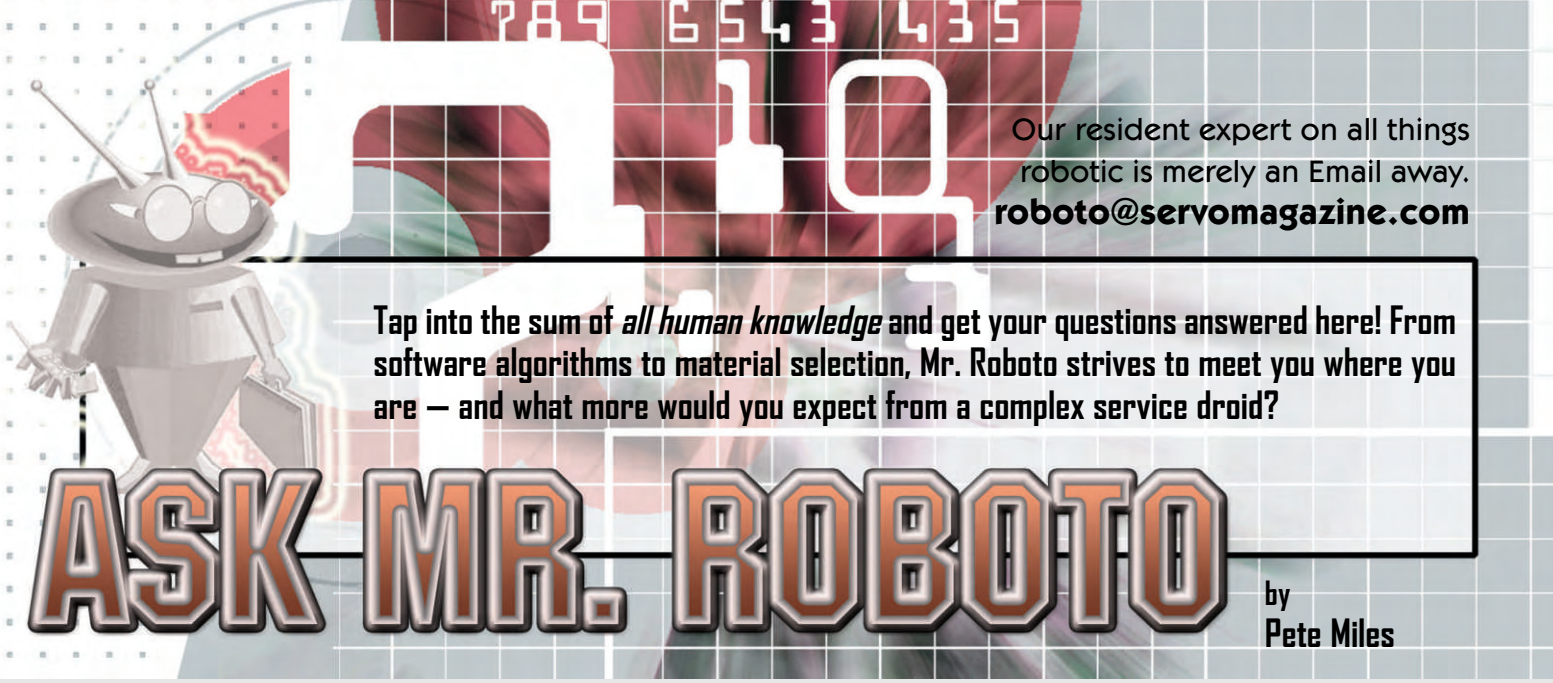


Our resident expert on all things robotic is merely an Email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

