

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

```
' -- Declarations
```

```
counter VAR Byte
```

```
l_values VAR Byte
```

```
r_values VAR Byte
```

```
m_values VAR Byte
```

```
IR_freq VAR Word
```

```
n VAR Word
```

```
x VAR Word
```

```
metal VAR Byte
```

```
TxPin CON 10
```

```
Baud19200 CON 32
```

```
rect VAR Word
```

```
'--Initialization
```

```
'n = 0
```

```
OUTPUT 0
```

```
OUTPUT 2
```

```
OUTPUT 4
```

```
INPUT 8
```

```
OUTPUT 9
```

main:

l\_values = 0

r\_values = 0

m\_values = 0

GOSUB IR\_navigation

' Forward navigation routine - if nothing is detected

forward:

PULSOUT 6, 850

PULSOUT 7, 600

' Keep the robot in our range

n = n + 1

IF n > 200 THEN

n = 0

IF NCD(m\_values) > 0 THEN main

GOSUB half\_turn

FOR x = 1 TO 50

PULSOUT 6, 850

PULSOUT 7, 645

```
    GOSUB IR_navigation
NEXT
    GOSUB half_turn
ENDIF
GOTO main
```

```
' -----Navigation routines-----
```

```
' Turn left by one pulse
```

```
left_turn:
```

```
    PULSOUT 6, 640
```

```
    PULSOUT 7, 645
```

```
GOTO main
```

```
' Turn right by one pulse
```

```
right_turn:
```

```
    PULSOUT 6, 800
```

```
    PULSOUT 7, 865
```

```
GOTO main
```

```
' Stop moving
```

```
dontmove:
```

metal = 0

n = 0 ' Reset counter

' Move towards the object a little more

FOR x = 1 TO 100

PULSOUT 6, 785

PULSOUT 7, 740

PAUSE 20

metal = metal + IN8 ' Check the material while moving towards object

NEXT

' Stop after getting close enough

PULSOUT 6, 760 'left

PULSOUT 7, 762 'right

' Detect the objects material

'DEBUG ? IN8

metal = metal + IN8

DEBUG ? metal

IF metal > 50 THEN GOTO nonmetal

IF metal < 80 THEN GOTO tilt

tilt:

```
FREQOUT 9, 2000, 4000 ' Beep if it senses metal
loop1:
  PULSOUT 11, 740
  GOSUB rctimeloop

  IF rct < 65535 THEN GOTO checkmetal

  GOTO loop1

checkmetal:

  IF rct < 30000 THEN GOSUB aluminum
  IF rct >30000 THEN GOSUB copper

tiltover:
  FOR x=1 TO 100
  PULSOUT 11, 800
  NEXT

nonmetal:
  FREQOUT 9, 2000, 100 ' Beep twice for non-metals

  GOSUB turn_away

GOTO main

' Move away from objects that have already been detected

turn_away:

  GOSUB backward

  FOR x = 1 TO 50

    PULSOUT 6, 800

    PULSOUT 7, 865

    PAUSE 20

  NEXT

RETURN
```

' Back up a small amount

backward:

FOR x = 1 TO 15

PULSOUT 6, 640

PULSOUT 7, 865

PAUSE 20

NEXT

RETURN

' Make a small turn to allow robot

' to come back at an angle

half\_turn:

FOR x = 1 TO 25

PULSOUT 6, 800

PULSOUT 7, 865

PAUSE 20

NEXT

RETURN

' -----IR Navigation Subroutine-----

IR\_navigation:

```

' Zoning: Emit different frequencies to detect
'     how far away the object is
FOR counter = 0 TO 4
    LOOKUP counter,[37500,38250,39500,40500,41500],IR_freq

    FREQOUT 2, 1, IR_freq      ' Middle sensor detection
    m_values.LOWBIT(counter) = ~IN3

    FREQOUT 0, 1, IR_freq      ' Right sensor detection
    r_values.LOWBIT(counter) = ~IN1

    FREQOUT 4, 1, IR_freq      ' Left sensor detection
    l_values.LOWBIT(counter) = ~IN5
NEXT

' Stop if it's close to the middle, or turn a pulse
' in the direction of the object
IF NCD(m_values) = 3 THEN dontmove
IF NCD(l_values) > 0 THEN left_turn
IF NCD(r_values) > 0 THEN right_turn

RETURN

rctimeloop:

```

```
LOW 12
RCTIME 12, 0, rct
```

```
rct = rct - 1
```

```
DEBUG DEC rct, TAB,
  REP "*" \NCD rct, CR
```

```
PAUSE 1000
RETURN
```

```
aluminum:
```

```
HIGH TxPin ' Set pin high to be a serial port
PAUSE 100 ' Pause for Serial LCD to initialize
SEROUT TxPin, Baud19200, ["Aluminum"]
```

```
FREQOUT 9, 2000, 4000
RETURN
```

```
copper:
```

```
FREQOUT 9, 1000, 5000
```

```
HIGH TxPin ' Set pin high to be a serial port
PAUSE 100 ' Pause for Serial LCD to initialize
SEROUT TxPin, Baud19200, ["Copper"]
RETURN
```