Chapter 7: Measuring Light

What's a Microcontroller?

PARALLAX inc

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

SIU

Presentation Index

Use and Copyright

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:

COPYRIGHTS AND TRADEMARKS

DISCLAIMER OF LIABILITY

Devices that Contain Light Sensors

Pushbuttons and dials are examples of pressure and rotation sensors, but there are a variety of sensors used for a variety of purposes: Temperature, smoke, tilt, vibration and so on.

Light sensors are also a type used in a variety of applications:

- ✓ Automatic street lights
- ✓ TV remotes using Infrared (non-visible light).
- ✓ Camera flash and exposure controls

Introducing the Photoresistor

While there are a variety of light sensors, a very popular one is the photoresistor in that it is easy to use and inexpensive.

As the name implies, it is a resistor that reacts to light. The active ingredient Cadmium Sulfide (CdS) allows electrons to flow more easily when light energy hits it, thus lowering it resistance (opposition to current flow).

The brighter the light the lower the resistance.

Just as with a carbon resistor, the photoresistor can be used with the BASIC Stamp in an RC circuit to obtain a value in relation to the amount of resistance, or in this case the amount of light hitting the sens

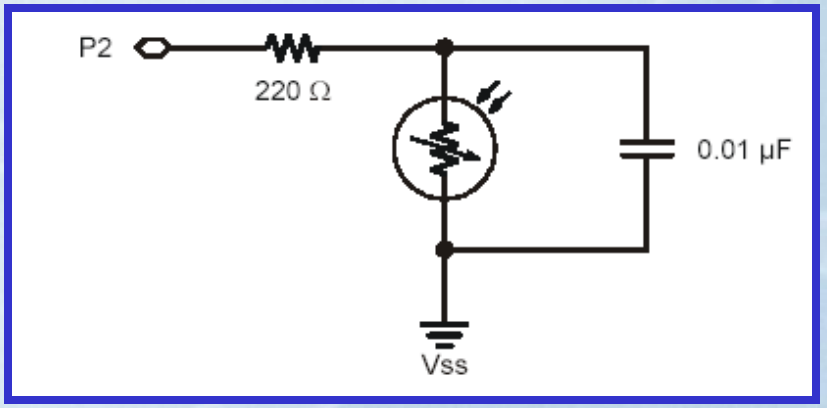
Figure 7-1
Photoresistor Schematic and Part Drawing.

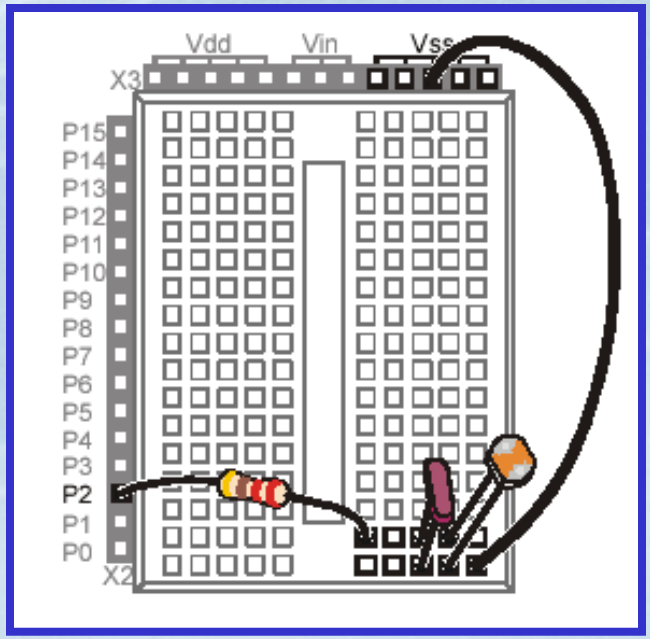
The photoresistor's cadmium sulfide coated light collecting surface is shown at the top of the part's drawing.

