



Then and NOW

TurtleBot: ROS Meets Kinect Meets Create

by Tom Carroll

*Turtle robots have been around since Grey Walter's Tortoise of the late '40s. When hobbyist robots started to be popular in the early '80s, robots such as the Tasman Turtle (shown in **Figure 1**) were popular. They were easy to interface and control by an Apple II or similar computer.*

Newer turtle-type robots such as the BBC Turtle robot shown in **Figure 2** or the Propeller-head Geek Turtle robot shown in **Figure 3** are still popular and are sold at many hobby robot outlets. These robots are simple to construct, interface, program, and operate as they are usually differentially-driven and have a clear plastic shell that shows the interior. They offer beginners a way to learn programming techniques to control a basic robot.

The term 'turtle' has recently taken on a new meaning with the introduction of two turtle-style experimenter's machines that are changing the face of hobby and advanced experimental robotics. I've reviewed many robots over the years, but have rarely come across two entirely different machines at the same time that have really offered so many capabilities and opportunities for serious robotics research. Both machines have a long list of users who have nothing but praises for their robot. With newer, powerful software suites coupled with equally powerful sensors, these two advanced turtle-style robots offer serious experimenters complete robotics packages that would rival the most advanced university level machines of five years ago — but at a fraction of the cost.

One is the Willow Garage ROS-based TurtleBot and the other is Eddie from Parallax that is based on Microsoft's Robotics Developer Studio, RDS4. Neither of these two robots actually utilize turtle shell construction and both feature Microsoft's powerful vision sensor: the Kinect. Each robot actually has so many features that I will do a comparison over the next couple columns. I'll concentrate on the TurtleBot this month. Next time, we'll take a look at Eddie.

The TurtleBot

In reading emails and other news items (and talking with WG personnel in early 2011), I heard rumors that WG was developing a small robot to serve as a base for their open source Linux-based software platform, Robot Operating System (or ROS). When I visited their facility in 2010 and got to know their amazing PR2 robot, I was also introduced to ROS which was the software for the PR2. I knew that WG was anxious for the robotics community to discover ROS and to implement it into various robot designs.

Willow Garage was formed in late 2006 by Scott

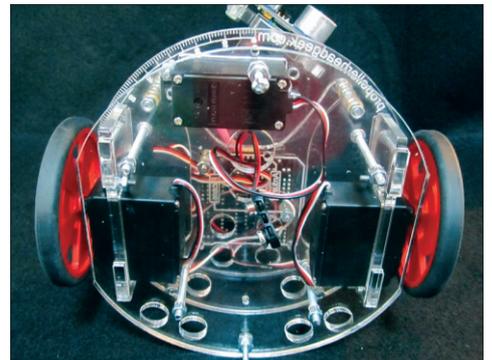
FIGURE 1. Tasman Turtle Robot from *theoldrobots.com*.



FIGURE 2. BBC Turtle Robot. (Courtesy of Paul Silver at flickr.)



FIGURE 3. Propeller-head Geek Turtle robot.



Hassan as a spawning ground for the development of non-military robotics and open source software. He felt that the impact of innovative ideas in robotics will, in turn, provide opportunities for return on investment of capital. He brought Steve Cousins on board as President and CEO, who helped form WG into a model of innovation incubation.

I have been favorably impressed with Willow Garage since that first visit, and briefly mentioned their new experimenter's platform robot and ROS in last month's column. When the TurtleBot was introduced this past summer, I felt that I just had to look seriously at this

advanced experimenter's platform that was significantly cheaper than the PR2. In mid-August, I flew down to Willow Garage in Menlo Park, CA and met with Melonee Wise and Tully Foote — the co-developers of the TurtleBot. TurtleBot is based on the iRobot Create — iRobot's Roomba minus the vacuum cleaner parts — and the Microsoft Kinect — the vision game sensor for the Xbox 360. Having previously seen many photos of the robot, I certainly wasn't expecting anything like the human-sized and very expensive PR2. Though smaller than the PR2, the TurtleBot has capabilities that I've rarely seen on any robot and certainly not one in the range of \$1,200 to \$1,400 for a 'ready-to-go' machine.

