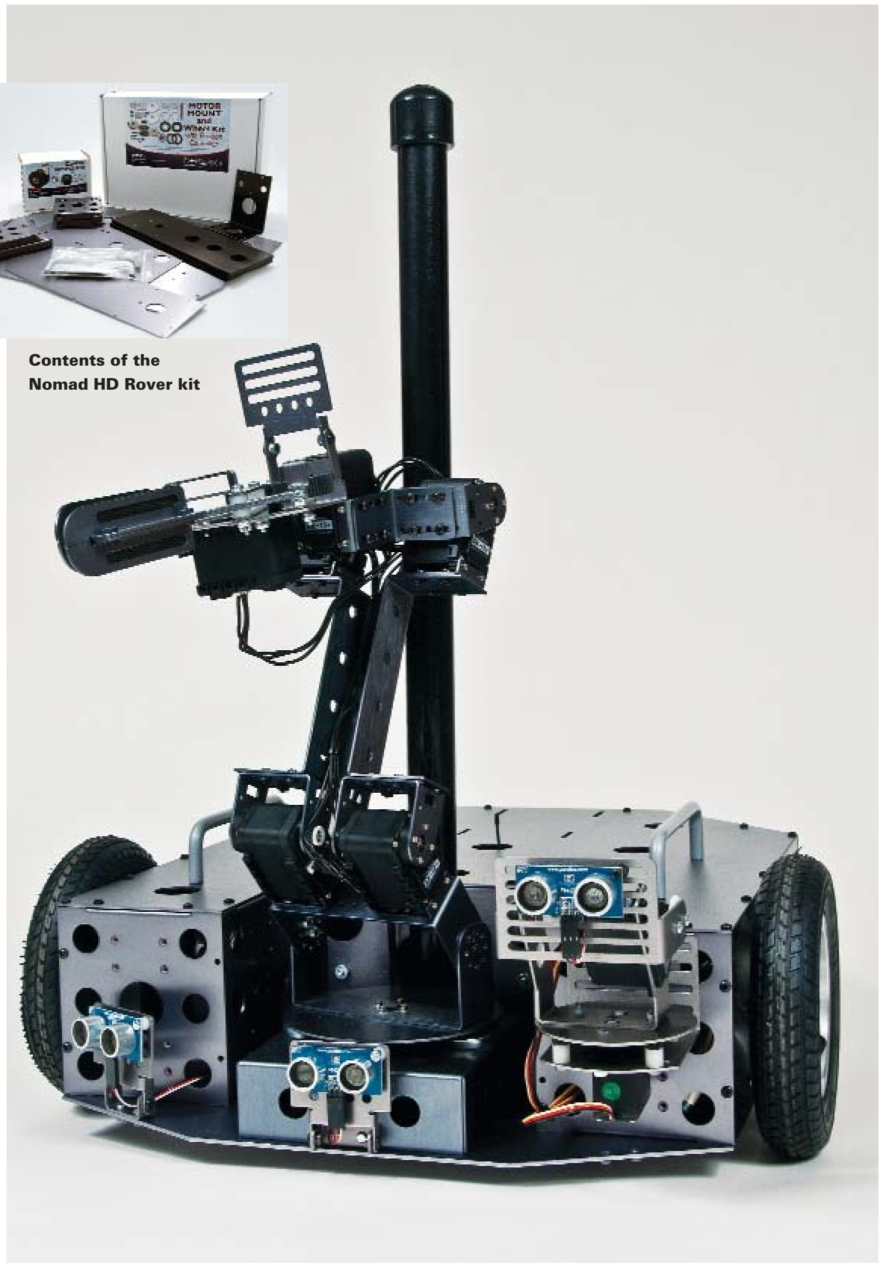
by Steve Norris

THE CRUSTCRAWLER Nomad HD Rover

A robust, rugged, versatile robot platform

I've always been a big fan of CrustCrawler's robot kits so when I heard they introduced a new two-wheeled rover using the Parallax Motor Mount Kit I just had to build one. CrustCrawler is well known for its high quality all metal robot kits and its latest, the Nomad HD Rover, follows that tradition. The HD stands for Heavy Duty and with 252 square inches of upper deck space and the ability to carry 10 pounds the Nomad lives up to its name. Despite being Heavy Duty the Nomad moves around as smooth and quiet as a cat. Over the next few articles we will build and expand the Nomad through a series of applications starting from simple encoder navigation to wall following and then compass navigation. As a final project we will add the CrustCrawler AX-12 Robotic Arm to the Nomad and build simple beverage can retrieval and recycle application. This will complement my recently completed Beverage Delivery System.

