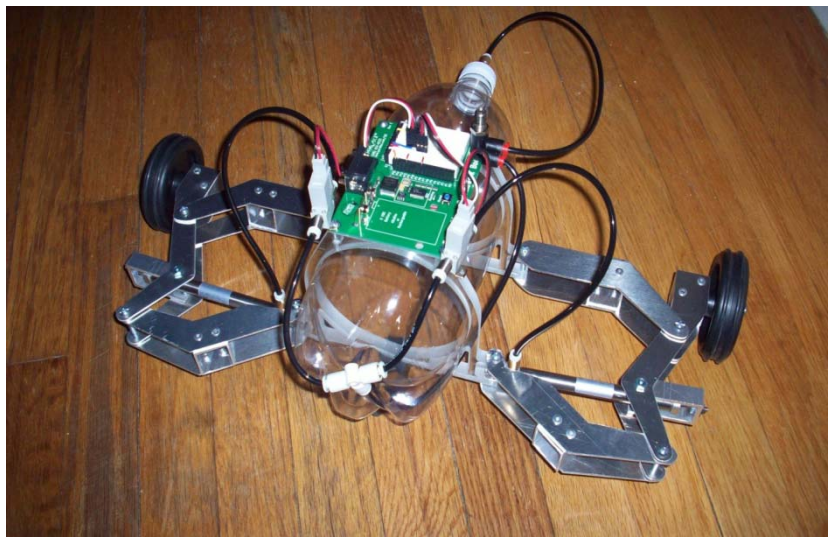
2.007 Design and Manufacturing I
Spring 2009

2.007 –Design and Manufacturing I

Microcomputers, Programming, Electronics, and Sensors

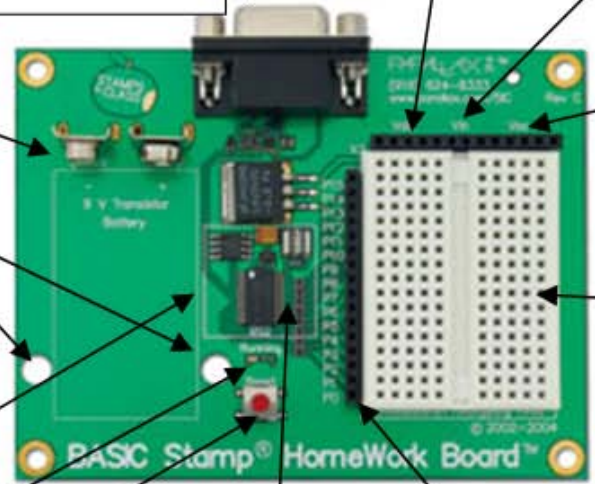


Images removed due to copyright restrictions. Please see
http://media.digikey.com/photos/Honeywell%20Photos/BZ-2RW82.jpg
http://media.digikey.com/photos/Parallax%20Photos/MFG_30056.jpg
http://www2.gpmd.com/imagem/f/mfutl0832.jpg

Dan Frey

31 MAR 2009

The Homework Board



Attachment for a serial cable for communicating with a computer (download programs from computer to BS2 and send DEBUG data from BS2 to your screen).

Vdd – a 5V regulated supply

Vin – a connection to the "+" battery terminal at 9V unregulated supply

Vss – a connection to the "-" battery terminal which serves as ground

Attachment place for a 9V "transistor battery"

Tie down holes for battery

BASIC Stamp 2 (BS2) PIC, EEPROM, and voltage regulator

Breadboard (solderless prototyping board) with 17 rows and two columns each with 5 electrically interconected "clips." Connections are horizontal rows separated by the trough.

Indicator light "ON" when a program is running

Button to reset your program

16 resistors at 220Ω attached to each I/O pin

16 headers to which you can connect to I/O pins through the built in 220Ω resistors

Each pin sources at most 20 *milli* Amps

What happens?

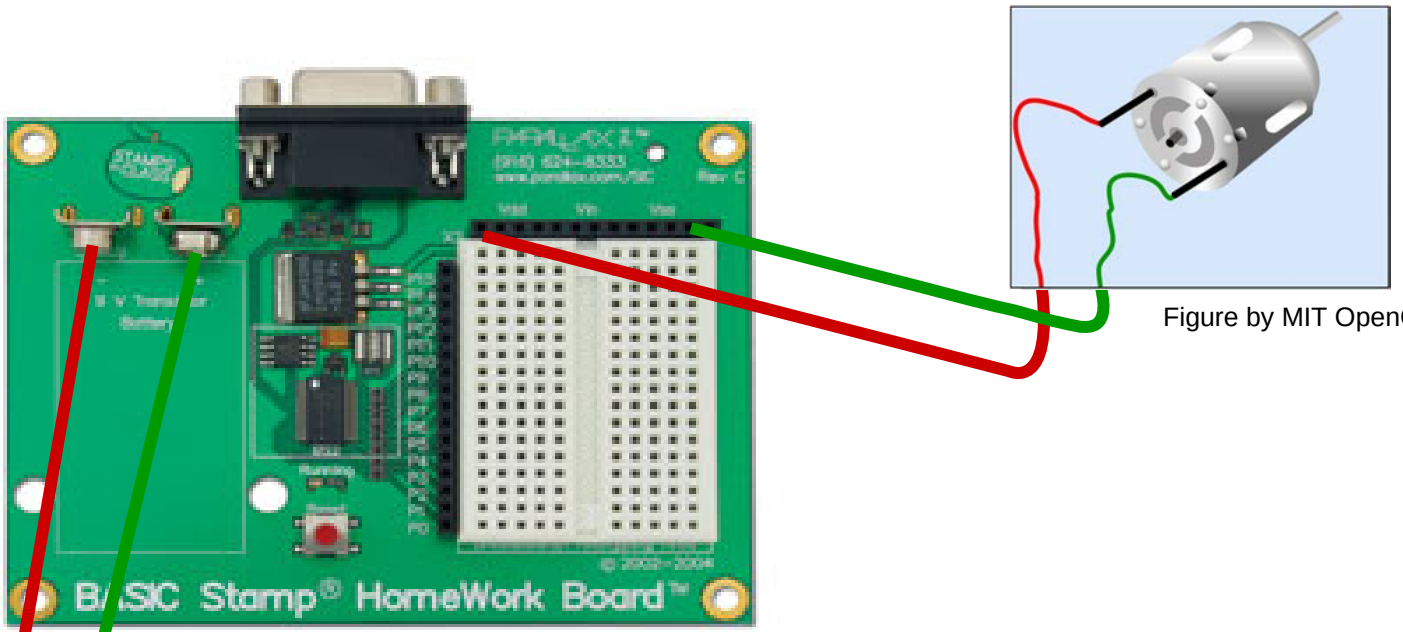


Figure by MIT OpenCourseWare.