Tully Foote, a Systems Engineer at WG, shown on the left of the TurtleBot in **Figure 4** and Melonee Wise, a Senior Engineer at WG, shown on the right co-wrote the initial proposal at Willow Garage, and then worked on the development of the various hardware components and the TurtleBot-specific software within ROS. It was to be an internal WG project that utilized the just-released Kinect to develop an entry-level mobile robot with many of the capabilities of their PR2. The goal was to produce a mobile robot with sufficient sensor/computing capability to perform multi-robot research and demonstrate how a simple robot design could take advantage of the many features of ROS. They wanted an inexpensive robot that was easy to assemble and was created from basic 'off-the-shelf' components. When visitors to WG saw the robots, they, of course, wanted one, so WG polished the design a bit, and now there are three vendors selling TurtleBots worldwide, with dozens of schools and institutions using them.

All of these systems and components have been available for over a year now and the TurtleBot is about as straightforward as you can get for a basic robot design. There is no cute little turtle shell body and there are no arms or appendages of any sort, however, these and any other device can easily be added at a later time. Experimenters can utilize the basic intelligence instruction set of the Create, add an iRobot Atmel ATMega 168-based command module (shown in **Figure 5**)



FIGURE 4. Tully Foote and Melonee Wise with a TurtleBot.
Photo by Jimmy Sastra.

for the most basic routines, or use the included powerful (for a netbook) Asus 1215N to process the Kinect's vision and other data.

Key Parts of the TurtleBot

Let's look closer at the four key components of the TurtleBot. The robot uses the Create platform as the base, power source, and for motive power and steering. Kinect serves as the main sensor suite and does so quite well. Willow Garage decided to use the Eee PC 1215N netbook from Asus with 2 GB of RAM and a 250 GB hard drive. This netbook uses the Intel dual-core Atom

D525 processor that is powerful enough to handle the demands of 3D data from the Kinect. For graphics, the Asus uses the NVIDIA ION Discrete Graphics Processor. Also included is a power and sensor board with an ADXR5613 single-axis gyro that can measure the robot's yaw rates up to 150 degrees/s. The board also provides regulated 12V power for the Kinect (the USB cannot supply the required current) and is mounted inside the Create. The rest of the components are the four round mounting boards and the spacers for the boards and Kinect mounting. The ROS software is provided on a separate Flash drive and is also downloadable from **ROS.com**.

Basic Mechanical Construction

The four 12-3/8" mounting plates are perfect for mounting experiments or even an inexpensive robotic arm. The bottom plate is affixed to the Create and has a 5-1/2" by 7-1/2" hole to allow access to the Create's 'cargo hold' and interconnection of various cables. The front of the lower plate is cut off to allow the Create's homing sensor to 'see' the charging beacon. The top plate is cut off at the front so it doesn't shadow the Kinect's field of vision. Originally, I had accidentally mounted the cut-off portion to the rear (to line up the ROS/WG graphics) but found that it affected the Kinect's operation. You can easily do your own modifications such as adding a bungee cord or long twist-tie to hold the netbook in place.

The bottom three plates are separated by two sets of 2" spacers, and the top plate is separated by four 8" spacers. I also tried removing one of the 2" spacers so I could swivel the netbook in and



FIGURE 5. Create command module installed.

out for access to the keyboard and screen while it was powered up. The top plate has a series of 1/8" holes spaced 3/4" apart in a 4-1/2" by 7-1/2" pattern that can be used for mounting experiments and sensors. The Kinect is mounted on two 3-1/4" spacers and is set to the back of the robot. It's elevated to allow the Kinect's fixed angle of vision to detect objects as close to the front of the robot as possible. The tilt base is not attached; instead, the body of the Kinect is fastened to the third plate. You have to remove two adhesive-attached strips from the bottom of the Kinect to access the threaded holes. The whole mechanical assembly is very simple and only took me a few minutes.

The Base: The iRobot Create

There have been literally thousands of robots constructed by using the iRobot Roomba series of robot vacuum cleaners that were hacked to remove only the vacuum system and brushes. iRobot quickly saw the potential of a market for robot experimenters and developed the Create. The basic model is priced at \$129. This Create does not include a rechargeable battery but rather a green case that holds a dozen alkaline batteries. I would recommend adding the 3,000 mAh Ni-MH battery and the iRobot fast charger for a total price of \$219, as the Create can literally 'eat' non-rechargeable alkaline batteries like they were candy. The \$60 iRobot Command module that I mentioned earlier is a great addition if you don't want to use just the ROS software that comes with the TurtleBot. iRobot has what they call the Premium Development Package which includes the above plus a remote control, two virtual walls, and the self-charging base for the charger.