Activity #1: Build & Test Light Meter

- Just as with the RC Time circuits in Chapter 5, the capacitor is charged by the output (P2 in this case) and the time to discharge though the resistor is measured.

In this case, as light level changes, discharge time will change.





- ✓What happens to the value of **time** as the light level changes? When is it lowest? Highest?

```
' What's a Microcontroller - TestPhotoresistor.bs2
' Read photoresistor in RC-time circuit using RCTIME command.
' {$STAMP BS2}
' {$PBASIC 2.5}

time VAR Word

DO
  HIGH 2
  PAUSE 100
  RCTIME 2, 1, time
  DEBUG HOME, ? time, " "
LOOP
```

Activity #2: Graphing Light Level

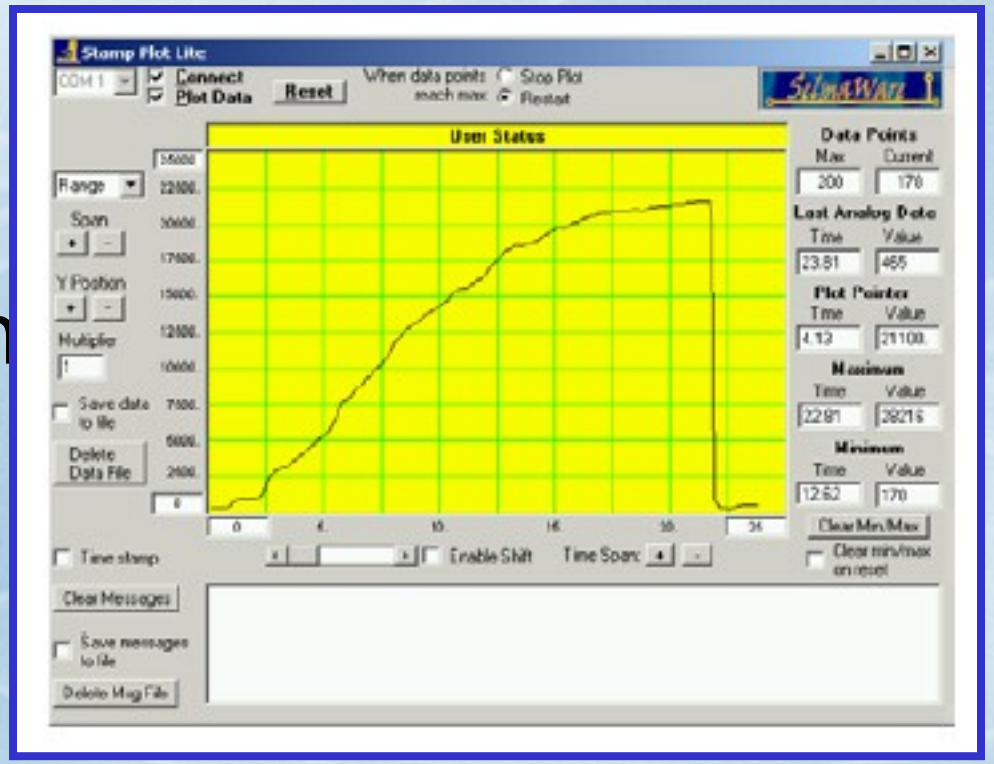
Monitoring of sensors is a need in industry to ensure systems are operating within specifications.

Stamp Plot Lite is a PC-based graphing program for monitoring data graphically.

Using Stamp Plot Lite

This image shows a plot of the light level. Note how the value increases from left to right then drops again suddenly?

What do the values represent as far as light level?



- ✓StampPlot Lite may be installed from your CD or downloaded from Parallax's website.
- ✓Winzip is required to extract and install the software.

Sending Measurements to Stamp Plot

Values to be graphed are sent using DEBUG and the DEC formatter. All data and instructions sent to Stamp Plot must end in a carriage return (CR).

