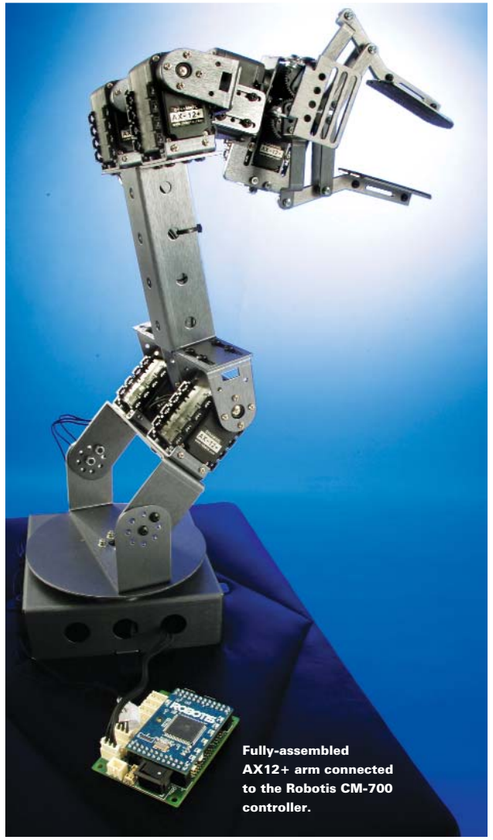
by Erick Royer

CRUSTCRAWLER AX12+ SMART ROBOTIC ARM

A versatile, robust platform with intelligent feedback and precise control

Editorial contributor Erick Royer will be writing about applications for this versatile robotic arm in a blog series that will be published at www.botmag.com with selected excerpts appearing in Robot magazine. Before diving into applications on the web, Erick wanted to revisit the details of the arm (initially reviewed by Steve Norris in our November/December 2008 issue in an article about his (Stonehenge) Clock issue) to both lure readers to his online reporting and place emphasis on the technical details that excite him most about this platform.
—the editors


Fully-assembled AX12+ arm connected to the Robotis CM-700 controller.

I am a History and Science Channel junkie. I love shows that highlight how things are made and the evolution of technology as it pertains to industry. Last weekend I was watching a documentary on Henry Ford and they showed some old photos and video clips of the first modern day production assembly production line. As labor intensive as it was, it demonstrated the latest innovation and technology of that time. A few days later I was watching another program called Ultimate Factories, which offered an inside look at the GM plant in Bowling Green, Kentucky, where they are producing the new Corvette Z06. This plant showcases the epitome of robotics technology and automation. What used to take a team of workers in the early 1900s days to complete can be done here in mere hours. One of the most fascinating parts of the factory for me was the use of robotic arms to perform precision welding. These massive and highly intricate robots perform welds in the exact same places with the exact same quality over and over—without a break or request for a raise.

I had been working on a review of the latest hardware/software combo of the AX12+ Smart Robotic Arm from Crustcrawler, trying to think of practical applications and ideas for this arm. After doing some research on the internet regarding larger scale industrial arms, I realized that they have many features and functions in common with the AX12+, from the ability to program finely tuned movements to receiving feedback such as pressure and temperature from the servos. I placed a phone call to Crustcrawler to confirm my theory that the AX12+ could be used as a mini simulator for programming and testing software to control the larger production robots. This fascinated me and I began to look at many remedial tasks that I do around my home which I could get the AX-12 to perform for me; from filling the dog's bowl with food to putting silverware away from the dishwasher tray. The AX12+ opened a whole new world of robotic projects for me, limited only by my imagination.

PHOTOS BY WALTER SIDAS

ASSEMBLY TIPS

Assembling the CrustCrawler AX12+ Smart Robotic Arm was simple fun requiring only three hours from box to programming. The included manual was very detailed and full of information and tips for each step. I first inventoried the parts to ensure that nothing was missing. CrustCrawler puts a huge emphasis on quality control so there was little likelihood that something was missing, but better safe than sorry.

When working on a project with a lot of small screws, washers and nuts, I like to use an egg carton from the grocery store. The 12 compartments are plenty for most projects to keep hardware organized and properly labeled. If you think of the manual as something you troubleshoot with rather than assemble with, I strongly suggest that you take time to study each step before you start turning screws. I found it also helpful to look ahead several steps prior to the one I was on to see how the component that I was working on would relate to future assembly. Another item to be cautious of is the number that is printed on each servo and servo box. This number represents the ID of each servo and it is important that each servo is assembled in the proper location.

The most delicate part of the assembly process is snapping the nuts into the correct location on the AX-12 servos. It is not hard, and the manual provides illustrated tips on how best to do this. Be sure to read this to avoid snapping off the plastic housings. Since the arm will be moving and stopping often as well as picking up and releasing objects, it is important to use a thread locker on all the screws where indicated.