The Create is based on the iRobot Roomba 400 series and is compatible with many of that series' accessories. According to the specifications, it is designed to handle a maximum payload of five pounds and has an extra wheel supplied to stabilize taller robot designs like the robot shown in **Figure 7** — another Create/netbook/camera configuration. The TurtleBot arm shown in **Figure 8** was originally developed by Michael Ferguson. WG summer intern, Helen Oleynikova, worked with the arm and demonstrated it for me. Needless to say, the 14.1 pound TurtleBot weighs a bit more.

The Create's wheel shown in **Figure 9** is beefy. The belt drive from the motor is connected to a planetary gear arrangement, and the shaft encoder (gray and black wires) is used for odometry. With 32 installed sensors (you can add more of your own), a 25-pin expansion port, and a series of 10 nice demos built in the Create is an ideal experimental platform on its own. Using either the command module or your own microcontroller and the Quick Start Guide (which gives you access to the compiler and a series of sample programs using C or C++ or some other programming languages via the iRobot Open Interface) will give you a great start for an advanced robot. If you want to use the Kinect, I highly recommend the use of ROS for its unique abilities with the Kinect sensor's more complex data stream (though Microsoft's RDS4 has been tweaked to also utilize the vast amount of visual data). I would never recommend hacking an old

Roomba in an attempt to convert it to a Create-like base, as you will probably end up with a poorly operating machine and not have anything like a Create. Keep the Virtual Walls/Lighthouses though, because they will work with the Create. At \$129, you'll be far happier buying a brand new model. Co-founder of iRobot, Helen Greiner, gave me a Create several years ago and I have found it to be the perfect 'already built' base for most robot experimenter's projects. I was quite grateful for the Create because it has so many applications in robot research. It is easy to see why WG chose this capable platform for the TurtleBot.

Eyes and Ears: The Xbox 360 Kinect

The advent of Microsoft's Xbox 360 Kinect sensor was the true impetus for the development of the TurtleBot and the Parallax/Microsoft Eddie. The Kinect is not just a camera for some robot's processor to make sense of a data stream sent by the camera. It is an intelligent vision/navigation/mapping system that — at a mere \$130 — can best systems that cost a hundred times more. Within hours after the Kinect was introduced in November '10, it was hacked, and many photos and hacked data soon appeared on the Internet. By the 60 day point — with sales averaging 133,333 units per day — it had made the Guinness World Records list as the best-selling consumer electronics device of all time.

You can use a Kinect with a Create utilizing a tutorial by Melonee Wise at www.ros.org/wiki/kinect/Tutorials/Adding%20a%20Kinect%20to%20an%20iRobot%20Create. This machine was her stepping stone to the TurtleBot. Microsoft has donated 2,500 Kinect sensor systems to FIRST high school competition robotics teams for use during the 2012 season — an ideal next step up for these steadily advancing series of robots. There is so much information available on the Internet about the Kinect itself, Kinect hacking, and robot applications that I won't go any deeper into it here. It is the ROS and the Kinect working together that makes the TurtleBot what it really is.

ROS: the Soul of TurtleBot

I had some uncertainties about just how adept I might be when installing and using ROS on the Asus netbook in the TurtleBot. As a metal-bender, electron-shover, software and programming have never been my strong suit. Fortunately, Melonee and Tully spent an afternoon guiding me through some of the basics, so I was able to adapt. I did make a few mistakes along the way, however.

After downloading the 99 page tutorial, I sat down and tried to immerse myself in it. Some parts came easily to me and some I really had to think about. I highly recommend all potential buyers of the TurtleBot first sit down with the free downloads of the open source ROS and the variations from **ros.org**, and get used to the way it operates. ROS is far more than a programming language as it is an amazing and quite popular extensible collection of libraries and tools for robotics. One of the nice features is the ability for the community of users and developers to be able to contribute a system package.

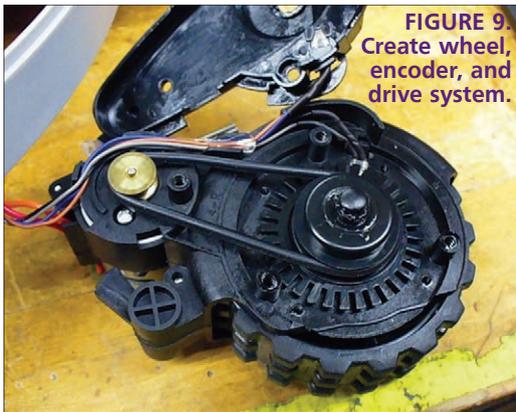


FIGURE 9. Create wheel, encoder, and drive system.

