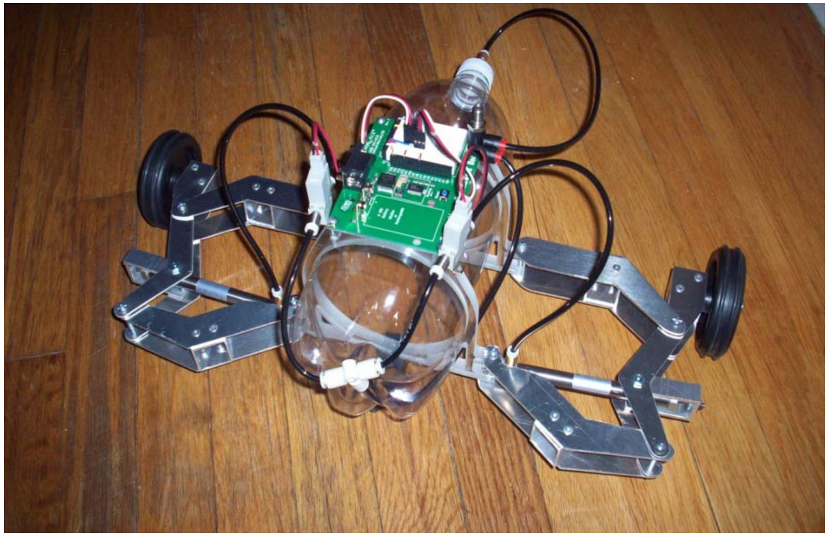
2.007 Design and Manufacturing I
Spring 2009

2.007 –Design and Manufacturing I

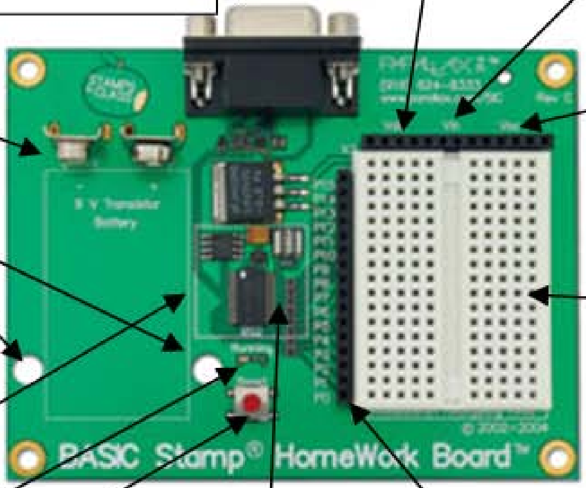
Microcomputers, Programming, Electronics, and Sensors



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

Attachment place for a 9V "transistor battery"

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

Tie down holes for battery

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

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

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?

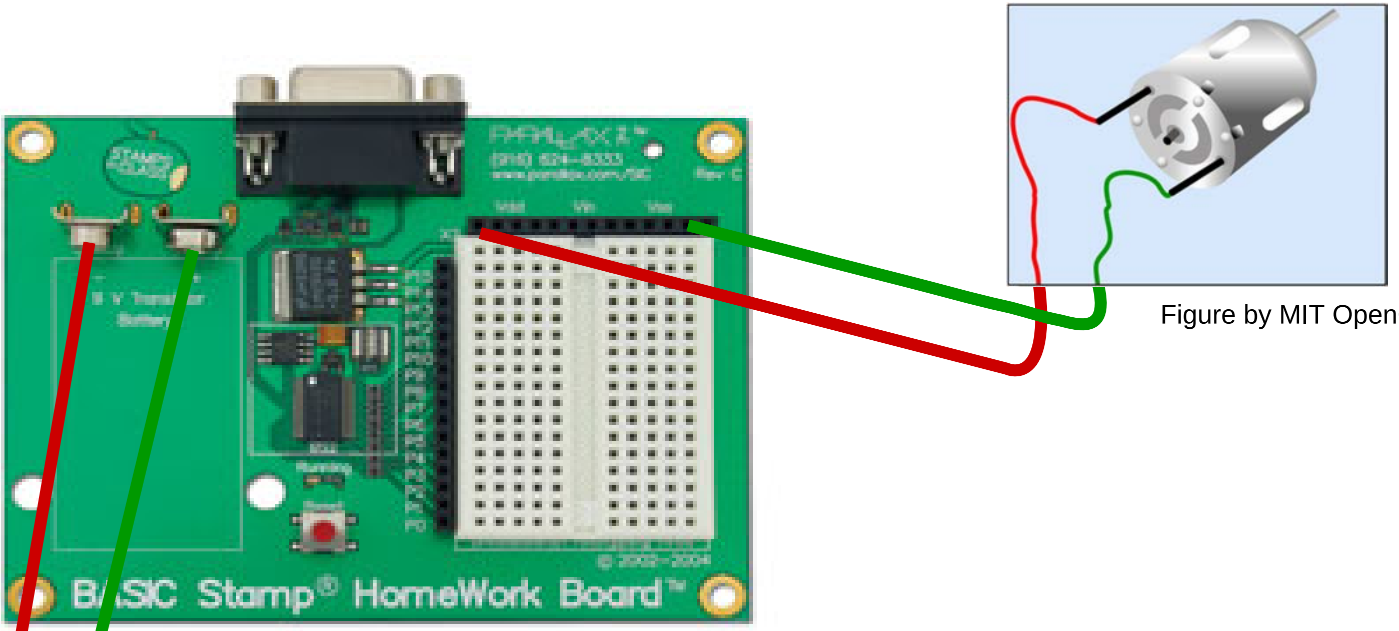


Figure by MIT OpenCourseWare.