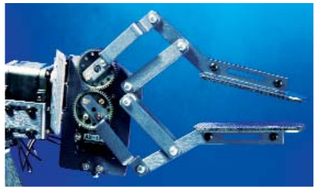
CrustCrawler did a tremendous job with the fit and finish of each component for this arm. Every part fit exactly as it was supposed to with no need to "muscle" anything into position. I particularly liked the use of Pem nuts on the brackets. A Pem nut is a small nut that is welded into the bracket eliminating the need for a separate nut. It really helps simplify the assembly process.

The Pem nuts are preinstalled, making assembly very easy.

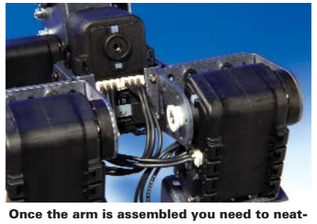
The manual has you install servos 2 and 3 as well as 4 and 5 so the servo horns face the outside of the arm, opposite each other. I originally installed this way, but once I started using the software, I realized that turning servos 2 and 4 around 180 degrees would simplify future programming, which I will explain below.

The trickiest part of the assembly process was probably the gripper, specifically the gears and gripper braces. Pay close attention to the instructions and any addendums that are in the manual regarding this process. The parts need to be assembled in a specific order to ensure that it works properly. It is also important not to over tighten the bolts on the gripper. If you put too much drag on the gripper's mechanics, it will stress out the servo and cause it to go into error mode.

The lower turntable brace has various holes in it to adjust the overall angle of the arm relative to the surface it is mounted to. I set mine per the manual for all my tests, but you have the option of mounting it in several positions. I can see adding two more servos (most likely the AX-18s since they have more torque to handle the entire arms weight) so the angle of the arm can be adjusted via the program or source code.



The AX-12 includes a sensor stand that gets mounted to the top of the gripper assembly. While I did not use it for this review, I will be working on some future projects where this will come in very handy to mount a camera, ultrasonic, light or motion sensors.



Once the arm is assembled you need to neaten up the wires using tie wraps. Since the bottom of the main channel is open, it makes the perfect place to conceal the wires. It is important to ensure that the wires do not impede the range of motion for each servo.

You definitely want to mount the base of your arm to a work surface or weighted board so it will remain secure when you are using it. The arm can swing pretty fast if you want it to, so make sure it is secure.

PROGRAMMING OPTIONS

Once the AX12+ Arm is assembled, you will need to choose a programming method and controller to bring this beauty to life. There are a lot of options, including the Robotis CM-5, USB2Dynamixel, Parallax Propeller board, any computer with a serial port, any controller that can communicate at 1mbps with a serial protocol, and just about any programming language. How's that for flexibility? CrustCrawler included the Robotis CM-700 control module and RoboPlus Software to be tested for this review. I used an external 9-volt power supply that is capable of providing 6 amps of continuous power to the arm.

The Robotis CM-700 Controller with sub-board attached.

CM-700 CONTROLLER

The CM-700 is a new modular controller which is perfect for scratch builders and more advanced hobbyists who wish to work with a mix of Dynamixel actuators. The microcontroller, with dual bus handles both TTL (for the AX-12 and AX-18F servos) and RS-485 (for the RX-64 servos) communication. It is designed to work with a daisy chain of similar servos or a mixture of AX-Series, RX-Series, or the EX-106+. The only limitation is that the input voltage must overlap the range of each servo used. It has an acceptable input voltage range from 7 to 27 volts.

In order to make use of the module you need a CM-700 SUB Board. The sub board manages the power with a battery/power jack and power switch. It features ports for 3- and 4-pin Dynamixel serial communication. There is a communication jack that is used in conjunction with a LN-101 USB interface to connect the CM-700 to a PC. This port can also be used with a wireless communication module such as the ZIG-110 and IR receiver. Lastly, there is a 5-pin peripheral device connection port that can be used to connect devices such as DMS, touch sensors, IR sensors and the like. Since the CM-700 will allow you to store programs on the board, there is a start and mode button used to activate motion programs.

ROBOPLUS SOFTWARE SUITE

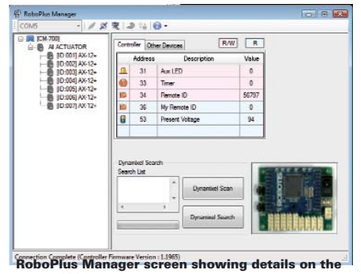
The CM-700 comes with the RoboPlus software suite which contains four separate programs; RoboPlus Task, RoboPlus Manager, RoboPlus Motion and RoboPlus Terminal. Additionally there is a Dynamixel Wizard program that will allow you to manage the Dynamixel's firmware as well as check the status and setup the necessary modes.