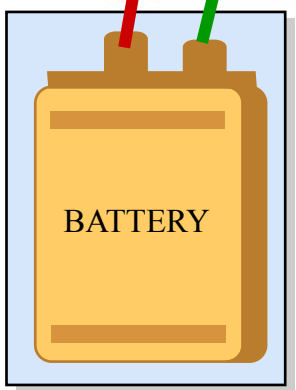
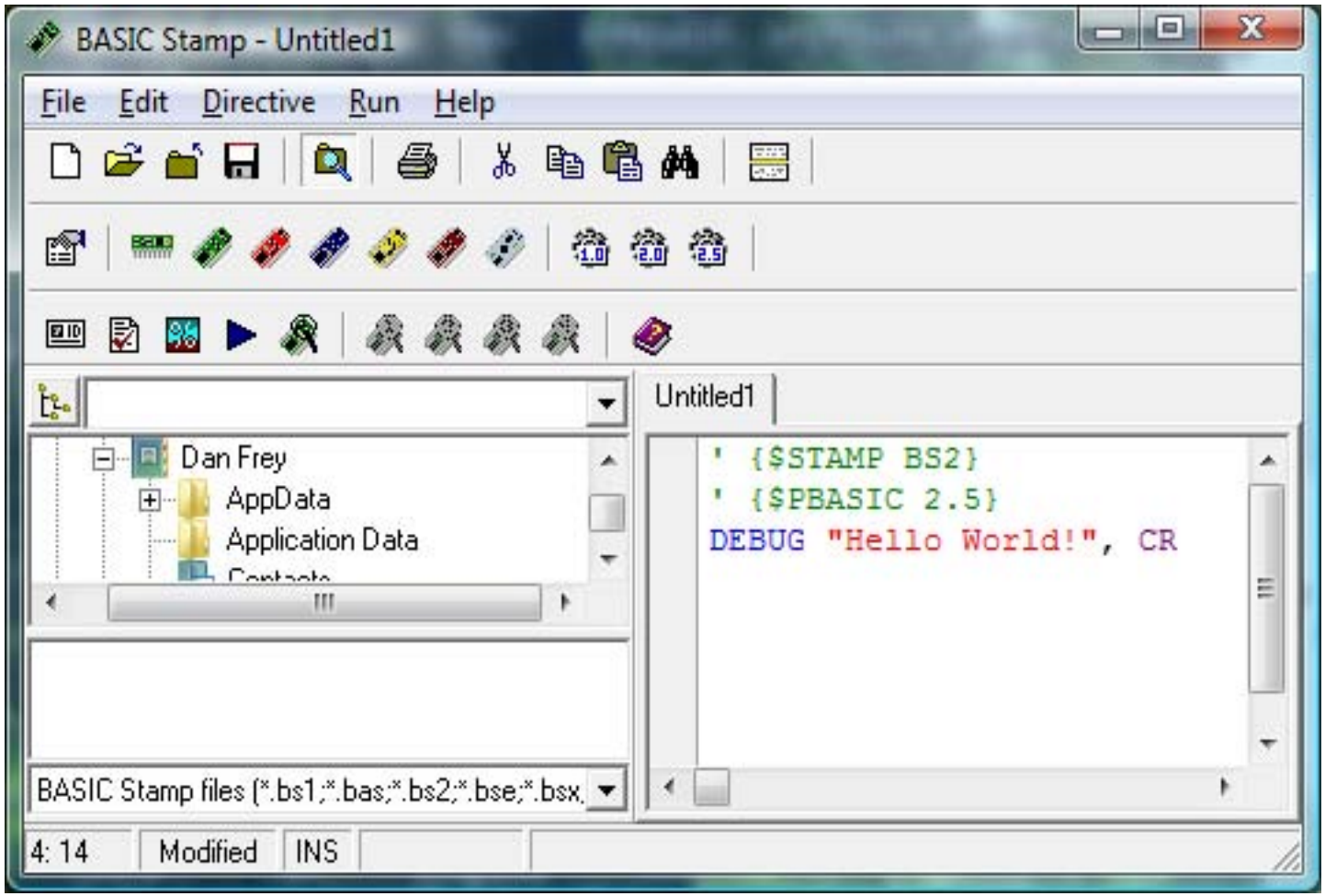


Figure by MIT OpenCourseWare.

The Basic Stamp Editor

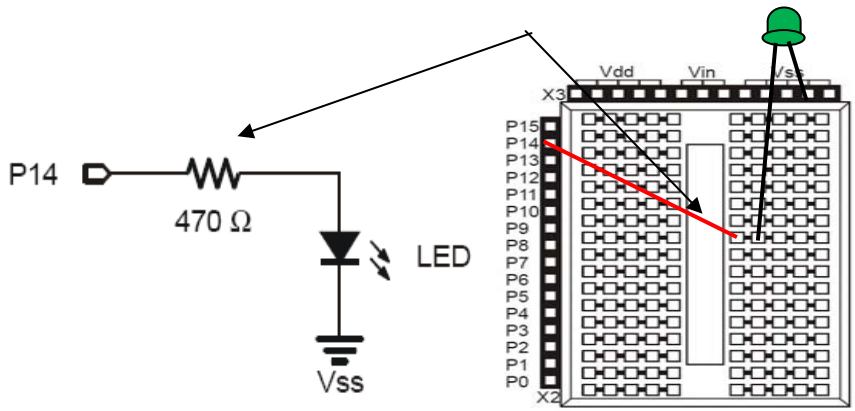


PBASIC Programming Language

- *name* VAR *size* (BIT, NIB, BYTE, WORD)
- IF … THEN
- FOR … NEXT
- GOTO *label* (define *label* like -- Loop:)
- PULSOUT *pin, period* (2μsec per unit)
- PAUSE *period* (1millisec per unit)
- DEBUG *OutputData* (to your PC screen)

Make an LED Flash

Just a jumper wire is needed because a 220Ω resistor is built into the pins of the Homework board




```
DO
HIGH 14
PAUSE 500
LOW 14
PAUSE 500
LOOP
```

The unit of the PAUSE command is *milli*seconds, so this line will result in a ½ sec pause.

What happens?

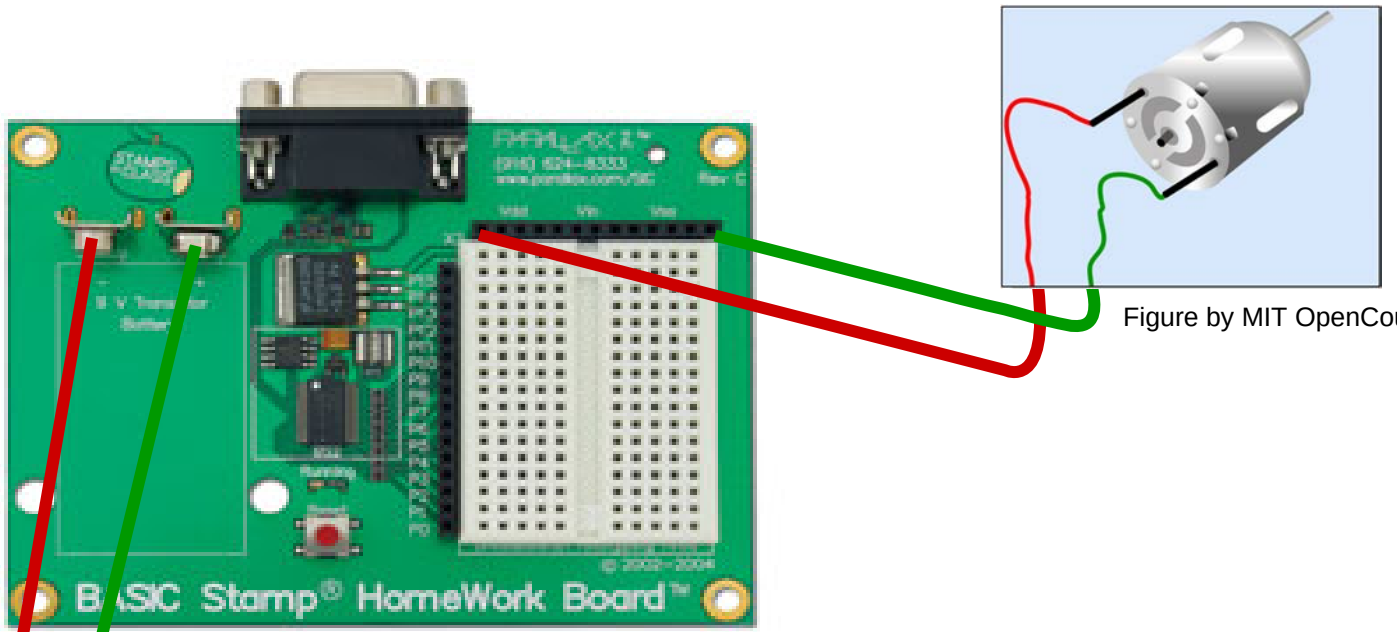


Figure by MIT OpenCourseWare.

