



SERVO MOTOR CONTROL

Servo motor angle rotation control by adjusting PWM ratio.

Index

- What is a servo motor ?
- How do servo motors work ?
- Information about Hitec HS-311 Standard servo motor.
- Algorithm for two servo motors control.
- Schematic diagram and operation analysis of the circuit.

What is a servo motor ?

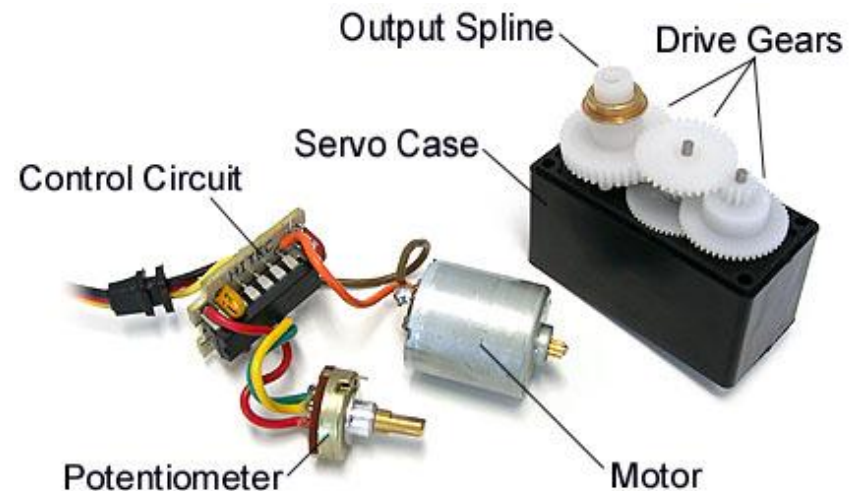
A Servo is a small device that incorporates a three wire DC motor, a gear train, a potentiometer, an integrated circuit, and an output shaft bearing (Shown in Figure). Of the three wires that stick out from the motor casing, one is for power, one is for ground, and one is a control input line. The shaft of the servo can be positioned to specific angular positions by sending a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, then the angular position of the shaft changes.

Servos come in different sizes but use similar control schemes and are extremely useful in robotics. The motors are small and are extremely powerful for their size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, doesn't consume much energy.

A very common use of servos is in Radio Controlled models like cars, airplanes, robots, and puppets. They are also used in powerful heavy-duty sail boats. Servos are rated for Speed and Torque. Normally there are two servos of the same kind, one geared towards speed (sacrificing torque), and the other towards torque (sacrificing speed).

Servos are constructed from three basic pieces; a motor, a potentiometer (variable resistor) that is connected to the output shaft, and a control board. The potentiometer allows the control circuitry to monitor the current angle of the servo motor. The motor, through a series of gears, turns the output shaft and the potentiometer simultaneously. The potentiometer is fed into the servo control circuit and when the control circuit detects that the position is correct, it stops the motor. If the control circuit detects that the angle is not correct, it will turn the motor the correct direction until the angle is correct. Normally a servo is used to control an angular motion of between 0 and 180 degrees. It is not mechanically capable (unless modified) of turning any farther due to the mechanical stop build on to the main output gear.

The amount of power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control.



How do servo motors work ?

Servos are controlled by sending them a pulse of variable width. The control wire is used to send this pulse. The parameters for this pulse are that it has a minimum pulse, a maximum pulse, and a repetition rate. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter clockwise direction. It is important to note that different servos will have different constraints on their rotation but they all have a neutral position, and that position is always around 1.5 milliseconds (ms).

The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse width Modulation. The servo expects to see a pulse every 20 ms. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position).

When these servos are commanded to move they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

Hitec HS-311 Standard

Detailed Specifications Control System: +Pulse Width Control

1500usec Neutral

Required Pulse: 3-5 Volt Peak to Peak Square Wave

Operating Voltage: 4.8-6.0 Volts

Operating Temperature Range: -20 to +60 Degree C

Operating Speed (4.8V): 0.19sec/60 degrees at no load

Operating Speed (6.0V): 0.15sec/60 degrees at no load

Stall Torque (4.8V): 42 oz/in (3.0 kg/cm)

Stall Torque (6.0V): 48.60 oz/in (4.5 kg/cm)