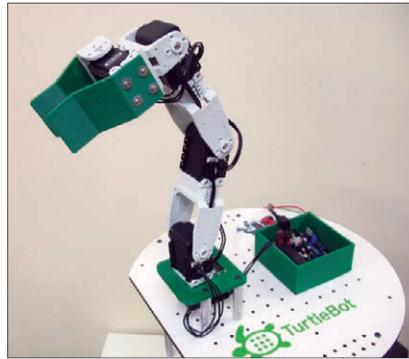


FIGURE 8. TurtleBot arm with Robotis AX-12 actuators.



FIGURE 7. Calliope Development Platform from robotshop.com.

ROS has progressed beyond ‘switchyard’ — the name given to it at Stanford’s SAIL (Stanford Artificial Intelligence Laboratory) in 2007. Further development of ROS is now centered at Willow Garage. It supports Ubuntu Linux and is an entirely open source software that is free for commercial and research use.

After the quick mechanical assembly, the robot is now ready for its software installation. The TurtleBot kit includes a green USB drive that contains ROS and Diamondback (the latest release). At ROS.org, you can go to Robots/TurtleBot/diamondback/Robot Setup – ROS Wiki for setup and installation instructions. One thing to remember is to NOT boot up the brand new laptop to bring up Windows because this will make it harder to access the BIOS screen. In fact, you can run into a bit of trouble by calling up Windows Office 2010 to purchase it at a later time — as I did — not thinking about what the netbook was intended for in the first place: the robot’s brain. Insert the USB thumb drive into the netbook and turn on the computer while repeatedly depressing “Esc” until the dialog “Please select boot device” appears. Select USB device and hit “Enter.” Follow the ‘robot setup Wiki’ instructions very carefully. You’ll shut down, remove the USB drive, re-start, and start up Ubuntu. The next steps are setting up NTP (Network Time Protocol), the root password, and the SDK (software development kit) on the computer. The following steps from the ROS Wiki site are designed to assist you in getting started with the TurtleBot and are a great introduction to understanding ROS:

1. TurtleBot Setup: These instructions will guide you through unpacking your netbook and installing the TurtleBot software.
2. TurtleBot SDK Setup on your computer: These instructions guide you through how to set up your computer to control the TurtleBot.
3. TurtleBot Networking Setup: Establishing communication between the TurtleBot and workstation.
4. TurtleBot Environment Setup: Set up your environment.
5. TurtleBot Bring-up: How to start a TurtleBot.
6. TurtleBot Teleoperation: How to teleoperate your TurtleBot with a keyboard or a joystick.
7. TurtleBot Follower Demo: How to make your TurtleBot follow whatever’s in front of it.
8. TurtleBot Odometry and Gyro Calibration: This will show you how to calibrate or test the calibration of a TurtleBot, which is highly recommended when running

any navigation-based application.

9. SLAM (simultaneous localization and mapping) Map Building with TurtleBot: How to generate a map using g-mapping.
10. Autonomous Navigation of a Known Map with TurtleBot: This tutorial describes how to use the TurtleBot with a previously known map.
11. Visualizing TurtleBot Kinect Data: This tutorial shows you how to look at data coming from the Kinect camera.
12. Using TurtleBot Interactive Markers: This tutorial describes how to use rviz interactive markers for controlling the TurtleBot.

This article is not intended as a primer on ROS because the subject is so vast. ROS Electric is the next upgrade after Diamondback, and should only be installed after a thorough knowledge of Diamondback, in my opinion. Some people have found learning ROS to be a fairly daunting task. TurtleBot and ROS are really for serious robot experimenters; it is not for a smaller robot running on a microcontroller only. You’re going to require a fairly powerful netbook at the least, but once you have your machine assembled and are running the software, you are literally going to be blown away with the numerous advanced capabilities of your machine. If you are even a little familiar with Unix or embedded Linux and know the C++ that ROS is written in, TurtleBot will be ready to go after loading the software. You’ll want another computer (Mac or Windows) as a workstation for Wi-Fi teleoperation, preferably with a joystick. If you have some trouble or questions about TurtleBot, ROS, Kinect for Windows SDK, applications, SLAM and gmapping, or any other topic, the TurtleBot wiki pages at ros.org are a great resource.

Final Thoughts

The TurtleBot may look like a simple robot, but the three main components working in conjunction with each other and ROS deliver a machine that is capable of tasks limited only by your imagination. Of course, you can build your own machine by buying or substituting some of the components, but the nice thing about purchasing the full TurtleBot kit is the included software which is already set up for the Create’s base and the positioning of the Kinect. After the software is loaded, it’s ready to go. **SV**