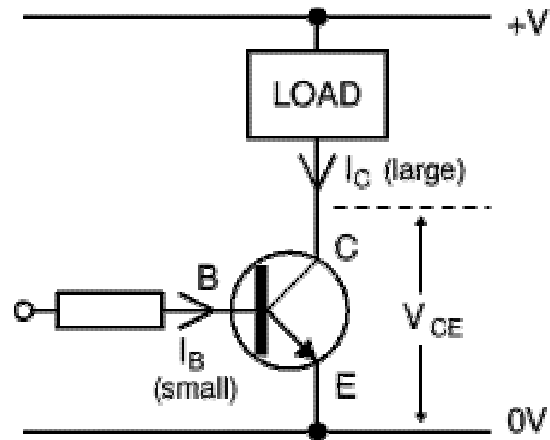


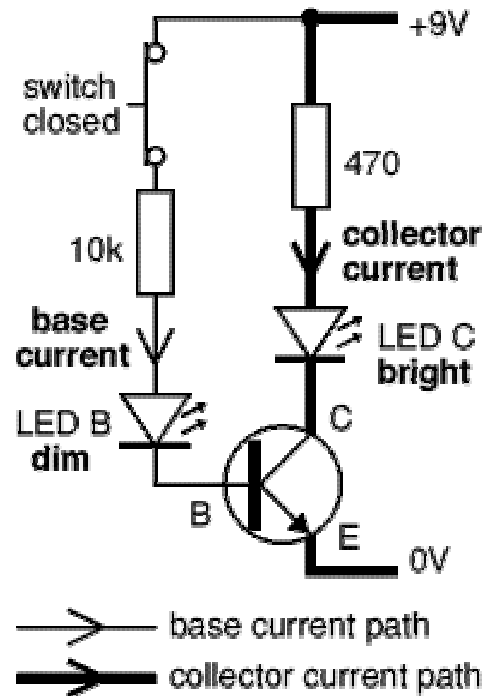
Transistor is a Current Amplifier



- Ratio of I_C to I_B is the transistor's "gain"
- Gain is sometimes abbreviated as h_{fe} or β
- β is typically around 100
- **Actual I_C may be limited by resistance of load**

$$\beta = \frac{I_C}{I_B} \approx 100$$

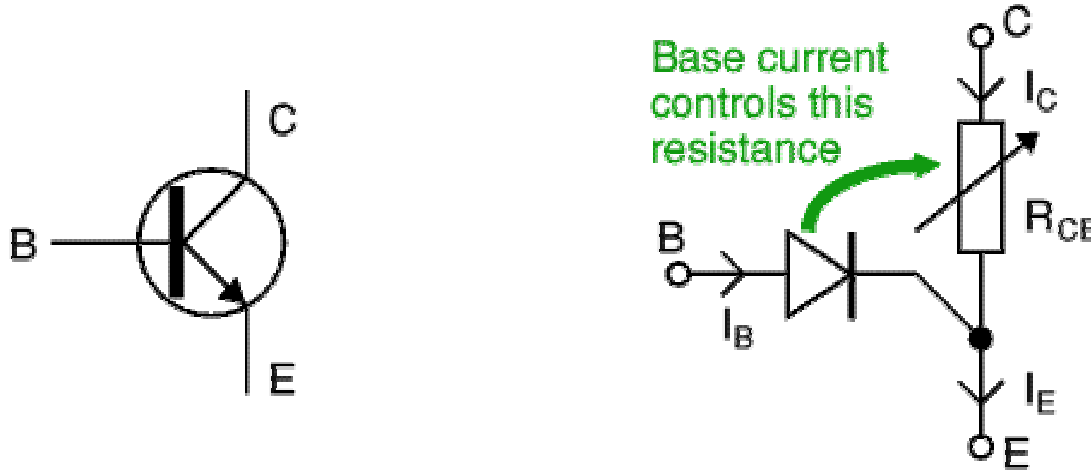
Transistors as Amplifiers



- Small base current controls large collector current
- Base current (in this circuit) $< 1\text{mA}$
- Collector current could be $> 100\text{mA}$