Current Drain (4.8V): 7.4mA/idle and 160mA no load operating

Current Drain (6.0V): 7.7mA/idle and 180mA no load operating

Dead Band Width: 5usec

Operating Angle: 40 Deg. one side pulse traveling 400usec

Direction: Clockwise/Pulse Traveling 1500 to 1900usec

Motor Type: Cored Metal Brush

Potentiometer Drive: 4 Slider/Direct Drive

Bearing Type: Top/Resin Bushing

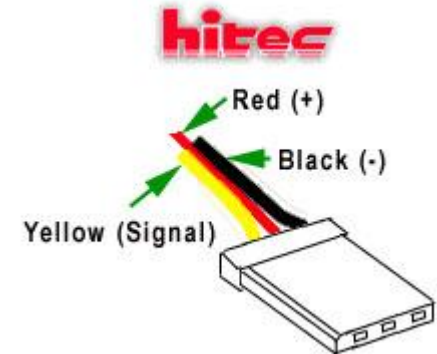
Gear Type: Nylon

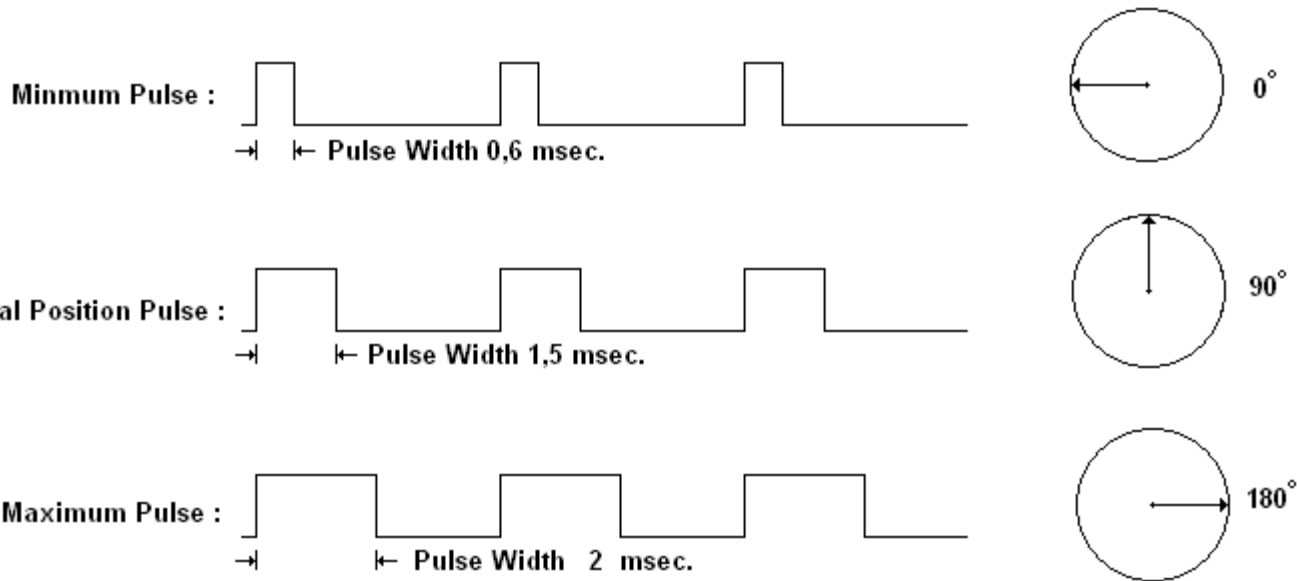
360 Modifiable: Yes

Connector Wire Length: 11.81" (300mm)

Dimensions: 1.57" x 0.79"x 1.44" (40 x 20 x 36.5mm)

