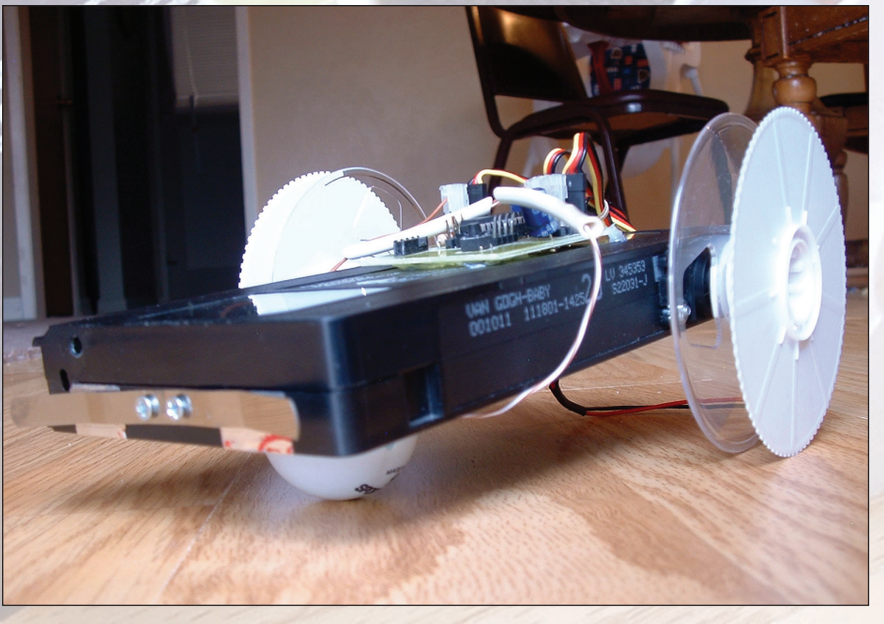
THE CASSETTE BOT

By Bryan Jackson

How do you make a robot out of a VHS cassette tape? What is a VHS cassette tape? Is this guy cheap or what? These questions and many more will be answered in this article about making a cheap VHS cassette robot.



It all started with a bang. My wife and kids had been to Grandma's house, and after a good day of terrorizing the grandparents, had brought home some borrowed video tapes. (Those are the things your parents used before DVDs were invented.) Immediately upon entering the house, my darling children accidentally dropped several of the video cassettes on the floor resulting in one VHS cassette suffering damage from a broken dust flap. Immediately after the obligatory lecture about being careful with other people's stuff, several pairs of expectant eyes turned toward me.

"Can you fix it?" my wife asked.

"If you have a cassette you don't care about I can pull parts from it," I replied.

So, I took the sacrificial tape apart and salvaged the aforementioned dust flap. I then expertly replaced the broken part on the offending tape and put everything back together again. Of course, it didn't play so I took it apart again and put everything in the right place this time, and got the thing working (thus remaining in the in-law's will and on their good side for at least the next week).

So, what do I do with the rest of the parts? Why make a robot, of course. It was so obvious looking at the parts strewn across the table. The small imp in charge of VHS

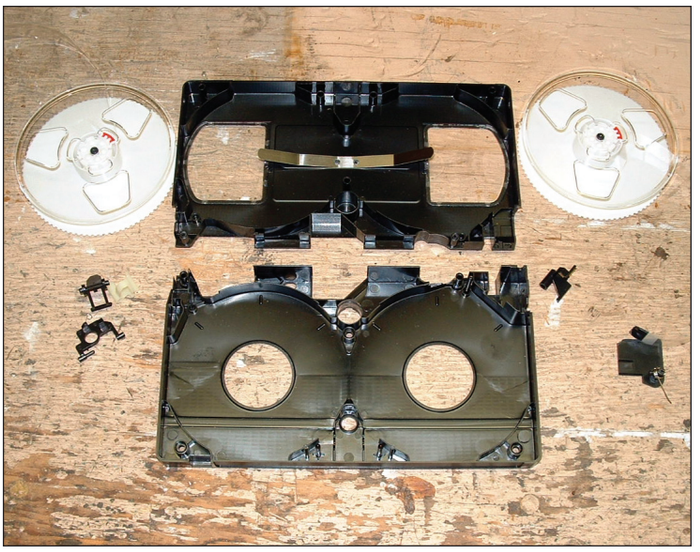


Figure 1. The disassembled VHS cassette.

