

**MAIN Scene FTBE**

**USER MANUAL**

Version 1.3.0 standalone

TABLE OF CONTENT

[1. PREFACE 3](#_Toc530867555)

[1.1 Legal notice 3](#_Toc530867556)

[1.2 Terms and conditions 3](#_Toc530867557)

[1.3 Disclaimer 3](#_Toc530867558)

[2. INTRODUCTION AND INSTALLATION 4](#_Toc530867559)

[2.1 – Installation and Upgrade Instructions 4](#_Toc530867560)

[2.2 - Global Variables For Time Based Events 5](#_Toc530867561)

[3. SUNRISE & SUNSET VD SETUP 7](#_Toc530867562)

[3.1 – Global variables added by Sunrise & Sunset VD 8](#_Toc530867563)

[3.2 – Setup default margin times in Sunrise & Sunset VD 9](#_Toc530867564)

[4. MAIN SCENE FTBE SETUP 10](#_Toc530867565)

[4.1 – Global variable settings 10](#_Toc530867566)

[4.2 – Repeating scenes setup example 12](#_Toc530867567)

[4.3 – Repeating VD setup example 13](#_Toc530867568)

[4.4 – Scheduled scenes setup example 14](#_Toc530867569)

[4.5 - Scheduled VD setup example 15](#_Toc530867570)

[4.6 - Scheduled global variable change setup example 17](#_Toc530867571)

[4.7 – User GPS location tracking setup and usage 18](#_Toc530867572)

[4.8 – Push messages and mobile devices setup 18](#_Toc530867573)

[4.9 – Change Darkness with light measuring device 20](#_Toc530867574)

[5. HOME STATUS VD SETUP 21](#_Toc530867575)

[6. HOME TIMERS VD SETUP 22](#_Toc530867576)

[7. APPENDIX 1 23](#_Toc530867577)

[7.1 Version History 23](#_Toc530867578)

# PREFACE

## Legal notice

**Fibaro** and **Home Center 2 (HC2)** are registered trademarks of **Fibar Group S.A.** registered in Poland and other countries.

## Terms and conditions

Permission to use, copy, modify, and distribute this software and its documentation for educational, research, personal use and non-profit purposes, without fee and without a signed licensing agreement, is hereby granted, provided that the above copyright notice, with "Terms and Conditions" and "Disclaimer" appear in all copies, modifications, and distributions.

It is strictly forbidden to sell, rent, lease and/or lend this software for profit without prior consent from the Author. Please contact the Author by e-mail: [z.sankovic@gmail.com](mailto:z.sankovic@gmail.com) for commercial licensing opportunities.

## Disclaimer

This software is provided by copyright owner "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the author and distributor be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Fibar Group S.A. and their employees **are not** responsible for support of the Main scene FTBE. Please contact the author, on the Fibaro Forum, for any questions or support required.

The Author however, wishes to acknowledge the help received from the Fibar Group S.A. in that they have kindly supplied the use of a HC2 for development purposes, free of charge.

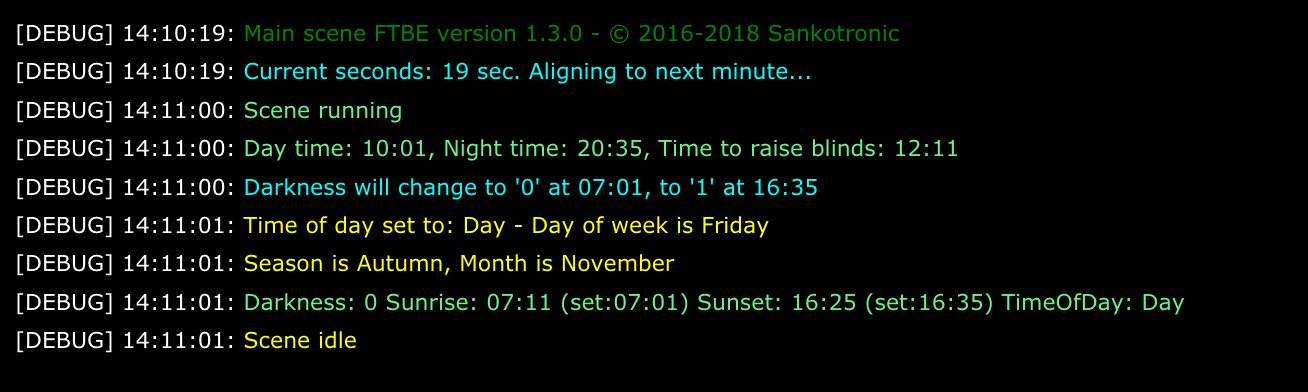
# INTRODUCTION AND INSTALLATION

Main scene FTBE is the beating heart of the HA system. It is the only scene necessary to loop endlessly because it is the source of most important events generated on our gateways and that are time based events! It is looping once every minute making sure that our gateways are aware of the current time, part of the day, week, month and season of the year. Many actions done by our gateways are depending on current time like raising or lowering blinds, lighting control, waking up of our home and us, brewing coffee at proper time (this is extremely nice!), reminding us of important events and many other things. This scene will take care of that  by changing various global variables at appropriate time thus activating other scenes triggered by them and the user can even configure it to start other scenes or press buttons on VD's at predefined intervals or at scheduled times.