Contents of the Nomad HD Rover kit



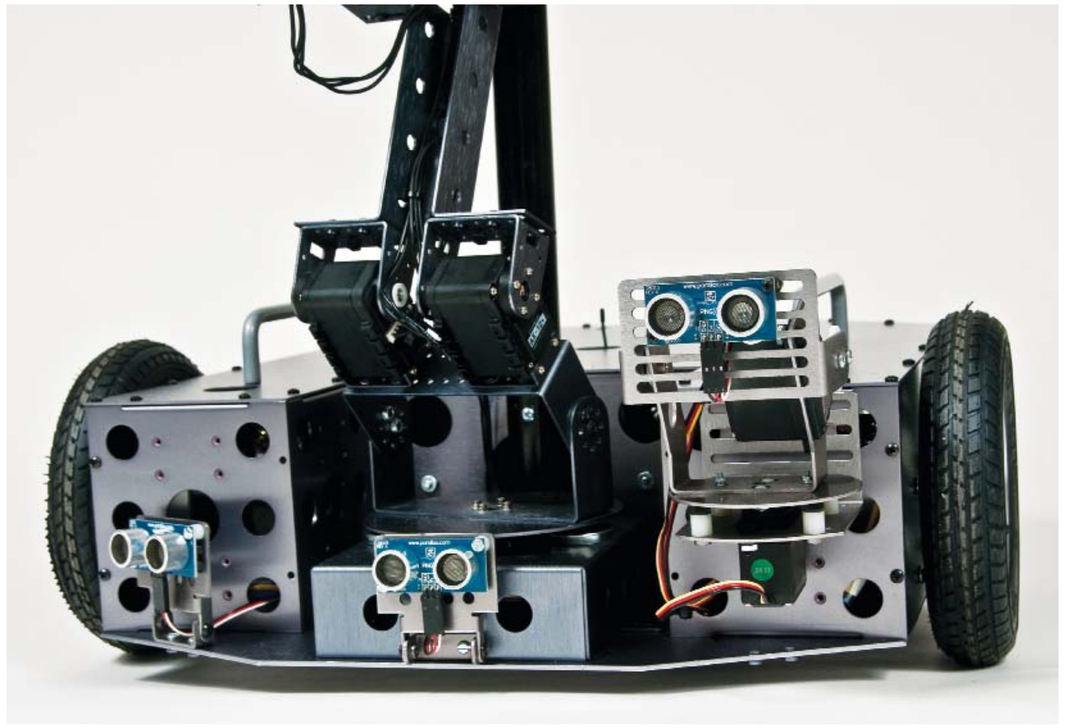
FEATURES

The Nomad is constructed from laser cut .063 gauge (sides and top plate) and .090 gauge (bottom plate) 5052 Type II anodized aluminum in gun barrel gray. Fully configured the Nomad weighs in at 12.6 pounds and has two grab handles mounted on the top deck for easy pickup. The lower inside and outer upper deck both measures 18 inches long and 14 inches wide with a body depth of 4 inches. This leaves plenty of room for electronics and batteries and can even accommodate most laptop computers. One unique feature of the Nomad is its "Robotic Arm Deck" which will accept any of CrustCrawler's robotic arms including the SG5-UT, SG6-UT, and AX-12 Smart Arm. The arm deck will allow the robotic arm to rotate a full 360 degrees. Depending on your configuration it is also possible to fold the arm back flat onto the top deck for storage. Like all of CrustCrawler's robots, the Nomad uses integrated Pem nuts for strength and ease of construction.

PHOTOS BY STEVE NORRIS

It is also loaded with slots and holes strategically located for wire routing and hardware add-ons. As an example, the dual front panels accept CrustCrawler's S3 Tilt/Pan systems for single or dual sensor/camera operation.

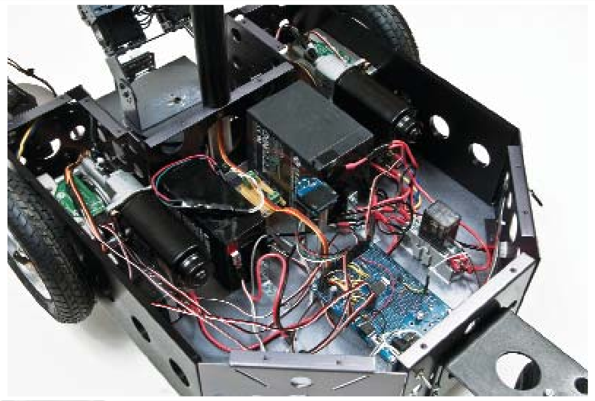
You can purchase the Nomad in number of configurations bundled with the Parallax Motor Mount Kit, the Parallax Caster Kit and robotic arm of your choice. The kits do not include microcontrollers, motor controllers, or sensors. These all will need to be purchased separately depending on your configuration. We will discuss my configuration in detail shortly.