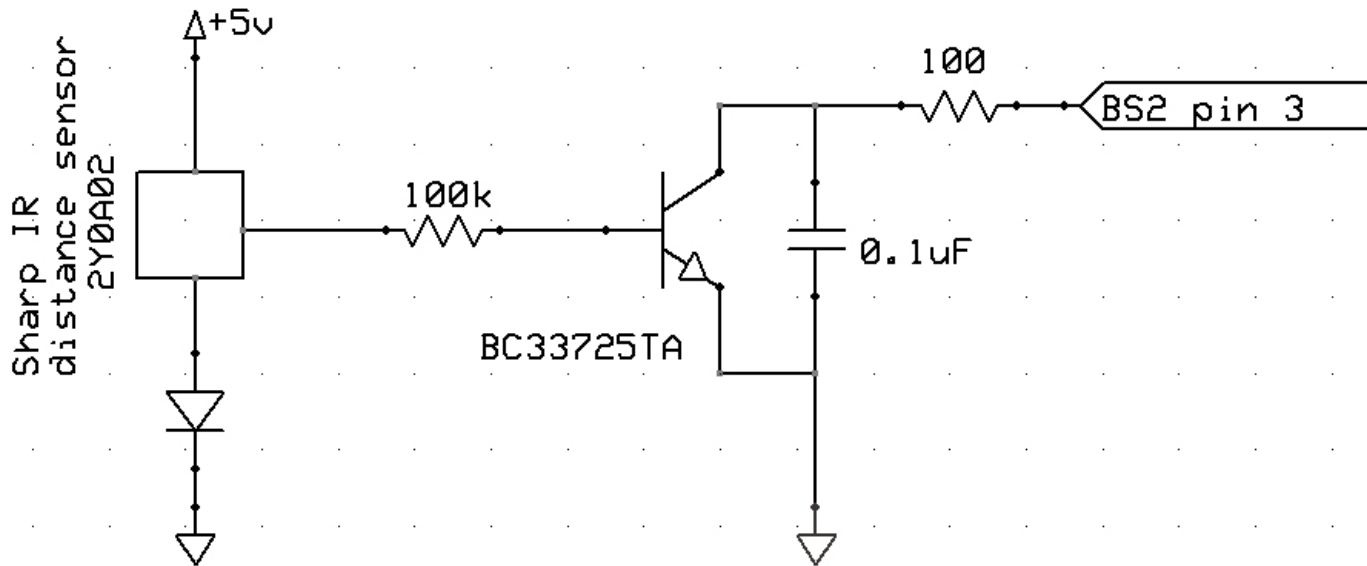
<http://www.kpsec.freeuk.com/trancirc.htm>