## ****– Installation and Upgrade Instructions****

**INSTALLATION**

IMPORTANT! Please follow bellow order to install Main scene FTBE and accompanying VD’S

1. Please first import "Sunrise & Sunset" VD to your HC.
2. Create new LUA scene and copy and paste code. Save to start scene and check debug window for any errors. If all good it should look like this: 
3. Other two VD "Timers" and "Home Status" can be imported last

**UPGRADE** procedure from previous versions to 1.3.0:

1. First copy settings of current Main scene FTBE to plain text editor
2. Copy and paste complete code of new Main scene version over the old code
3. Copy your settings from old to new scene part by part since there are some changes in layout of code in new scene that should not be changed or deleted
4. Delete old Sunrise & Sunset VD and import new version.

## - Global Variables For Time Based Events

Main scene is taking care that these global variables change their values at proper time and must be added to global variable panel before start using the scene. This global variables are also shown on Home Timers VD and their value can be changed by this VD:

1. **Darkness** - ***global variable*** *(upper part of variable panel)*with possible values **0** for day time (**Light**) and **1** for night time (**Dark**). Value is changed at adjusted time for sunrise and sunset time or depending on measured outside light level. This variable is also used in Very Smart Lights™ scene.
2. **TimeOfDay** - is ***predefined global variable*** *(lower part of variable panel)* with possible values "**Morning**", "**Day**", "**Evening**", "**Night**" or equal values in users language which then can be mapped to English values used by this scene. Time of day will be changed as follows:
   * + Night to Morning - at adjusted sunrise time or when global SleepState change to Awake if useSleepState is set to true.
     + Morning to Day - after defined time interval from adjusted sunrise time.
     + Day to Evening   - at adjusted sunset time.
     + Evening to Night - after defined time interval from adjusted sunset time or when global SleepState change to Sleep.

Virtual device "**Sunrise & Sunset**" can be used to adjust sunrise and sunset times for **+/- 59** minutes and also define time intervals after which Day or Night time will be set. I use two scenes to change Global variable **SleepState**. Scene "Good night" is started by FIBARO swipe or switch 2 on dimmer in our bedroom and sets this variable to Sleep after checking that selected doors and windows are closed, all lights are turned off, etc. Scene "Good morning" is either started by Alarm clock or manually with swipe or switch 2 in our bedroom and after turning on slowly selected lights and starting and slowly raising volume of our Sonos speakers and starting coffee machine changes value of variable **SleepState** to Awake. This variable is also important for Very Smart Lights™ scene.

1. **WeekDay** - is ***predefined global variable*** *(lower part of variable panel)* possible values are name of days in user preferred language with possibility to map names so no need to change scene code. This can be used to display week day on tablet controller or even use for triggering some events that must happen at midnight of each or some days of the week.
2. **Month** - is ***predefined global variable*** *(lower part of variable panel)* possible values are names of months in user preferred language with possibility to map names so no need to dig into scene code.
3. **SeasonState** - is ***predefined global variable*** *(lower part of variable panel)* possible values are names of four seasons in user preferred language with possibility to map names.

Following global variables are used for home state, alarm state and demonstration mode and their value can be changed by Home Status VD. User can add them and map to preferred language:

1. **SleepState** - is ***predefined global variable*** *(lower part of variable panel)* with values "Awake", "Sleep" or values in your language that you can map in scene code. Value is changed by other scenes or VD
2. **PresentState** - is ***predefined global variable*** *(lower part of variable panel)* with values "Home", "Away" and "Holiday" or values in your language that you can map in scene code. Value is changed by other scenes or VD
3. **HouseAlarm** - is ***predefined global variable*** *(lower part of variable panel)* with values "Armed", "Partly armed", "Arming", "Disarmed" and "Disarming" or values in your language that you can map in scene code. Value is changed by other scenes or VD
4. **AlarmState** - is ***predefined global variable*** *(lower part of variable panel)* with values "Safe" and "Breached" or values in your language that you can map in scene code. Value is changed by other scenes or VD
5. **DemoMode** - predefined global variable with values: "Yes", "No" or can be set in user preferred language due to mapping possibility. Value changed by Home status VD and if set to "Yes" then scene will no more change time based variables like Darkness, TimeOfDay, SeasonState so they can be changed by VD to make test of the system or just demonstration at different times of day.