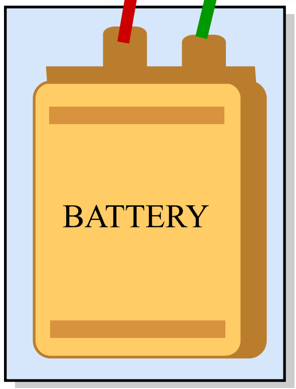
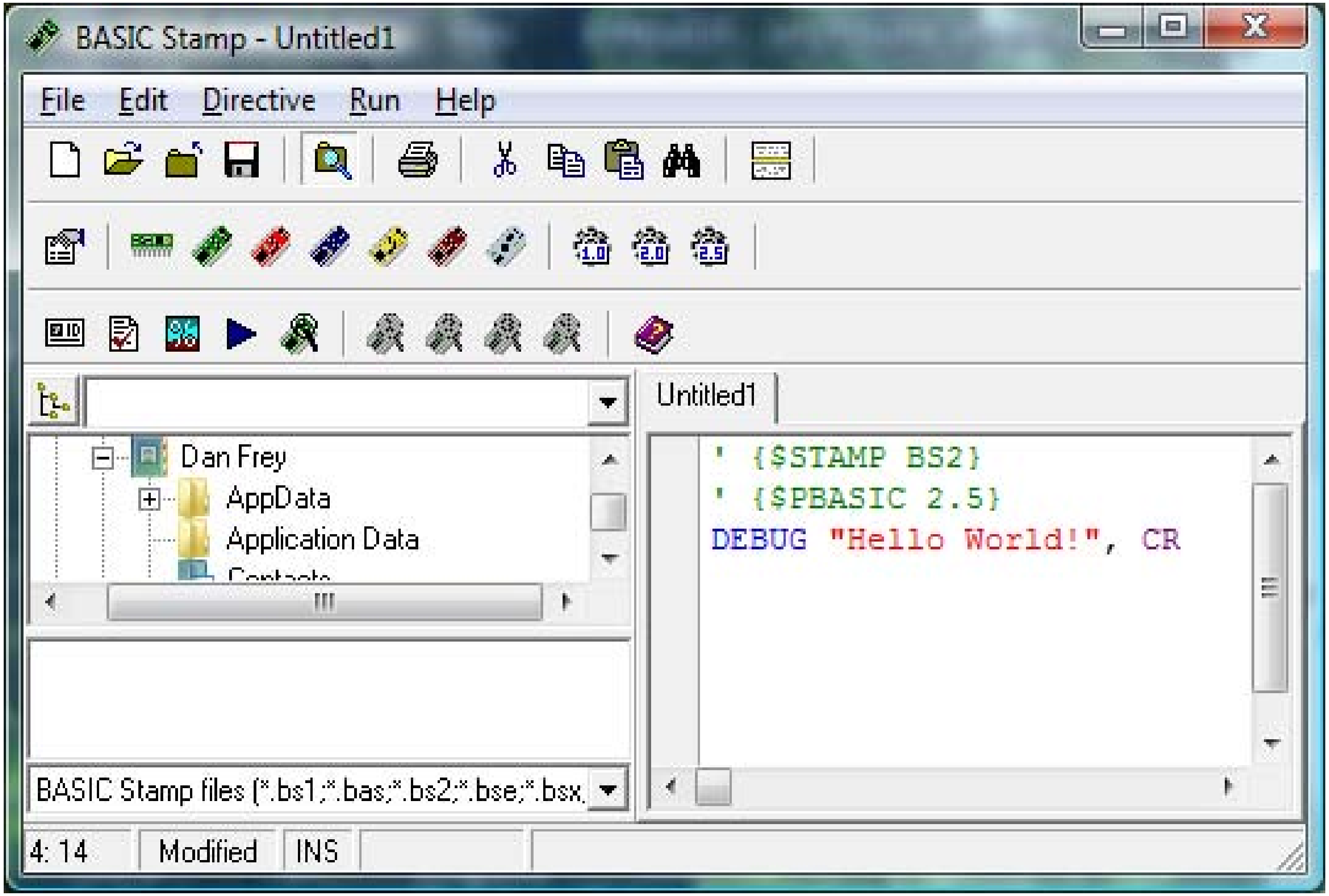


Figure by MIT OpenCourseWare.

NOTE: As we discussed in class, the motor will turn accelerating to its no load speed for 5V. Since the connection is to Vdd (not Vin) the motor “sees” 5V from the regulated supply, thus it is essentially independent of the current draw as long as the 9V battery is not dragged down to below 5V. This particular regulator can’t “buck” the voltage back up.

The Basic Stamp Editor



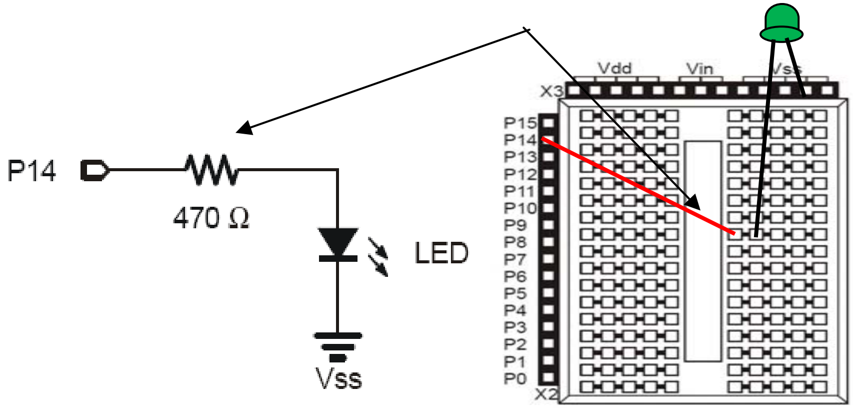
```
' {$STAMP BS2}
' {$PBASIC 2.5}
DEBUG "Hello World!", CR
```

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?

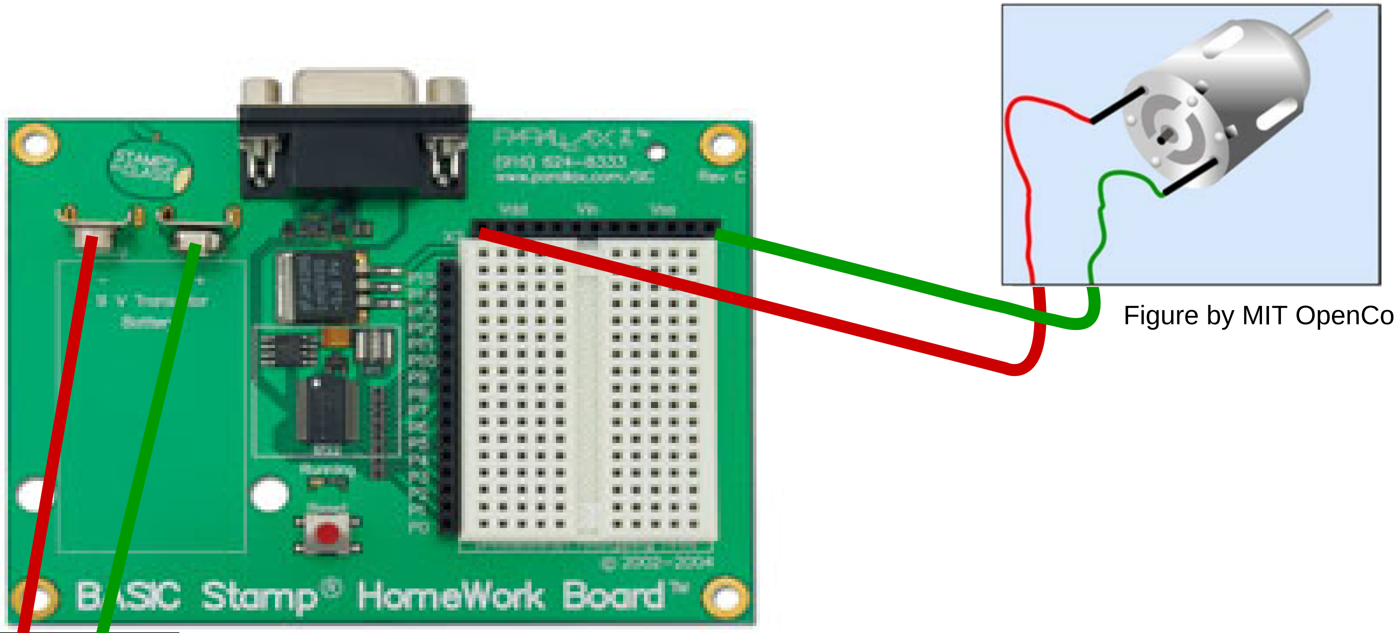


Figure by MIT OpenCourseWare.

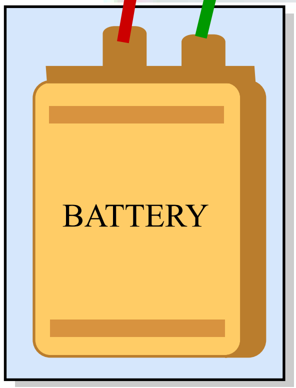


Figure by MIT OpenCourseWare.

