



Chapter 8: Frequency and Sound



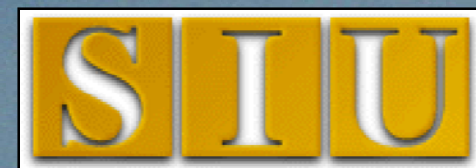
What's a Microcontroller?



Presentation based on:
"What's a Microcontroller ?"
By Andy Lindsay
Parallax, Inc

Presentation developed by:
Martin A. Hebel
Southern Illinois University Carbondale
College of Applied Sciences and Arts
Electronic Systems Technologies

9/10/03





Presentation Index

What's a Microcontroller?

- ✓ [Use and Copyright](#)
- ✓ [Electric Beeps](#)
- ✓ [Piezoelectric Speaker](#)
- ✓ [ACTIVITY #1: Building and Testing the Speaker](#)
- ✓ [Programming Speaker Control](#)
- ✓ [ACTIVITY #2: Action Sounds](#)
- ✓ [Example Nested Loop](#)
- ✓ [Two Frequencies at Once](#)
- ✓ [ACTIVITY #3: Musical Notes and Simple Songs](#)
- ✓ [Storing and Retrieving Data](#)
- ✓ [TwinkleTwinkle.bs2 – Abbreviated version](#)
- ✓ [NotesAndDurations.bs2](#)
- ✓ [Activity #4: Microcontroller Music](#)
- ✓ [Select Case](#)
- ✓ [Review Questions](#)
- ✓ [Links](#)



Use and Copyright

What's a Microcontroller?

This presentation supplements "**What's a Microcontroller**" by Andy Lindsay. ([Link to text](#) at Parallax)

- ✓ This presentation is not a replacement for the text.
- ✓ Important concepts of the text are highlighted.
- ✓ In some cases, additional material has been added to augment the text. Denoted by titles colored **gold**.
- ✓ Full program listings are generally not provided in the presentation.

Distribution:

This presentation may be freely distributed without modifications. Modifications are permitted by schools and organizations for internal use only. Credits, use and copyright slides must remain.



COPYRIGHTS AND TRADEMARKS

This documentation is Copyright 2003 by Parallax, Inc. By downloading or obtaining a printed copy of this documentation or software you agree that it is to be used exclusively with Parallax products. Any other uses are not permitted and may represent a violation of Parallax copyrights, legally punishable according to Federal copyright or intellectual property laws. Any duplication of this documentation for commercial uses is expressly prohibited by Parallax, Inc. Check with Parallax for approval prior to duplicating any of our documentation in part or whole for any use.

BASIC Stamp is a registered trademark of Parallax, Inc. If you decide to use the name BASIC Stamp on your web page or in printed material, you must state that "BASIC Stamp is a registered trademark of Parallax, Inc." Other brand and product names are trademarks or registered trademarks of their respective holders.

DISCLAIMER OF LIABILITY

Parallax, Inc. and Southern Illinois University are not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, or any costs of recovering, reprogramming, or reproducing any data stored in or used with Parallax products. Parallax is also not responsible for any personal damage, including that to life and health, resulting from use of any of our products. You take full responsibility for your BASIC Stamp application, no matter how life threatening it may be.



Electric Beeps

What's a Microcontroller?

From your alarm clock to microwave to automobiles and ATM machines your day is full of devices sounding beeps to alert you or indicate actions to be taken.

