XMC_Servo Library Examples

RC Servo Examples for Infineon XMC Series (Arduino compatible API)

History

V1.0.0 June 2020 - Original Version

Infineon XMC series hardware PWM driven RC Servo output library using Infineon XMC-for-Arduino (https://github.com/Infineon/XMC-for-Arduino)

Examples General Information

All examples perform on 2 or 4 Servo outputs, and are shown using setting as degrees (0 to 180) or time (min to max µs). First pair of outputs is run anti-phase so one increases while other decreases on default range, second pair (Sweep examples) does the same as first pair but with a limited sweep range.

Error conditions

All examples use two on-board LEDs for various indications, most notably to indicated a pin number that is NOT a PWM pin number has been specified for the board compiled with. Each board has **different** pin numbers for PWM outputs as well as how many PWM outputs are available.

When BOTH LED1 and LED2 are ON permanently than a wrong pin number has been specified for the Servo to attach to.

Default configuration

Default pin configuration is for a XMC1100 Boot Kit Kit using

Function	Pin	Purpose	Examples used in
Servo A	3	1st Servo	All
Servo B	4	2nd Servo	All
Servo C	6	3rd Servo	Sweep only
Servo D	9	4th Servo	Sweep only
Knob/Pot	A1	Pot input	Knob only

Other Variants available PWM Pins

Some Board variants	PWM Pins
XMC1100 Boot Kit	3, 4, 6, 9
XMC1100 XMC2Go	8, 3, 2, 1
XMC1300 Boot Kit	31, 26, 32, 33
XMC4700 Relax	3, 10, 11, 95, 72, 96, 63, 36, 78, 90, 91,

5, 6, 9, 53, 39, 64, 38, 68, 79, 83, 82, 81

Tested on

Initial testing has been performed using

- XMC-for-Arduino V1.50
- XMC1100 Boot Kit
- XMC1300 Boot Kit

Example sketches

Example's Name	Purpose
Knob-degrees	Set two outputs using pot input by setting position in degrees
Knob-time	Set two outputs using pot input by setting position in time (µs)
Sweep-degrees	Continuously Sweep four outputs using programme by setting position in degrees
Sweep-time	Continuously Sweep four outputs using programme by setting position in time (µs)

These are located in standard Arduino Examples under

- Examples
    - XMC_Servo
        - Knob
            - Knob-degrees
            - Knob-time
        - Sweep
            - Sweep-degrees
            - Sweep-time

Knob-degrees Description

Set two outputs using pot input by setting position in degrees to update PWM output

Uses

- Servo A output running in same direction as pot input
- Servo B output running in **opposite** direction to Servo A
- LED1 on-board LED to indicate at centre position
- Pot (value > 1k) between Vcc and Gnd with wiper connected to pin A1
    - Check for 3V boards top of Pot is on 3V

Operation

1. Initialises hardware and LED1 OFF
2. Reads pot on analogue input
3. Scales pot from 0-1023 to 0-180 degrees
4. Writes to RC output (Servo A) as degrees
5. Writes to RC output (Servo B) as 180 - calculated degrees to give reverse direction
6. if calculated degrees is exactly 90° Lights LED1
    - Else turns LED1 OFF
7.

Wait 100 ms

8. Repeat from step 2

Knob-time Description

Set two outputs using pot input by setting position in time (µs) to update PWM output

Uses

- Servo A output running in same direction as pot input
- Servo B output running in **opposite** direction to Servo A
- LED1 on-board LED to indicate at centre position
- Pot (value > 1k) between Vcc and Gnd with wiper connected to pin A1
  - Check for 3V boards top of Pot is on 3V

Operation

1. Initialises hardware and LED1 OFF
2. Reads pot on analogue input
3. Scales pot from 0-1023 to 600-2400 µs pulse width
4. Writes to RC output (Servo A) pulse width
5. Writes to RC output (Servo B) as 2400 - pulse width to give reverse direction
6. if calculated pulse width is 1490-1510 µs Lights LED1
   - Else turns LED1 OFF
7. Wait 100 ms
8. Repeat from step 2

Sweep-degrees Description

Continuously Sweep up and down four outputs using programme by setting position in degrees to update PWM output

Uses

- Servo A output running in same direction as pot input
- Servo B output running in **opposite** direction to Servo A
- Servo C output running in same direction as pot input (using range 1000 to 2000)
- Servo D output running in **opposite** direction to Servo A (using range 1000 to 2000)
- LED1 on-board LED to indicate at centre position

Operation

Limited range on Servo C and Servo D will mean smaller step changes in timing as 180° translates over a smaller timing range.

1. Initialises hardware and LED1 OFF, initialise degrees to 0 and direction = up
2. Writes to RC output (Servo A) as degrees
3. Writes to RC output (Servo B) as 180 - calculated degrees to give reverse direction
4. Writes to RC output (Servo C) as degrees
5. Writes to RC output (Servo D) as 180 - calculated degrees to give reverse direction
6. If calculated degrees is exactly 90° Lights LED1
   - Else turns LED1 OFF

7. If direction = up
    - If degrees >= 180 set direction = down
        - Else Increment degrees
    - Else If degrees <= 0 set direction = up
        - Else Decrement degrees
8. Wait 100 ms
9. Repeat from step 2

Sweep-time Description

Continuously Sweep up and down four outputs using programme by setting position in time (600 - 2400 µs) to update PWM output

Uses

- Servo A output running in same direction as pot input
- Servo B output running in **opposite** direction to Servo A
- Servo C output running in same direction as pot input (using range 1000 to 2000)
- Servo D output running in **opposite** direction to Servo A (using range 1000 to 2000)
- LED1 on-board LED to indicate at centre position
- LED2 on-board LED to indicate Servo C and Servo D in operating range and values not being clipped

Operation

Limited range on Servo C and Servo D will mean any passed in value < 1000 or > 2000 are clipped to those values, so no change of movement will be seen for periods at end of sweeps while LED2 is OFF.

1. Initialises hardware and LED1 and LED2 OFF, initialise pulse width to 600, direction = up
2. Writes to RC output (Servo A) pulse width
3. Writes to RC output (Servo B) as 2400 - pulse width to give reverse direction
4. Writes to RC output (Servo C) pulse width
5. Writes to RC output (Servo D) as 2400 - pulse width to give reverse direction
6. if calculated pulse width is 1490-1510 µs Lights LED1
    - Else turns LED1 OFF
7. if calculated pulse width is 1000-2000 µs (Servo C and Servo D range)
    - Lights LED2 to show in active range for Servo C and Servo D
    - Else turns LED2 OFF
8. If direction = up
    - if Pulse width > 2400 set direction = down
        - Else Increment Pulse width
    - Else If Pulse width < 600 set direction = up
        - Decrement Pulse width
9. Wait 100 ms
10. Repeat from step 2

=== End of Document ===