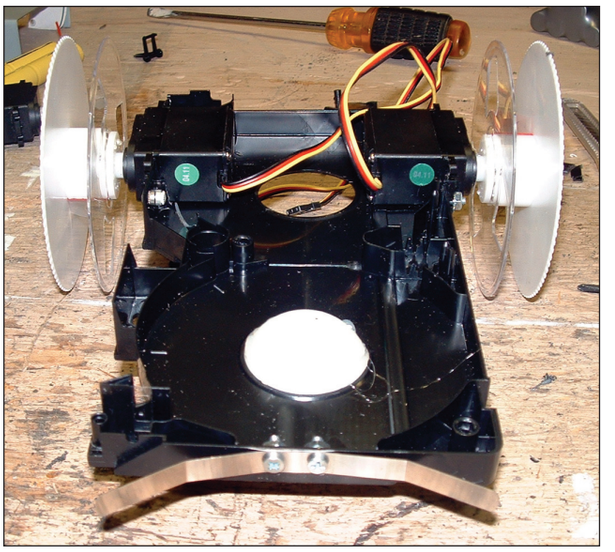


Figure 2. Open cassette case showing mounted motors and sensor.

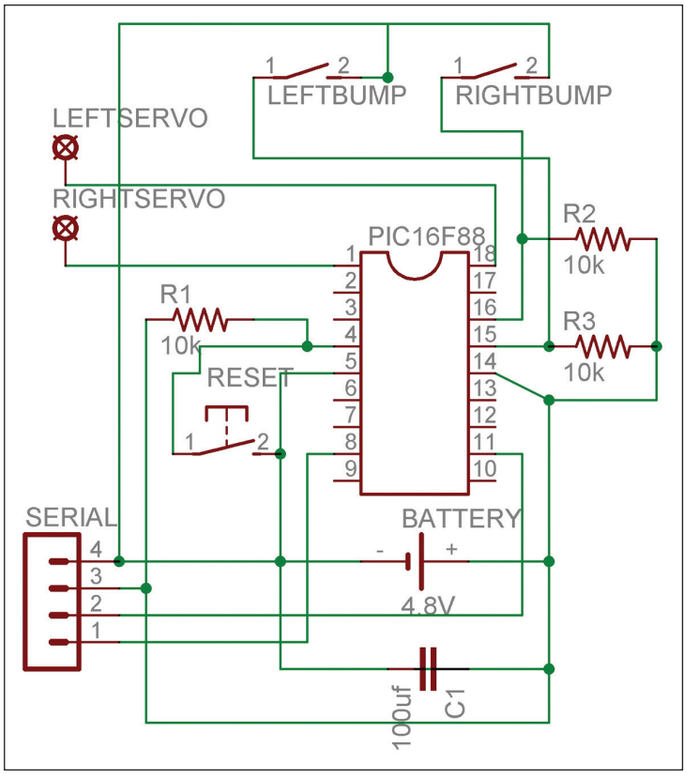
tapes probably planned it that way. The two tape reels were the perfect size for wheels. The body of the tape is a good size for the frame and there is even a long strip of springy steel that would make a perfect bump sensor (**Figure 1**).

The first step was to rip out all the tape from the reels. This takes a surprising amount of time and you may want to enlist the finely honed skills of a two-year-old child in this endeavor. Once the reels are empty, cut the tape from the reels and throw it away. The rest is a matter of personal style and available materials.

I first pulled some servos from an old robot frame. These servos had already been modified for continuous rotation. I cut rectangular holes in the side of the cassette tape body for the servos and drilled a couple of holes for the mounting bolts. Next, I drilled a small hole in the center of each reel. I bolted the servos to the body, and then removed the screw from the servo horn and used it to screw the reel to the servo. (Rinse and repeat for the other side.) A ping pong ball hot glued to the hole in the bottom front of the body is the third "wheel." This works quite well for a small light robot that generally stays on smooth surfaces.

The bump sensor is very simple but effective. I drilled a couple of holes in the front of the case using the holes in the spring strip as a guide. Then, I bolted the strip to the front of the case (**Figure 2**). I used a wire under the nut to connect the strip to ground on the controller. For the other half of the bump switch, I used duct tape (the aluminum tape for real duct work). If you don't have a roll, you need one now. Just remember the glue on the sticky side is a pretty good insulator, so make connections on the top. I cut a couple of strips, stuck them under the bump sensor spring on each side, and ran the excess under the body of the bot. Then, I used another piece of tape to stick the stripped end of a wire to the duct tape (**Figure 3**). A pull-up resistor and some wiring complete the sensor (**Schematic 1**).

The battery is from a 9.6V rechargeable pack that I cut in half lengthwise to make a flat 2x2 4.8V pack. I have had good success running 5V microcontrollers from a 4.8V NiCd pack without any voltage regulation.



Schematic 1.

