

Dynamixel AX-12A and Arduino: how to use the Serial Port

Share:

I bought two servos [Dynamixel](#), the AX-12A, which are the basic servos version of the Korean company Robotis inc.

Dynamixel servomotors are like Ferrari in the world of the cars : the best servomotors on the market.

Some characteristics of these servos:

- Operational Torque: 144 oz-in (10.4 kg.cm)
- 1 / 2 duplex multi-drop serial bus
- 1M bps serial communication
- Reports position, speed, load, voltage, and temperature
- Full rotation mode
- 300 ° angular position in 1024 increments
- Speed and torque control in 1024 increments
- Built in LED status indicator / alarm
- Shutdown on max / min voltage, load or temperature
- Single cable network connection

They are not particularly expensive: about \$ 45 U.S., in Europe about 45-50 euros.



These servomotors have difficulty connecting to the Arduino. In fact, the half-duplex communication to 1Mbps requires additional circuitry to make connections to Arduino if there are several servos to be connected. A

VISITOR MAP

[Large Visitor Map](#)

ARCHIVE

- [March 2016](#)
- [October 2014](#)
- [June 2013](#)
- [May 2013](#)
- [December 2012](#)
- [May 2012](#)
- [April 2012](#)
- [March 2012](#)
- [December 2011](#)
- [October 2011](#)
- [September 2011](#)
- [August 2011](#)
- [July 2011](#)
- [June 2011](#)

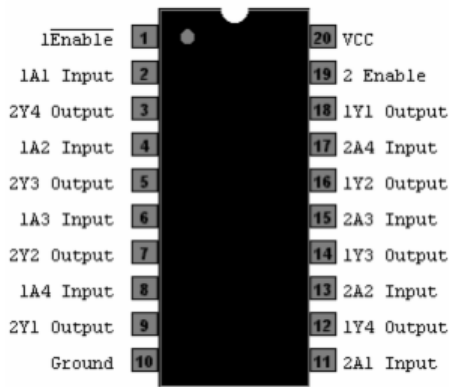
CATEGORIES

- [Aduino](#)
- [compound eye](#)
- [delta robot](#)
- [Dongbu Herkulex](#)
- [Dynamixel](#)
- [GRBL](#)
- [OpenCv](#)
- [robot brush](#)
- [robot drawing](#)
- [Robottini](#)
- [Senza categoria](#)
- [Speech](#)
- [Tips](#)
- [vacuum gripper](#)

ROBOT BLOG

- [billporter](#)
- [Patrick Mc Cabe Makes](#)
- [Savage electronics \(in spanish\)](#)
- [seriousrobotics](#)

single servo can be connected directly to the Arduino, in the case of several actuators it is necessary to use a tri-state buffer, which is placed between the Arduino and AX-12A. A simple tri-state buffer is the [74LS241N](#).



74LS241

The Dynamixel protocol is a serial protocol, so, Arduino side, the buffer 74LS241 must be connected to the serial port and then on pins 0 and 1. This is a problem: if the serial port is used byDynamixel, how to display information through the Arduino IDE Serial monitor?

We need another serial port. The Arduino library NewSoftSerial can help us, but we also need a USB-serial converter. I chose the converter [USB2Serial](#) built from Arduino team.

This is a picture of all the components needed to use the Dynamixel Servos.



The 2 'strange' components: the 74LS241 and the USB2Serial converter:

ROBOT SITES

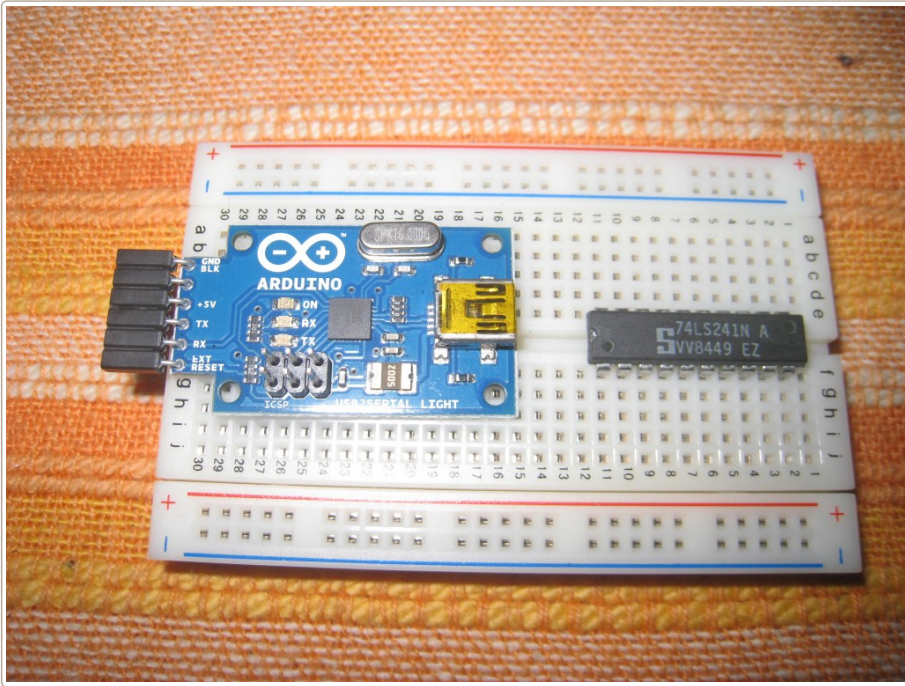
- [JavaCvPro site](#)
- [letsmakerobots](#)
- [Privacy policy](#)
- [Robotics guide](#)
- [Robotics Portal](#)
- [societyofrobots](#)

