

# Servo Demo

Ariel, Harsh, Mathew, Gabe

# Functionality

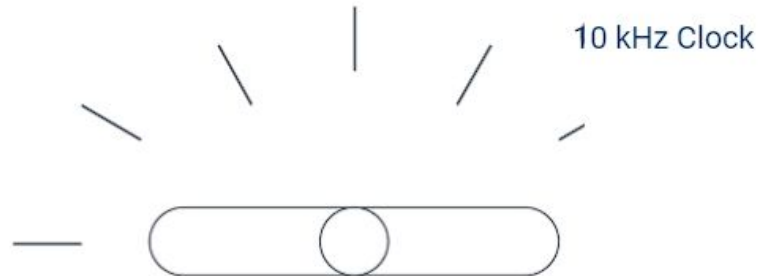
# Specifications and Functionality

- PWM generator for servo applications - Base Functionality
  - Pulse widths are bound between 0.5ms to 2.5ms of high time with 20 ms period
- Precision based approach
  - Faster 90 kHz clock compared to original 10 kHz clock
  - Scale factor to correct for manufacturer specifications
- Two Hardware Modes
  - Degree Control Mode:
    - Input degree
    - Precision to the degree
    - SCOMP outs 0-180
  - Variable Pulse Oscillation Mode:
    - Input speed
    - 256 speeds using a smooth, variable clock

# Precision Factor - 10 kHz vs 90 kHz Clock

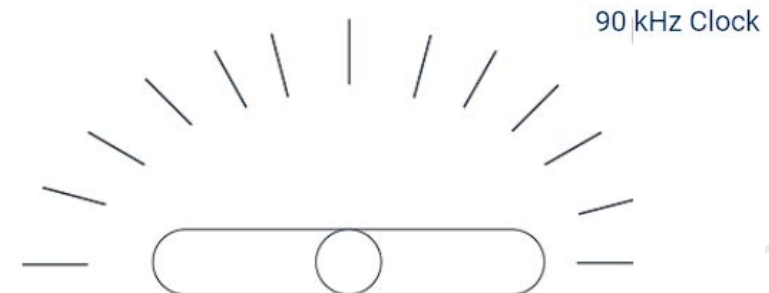
- 10 kHz Clock

- 1ms = 10 counts, so with a 2ms range, 20 counts + 1 since inclusive
- Would only be able to turn to angles with differences of 9 degrees



- 90 kHz Clock

- The range for the counter is 2ms:
  - 1 ms = 90 counts -> 2 ms = 180 counts
- 180 + 1 counts gives 181 possible values for pulses between 0.5ms and 2.5ms
- SCOMP OUTs a value for the peripheral to count to at 90 kHz, known as command



# Precision Factor - Scale

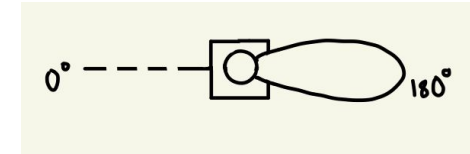
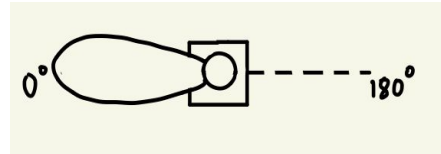
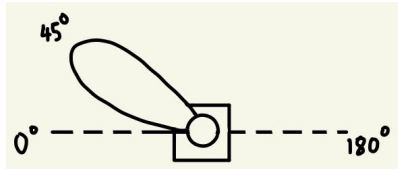
- SCOMP OUT: Command
  - Values sent to the peripheral are converted to a safe value, that stays within 0.5 - 2.5ms
  - Theoretically: The value inputted to the peripheral corresponds to the exact degree based on manufacturer specification range (0-180° with 180 counts)
  - Experimentally: Actual servo ranges from 0-200°
- Scale
  - Based on actual servo ranges, command inputted must be scaled by 8/9, to correspond to the exact degree since actual range of servo is ~ 200 degrees
    - Example: 160 OUT corresponds to 180 degree turn
  - This scale factor can be adjusted to account for the variability between different as well as similar servos