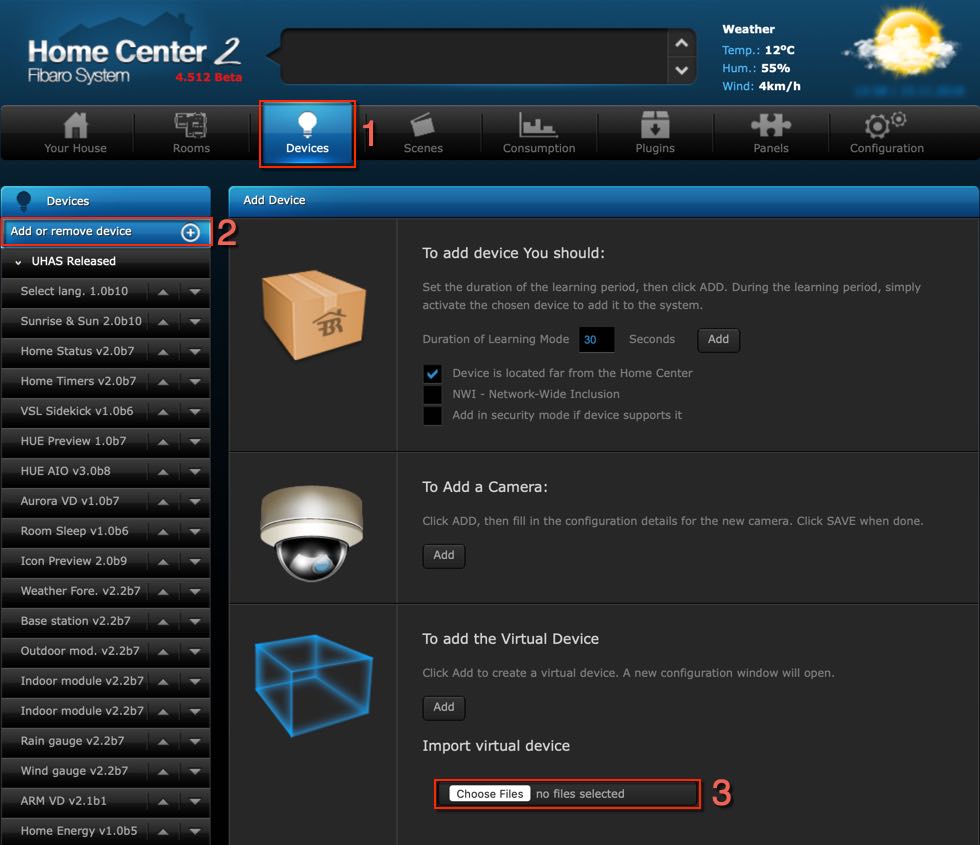
This global variable is not necessary to be used:

* **HomeTable** - predefined global variable table with device and scene IDs. Recommended to use since z-wave devices can change their ID with re-inclusion and then is necessary  to edit only scene which make this table and only device ID in scene headers. Much less time and effort is needed than without that option! So if you use this table then uncomment line bellow:

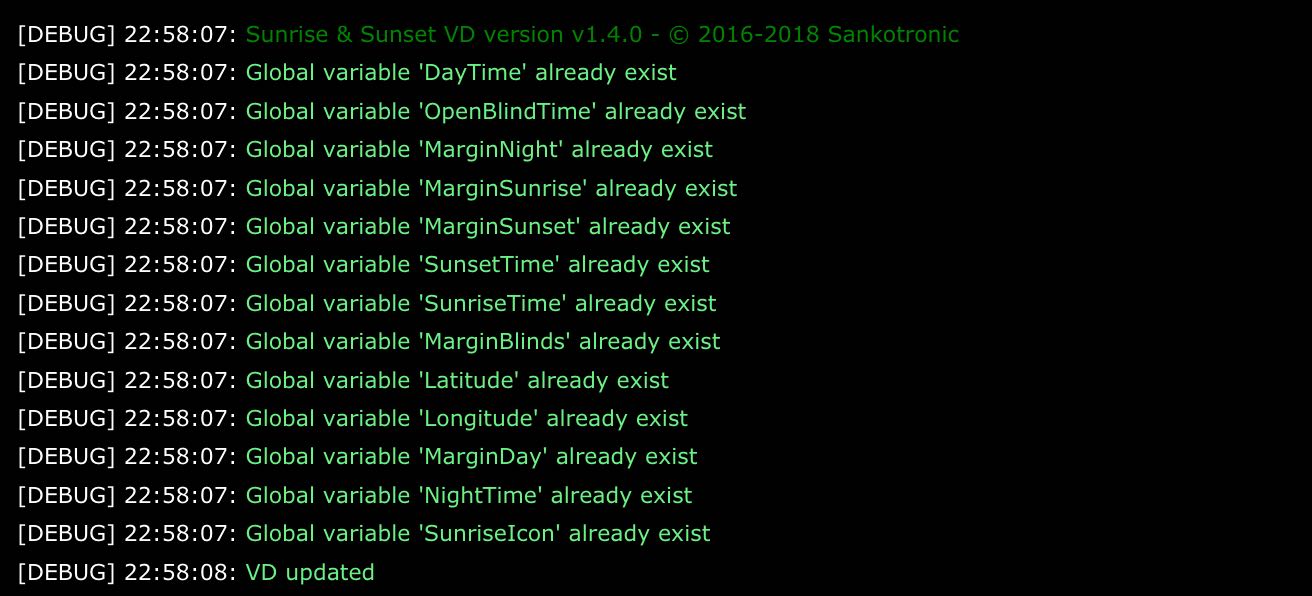
|  |
| --- |
| -- get the table from global if not using then delete this line!!!  local jT = json.decode(fibaro:getGlobalValue("HomeTable")) |