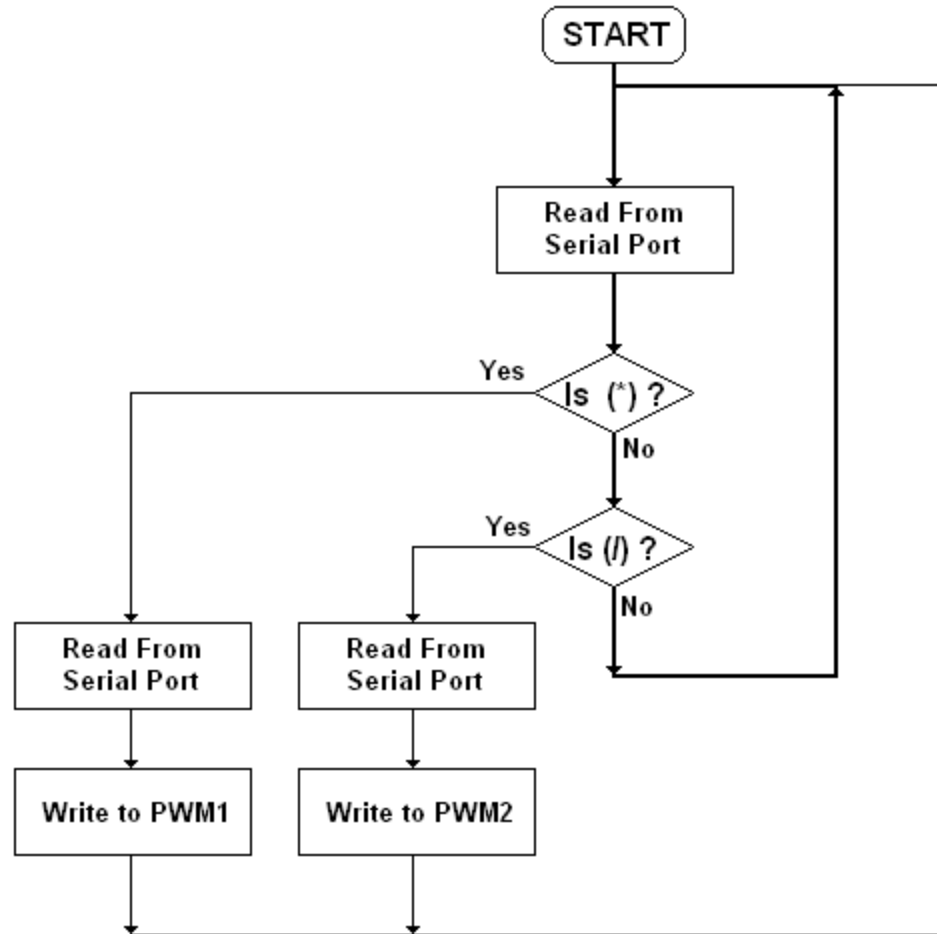
Weight: 1.52oz (43g)






When a pulse is sent to a HS-311 that is less than about 0.6 ms the servo does not rotate to any position and holds its output shaft at zero degree. If the pulse is wider than 0.6 ms the servo rotates. For example, if pulse width is equal to 1,5 ms HS-311 will rotate 90 degrees and for pulse width is equal or greater than 2 ms, HS-311 will make rotation of 180 degrees. The minimal width and the maximum width of pulse that will command the servo to turn to a valid position are functions of each servo. Different brands, and even different servos of the same brand, will have different maximum and minimums. Generally the minimum pulse will be about 1 ms wide and the maximum pulse will be 2 ms wide.

Algorithm





When the program is executed, data starts to read from serial port. The program algorithm consists of two loops, and after first data read from serial bus first loop of the program looks for characters star (*) and slash (/) for detection of data for which PWM module belongs to. In the second loop of the program, data write an appropriate PWM module

Data come from serial port is firstly checked for star character. If data is star character coded in binary, the following data will be written to PWM1 module of PIC 16F877, then program will turn to start for another read operation. If this time first character of data come from USART is not a star ASC II character, then program checks data for first character could be a slash ASC II character. If the character is slash then following binary data will be send to PWM2 module as a PWM data information, after that program will turn start again. If data come from serial port is neither a star character nor a slash character, then program turns to start for a new read operation, and follow path of first loop until one of these two criteria is satisfied.

There is no need draw anything about algorithm of Visual Basic part of the circuit. If we need to point something about it ; for simultaneous two servo motor operation, first set PWM value of motor 2 then set PWM for motor 1 and hold the button pressed for required operation time. As long as mouse button is pressed motors hold given positions in PWM signals.

PWM Mode of PIC16F877

In Pulse Width Modulation mode, the CCPx pin of PIC16F877 produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot TOSC \cdot (\text{TMR2 Prescale Value})$$

Operation of Circuit

The main purpose of this circuit is observing and understanding the operation principles of servo motors in electronic circuits. Thus, we only required a PWM generator with two undepended outputs. Therefore we used PIC 16F877 microcontroller just for its PWM modules and also for its other advantageous properties like as its price, easy programmability etc. Beside PIC 16F877, we use MAX 232 for PC communication. We only connect receiver pin of PIC 16F877 to MAX 232 since we don't require transmitting any data to PC. We made standart (prerequired) connections of both MAX 232 and PIC16F877. Since servo motors have three input pins two for supply and one for control signal, we do not use any buffer or driver for IC for drive them by PIC 16F877. Although, each servo motor draws about 0,5A from power supply, their control signal input require low power PWM signal.

As we mention about, we apply continuous PWM signal as long as mouse button is pressed. This cause overheat on motors. So, when mouse button is unpressed servo motor rotate to zero degree simultaneously. After the motors reach zero degree, PWM signal will be still available at inputs of motors for four seconds. Then PWM signal will be stopped by Visual Basic controlled program part. However servos work rightfully as long as PWM signal is applied, so, after this point servos couldn't save their high torque ability.

Sources

- www.servocity.com
- www.microchip.com
- www.hitec.com

**/ And special thanks to Abdül Balıçı and other dear friends for their helps.