Robotic Sunflower Lesson 3: Creating a Light-Tracking Servo

AUTHOR: Pat Blount

DESCRIPTION: Student will learn how to program the Basic Stamp to use information from two photoresistors to point a servo at a light source. This will be the first degree of freedom for the flower head.

GRADE LEVEL(S): 9, 10, 11, 12

SUBJECT AREA(S): Electricity, electronics, computer science, applied physics

ACTIVITY LENGTH: 1 hour, 40 minutes

LEARNING GOAL(S):
Students will synthesize the previous lesson for light metering and servo control to design a servo controlled by two photoresistors that will track a light source.

STANDARDS MET:
Common Core:

- CCSS.ELA-Literacy.RST.11-12.9. Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.
- CCSS.ELA-Literacy.RST.9-10.3. Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
- CCSS.ELA-Literacy.RST.9-10.7. Translate quantitative or technical information expressed in words in a text into visual form (e.g., a table or chart) and translate information expressed visually or mathematically (e.g., in an equation) into words.

Next Generation Science Standards:

- HS-PS3-1. Create a computational model to calculate the change in the energy of one component in a system when the change in energy of the other component(s) and energy flows in and out of the system are known.
- HS-PS3-3. Design, build, and refine a device that works within given constraints to convert one form of energy into another form of energy.
- HS-ETS1-1. Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.

- HS-ETS1-2. Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- HS-ETS1-3. Evaluate a solution to a complex real-world problem-based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.

……

Student Background:

This lesson was developed as the third in a unit that culminates in the construction of a robotic sunflower that tracks the sun. My students will already have completed the following lessons beforehand:

- Robotic Sunflower Lesson 1: Measuring Voltage with a Microcontroller
- Robotic Sunflower Lesson 2: Controlling a Servo
- Robotic Sunflower Lesson 2.1: Extension to Controlling a Servo

Educator Background:

This lesson is a culmination of the previous lessons. There is no new instruction required to accomplish the design. Students will take the design from the day before with a photoresistor controlling a servo and add an additional photoresistor. Set up the new photoresistor the same as the first but on a different pin. At this point groups will have chosen different solution paths, some using the voltage divider circuit and some using the RC circuit. One item to consider is how to compare the values on the two photoresistors. A class discussion of the solution paths and their pros and cons should be held prior to starting the build. The instructor will need to become familiar with the following conditional statements:
IF...THEN
IF...THEN...ELSE
SELECT...CASE
Comparison Operators and Branching

Following is a sample program using the RC circuits to control the servos.

```
' {$STAMP BS2}
' {$PBASIC 2.5}


'servo control with dual photodiodes
'this program allows the user to control a servo by
'controling the light level on a pair of photodiodes.



'constants and pin assignments-----------------------------------------------------------------------------------

left_light PIN 14              'assigns the word "left_light" to a pin number. this allows the user to change the pin assignment
'for different circuit configurations without change multiple occurrences in the program.
right_light PIN 0              'left_light is the left photoresistor and right_light is the right.
LR_servo PIN 12                 'assigns the word LR_servo to a pin number. this allows the user to change the pin assignment
                                 'for the servo controlling Left/Right Motion for different circuit configurations without changing
                                 'multiple occurrences in the program.

differential CON 100            'the stepping angle for the servos
LR_servomin  CON 250            'min servo pulsout value, find these values with servo
LR_servomax  CON 1150           'max servo pulsout value
scale        CON 569            'scale factor for the rct times, determined by dividing 400 (servo max deviation from center)
                                      'by max sensor value and x 256 and rounding off

'variables-------------------------------------------------------------------------------------------------------

L_rct VAR Word                  'variable for the RCTIME count for photoresistor 1.is declared as a word, 0 to 65535, to handle
higher number counts
R_rct VAR Word                  'variable for the RCTIME count for photoresistor 2.is declared as a word, 0 to 65535, to handle
higher number counts
counter VAR Nib                 'makes the word "counter" a 4 bit (0 to 15) sized variable used to count a loop control.
position VAR Word               'makes the word "position" a 16 bit sized variable to represent the servo position from 200 to 1200

'initializations-------------------------------------------------------------------------------------------------

position = 700                  'sets the initial value of position
HIGH left_light                 'sets the pin to high, 5V, thus setting both plates of the cap to 5V. this allows
HIGH right_light                'the RCTIME command to time the fall of the voltage to 1.3V when the pin goes low.
FOR counter = 1 TO 10
  PULSOUT LR_servo,position     'sets LR_servo to a middle position at startup
  PAUSE 20
NEXT

'main program----------------------------------------------------------------------------------------------------

PAUSE 200                       'a short pause for startup time

DO

GOSUB lightlevel                'sends program control to measure the light level
DEBUG "original L_rct",TAB,"original R_rct",CR
DEBUG TAB,DECL_rct,TAB,TAB,DECR_rct,CR
IF (position <LR_servomin) OR (position >LR_servomax) THEN
   ' NAP 7
    GOSUB servoreset
    GOSUB moveservo
  ELSE
    GOSUB moveservo             'moves the servo based on the light level
ENDIF

PAUSE 3000
'NAP 6                          'places BS2 in low power mode (50 micro-amps vs 3 mA)for 1.152 secs, will replace w/ sleep

DEBUG "Left photo",TAB,"rightphoto",TAB,"position",CR
DEBUG DEC L_rct, TAB,TAB,DECR_rct,TAB,TAB,SDEC position
```

```
PAUSE 4000
DEBUG CLS
LOOP

'subroutines
==========================================================================================
======

'read light sensors-----------------------------------------------------------------------------------------

lightlevel:                         'subroutine for reading the light level

DO                                  'loop to verify that rctime interval count is not outside range of 65535
  RCTIME left_light,1,L_rct         'rctime arguments include pin number, initial state, and variable to store time.
  RCTIME right_light,1,R_rct        'initial state indicates wheather the capacitor starts high, 5V, or low, <1.3 V.
  HIGH left_light                   'cap has discharged so needs to be recharged
  HIGH right_light
LOOP UNTIL (L_rct<>0)AND (R_rct<> 0) 'makes sure that the sensors are not overloaded i.e. does not exceed 65535

RETURN                              'returns control of the program to the main loop

'move servo-------------------------------------------------------------------------------------------------

moveservo:                                          'subroutine for moving the servo

L_rct = (L_rct */ Scale) MAX 400  ' scale RCTIME to 0-400
R_rct = (R_rct */ Scale) MAX 400

position = L_rct - R_rct            ' calculate position as 700 (center servo value) plus or minus 400 (-400 to 400). this gives a
range from 300 to 1100

FOR counter = 1 TO 15
PULSOUT LR_servo, (700 + position)     ' move the servo
PAUSE 20
NEXT

RETURN

'servorest ------------------------------------------------------------------------------------------------

servoreset:

position = 700
FOR counter = 1 TO 10
  PULSOUT LR_servo, position
  PAUSE 20
NEXT

RETURN
```