# SUNRISE & SUNSET VD SETUP

Sunrise & Sunset VD can be imported into HC2 by choosing "Add or remove device" on the devices page by choosing the option to import a Virtual Device:



After importing VD click on Advanced tab, scroll all the way down to main loop code window and click on Debug button. Debug window will open and this debugging messages will be shown:



## – Global variables added by Sunrise & Sunset VD

Following global variables are automatically added to global variable panel and set to default values by Sunrise & Sunset VD when imported so user don't need to add them:

* **MarginSunrise** - global variable with possible values from **-59 to +59 Min**, used to adjust sunrise time
* **MarginSunset** - global variable with possible values from **-59 to +59 Min**, used to adjust sunset time
* **MarginDay** - difference time from sunrise when morning starts until time when day time starts in format "00:00", minimum 1:00h, maximum 6:59h difference
* **DayTime** - exact time when morning time changes to day time in format "00:00", shows exact time when Morning will change to Day time
* **MarginNight** - difference time from sunset when evening starts until time when night time starts in format "00:00", minimum 1:00h, maximum 6:59h difference
* **NightTime** - exact time when evening time changes to night time in format

"00:00", shows exact time when Evening will change to Night time.

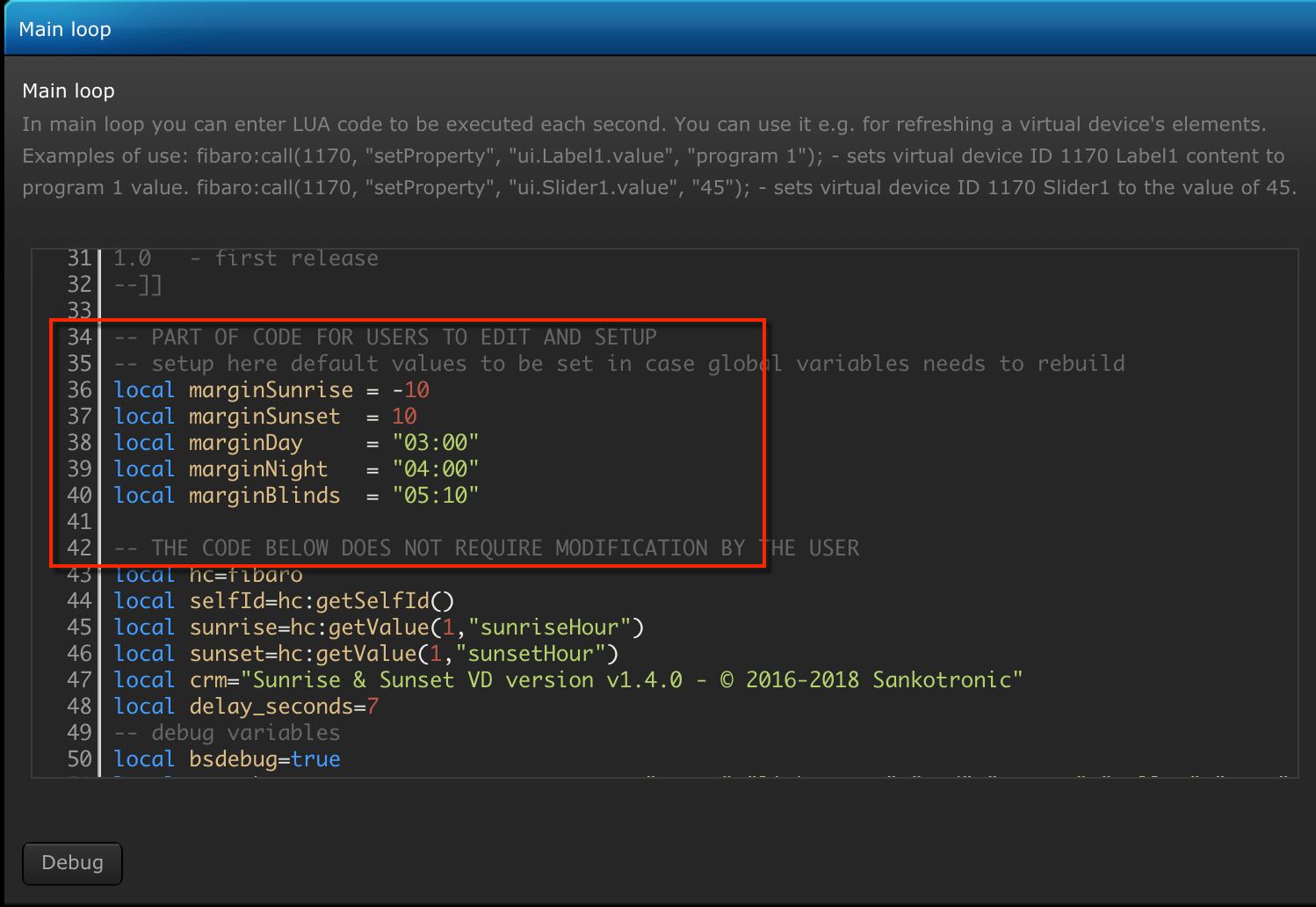
* **MarginBlinds** - difference time from sunrise until time to open selected blinds in format "00:00", minimum **00:00h**, maximum 12:59h difference
* **OpenBlindTime** - exact time when blinds will open in format "00:00", shows exact time when blinds will be open
* **Latitude** - Position of your HC
* **Longitude** - Position of your HC