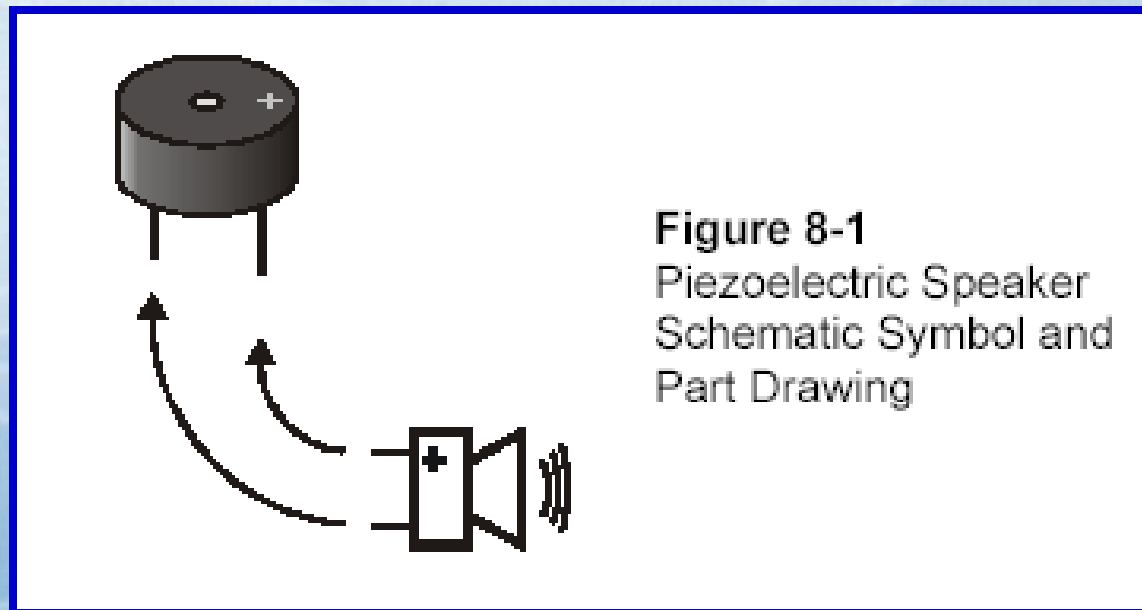
Microcontrollers produce sounds by sending high/low signals very quickly to a speaker. The speed at which the signal repeats is called **frequency** and is measured in **cycles per second** or **Hertz (Hz)** which produce the desired tone or pitch.



Piezoelectric Speaker

What's a Microcontroller?

- ✓ The piezoelectric speaker is common, small and inexpensive speaker used in many devices though it lacks in audio quality.





ACTIVITY #1: Building and Testing the Speaker

What's a Microcontroller?