The three front mounted Pings

CONSTRUCTION

The Nomad comes with a 15 page printed assembly guide. The Parallax Motor Mount Kit and the Parallax Caster Kit both have their own construction manuals. They all have clear illustrations and the text is well written. You should first build the Motor Mount Kit but be sure not to install the cotter pins and wheels until the motors have been installed into the Nomad chassis. The build process is straightforward and should take you an hour or two.

The Nomad is essentially a tail dragger. The two Motor Mount wheels are mounted forward with a Parallax Caster Wheel Kit in the rear. The caster kit includes dual three inch Du-Bro sealed pneumatic rubber tires and precision machined aluminum parts. It is a bit expensive but has to be the smoothest and quietest caster wheel assembly I have ever encountered.

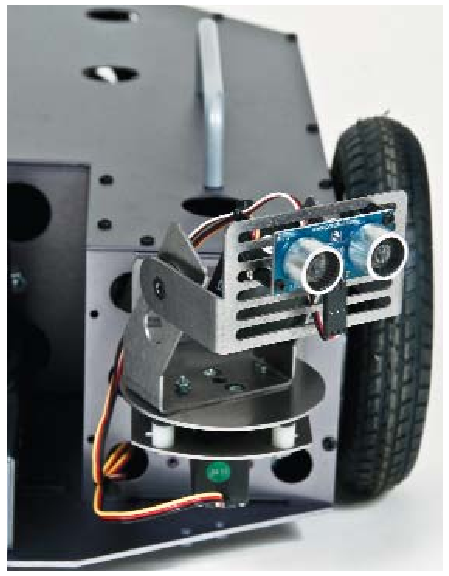

Inside view of electronics-propeller board, power relay and batteries.

MOTOR DRIVE SYSTEM

As mentioned, the drive system for the Nomad is based on the Parallax Motor Mount and Wheel Kit. The kit contains two 12 VDC motors that are combined with precisely machined aluminum hardware and two six inch pneumatic rubber tires mounted on aluminum rims. In addition the kit comes with Position Controllers that use a quadrature encoder system to track the position and speed of each wheel and report the data on demand. Optionally the Position Controllers can be interfaced to two Parallax HB-25 motor controllers to provide smooth speed control and ramping. This is the configuration that I use for all my robots. A word of caution, the HB-25s can be a bit fussy and should be powered on after the microcon-

Rear view showing caster kit, power switch, programming and charging cables.

troller you are using is fully initialized. You can do this by using two switches, one for the microcontroller and its associated electronics and then another switch for the HB-25s. I prefer to use a 12V DPDT relay, which is controlled by the microcontroller. In this case I use one switch to power up for the microcontroller and it will in turn power up the HB-25s and their associated motors using the relay. This configuration also allows for a low power "sleep" mode where the microcontroller shut offs the HB-25s and motors. The whole drive system is controlled by sending simple serial commands to select the speed and travel distance of each wheel independently.


Closeup of left Ping mounted on an S3 pan and tilt bracket.

POWER

Power for my Nomad is provided by two SLA (Sealed Lead Acid) batteries. The drive motors run on 12 volts and require at least 1.5 amps each. To provide this power I used a single 12V 5 AH battery. The processor and all other electronics use a separate 6V 4.5 AH SLA battery for their power needs. These two batteries give the Nomad about two hours of continuous operation. Both batteries are mounted inside the Nomad just behind the arm deck. I used a couple of two inch "L" brackets and rubber feet to hold them securely and allow easy removal if necessary. To charge the batteries I ran two R/C power cables directly from each of the batteries and out to the lower holes in left and right rear panels. A SPST toggle switch is mounted on the rear panel which used to power off the microcontroller and electronics when the robot is not in use.

COMMUNICATIONS

To receive commands and send telemetry data my Nomad uses an XBee RF module. Currently I have three remote host devices that can send and receive RF commands using XBee. I can use my handheld terminal, my PC using a Parallax XBee USB Adapter or a web server based on the Parallax PINK. The XBee modules provide two modes of communication. The first is a simple serial method of transmit/receive and the second is an advanced framed mode. XBees can be configured through a PC utility or directly from the microcontroller. These modules can communicate point to point, from one point to a PC, or in a mesh network.