Since version 1.4.0 - new global variable same as above, added to variable panel by Sunrise & Sunset VD:

* **SunriseIcon** – Remembers VD main icon ID that was selected and is used for function that sets same icons on all VD buttons.

## – Setup default margin times in Sunrise & Sunset VD

User need to setup global variable names and map values to his preferred language in Main loop and all buttons code same way as it is explained for main scene in next chapter. Also it is possible to setup margin times default values that will be used to rebuild global variables in case are corrupted or deleted:



# MAIN SCENE FTBE SETUP

This scene can be setup to start different scenes or press button on different VD's at predefined interval that can be from 1 minute to 1 year. Main scene can also be setup to run different scenes or press buttons on different VD's at scheduled times (more than one per day) and can be defined for each day of the week. But let us first setup global variables.

## – Global variable settings

For setting up global variables there is detailed explanation in scene code, but here is how looks that part (NOTE! this is not complete code, but just part to setup global variables):

|  |
| --- |
| -- GLOBAL VARIABLES  -- enter names and value mapping of your global variables or leave as it is  -- get the table of device & scene ID's from global variable HomeTable. If  -- using then uncomment bellow line else leave it as it is!  -- local jT = json.decode(hc:getGlobalValue("HomeTable"))  -- "Darkness" is global variable with two possible states 0 - for day time and 1  -- for night time and it is changed at sunrise & sunset by main scene that is  -- responsible for all time based events. See my other posts for details  local darkness = "Darkness"  local darknessMapping = {Light="0", Dark="1"}  -- "SleepState" is predefined global variable with possible values:  -- "Sleep", "Awake".  local sleepState = "SleepState"  local sleepStateMapping = {Sleep="Sleep", Awake="Awake"}  -- if you want that TimeOfDay changes to Morning (or Day) if you wakeup  -- before sunrise time inestead at sunrise time then set useSleepState  -- to true therwise leave it false  local useSleepState = false  -- predefined global variable that keeps current day of week.  -- make sure that value mapping corresponds to day names in  -- your language  local weekDay = "WeekDay"  local weekDayMapping = {Sunday ="Sunday",  Monday ="Monday",  Tuesday ="Tuesday",  Wednesday ="Wednesday",  Thursday ="Thursday",  Friday ="Friday",  Saturday ="Saturday"}  -- enter week days in your language and mapped in previous variable  local weekDayMap = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}  -- predefined global variable that keeps current month name.  -- make sure that value mapping corresponds to month names in  -- your language  local month = "Month"  local monthMapping = {January ="January",  February ="February",  March ="March",  April ="April",  May ="May",  June ="June",  July ="July",  August ="August",  September ="September",  October ="October",  November ="November",  December ="December"}  -- enter month names in your language and mapped in previous variable  local monthMap = {"January", "February", "March", "April", "May", "June",  "July", "August", "September", "October","November", "December"}  -- predefined global variable that keeps current season and map  -- your values  local seasonState = "SeasonState"  local seasonStateMapping = {Spring="Spring", Summer="Summer", Autumn="Autumn", Winter="Winter"}  -- predefined global variable that keeps current time of day and map  -- your values  local timeOfDay = "TimeOfDay"  local timeOfDayMapping = {Morning="Morning", Day="Day", Evening="Evening", Night="Night"}  -- predefined global variable that determines normal or demo  -- mode of the system. Demo mode can be used for testing or  -- demostration of the system set by "Home state" VD  local demoMode = "DemoMode"  local demoModeMapping = {On="Yes", Off="No"}  -- GPS LOCATION AND USERS SETUP  -- define users for which you want to get GPS location. Enter users inside  -- braces separated by comma  local userID = {}  -- PUSH MESSAGES AND MOBILE DEVICE SETUP  -- define mobile devices to send push messages. Enter devices inside  -- braces separated by comma  local iosDeviceID = {}  -- flag for each user if 1 then send notifications else if 0 do not  -- send notifications. You can add code in function extraUserCodeFirst()  -- where you can change user flags depending on some global variable.  local iosDeviceFlag = {} |