# SCOMP to Peripheral Communication

# Modes and Functions

## Degree Control Mode Functions

- DegreeTurn
  - Using a degree value stored in AC prior to CALL, this function turns to the exact degree requested
  - Automatically displays value in decimal on Hex Display
  - SetMax, SetZero, SetMid are subfunctions
  - Scale preset to 8/9

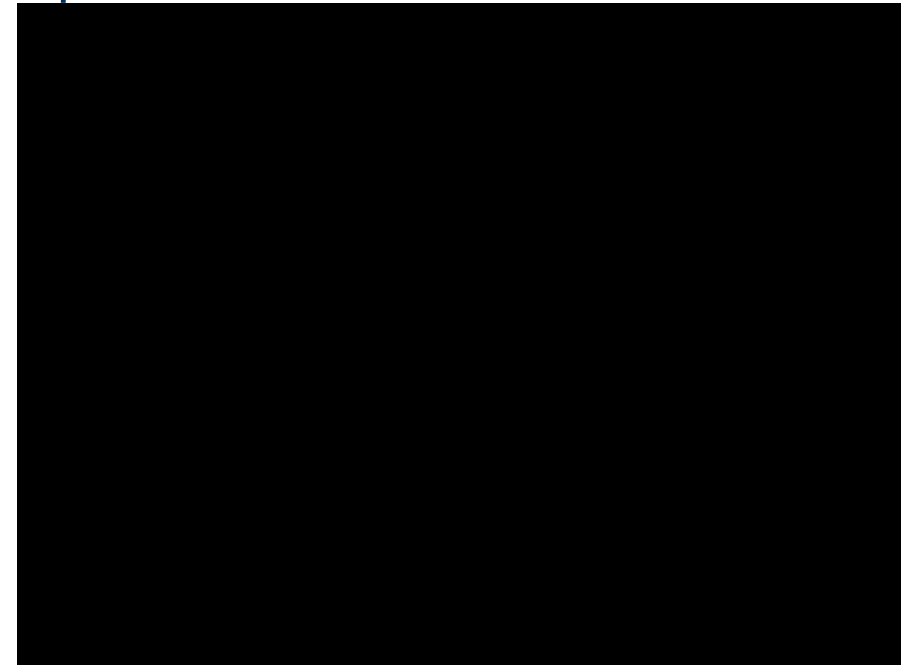
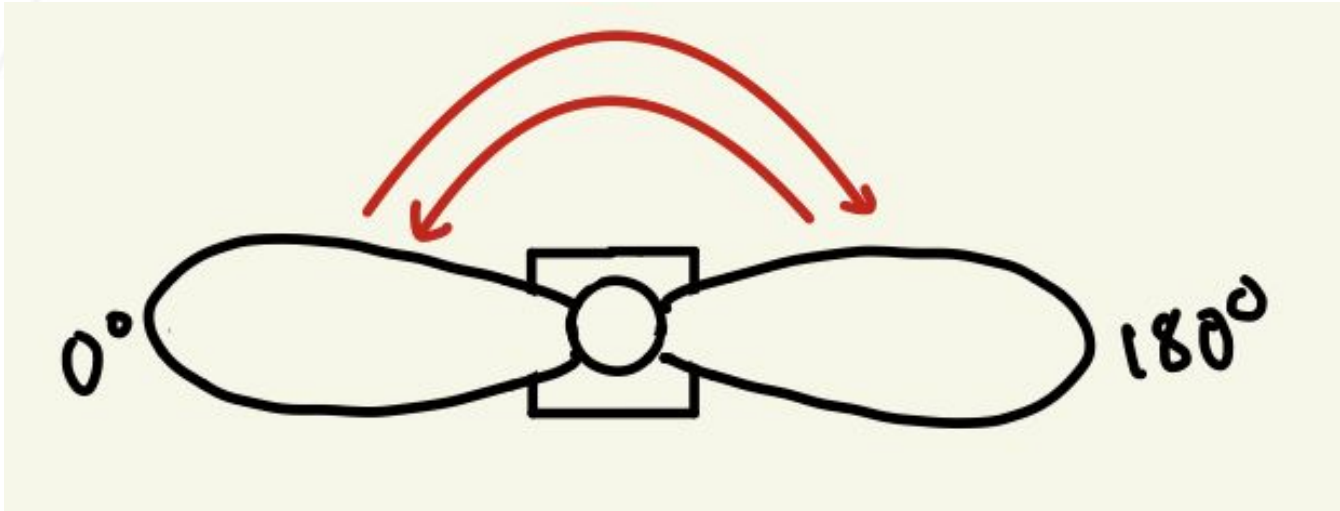


