

# Notes on Servo Adjustment For Ardustation 2

---

The Ardustation 2 software originally targeted a fixed tracking antenna platform: The ReadyMadeRC plywood antenna tracker. This system has two servos, both Hitec 485.



I originally modified the Ardustation software to accommodate this servo on the RMRC kit and calibrated the motion through the use of the Arduino servo library and the “attach” function. Here are the syntax and parameter definitions from the Arduino documentation:

## Syntax

```
servo.attach(pin)  
servo.attach(pin, min, max)
```

## Parameters

servo: a variable of type Servo  
pin: the number of the pin that the servo is attached to  
min (optional): the pulse width, in microseconds, corresponding to the minimum (0-degree) angle on the servo (defaults to 544)  
max (optional): the pulse width, in microseconds, corresponding to the maximum (180-degree) angle on the servo (defaults to 2400)

The second form of the call is where the range of motion of the servo can be adjusted. The software has one call for each servo: tilt and pan and only needs to be made at startup of the software. The “min” and “max” pulse widths (measured in microseconds) determine the “endpoints” for the movement of the servo and needs some explanation.

The Arduino function call to cause a servo to move uses the “write” command with an argument of an angular position in degrees. The valid range is 0 to 180 degrees.

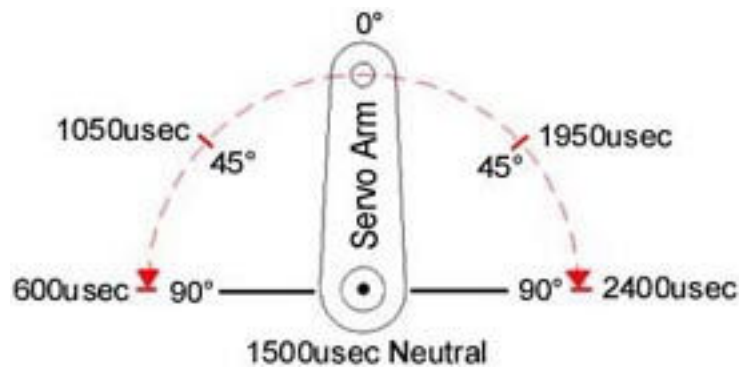
## Syntax

```
servo.write(angle)
```

## Parameters

servo: a variable of type Servo  
angle: the value to write to the servo, from 0 to 180

This normally would map to the actual angular position of the servo arm as pictured in this diagram for the Hitec 485:



The timing measurement in microseconds (such as 600usec) refers to the pulse width signaling method sent by the hardware to the servo to command a position. For the 485 servo, sending a command of “servo.attach(9,600,2400)” would correspond to a commanded position of 0 degrees sending the servo arm to the full counterclockwise position (marked as 600usecs in the diagram above) and 180 degrees sending the servo to the full clockwise position (marked as 2400 usec). In practical use, these calibration values are close, but need to be slightly tweaked for the actual servo in use. For my servos, I determine the best values where: servo.attach(9,609,2255) for the pan and servo.attach(10,591,2235) for the tilt. It is important to note that before the “attach” calls are made at power up, the **Arduino hardware commands both servos to the 90 degrees, 1500 usec position**. Therefore, before calibration, the servos will always rotate to this position. To avoid physical damage, any hardware attached to the servo arms must be able to operate safely at the power up 1500 microsecond commanded position since there is no application software override of this initialization behavior.

The actual mechanical configuration for the servos on the RMRC kit is different between pan and tilt. The pan servo is actually geared so that the full 360 degree rotation of the antenna can be reached. Therefore, the 2:1 gearing causes a commanded rotation of 180 degrees to the Arduino “write” call to correspond to a maximum rotation of 360 degrees clockwise of the antenna. The pan axis is mapped to a desired bearing commanded to the antenna (North – 0 degrees, East – 90 degrees, South – 180 degrees, West – 270 degrees) so the computed bearing must be divided by 2 to determine the value to pass to the Arduino write function such as in this pseudo code:

```
Pan.write(Bearing_to_target/2)
```