NOTE: As we discussed in class, the motor will turn accelerating to its no load speed for 5V. Since the connection is to Vdd (not Vin) the motor "sees" 5V from the regulated supply, thus it is essentially independent of the current draw as long as the 9V battery is not dragged down to below 5V. This particular regulator can't "buck" the voltage back up.

What happens?

NOTE: Same as the last slide. The proto board area makes connections between the rows of 5 “clips”. For more background, see http://en.wikipedia.org/wiki/Breadboard

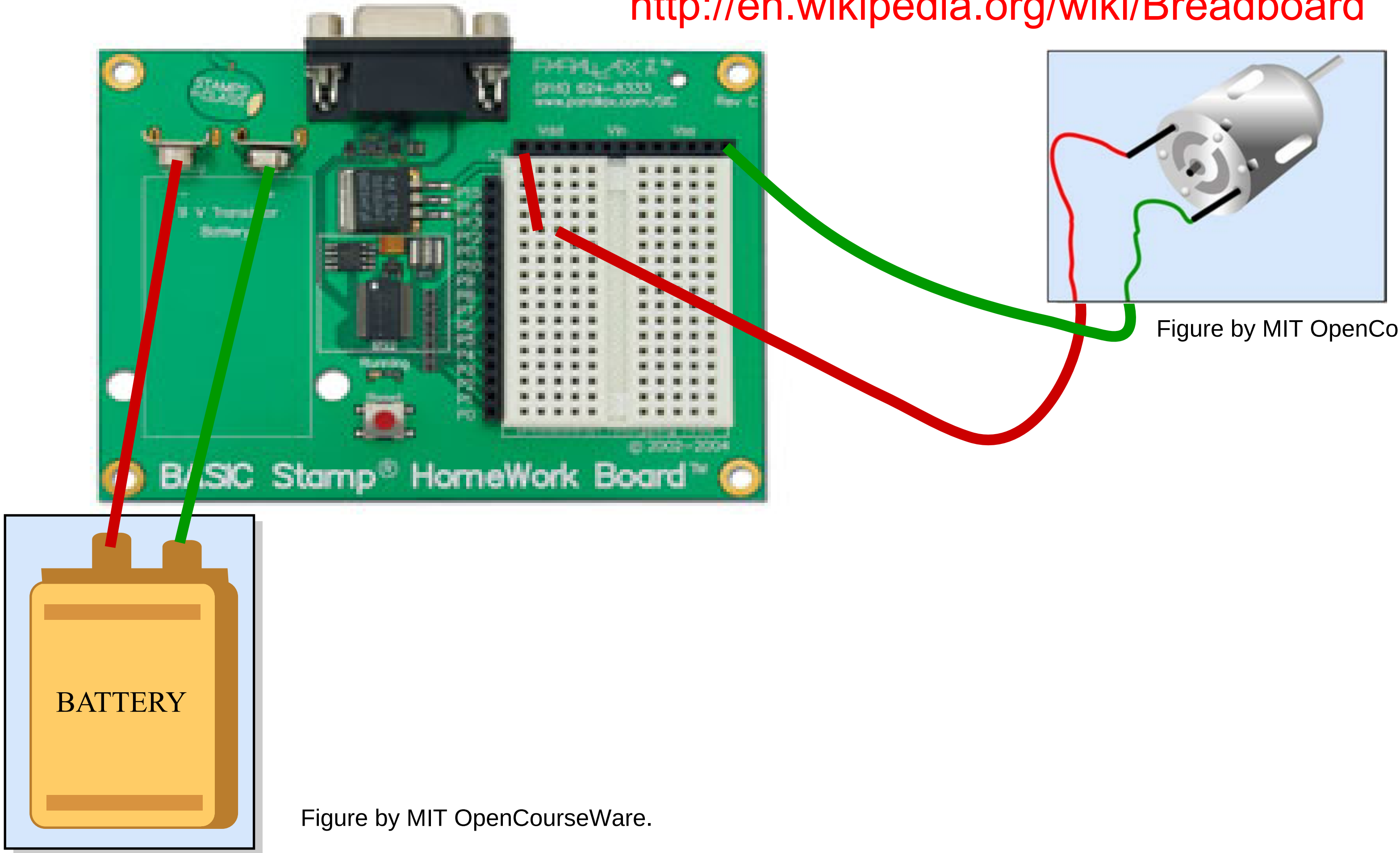


Figure by MIT OpenCourseWare.

Figure by MIT OpenCourseWare.

What happens?

NOTE: Nothing happens -- unlike the last slide. The proto board area does not make connections across the “trough” (unless you make such a connection with a wire).

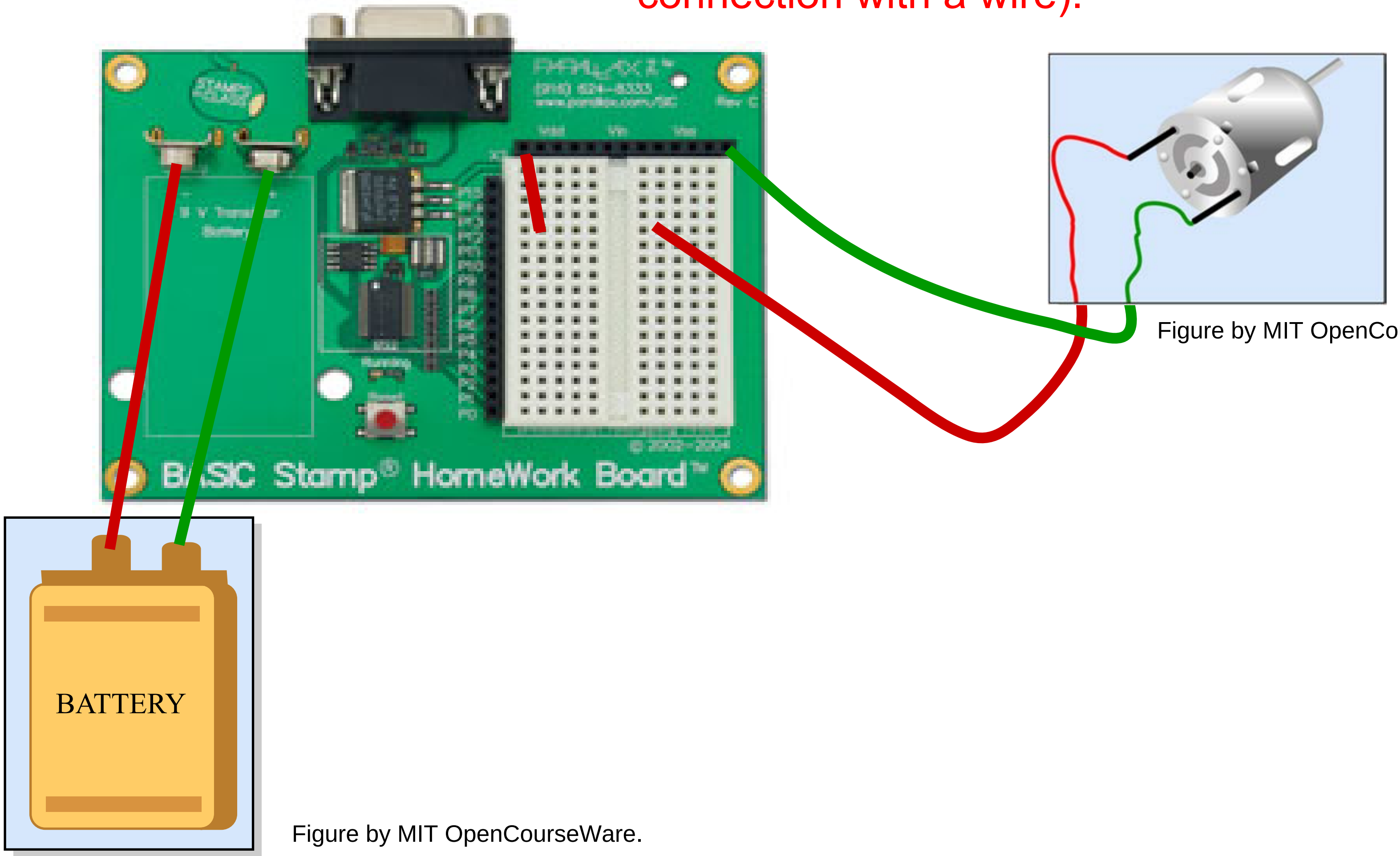


Figure by MIT OpenCourseWare.