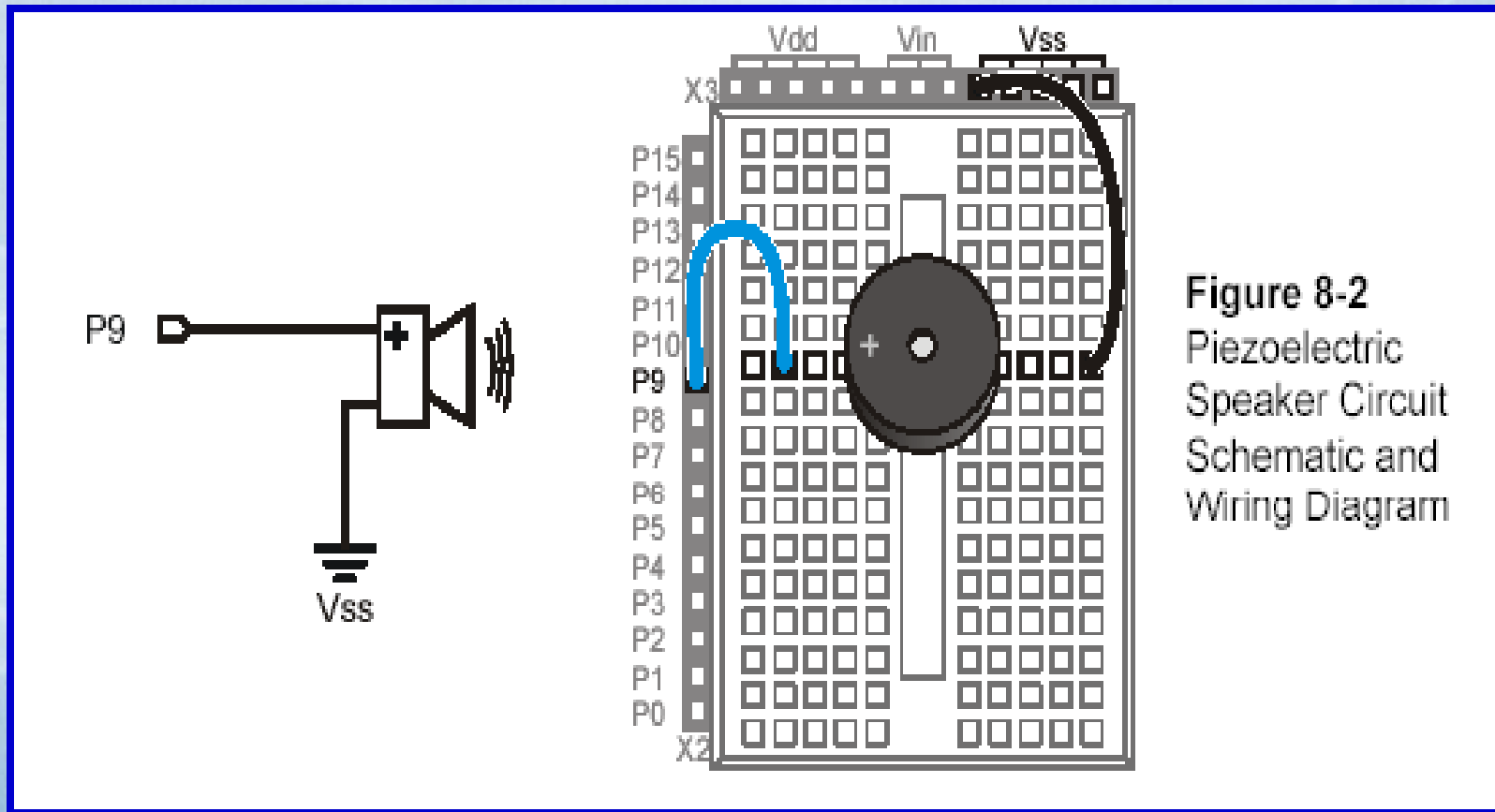


Figure 8-2
Piezoelectric
Speaker Circuit
Schematic and
Wiring Diagram



Programming Speaker Control

What's a Microcontroller?

The `FREQOUT` command sends high/low signals to the specified pin at the frequency and for the duration defined.

`FREQOUT` Pin, Duration, Freq1, {Freq2}

To play a note at 2000Hz which lasts 1.5 seconds:

```
' What's a Microcontroller - TestPiezoWithFreqout.bs2
' Send a tone to the piezo speaker using the FREQOUT command.
' {$STAMP BS2}
' {$PBASIC 2.5}
DEBUG "Tone sending...", CR
FREQOUT 9, 1500, 2000
DEBUG "Tone done."
```




ACTIVITY #2: Action Sounds

What's a Microcontroller?

In ActionTones.bs2 a variety of tones are played. Alarm and Robot Reply are a sequence of tones sent to the speaker.

In Hyperspace, a **nested loop** is used where FREQOUT cycles through durations from 15 to 1. For each duration it cycles through frequencies from 2000 to 2500 in increments of 20.



What's a Microcontroller?

Outer
Loop

```
FOR duration = 15 TO 1 STEP 1
  FOR frequency = 2000 TO 2500 STEP 20
    FREQOUT 9, duration, frequency
  NEXT
NEXT
```

Inner
Loop

The inner loop is performed fully every repetition of the outer loop.



Example Nested Loop

What's a Microcontroller?

```
duration VAR Word
frequency VAR Word

DEBUG "Duration Frequency", CR,
"-----", CR

FOR duration = 4000 TO 1000 STEP 1000
  FOR frequency = 1000 TO 3000 STEP 500
    DEBUG " ", DEC5 duration,
          " ", DEC5 frequency, CR
    FREQOUT 9, duration, frequency
  NEXT
  DEBUG CR
NEXT
```



Two Frequencies at Once

What's a Microcontroller?

The FREQOUT command has an optional parameter called Freq2. This allows playing 2 frequencies simultaneously.

At times the frequencies will combine to aid and at other times oppose creating a beat frequency at the difference between the two.

