XMC_Servo

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

History

V1.0.0 July 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)

- Uses PWM outputs on board by pin number
- **1 µs resolution** (steps)
- Uses separate min, max limits on each channel
- Quadrant safe (scales each channels' 0° = min. 90° = 1500 µs, 180° = max)
- analogWriteResolution of 16 bits for accuracy
- Hardware PWM ensures repetition frequency is accurate
- Low software overhead for easier integration
- Class methods same as Arduino standard Servo library
- Remove all possibilities of accessing data or updating for non-existent pin data.
- Additional functions extend capabilities.
- documentation included in Arduino/libraries/XMC_Servo/extras folder

Library Comparisons

Comparison To XMC_Servo_Adv Servo Library

The advanced version adds more useful methods and better accuracy down to 500 ns (0.5 µs)

Comparison To Arduino Standard Servo Library

Servo built-in library is an **Interrupt** driven module, where one or more 16 bit timer(s) are set to interrupt at a determined time interval to handle the first of the required output changes, then set the next time interval of up to 12 PWM outputs, however this makes each version **Hardware Specific** and depending on the hardware being run on depends on how close the timing is for each output to not miss changes or exceed the standard 20 ms repetition frequency. The software has large sections in machine code (assembler) specific to the processor being run on.

Using with unsupported hardware will give the **compilation** error

```
error "This library only supports boards with an AVR, SAM, SAMD, NRF52 or STM32F4 processor."
```

Known problems with Standard Arduino Servo Library

- Outputs can only be resolved **down to 4 µs steps**.
- Multiple servo channels can be assigned to the **SAME PIN**

- No checks for existing pin in use
- User defined limits in *attach()* of *min* and *max* can be set but are **never** used
    - If user defined *min* and *max* limits were used any large unbalanced limits would not scale from degrees correctly.
    - Not quadrant safe using degrees for *write()*
- *detach()* does not release resources or make channel available to reuse (no reattach)
- No invalid pin checks
- Changing to use PWM outputs would affect internal timings as same timers for PWM used for other functions
- Refresh on channels not guaranteed to be 50 Hz, each channel pulse length affects refresh interval and position jitter

Where as XMC_Servo library uses the **HARDWARE PWM** on XMC boards to more accurately (to within **1 µs**) control Servo outputs with minimal software overhead. The limit of number of Servo outputs depends on number of PWM outputs available on the Infineon XMC board being used, even the XMC2Go with only 16 pins has 4 PWM outputs.

Comparison To Infineon ServoC Library

ServoC is available from https://github.com/Infineon/ServoC however this can **ONLY** support **ONE** Servo output, whilst the servo output is ON (High) NOTHING else can be handled by the software as the pulse is determined by software time delays (delaymicrosecond), and the repetition of the servo pulse has to be hand tuned in your application.

These functions are blocking and require much hand tuning of your application.

XMC_Servo library using Hardware PWMs handles the many Servo outputs in hardware with no software overhead limits.

Installation

Standard Installation into Arduino IDE from Library Manager or from ZIP file

Usage notes

XMC_Servo library uses hardware PWMs at 16 bit resolution, by setting

```
analogWriteResolution( 16 );
```

Changing this setting in other parts of your application will have strange effects on the outputs, so do not use *analogWriteResolution* in your application.

Setting new values of Servo timing within same 20 ms repetition cycle will not necessarily take effect and the last value sent may not appear until NEXT 20 ms period.

Methods, Parameters and Return Values

Arduino Standard Servo Library Compatibility

Common terms

- degrees is a value between 0 and 180 (minimum and maximum)
- min is the minimum time value for a Servo channel output default is 544 µs
- max is the maximum time value for a Servo channel output default is 2400 µs
- centre (90°) is 1500 µs
- channel is this class implementation for a pin from *attach()*

Method for channel	**Actions for Methods**	**Parameters description**	**Parameter Type**	**Returns value/function**	**Return Type**
attach	Attach Servo to PWM Pin use default minimum = 544 maximum = 2400	Pin	int	Channel or 255 for error	uint8_t
attach	Attach Servo to PWM Pin pass in limits	Pin, min, max	int,int,int	Channel or 255 for error	uint8_t
detach	Detach pin from Servo PWM	none		none	
attached	Check Servo is attached to a pin	none		bool	true/false
write	Write value as degrees or Time (µs)	degrees < 500 (µs) > 500	int	none	
writeMicroseconds	Write value as Time (µs)	µs > 500	int	none	
read	Read last value written as degrees	none		Degrees	int
readMicroseconds	Read last value written as Time (µs)	none		µs	int

Note in many methods values are limited to

Type	**Minimum**	**Maximum**	**Note**
Degrees	0	180	
Time (µs)	544 or min	2400 or max	min and max are the channel defined limits

XMC_Servo Extensions

Method for channel	**Actions for Methods**	**Parameters description**	**Parameter Type**	**Returns value/function**	**Return Type**
attach	Attach Servo to PWM Pin define maximum use default maximum = 2400	Pin, min	int, int	Channel or 255 for error	uint8_t

Improved functionality

- *write()/writeMicroseconds()* have increased resolution of **1 µs** and quadrant safe.
- *detach()* Correctly sets output to low(0) and frees internal resources
- *attach()* Checks for valid PWM pin and not already used
- *attach()* Allows outputs to be assigned again after a *detach()*
- *attach()* Checks for min and max parameters within absolute ranges, and min < max

Examples

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, second pair (Sweep examples) does the same as first pair but with a limited sweep range.

Default pin configuration is for a XMC1100 Boot Kit or XMC1400 Boot 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

Example sketches are

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)