

Guide to Dynamixel AX-12A servos

The AX-12A Control Table

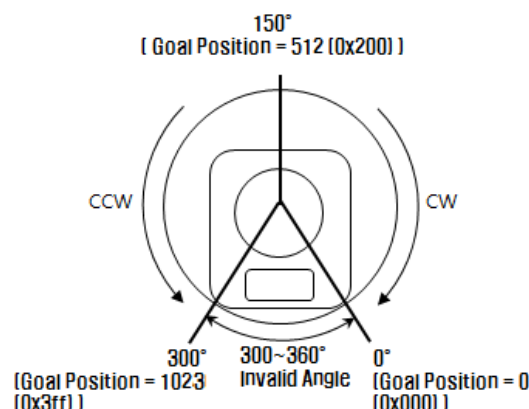
The AX-12A servos contain onboard memory, allowing the user to read the status of the motor or write configuration settings to it. This data is presented in the Control Table, which is divided in two fields: one for data stored in the non-volatile EEPROM and one for data stored in the volatile RAM. The control table, which is presented in the online e-Manual, also contains information about the default settings, as well as whether the address is read-write or read-only. Next some especially important control table addresses in the EEPROM are presented, which need to be considered when configuring the AX-12A servos, e.g., in the Real-Ant robot. Below is a subset of the table presented in the e-Manual:

Address	Size (Byte)	Data Name	Description	Access	Initial value
3	1	ID	DYNAMIXEL ID	RW	1
4	1	Baud Rate	Communication Speed	RW	1
6	2	CW Angle Limit	Clockwise Angle Limit	RW	0
8	2	CCW Angle Limit	Counter-Clockwise Angle Limit	RW	1023

The DYNAMIXEL ID is used to identify the device in the network on the Dynamixel bus, and must thus be a **unique value** for each device. Thus, the default ID of 1 must be changed if multiple AX-12 servos are used. **The DYNAMIXEL ID 200 is the default ID for the OpenCM9.04 controller and should thus not be assigned to any other device.**

The Baud Rate determines the communication speed on the Dynamixel network (servos and a control board), and the initial value of 1 corresponds to 1 000 000 bps. For the baud rate of other values see the e-Manual.

The clockwise and counter-clockwise angle limits are used to limit the rotation range. The initial values of 0 and 1023 should not limit the joint motion, but if the servos for some reason have restricted motion when they are taken in use, this parameter should be checked and changed if necessary. See picture from e-Manual below for angles corresponding to range 0 – 1023:



Accessing the Control Table via the OpenCM9.04 controller

The OpenCM9.04 is a microcontroller used in conjunction with the Dynamixel servos (other options are also possible via e.g. standard Arduino controllers and expansion boards). If a OpenCM9.04 “A Type” board is acquired, it will have no buses connected, so the “OpenCM9.04 - Connectors and Accessory Set” needs to be acquired and the Dynamixel bus connectors need to be soldered manually. The AX-12A servos also need **external power** (while the OpenCM9.04 itself can be powered via USB), so this needs to be arranged for the controller e.g., via the SMPS2Dynamixel adapter. See the OpenCM9.04 e-Manual for instructions for how to solder the connectors.

The OpenCM9.04 is programmed via the Arduino IDE, and functions for the Dynamixel servos are found in the Dynamixel2Arduino Library. The OpenCM9.04 e-Manual contains detailed instructions how to configure the Arduino IDE and install the library, and the library contains useful example code under File->Examples->Dynamixel2Arduino in the IDE, but below some functions of the library needed to configure the Dynamixels are highlighted.

The example codes start of with code that needs to be altered depending on the hardware used. Pay attention to the comments; if the OpenCM904 board is used without any expansion boards the examples will have to be edited so that serial is Serial1 (not Serial3 as by default) and DXL_DIR_PIN is 28 (not 22 as by default). The correct ID also needs to be selected, if it has been changed to some other ID from the default value of 1. Finally, the correct Dynamixel protocol version must be selected (for the Real-Ant project protocol version 1.0 was used).

```
//Board settings
#define DXL_SERIAL Serial1 //OpenCM9.04 EXP Board's DXL port Serial. (Serial1 for the DXL port on the OpenCM 9.04 board)
#define DEBUG_SERIAL Serial
const uint8_t DXL_DIR_PIN = 28; //OpenCM9.04 EXP Board's DIR PIN. (28 for the DXL port on the OpenCM 9.04 board)

const uint8_t DXL_ID = 6;
const float DXL_PROTOCOL_VERSION = 1.0;
```

When the communication is started by calling `dxl.begin()`, this function takes the baudrate as its input. In the examples the baudrate is 57600 bps, but if the Dynamixels are used with their default settings the baudrate will be 1 000 000 bps, so it needs to be changed when calling the function as pictured below:

```
dxl.begin(1000000);
```

The `dxl.setID()` can be used to set a new ID (New_ID in the picture below) for the Dynamixel servo, provided the current ID (DXL_ID in the picture below) is known. AX-12A servos should have ID = 1 when coming from the factory, but if previously used servos are used, look at the label in case the previous user has written down a new ID.

```
//Setting new ID
dxl.setID(DXL_ID, New_ID);
```

Using the `dxl.setOperatingMode()` function we set the mode to “OP_POSITION”, i.e. position mode, which also reset the CW and CCW Angle limits to 0 and 1023 respectively.

```
// Turn off torque when configuring items in EEPROM area
dxl.torqueOff(DXL_ID);
dxl.setOperatingMode(DXL_ID, OP_POSITION);
dxl.torqueOn(DXL_ID);
```

The current value of any Control Table address can be read and printed to the Arduino Serial monitor (remember to use correct baudrate in the monitor!) this way:

```
//Memory read test
//ID, CW & CCW limits
DEBUG_SERIAL.print("Current (new) ID:");
DEBUG_SERIAL.print(dxl.readControlTableItem(ID, New_ID));
DEBUG_SERIAL.print(" CW:");
DEBUG_SERIAL.print(dxl.readControlTableItem(CW_ANGLE_LIMIT, New_ID));
DEBUG_SERIAL.print(" CCW:");
DEBUG_SERIAL.print(dxl.readControlTableItem(CCW_ANGLE_LIMIT, New_ID));
DEBUG_SERIAL.print("\n");
```

Other address parameter names than ID, CW_ANGLE_LIMIT and CCW_ANGLE_LIMIT presented above can be found in the OpenCM9.04 e-Manual under 8.4.1.1 Dynamixel2Arduino Class > ReadControlTableItem(), or in the following link:

https://emanual.robotis.com/docs/en/popup/arduino_api/readControlTableItem/.

If writing directly to the control table is needed, e.g. to set some CW and CCW limits if full range of motion is not desired, a similar function called writeControlTableItem() can be used. This function can also be found in the OpenCM9.04 e-Manual under 8.4.1.1 Dynamixel2ArduinoClass > WriteControlTableItem(), or in the following link:

https://emanual.robotis.com/docs/en/popup/arduino_api/writeControlTableItem/

Sources:

AX-12A e-Manual: <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>

OpenCM9.04 e-Manual: <https://emanual.robotis.com/docs/en/parts/controller/opencm904/>

OpenCM9.04 Accessory Set: <https://www.robotis.us/opencm9-04-connectors-and-accessory-set/>

SMPS2Dynamixel adapter: <https://www.robotis.us/sm��2dynamixel/>