## Release Notes (3.20, December 2021)

### Unicode tables are updated to the latest standard

The Unicode tables now reflects the latest 2021 Unicode set.

### INKEY$ is initialized correctly

Originally, the INKEY$ value would possibly be set by the “F5” press used to start a program. This has been corrected.

### Gopher.Start(..., port)

The Gopher.Start method now takes an addition port number as a parameter. Normally the Gopher server is on port 70; you can set it to any other internet port number.

### Skoobot updates!

The Skoobot is a tiny Bluetooth-controllable robot from <https://www.william-weiler-engineering.com/> . Sample programs include the simplest possible program to set the robot into Rover mode, a keyboard-driven robot program, and a GUI program.

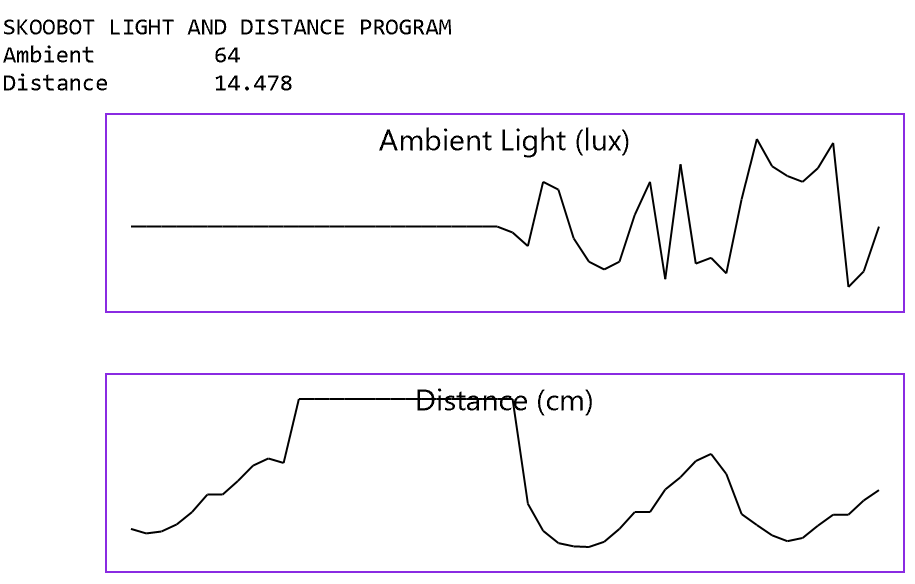
New in this release: the Skoobot specialization can provide a stream of distance and light values. You just have to pass in the function you want called and the number of milliseconds between callbacks.

The light callback gets a value in Lux. The distance callback gets a distance in centimeters (cm).

|  |  |
| --- | --- |
| **Category** | **Methods** |
| Data command | SetupLight(“function”, ms)  SetupDistance(“function”, ms) |

Example: complete program to graph Light data

This example is a complete program that demonstrates setting up an ambient light callback, adding data to an array, and automatically graphing that data.



CLS WHITE BLACK

PRINT "SKOOBOT LIGHT AND DISTANCE PROGRAM"

REM

REM Pick a Skoobot. If there's only one, select it

REM automatically with no user intervention.

REM

devices = Bluetooth.DevicesName("Skoobot\*")

IF (devices.Count = 0)

PRINT "No Skoobot devices found!"

END IF

IF (devices.Count = 1)

device = devices[1]

ELSE

device = Bluetooth.PickDevicesName ("Skoobot\*")

IF (device.IsError) THEN

PRINT "No device selected"

STOP

END IF

END IF

skoobot = device.As ("Skoobot")

REM Set up a Light array

DIM lightData()

lightData.MaxCount = 50

lightData.RemoveAlgorithm = “First”

gl = Screen.Graphics (50, 50, 100, 400)

gl.Title = "Ambient Light (lux)"

gl.GraphY (lightData)

REM Set up a distance array

DIM distanceData()

distanceData.MaxCount = 50

distanceData.RemoveAlgorithm = “First”

gd = Screen.Graphics (50, 180, 100, 400)

gd.Title = "Distance (cm)"

gd.GraphY (distanceData)

REM

REM Set up the Light and Distance functions.

REM Each will be called back about every 100 milliseceonds.

REM

skoobot = device.As (“Skoobot”)

skoobot.SetupLight (“Light”, 100)

skoobot.SetupDistance ("Distance", 100)

FOREVER WAIT

REM Called with light values in LUX

FUNCTION Light (currskoobot, lux)

GLOBAL lightData

lightData.Add (lux)

Screen.ClearLine (2)

PRINT "Ambient", lux

RETURN

REM Called with an approximate distance in cm (centimeter)

FUNCTION Distance (currskoobot, cm)

GLOBAL distanceData

distanceData.Add (cm)

Screen.ClearLine (3)

PRINT "Distance", cm

RETURN

There are complete Skoobot sample programs including

**“A first control program**” which is the simplest possible real Skoobot program. It demonstrates how to connect to a Skoobot and to get a Skoobot specialization.

**“Control program for Skoobot”** is a GUI control program

**“Gopher-of-things for Skoobot”** shows how to control the Skoobot over the internet using the Gopher protocol to a local gateway. The local gateway runs the Gopher-of-things program, connecting to the nearby Skoobot using Bluetooth and offering up a Gopher interface to any Gopher client over the internet.

**“Keyboard-driven Skoobot program”** is a command-line program for the Skoobot.

**“Light and Distance”** demonstrates getting just light and distance data straight from the Skoobot and graphing the results. BC BASIC is particularly good at making it really simple to make graphs that automatically update themselves.