Configure the servo:

* 1. Connect the servo to the DPC-CAN device as well as servo power. Connect the DPC-CAN device via USB to your Windows computer. Note the COM port associated with the DPC-CAN.
	
	2. With the correct COM port selected, press [Open] and verify the DPC-CAN-VER text field populates with the correct data. Assuming it does, select the “UAVCAN” radio button to detect the servo
	3. In the “SERVO Configuration” area in the top right section of the app, press [All] to select all the check boxes on that tab, and press the [READ] button to read those values from the servo in to the app
	4. Next, you need to enter some basic configurations for this servo. For this example, you want this servo to be “Servo 2” in ArduPilot, and don’t forget ArduPilot is on the bus as Node ID 10. We’ll also configure data streaming at 50Hz. The Servo is not a DNA Client on the bus, meaning it does not get its NodeID from the DNA Server (ArduPilot, in this example.) Because of this, you need to set the NodeID manually, and you need to calculate what that will be. Remember that NodeID of “1” is reserved, so if you were configuring for “Servo 1” you’d have to choose a different NodeID. Because of this, we recommend starting at a higher number. In order to configure the servo to be its own statically-assigned NodeID, you also have to add 256 to the NodeID value. In this example, since we are configuring Servo 2, we will set both the ActuatorID (ID2 in the configurator) and NodeID (ID1 in the configurator) accordingly, but we have to compute for the addition of 256 as well as our own custom node offset on the bus. For this example, we will add 276 to our desired Servo number for a value of 278 in NodeID / ID1. The breakdown of this is (256 + 20) + 2 (Actuator number). We get the 20 from just being an offset of 20 from base 1, you can use any number you prefer. A future firmware update will make this easier to understand and configure, but for now we need to perform this calculation manually. You’re not locked to the value of “+20”, but we want to avoid NodeID values of “1” and “10” after factoring for the addition of “256”.

	Here is a table of the values you’d configure for ID2 and ID1 when configuring for a given “Servo Number” in ArduPilot:

|  |  |  |
| --- | --- | --- |
| **ArduPilot Servo Number** | **ID2 (NodeID on UAVCAN Bus)** | **ID1 (ActuatorID)** |
| 1 | 277 | 1 |
| 2 | 278 | 2 |
| 3 | 279 | 3 |
| 4 | 280 | 4 |

Here is an example of the “SERVO Configuration” with the values set for ActuatorID 2 as described, including Streaming of servo telemetry data to ArduPilot at 50Hz. Do not forget to press each configuration item’s “Set” button, followed at the end by “Save” and then “Servo Reset.” Like in the example, the word “Reset” may not show up on your screen. This will also be addressed in a future version:


* 1. The next thing we’ll want to change is how far the servo travels for a given input command. By default, some of these Hitec UAVCAN servos will travel as far as -150 degrees all the way to +150 degrees for a total travel of 300 degrees. For most applications, you will want a 90-degree full-travel with 45 degrees on each side of zero.
* First we want to update the following two fields to -45.00 and 45.00 (-45 to +45 = 90-degrees of servo travel) Set your end-points in accordance with your application requirements 
* I’ll then press that section’s [Left] button to swing the servo to its “Min” value and note the number it displays, in this case 6144
* Next I’ll press the [Right] button and note the value it displays, in this case 10240

* Keep in mind, this has not changed any parameters in the servo yet. So far you have only commanded the servo to go to a specific point so that you can get the values associated with the encoder in those positions.
* In order to save the endpoints to the servo, you need to change its configuration. In the “Servo Configuration” section, select the “Mode” tab and enter those values for the “POSITION MAX\_LIMIT” and “POSITION MIN\_LIMIT” fields, press each of their corresponding [SET] buttons, press [SAVE] and finally reboot the servo by pressing the [SERVO RESET] (Reminder, it probably just says [SERVO] on your screen like above)
* When the servo reboots you can use the left buttons to command unitless values of -1.000 and +1.000 to validate that those values are giving me -45 and +45 degrees of servo travel. Sending this “unitless” type of command emulates with ArduPilot will be sending to the servo when it expects max travel. You can also then press the [Center] button to command the servo to center, and then place your control horn per your application’s requirements:


In the next document you can now proceed with AutoPilot configuration.