This is the main RoboPlus screen with links to each program.

RoboPlus Task. RoboPlus task refers to a "task" as a set of motions that perform specific actions. The source code that specifies the tasks to be executed are called "task code." The robot will move and perform actions according to the task codes that you write. The RoboPlus Task application facilitates and simplifies writing these codes. While I did not write any code with the Task application, I did use it to look at the motion files created by RoboPlus Motion.

RoboPlus Terminal. This application is a tool designed to help users work with the controller via a text-based user interface. The terminal program communicates with the controller in ASCII code and prints various information transmitted by the controller for the users. This application was included for advanced users who can manage robot controllers directly with ASCII code.

RoboPlus Manager. This program is used to handle devices that are used by a robot. It allows you to manage the controller's firmware with updating and restoring functions. You can also inspect the status of and test the controller and any connected peripheral devices. One very handy use for this application is to check the settings for each AX-12 servo, such as servo ID. When you connect the AX12+ Arm to your PC and run the Dynamixel Scan from within the RoboPlus Manager program, it will return a list of all the servos connected and you can click on each one to obtain more information.

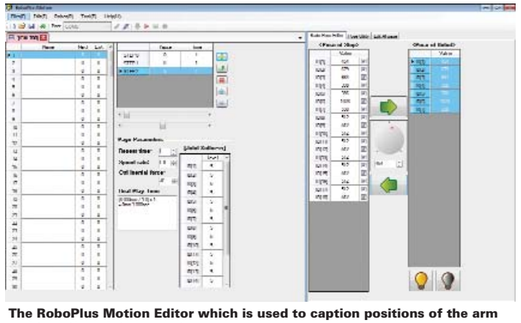


RoboPlus Manager screen showing details on the seven AX-12 servos.

RoboPlus Motion. This application is the bread and butter of this software suite, especially as it pertains to the AX12+ Arm. If you remember Claymation cartoons, then you are familiar with stop-motion technology where basically you move an object and take a photo, then move it again and so on. When you are finished you can play all the photos together and they will create motion. The RoboPlus Motion application works in a similar fashion. The Dynamixel servos can report positioning information back to the controller giving you the ability to move the arm to a position you want and "capture" the settings into the software. This is referred to as the "pose" of the robot within the application.

There are three basic parameters that you will be editing within the Motion application; Pose, Step, and Page. A "motion page" is defined a series of steps that make up a specific action.

For example, you could define a page called "return to home" which would run the steps necessary to return the arm to what you define as the home position. If you are using the AX12+ to pick up and move an object from one defined position to another you can create a page called "pickup" and another called "place." The great thing about this is once you define the pages you can rearrange them in



The RoboPlus Motion Editor which is used to caption positions of the arm and turn them into pose steps.

ROBOTIS DYNAMIXEL AX-12 SERVO

Each servo has an input and output port for the wiring. It is important to ensure that they are connected in sequential order and that you have the wires in the correct locations.

Hobby servos that we see in radio control models are familiar ground. The Dynamixel AX-12 servo will do anything a hobby servo can, but it has clear advantages designed specifically for robotics. The AX-12 has the ability to track its speed, temperature, shaft position, voltage, and load. The servo's sensor management and position control is handled by the built-in microcontroller rather than your main controller.

The AX-12 is more of a robotic actuator than a servo. Its design incorporates a gear reduction system, a precision DC motor, and a microcontroller with networking functionality. Dynamixel servos use serial communication in which you daisy chain wires from one servo to the next, thus requiring fewer wires to the microprocessor.

The servos operate over TTL RS-232 serial protocol and support communication speeds of up to 1MBPS, which makes them compatible with a variety of controllers on the market including the USB2Dynamixel adaptor, CM-5 programming bundle, or the new CM-700 controller. Each servo is assigned a unique ID allowing the controller to communicate directly with that servo. The AX-12's position, velocity, compliance, and torque can be set with a single command packet allowing the main processor to control many servos with very few resources.

The AX-12 is capable of precise control response with position and speed resolution programmed in 1024 steps. Feedback can be sent to the controller for angular position, angular velocity and load torque. The servo also features a status display LED which can be programmed to indicate errors, including high temperature, overload, over voltage, and low voltage conditions. These servos are very modular in nature and there are numerous brackets on the market designed specifically for them, thus giving maximum flexibility to create custom robot projects.

any order you want. Each page you define can be comprised of up to seven steps. Each step is a single motion of the AX12+ arm. You can define a pause and the duration for each step allowing you to control the overall speed of that particular motion. Steps, like pages, can be rearranged, edited or removed as needed.

