

# XMC\_Servo Class and Methods Descriptions

## RC Servo Library for Infineon XMC Series (Arduino compatible API)

### History

**V1.0.0** June 2020 - Original Version

**Copyright © 2020 - Paul Carpenter - PC Services. All right reserved.**

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

### Common Limits and Definitions

Beware some values can be passed as *signed* values but only use unsigned ranges

### Degrees

Value between 0 and 180 (minimum and maximum) often 0° is fully left/down and 180° is fully right/up

### Servo ranges (Time width of pulse)

Normal specification is measured as time in  $\mu$ s with ranges

- 500 - Minimum value or most negative or 0°
- 1500 - Centred or 'zero' point or 90°
- 2500 - Maximum value or most positive or 180°

Standard Arduino Servo library uses ranges of

- 544 - Minimum value or most negative or 0°
- 1500 - Centred or 'zero' point or 90°
- 2400 - Maximum value or most positive or 180°

### Useful Constants

Constant Name	Description	Normal Value
MIN_PULSE_WIDTH	Hard Lower limit for Servo	544
MAX_PULSE_WIDTH	Hard Upper limit for Servo	2400
DEFAULT_PULSE_WIDTH	Centred position	1500
DEFAULT_ANGLE	Centred Angle	90
INVALID_SERVO	Main returned error code	255
REFRESH_FREQUENCY	Servo Update frequency (Hz)	50
REFRESH_TIME	Servo Update time (microseconds)	20000

When performing *attach( pin, min, max )* ranges can be set **WITHIN** these Pulse Width limits to smaller range to

match a servo interface device.

## Servo Channels (Outputs)

A channel is the class implementation that is then assigned a pin using valid Hardware PWM pins, for example

```
#include <Servo.h>           // Include library header

// Instantiate classes
Servo motor_left( );         // define a channel for motor Left servo drive
Servo motor_right( );        // define a channel for motor Right servo drive
.....
```

## Repetition Frequency

Each channel has output repeated every 20 ms (50 Hz) as per standards. This is fixed in library.

## Methods (Functions)

### Attach Servo to PWM Pin (attach)

Defines PWM pin to use and optionally minimum and maximum ranges within default range. Each Servo channel can have its own minimum and maximum range values. Note at attach the PWM is started with a centred servo (1500  $\mu$ s).

#### Parameters

Parameter	Type	Parameter description	Range allowed
pin	int	Arduino pin number to assign	Valid PWM Pin for current board
min	int	Optional minimum range in $\mu$ s	544 to 2399 or < max passed in
max	int	Optional maximum range in $\mu$ s	> min passed in to 2400

#### Calling examples

```
motor_left.attach( 9 );           // Attach pin 9 using default range min = 544, max = 2400
motor_right.attach( 11, 600 );    // Attach pin 11 using min = 600, and using default max = 2400
steering.attach( 12, 800, 2200 ); // Attach pin 12 using min=800 and max=2200
```

#### Returns/Errors

Returns a channel number as *int* of ranges

- 255 Error occurred (min  $\geq$  max, min or max outside default limits, invalid PWM pin, pin already in use)
- 0 to maximum number of PWM pins supported for valid PWM pin

### detach - Detach pin from Servo PWM

Stops PWM output, asserts pin level to Low (0), frees up resources. Can be reassigned by calling *attach()* again.

#### Parameters

None

### Calling examples

```
steering.detach();
```

### Returns/Errors

None

### attached - Is a PWM pin attached to Servo

Returns boolean for attached status

### Parameters

None

### Calling examples

```
value = steering.attached();
```

### Returns/Errors

boolean true is attached

### write - Write value as degrees or Time ( $\mu$ s)

Set output repeated pulse width as either a degrees or time width.

### Parameters

Two ranges of *int*

- < 500 - clipped to 0 to 180 as degrees
- > 500 - clipped to channel min to channel max (default 544 to 2400) as time in  $\mu$ s

### Calling examples

```
motor_left( 120 );           // Set output equivalent of 120 degrees
motor_right( 1800 );         // Set output pulse width to 1800 micro seconds
```

### Returns/Errors

None

### writeMicroseconds - Write value as Time ( $\mu$ s)

Set output repeated pulse width as time width.

#### Parameters

Value as *int* clipped to channel min to channel max (default 544 to 2400) as time in  $\mu$ s

#### Calling examples

```
motor_right( 1800 );           // Set output pulse width to 1800 micro seconds
```

#### Returns/Errors

None

#### read - Read last value written as degrees

Fetch last written value as degrees, if no write has occurred this is 90 degrees from attach.

#### Parameters

None

#### Calling examples

```
value = motor_left.read();
```

#### Returns/Errors

*int* in range 0 to 180 as degrees

**Note** 0 can also be Invalid Servo error

#### readMicroseconds - Read last value written as Time ( $\mu$ s)

Fetch last written value as time, if no write has occurred this is 1500 ( $\mu$ s) from attach.

#### Parameters

None

#### Calling examples

```
value = motor_left.readMicroseconds();
```

#### Returns/Errors

*int* in range min to max for that channel as time ( $\mu$ s)

**Note** 0 denotes invalid servo (not attached)

=== End of Document ===