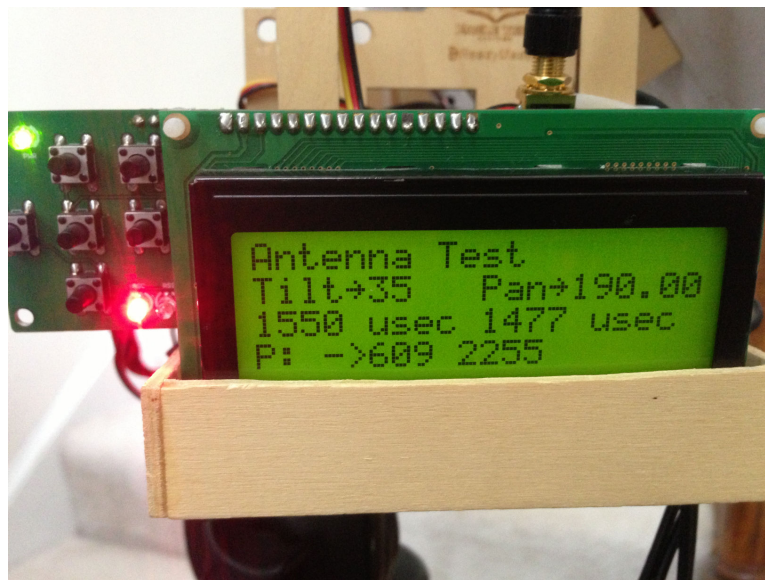
In actual practice, there is additional code to allow an offset in the pointing angle, but essentially all pan servo bearing angles must be divided by 2.

The tilt servo is used to point the antenna from the horizon, vertically to directly overhead in the sky (zenith) and only needs a range of rotation of 90 degrees. This servo is directly connected to the antenna without gearing, so the Arduino calls use the range of 50 to 140 degrees which works well with the physical mounting of the 485 servo on the RMRC kit. The calibration statement for the tilt “servo.attach(10,591,2235)” was determined by experiment to match the physical rotation of the antenna on my RMRC kit – keeping in mind the physical stops of the physical construction of the

antenna mount frame. I had to tweak the values to not overdrive the servo. I used the endpoints to adjust the physical stops instead of modifying the commanded angle positions (50 to 140).

## Using Ardustation 2.0.20 With Other Servos and Mounts

Numerous individuals have wanted to use Ardustation 2 with other mounts and servos, which proved to be difficult because there weren't easy provisions in the software or documentation to describe the existing configuration and design decisions. In addition, calibrating the servos required constant compilation of the software to adjust the two servo Arduino endpoint commands. I have made some modifications in the latest version (2.0.20) to make this easier. The changes are in the "Antenna Test" menu item which now display the physical antenna pointing values (bearing and pitch) and also displays the servo timing value for each test position of the antenna. A servo endpoint modification feature has



also been added to change the values to the Arduino "attach" command without requiring compilation of the software.

When adding other servos or mount configurations, it is important to note that Ardustation 2 always assumes that there is a fixed mapping between commanded antenna heading (0 to 360 degrees) and the pan servo position. Therefore, multiple turn servos/winch servos are not easy to add without substantial modification of the software. There are workarounds that can be used to support winch servos as long as the servo supports a normal servo positioning mode with potentiometer feedback in addition to the normal free running winch servo modes.

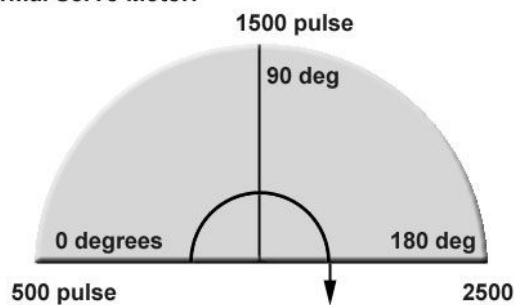
## Adapting Servos To Ardustation 2 Pan Function

In order to match a servo/mount to the Ardustation 2 software, it must be evaluated how much servo rotation is required to drive the pan and tilt axes. For pan, the software assumes a single rotation with

