

;; SciBorg line follower code

```
global [songon black]

to follow-line
  go-straight
  loop [
    if sees-black? left-sensor [turn-left]
    if sees-black? right-sensor [turn-right]
    if front-bumper? [stop-motion play-song]
  ]
end

to turn-left
  left-wheel off
  right-wheel on thisway
end

to turn-right
  right-wheel off
  left-wheel on thisway
end

to go-straight
  left-wheel on thisway
  right-wheel on thisway
end

to stop-motion
  left-wheel off
  right-wheel off
end

to left-wheel
  a,
end

to right-wheel
  b,
end

to sees-black? :sensor-value
  output :sensor-value > black
end

to left-sensor
  output sensor 0
end

to right-sensor
  output sensor 1
end

to front-bumper?
  output switch 7
end

to back-bumper?
  output switch 8
end
```

```

to black+10
  setblack black + 10
  printblack
end

to black-10
  setblack black - 10
  printblack
end

to printblack
  type [black=] print black
end

; Call this from command center to get default values
to initialize
  setblack 100 ; default threshold
  left-wheel setpower 8
  right-wheel setpower 8
  printblack
end

to play-song
  setsongon 1
  beethoven
end

to stop-song
  setsongon 0
end

to beethoven
  loop[wait 6
    repeat 3 [play_note 90 1 1]
    play_note 86 7 5
    repeat 3 [play_note 88 1 1]
    play_note 85 7 5
    repeat 3 [play_note 90 1 1]
    play_note 86 1 1
    repeat 3 [play_note 91 1 1]
    play_note 90 1 1
    repeat 3 [play_note 98 1 1]
    play_note 95 2 5
    repeat 3 [play_note 90 1 1]
    play_note 85 1 1
    repeat 3 [play_note 91 1 1]
    play_note 88 1 1
    repeat 3 [play_note 100 1 1]
    play_note 97 2 5
  ]
end

to play_note :freq :duration :wait
  if back-bumper? [setsongon 0]
  if songon
    [note :freq :duration
    wait :wait]
end

```