Transistors as Amplifiers



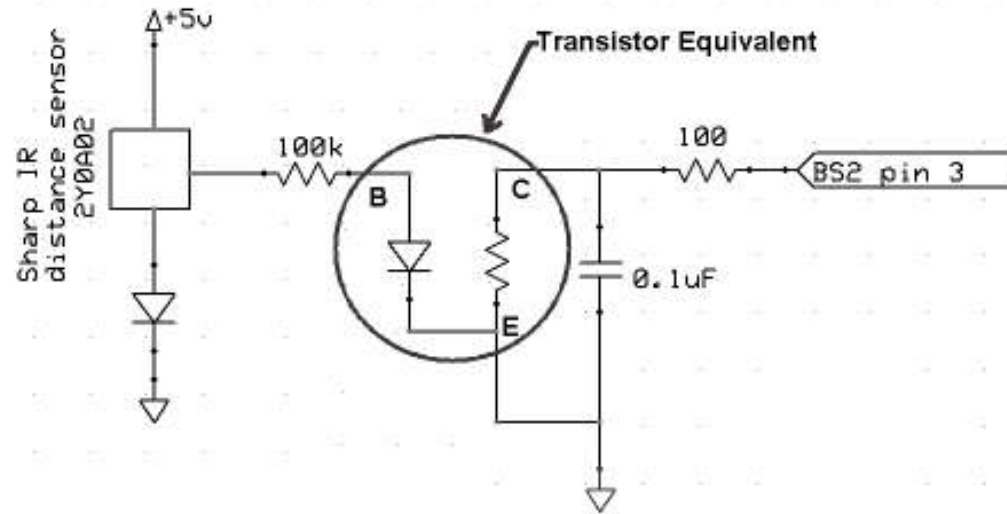
- Transistor is equivalent to a variable resistor controlled by the current through a diode
- A transistor contains an internal diode between Base & Emitter
- The voltage across this diode will always be about 0.6V when there is current flowing into the base (transistor on)
- The current through the diode controls the current through the CE resistor (R_{CE})

Interfacing an Analog Sensor to the BS2

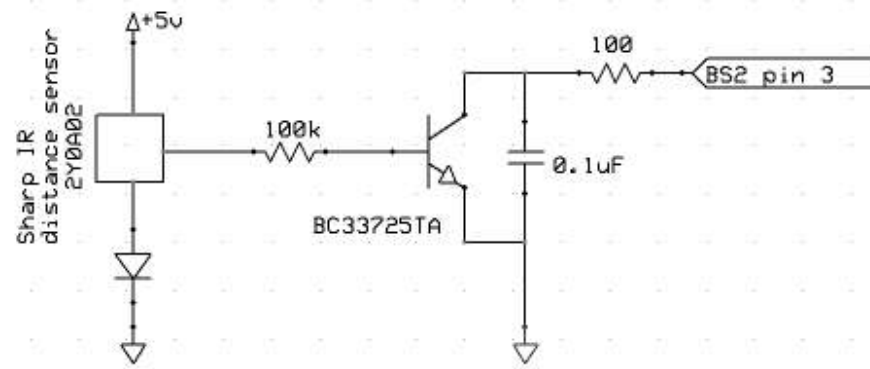


- Sensor output can be 0-5V
- C-E pins of transistor replace resistor
- Can be read with RCTIME

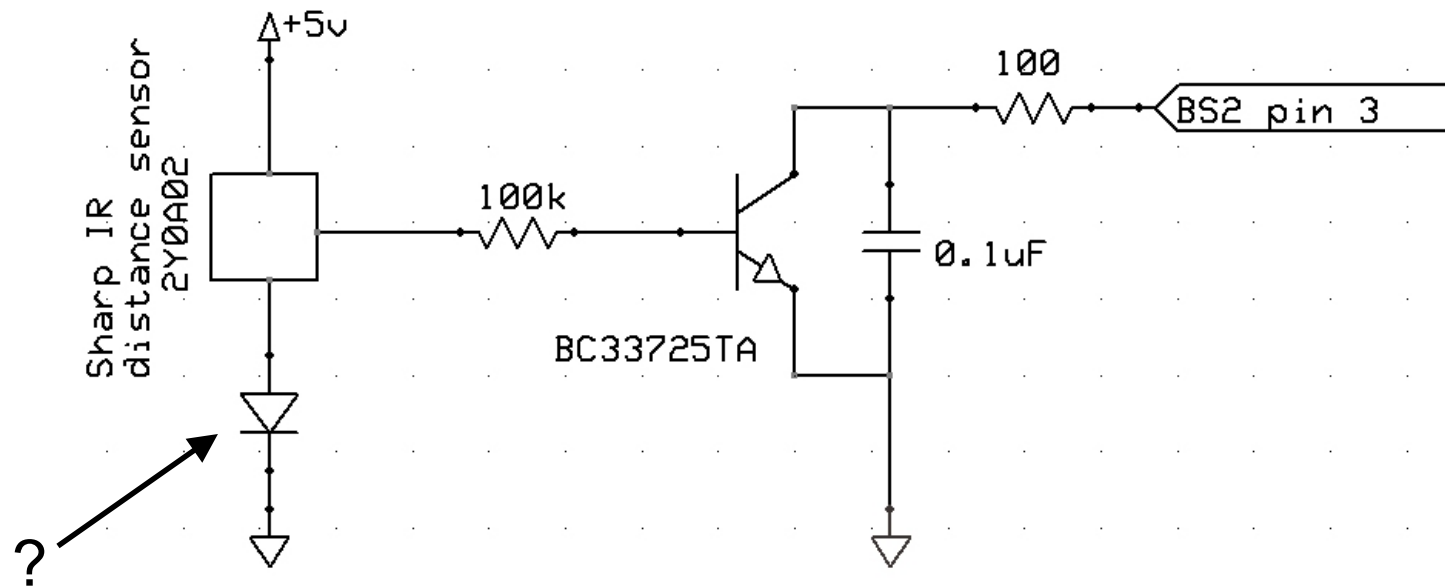
How does it work?