For example if global variable for **Darkness** is called **LightDark** on user system and values for **Light** is **day** and for **Dark** is **night** then user will edit this part to look like this:

|  |
| --- |
| local darkness = "LightDark"  local darknessMapping = {Light="day", Dark="night"} |

Same is for all other global variables. For example for weekDay and Month it is possible to use names of weekdays and months in preferred language only what is needed is to correct values above that are inside quotes (green text). When that is done scene is ready to run!

**REMINDER** – Same setup with global variables must be done in Sunrise & Sunset VD, Home Status VD and Home Timers VD.

## – Repeating scenes setup example

Instructions how to set up this part are in code itself, but let see first how to setup scene to run at predefined interval. Here is that part of code:

|  |
| --- |
| -- REPEATING SCENES SETUP ------------------------------------------------  -- here you can setup scenes that will be executed at predefined interval  -- in minutes. Lowest value is every 1 minute and longest interval is up to  -- you. 24h = 1440m, 1week = 10080m, 30d = 43200m etc.  -- enter scene names that will run repeatedly every X minutes separated  -- by comma. This is needed for debugging and can be used to send push  -- notification  local runSceneRepeatName = {"Alarm clock check", "Netatmo weather update", "Weather VD refresh"}  -- enter scene IDs that will run repeatedly every X minutes separated  -- by comma.  local runSceneRepeatID = {jT.scene.AlarmClock, jT.scene.Netatmo, jT.scene.Weather\_refresh};  -- enter at which interval time in minutes will scenes run separated  -- by comma.  local runSceneRepeatTime = {1, 5, 5};  -- enter here flag for sending push message when scene is executed  -- 1 for sending message, 0 no messages  local runSceneRepeatPushFlag = {0, 1, 0};  -- enter here message content that will be sent to you when scene is  -- executed. If above flag is 0 for any of the scenes then just put  -- empty "" for it.  local runSceneRepeatPushMessage = {"", "Netatmo weater station data updated", ""};  -- for each scene enter variable with name "count" (must be all same name)  -- separated by comma. This is used to count interval at which scene will  -- be started and is neccessary for proper execution!  local runSceneRepeatCount = {count, count, count};  -- enter here "Yes" for each scene if you want them to run immediatelly when  -- this scene starts running after reboot or saving or enter "No" if you  -- want scene to run after defined interval after scene is restarted  local runSceneAtStart = {"Yes", "Yes", "Yes"}; |

As can be seen from above example, three scenes are setup to run repeatedly. First scene is for checking Alarm clock and it is run every minute with PushFlag set to 0 and empty text for PushMessage. Other two scenes are for updating weather and they will run every 5 minutes since **runSceneRepeatTime** is set to **5**. When Netatmo weather scene is started also push message is sent.

In above example global variable **HomeTable** is used which keeps ID numbers of all devices and scenes in a table. That is why in setup **runSceneRepeatID** are pointing variables **jT.sceneAlarmClock**, **jT.scene.Netatmo** and **jT.scene,Weather\_refresh**, but they are actually representing ID numbers of the appropriate scene. if HomeTable is not used then above setting would look like **runSceneRepeatID = {61, 76, 99}**. If any mistake is made in setup and any parameter is missing or wrong then scene will inform user during first start!