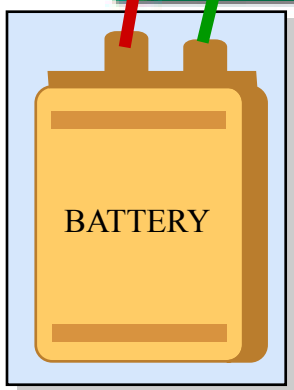


Figure by MIT OpenCourseWare.

What happens?

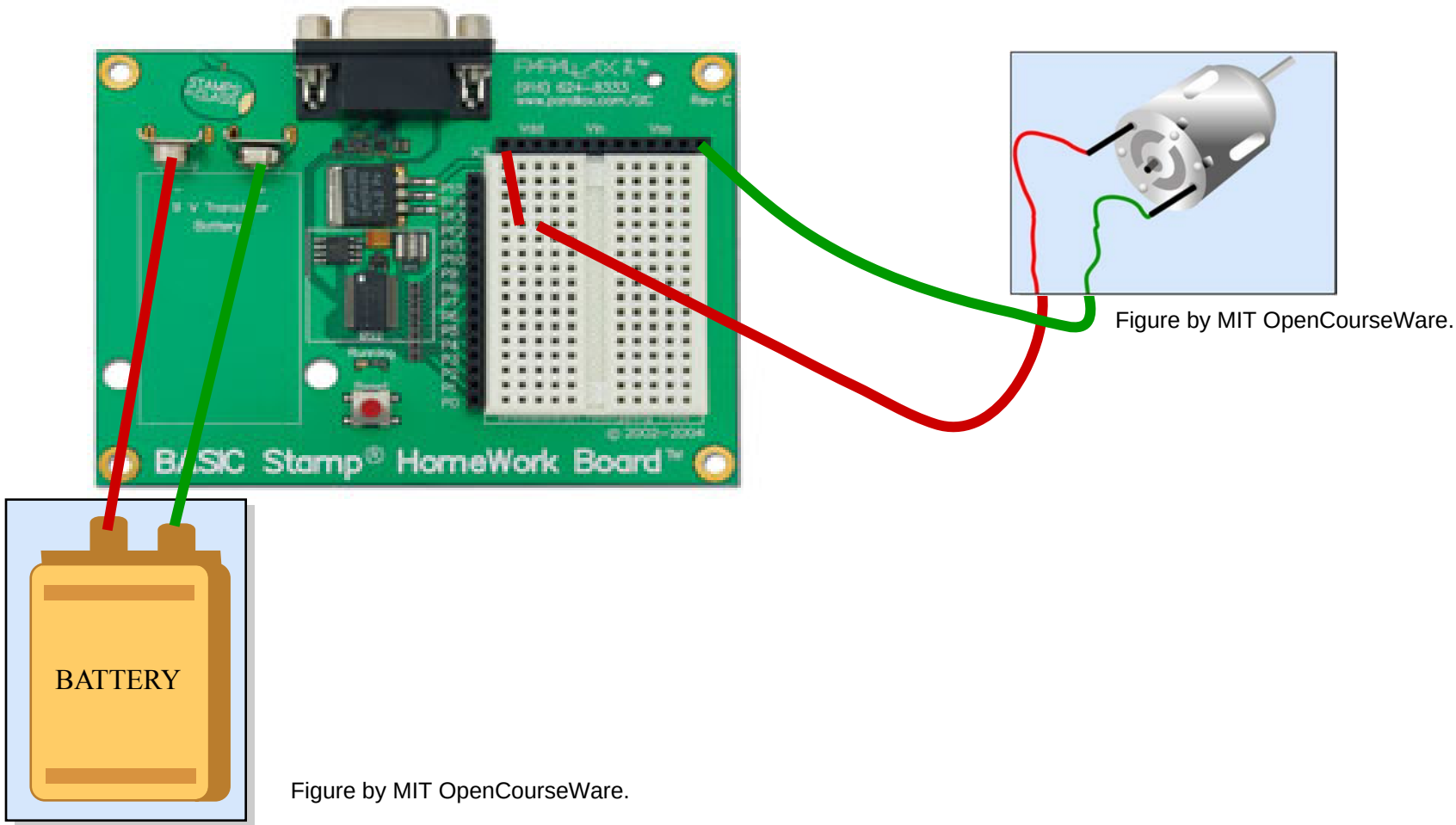


Figure by MIT OpenCourseWare.

Figure by MIT OpenCourseWare.

What happens?

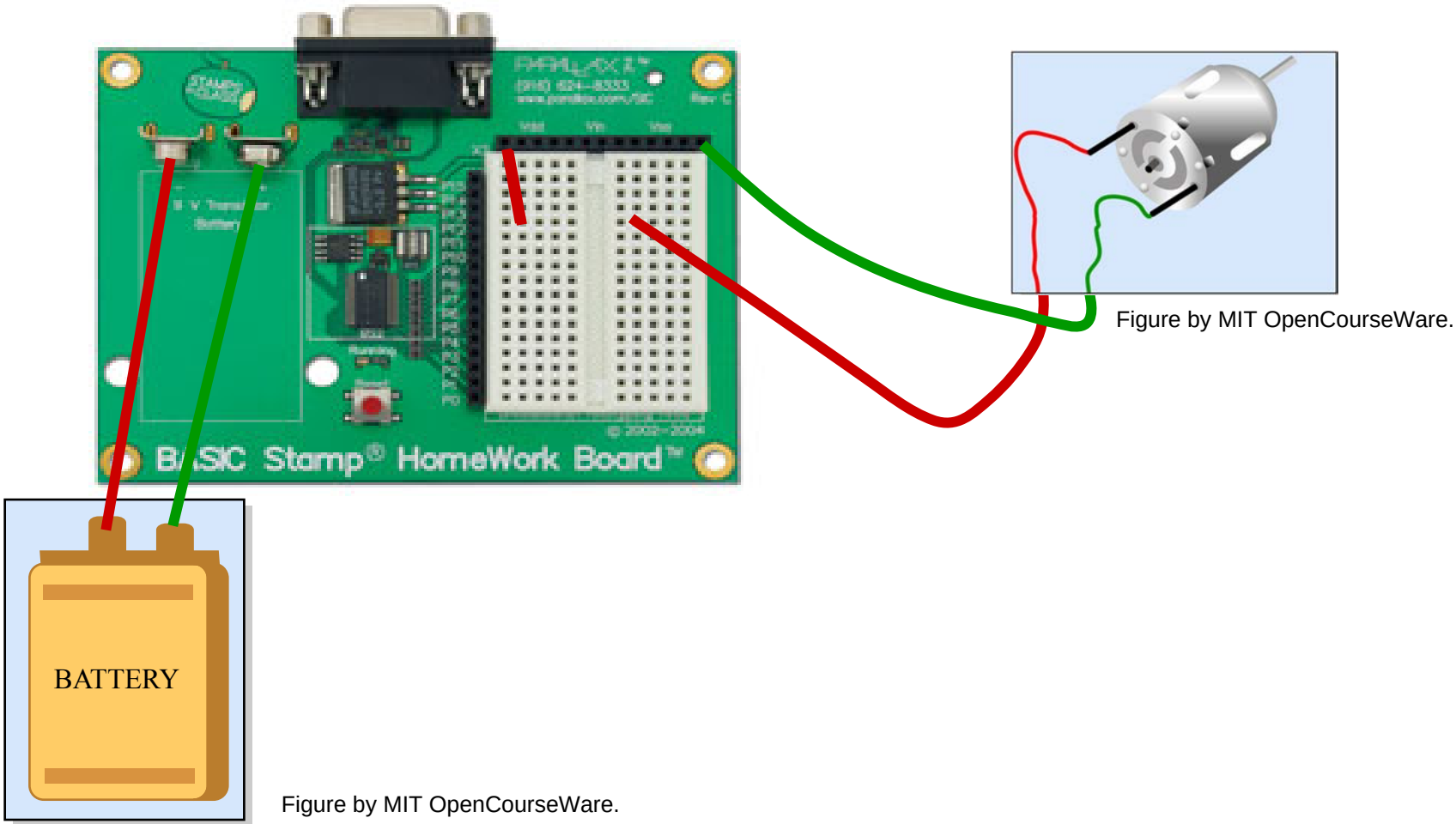
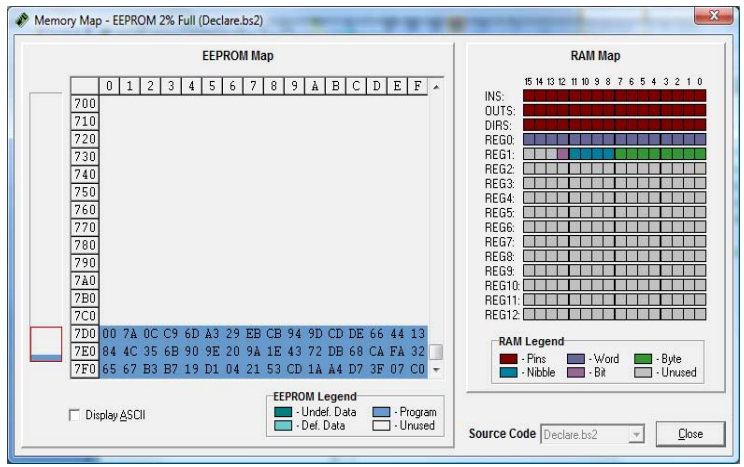