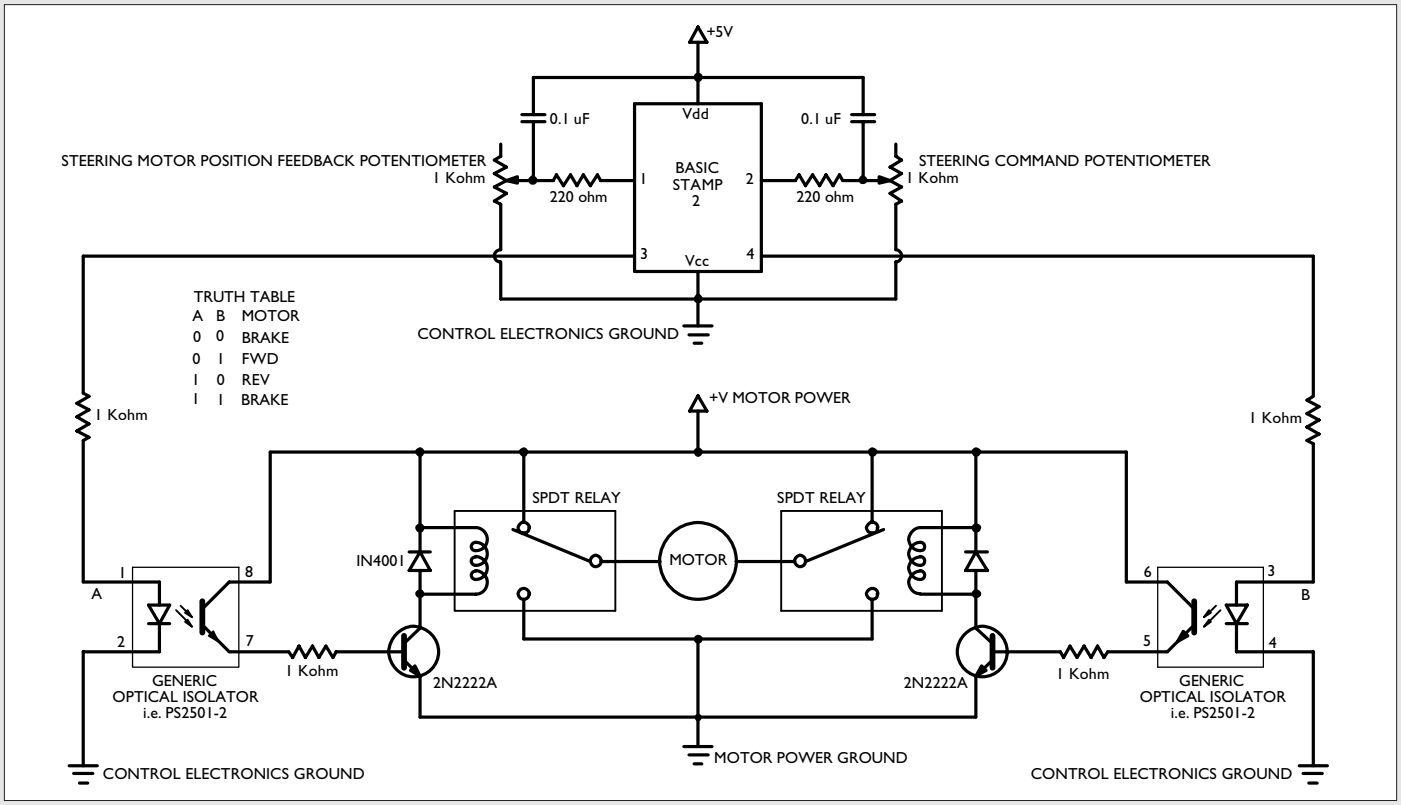
by
Pete Miles

Q. I am trying to improve upon the steering capabilities of our new double stroller. I would like to add "power steering" to it and, eventually, a power drivetrain. My basic idea (as I am new to electronics and robotics) is that I will have to hook up a pretty beefy DC motor for the steering. Someone at Parallax recommended that I incorporate the Motor Mind C that will control up to two 24 V motors at 4 amps, continuous, for the steering along with the Motor Mind C Carrier Board to simplify the process.

Well, the process couldn't get simple enough for me because I have no idea how to do any of this. Anyway, I would like to activate this motor using two pressure sensitive pushbuttons — one for each hand. Push the left-side button and the wheels turn to the left and so on. The harder you press, the quicker the wheels turn. Of course, the wheels should return to the default, straight ahead position when pressure on either button is removed, just like an R/C car, but with a straight DC rather than servo motor.

I think the stroller should also be able to roll freely, unless the motor is activated — much like an electric bike. Am I getting in over my head here? Conversely, if I did decide to have the stroller only powered, I would like the motor to be

Figure 1. Relay-based steering servo motor controller.



able to act as a brake, also.

Is the Motor Mind C an actual motor controller, along the lines of the much, much more expensive AmpFlow controllers? How hard would it be for a novice to learn how to program the code for such a project?

I'm sure I will be a regular customer and visitor to your site. If you could offer any advice, that would be great.

— Mike Harman
via Internet

A. No, you are not over your head on this type of a project. You just need to break it down into several smaller projects. This project has two major tasks: powered steering and powered drive. Strollers typically have four wheels, where the front wheels are mounted on casters and the rear wheels are fixed. Let's begin with the powered steering aspect of this project. You mentioned using pressure sensitive switches to control how fast the wheels turn. I am going to assume that what you meant here is that the harder you press the button, the greater the turning angle the front wheels will go through. Since you wanted the wheels to return to their straight ahead position when the pushbuttons are released, then you are going to need some sort of feedback to sense the position of the wheels/motors.