- Sprinkler
  - Number of cycles stored in AC
  - Autonomous function that to behave like a sprinkler by moving forward 16 degrees, pausing for 0.2 seconds, then moving back 8 degrees

# Modes and Functions

## Variable Pulse Oscillation Mode Functions

- Bounce Function
  - Speed stored in AC (deg/sec): Higher value -> Faster oscillation (1 - 256)
  - Autonomous function that controls the speed bounce functionality of the Servo
  - Oscillates between min and max position with variable speeds





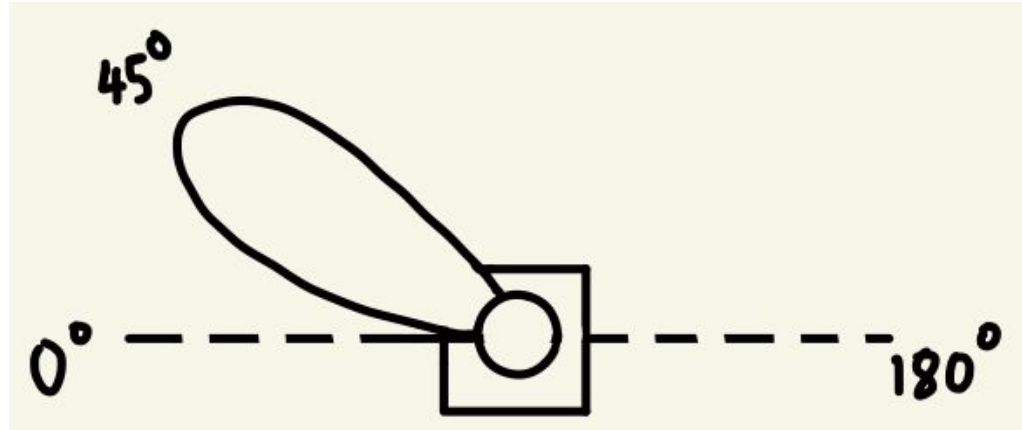
# Applications

# Switch Input

- Input degree value using switches
  - The 8 bit value from the switches is outed directly to the servo
  - LSB is the mode
    - 0 is degree mode
    - 1 is speed mode
  - Total value of switches results in oscillating speed or degree position
- Can be used to test and calibrate servo

# Precision Degree Turns

- Input value as a degree value in assembly as follows:  
LOADI [DEGREE]  
CALL DegreeTurn
- This application will demonstrate the precision of this function by turning to:
  - 10°
  - 20°
  - 30°
  - 40°
  - 60°
  - 90°



