1 Introduction

There are endless sensors available or easily modifyable for the HandyBoard. The sections below concentrate on sensors in ESG's Robotics lab.

HandyBoards recognize two classes of sensors– digital and analog– plus a few special sensors. Digital sensors are either true or false. Because of the way switches are normally wired into the HandyBoard, "false" equates to a high voltage read on the sensor's signal pin. Analog sensors return a value between 0 and 255. Analog sensors can be read with the digital inputs, and any value above 128 will be treated as true.

Section 5.1 has several diagram for how to prepare sensors for use with the HandyBoard. Section 5.3 describes how the HandyBoard interacts with sensors in general, and switches, photocells, and infrared reflectors in particular.

2 Touch Sensors

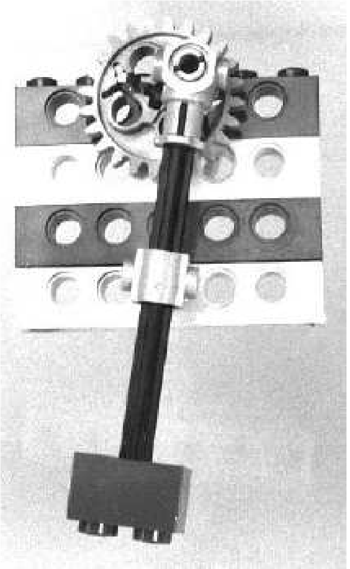
Digital touch sensors are also known as switches. They are most often used to detect collisions and for giving robots user interfaces. Any kind of switch can be used. If your switches doesn't have a lego backing or a very long action arm, you can hot glue one on.

We also have "Bend" sensors, which are strips of material which have different resistence spending on how curved they are. Generally, the full range of values they produce on the HandyBoard is only a small part of the range from 0 to 255, so callibrate accordingly.

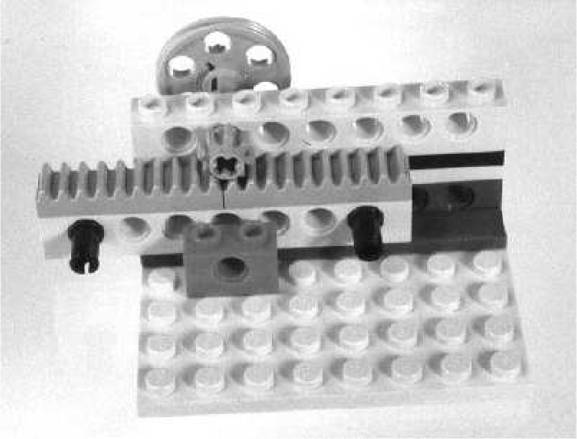
3 Turn Sensors

Turn sensors are potentiometers. They can be used to detect the internal configuration of the robot (e.g., how far a wheel is turned)

Although almost all turn sensors have a limited angle through which they can turn, they can be rigged to detect continuous turning. For example, a turn sensor can be attached to the back of the lego leg as below.



There are also linear potentiometers, which have a dongle that can slide within a long, narrow box. These are particularly useful when you have geear racks or other linear motion in your robot.



Be sure to check the range of your sensor before using it.

4 Light Sensors

Light sensors of photocells. They can allow a robot to follow a line, detect an obstacle at a distance, or become a photovore.

There are two kinds of special IR detector-reciever modules (reflective optosensors). If both detector and reciever are facing the same direction, it's a distance sensor. If they're facing eachother with a gap, it's an interruption sensor. Interruption sensors may be used to calculate the speed of a spinning motor, by placing a lego wheel with the holes aligned between the sensors.

5 Special Sensors

Microphones and cameras can also be used with the HandyBoard, but often require too much computational power. For a microphone, consider building a small circuit to estimate the energy of the noise the microphone is detecting, rather than collecting individual samples. For a camera, expect to only be able to process small parts of the image, or only be able to take images every so often. That said, I have heard of an assembly code speech recognition library built for the HandyBoard.

There are "orientation sensors" which can tell when the robot is on an incline or upside-down. The simplest kind consists of a conductive ball in s small box, with wires coming out of the various sides, which are connected when that side of the box is "down".