## – Repeating VD setup example

|  |
| --- |
| -- REPEATING VIRTUAL DEVICE SETUP ----------------------------------------  -- here you can setup VD that will be executed at predefined interval  -- in minutes. Lowest value is every 1 minute and longest interval is up to  -- you. 24h = 1440m, 1week = 10080m, 30d = 43200m etc.  -- enter VD IDs that will run repeatedly every X minutes separated  -- by comma.  local runVDRepeatID = {jT.waether.update, jT.watchdog};  -- enter here which button to press to execute command  local runVDRepeatButton = {"11", "2"};  -- enter at which interval time in minutes will VD run separated by comma.  local runVDRepeatTime = {5, 1440};  -- enter here flag for sending push message when VD is executed 1 for  -- sending message, 0 no messages  local runVDRepeatPushFlag = {0, 1};  -- enter here message content that will be sent to you when VD is  -- executed. If above flag is 0 for any of the VD then just put  -- empty "" for it.  local runVDRepeatPushMessage = {"", "Main scene is running ok"};  -- for each VD enter variable with name "count" (can be all same name)  -- separated by comma. This is used to count interval at which VD will  -- be executed and is neccessary for proper execution!  local runVDRepeatCount = {count, count};  -- enter here "Yes" for each VD if you want them to run immediatelly when  -- this scene starts running after reboot or saving or enter no if you  -- want VD to run after defined interval  local runVDAtStart = {"No", "Yes"}; |

Same as for repeating scenes user can also setup to press button on virtual devices at predefined time interval. In above example Weather update VD button "11" to be pressed every 5 minutes is setup. Number of the button to be pressed is easy to find out, just open device and count all labels, sliders and buttons from up down and from left to right. Example how to do that can be found here: <https://forum.fibaro.com/index.php?/topic/23283-need-some-help-with-some-scene-problems/&do=findComment&comment=104166>

Anyway the other VD is more interesting. It is a Watchdog VD which button "2" is pressed every 24h (1440 min) and push notification is sent to user to verify Main scene is running. "Watchdog" VD button will be pressed immediately after scene is started since **runVDAtStart** is set to "**Yes**", while "Weather update" VD button will be pressed after 5 minutes from starting this scene. Same rules apply for REPEATING SCENE SETUP.

## – Scheduled scenes setup example

|  |
| --- |
| -- SCHEDULED SCENES SETUP ------------------------------------------------  -- here you can setup scenes that will be executed at predefined time of  -- day in format "00:00". You can set up more than one time during day.  -- Also you can enter flag for each day of the week when scene will be  -- executed or not. See more details bellow.  local runSceneSchedName = {"Weekly battery check", "Open blinds after sunrise"}  -- enter scene IDs that will run at scheduled time separated by comma.  local runSceneSchedID = {jT.scene.Batt\_check, jT.scene.Blinds\_open};  -- enter at what time of day will scenes run separated by comma.  -- You can define more than one time in format "00:00" for one scene  -- or even use local or global variables like sunsettime or sunrisetime  -- Example: {{"13:30", "19:20"}, {sunrisetime, "13:00", sunsettime}}  -- in above example first scene will run at 13:30 and 19:20 while  -- second scene will run at sunrisetime, 13:00 and at sunsettime  -- IMPORTANT NOTE - if you use variable to set time then make sure that  -- you add refreshing of value in function extraUserCodeFirst() like  -- this for above example:  -- runSceneSchedHour[2][1] = sunrisetime  -- runSceneSchedHour[2][3] = sunsettime where first square brackets  -- [2] with number inside refers to second scene and second square  -- brackets [1] with number inside refers to time in order which means  -- sunrisetime is 1st time [1] for 2nd [2] scene and sunsettime is  -- 3rd [3] time for 2nd [2] scene in a row added to the variable.  local runSceneSchedHour = {{"10:00"}, {blindtime}};  -- enter here flag for each day of the week for scene to run at above  -- scheduled time. if flag 1 the scene will run and if flag 0 then  -- will not be run for that day. Remeber, Sunday is first day of the  -- week on HC2! Example for two scenes:  -- {{1, 0, 0, 0, 0, 0, 0}, {0, 1, 1, 1, 1, 1, 0}} - in this example  -- first scene will run only on Sunday, and second scene will run on  -- weekdays but not on weekends (Saturday and Sunday).  local runSceneSchedWeek = {{1, 0, 0, 0, 0, 0, 0}, {1, 1, 1, 1, 1, 1, 1}};  -- enter here flag for sending push message when scene is executed  -- 1 for sending message, 0 no messages  local runSceneSchedPushFlag = {1, 0};  -- enter here message content that will be sent to you when scene is  -- executed. If above flag is 0 for any of the scenes then just put  -- empty "" for it.  local runSceneSchedPushMessage = {"Battery check scene activated", ""}; |

