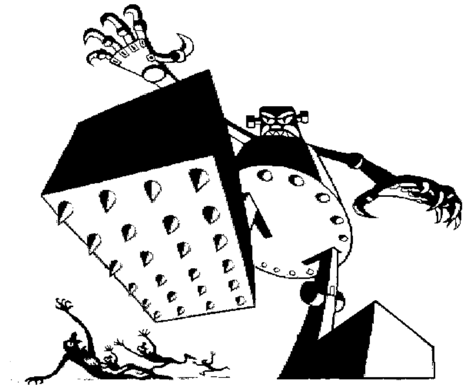
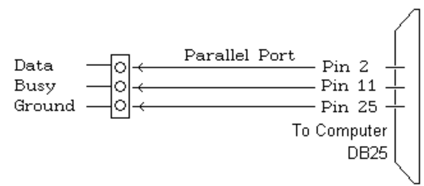
P.A.R.T.S

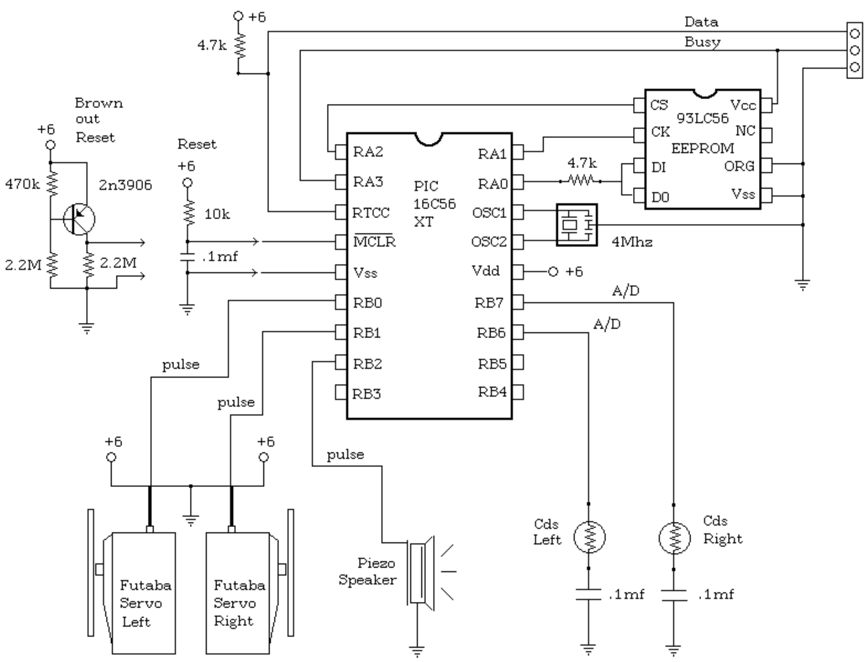
Marvin Green marvin@agora.rdrop.com
821 SW 14th St. Troutdale OR 97060 (503) 666-5907
Issue # 13 Portland Area Robotics



Simply Basic Robots.

Last issue I made a comparison between the 68HC11 and the Basic Stamp chip set. From the comparison I believe the Basic Stamp shows promise for small robotic applications. With this chip set I decided to build a simple light following robot. I wanted to make it easy to build and easy to program. As you see from the schematic, the entire robot uses only two chip. One is the BASIC Stamp interpreter, and the other is the EEPROM chip. The robot is programed from a simple connection to a PC printer port.





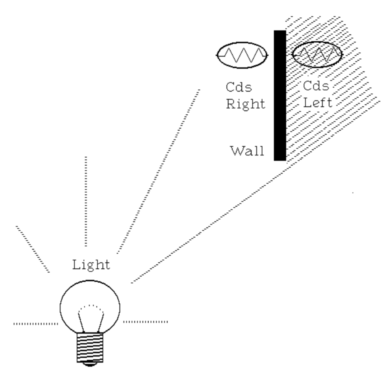
I will not go into detail about the Basic Stamp functions, there are several magazine articles that do that. * However, the BASIC Stamp is a very clever design and makes excellent use of it's limited resources.

I built this robotic circuit on a piece of prototyping board that was 2x3 inches. Then I added connectors for the power supply, servos, and accessories. The construction time took about two hours. I then mounted the circuit board onto on of my B-BOT frames and used two Futaba S148 servos. Modifying the servo and building the frame took about 35 minutes.