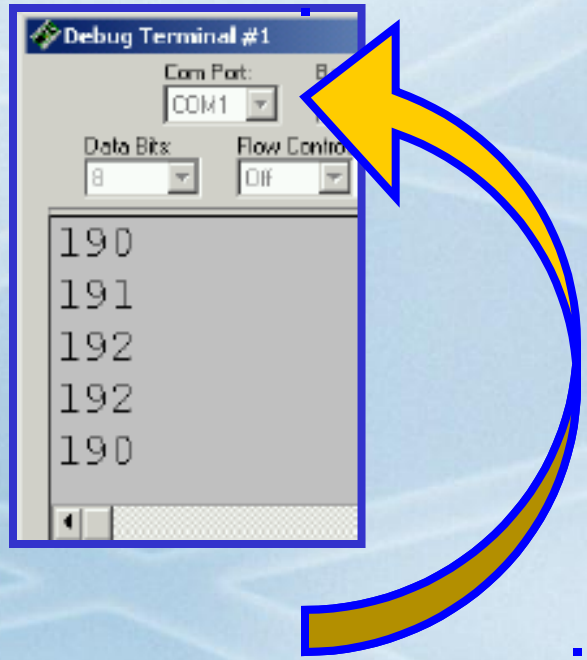
For example, to plot the value of Time, the code would be:

```
DEBUG DEC Time, CR
```

Stamp Plot may also be configured by sending control instructions instead of having to click the settings on the plot:

```
DEBUG "!AMAX 1250", CR,
      "!TMAX 25", CR,
      "!TMIN 0", CR,
      "!SHFT ON", CR,
      "!RSET",CR
```

- ✓Load and run PlotPhotoresistor.bs2
- ✓Verify the output in the DEBUG Window is a single column of values:



- ✓Note the COM port

- ✓ Open Stamp Plot Lite
  Menu→Program→Stamp Plot→ Stamp Plot Lite

- ✓ Change the COM port to match the one from the DEBUG Window.

- ✓ Close the DEBUG Window. *Only 1 applications can have access to the COM Port at anyone time.*

- ✓ Click **Connect** then **Plot Data** in Stamp Plot Li

- ✓ Press and release the **Reset button** on your HomeWork board to re-start your program to catch the DEBUG statements at the start of the program.
- ✓ The light level should now be plotting.
- ✓ Experiment with the Span and Time Span + and – buttons to adjust your plot.

Activity #3: Tracking Light Events

Programs remain in memory even when power is removed because it resides in special type of memory, EEPROM (E-E-PROM).