Figure by MIT OpenCourseWare.

Figure by MIT OpenCourseWare.

Memory and Variable types

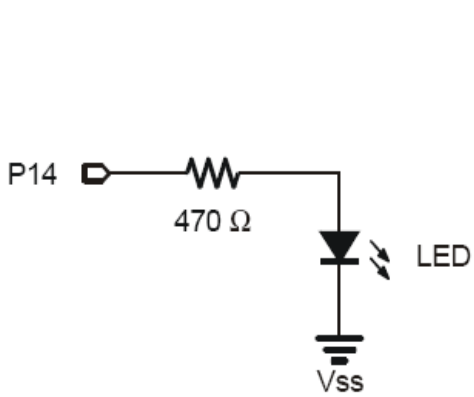
```
Mouse VAR BIT    ' Mouse is a variable that takes values 0 or 1
Cat VAR NIB      ' Cat is a variable that uses four bits
'NOTE:  The term "NIB" is short for a "Nibble" which is a small Byte
Dog VAR BYTE     ' Dog is a variable that uses eight bits
Horse VAR Word   ' Horse is a variable that that uses 16 bits
Dog = 250  ' Assign a value to the byte sized variable
DEBUG ? Dog  ' Display the result to the screen
Dog = 260  ' Try to assign a value larger than the byte data type can hold
DEBUG ? Dog ' Display the result to the screen
```

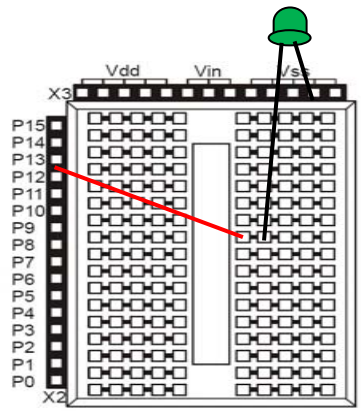


Text removed due to copyright restrictions. Please see
http://en.wikibooks.org/wiki/PBASIC_Programming/Loops#FOR_.2F_NEXT

Making an LED Blink Increasingly Faster

```
Delay VAR Nib
FOR Delay= 1 TO 15
HIGH 14
PAUSE 20-(Delay*100)
LOW 14
PAUSE 20-(Delay*100)
NEXT
```





What Happens?

```
Delay VAR Nib
FOR Delay= 1 TO 15
HIGH 14
PAUSE 20-(Delay*100)
LOW 14
PAUSE 20-(Delay*100)
NEXT
```

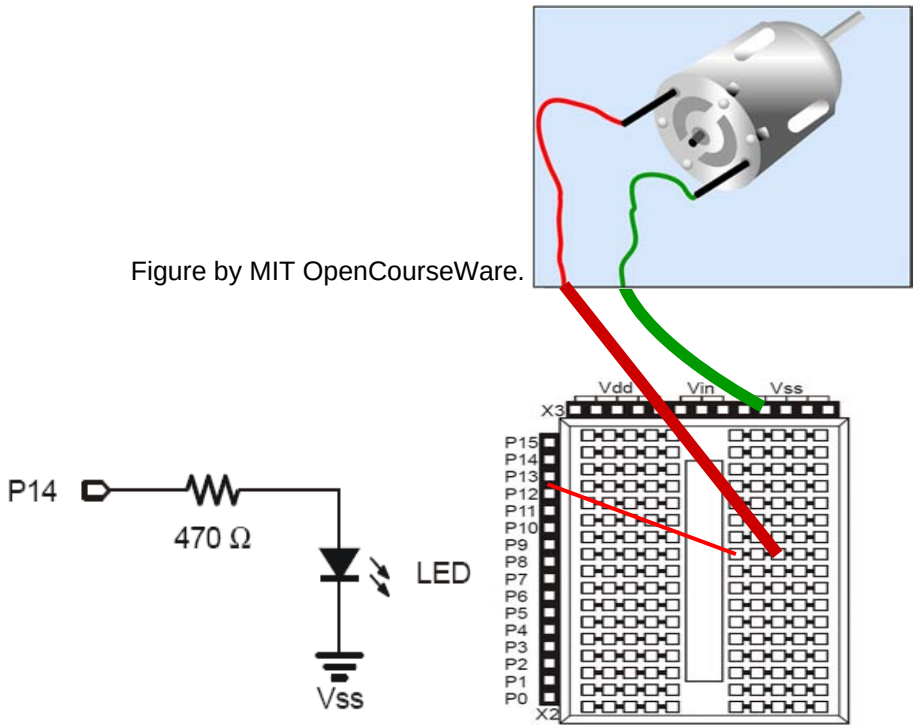
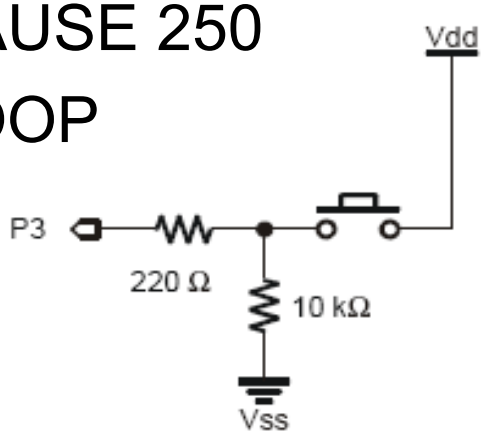