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

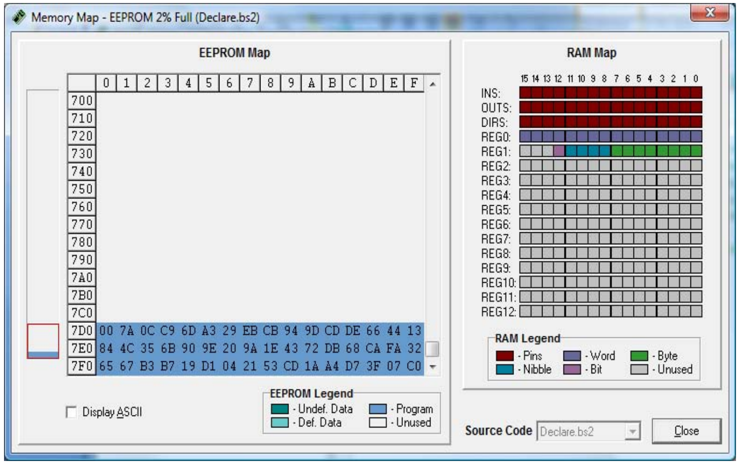
Horse VAR Word ' Horse is a variable that that uses 16 bits

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



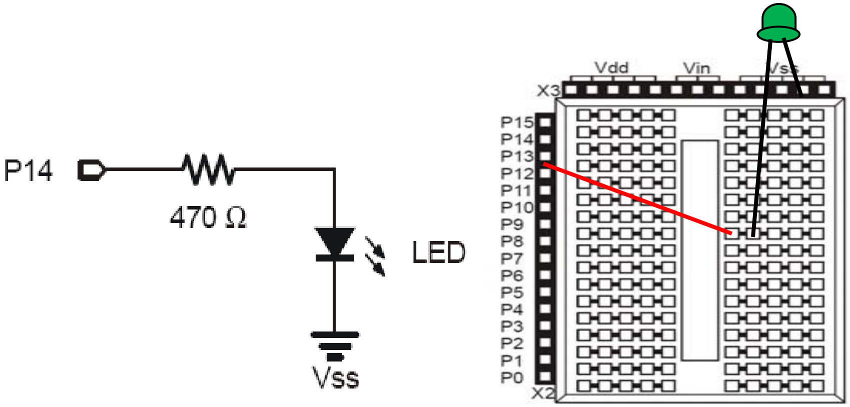
Memory Map - EEPROM 2% Full (Declare.bs2)

EEPROM Map

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
700
710
720
730
740
750
760
770
780
790
7A0
7B0
7C0
7D0 00 7A 0C C9 6D A3 29 EB CB 94 9D CD DE 66 44 13
7E0 84 4C 35 6B 90 9E 20 9A 1E 43 72 DB 68 CA FA 32
7F0 65 67 B3 B7 19 D1 04 21 53 CD 1A A4 D7 3F 07 C0
```

Making an LED Blink Increasingly Faster

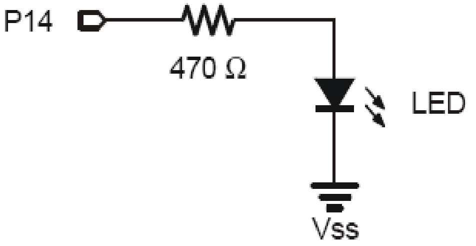
```
Delay VAR Nib
FOR Delay= 1 TO 15
HIGH 14
PAUSE 200-(Delay*10)
LOW 14
PAUSE 200-(Dela
NEXT
```



NOTE: The microcomputer will place the LED on high for 0.2 seconds and low for 0.2 sec and ramp down the duration to 0.05 sec as the loop executes. The LED will at first be blinking noticeably, but later just appear as if it's dimmer as it's seeing 2.5V rather than 5V sometimes and OV other times. This is pulse width modulation (PWM) of a source.

What Happens?

```
Delay VAR Nib
FOR Delay= 1 TO 15
HIGH 14
PAUSE 200-(Delay*10)
LOW 14
PAUSE 200-(Delay*10)
NEXT
```



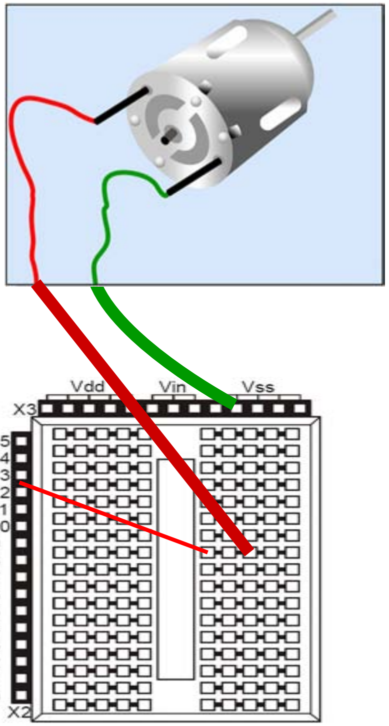
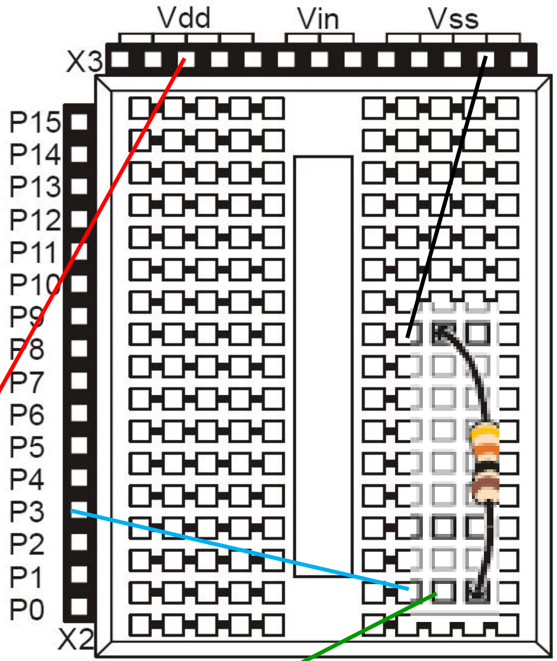


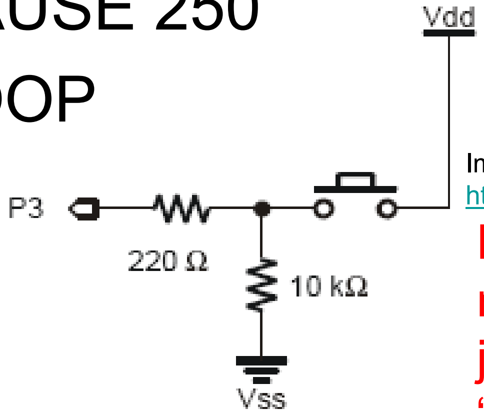
Figure by MIT OpenCourseWare.

NOTE: Basically nothing. The pins of the Stamp can only source 20milli Amps which cannot turn this motor at all. Maybe a very nice motor could turn at a slow rate. This motor costs pennies. Don’t expect too much.

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
```