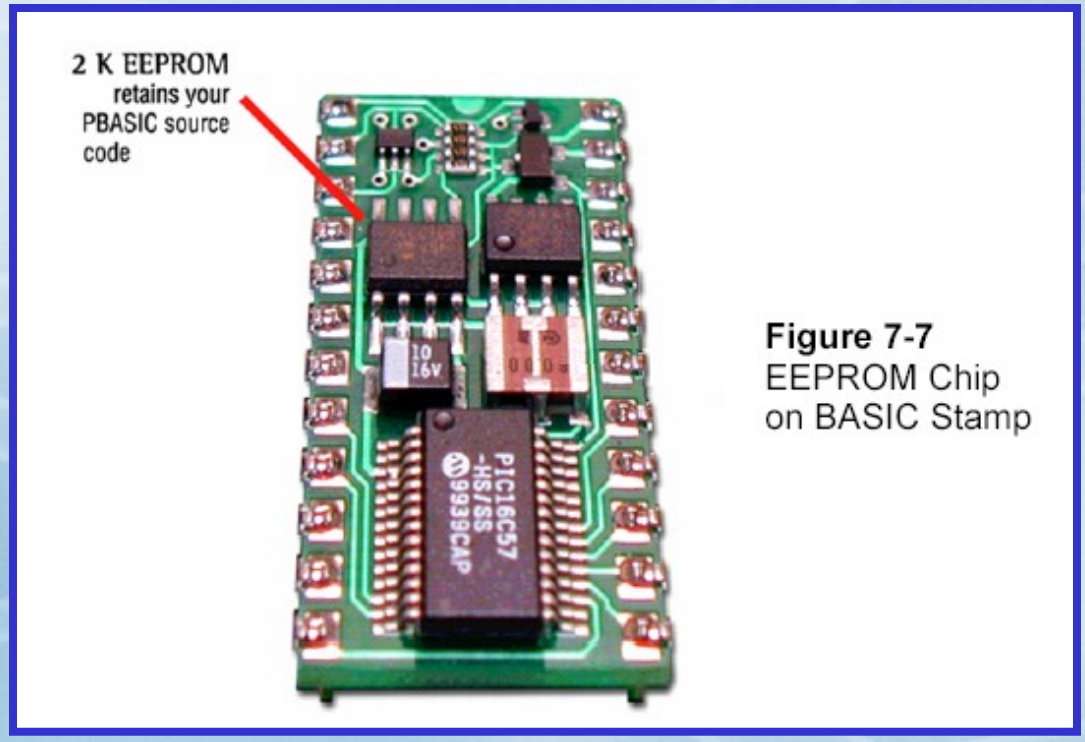


Figure 7-7
EEPROM Chip
on BASIC Stamp

The Memory Map (Click Run→Memory Map) shows RAM usage for variables and EEPROM usage for the current program. Notice that programs are stored in EEPROM from bottom-up.