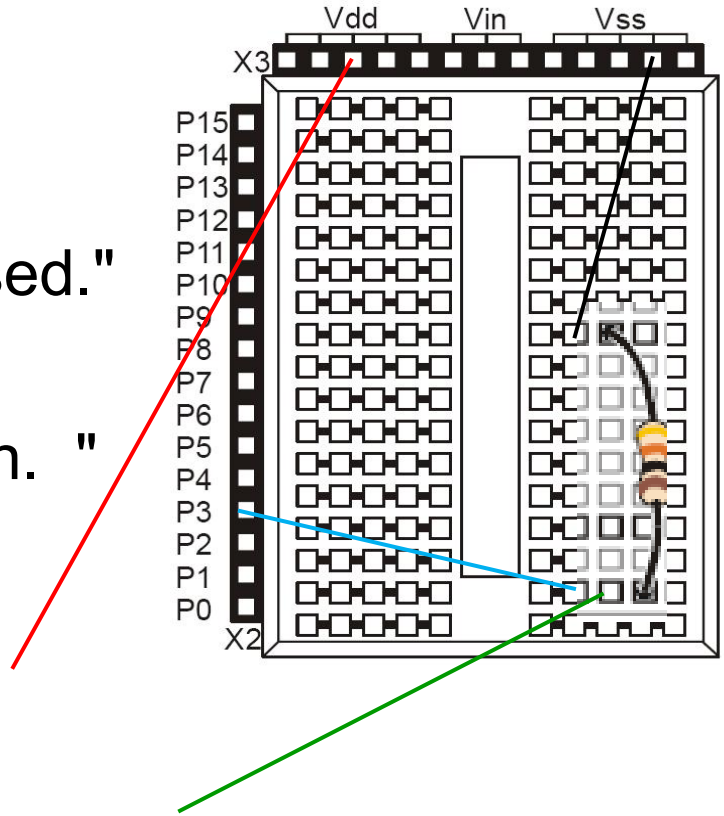
Figure by MIT OpenCourseWare.

Text removed due to copyright restrictions. Please see
http://en.wikibooks.org/wiki/PBASIC_Programming/Branches#IF_.2F_THEN_Branches

Checking the State of a Switch

```
DO
IF (IN3 = 1) THEN
  DEBUG HOME, "YES! Switch pressed."
ELSE
  DEBUG HOME, "NO! Switch is open.  "
ENDIF
PAUSE 250
LOOP
```

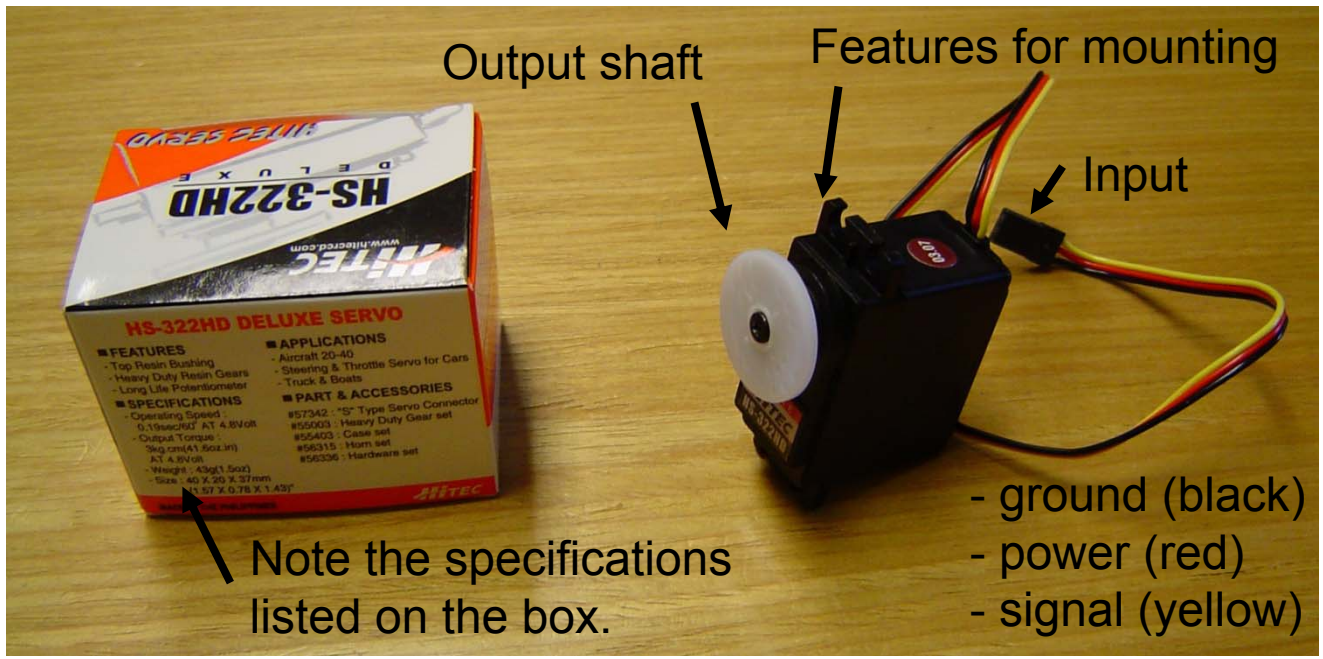




Images removed due to copyright restrictions. Please see
http://media.digikey.com/photos/Honeywell%20Photos/BZ-2RW82.jpg

Servo Motors

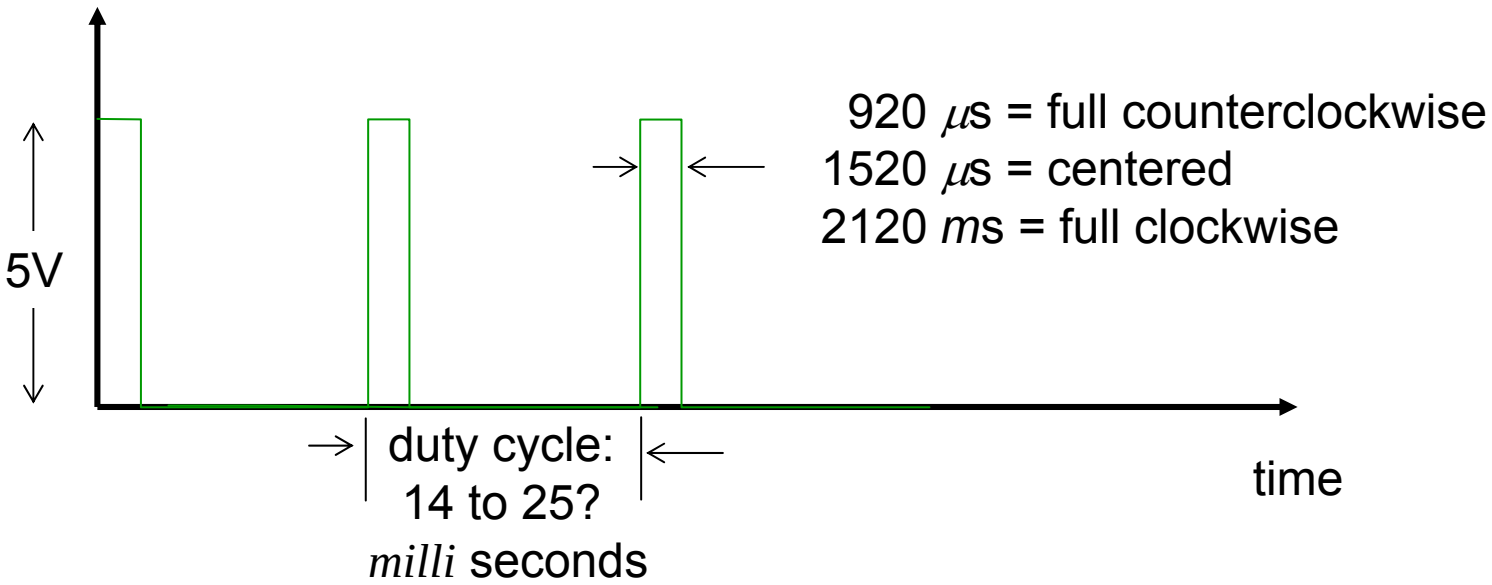
- Actuators that attain and hold a commanded position
- The type you have are commonly used in radio controlled cars and planes