Next I connected two Cds cells and played with some BASIC code to read the sensors. I was surprised at how easy the code was to write. The BASIC Stamp chip set has very power commands to read its I/O pins. One of the commands lets you read variable resistors, such as Cds cells.

The Cds cells I used came from Electronic Goldmine [G5818] (602) 451-7454, but the Radio Shack [276-1657] ones should work as well.

I then mounted a bridge between the two Cds cells to cast a shadow upon the opposite cell when a bright light shines at an angle. Just for fun I then added a piezo electric speaker [RS 273-060] for simple sound effects.



The Program.

```
'**********************************************
'  b0 =  variable for right servo   pin = 0
'  b1 =  variable for left servo    pin = 1
'        Piezo speaker               pin = 2
'
'  b6 =  variable for right eye      pin = 6
'  b7 =  variable for left eye       pin = 7
'
'**********************************************
Symbol  counter = b4    ' Counter
Symbol  Reye    = b6    ' Right Eye
Symbol  Leye    = b7    ' Left Eye
Symbol  LwSig   = b3    ' Low Signal of light.
'**********************************************
Init:
        pins  = 0       ' Initialize.
        dirs  = %00001111
        LwSig = 255
        gosub Start

' PART 1 ******  Find Bright Spot ************
'        ******* Then Point there  ************

FindHigh:
      gosub spin                 ' Normal fwd speed
      gosub ReadE                ' Read eyes
      counter = counter + 1      ' Inc. counter
      if counter > 90 then Find  ' Complete 1 spin?
      if Reye = LwSig then MoreLt ' At bright spot?
      goto ContFnd
MoreLt:
      LwSig = Reye     ' Save location of high
      goto Continue
Find:
      if Reye = LwSig then Follow ' At Bright Spot?
      goto Continue
ContFnd:
      gosub Pulse
      goto FindHigh

'PART 2 *********  Follow Light  ******************

Follow:
      gosub ReadE               ' Read eyes
      gosub Forward             ' Normal fwd speed
      if Reye > Leye then TrnRt
      if Leye > Reye then TrnLt
      goto Continue             ' If eyes are equal
TrnRt:
      gosub Right       ' Cut right servo speed
      goto Continue
TrnLt:
      gosub Left        ' Cut left servo speed
      goto Continue
Continue:
      gosub Pulse
      goto Follow

'*********    SUBROUTINES   **************

ReadE:
      pot 6,255,b6       ' Read right CDS cell
      pot 7,200,b7       ' Read left CDS cell
      b6 = b6 / 16       ' Make it smaller
      bb7 = b7 / 16
      return
Pulse:
      b0 = 125+b0       ' Adjust for + = forward.
      b1 = 128-b1       ' Adjust for + = forward

       pulsout 0,b0     ' right servo
       pulsout 1,b1     ' left servo
       return
Forward:
        b0 = 50          ' Normal forward speed
        b1 = 50          ' Normal forward speed
        return
Spin:
        b0 = 10          ' Spin the bot
        b1 = -10         ' Spin the bot
        return
Right:
        b0 = 2           ' Slow to turn right
        return
Left:
       b1 = 2           ' Slow to turn left
       return
Sing:
      sound 2,(60,15,75,15,90,15,60,30)
      return             ' Make a sound
Start:
      gosub sing         ' Sing a song
      gosub sing
      gosub sing
      pause 1000
      return
```

As you can see this code is broken into two parts. The first part of the program spins the robot 180 degrees , takeing a sensor reading each step. The program remebers the location of the brightest spot. After the first full spin, the robot, the robot continues to spin until it reaches the location of the brightest spot.

The second part of the code allows the robot to follow the sensor with the brightest light on it. If the two sensors are equal, then the robot just goes forward. It may surprise you that this code compiles down to about 100 bytes of code space in the BASIC Stamp.

* Take a look at the Parallax BASIC Stamp for $39. (916) 624-8333. Their BBS has lots of good information at (916) 624-7101. Nuts and Volts Magazine issues May and June 1994 has good articles on constructing the Faux Stamp, a BASIC Stamp clone. Contact Scott Edwards at (602) 459-4802 or 72037.2612@compuserve.com. Also check out the January/February 1994 issue of Micro Computer Journal.

Computer chips are getting smarter, faster and easier to use, not to mention cheaper. This is all great news for people who like to build robots. Happy building.