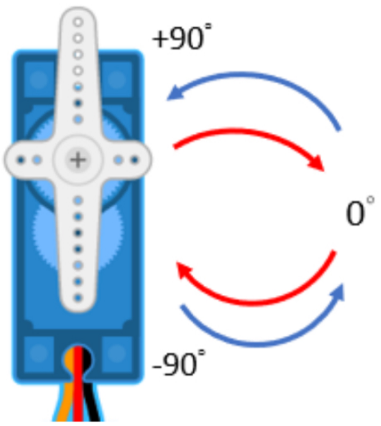
LAB 08 Requirement Description

- **CCP Module**
  - **Video Link:** [Lab08: CCP Module- YouTube]
  - **HackMD Link:** [Lab08: CCP Module- HackMD]

- **Basic(70%)**
  - **Description:**

    Utilize RB0 as a motor control button to control motor rotation according to the following steps:

    1. Set the initial position of the motor to -90°.
    2. When the button is pressed, the motor should rotate 90° counterclockwise.
    3. Once the motor reaches +90°, the rotation direction should reverse to clockwise.
    4. Press the button again when the motor rotates back to -90°, and repeat steps 2 through 4.

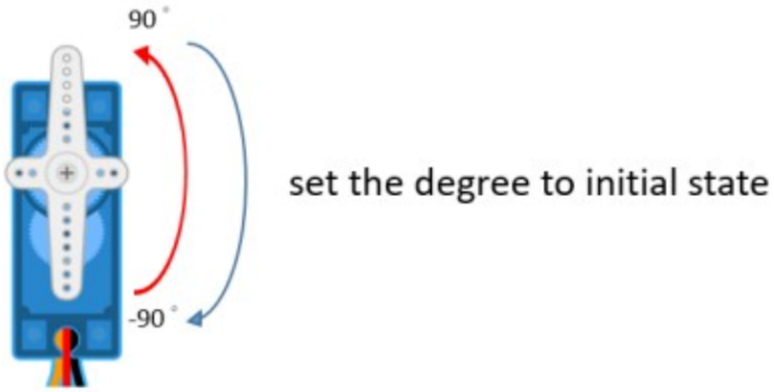


  - **Grading Criteria:**
    1. Ensure that the CCP1CON<5:4> bits are properly handled when configuring the duty cycle.
    2. Both C and assembly language implementations are acceptable.

- **Advanced(30%)**
  - **Description:**

    Utilize RB0 as a motor control button to control motor rotation according to the following steps:

    1. Set the initial position of the motor to -90°.
    2. When the button is pressed, the motor should rotate from -90° to +90°.
    3. Upon reaching +90°, reset the motor position back to the initial state of -90°.
    4. Press the button again to repeat the process from steps 2 through 3.



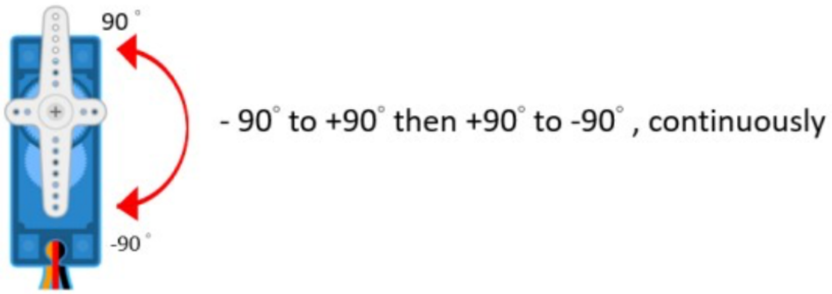
  - **Grading Criteria:**

    1. Do not limit the motor to only -90° and +90° positions. The program should process every intermediate degree, incrementing CCPR1L and adjusting CCP1CON<5:4> step by step.
    2. Ensure that the CCP1CON<5:4> bits are properly handled when configuring the duty cycle.
    3. Both C and assembly language implementations are acceptable.

- **Bonus (20%)**
  - **Description:**

    Utilize RB0 as a motor control button to control motor rotation according to the following steps:

    1. Set the initial position of the motor to -90°.
    2. When the button is pressed, the motor should rotate smoothly from -90° to +90° and then return to -90°, repeating the motion continuously.

  - **Grading Criteria:**

    1. Do not limit the motor to only -90° and +90° positions. The program should process every intermediate degree, incrementing CCPR1L and adjusting CCP1CON<5:4> step by step.
    2. Ensure that the CCP1CON<5:4> bits are properly handled when configuring the duty cycle.
    3. Both C and assembly language implementations are acceptable.

  - **Hint:**

    The following steps should be taken when configuring the CCP module for PWM operation:

    1. Set the PWM period by writing to the PR2 register.
    2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4>bits.
    3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
    4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
    5. Configure the CCPx module for PWM operation.