Integrating the BASIC Stamp BS2 with Range Finding Sensors for Trackless Navigation

ELEC471
Spring 2009

Max Nielsen
Dan Brosnan
Dana Howes
April 29, 2009
This is a program for the BASIC stamp BS2 to control two infrared distance measurement sensors, an ultrasonic distance measuring sensor (Parallax, ‘PING)))’ sensor), and an integrated bumper sensor, with the code to program a motor controller to control two individual motors for obstacle avoidance in a robotics application. The program for this is written, but will have explanations in between the code. The actual code will be highlighted below. A diagram of the obstacle avoidance can be seen below in Figure 1.

**Figure 1 – Obstacle Avoidance Diagram**
A photo of an application can be seen in Figure 2.

The first part of this program defines the basic stamp module being used, and the language that is being used. This is identified by the compiler, and helps to define the functions.

`[STAMP BS2]
[SPBASIC 2.5]

The next part is a simple description of what the program does. Anything after the apostrophe, "' " mark will be recognized as a comment.

'-----[ Program Description ]---------------------
' This program demonstrates reading the distance in centimeters from the Sharp GP2D12 Analog Distance Sensor. It also combines this with the

' PING)) Ultrasonic sensor from Parallax, and displays all values in both inches and cm.

The next part of the program sets the I/O Definitions. It defines the variables to what I/O pin they are plugged into. Again, to the right after the apostrophe is a comment with a description of what each definition does. There is a repeat, one for the left infrared sensor, one is for the right infrared sensor.

'-----[ I/O Definitions ]---------------------
Adc0831R PIN 0   ' ADC0831 Chip Select (ADC0831.1)
AdcClockR PIN 1   ' ADC0831 Clock (ADC0831.7)
AdcDataR PIN 2   ' ADC0831 Data (ADC0831.6)
Adc0831L PIN 3   ' ADC0831 Chip Select (ADC0831.1)
AdcClockL PIN 4   ' ADC0831 Clock (ADC0831.7)
AdcDataL PIN 5   ' ADC0831 Data (ADC0831.6)

To see how the infrared sensors work and how they are connected, see Appendix A.

The following is the variable definitions and the constants used to make conversions and calculations from the analog IR sensors.

'-----[ Constants ]---------------------
span CON 5 ' 5 cm Per Data Point

'-----[ Variables ]---------------------
resultR VAR Byte   ' ADC8031 Result
voltsR VAR Word   ' Volts (0.01 Increments)
Since the IR sensors are analog sensors, they need to be connected to an Analog-to-Digital converter (ADC) to be used with the digital input on the BASIC stamp microcontroller. The next part is a data table that is established to reference from the analog IR sensors.

The ADC’s need to be activated by setting one of the pins high, which pins are defined at the beginning of the program.
Here is the actual programming code. This is what the microcontroller will run when it is turned on. At this point, it will loop until the bumper sensor is hit, in which it will back up and turn off. The comments to the right explain each line of the code.

'----[ Program Code ]----------------------------------------
DO
  GOSUB Ultrasonic  'runs the Ultrasonic subroutine
  DEBUG HOME, "Ultrasonic Distance = ", DEC3 cmDistance, " cm"
  DEBUG CR, "Ultrasonic Distance = ", DEC3 inDistance, " in"
  GOSUB Read_GP2D12R  'Read Right IR Sensor Value
  GOSUB Calculate_DistanceR   'Convert Value To cm
  DEBUG CR, "Left IR Distance = ", DEC cmR, " cm"
  GOSUB Read_GP2D12L  'Read Left IR Sensor Value
  GOSUB Calculate_DistanceL  'Convert Value To cm
  DEBUG CR, "Right IR Distance = ", DEC cmL, " cm"
  DEBUG CR, ? IN8  'display left IR distance
  DEBUG "YES"
  DEBUG "NO"
  IF(IN8 = 1)THEN   'if the push button is pressed
    speed = 0
    SEROUT 14,84,[80,0,3,speed]  'stop motor 1
    SEROUT 14,84,[80,0,0,speed]  'stop motor 0
    PAUSE 500
    SEROUT 14,84,[80,0,1,40]  'motor 0 backwards
    SEROUT 14,84,[80,0,2,40]  'motor 1 backwards
    PAUSE 3000
    SEROUT 14,84,[80,0,1,0]  'motors stopped
    SEROUT 14,84,[80,0,2,0]
  ENDIF
  IF(inDistance <= 30) THEN     'if the ultrasonic data sees less than or equal to 30"s, turn right
    GOSUB turnright
  ELSEIF (cmL <= 20 )THEN      'if the Left IR sensor reads less than or equal to 20cm, turn right
    GOSUB turnright
  ELSEIF (cmR <= 20 )THEN      'if the right IR sensor reads less than or equal to 20cm, turn left
    GOSUB turnleft
  ELSE
    SEROUT 14,84,[80,0,3,55]   'if none of these conditions, move straight forward
    SEROUT 14,84,[80,0,0,40]
  ENDIF
LOOP    'loop the program
END
In the program, the GOSUB routine runs the subroutine following the command. The subroutines are defined below.