- Current through BE diode controls CE resistor
- I_{CE} will be about $\beta = 100$ times larger than I_{BE}



What's the diode doing?



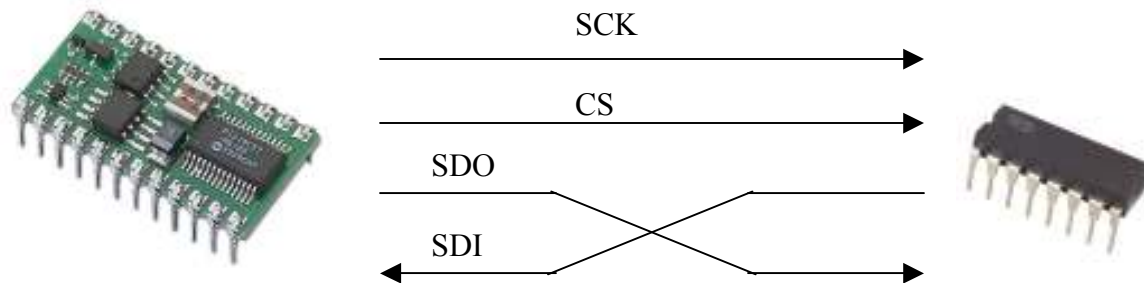
- Transistor contains internal diode (between B & E)
- Base must be above 0.6V for transistor to begin turning on
- Diode prevents sensor output from going below 0.6V

SPI Communications

- Yet another serial communications protocol!
- Similar to I2C
- Used primarily for inter-IC (chip to chip) communications
- Interface to:
 - Digital memory
 - A/D converters (receive analog info from a sensor)
 - D/A converters (generate an analog voltage)
 - Sensors (Accelerometers, temperature, ...)