We also have heat sensors, and Hall effect sensors are easy to find.

6 Programming Sensors

The basic sensor functions are `digital(int n)` and `analog(int n)`. These return a boolean or a value between 0 and 255, respectively.

The HandyBoard also has an interrupt system. This allows you to register functions with the HandyBoard's core system, which are called (and the current program halted) when certain actions happen. This is far more efficient than standard polling of the various sensors.

7 The Knob, The Start, and Stop Button

The HandyBoard comes with three built-in sensors, for a robot user interface: the knob, and the start and stop buttons. The knob is a turn sensor, callibrated to give a value between 0 and 255. Get it's current reading with the `knob()` function. The start and stop buttons return true (1) through the `start_button()` and `stop_button()` functions when the associated button is pressed. It can be useful to write loops like:

```
while (!stop_button())  do_something();
```

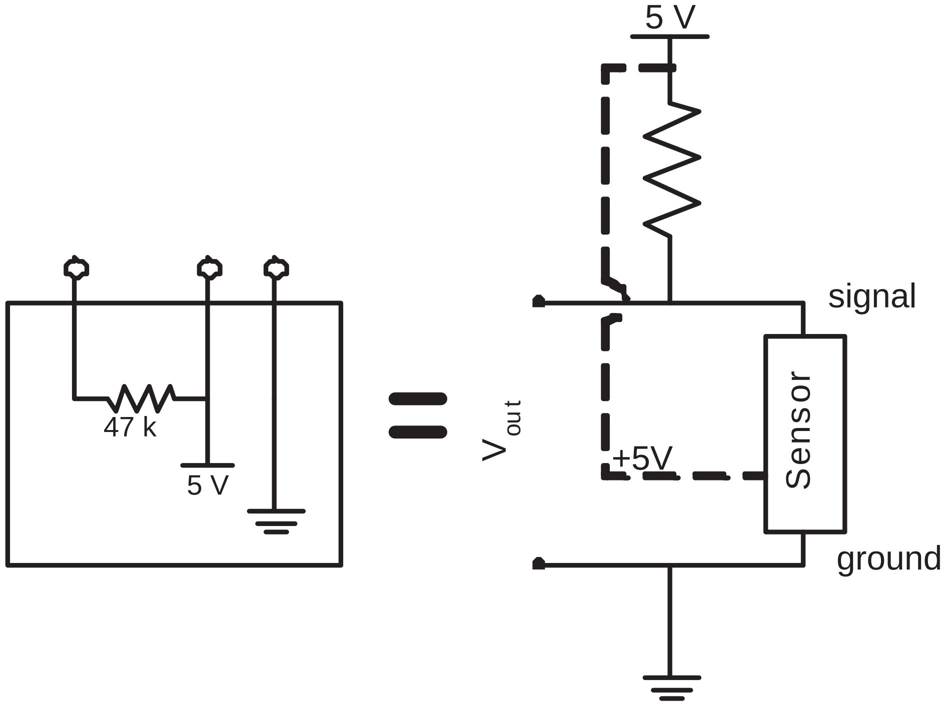
(but for this to work, the stop button must be held until it's registered.

8 The Infrared System

The HandyBoard comes with a built-in system for emitting and detecting infrared. While its possible to read the IR sensor directly, these are most useful with a set of assembly code available with the lego robotics software. This code allows you to set up IR beacons and emittors that output IR at a particular frequency (quickly alternating output it and not). The assembly code tracks noisy sequences of "reading" and "not reading" IR input, to determine if the underlying frequency of the input matches the one its looking for. This can be useful, for example, if there are two "goals areas", each marked by an IR beacon with a different frequency.

9 Adapting Sensors

The HandyBoard digital and analog input ports are effectively ohm meters. Every resistive sensor produces a voltage divider with a 47 kΩ resistor inside the HandyBoard. When the sensor has infinite resistance (open circuit), the signal wire will read a high voltage (5 V). When the sensor has zero resistance (closed circuit), it will read 0 V. The diagram below shows this.



The template program is useful for checking the output range of a sensor.

MIT OpenCourseWare
http://ocw.mit.edu

ES.293 Lego Robotics
Spring 2007

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.