For the DC motor, I would recommend a low speed gear motor with an output shaft speed that doesn't move any faster than about 60 RPM. Automotive windshield wiper or window opening motors work well here and they typically run at 12 V. The motor can be connected directly to the front wheel's caster axle (which will require two motors) or the front wheels can be configured like a rack and pinion system, typically used in cars. In that case, only one steering motor would be needed.

A potentiometer can be used as a simple feedback sensor for monitoring the steering wheel's position. This should be mounted directly to the steering system in such a way as to be able to measure the full range of the steering mechanism while maximizing the full range of motion of the sensor. Mounting directly to the caster's axle is one suggestion.

Probably the hardest part of this project is installing the pressure sensitive pushbuttons. The pressure sensitive sensors that you are talking about here are known as Force Sensors. These sensors are generally very expensive and designed more for industrial applications. One low cost option is the IESP-12 force sensor, made by CUI Stack, Inc. (**www.cui.com**), and available from Digi-Key (**www.digikey.com**) and HVW Technologies (**www.hvwtech.com**). This sensor acts like a resistor, where the resistance changes as the force on the pushbutton increases. Another option is to use a rotary or linear potentiometer, a lever, and a spring attached to the lever to move the lever back to the home position when the pressure is released from the lever.

Figure 1 shows an electrical schematic on how to implement a simple steering control circuit using a potentiometer as a steering feedback sensor, a potentiometer for the pushbutton control, and a couple of SPDT relays for the motor direction control. A BASIC Stamp 2 is used as the microcontroller in this application. The capacitors wired to the potentiometer are there for the RCTIME function of the BASIC Stamp. They can be removed from the circuit if you are using a microcontroller that has an analog-to-digital converter. The PS2501-2 is an optical isolator to protect the microcontroller from voltage surges when driving the motor.

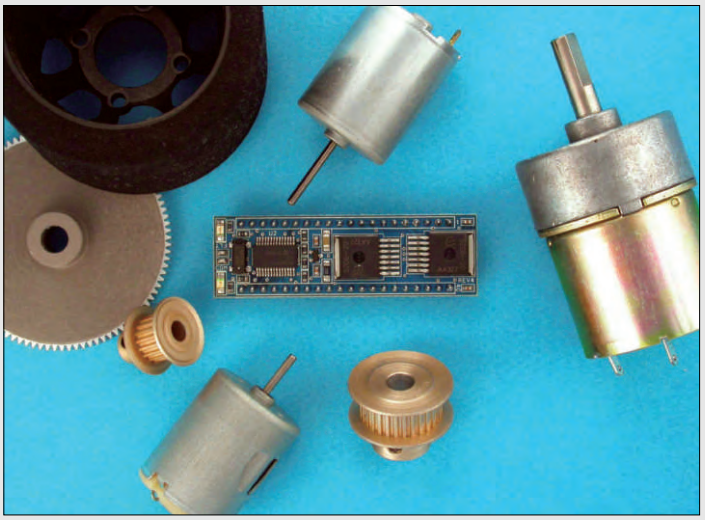


Figure 2. The Motor Mind C from Solutions Cubed.

The program shown in Listing 1 is a simple example program for controlling the steering direction of the motor using a single input potentiometer as the pushbutton sensor.

All this program does is compare the position of the steering feedback potentiometer position and the pushbutton position. When the difference between the two readings is greater than the dead band, then the logic will tell the motor to move until the difference is less than the dead band. The dead band is needed to reduce the amount of motor oscillating about the end position. This value can be increased or decreased, depending on how much sensitivity you want in your system.

Listing 1

```
'{$STAMP BS2}
'{$PBASIC 2.5}

' Sample program that demonstrates how to use a regular DC gearmotor,
' a position feedback potentiometer, and a steering position command
' potentiometer to perform as a servo motor.

Motor_Pos       VAR  Word       ' Steering motor position feedback
Steering_cmd    VAR  Word       ' Commanded steering motor position
Dead_Band       VAR  Word       ' Dead band size variable
tmp             VAR  Word       ' temp variable

' Initialize steering motor to its current brake position
LOW 3                       ' H-Bridge relay control A
LOW 4                       ' H-Bridge relay control B
Dead_Band = 20              ' Set dead band width to 20

Main:   ' Main loop
  GOSUB Read_Motor_Position
  GOSUB Read_Steering_Command
  IF Motor_Pos > Steering_cmd THEN
    tmp = Motor_Pos - Steering_cmd
    IF tmp > Dead_Band THEN
      GOSUB Turn_Left
    ELSE
      GOSUB Stop_Motor
    ENDIF
  ENDIF
  IF Steering_cmd > Motor_Pos THEN
    tmp = Steering_cmd - Motor_Pos
    IF tmp > Dead_Band THEN
      GOSUB Turn_Right
    ELSE
      GOSUB Stop_Motor
    ENDIF
  ENDIF
GOTO main

Read_Motor_Position:      ' Read current steering motor position
  HIGH 1
  PAUSE 1
  RCTIME 1, 1, Motor_Pos
  RETURN

Read_Steering_Command:    ' Read current steering commanded position
  HIGH 2
  PAUSE 2
  RCTIME 2, 1, Steering_cmd
  RETURN

Stop_Motor:               ' Stop the steering motor
  LOW 3
  LOW 4
  RETURN

Turn_Left:                ' Turn the steering motor to the left
  LOW 3
  HIGH 4
  RETURN

Turn_Right:               ' Turn the steering motor to the right
  HIGH 3
  LOW 4
  RETURN
```