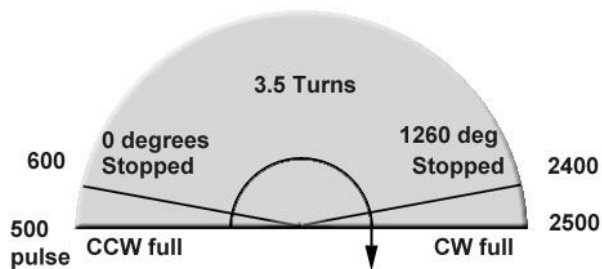
bearing values calculated from 0 to 360 degrees. The mechanical portion of an antenna mount must be evaluated to how much servo rotation is required. The HITEC 485 can move a total of 180 degrees, so gearing was added to the plywood mount to get the full 360 degree. If a servo is directly driving the pan axes, an optimum servo would provide a full 360 degrees of rotation with sufficient torque to overcome the lack of any gearing. There are very few servos that can do a full 360 degree rotation with potentiometer feedback on, so it may be necessary to use the partial rotation of a multiturn servo to drive the antenna.

As an example, consider the HS-785 sail winch servo. It differs from a normal servo as in the following diagram:

**Normal Servo Motor:**



**Hitec HS-785HB Winch Servo Motor:**



Rotates a full 3.5 turns  
Provides variable speed CW and CCW continuous rotation  
<600 provides increasing CCW rotation  
>2400 provides increasing CW rotation  
Between 600 and 2400: angular position

The servo can turn a total of 3.5 turns (1260 degrees) using pulse width timing values of 600 usecs to 2400 usecs. Pulse width values of less than 600 usecs or greater than 2400 usecs will cause the servo to

rotate freely. These values must be avoided since many antennas would not stand continuous rotation unless a slip ring is used. Therefore, this servo can only be used on the pan axis if the rotation is limited to 360 degrees. The Arduino servo calibration value ranges must surround 1500 usecs for the hardware power up issue, so the only way to make this work is to figure out the timing value range for 360 degrees. The total timing range for 3.5 turns is  $2400 - 600 \text{ usecs} = 1800 \text{ usecs}$ . The timing range for one turn is then  $1800 \text{ usecs} / 3.5 = 514 \text{ usecs}$ . So the final values would be: low =  $1500 - (514/2) = 1242 \text{ usecs}$ . High value:  $1500 + (514/2) = 1757$ . Therefore, to use this servo, the following variables would be set in the Ardustation 2 source code:

```
#define PANLOW 1242
```

```
#define PANHIGH 1757
```

When the software computes a antenna heading value, it computes the bearing as 0 to 360 degrees, and divides this by 2 to be compatible with the Arduino servo “write function” which expects angle values of 0 to 180 degrees. The 1242 is the pulse timing for 0 degrees and 1757 is the pulse timing for 180 degrees so the full rotation would be achieved.

The problems that may occur include a slightly different amount of rotation than the 360 degrees desired or non linearity of the rotation over the range. The first problem is easily solved by adjusting the two pulse width values using the new Antenna Test screen. The second issue would not be solvable, unless code would be added to Ardustation 2 to support nonlinearity of the servo commanded rotation values.

The suitability of the HITEC HS-785 would have to be evaluated based on how large an antenna is moved, and how much accuracy is needed for pan values. In general, this approach can be used to consider other servos, especially if gearing is used.

The approach should be to figure out how much servo rotation is required (usually less than 180 degrees is available with most servos). If less than 180 degrees of rotation is needed, figure out the total rotation needed in degrees. As an example consider that 75 degrees of servo rotation is needed to fully rotate the antenna 360 degrees of actual rotation. From the data sheet for the servo, figure out its maximum rotation range and what the low and high pulse timing values are.

Example: A servo data sheet gives the following timing values for a given max rotation of 1000 usecs low to 2000 usecs high gives 90 degrees of rotation. So  $(2000 - 1000) / 90 \text{ degrees} = 11.11 \text{ usecs per 1 deg}$  of rotation. To get the necessary pulse timing range multiply the rotation needed (75 degrees) by this computed value to get:  $75 * 11.11 \text{ usecs per deg} = 833 \text{ usecs}$ . Therefore the range of servo rotation needs to be 833 usecs surrounding the power on init servo position of 1500 usecs. PANLOW would be  $1500 - (833/2) = 1084$  and PANHIGH would be  $1500 + (833/2) = 1916$ .

These values would be entered into the Ardustation 2 software, compiled, and loaded and then evaluated with the Antenna Test screen.