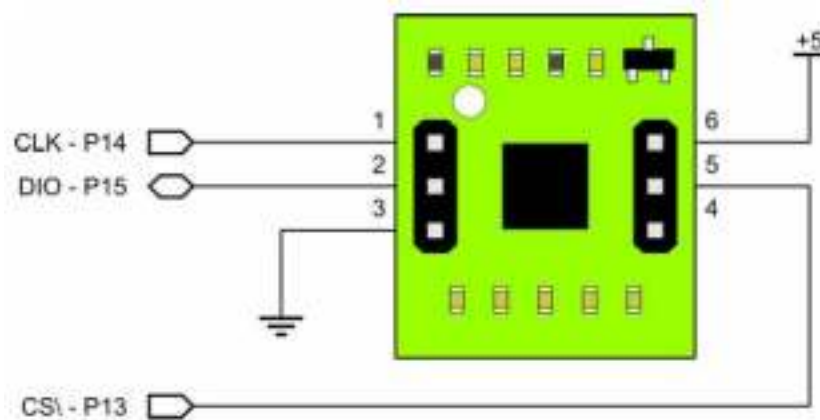
SPI Communications

- The SPI bus specifies four logic signals.
 - SCLK — Serial Clock (output from master)
 - MOSI/SIMO — Master Output, Slave Input (output from master)
 - MISO/SOMI — Master Input, Slave Output (output from slave)
 - SS — Slave Select (active low; output from master)
- Alternative naming conventions are also widely used:
 - SCK — Serial Clock (output from master)
 - SDI, DI, SI — Serial Data In
 - SDO, DO, SO — Serial Data Out
 - nCS, CS, nSS, STE — Chip Select, Slave Transmit Enable (active low)
- The SDI/SDO (DI/DO, SI/SO) convention requires that SDO on the master be connected to SDI on the slave, and vice-versa.



SPI Communications

- Simplified implementations are common (“Three Wire Interface”)
- Sometimes data only needs to go in one direction
- Some chips can share one data line (DIO)



H48C accelerometer

BASIC Stamp SPI

- Uses SHIFTIN and SHIFTOUT commands
- SHIFTOUT sends a number as a serial stream of bits
- SHIFTIN receives a number as a serial stream of bits
- Commands automatically coordinate clock and data signals

```
'H48C accelerometer subroutine
```

```
Dio PIN 15 ' data to/from module
```

```
Clk PIN 14 ' clock output
```

```
CS PIN 13 ' active-low chip select
```

```
Get_H48C:
```

```
  LOW CS 'activate accelerometer
```

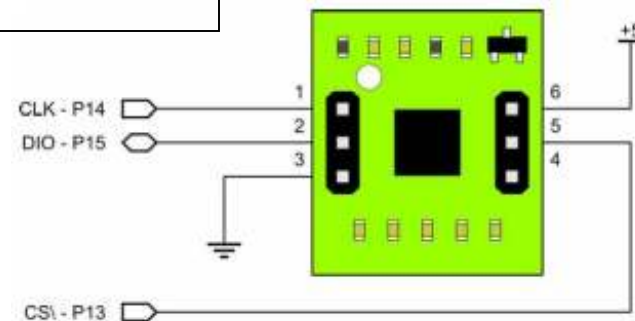
```
  SHIFTOUT Dio, Clk, MSBFIRST, [%11\2, axis\3] ' select axis
```

```
  SHIFTIN Dio, Clk, MSBPOST, [axCount\13] ' read axis voltage counts
```

```
  HIGH CS
```

```
RETURN
```

```
END
```



SPI Accelerometer Code

- 1) Set axis
- 2) GOSUB Get_H48C
- 3) Acceleration along the axis comes back in axCount variable

```
H48C Accelerometer demonstration loop
```

```
DO  
  axis = 0  
  GOSUB Get_H48C  
  x=axCount  
  axis = 1  
  GOSUB Get_H48C  
  y=axCount  
  axis = 2  
  GOSUB Get_H48C  
  z=axCount  
  
  DEBUG DEC x," ",DEC y," ",DEC z, CR  
LOOP
```

H48C Accelerometer

- can determine 3D static orientation
- can detect movement in 3D

What is MaxMSP/Jitter

- Max – virtual environment for routing and processing MIDI signals and for making signal and sensor interconnections
- MSP – virtual environment for implementing real time audio (and some video) processing
- Jitter – advanced video extension package

What is MaxMSP/Jitter

- “Graphic” or “Visual” programming language
- Programs written by making wire-like connections between “objects”
- Programs run as a network of “objects” sending “messages” to other objects

Intro to MaxMSP

- Object boxes
- Inputs on top, outputs on bottom
- Online help is available for all objects
- Print object
- BANG signal
- Pushbutton and Toggle objects
- Serial object
- Noteout object