Program 6: RC dual photoresistor servo control, sample a.

Program 6 calculates position by finding the difference between the two sensor values and then adjusting the servo accordingly. Another approach is illustrated in Program 7 in which the sensor values are placed into a range and then the servo is moved based on the range of values. Each program has its merits.

```
' {$STAMP BS2}
' {$PBASIC 2.5}


'servo control with dual photodiodes
'this program allows the user to control a servo by controlling the light level on a pair of photodiodes.

'constants and pin assignments-----------------------------------------------------------------------------------

sensor1 PIN 14                  'assigns the word "sensor1" to a pin number. this allows the user to change the pin assignment
'for different circuit configurations without change multiple occurrences in the program.
sensor2 PIN 0
servo1 PIN 12                   'assigns the word servo1 to a pin number. this allows the user to change the pin assignment
  'for different circuit configurations without change multiple occurrences in the program.
differential CON 10         'the stepping angle for the servos
servomin     CON 300      'min servo position
servomax     CON 1100   'max servo position

'variables-------------------------------------------------------------------------------------------------------

rct1 VAR Word                 'variable for the RCTIME count for photodiode 1. is declared as a word, 0 to 65535, to handle
                                 'higher number counts
rct2 VAR Word                 'variable for the RCTIME count for photodiode 2. is declared as a word, 0 to 65535, to handle
                                 'higher number counts
counter VAR Nib               'makes the word "counter" a 4 bit (0 to 15) sized variable used to count a loop control.
position1 VAR Word          'makes the word "position1" a 16 bit sized variable to convert rct1 to values that represent
                                 'ranges.
position2 VAR Word          'makes the word "position2" a 16 bit sized variable to convert rct2 to values that represent
                                 'ranges.
position VAR Word           'makes the word "position" a 16 bit sized variable to represent the servo position from 200 to
                                 '1200

'initializations-------------------------------------------------------------------------------------------------

position = 700                 'sets the initial value of position
HIGH sensor1                   'sets the pin to high, 5V, thus setting both plates of the cap to 5V. this allows
HIGH sensor2                   'the RCTIME command to time the fall of the voltage to 1.3V when the pin goes low.
FOR counter = 1 TO 10
  PULSOUT servo1,position       'sets servo1 to a middle position at startup
  PAUSE 20
NEXT

'main program----------------------------------------------------------------------------------------------------

DO

GOSUB lightlevel                    'sends program control to measure the light level
GOSUB moveservo                         'moves the servo based on the light level
NAP 6                          'places BS2 in low power mode (50 micro-amps vs 3 mA)for 1.152 secs, will replace w/ sleep
IF position >servomax OR position <servomin then
SLEEP  7000'places the BS2 in a low power mode for about 2 hours. this occurs when the servo max/mins out.
LOOP                           'which would happen at sunset or sunrise.

'subroutines
==========================================================================================================
======

'read light sensors----------------------------------------------------------------------------------------------

lightlevel:                    'subroutine for reading the light level

DO                             'loop to verify that rctime interval count is not outside range of 65535
  RCTIME sensor1,1,rct1              'rctime arguments include pin number, initial state, and variable to store time.
  RCTIME sensor2,1,rct2              'initial state indicates whether the capacitor starts high, 5V, or low, <1.3 V.
  HIGH sensor1                     'cap has discharged so needs to be recharged
  HIGH sensor2
LOOP UNTIL (rct1 <>0)AND (rct2 <> 0) 'makes sure that the sensors are not overloaded i.e. does not exceed 65535
```

```
RETURN                          'returns control of the program to the main loop

'move servo-----------------------------------------------------------------------------------------------------------

moveservo:                      'subroutine for moving the servo

SELECT rct1                      'divides the rct1 readings into 10 intervals and assigns a number based on
  CASE 0 TO 600                    'the range of values that rct1 falls into
    position1 = 0
  CASE 601 TO 1200
    position1 = 1
  CASE 1201 TO 1800
    position1 = 2
  CASE 1801 TO 2400
    position1 = 3
  CASE 2401 TO 3000
    position1 = 4
  CASE 3001 TO 3600
    position1 = 5
  CASE 3601 TO 4200
    position1 = 6
  CASE 4201 TO 4800
    position1 = 7
  CASE 4801 TO 5400
    position1 = 8
  CASE 5401 TO 6000
    position1 = 9
ENDSELECT

SELECT rct2
  CASE 0 TO 600
    position2 = 0
  CASE 601 TO 1200
    position2 = 1
  CASE 1201 TO 1800
    position2 = 2
  CASE 1801 TO 2400
    position2 = 3
  CASE 2401 TO 3000
    position2 = 4
  CASE 3001 TO 3600
    position2 = 5
  CASE 3601 TO 4200
    position2 = 6
  CASE 4201 TO 4800
    position2 = 7
  CASE 4801 TO 5400
    position2 = 8
  CASE 5401 TO 6000
    position2 = 9
ENDSELECT

IF position1 < position2 THEN              'decides if rct1 or rct2 is larger then increases or decreases the position
position = position + differential      'of the servo by the deferential.
    FOR counter = 1 TO 10
      PULSOUT servo1,position
      PAUSE 20
    NEXT
ELSEIF position1 > position2 THEN
position = position - differential
    FOR counter = 1 TO 10
      PULSOUT servo1,position
      PAUSE 20
    NEXT
ENDIF

DEBUG CLS, "rct1", TAB,"rct2",TAB,"position1",TAB,"position2",TAB,"position",CR              'Displays the rct1 , rct2, and the
turn                                                                                        'values for adjustments
```

```
DEBUG DEC rct1,TAB, DEC rct2,TAB,TAB,DEC position1,TAB,TAB,DEC position2,TAB, DEC position

RETURN
```

