

' PROGRAM: ROBOT.bas

' First Coding 04/05/98

'Steering Servo Definitions

Symbol SERVO_STEER = 0 ' Defines Steering Servo (pin 0)
Symbol LEFT_90 = 30 ' Steer 90 degrees left
Symbol RIGHT_90 = 216 ' Steer 90 degrees right
Symbol LEFT_45 = 75 ' Steer 45 degrees left
Symbol RIGHT_45 = 168 ' Steer 45 degrees right
Symbol STRAIGHT = 121 ' Steer Straight
Symbol RIGHT_0 = STRAIGHT ' Ditto
Symbol LEFT_0 = RIGHT_0 ' Ditto
Symbol STEER_ANGLE = b2 ' Variable that keeps steering command
Symbol LAST_STEER = b3 ' Last command sent
' (sent to steering subroutine)

'Detector Led Definitions

Symbol LEFT_LED = pin5 ' Defines the left detector
Symbol RIGHT_LED = pin7 ' Defines the right detector
Symbol CENTER_LED = pin6 ' Defines the center detector
Symbol ON_TABLE = 0 ' A detector that says this is ON
' the table
Symbol OFF_TABLE = 1 ' Opposite of above
Symbol LED_MASK = %11100000 ' Bits on LED port

'Drive Motor Definitions

Symbol SERVO_DRIVE = 1 ' Defines the output for this servo (pin 1)
Symbol FWD_FAST = 200 ' Drive fast fwd
Symbol FWD_SLOW = 178 ' Drive fwd slowly
Symbol REV_FAST = 30 ' Drive backwards FAST
Symbol REV_SLOW = 170 ' Drive backwards SLOW
Symbol STOP_HARD = 173 ' Stop with wheels locked (865?)
Symbol STOP_SOFT = 0 ' Stop w/o power
Symbol DRIVE_SPEED = b4 ' Variable that holds the drive speed
Symbol LAST_DRIVE = b5 ' Last command sent

'Main Program

MAIN:

Symbol counter = w3 ' Set up a counter for general stuff
Symbol register = w4 ' Used as a general register
Symbol leds = b0 ' Stores the value of the leds

' shifted over to bit 0

INIT:

```
DIRS          = %00011111      ' Sets LEDS ad inputs, all else
                                ' as outputs
Low SERVO_STEER      ' Sets steering servo pin low
Low SERVO_DRIVE     ' Dito for the drive servo
STEER_ANGLE        = STRAIGHT   ' Point this puppy straight
```

'-----
' QUICK ABORT - Stop Quickly!
'-----

QUICK_ABORT:

```
DRIVE_SPEED = STOP_HARD      ' Set-up command to STOP
for counter  = 1 to 75       ' Send it enough time to make
    gosub UPDATE_SERVOS     ' it work!
next
```

' NOW FALL INTO CHECK_LEDS
'
'-----

' CHECK_LEDS - Check the state of the

' LEDS and decide what to do.

' LEFT CENTER RIGHT ACTION

```
'   -----
'   ON  ON  ON    STRAIGHT
'   ON  ON  OFF   LEFT_90_TURN
'   ON  OFF ON    LEFT_90_TURN
'   ON  OFF OFF   LEFT_90_TURN
'   OFF ON  ON    RIGHT_90_TURN
'   OFF ON  OFF   BACK_UP & RIGHT_45_TURN
'   OFF OFF ON    RIGHT_90_TURN
'   OFF OFF OFF   BACK_UP & LEFT_45_TURN
```

'-----
' CHECK_LEDS:

```
    gosub GET_LEDS          ' Get current value of leds
    branch leds,(STRA,190t,190t,190t,r90t,bl45t,r90t,bl45t)
end
```

