# Servo Motor Interfacing with PIC18F4550

## Overview of Servo Motor

Servo Motor

A servo motor is an electric device used for precise control of angular rotation. It is used where precise control is required, like in the case of control of the robotic arm.

- It consists of a suitable motor with control circuitry for precise position control of the motor shaft.
- It is a closed-loop system.
- The rotation angle of the servo motor is controlled by applying a PWM signal to it.
- By varying the width of the PWM signal, we can change the rotation angle and direction of the motor.

For more information about Servo Motor and how to use it, refer to the topic Servo Motor (http://electronicwings.com/sensors-modules/servo-motor) in the sensors and modules section.

## Generating PWM using PIC18F4550

SG90 servo has a practical duty cycle time for -90° to +90 rotation that is different from ideal.

At ~0.6ms (3% duty cycle) we get shaft position at -90° of its rotation.

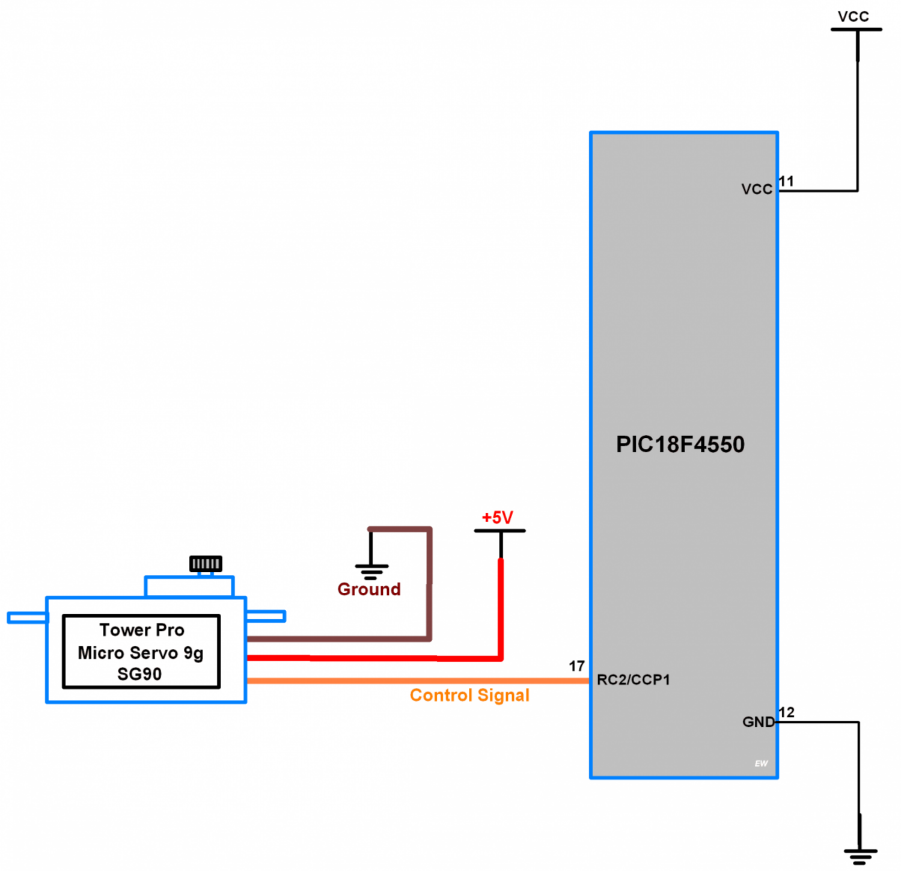
At ~1.4ms (7% duty cycle) we get shaft position at 0° (neutral) of its rotation.

At ~2.4ms (12% duty cycle) we get shaft position at +90° of its rotation.

To control servo motor in between -90° to +90° rotation. We need to generate a PWM waveform of 50Hz with duty cycle variation from ~0.6ms to ~2.4ms. We can use a fast PWM mode of PIC18F4550 using Timer1.

For more information about generating PWM in PIC18F4550 and how to use it, refer to PIC18F4550 PWM (http://electronicwings.com/pic/pic18f4550-pwm).

## Connection Diagram of Servo Motor to PIC18F4550



PIC18F4550 Interface with Servo Motor

## Servo Motor Example using PIC18F4550

Now let’s program PIC18F4550 to generate 50Hz PWM to control Servo Motor in an angle between -90° to +90° rotation.

## Servo Motor Code for PIC18F4550

```
/*
 * Servo control using PIC
 * http://www.electronicwings.com
 */

#include <pic18f4550.h>
#include <stdio.h>
#include <math.h>
#include "Configuration_header_file.h"

#define MINTHR          8000
#define RESOLUTION      488

#define InternalOsc_8MHz    8000000
#define InternalOsc_4MHz    4000000
#define InternalOsc_2MHz    2000000
#define InternalOsc_1MHz    1000000
#define InternalOsc_500KHz  500000
#define InternalOsc_250KHz  250000
#define InternalOsc_125KHz  125000
#define InternalOsc_31KHz   31000
```

## Change the Angle of Servo Motor with Potentiometer using PIC18F4550

Now let’s program PIC18F4550 to generate 50Hz PWM to control Servo Motor in an angle between -90° to +90° rotation using external potentiometer knob.

- Here we are using ADC channel 0 of PIC18F4550 to read external potentiometer knob and according to the ADC value, we are varying duty cycle of PWM.
- Refer ADC in PIC18F4550 (http://electronicwings.com/pic/pic18f4550-adc) for more information on ADC in PIC18F4550.

