

Controlling the Speed of a DC Motor

PROBLEM

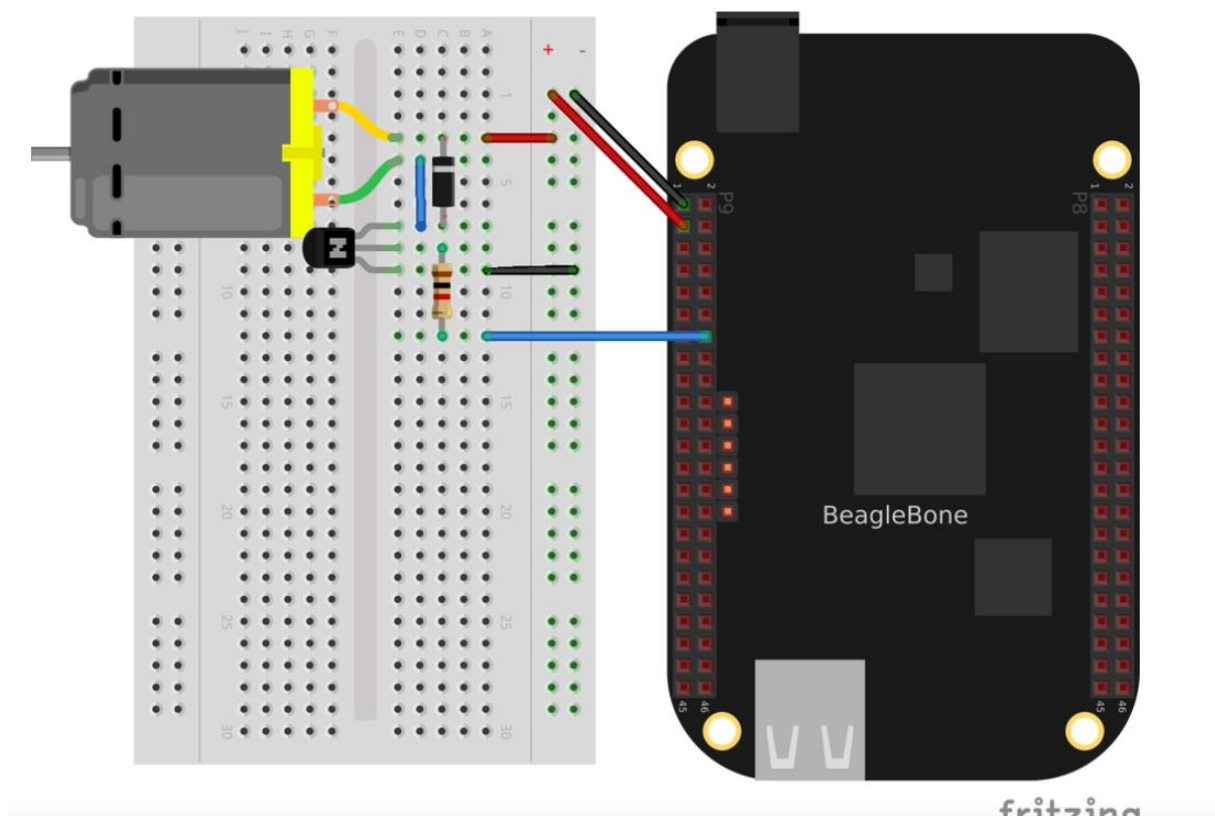
You have a DC motor (or a solenoid) and want a simple way to control its speed, but not the direction.

SOLUTION

It would be nice if you could just wire the DC motor to BeagleBone Black and have it work, but it won't. Most motors require more current than the GPIO ports on the Bone can supply. Our solution is to use a transistor to control the current to the bone.

Here's what you will need:

- 3V to 5V DC motor
- Breadboard and jumper wires (see [Prototyping Equipment](#))
- 1k Ω resistor (see [Resistors](#))
- Transistor 2N3904 (see [Transistors and Diodes](#))
- Diode 1N4001 (see [Transistors and Diodes](#))
- Power supply for the motor (optional)



```
Code : #!/usr/bin/env node

// This is an example of driving a DC motor

var b = require('bonescript');
```

```

var motor = 'P9_16', // Pin to drive transistor
    min = 0.05,      // Slowest speed (duty cycle)
    max = 1,          // Fastest (always on)
    ms = 100,         // How often to change speed, in ms
    speed = 0.5,      // Current speed;
    step = 0.05;      // Change in speed

b.pinMode(motor, b.ANALOG_OUTPUT, 6, 0, 0, doInterval);

function doInterval(x) {
    if(x.err) {
        console.log('x.err = ' + x.err);
        return;
    }
    var timer = setInterval(sweep, ms);
}

function sweep() {
    speed += step;
    if(speed > max || speed < min) {
        step *= -1;
    }
    b.analogWrite(motor, speed);
    console.log('speed = ' + speed);
}

process.on('SIGINT', function() {
    console.log('Got SIGINT, turning motor off');
    clearInterval(timer); // Stop the timer
    b.analogWrite(motor, 0); // Turn motor off
});

```

DISCUSSION

This is actually the same code that used to drive the servo motor ([Controlling a Servo Motor](#)). The only difference is how the parameters are set. We are now using the PWM hardware to control the speed, rather than the position, of the motor. If the duty cycle is 1, the voltage to the motor is on all the time and it runs at its fastest. If the duty cycle is 0.5, the voltage is on half the time, so the motor runs slower. A duty cycle of 0 stops the motor.

This same setup can be used to drive a solenoid. After all, a solenoid is a DC motor that goes back and forth, rather than spinning. Generally, you would drive a solenoid with an on or off signal, rather than using a PWM signal.

At the end of the code, the `process.on()` function detects when the user has pressed ^C (CTRL-C), stops the timer, and turns off the motor. If you don't turn off the motor, it will keep spinning.