Program 7: RC dual photoresistor servo control, sample b.

In order to find the range, program 7 uses the SELECT…CASE command structure. This will take the rctime values and find the range in which it falls. 10 ranges were used to make the division easy. A higher resolution can be achieved with more ranges. Values for the rctime should be measured for the site location so appropriate ranges can be set.

Science Kit Materials List:

- Multimeter (1 per group)
- LED (1 per group)

Other Materials List:

- "Robotic Sunflower Lesson 3: Student Guide"
- Photoresistor (2 per group)
- 220 Ohm resistor (2 per group)
- 440 Ohm resistor (2 per group)
- 2k Ohm resistor (2 per group)
- 10K Ohm resistor (2 per group)
- 0.01 uF Capacitor (2 per group)
- 5k Ohm potentiometer (2 per group)
- Basic Stamp Homework Board (1 per group)
- Jumper wires (1 set per group)

Vocabulary:

- IF…THEN
- IF…THEN…ELSEIF
- SELECT…CASE
- Conditional statements
- Branching

……

BONNEVILLE ENVIRONMENTAL FOUNDATION
240 SW 1st Avenue
Portland OR 97204
503-248-1905
www.b-e-f.org

Lesson Details:

Planning and Prep

Ensure that there are enough materials for each group. In addition, students should be able to access a computer. If not already there, download pBasic 2.5 onto the computers for programming the Basic Stamps. It's a free program available at www.parallax.com.

Class Sequence

- Take about 30 minutes and introduce students to IF…THEN…ELSE, conditional statements and SELECT…CASE control structures if needed.
- Challenge the students to come up with a program that uses the information from a pair of photoresistors to control the direction a servo turns. The design constraints would be to have the servo move right when the right photoresistor is more illuminated, likewise left, and stay in its current position when the photoresistor are equally illuminated.
- At the end of the day team solutions should be able to track a high intensity light source.

Assessment

Use the rubric in Table 1 to assess student work.

Program	Max Pts	Pts Earned	Grading Criteria		Instructor Initial
Servo Control w/ 2 photo	3		Circuit is wired correctly. Program entered correctly, downloads and runs as specified.	___/1 ___/2	
Document	2		Documentation and comments.	___/2	
Teamwork	2		Worked well and contributed.	___/2	
Total:					

Table 1: Rubric for servo control with two photoresistors.

BONNEVILLE ENVIRONMENTAL FOUNDATION
240 SW 1st Avenue
Portland OR 97204
503-248-1905
www.b-e-f.org