The Nomad (and most of my robots) uses the XBee in the simple serial mode with my own communications protocol. The protocol was designed to be easy to implement and also allow communications packets to be typed directly from a keyboard using any terminal emulator program like HyperTerminal or the Parallax Serial Terminal. All packets start with a three character header and end with an ASCII Return character. The first character of the header is always an "!" and the second character is the Network ID such as "R" for robot. The third is the destination device (robot) ID. The special character "#" is used to specify a broadcast to all devices. Data following the three character header is device dependent. Here is an example for a Move Forward command sent to device ID 3 in the robot network:

```
!R3FW (return)
```

Telemetry returned from a device is formatted in a similar manner but with the header starting with an "*" instead. Although this protocol is certainly not as robust as others it has proven to be quite adequate for my robot and electronics applications.

SENSORS

A robot is nothing more than a remote control car without sensors. For forward distance measuring and object detection I installed three Parallax Ping ultrasonic sensors to the front of the Nomad. The Nomad provides two sets of Pem nuts mounted at a 20 degree angle to the left and right side of the arm deck. These are drilled to accommodate the CrustCrawler S2 mounting brackets which are provided with the Nomad kit. The S2 brackets are pre-drilled to hold a variety of sensors including the Parallax Ping. I wanted full frontal coverage so I installed an S2 and Ping on the right side using the provided Pem nuts. There are no holes for a center bracket because this would interfere with the mounting of an arm on the arm deck. As an alternative I mounted the second S2/Ping directly to the base of the AX-12 Smart Arm. This required drilling a few holes into the base. We will explore the mounting and use of the Smart Arm on the Nomad in a future article.

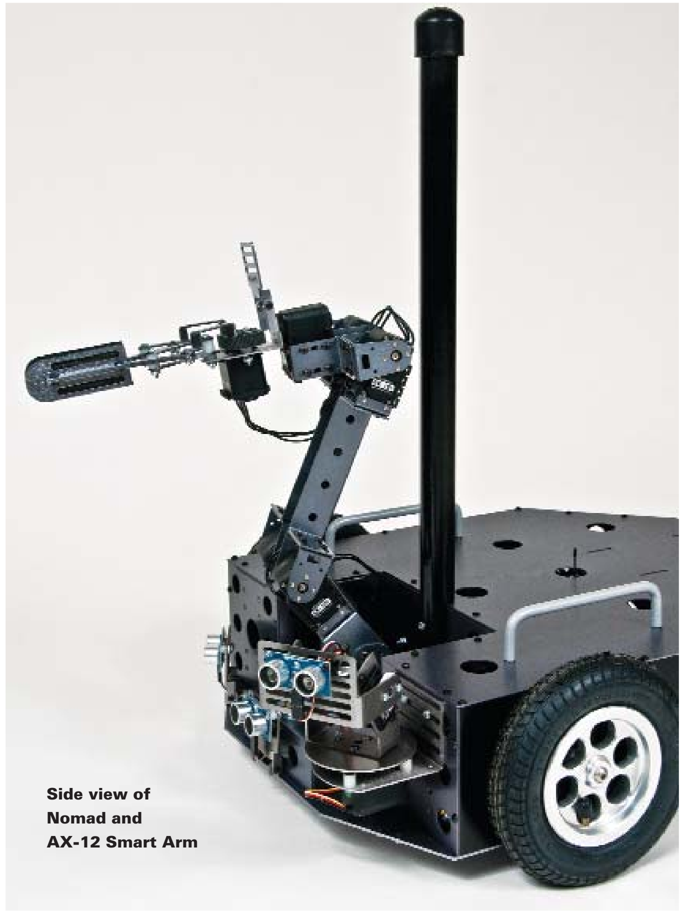
For the left side I opted for the more advanced CrustCrawler S3 Tilt and Pan bracket. The S3 comes with two servos that allow the Ping to be panned left and right 90°as well as up and down about 45°. The S3 easily mounts to the left front panel using a set of provided Pem nuts. As we will see when we discuss the software, I use this Ping for left object avoidance by positioning it at 20 degrees or for wall following by positioning it at 90 degrees.

For simple dead reckoning navigation I installed a Honeywell HMC6352 Compass Module. You can get these from either SparkFun (where I got mine) or Parallax. I've never had much luck in the past with electronic compasses but the HMC6352 boasts accuracy to the nearest 0.1° and a built-in calibration for an average error of 2.5°. I also like that it will internally calculate the heading, thus simplifying the software post processing requirements. I was a bit paranoid about magnetic interference generated from the motor drive system so I mounted the HMC6352 on top of a 16-inch piece of $^3/_4$" PVC pipe using a little hot glue. This may be overdoing it but I can say that I don't have any interference. The pipe is mounted in the center of the chassis and along the wheel centerline to prevent any turning parallax errors.

