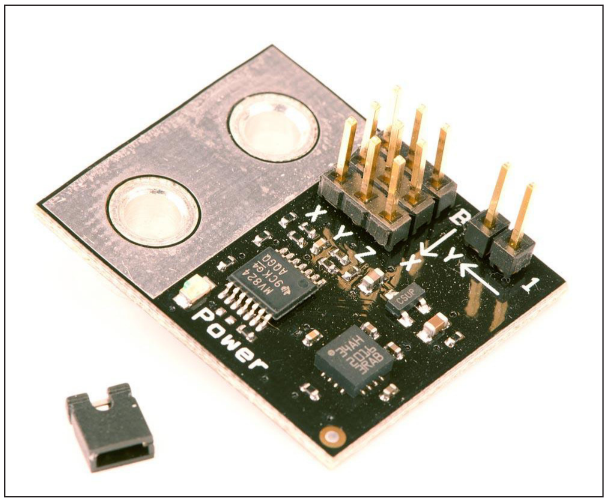
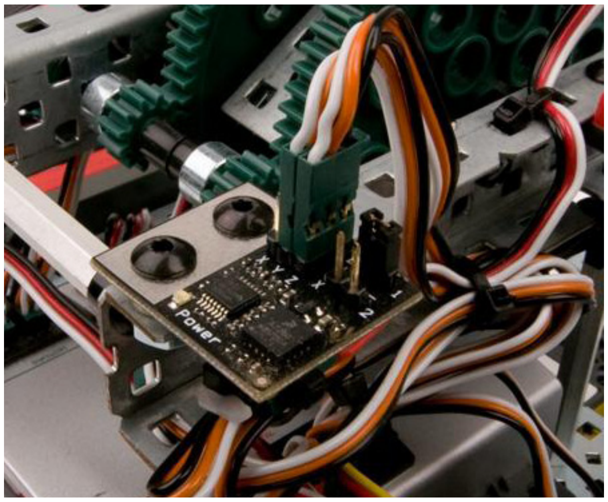
Accelerometer



Accelerometer
The VEX Accelerometer comes without plastic casing, which keeps the size of the sensor much smaller.

The VEX Accelerometer is an Analog Sensor that is able to measure acceleration on three axes (up/down, left/right, and forward/backward) at the same time. This data could allow the robot to calculate its velocity or react to collisions that do not hit a touch sensor. For convenience, there is an indicator light on the sensor itself that will light up solid green when the sensor detects that it is wired properly. Additionaly, the Accelerometer has connections for mounting directly to other VEX construction parts, making it easy to integrate on almost any robot.



Accelerometer Mounted on a Robot

There are two sensitivity modes on the Accelerometer: a +2g mode when the jumper is not attached, and a +6g mode when the jumper is attached. You may find that, while the more sensitive mode gives more accurate readings, the less sensitive mode will be more reliable for measurements that change quickly.

While the Accelerometer has the ability to take measurments in three axes of space, it is important to note that each axis needs its own Analog Port, meaning that you could potentially use 3 Analog Ports per sensor.

Accelerometer Natural Language Sample Code

Wait for Acceleration
This code has the robot wait until the robot receives a slight push forward, and then move forward for one second. The robot uses the Accelerometer to sense when it receives the forward push. The program also accounts for the bias in the sensor. It does so by treating its initial acceleration value as a reference point, and comparing its readings to that reference point.

The sample code uses the x-axis as the forward direction, but you may need to use a different axis depending on how your sensor is mounted. The sensor "xAccel" is the Accelerometer's x-axis readings

```
task main ()
{
 robotType(recbot);  //Specifies the robot type

 wait(2); //wait before running, allows sensor to initalize

 //Creates a variable to store the updated value of the sensor
 int currentX;

 //Creates a variable to store the initial sensor reading,
 // used a the "reference point" for future readings
 int xBias = abs(SensorValue[xAcecel]);

 //using a do-while loop has the calculation run once before
 //checking values
 do
 {
   //Takes a reading, subtracts out the inital bias
   currentX = abs(SensorValue[xAccel]) - xBias;

   //Take absolute value of reading, insures math stays correct
   currentX = abs(currentX);

   wait(0.01); //small wait statement, helps eliminate
               // irregular readings
 }

 //Loop while the current Accelerometer value is either zero or
 // very small
 while(currentX < 3);

 //Move the robot forward for one second, then stop.
 forward(63);
 wait(1);
 stop();
}
```