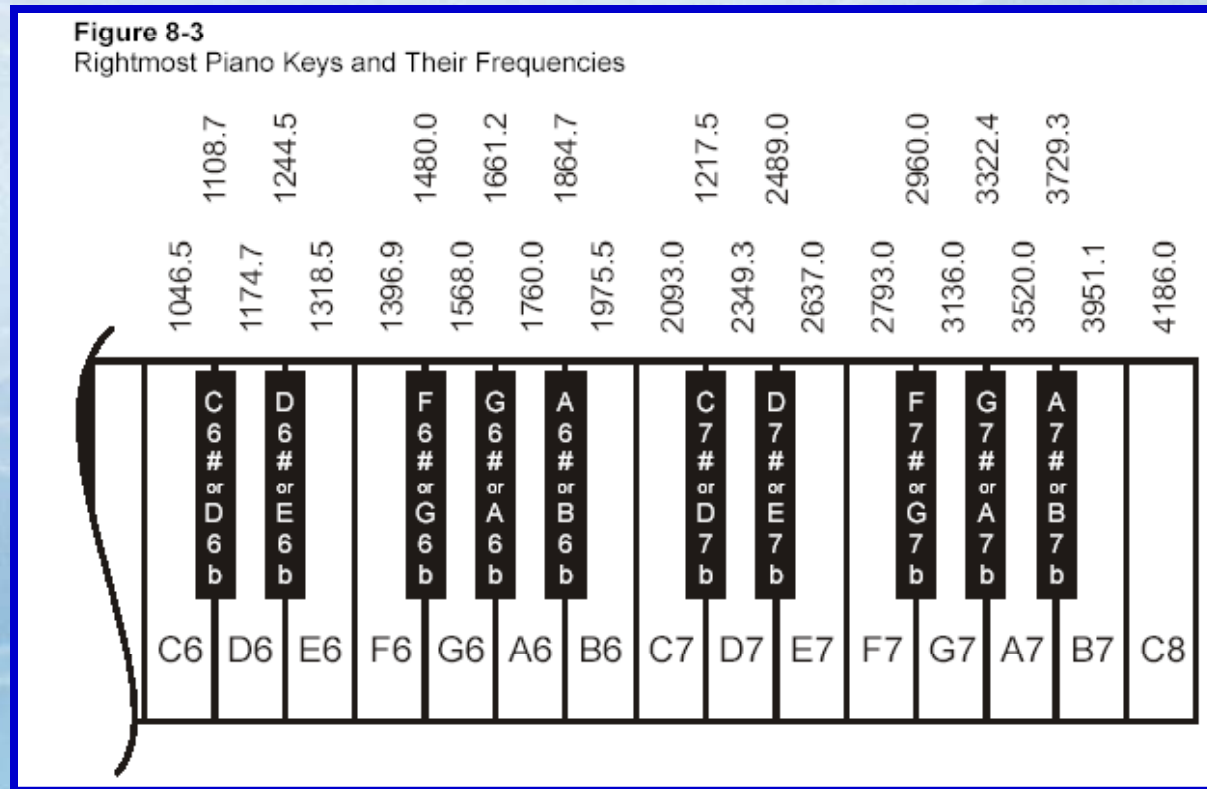
FREQOUT 9, 5000, 2000, 2005



ACTIVITY #3: Musical Notes and Simple Songs

What's a Microcontroller?

Each key on a piano is a specific frequency corresponding to a note. There are 12 groups of notes, each at a higher octave. An octave is a doubling of frequency, so C7 is double the frequency of C6.





Storing and Retrieving Data

What's a Microcontroller?

The DATA command is similar to WRITE but stores a list of expressions.

```
{Symbol} DATA {Word} DataItem1,{DataItem2,... }
```

For example:

```
Notes DATA "C","C","G","G","A","A","G"
```

Stores the characters in EEPROM, with the 1st location called Notes. Each subsequent address is Notes+index value.



What's a Microcontroller?

Notes DATA "C", "C", "G", "G", "A", "A", "G"

Memory Map - EEPROM 6% Full (Untitled1)

Detailed EEPROM Map

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	C	C	G	G	A	A	G									
010																
020																
030																
040																
050																
060																
070																
080																
090																
0A0																
0B0																
0C0																
0D0																
0E0																
0F0																

RAM Map

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INS:																
OUTS:																
DIRS:																
REG0:																
REG1:																
REG2:																
REG3:																
REG4:																
REG5:																
REG6:																
REG7:																
REG8:																
REG9:																
REG10:																
REG11:																
REG12:																

Condensed EEPROM Map

Source Code Untitled1

EEPROM Legend

- - Undef. Data
- - Def. Data
- - Program
- - Unused

RAM Legend

- - Pins
- - Word
- - Byte
- - Nibble
- - Bit
- - Unused

Display ASCII

Close

Check to view ASCII



This code stores a list referenced by Frequencies with each taking 2 bytes because they are greater than 255 and stored as Words. As such, when read, $\text{index} * 2$ is used to jump 2 at a time.

**Frequencies DATA Word 2093, Word 2093,
Word 3136, Word 3136,
Word 3520, Word 3520,
Word 3136**



TwinkleTwinkle.bs2 – Abbreviated version

What's a Microcontroller?