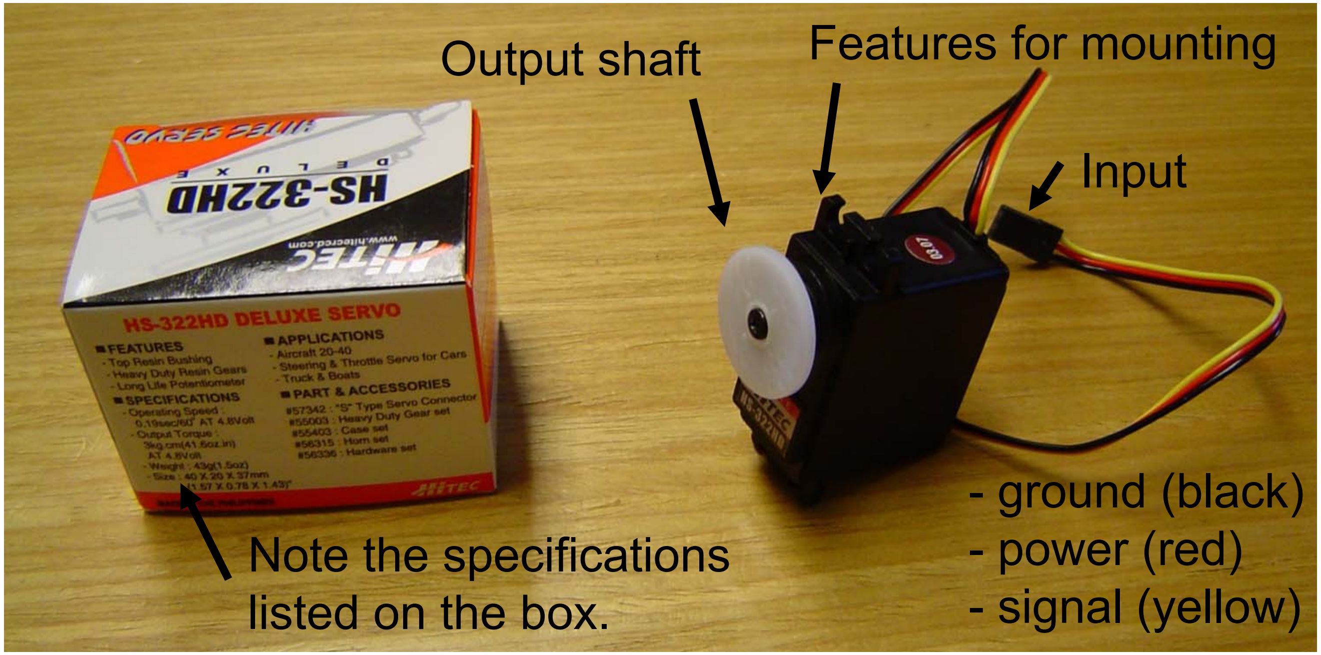


Images removed due to copyright restrictions. Please see http://media.digikey.com/photos/Honeywell%20Photos/BZ-2RW82.jpg

NOTE: There was a good question in class. Why do you need a 10kOhm resistor? Could a plain old wire do the job in its place? The item in question is functioning as a “pull up” resistor. A good explanation can be found at http://en.wikipedia.org/wiki/Pull-up_resistor

Servo Motors

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



Output shaft

Features for mounting

Input

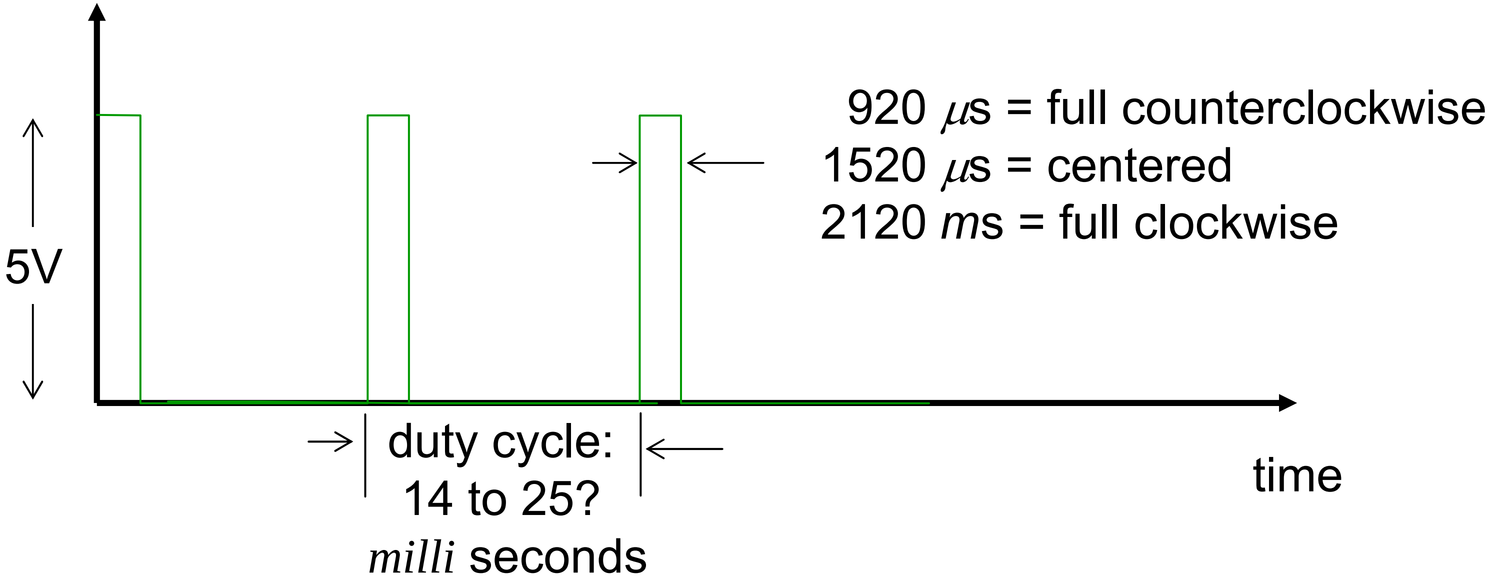
Note the specifications listed on the box.