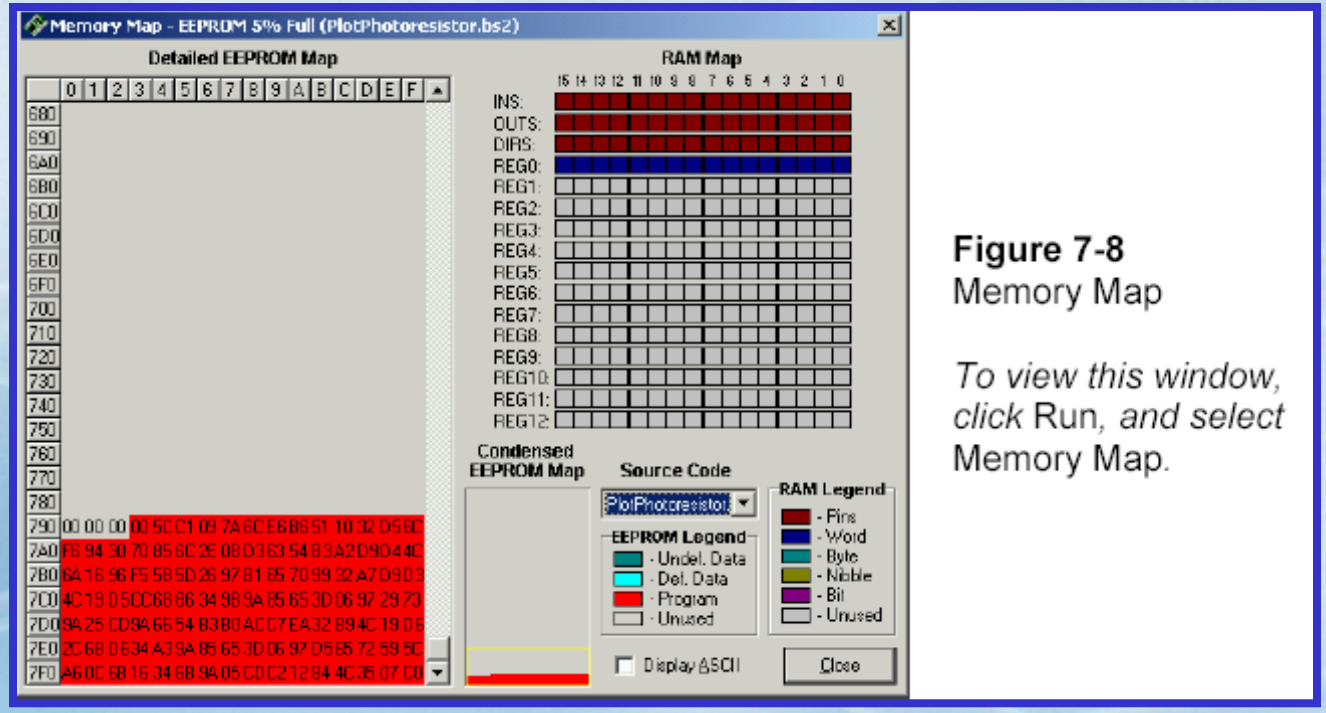


Figure 7-8
Memory Map

To view this window, click Run, *and select* Memory Map.

Writing to EEPROM

BASIC Stamp programs can also use the EEPROM for data storage.

WRITE Locations, {WORD} Data Item

For example, to store the value of 195 in address location 7:

WRITE 7, 195

To store values greater than 255, the Word parameter must be specified:

WRITE 8, Word 659

WRITE 10, Word 50012

Note that location 9 is skipped because using Word required 2 bytes to hold the value.

StoreLightMeasurementsInEeprom.bs2

The StoreLightMeasurementsInEeprom program stored 30 measurements of light into EEPROM (from locations 0 to 58 with each taking 2 bytes).

```
FOR eepromAddress = 0 TO 58 STEP 2
  HIGH 2
  PAUSE 5000
  RCTIME 2, 1, time
  DEBUG DEC2 eepromAddress,
        " ", DEC time, CR
  WRITE eepromAddress, Word time
NEXT
```

Reading From EEPROM

The READ instruction is used to read data from EEPROM:
READ Location, {Word} Data Item

- ✓ Read a byte and save in byte variable littleRR:
  READ 7, littleRR

- ✓ Read a word value and store in Word variable eepromValueA:
  READ 8, Word eepromValueA

ReadLightMeasurementsFromEeprom.bs2

The ReadLightMeasurementsFromEeprom will read and display the values from EEPROM

```
FOR eepromAddress = 0 TO 58 STEP 2
  READ eepromAddress, Word time
  DEBUG DEC2 eepromAddress, " ", DEC time, CR
NEXT
```

Activity #4: Simple Light Meter

Simple light meter uses the 7-segment LED to indicate the light strength.