When index = 0

```
Notes DATA "C" "C", "G", "G", "A", "A", "G"  
Frequencies DATA Word 2093, Word 2093, Word 3136, Word 3136,  
Word 3520, Word 3520, Word 3136
```

```
Durations DATA Word 500, Word 500, Word 500, Word 500,  
Word 500, Word 500, Word 1000
```

```
index VAR Nib  
noteLetter VAR Byte  
noteFreq VAR Word  
noteDuration VAR Word
```

```
FOR index = 0 TO 6  
  READ Notes + index, noteLetter  
  
  READ Durations + (index * 2), Word noteDuration  
  
  READ Frequencies + (index * 2), Word noteFreq  
  
  FREQOUT 9, noteDuration, noteFreq  
NEXT
```



When index = 1

```
Notes DATA "C" "C", "G", "G", "A", "A", "G"
Frequencies DATA Word 2093, Word 2093, Word 3136, Word 3136,
Word 3520, Word 3520, Word 3136

Durations DATA Word 500, Word 500, Word 500, Word 500,
Word 500, Word 500, Word 1000
```

```
index VAR Nib
noteLetter VAR Byte
noteFreq VAR Word
noteDuration VAR Word
```

```
FOR index = 0 TO 6
  READ Notes + index, noteLetter

  READ Durations + (index * 2), Word noteDuration

  READ Frequencies + (index * 2), Word noteFreq

  FREQOUT 9, noteDuration, noteFreq
NEXT
```



Activity #4: Microcontroller Music

What's a Microcontroller?

Note durations in music are defined as whole, half, quarter, eighth, sixteenth and thirty-second of a whole note.

The duration of the whole note depends on the tempo of the music. Some music has a vary fast tempo, others very slow.

Rests are durations when no tones are played.



NotesAndDurations.bs2

What's a Microcontroller?

This program uses a combination of Data, lookup and lookdown to play a piece of music.

The music to be played is stored using DATA. For durations: 1=whole, 2= 1/2 and so on.

```
Notes DATA "C", "D", "E", "C", "C", "D", "E", "C", "E", "F",  
           "G", "E", "F", "G", "Q"
```

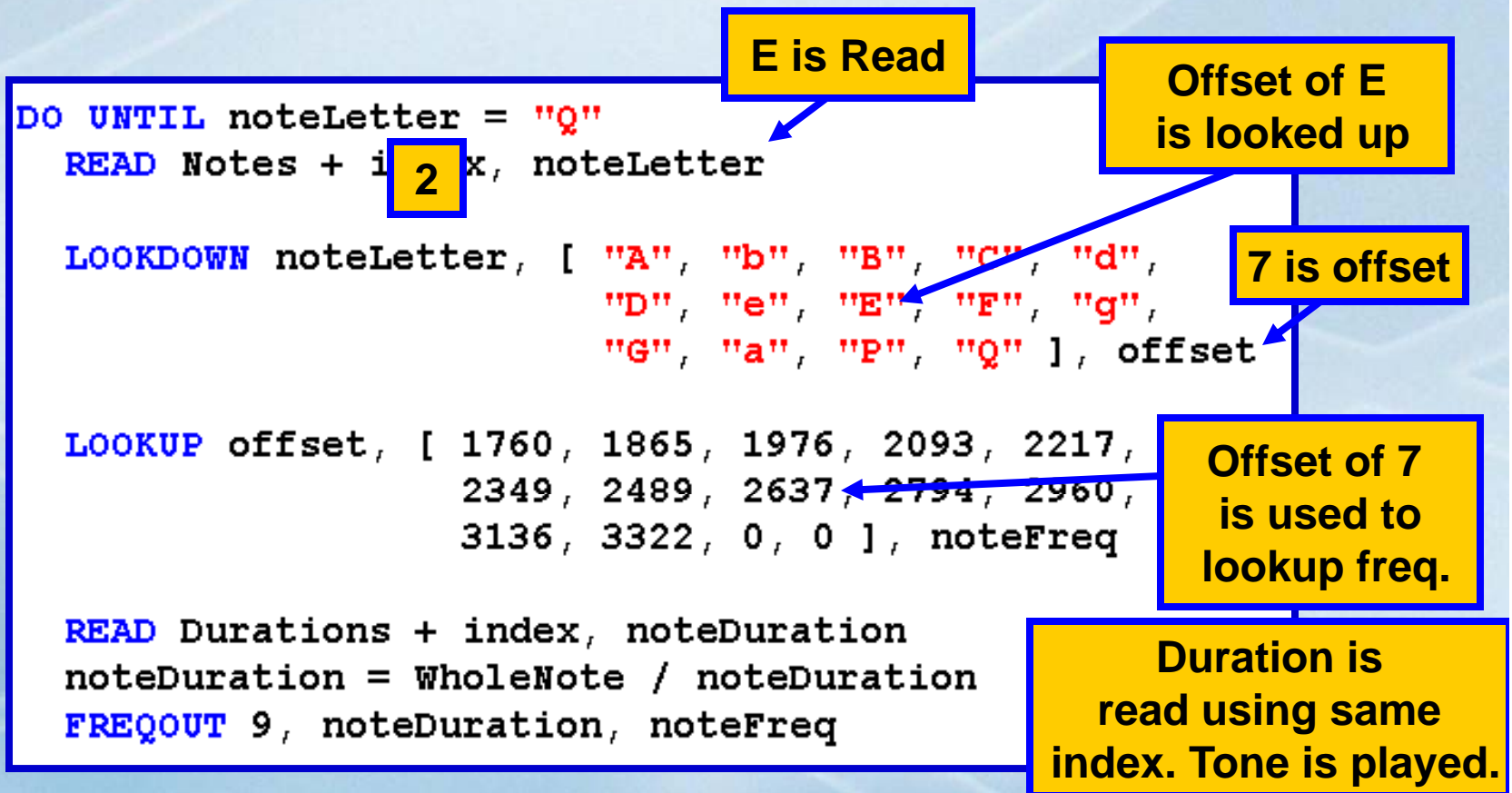
```
Durations DATA 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,  
              2, 4, 4, 2
```

```
WholeNote CON 2000
```