TAG

Arduino

audio chip auduino

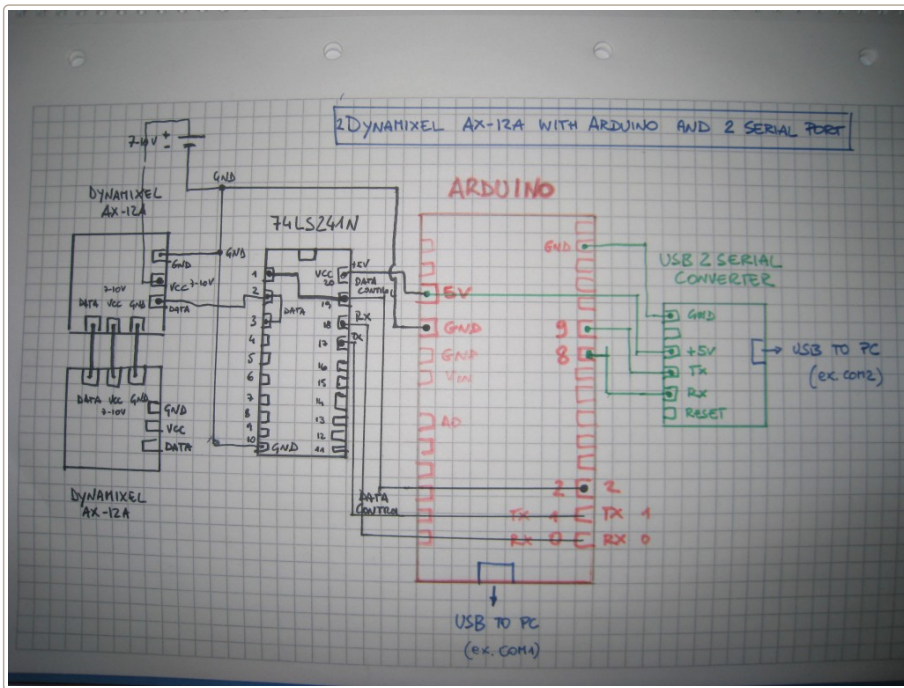
complementary filter compound eye Delta
Robot diffident robot dog robot Dongbu
graphic head robot Herkulex IR sensor javacv
JavaCvPro Kalman filter Line follower
me motor OpenCV plot Processing
quadrupod robot robbottini Self-balancing
Serial Port servo servos controller SOMO-
14D tail robot Ultrasonic sensor vacuum
gripper



These are all the components wired:



This is the wiring schema (sorry for the manual drawing):



Dynamixel servos require a software library to work with Arduino. An awesome library, realized by Savage Electronics, can be found [here](#). Instructions can be found [here](#).

This is a short video showing the Dynamixel servos in action. It is possible also to see the values printed in the Serial monitor with the NewSoftSerial throw the Serial port between the pins 8 and 9.

Dynamixel AX-12A with Arduino and 2 Serial ports



This is the code used:

```
#include <Dynamixel_Serial.h>
#include <NewSoftSerial.h>

NewSoftSerial mySerial(9,8); //9=tx, 8=rx
int k=0;

void setup(){ // Configuration

Ax12.begin(1000000,2); //2=data control
mySerial.begin(9600); //speed of serial port used by NewSoftSerial
}

void loop(){

// move servo 1 and 2 randomly
Ax12.move(1,random(200,600));
Ax12.move(2,random(200,600));
mySerial.print("Number:");
mySerial.println(k++);

delay(1000);

mySerial.print("Number2:");
mySerial.println(random(100));

}
```