## Connection Diagram of Servo Motor and Pot to PIC18F4550

PIC18F4550 Interface with Servo Motor and POT

## Code for Servo control using POT with PIC18F4550

```
/*
 * Servo control using POT with PIC
 * http://www.electronicwings.com
 */

#include <pic18f4550.h>
#include <stdio.h>
#include <math.h>
#include "Configuration_header_file.h"
#include "ADC_Header_File.h"

#define MINTHR          8000
#define RESOLUTION      488

#define InternalOsc_8MHz    8000000
#define InternalOsc_4MHz    4000000
#define InternalOsc_2MHz    2000000
#define InternalOsc_1MHz    1000000
#define InternalOsc_500KHz  500000
#define InternalOsc_250KHz  250000
#define InternalOsc_125KHz  125000
```

## Video of Controlling Servo Motor using Potentiometer with PIC18F4550

## Components Used

Powered By (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

| | | |
|---|---|---|
| **Servo Motor MG995**<br>Servo Motor MG995 | X 1 | 🛒 (https://www.mouser.in/ProductDetail/Adafruit/1143?qs=GURawfaeGuDkq89VowcgJw%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) |

## Components Used

Powered By (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

| | | |
|---|---|---|
| **PIC18f4550**<br>PIC18f4550 | X 1 | (https://www.mouser.in/ProductDetail/Microchip-Technology/PIC18F4550-I-P?qs=oKK8NaWdAJs8nLDXBGwMXw%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)<br>Datasheet (/components/pic18f4550/1/datasheet) |
| **PICKit 4 MPLAB**<br>PICKit 4 MPLAB | X 1 | (https://www.mouser.in/ProductDetail/Microchip-Technology/PG164140?qs=r5DSvlrkXmLKDuYNJImLWw%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)<br>Datasheet (/components/pickit-4-mplab/1/datasheet) |

## Downloads

PIC Interface with Servo Motor Project Files

Download (/api/download/platform-attachment/349)

## Comments

Comment

nglethanhtrung (/users/nglethanhtrung/profile)
2018-04-23 10:07:17

Hi, i don't understand define MINTHR why is 8000 and next line of code why is 488. thanks

Reply Like

blarblublublar (/users/blarblublublar/profile)
2019-04-19 14:13:21

what is MINTHR?

Reply Like

blarblublublar (/users/blarblublublar/profile)
2019-04-19 14:14:22

and Why MINTHR and resolution are 8000 and 488?

Reply Like

lokeshc (/users/lokeshc/profile)
2019-04-19 15:30:12

Minimum threshold to decide which timerprescale bit to be used and resolution for decide internal frequency to gain possible accuracy with period

Reply Like 1 👍

blarblublublar (/users/blarblublublar/profile)
2019-04-20 14:25:03

Now i know overview but specifically why 8000 and 488 values are used?

Reply Like

manaskant1993 (/users/manaskant1993/profile)
2020-04-19 18:00:50

i could have a possible explanation for these values (of course i could be wrong....but still)...........typical operating frequency (Fosc) of pic is 8 MHz......now minimum and maximum prescale values of timer2 are 1 and 16 respectively........and the formula to find out value of PR2 register is PR2 = (Fosc/(4*Fpwm*TMR2_prescale_value))-1 as per datasheet.......since PR2 is an 8 bit register, the min and maximum values it can take for generating a valid pwm frequency is 1 and 255 respectively.......so to find out MINTHR value we substitute in above formula PR2 = 255 & TMR2_prescale_value = 1, and we get

Fpwm = 7812.5 Hz which can be rounded off to 8000 Hz approx., and thus MINTHR = 8000.............and then to find out RESOLUTION value we substitute in above formula PR2 = 255 & TMR2_prescale_value = 16, and we get Fpwm = 488.28125 Hz which can be rounded off to 488 approx., and thus RESOLUTION = 488..........I guess this is the reason for choosing 8000 and 488..........if somebody as a different explanation then we all would be glad to know.

Reply Like

pratiksawarkar09 (/users/pratiksawarkar09/profile)
2020-06-29 15:10:56

Duty_Scale = (((float)(ADC_Read(0)/4.0)*9.0)/255.0) + 3.0;
please explin me this calculation

Reply Like

STIPpro (/users/STIPpro/profile)
2023-07-13 09:08:55

mee too

Reply Like

pratiksawarkar09 (/users/pratiksawarkar09/profile)
2020-06-29 17:55:29

if my oscillator frequency is constant to 8MHZ then can i use timer2 postscallar to obtain 50HZ as output frequency

Reply Like

gopalbiswas (/users/gopalbiswas/profile)
2021-07-19 23:22:23

HI

Reply Like

leeyunjai1982 (/users/leeyunjai1982/profile)
2022-11-14 13:35:27

Can this chip (PIC18F4550) control many servo motors (12 items) simultaneously?

Reply Like

STIPpro (/users/STIPpro/profile)
2023-07-13 08:39:44

what happen if the input digital? how the conversion looks like?

Reply Like

**About Us (/about)**
**Business Offering (/business-services)**
**Host Platform (/launch-platform)**
**Contact Us (/contactus)**

**Connect On:**
Facebook(https://www.facebook.com/electronicwings)
LinkedIn(https://www.linkedin.com/company/electronicwin
Youtube(https://www.youtube.com/channel/UCNdqkukBtk4
Instagram(https://www.instagram.com/electronicwings_coi
igshid=1cip10jijttko)

Terms of Service (/terms-of-service)
Cookies Policy (/cookie-policy)
Privacy Policy (/privacy-policy)