## C:\Users\Almighty\AppData\Local\Microsoft\Windows\INetCache\Content.Word\threePucksCropped.pngTODO: Infinion Puck.Js

The

## Updates to Sensor Extension

### Sensor.Compass (function)

The compass sensor callback returns three values: the direction to magnetic north, the heading accuracy, and the direction to true north. Many systems do not report true north. All angles are reported in degrees.

See the full example below. Not all systems include this sensor, and it’s only available in the IOT Edition.

### Sensor.Inclinometer(function)

The inclinometer callback reports on the pitch, roll and yaw and the yaw accuracy. All values are in degrees.

See the full example below. Not all systems include this sensor, and it’s only available in the IOT Edition.

### Sensor.Light(function)

The light sensor will report the light intensity level in lux (lx). From Wikipedia, common lux values include

|  |  |
| --- | --- |
| Illuminance (lux) | Surfaces illuminated by |
| 0.0001 | Moonless, overcast night sky ([starlight](https://en.wikipedia.org/wiki/Star#Radiation))[[3]](https://en.wikipedia.org/wiki/Lux#cite_note-radfaq-3) |
| 0.002 | Moonless clear night sky with [airglow](https://en.wikipedia.org/wiki/Airglow)[[3]](https://en.wikipedia.org/wiki/Lux#cite_note-radfaq-3) |
| 0.05–0.36 | Full moon on a clear night[[4]](https://en.wikipedia.org/wiki/Lux#cite_note-4) |
| 3.4 | Dark limit of civil [twilight](https://en.wikipedia.org/wiki/Twilight) under a clear sky[[5]](https://en.wikipedia.org/wiki/Lux#cite_note-5) |
| 20–50 | Public areas with dark surroundings[[6]](https://en.wikipedia.org/wiki/Lux#cite_note-NOAO_CaRLLI-6) |
| 50 | Family living room lights (Australia, 1998)[[7]](https://en.wikipedia.org/wiki/Lux#cite_note-energyrating-7) |
| 80 | Office building hallway/[toilet](https://en.wikipedia.org/wiki/Toilet_(room)) lighting[[8]](https://en.wikipedia.org/wiki/Lux#cite_note-8)[[9]](https://en.wikipedia.org/wiki/Lux#cite_note-9) |
| 100 | Very dark overcast day[[3]](https://en.wikipedia.org/wiki/Lux#cite_note-radfaq-3) |
| 320–500 | Office lighting[[7]](https://en.wikipedia.org/wiki/Lux#cite_note-energyrating-7)[[10]](https://en.wikipedia.org/wiki/Lux#cite_note-10)[[11]](https://en.wikipedia.org/wiki/Lux#cite_note-11)[[12]](https://en.wikipedia.org/wiki/Lux#cite_note-12) |
| 400 | [Sunrise](https://en.wikipedia.org/wiki/Sunrise) or [sunset](https://en.wikipedia.org/wiki/Sunset) on a clear day. |
| 1000 | Overcast day;[[3]](https://en.wikipedia.org/wiki/Lux#cite_note-radfaq-3) typical [TV studio](https://en.wikipedia.org/wiki/TV_studio) lighting |
| 10,000–25,000 | Full [daylight](https://en.wikipedia.org/wiki/Daylight) (not direct sun)[[3]](https://en.wikipedia.org/wiki/Lux#cite_note-radfaq-3) |
| 32,000–100,000 | Direct [sunlight](https://en.wikipedia.org/wiki/Sunlight) |

Not all systems include this sensor, and it’s only available in the IOT Edition.

### Full Sensor example

This example shows how to connect to all of the sensors offered in the IOT edition. Each sensor has a similar pattern: first call <type>sensor = Sensor.<type>(“my<type>function”) to start listening. Then make a my<type>function function to get callbacks. Note that the callbacks will be throttled: if too many are sent at once, only the last callback will be delivered. In the example, a timer is fired periodically; after MAXTIME number of seconds, the program will automatically stop.

CLS GREEN

PRINT "All sensors"

PRINT "(not all computers have these)"

maxVolume = 0

REM MAXTIME is the max time to run

MAXTIME = 30

startTime = DateTime.GetNow()

compass = Sensor.Compass("Compass")

inclin = Sensor.Inclinometer("Inclinometer")

light = Sensor.Light("Light")

location = Sensor.Location("Location")

mic = Sensor.Microphone("Microphone")

System.Trace (1)

System.SetInterval ("Timer", 2000, "")

FOREVER

PRINT AT 12,1 "All done!"