- ground (black)
- power (red)
- signal (yellow)

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



920 μs = full counterclockwise

1520 μs = centered

2120 *ms* = full clockwise

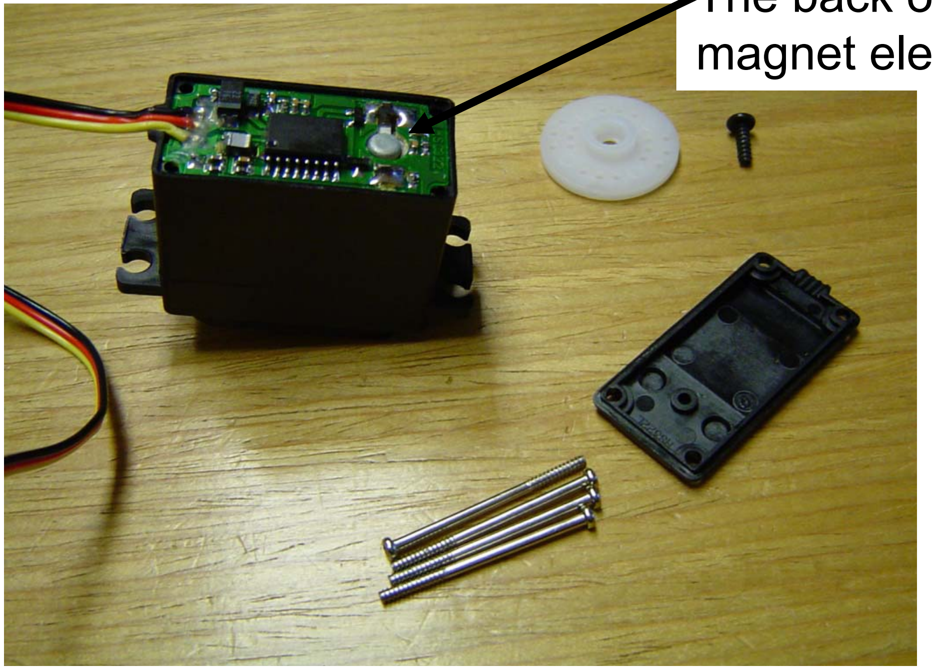
5V

duty cycle:
14 to 25?
milli seconds

time

Electronics Within the Servo

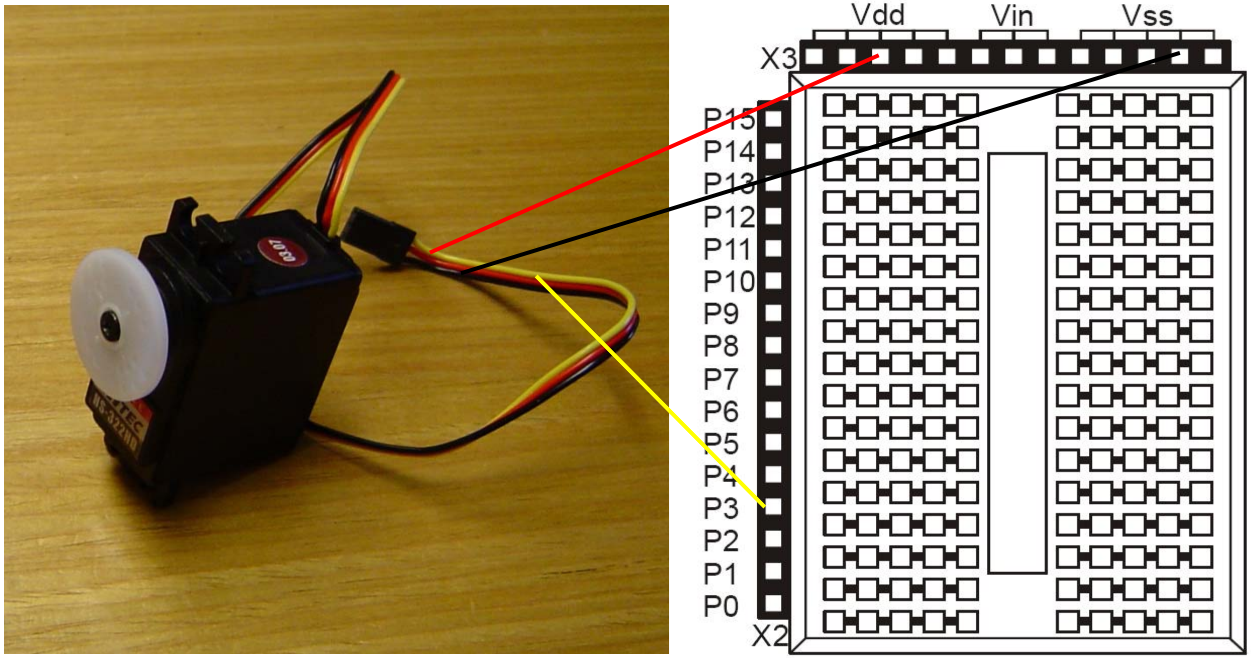
- Receive the commanded position
- Sense the position of the output shaft
- Supply voltage to the motor (either polarity) depending on the error



The back of a small, DC, permanent magnet electric motor

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
```



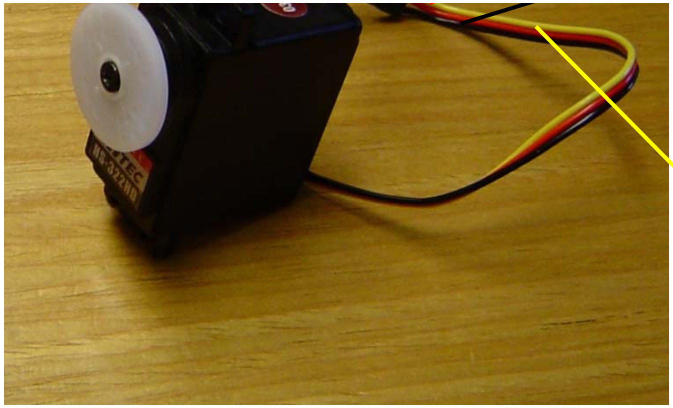
If I declare Reps as type Nib, what happens?

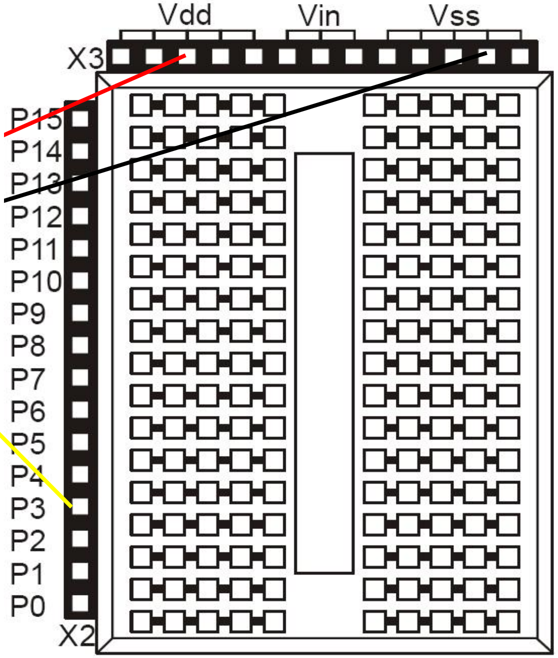
1) error message generated
2) program never leaves the first FOR loop
3) program leaves each FOR loop sooner
4) no difference

NOTE: The answer is “2”, the program never leaves the first FOR loop. A variable declared as a Nib has 4 bits and can only represent integers 0 through 15. When the FOR loop increments and Rep=15, there will be an overflow and Rep will go to zero. The BASIC stamp is very simple and so does not do much to help deal with run time errors like this. Other programming environments like Python will handle the error differently.

e Stamp

```
  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)

