



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

Project 2: MicroPython and STM32-F411RE

Jesús Gamero Tello

02/06/2020

INDICE

Explicación del proyecto y funciones	3
Componentes utilizados	4
Fritzing	4
Imagen de conexiones	5

Explicación del proyecto y funciones

Mi practica de micropython está basada en el módulo visto en clase, A9: Mover un servomotor de acuerdo con el ángulo de un potenciómetro. Para lo cual necesitaremos un potenciómetro y un servomotor. También he decidido añadir un botón para cubrir la parte de FSM ya que sin el botón no veía posible cubrirlo.

Para lo cual hemos usado las funciones de la librería de micropython ADC, Timer y Pin.

- **La clase ADC** nos permite la conversión analógica a digital, he configurado el pin PA1 como ADC, en este pin irá enchufado el potenciómetro el cual nos permitirá girar el servo.
- Con **la clase Timer** he configurado dos timers el primero el cual es esencial para poder mover el servo y el segundo para hacer uso de las interrupciones, la interrupción de este último saltara cuando llegue al preescaler del primero cogiendo así el valor en ese momento del pin configurado como ADC.
- Con **la clase Pin** se ha configurado un pin extra para implementar un FSM sencillo del que se hablara posteriormente

También he usado la función `micropython.alloc_emergency_exception_buf()` para “reservar” 100 bytes de emergencia en la RAM, ya que es recomendable usarlo cuando se usan interrupciones.

Mediante `pyb.freq` he cambiado la frecuencia del reloj (sysclock) a 84 MHz para poder dejar los valores de prescaler y el period del timer iguales a los que usábamos en la practica en C. También quiero comentar que era posible dejar la frecuencia de la placa sin modificar, pero modificando el prescaler el cual se me quedaba a 1919.




En cuanto a la formula usada para mover el servomotor, se ha usado la misma que se uso en la practica hecha en C.

```
formula=(val*100)/4095
grados=formula+25
ch.pulse_width(int(grados))
```

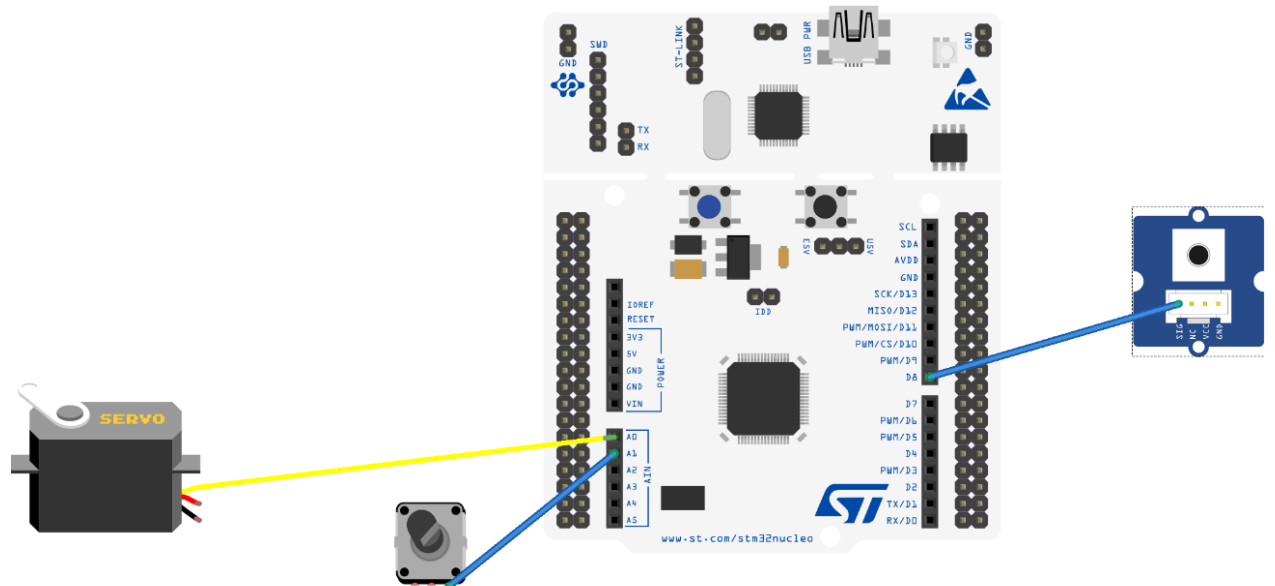
En cuanto al FSM, lo he realizado mediante dos casos:

- **Primer caso:** Si no se pulsa el botón la aplicación funcionara con normalidad, es decir, si se gira el potenciómetro se girará el servomotor
- **Segundo caso:** Si se pulsa el botón el servomotor regresara a la posición inicial durante un periodo de 5 segundos independientemente de donde se encontrase en el momento de pulsar el botón, posteriormente después de esos 5 segundos regresara a la posición que marque el potenciómetro

Componentes utilizados

Componente	Pin	Imagen
botón	PA9	
Servomotor	PA0	
Potenciómetro	PA1	

Fritzing



fritzing

Imagen de conexiones