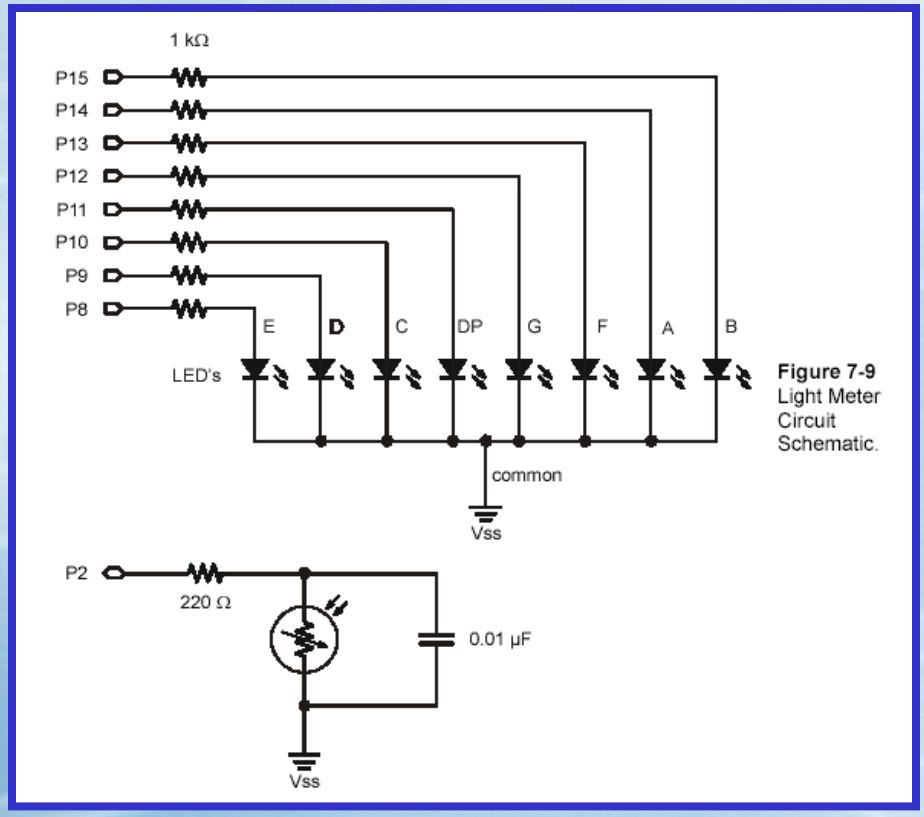
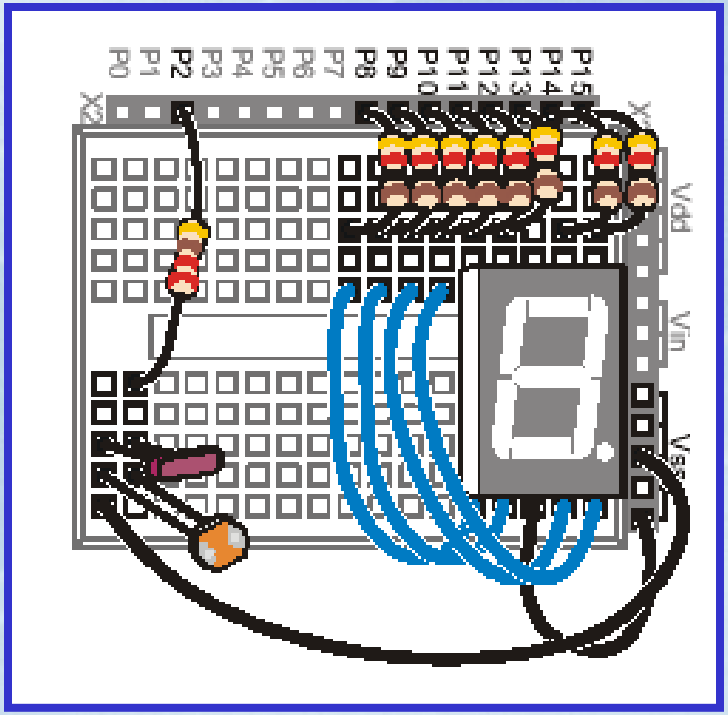


Figure 7-9 Light Meter Circuit Schematic.



Using GOSUB

A **GOSUB** is a branch to another section of the program defined by a label. Once complete, the execution returns to after the **GOSUB** call.