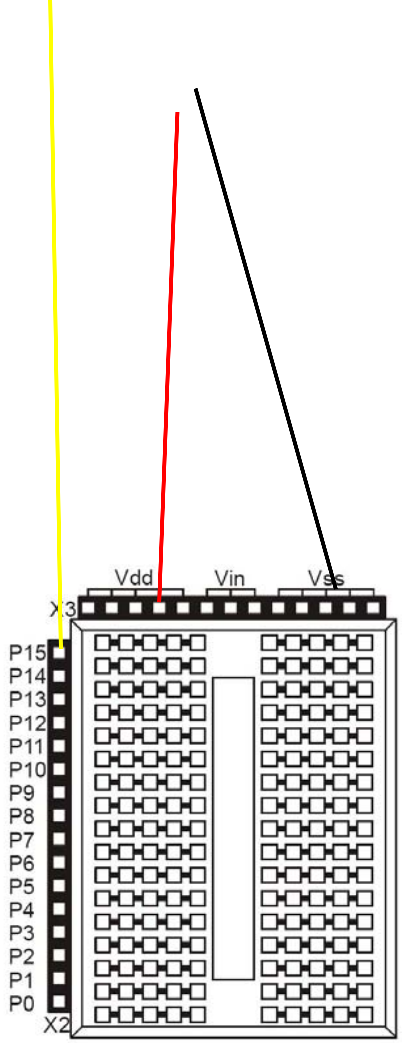
FM transmission

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

Elevator (ch2)

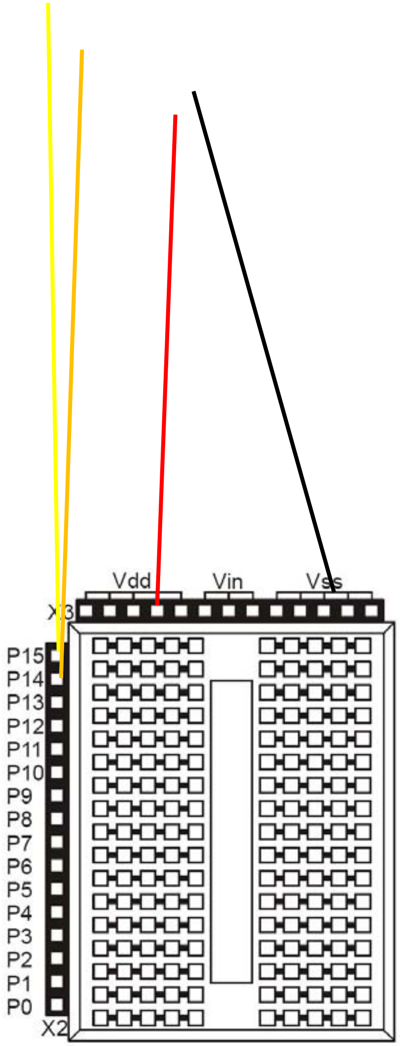
Rudder (ch 4)

Aileron (ch 1)



Getting Signals into the Stamp

```
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

An Issue with Arithmetic

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

DO
PULSIN 15, 1,throttle
PULSIN 14, 1, rudder
DEBUG home, ?
DEBUG ? rudde
```

```
result=throttle-2*rudder
DEBUG ? result

PAUSE 200
LOOP
```

NOTE: As discussed in class, the DEBUG here prints out a large number like 34000 because it executes the commands left to right. We are all used to languages assuming the usual priority of operators like first exponents, then multiplications, then addition. So watch out and use parentheses liberally.

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.

Another Issue with Arithmetic

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

DO
PULSIN 15, 1,throttle
PULSIN 14, 1, rudder
DEBUG home, ?
DEBUG ? rudde
```

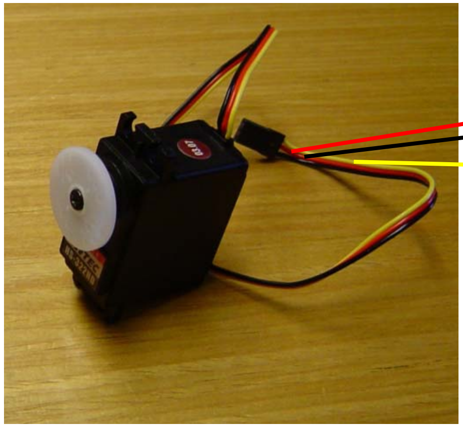
```
result=(throttle/rudder)*10
DEBUG ? result

PAUSE 200
LOOP
```

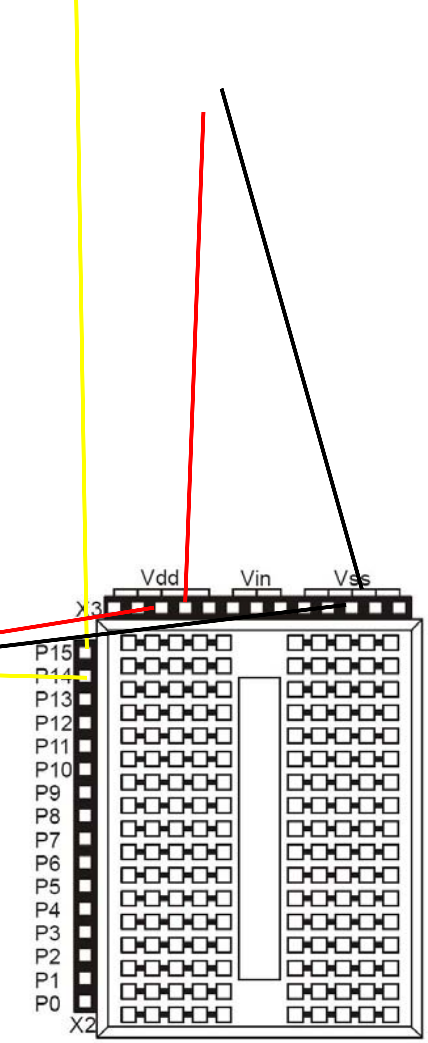
NOTE: As discussed in class, the DEBUG here prints out 0 or 10 skipping intermediate values we wanted like 7. We are all used to languages supporting real values. But this language represents all intermediate results as integers, so 740/760 is zero and 760/740 is 1.

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
```

