipo de evento (presionado o liberado).

### Periféricos (1 o 2):

UART y GPIO.

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

Diagrama

Descripción generada automáticamente

### Incluir una breve descripción de cada estado

* **INICIO/REINICIO:** es el estado por defecto con el que se inicia la máquina. Se espera que las teclas estén liberadas y la bandera disponible en 1, para recibir datos.
* **REGISTRO DE ENTRADA:** en este estado una está siendo presionada por el usuario, se almacena en un buffer, se valida la tecla presionada, se envía a escritura y la bandera disponible se coloca en 0.
* **ESCRITURA UART:** se recibe el carácter de la tecla presionada y se usan las funciones para escribir en la UART. En este estado se escribe la tecla y el estado de la misma, se habilita la bandera disponible para esperar otro dato.

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

API\_KEYPAD.c y API\_KEYPAD.h para la lectura del teclado matricial 4x4 y el anti-rebote de sus teclas.

API\_controlUart.c y API\_conrolUart.h para el control de escritura de teclado, código u otro periférico en la UART-

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

**API\_controlUart**

* ***typedef uint8\_t mensaje\_t*** // esta variable permite enviar un mensaje de máximo 8 bits de longitud.
* ***void uartInit(void)*** // Esta función inicializa el periférico UART6 de mi micro con un baudrate de 9600, sin bit de paridad, sin control de flujo, con una longitud de palabra de 8bits y un bit de stop.
* ***void uartPrintSerial( mensaje\_t mensaje)*** // Esta función permite imprimir un mensaje por la terminal serie.
* ***typedef enum{DISPONIBLE = 0, REGISTRO\_ENTARDA, ESCRITURA\_UART} uartState;*** // esta es la enumeración de los estados de la máquina de estados para controlar la escritura en la UART.

**API\_keypad**

* ***typedef uint8\_t tecla\_t*** // esta variable permite enviar un mensaje de máximo 8 bits de longitud.
* ***keypadInit()*** // Inicializa la configuración de pines GPIO para conectar el teclado matricial.
* ***tecla\_t keypadRead()*** // Excita los pines de escritura y consulta los estados de los pines de lectura conectados al teclado. Si recibe una señal, según sea el código binario de éste, esa será la tecla presionada. Devolverá la tecla presinada con su correspondiente código ASCII.