FUNCTION Compass (direction, accuracy,↲

trueNortHeading)

Screen.ClearLine (5)

PRINT "Heading", ↲

Math.Round(direction), ↲

accuracy, ↲

Math.Round (trueNorthHeading)

END

FUNCTION Inclinometer (pitch, roll, yaw, accuracy)

Screen.ClearLine (6)

PRINT "Inclin.", ↲

Math.Round(roll), ↲

Math.Round(pitch), ↲

Math.Round(yaw)

END

FUNCTION Light (lux)

Screen.ClearLine (7)

PRINT "Light", lux

END

FUNCTION Location(latitude, longitude, altitude, accuracy)

Screen.ClearLine (8)

PRINT "Position", ↲

Math.Round(latitude, 2), ↲

Math.Round(longitude, 2),↲

accuracy

END

FUNCTION Microphone (data)

v = data.SumOfSquares / data.Count

v = SQR(v)

Screen.ClearLine (9)

PRINT “Volume”, Math.Round (v, 4)

GLOBAL maxVolume

maxVolume = Math.Max(maxVolume, v)

Screen.ClearLine (10)

PRINT “Max Vol.”, ↲

Math.Round (maxVolume, 4)

END

REM Sets us up to wait for 10 seconds

REM and then call FOREVER STOP to

REM stop the 'FOREVER' statement.

FUNCTION Timer (arg)

curr = DateTime.GetNow()

GLOBAL startTime

delta = curr.Subtract (startTime)

Screen.ClearLine (4)

PRINT "TIME", curr.Time,

GLOBAL MAXTIME

IF (delta > MAXTIME) THEN ↲

FOREVER STOP

END

## System Extension

### System.Errors

The System.Errors value is the most recent set of errors encountered while running programs. It persists from one program run to the next (but doesn’t persist after you shut down Best Calculator). This can be useful when debugging Bluetooth issues.

### System.SetInterval (function, interval, argument)

The SetInterval method will start calling functionName at the specified interval (in milliseconds), passing in argument.

In this example, System.SetInterval() sets up a callback on the “showtime” function. That function displays the time. When enough time has passed (5 seconds), it also calls FOREVER STOP which breaks out of the FOREVER statement.

CLS GREEN

start = DateTime.GetNow()

System.SetInterval ("showtime", 500, "arg")

PRINT "About to FOREVER"

FOREVER

PRINT "All done!"

FUNCTION showtime(arg)

Screen.ClearLine (10)

time = DateTime.GetNow()

PRINT "The current time is", time.Time

REM Stop after 5 seconds

GLOBAL start

delta = time.Subtract (start)

IF (delta > 5) THEN FOREVER STOP

END

### System.Version

The System.Version value lets you see the current version and edition of Best Calculator

Example

PRINT System.Version

Which might print out something like

Best Calculator 3.8 (X64)

The (X64) shows that the system architecture is a standard X64 Intel-type computer.

## Release Notes (September 2017)

Pasting from PDF files and Word documents pastes only the text, not the formatting.

The Graphics object now includes g.H and g.W which return the size of the graphics windows in pixels.

The graphics.Rectangle object is now a full object. You can animate a rectangle by setting the X1 and Y1 properties.

Added a System object with useful system-wide properties including

* System.Errors is a list of the most recent errors. PRINT System.Errors and CONSOLE System.Errors are some of the easy ways to see the current set of errors. The error data persists between program runs, but not between restarts of the app.
* System.SetInterval (<function>, <delayInMilliseconds>, <argument>) calls function periodically.
* System.Trace (0) turns off tracing, System.Trace(1) turns on minimal tracing
* System.Version is the current version (e.g., 3009 is version 3.9)

The Sensor object was updated with Sensor.Compass Sensor.Inclinometer and Sensor.Light . These are in addition to the existing Sensor.Location and Sensor.Microphone.

Multiple callbacks are more robust. All callbacks (like from Bluetooth devices) are serialized and will be called between regular statements, during PAUSE statements and during FOREVER statements.