'----[ Subroutines ]--------------------------------------------

Ultrasonic:  
- PULSOUT 6, 5  
- PULSIN 6, 1, time  
- cmDistance = cmConstant ** time  
- inDistance = inConstant ** time

RETURN

Read_GP2D12R:  
- voltsR = 0  
- FOR indexR = 0 TO 2  
-  
- LOW Adc0831R  
-  
- SHIFTIN AdcDataR, AdcClockR, MSBPOST, [resultR\9]  
-  
- HIGH Adc0831R  
-  
- voltsR = voltsR + resultR  
- PAUSE 30  
- NEXT  
- voltsR = voltsR / 3  
- RETURN

Calculate_DistanceR:  
- FOR indexR = 0 TO 15  
- READ (VoutR + indexR), test2R  
- IF (test2R <= voltsR) THEN EXIT  
- NEXT  
- SELECT indexR  
- CASE 0  
- cmR = 10  
- CASE 1 TO 14  
- cmR = 10 + (5 * indexR)  
- IF (test2R < voltsR) THEN  
- READ (VoutR + indexR - 1), test1R  
-  
- slopeR = (test1R - test2R) * 10 / span  
-  
- cmR = cmR - ((voltsR - test2R) * 10 / slopeR)  
-  
- inR = (cmR ** 25801)  
- ENDIF  
- CASE 15  
- cmR = 80  
- ENDSELECT  
- RETURN

Read_GP2D12L:  
- voltsL = 0  
- RETURN
FOR indexL = 0 TO 2
  ' Read 3 Times
  LOW Adc0831L
  ' Enable ADC0831
  SHIFTIN AdcDataL, AdcClockL, MSBPOST, [resultL\9]
  ' Read The Voltage
  HIGH Adc0831L
  ' Disable ADC0831
  voltsL = voltsL + resultL
  ' Add The Values
  PAUSE 30
NEXT
voltsL = voltsL / 3
  ' Average The Readings
RETURN

Calculate_DistanceL:
FOR indexL = 0 TO 15
  ' Search DATA Table For Value
  READ (VoutL + indexL), test2L
  ' Get Value From DATA Table
  IF (test2L <= voltsL) THEN EXIT
  ' Found Value
NEXT
SELECT indexL
CASE 0
  cmL = 10
  ' Set To Minimum Distance
CASE 1 TO 14
  cmL = 10 + (5 * indexL)
  ' Calculate Distance
  IF (test2L < voltsL) THEN
    ' Estimate Using Interpolation
    READ (VoutL + indexL - 1), test1L
    slopeL = (test1L - test2L) * 10 / span
    ' Calculate Slope
    cmL = cmL - ((voltsL - test2L) * 10 / slopeL)
    inLE = (cmL ** 25801)
  ENDIF
CASE 15
  cmL = 80
  ' Set To Maximum Distance
ENDSELECT
RETURN
The following subroutines are to tell the motors which direction to turn.
During configuration, you should connect just one motor controller at a time to your serial line unless you want to configure each motor controller the exact same way. The default configuration is for two-motor control with motor numbers 2 and 3.

If the motor supply is connected, the motors will be pulsed as the LEDs flash, so it is recommended that the motors or the motor supply be disconnected prior to configuration.

Configuration is achieved by sending a three-byte packet consisting of the start byte, a configuration command byte, and the new configuration byte:

| start byte = 0x80 | change configuration = 0x02 | new settings, 0x00-0x7F |

The new settings byte contains two parts: a six-bit motor number and a one-bit flag specifying one-motor or two-motor mode.

- Bits 0-5 specify the motor number(s) to which the motor controller will respond. In single-motor mode, the number you choose sets the number to which the motor controller will respond. In two-motor mode, the motor controller will respond to two consecutive numbers. If you set an even motor number, the motor controller will control that motor number and the one above it; if you set an odd motor number, the motor controller will control that motor number and the one below it. Note that all motor controllers will respond to motor number 0 (and 1, if in two-motor mode).