The Honeywell HMC6352 Compass Module

THE NOMAD BRAIN

Like most of my robots in the past the Parallax Propeller chip was used as the main controller. The Propeller is ideal for robotic work and is in fact actually eight individual microcontrollers called Cogs built into one chip. This means you can have up to eight separate operations running at the same time. For the Nomad this includes sensor processing, motor control and RF communications. Each of the Cogs has its own memory in addition to the main 32K of RAM which is shared by all eight. The Cogs can perform simultaneous tasks cooperatively or individually while sharing resources through a common hub. The Propeller also supports 32 input/output pins. With eight cogs and 32 I/O pins you can connect and manage a wide range of sensors and motor controllers.


Side view of Nomad and AX-12 Smart Arm

The Parallax Propeller Proto Board was used to simplify the use of the Propeller Chip. It contains all the support hardware needed for the Propeller. This includes power supply, EEPROM, 5 MHz crystal, and programming connector. All the Propeller I/O pins are easily accessible using this board. All the wiring was done using point to point soldered wiring. I also added an extension cable to route the Propeller programming pins for the Propeller Plug out through the rear panel.

Don't let my obvious bias to the Propeller Chip sway you away from other choices. There are many other microcontroller options to empower your Nomad. As long as your application is not to taxing you could certainly use the venerable Basic Stamp or the popular Arduino. They are both great processors for the beginner or if you are intimidated by the Propeller. But do keep in mind that both these processors have a single thread of execution and may eventually limit the level of sophistication of your application.

Another option for the Nomad's brain is to use a laptop computer. The Nomad has plenty of room and weight capacity to carry most any laptop on its top deck. A possible configuration is to use an Arduino connected to the laptop via a USB port. The Arduino can handle the reading of sensors and issuing low level commands to the motor drive while the laptop can handle the heavy computational load required by applications for image processing and navigation.

SOFTWARE

The Propeller software for my Nomad uses a behavior-based programming architecture similar to the one I used for my Parallax Stingray. This architecture uses a collection of simple behaviors that take inputs from one or more sensors and then triggers an action through the robot's motor drive system. Each behavior is written as a single Spin method that is called sequentially based on its priority. Using the multiprocessing capabilities of the Propeller you could also start a behavior in its own Cog and run them in parallel. The choice is based largely on the performance required by the behavior. As mentioned before a behavior has an assigned priority which determines when it can get access to the motor drive system. The system I'm working on has five behaviors in the following order:

Mona prepares to rescue her little sock toy from a rouge robot.

1. Blocked
2. Avoid
3. Compass Navigate
4. Follow Left Wall
5. Move Forward

The Blocked behavior has the highest priority and will override all the other behaviors. It uses the forward facing Ping to determine if the path ahead is blocked and issues a simple "move back 12 inches" instruction to the motor drive when triggered. The Avoid behavior has the second highest priority and uses the 20 degree right Ping and the left Ping set to 20 degrees to steer the robot away from potential obstacles found about 3 feet ahead on the left or right. The Compass Navigate behavior attempts to hold the robot on a set course. This behavior has the additional capability to follow map directions and make turns based on distances traveled. The Follow Wall behavior uses the left Ping set to 90 degrees and tries to lock on and follow the wall on the left if there is one. Finally the lowest priority behavior, Move Forward, simply instructs the motor drive to drive forward.

As you can see the behaviors are rather simple but will result in surprisingly complex behavior. In many cases the robot will execute unexpected behavior that at times may seem quite clever and more intelligent than the individual behaviors themselves. This is what is called "emergent behavior." As an example of this emergent behavior, the Nomad is able to execute perfect three-point turns even though there is no explicit code written for it. This advanced behavior is created by the Blocked behavior wanting to back up the robot and the Avoid behavior wanting to turn it away from an obstacle.

WHAT'S NEXT

The next article will continue to expand the Nomad by adding a video camera, an AX-12 Smart Arm, and Beacon and Line navigation sensors. We will also write behaviors that will used these additional sensors.

The source code and schematics for this version of my Nomad can be downloaded from www.botmag.com/issue25.

Links

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

Parallax, www.parallax.com, (888) 512-1024

Spark Fun Electronics, www.sparkfun.com

Steve Norris website, www.norrislabs.com

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