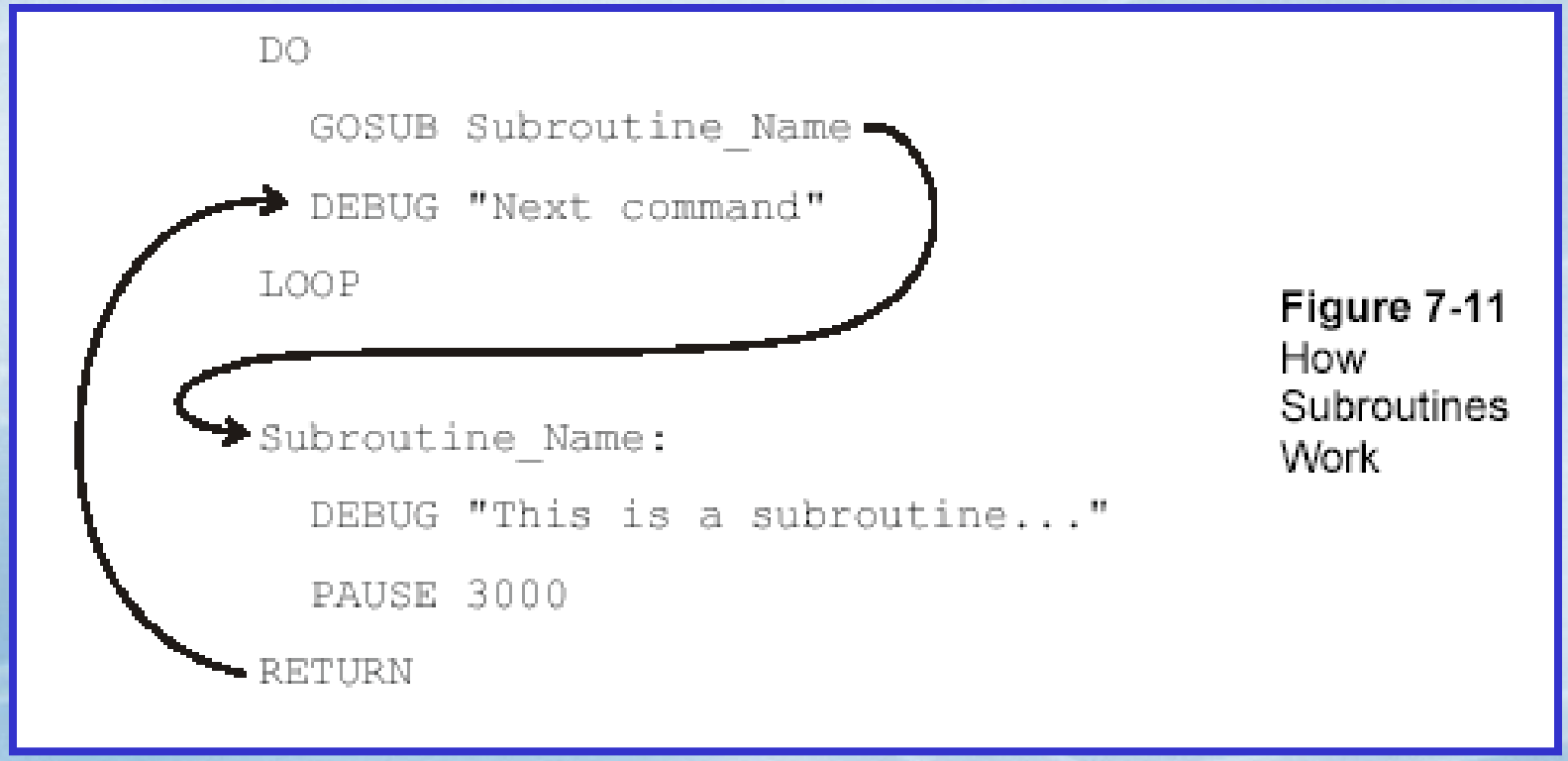


Figure 7-11 How Subroutines Work

All subroutines called with a **GOSUB** MUST end with a **RETURN**.

If GOSUBs and RETURNs are not properly matched, it can lead to improper operation of your BASIC Stamp program.

GOSUBs allow code to be re-used and makes coding cleaner by having the main DO...LOOP call subroutines to perform the various tasks of the program. From the main routine, you can get an idea of what the program will perform.

```
DO                                  ' Main routine.
   GOSUB Get_Rc_Time
   GOSUB Delay
   GOSUB Update_Display
LOOP
```

LightMeter.bs2 uses the principles of RCTIME with the Photoresistor and LOOKUP with the 7-segment display to create a light meter.

By using a PAUSE length defined by the time value, LED segments will cycle at a speed dependent on light level.

Chapter #6 Review

1. A photoresistor's _____ changes based on the amount of light.
2. The _____ command is used to send values to Stamp Plot for plotting.
3. Programs are stored to _____ memory.
4. The _____ command stores data in the EEPROM.
5. The _____ command reads data from EEPROM.
6. The _____ command causes execution to branch to a subroutine.
7. The _____ command must be used at the end of a subroutine.

Links

- ✓ 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