Pulse Width Modulation (PWM)

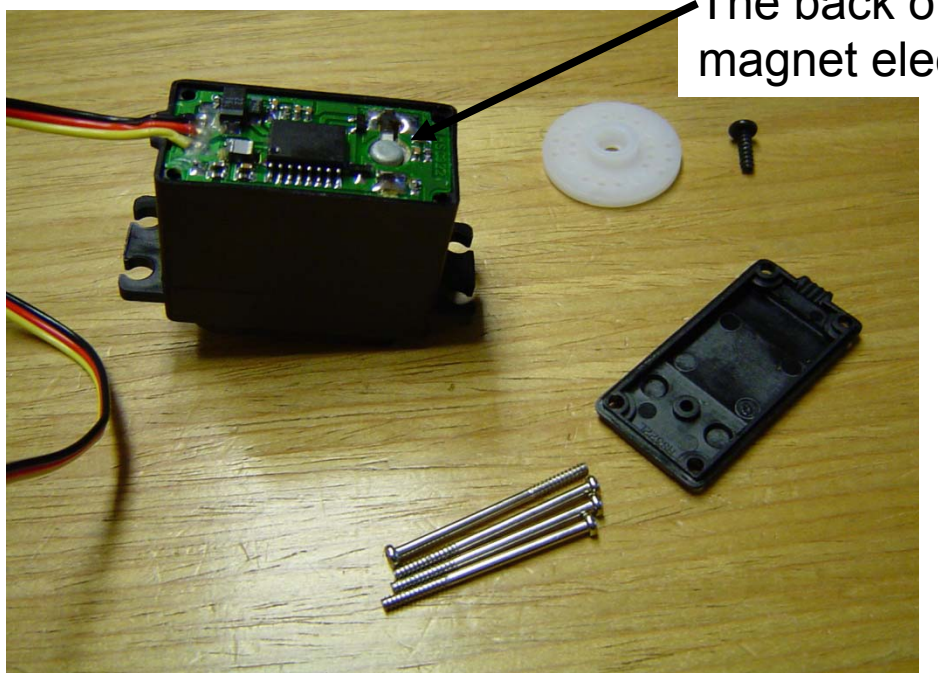
- The duration of the pulse is interpreted as a commanded position
- PULSOUT *pin, period* (2μsec per unit)
- PAUSE *period* (1millisec per unit)

Voltage on yellow wire



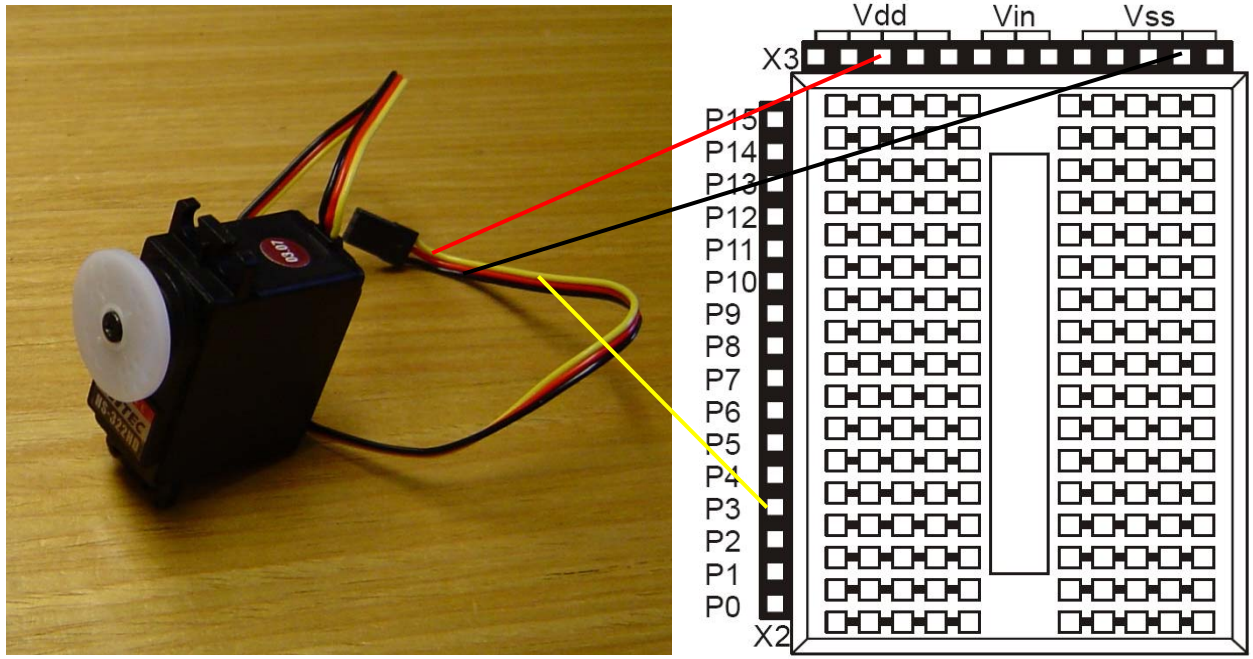
Electronics Within the Servo

- Receive the commanded position
- Sense the position of the output shaft
- Supply voltage to the motor (either polarity) depending on the error



Driving a Servo with the Stamp

```
DO
Reps VAR Byte
FOR Reps=1 TO 20
  PULSOUT 3, 750
  PAUSE 16
NEXT
FOR Reps=1 TO 20
  PULSOUT 3, 1100
  PAUSE 16
NEXT
LOOP
```



If I declare Reps as type Nib, what happens?

1) error message generated
2) program never leaves the first FOR loop
3) program leaves each FOR loop sooner
4) no difference

Radios

Image removed due to copyright restrictions. Please see
http://www.modelimport.com/marcas/futaba/Receptores/41007902%20R168DF.jpg

Throttle (ch3)

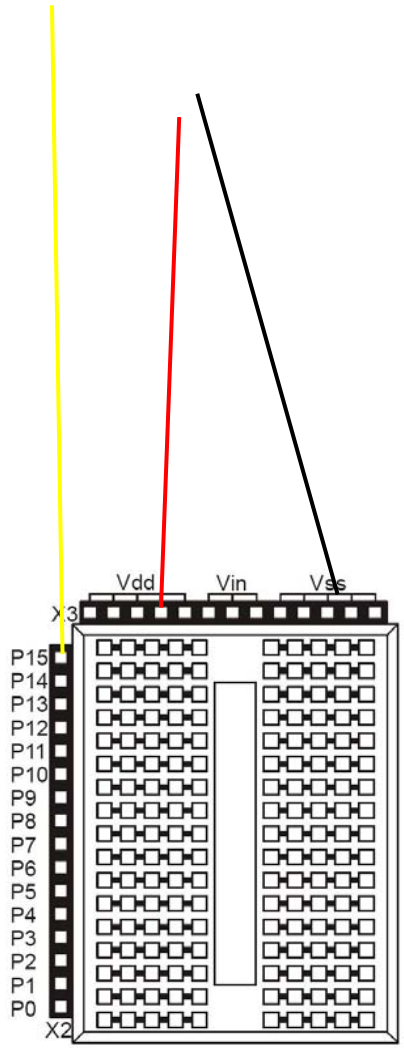
FM transmission

Please see http://www2.gpmd.com/image/t/towj41.jpg

Elevator (ch2)

