

Purpose:

This module provides a set of interface functions to generate up to 8 channels of pulse train suitable for driving RC servos. It is based on the output compare system on the Freescale MC9S12C32 microcontroller. The initialization function should be called before any use is made of Port T. After the initialization is complete, pins not assigned to the servo library can be used as ordinary I/O port pins. The library is provided in source form to be incorporated into your CodeWarrior project.

Header:

The header file **ServoLib.h** should be copied into the Headers folder of each project that want to use the library and that header file should `#included` in any module wishing to use the functions provided by this module.

Revision History:

November 5, 2011, First release of documentation for code written in 2010.

Initialization**Function:**

`Servo12_Init`

Parameters

`char [9]` A null terminated string of 8 ASCII characters to describe the mode of each of the pins on Port T. the legal values are:
'S' for servo output, 'x' for other function controlled by other modules.
The string positions, reading left to right, correspond to the pins MSB to LSB (`modeString[0]=MSB`, `modeString[7]=LSB`)

Returns

`Servo12ReturnTyp`
`SERV012_Err` if the input string is malformed
`SERV012_OK` if the mode string was OK

Description

Initializes the timer to a 3MHz count rate (assuming a 24MHz clock) and configures output compare system to take control of the selected pins and programs the toggle on overflow function to generate the pulse train with no software overhead.

Notes

Assumes a 24MHz bus clock, initializes to a 1.5mS pulse width

Usage:

```
if (Servo12_Init("SSxxxxxx") == SERV012_OK)
```

would initialize the library to output servo pulse trains on bits 6 & 7 as outputs and test to see that the initialization was successful.

Set Servo Pulse-width

Function:`Servo12_SetPulseWidth`**Parameters**

`unsigned char ChannelNum` 0-7 to define which bit to change
`unsigned int NewWidth` width of the desired pulse in micro-seconds

Returns

`Servo12ReturnTyp`
`SERV012_Err` if the channel number is bad, or channel not configured for servo output.
`SERV012_OK` if the mode string was OK

Description

programs a new value for the pulse width accounting for the fact that the value that we program is 0xFFFF - DesiredWidth and the that timer is in timer ticks not microseconds.

Notes

Assumes a 24MHz bus clock

Usage:

```
If (Servo12_SetPulseWidth(7, 2000) != SERV012_ERR)
```

Would set the pulse width on servo channel 7 to 2mS and check the return for errors.