- Bit 6 specifies whether the motor controller is in one-motor mode or in two-motor mode. If this bit is clear, the motor controller will be in two-motor mode; if the bit is set, the motor controller will be in 1-motor mode.

After sending the change configuration command, the motor controller will flash the red LED for one-motor mode and the green LED for two-motor mode. For instance, if you configured your motor controller to control one motor, motor number 3, the red LED should flash 4 times. (The reason for the extra flash is so that you get some response if you set the motor number to zero). After configuration, the motor controller must be reset (either by turning it off and back on or by using the reset line) before you can continue using it.
A typical two-motor setup.

turnright:
  SEROUT 14,84,[$80,0,1,40]
  SEROUT 14,84,[$80,0,3,40]
  RETURN

turnleft:
  SEROUT 14,84,[$80,0,0,40]
  SEROUT 14,84,[$80,0,2,40]
  RETURN
Sharp GP2D12 Analog Distance Sensor (#605-00003)

General Description

The Sharp GP2D12 is an analog distance sensor that uses infrared to detect an object between 10 cm and 80 cm away. The GP2D12 provides a non-linear voltage output in relation to the distance an object is from the sensor and interfaces easily using any analog to digital converter.

Features

- High immunity to ambient light and color of object
- No external control circuitry required
- Sensor includes convenient mounting holes
- Compatible with all BASIC Stamp® and SX microcontrollers

Application Ideas

- Robot range finder
- Halloween prop activation

Quick Start Circuit Connecting and Testing

Connect the GP2D12 to your analog to digital converter as shown in the circuit on the previous page. The potentiometer connected to the Vref pin on the ADC0831 is being used as a voltage divider to set the reference voltage to 2.55 volts. On the ADC0831 this will give a value of 0 to 255 for an input voltage of 0 to 2.55 volts. This gives us a resolution of 0.01 volts per step from the ADC. If you are using a different analog to digital converter, you may want to adjust the potentiometer to get the best results from your particular ADC.
Calibration

Because the output of the GP2D12 is not linear, we need a way to determine what distances correspond to what voltages. One way of calibrating your sensor is by measuring the voltage output of the GP2D12 at given fixed distances, in centimeters, as shown in the chart below. Once you have this information you can plug these numbers into the EEPROM DATA statements in the program. The table of data is used by a routine in the program to calculate the distances, which are then displayed on the Debug Terminal, along with the voltage output from the sensor.

![GP2D12 (voltage vs distance)](image)

Sensitivity

The usable range of the GP2D12 is between 10 cm and 80 cm. The readings for objects closer than 10 cm are unstable and therefore not usable.

Resources and Downloads

Check out the Sharp GP2D12 Analog Distance Sensor product page for example programs, the manufacturer datasheet and more:


Specifications
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Quantity</th>
<th>Minimum</th>
<th>Typical</th>
<th>Maximum</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vcc</td>
<td>Supply Voltage †</td>
<td>4.5</td>
<td>5.0</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>Topr</td>
<td>Operating Temperature †</td>
<td>-10</td>
<td>-</td>
<td>+60</td>
<td>°C</td>
</tr>
<tr>
<td>Tstg</td>
<td>Storage Temperature †</td>
<td>-40</td>
<td>-</td>
<td>+70</td>
<td>°C</td>
</tr>
<tr>
<td>ΔL</td>
<td>Distance Measuring Range †</td>
<td>10</td>
<td>-</td>
<td>80</td>
<td>cm</td>
</tr>
<tr>
<td>Vo</td>
<td>Output Terminal Voltage (L=80 cm) †</td>
<td>0.25</td>
<td>0.4</td>
<td>0.55</td>
<td>V</td>
</tr>
<tr>
<td>ΔVo</td>
<td>Output change at L=80 cm to 10 cm †</td>
<td>1.75</td>
<td>2.0</td>
<td>2.25</td>
<td>V</td>
</tr>
<tr>
<td>Icc</td>
<td>Average Dissipation Current (L=80cm) †</td>
<td>-</td>
<td>33</td>
<td>50</td>
<td>mA</td>
</tr>
</tbody>
</table>

† data obtained from Sharp’s GP2D12 datasheet

### Pin Definitions and Ratings

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vo</td>
<td>Voltage Output</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>3</td>
<td>Vcc</td>
<td>Supply Voltage</td>
</tr>
</tbody>
</table>

*All units in millimeters (mm)*