'-----
' GET_LEDS - Gets the value of the

' LEDS and returns it in the LEDS

' variable (which is 0 thru 7)
'-----

GET_LEDS:

```
    leds = 0                ' Start with zero
    bit0 = RIGHT_LED
    bit1 = CENTER_LED      ' Read in state of pins and
    bit2 = LEFT_LED        ' separate out LED bits
    return                  ' return to sender!
```

```

'
'-----
' Update Servos - This routine called
'   to send update pulses to the
'   servos. (update data kept in
'   STEER_ANGLE and DRIVE_SPEED)
'-----
UPDATE_SERVOS:
  pulsout SERVO_STEER, STEER_ANGLE 'Send steering info first
  LAST_STEER = STEER_ANGLE      'Note the last command
  pulsout SERVO_DRIVE, DRIVE_SPEED 'Send drive speed next!
  LAST_DRIVE = DRIVE_SPEED      'Note the last command
  pause 15                      'All pauses will happen here
  return
'
'-----
' NEW_STEER - Turns wheel slowly from
' LAST_STEER to STEER_ANGLE
'-----
NEW_STEER:
  for counter = 1 to 35
    gosub update_servos
  next
  return
'-----
' GET_RANDOM - Gets a random
' (returns result in REGISTER)
'-----
GET_RANDOM:
  random register      ' Get a number
  register = register & $0ff + 20 & $0ff
                      ' Cut it down below 256
                      ' but above 20

  return
'
'-----
' LEFT_90_TURN - This routine turns
' robot 90 degrees and makes sure the
' appropriate LEDS do NOT light. If
' they do, we abort and make another
' decision
'-----
L90T:
LEFT_90_TURN:
  if LEFT_LED = OFF_TABLE THEN QUICK_ABORT
  ' Stop FAST, ICEBERG!
  DRIVE_SPEED = STOP_HARD      ' Make sure we are stopped!
  STEER_ANGLE = LEFT_90        ' Point it to go left90 degrees

```

```

gosub new_steer      ' Make it so
gosub get_rand      ' Get a random number to run
DRIVE_SPEED = FWD_SLOW      ' Engine room..ahead slow!
for counter = 1 to register  ' Do this a random amount
    if LEFT_LED = OFF_TABLE THEN QUICK_ABORT
        ' Stop FAST, ICEBERG!
        gosub update_servos  ' Send a command to the servos
next
goto CHECK_LEDS      ' Go and make another decision

```

```

'-----
' BACKWARD_LEFT_45_TURN - This routine
' turns robot 90 degrees and makes
' sure the appropriate LEDS do NOT
' light. If they do, we abort and
' make another decision
'-----

```

BL45T:

```

BACKWARD_LEFT_45_TURN:
DRIVE_SPEED = STOP_HARD      ' Make sure we are stopped!
STEER_ANGLE = STRAIGHT      ' Go Straight
gosub new_steer      ' Make it so
DRIVE_SPEED = REV_SLOW      ' Push us backward slow
for counter = 1 to 76      ' Do this a fixed amount
    gosub update_servos  ' Send a command to the servos
next
goto LEFT_90_TURN      ' After backing up, turn left
                        ' and continue to manuver

```

```

'-----
' RIGHT_90_TURN - This routine turns
' robot 90 degrees and makes sure the
' appropriate LEDS do NOT light. If
' they do, we abort and make another
' decision
'-----

```

R90T:

```

RIGHT_90_TURN:
if RIGHT_LED = OFF_TABLE THEN QUICK_ABORT
                        ' Stop FAST, ICEBERG!
DRIVE_SPEED = STOP_HARD      ' Make sure we are stopped!
STEER_ANGLE = RIGHT_90      ' Point it to go left90 degrees
gosub new_steer      ' Make it so
gosub get_rand      ' Get a random number to run
DRIVE_SPEED = FWD_SLOW      ' Engine room..ahead slow!
for counter = 1 to register  ' Do this a random amount
    if RIGHT_LED = OFF_TABLE THEN QUICK_ABORT
        ' Stop FAST, ICEBERG!
        gosub update_servos  ' Send a command to the servos
next

```

```

        goto CHECK_LEDS          ' Go and make another decision
,
'-----
' STRAIGHT - This is the routine
' we spend most of our time in. It
' is the "straight and fast" mode
' where we look for edges of table
' and then JUMP to those routines
'-----
STRA:
    DRIVE_SPEED = STOP_SOFT      ' Go straight and FAST!
    STEER_ANGLE = STRAIGHT      ' Keep it stopped
    gosub new_steer              ' Make it so
    DRIVE_SPEED = FWD_FAST      ' Go forward
STRA_LOOP:
    gosub GET_LEDS              ' Get current LEDS value
    if leds <> 0 THEN QUICK_ABORT ' Stop FAST, ICEBERG!
    gosub update_servos        ' Send a command to the servos
    goto STRA_LOOP              ' Keep going straight if all
                                ' is well!
,

```