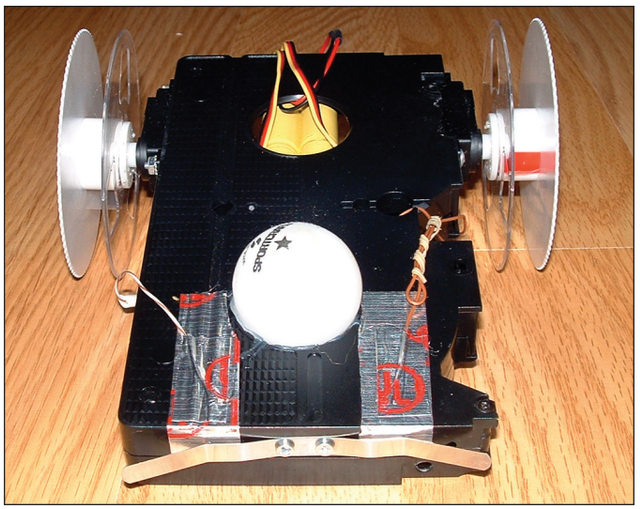
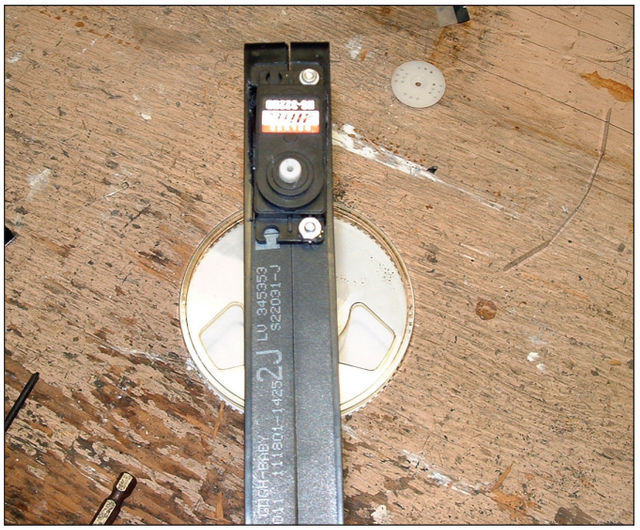


Figure 3. Bottom view showing bump sensor connections.

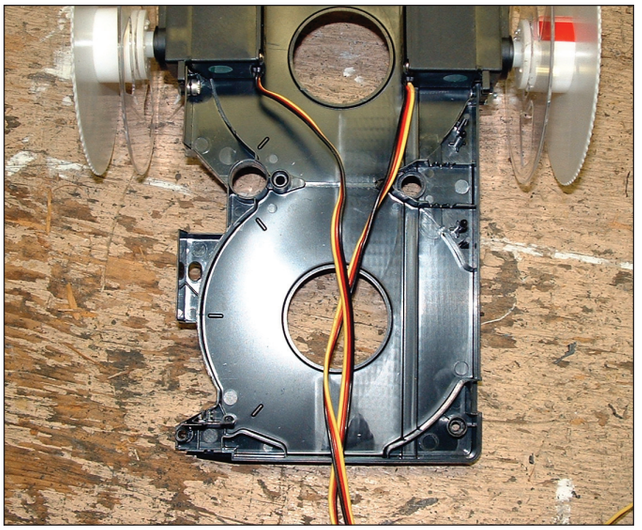
The controller is a PIC16F88 loaded with the Tiny bootloader and programmed using a MAX232a TTL to RS-232 converter. The circuit board was left over from another project, and fit quite nicely. Schematics and artwork for the board can be found on my website (**www.highschoolrobots.com**).

Programming was quick and dirty. I use JAL (Just Another Language), and Tiny bootloader as my development platform. I like the JAL compiler because it's free, and the code is easy to read. The self-programming capability of the 16F88 coupled with a bootloader make it very easy to program. The source code for this project is also available on my website.

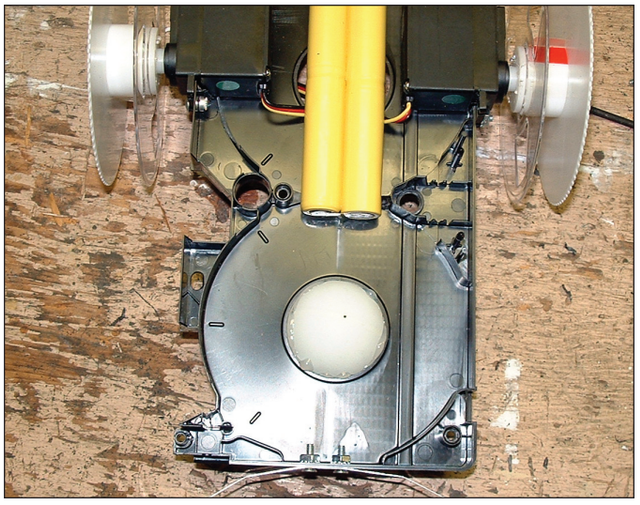
After half an hour or so, I had the code working and the bot was happily bumping into stuff. The code is pretty simple. I just have the bot go forward until one of the sensors is tripped, then back up a little and turn in the opposite direction of the sensor that was bumped. I noticed



