

Movimiento Servos desde MBED

Juan David Sandoval

Juandsandoval8@gmail.com, Ingeniería Electrónica

Resumen—Las tarjetas STM tiene variedad de usos y aplicaciones una de ellas en el tratamiento de conversores ADC como ancho de pulso (DUTY CYCLE) para mover un servomotor a la posición que deseemos, con algún sensor externo.

Abstract— STM cards have a variety of uses and applications, one of them in the treatment of ADC converters such as pulse width (DUTY CYCLE) to move a servomotor to the position we want, with an external sensor.

1. INTRODUCCIÓN

En este informe se explicará cómo mover nuestros servomotores partir del ADC de un módulo joystick para posicionar nuestro servo a la posición deseada.

2. METODOLOGÍA

A. PROBLEMA

Se ha planteado en clase que se debe mover los dos servomotores de la base “pantilt” a partir la conversión análoga-digital de dos potenciómetros que constituyen el modulo joystick.

B. MATERIALES

- *STM32F411RE
- *Modulo joystick
- *Servomotores

C. COMO USARLOS

- STM32F411RE: Como ya se vio en los anteriores informes y guías usaremos Mbed para programar la tarjeta STM, una vez creado nuestro proyecto trabajaremos el ADC pues será quien moverá nuestro servo.

```
#include "mbed.h"

AnalogIn josx(A0);
AnalogIn josy(A1);
PwmOut Servo_1 (PC_9);
PwmOut Servo_2 (PC_8);
```

- Lo primero que haremos será declarar las entradas análogas que queremos convertir estas pertenecen a las salidas de los dos potenciómetros del módulo joystick.

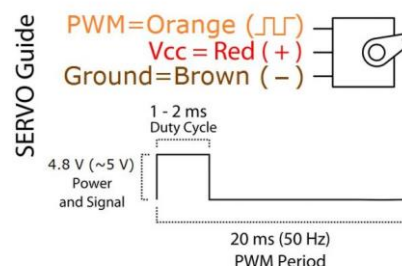


- Lo siguiente será declarar la salida las salidas del tipo PWM, que serán las salidas hacia los servomotores.

- Donde X-axis y Y-axis serán las entradas A0 y A1.

```
int main ()
{
    Servo_1.period_ms(20);
    Servo_1.pulsewidth_us(1000);
    Servo_2.period_ms(20);
    Servo_2.pulsewidth_us(1000);
    uint16_t v_read_x;
    uint16_t v_read_y;
    int x,y;
```

- En el código principal vendrá las variables de inicio de cada servomotor, estas variables (period y pulsewidth), son sacadas de la ficha técnica del servomotor.



- La alimentación de los servos si se conectan desde la tarjeta debe hacerse a 3.3 y si se hace externamente se debe poner a 5

- También se tienen dos variables “v_read_y” e “v_read_x”, esta variable será donde se almacene la conversión análoga-digital de los dos potenciómetros conectados en los puertos A0 y A1 estas estarán guardadas en variables del tipo uint_16t.

- Por último, se tiene dos variables del tipo entero “x” e “y”, que tendrán los valores para mover los servomotores.

```

while(1) {
    v_read_x = josx.read_u16(); // de 0 65535
    x=(v_read_x/65.535)+1000;
    if (x>2000) {
        x=2000;
    }
    if (x<1000) {
        x=1000;
    }
    wait_ms(20);
    Servo_1.pulsewidth_us(x);
    v_read_y = josy.read_u16(); // de 0 65535
    y=(v_read_y/65.535)+1000;
    if (y>2000) {
        y=2000;
    }
    if (y<1000) {
        y=1000;
    }
    Servo_2.pulsewidth_us(y);
}
}

```

- Dentro del ciclo infinito se tiene la conversión analógica a digital, como este dato se recibe de 0 a 65535 y se requiere valores entre 1000 a 2000. Lo que se hará es tratar esa conversión para lograr esos valores y guardarlos en las variables del tipo **int** que se mencionaron anteriormente (x,y).

```

* v_read_x=josx.read_u16();
x=(v_read_x/65.535)+1000;
* v_read_y=josy.read_u16();
y=(v_read_y/65.535)+1000;

```

- Con esto ya se tiene valores entre 1000 a 2000.

Con la sentencia “if” lo único que se hará es decirle que no se pase por encima de 2000 y que no tome valores por debajo de 1000, y así no se tendrán valores que afecten el movimiento del servo teniéndolo dentro del rango establecido.

- Ya por último se tiene las salidas que quedaron almacenadas en “y” y “x” estos valores numéricos tendrán valores entre 1000 y 2000 y serán enviados a él comando pulsewidth y será leído en microsegundos.

```

Servo_1.pulsewidth_us(x);
Servo_2.pulsewidth_us(y);

```

* Una vez hecho esto se compilará la tarjeta y se harán las conexiones correspondientes para los servos, recordando que cada servo debe ir alimentado a 3.3v ya que a ese voltaje trabaja nuestra tarjeta STM.

3. CONCLUSIONES

Se puede concluir que es bastante útil el tratar de mover un servomotor desde un mando manual y acomodarlo a alguna necesidad en común; la base “pantilt” nos permite un movimiento en X y Y lo cual es bastante útil para próximos laboratorios.

5. BIBLIOGRAFIA

- *Docs.zephyrproject.org. (2019). ST Nucleo F411RE — Zephyr Project Documentation. [online] Available at: https://docs.zephyrproject.org/latest/boards/arm/nucleo_f411re/doc/index.html [Accessed 29 Nov. 2019].*
- *Arm Mbed. (2019). Home | Mbed. [online] Available at: <https://www.mbed.com/en/> [Accessed 29 Nov. 2019].*