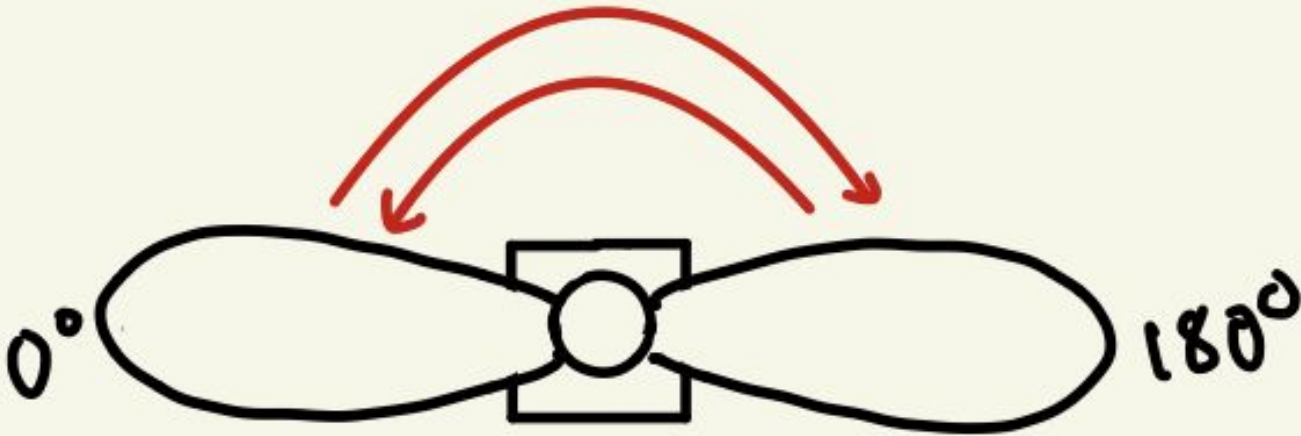
# Sprinkler Pulsator

- A function that takes in number of cycles and makes the servo move like a sprinkler
- The servo will move forward by 16 degrees then back by 8 degrees to imitate the pulsing motion of a sprinkler
- This application was implemented by calling DegreeTurn to turn clockwise and counter-clockwise, and timer for a slight delay



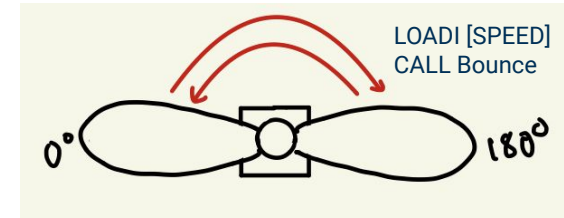
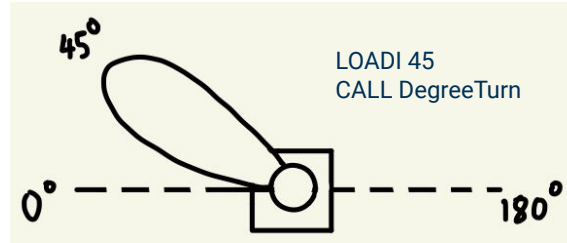
# Oscillating Tower Fan

- The servo will oscillate between  $0^\circ$  and  $180^\circ$  based on the output speed in assembly
  - This is implemented using the built in speed mode of the servo
- The application demonstrates the variable speeds, as well as the instantaneous change
  - 3 speeds are shown, and can change anytime
  - Example: `LOADI 32`  
`CALL Bounce`

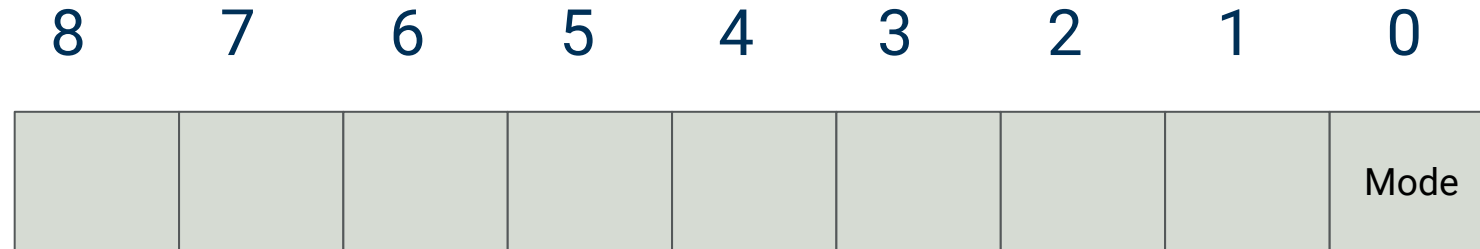


**Thank you!**





16-9 Unused



Input Values

1 - Speed  
0 - Degree