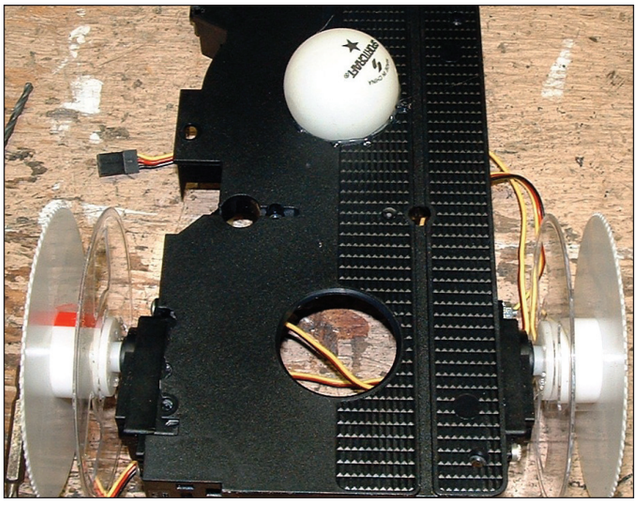
Side view showing servo motor mount.



Open cassette case showing motors.



Open cassette case with battery pack.

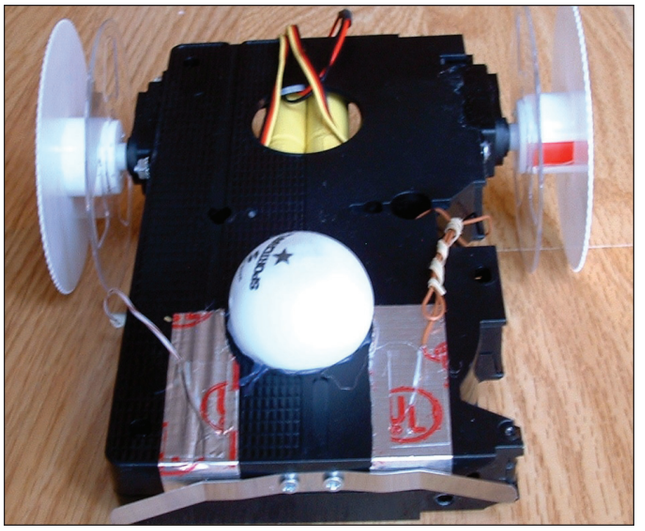


Bottom view showing ping pong ball.

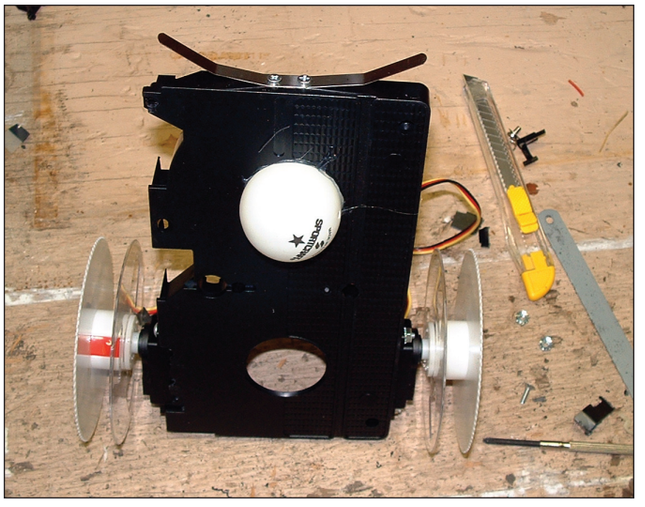
that the bump sensor was a little too stiff, so I pulled it off and put a couple extra bends in it to put the contact portion closer to the duct tape. This helped quite a bit. The bot still gets stuck if it hits a corner dead center, and it can get stuck with a wheel on a chair leg and drive circles around the leg forever.

My next modification will be to extend the bump sensor out to cover the wheels and maybe put something in the middle to get rid of that blind spot.

It was a fun little project and the kids got a kick out of it. They had a lot of fun poking the bump sensor to make the car back up and turn away. This little bot is capable of quite a lot of modification. There is plenty more space inside the case. The clear inside disk of the wheels is screaming for some sort of optical odometry; the clear windows on top need some sort of display; and, of course, it needs more sensors. I wonder what I did with that old Polaroid camera with the sonar range finder ... **SV**



Bottom view of completed car.



Bottom view showing bump sensor.