Rudder (ch 4)

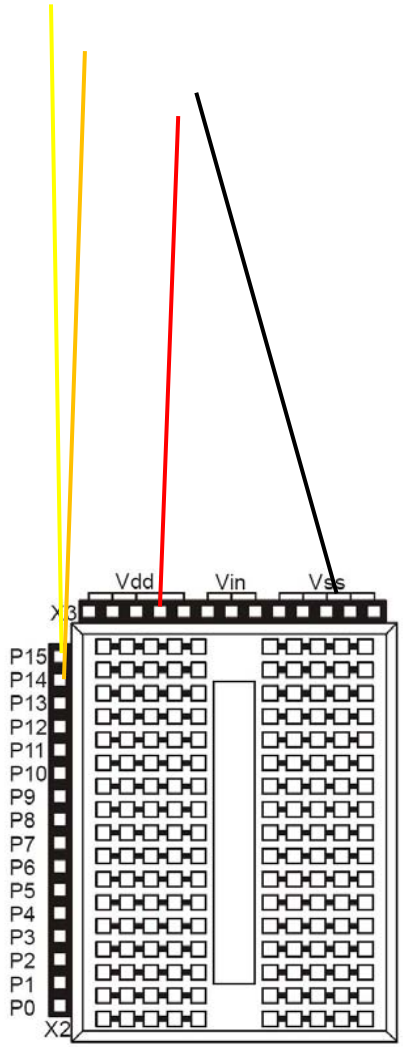
Aileron (ch 1)



Getting Signals into the Stamp

Image removed due to copyright restrictions. Please see
http://www.modelimport.com/marcas/futaba/Receptores/41007902%20R168DF.jpg

```
throttle VAR Word
rudder VAR Word
DO
PULSIN 15, 1,throttle
PULSIN 14, 1, rudder
DEBUG home, ? throttle
DEBUG ? rudder
PAUSE 200
LOOP
```



An Issue with Arithmetic

```
throttle VAR Word
rudder VAR Word
result VAR Word

DO
PULSIN 15, 1,throttle
PULSIN 14, 1, rudder
DEBUG home, ? throttle
DEBUG ? rudder
result=throttle-2*rudder
DEBUG ? result

PAUSE 200
LOOP
```

Get in the habit of using brackets to indicate desired order of operations

Another Issue with Arithmetic

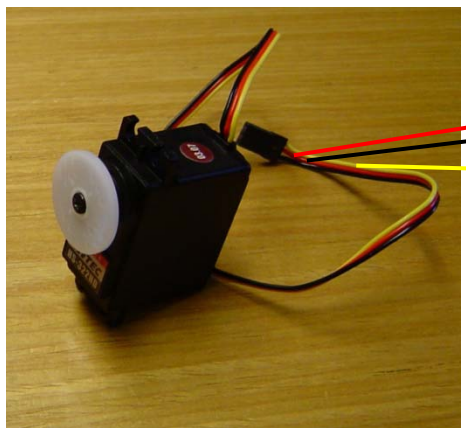
```
throttle VAR Word
rudder VAR Word
result VAR Word

DO
PULSIN 15, 1,throttle
PULSIN 14, 1, rudder
DEBUG home, ? throttle
DEBUG ? rudder
result=(throttle/rudder)*10
DEBUG ? result

PAUSE 200
LOOP
```

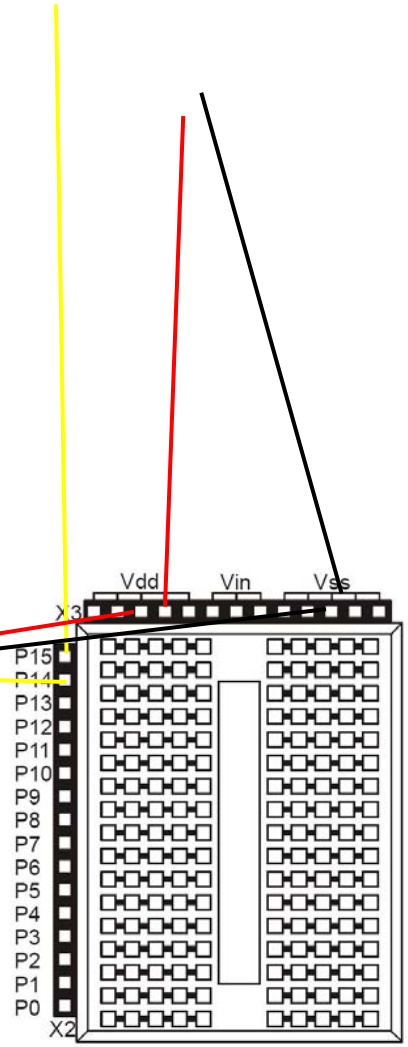
Intermediate results are stored in the same kind of variable as the final result. Watch out for underflow.

Expanding the Servo Range

Image removed due to copyright restrictions. Please see
http://www.modelimport.com/marcas/futaba/Receptores/41007902%20R168DF.jpg

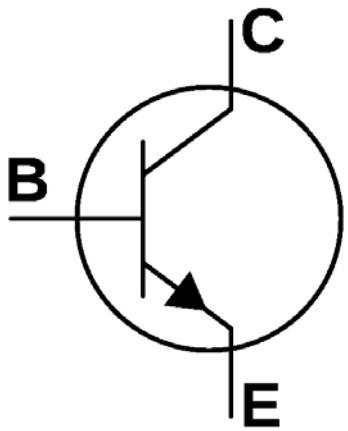
```
throttle VAR Word
response VAR Word
DO
PULSIN 15, 1,throttle
DEBUG home, ? throttle
IF (throttle>500)AND(throttle<1000) THEN
response=((throttle-750)*2)+750
ELSE
response=throttle
ENDIF
PULSOUT 14, response
PAUSE 10
LOOP
```





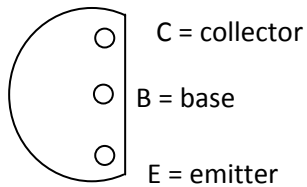
Switching On/Off a Load

Image from Wikimedia Common,
http://commons.wikimedia.org



The symbolic representation of the transistor

How the transistor (as packaged) looks literally



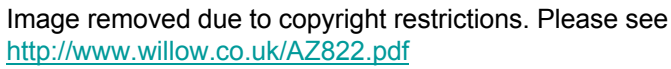


Image removed due to copyright restrictions. Please see
http://www.willow.co.uk/AZ822.pdf

H Bridge

- Reversible control of a load such as a DC motor


Image removed due to copyright restrictions. Please see
http://commons.wikimedia.org/wiki/File:H_bridge_operating.svg