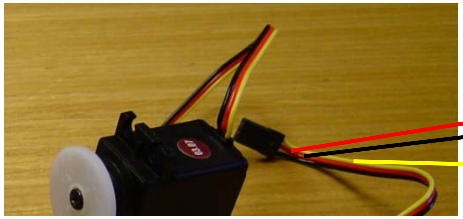


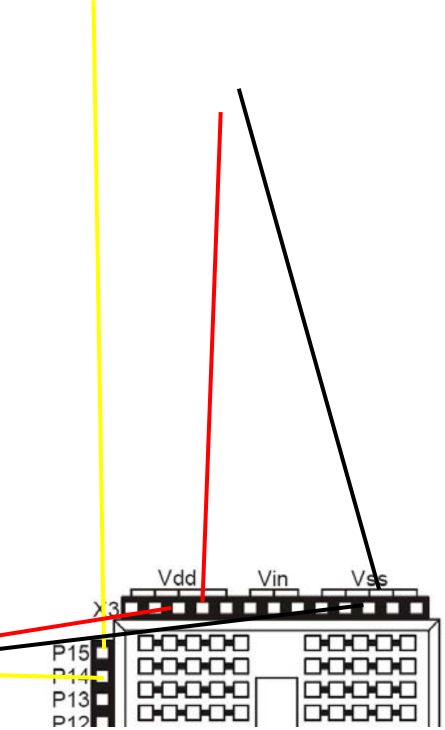


Expanding the Servo Range

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, respons
PAUSE 10
LOOP
```





NOTE: I also purchased "servo stretcher" devices so you don't necessarily have to do what on this slide. Still, you may want to do it if you're using the HW board anyway or if you want to mix signals or want the machine to operate autonomously sometimes or just don't want the hassle of sharing electronic modules.

Switching On/Off a Load

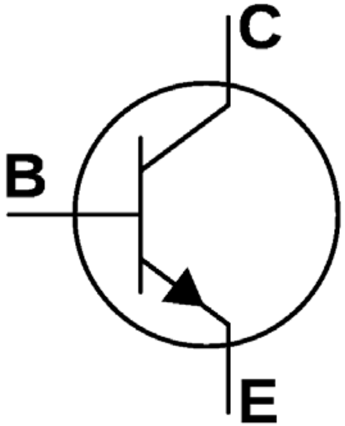


Image from Wikimedia Common, http://commons.wikimedia.org

The symbolic representation of the transistor
How the transistor (as packaged) looks literally

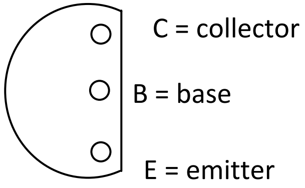


Image removed due to copyright restrictions. Please see
http://www.willow.co.uk/AZ822.pdf

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

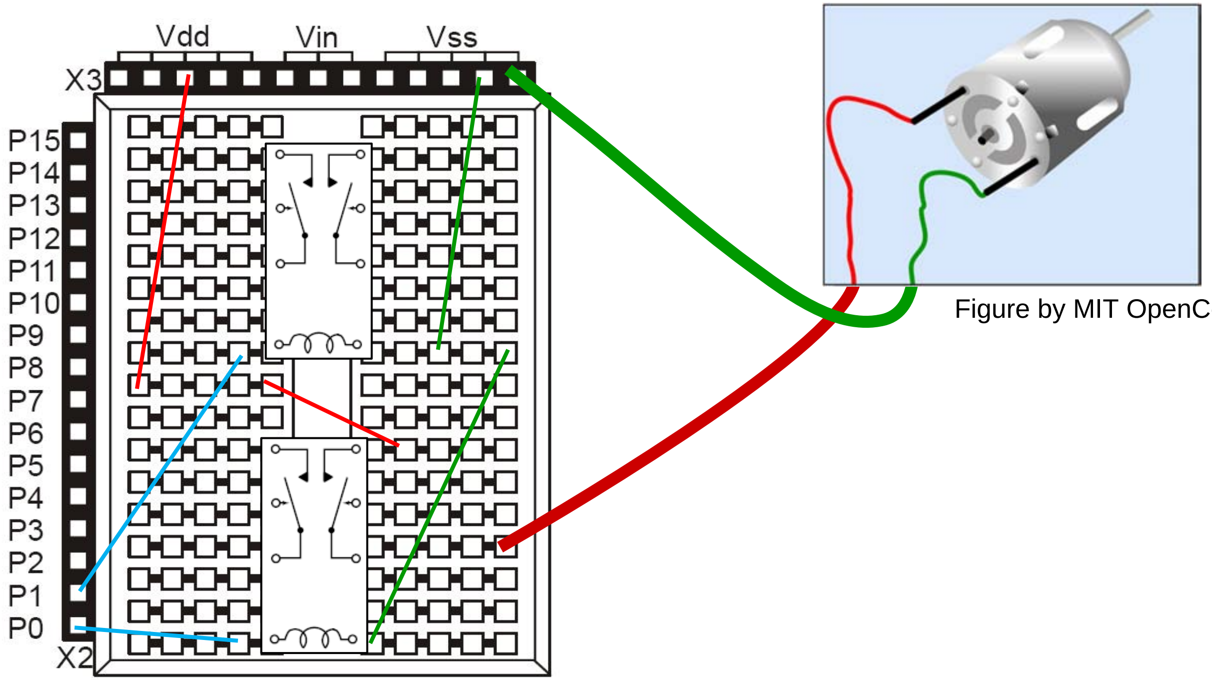


Figure by MIT OpenCourseWare.

How Would I Make a Reversible Control?

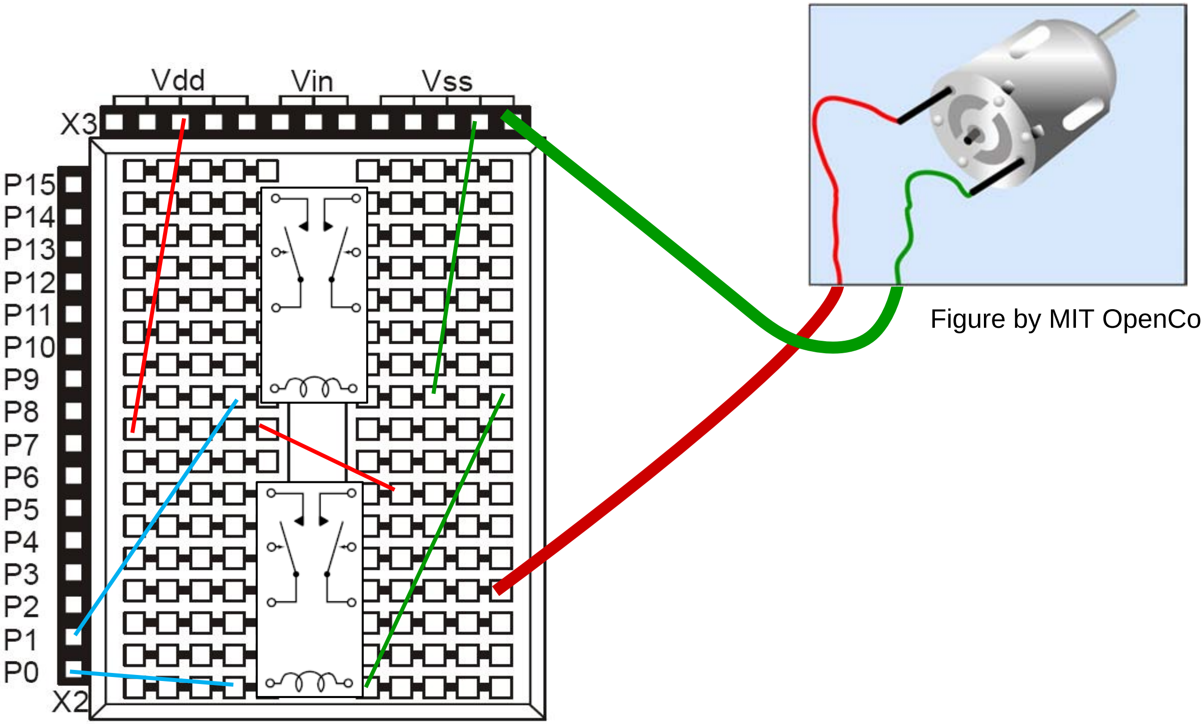


Figure by MIT OpenCourseWare.

NOTE: The practice exam solution nearly gives the solution, but not quite.

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

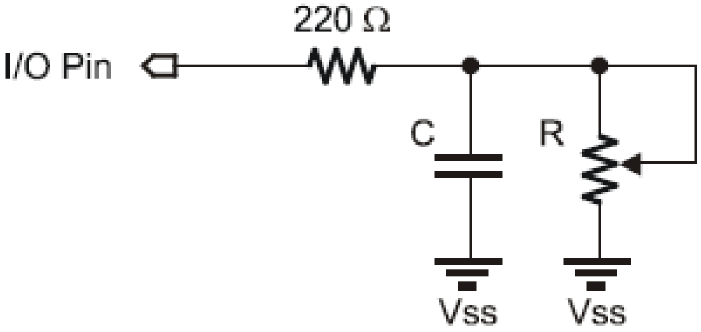
Force Measurement

- “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

Next Steps

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