CREATING A POSE

On the right side of the screen there is a column of settings for "pose of step" and for "pose of robot". The value column of the "pose of robot" column will contain either "ON" or "OFF" or the current position of the servo. The on and off setting refers to the torque of the servo. This setting is adjusted by pressing the yellow or gray light bulb button. When the servo is set to off, it allows you to move it freely by hand. When set to on, the servo is locked in that position and should not be moved by hand. You can select a single servo to adjust or multiple by pressing the CTRL key while you click on each servo. For setting the initial step, I usually select all the servos and turn the torque OFF allowing me to move the entire arm freely. Then once I get it in the position I want, I press the yellow light bulb and the arm will remain in that position while sending the position for each servo. Then, by pressing the left arrow button, the servo positions are sent to the "pose of step" column. Those settings now become the pose of the arm for that step.

You can then add a new step and repeat the process. On subsequent poses, I tend not to select all the servos since doing so will cause the arm to lower under its own weight. I will usually only turn off the servos that I need to move. A nice feature of this program is the wheel icon between the two pose boxes. This allows you position a servo or group of servos by rotating the dial in small increments. It is great when you need to control precise positions of the arm or gripper.

Earlier I mentioned that I rotated servos 2 and 4 on the arm. The reason I did this was to be able to select servos 2 and 3 as well as 4 and 5 and move them together with the wheel. Using the setup in the manual, these servos actually move opposite each other, making it impossible to select the pair that controls that part of the arm and use the wheel to dial in small adjustments because one number will have to increase while the other has to decrease. After repositioning the servos I can now make the adjustments together.

You can test the steps and pages by pressing the play button. The speed and repeat time for the page can be set as well. Once you are happy with the motions you created, you can save the file as an .mtn file. This file can be recalled back into the RoboPlus Motion application or you can open it in the RoboPlus Task application for more detailed code editing. You can also download the motion file directly to the CM-700 controller to run the motion without being connected to a PC for fully independent operation.

CONCLUSION

I really enjoyed learning how to control the AX12+ Smart Arm with the CM-700 controller as well as the use of the RoboPlus software. When I first assembled the arm and started looking at the RoboPlus Motion application I thought it might be fairly complicated to get going. But once I took a little time to understand the program and how it interacts with the arm, it was actually very simple to make the arm do just about anything. The addition of sensors, a mobile platform and a camera system will greatly enhance the capabilities of the AX12+ by adding logic and motion so that it can react to its environment. No matter what type of project you can think of, this combination of hardware and software gives you industrial-grade quality and performance in an affordable package.

Links
CrustCrawler, www.crustcrawler.com, (480) 577-5557

Robotis, www.robotis.com

For more information, please see our source guide on page 89.

INTERVIEW WITH ALEX DIRKS, OWNER OF CRUSTCRAWLER

ROBOT: What led you to design the first Crustcrawler robotic arms, and specifically the AX-12?

ALEX: Robotic arms are an essential part of a lot of robotic applications. What good is having a moving robotic platform if it can't manipulate objects in its environment? The AX12+ and AX-18 Smart Robotic arms where a natural progression in technology away from the standard RC-based robotic arm offerings.

ROBOT: Why did you choose the AX-12 servos for the basis of this arm?

ALEX: If you examine any industrial based robotic arm, one of the essential factors for precision control of the arm in 3D space is position feedback. Without position feedback, there is no way the hosting computer can gauge where the arm is in 3D space at any one time. The AX12+ / AX-18 servos are the only servos on the market that can not only provide position feedback but also temperature, voltage and current feedback. Also, at 220 oz.-in. of torque for $44, they are a lot more affordable and provide a lot more features than standard RC servos with the equivalent torque rating. We selected the AX12+ and AX-18 servos for the core of our Smart Robotic Arm design so we could provide our customers with a truly intelligent, affordable, hi-torque robotic arm for use in a wide variety of applications.

ROBOT: Can you describe some applications for which your customers are using the AX12+ Smart Arm?

ALEX: Our AX12+ and AX-18F Smart Robotic Arms are used in the pharmacy industry for "pick and place" functions, University education and research, light industrial functions for moving parts and sorting bins, and for ROV applications. We also sell a lot of arms to students for their engineering projects.

ROBOT: What might we expect for future robotic arms or accessories from Crustcrawler?

ALEX: We are currently in beta testing of our next generation of "light industrial" robotic arms that will include all of the feedback capabilities of the AX-12 and AX-18 series of servos but will feature RS-232 and RS485 communication and larger servos with even more torque capability. As with the AX12+ and AX-18 Smart Robotic arms, we will also be providing power supplies, additional grippers and camera units.