What's a Microcontroller?

Index is used to read each note and duration, the note's frequency is looked up. For example, for note 3 (index=2).





Select ... Case

Select...Case is a very clean method of performing a code block based on a value.

```
SELECT expression  
  CASE condition(s)  
    statements  
ENDSELECT
```

Depending on the value of expression, the code of any CASE blocks will be ran.



SelectCaseWithValues.bs2

What's a Microcontroller?

```
value VAR Word
DEBUG "Enter a value from", CR,
"0 to 65535: "
DO
  DEBUGIN DEC value
  SELECT value
    CASE 0, 1
      DEBUG "Bit", CR
      PAUSE 100
    CASE 2 TO 15
      DEBUG "Nib (Nibble)", CR
      PAUSE 200
    CASE 16 TO 255
      DEBUG "Byte", CR
      PAUSE 300
    CASE 256 TO 65535
      DEBUG "Word", CR
      PAUSE 400
  ENDSELECT
  DEBUG CR, "Enter another value: "
LOOP
```

Variable expression
to be checked

Conditions to check
expression against

If condition is true,
Code will be ran. If not,
it will be skipped

Each CASE
will be checked

Defines the end of the
SELECT...CASE block



What's a Microcontroller?

The CASE conditions are very versatile depending on your need:

CASE 100

CASE "A"

CASE "A" TO "Z"

CASE 50 TO 100

CASE >100, <50

CASE <>"q" (not equal too)

CASE "A", "a"



Review Questions

What's a Microcontroller?

- ✓ The command to generate a frequency is **FREQOUT**
- ✓ The 1st parameter or argument in the command defines the pin. The 2nd defines the **DURATION** and the 3rd the **FREQUENCY**
- ✓ A FOR-NEXT Loop inside another is called a **NESTED** Loop.
- ✓ When using the DATA command, the expressions are stored in **EEPROM**.
- ✓ Given the code fragment, what would X be? **3**
Vals DATA 1,2,3,5
READ Vals+2,X
SELECT...CASE uses a specific expression and will run a code block depending on the condition of the expression.



Links

What's a Microcontroller?

- ✓ [BASIC Stamp Home](#)
- ✓ [Stamps In Class Home](#)
- ✓ [BASIC Stamp Software](#)
- ✓ [BASIC Stamp Robots](#)
- ✓ [BASIC Stamp Yahoo Group](#)
- ✓ [Stamps In Class Yahoo Group](#)
- ✓ [SIUC EST Degree](#)
- ✓ [StampPlot Lite Software](#)