Running a Motor with Relays

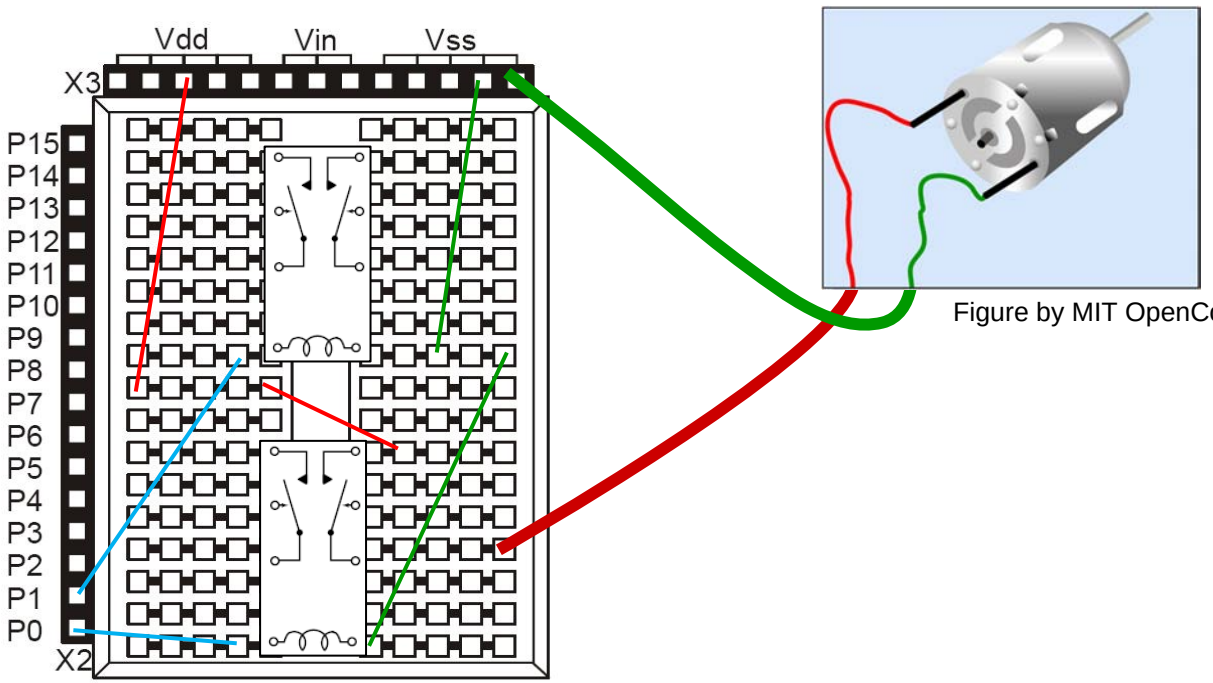


Figure by MIT OpenCourseWare.

How Would I Make a Reversible Control?

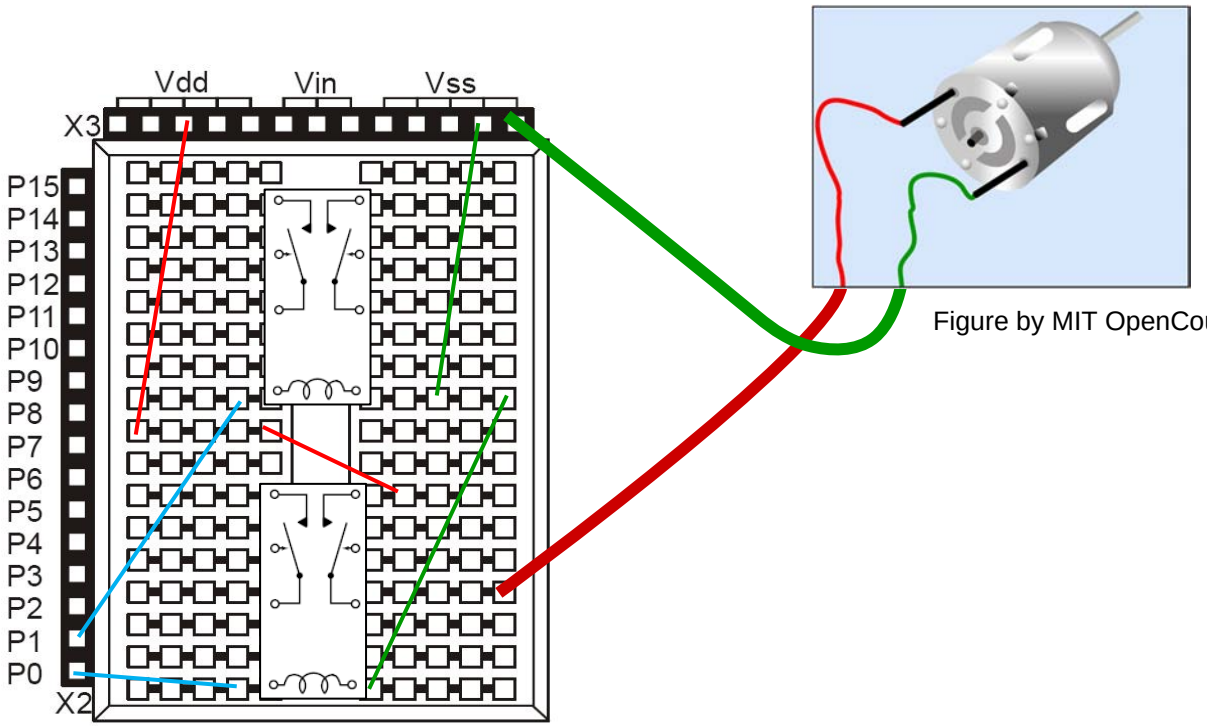


Figure by MIT OpenCourseWare.

Sensors

- Contact (mechanical)
- Proximity (optical)
- Range (acoustic)
- Force (piezo)

Images removed due to copyright restrictions. Please see
http://media.digikey.com/photos/Honeywell%20Photos/BZ-2RW82.jpg
http://www.trossenrobotics.com/store/i/is.aspx?path=/images/Pimages/S-10-GP2D120.jpg
http://www.parallax.com/Portals/0/Images/Prod/2/280/28015-M.jpg
http://media.digikey.com/photos/Parallax%20Photos/MFG_30056.jpg

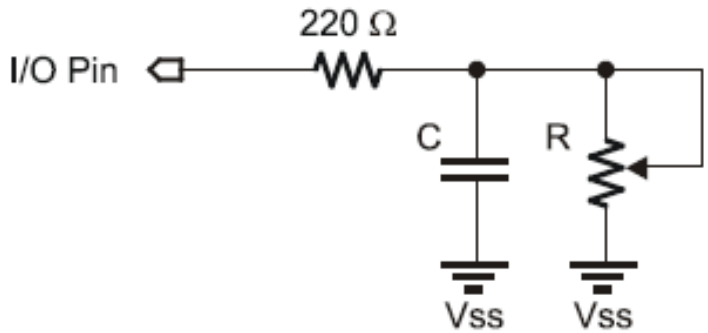
Force Measurement

- "piezoresistive"
  - (NOT piezoelectric)

Image removed due to copyright restrictions. Please see
http://media.digikey.com/photos/Parallax%20Photos/MFG_30056.jpg

http://www.tekscan.com/pdfs/DatasheetA201.pdf

RCTIME




```
RC PIN 7
result VAR Word
DO
HIGH RC ' charge the cap
PAUSE 1 ' for 1 ms
RCTIME RC, 1, result ' measure RC discharge time --the
    arguments are PIN, state (1=diagram “a”), and variable
DEBUG HOME, DEC result ' display value
PAUSE 50
LOOP
```

Acoustic Ranging/Detection

- Ultrasonic pulse
- Distance-to-target is by measuring the time required for echo

Image removed due to copyright restrictions. Please see
http://www.parallax.com/Portals/0/Images/Prod/2/280/28015-M.jpg
pp. 2 and 4 in http://www.parallax.com/Portals/0/Downloads/docs/prod/acc/28015-PING-v1.5.pdf

Next Steps

- Thursday 2 April
  - No lecture
  - Lab times that day instead
- Tuesday 7 April
  - Lecture on sensors and batteries
  - HW#3 is due!