Depending on the potentiometer values and the range of motion of the steering mechanism and input potentiometers, the measurements for the potentiometers may need to be scaled up or down. This can be done by multiplying the values by some constant. The constants will depend on the actual system built. A second input potentiometer for controlling the motor position in the other direction can be added to the circuit. As for the logic, averaging the two input positions together will work for control of the motor position. In essence, this is a simple circuit for making a servo motor using a regular DC gear motor. The second part of this project is the drive system. Since the front wheels are being used for steering, only a single motor is needed to drive the rear wheels. Because strollers are fairly light, the same windshield wiper/window opening motors should be sufficient to drive the stroller. A heavy duty motor choice would be to use a wheel chair motor from National Power Chair (**www.npcrobotics.com**).

The advantage of these motors is that they are low speed and have a lot of torque for driving a stroller. The drawback to these motors is that they really don't spin freely, due to their gear boxes. So, the requirement that the stroller move freely when the motors are not in use will require some sort of a mechanism to disengage the motor. Another option here is to use a plain DC motor with a 3/8 inch or 1/2 inch diameter shaft and press the shaft against the wheel of the stroller or use a large disk on the wheel's axle and use a flat belt between the disk and motor shaft. There is more chance for the motor to slip with this case, but — when the power is removed from the motor — it will freely spin. If you want the motors to act like a brake when using the stroller in the powered mode, then you should use a gear motor.

The Motor Mind C (**www.solutionscubed.com**) is a true motor controller and is very similar to the AmpFlow (**www.ampflow.com**) motor controller. Both of these motor controllers will work in this application. They both can control one or two different motors. They both can accept RS-232 serial communication from a microcontroller, a 0-5 V analog input signal from a potentiometer, or a standard 1-2 ms R/C style pulse width for controlling motor speed. The main difference between the two controllers is how much current they can handle. The Motor Mind C has a maximum current draw of 4 amps, whereas the AmpFlow motor controller has a maximum current draw of 160 amps. Which motor controller you use really comes down to how much current your drive motor requires.

The Motor Mind C Carrier Board is a convenient board that holds the 40-pin Motor Mind C module

and a BASIC Stamp 2 module. The board has an RS-232 port along with screw terminals for the motors and motor batteries. The RS-232 port is for programming the BASIC Stamp that is used to control the Motor Mind C. There is a 16-pin header for the Stamp to connect to different sensors. This system can be used to control both the steering and drive motors. The Motor Mind C can replace the SPDT relays for driving the steering motor.

The AmpFlow motor controller has another advantage over the Motor Mind C; it has both velocity and position control for the two motors it controls. Thus, this controller can be wired directly to a drive motor and steering motor with an encoder and no special electronics will be needed to control the motors (other than the pressure sensitive pushbuttons).

Solutions Cubed also has two other products that will work well at directly controlling the position and velocity of the steering motor. They are called the Mini PID and ICON PID motor controllers and they have 3.75 and 12 amp continuous current handling capabilities, respectively. Another option is to look at the motor cotrollers from Vantec (**www.vantec.com**). They offer both velocity and position controllers. They have a unique controller called the RBSA Bully servo amplifier that works just like the one shown here.

As for the input sensor for setting the speed of the stroller, you could use a similar pushbutton type of control like one being used for the steering or you can use a spring controlled lever like those seen on lawn mowers. Here, a potentiometer and a spring connected to the lever are used for the speed control sensor. You have to hold it down in order for the motor to run. You should set up about half of the range of the lever to keep the motor in the stopped position and the final half of the range of motion would be the variable motion that is used to control the speed of the motor. This way, if you let go of the lever, then the stroller will immediately stop.

This will be a fun project. All you need to do is break it down into several smaller parts, get each part working by itself, and then combine the parts together. I hope this gives you enough information to get started. SV