## Adapting Servos to the Tilt Function

On the RMRC antenna, the Hitec 485 directly drives the antenna from straight to the horizon to the zenith for 90 degrees of rotation. The way the servo is mounted leads to the best range of servo pulse widths of 50 to 140 degrees. The endpoint settings are still set to allow full 180 degree rotation, but only the portion from 50 to 140 degrees is used. Other antenna mounts and servos which directly drive the antenna may have an alternate range that is best suited to obtain 90 degrees. To set an alternate range of values, use the following 2 defines:

```
#define tilt_pos_upper_limit 140 // Upper tilt limit (antenna points to the horizon)
```

```
#define tilt_pos_lower_limit 50 // Lower tilt limit (antenna points to the sky)
```

If no gearing is used, the delta between these two values must be 90 degrees. Otherwise, the delta must match the angle rotated going into the gear.

For both the tilt and pan servos, it is possible to reverse the rotation using these two variables:

```
#define PANREVERSE 0 // Set = 1 if the pan servo should be reversed for bearing output
```

```
#define TILTREVERSE 0 // Set = 1 if the tilt servo should be reversed for pitch up angle
```

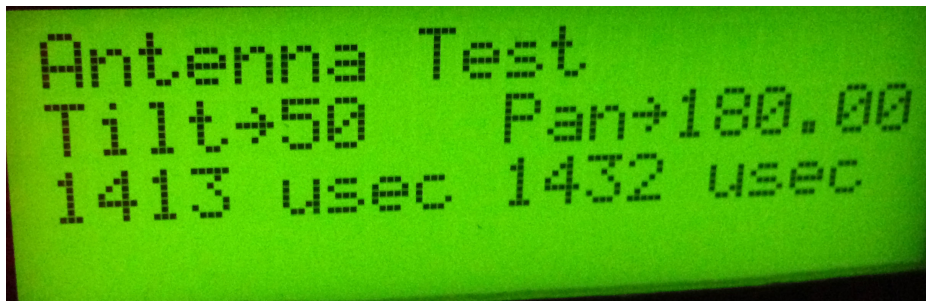
## Using the Antenna Test Screen to Tune Servo Settings

The Antenna Test Screen is reached from the top menu – selecting Antenna Test with the center button.

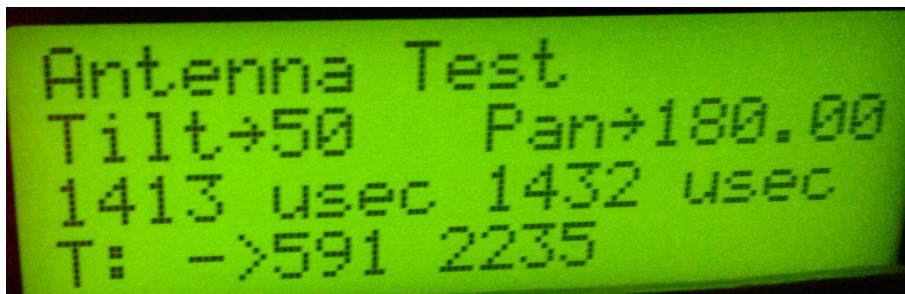


The top level screen displays the tilt value in degrees from 0 to 90 and the pan value in degrees from 0 to 360. Below the antenna angles is displayed the servo pulse with timing value for both servos. Pressing the “up” and “down” buttons changes the tilt position of the antenna. Pressing the “right” and “left”

buttons changes the pan position.



Pressing the center button will allow toggling through the servo endpoint values for both the pan and tilt servos. The marked servo endpoint can be modified by pressing the up and down buttons from 400 to 2400 microseconds. Not all servos can take timing values less than 1000 or more than 2400 microseconds and this can be detected if buzzing sounds come from the servo.



The servo endpoints use the default values which are set in the software as follows:

```
#define PANLOW 609
```

```
#define PANHIGH 2255
```

```
#define TILTLOW 591
```

```
#define TILTHIGH 2235
```

Changing these values will immediately change the servo endpoints and changes to the antenna position can be verified through the full range of rotation. When the optimal values are obtained, these values must be changed in the above lines within the source code for them to apply as defaults.

The operation of this screen can be viewed in the following YouTube video:

<http://youtu.be/woa-6Slzbjg>

Any bugs, issues, or questions can be submitted on the DIYDRONES Arudstation forum or by email at [heyno@heino.com](mailto:heyno@heino.com).

Heino R. Pull 8/3/2013