Here user can setup scenes that are executed at scheduled times (can be more than one time) every day in a week, or just selected days in a week. In above example scene "Weekly battery check" to run at 10:00h only on Sunday is setup. Push message "Battery check scene activated" will be sent since **runSceneScedPushFlag** is set to **1**. The other scene "Open blinds after sunrise" run time is set by variable **blindtime** and for every day of the week. Since variables need to be refreshed if used for scheduled activation of the scenes, an equation also need to be added in the part of scene that is called **extraUserCodeFirst()** as follows:

|  |
| --- |
| -- EXTRA FUNCTION WHERE YOU CAN ADD YOUR CODE ----------------------------  -- use this function to add code that will be executed before all other  -- code in the loop  function extraUserCodeFirst()  -- if you use variable for scheduled execution of scene or VD then you  -- need to add here equation to get last value calculated  runSceneSchedHour[2][1] = blindtime  -- your code goes here  end |

## - Scheduled VD setup example

User can setup scheduled VD and have pressed different buttons on same VD at different times! So if for one VD user define only one button it still must be inside double braces {{"2"}}. For same VD more buttons {{"2", "8"}}. Number of buttons for one VD must be equal to number of scheduled times!! See example bellow:

|  |
| --- |
| -- SCHEDULED VD SETUP ---------------------------------------------------  -- here you can setup VDs that will be executed at predefined time of  -- day in format "00:00". You can set up more than one time during day.  -- Also you can enter flag for each day of the week when VD will be  -- executed or not. See more details bellow.  -- enter VD IDs that will run at scheduled time separated by comma.  local runVDSchedID = {384, 255};  -- enter here which button to press to execute command for each time that  -- you enter bellow. Example for Sonos remote VD from Krikroff. If you set  -- runVDSchedButton = {{"7", "9"}} and  -- runVDSchedHour = {{"07:00", "11:00"}}  -- Then button "2" (Play) will be pressed at 07:00 to start music and  -- button "9" (Stop) will be pressed at 11:00 to stop music.  local runVDSchedButton = {{"7", "9"}, {"2"}};  -- enter at what time of day will VD button be pressed separated by comma.  -- You can define more than one time in format "00:00" for one VD  -- or even use local or global variables like sunsettime or sunrisetime  -- Example: {{"13:30", "19:20"}, {sunrisetime, "13:00", sunsettime}}  -- in above example first VD button will be pressed at 13:30 and 19:20  -- while second VDs button will be pressed at sunrisetime, 13:00 and  -- at sunsettime  local runVDSchedHour = {{"07:00", "22:00"}, {"10:00"}};  -- enter here flag for each day of the week for VD to press button  -- at above scheduled time. if flag 1 the VD button will be pressed and  -- if flag 0 then will not be pressed for that day. Remeber, Sunday is  -- first day of the week on HC2! Example for two VDs:  -- {{1, 0, 0, 0, 0, 0, 0}, {0, 1, 1, 1, 1, 1, 0}} - in this example  -- first VDs button will be pressed only on Sunday, and second VDs  -- button will be pressed on weekdays but not on weekends (Saturday  -- and Sunday).  local runVDSchedWeek = {{1, 0, 0, 0, 0, 0, 0}, {0, 1, 1, 1, 1, 1, 0}};  -- enter here flag for sending push message when VD buttons is pressed  -- 1 for sending message, 0 no messages  local runVDSchedPushFlag = {0, 0};  -- enter here message content that will be sent to you when VD is  -- executed. If above flag is 0 for any of the VDs then just put  -- empty "" for it.  local runVDSchedPushMessage = {"", ""}; |

Setting is completely same as for scheduled scenes except that it is needed to enter ID of VD's and their number of button that is to be pressed at scheduled time. In above example "Sync time" VD is setup to be activated every day at 2:00h to synchronize gateway clock since it is not doing it by its own settings. It is also possible to use variables to schedule time but then also equation to the function extraUserCodeFirst() must be added to take value from global variable like this:

|  |
| --- |
| -- EXTRA FUNCTION WHERE YOU CAN ADD YOUR CODE ----------------------------  -- use this function to add code that will be executed before all other  -- code in the loop  function extraUserCodeFirst()  -- if you use variable for scheduled execution of scene or VD then you  -- need to add here equanation to get last value calcualted  runVDSchedHour[1][1] = fibaro:getGlobalValue("YourGlobalVariableName");  -- your code goes here  end |

This actually gives user endless possibilities because it is possible to add global variables in this scene and then use them to manipulate scheduled time of the scenes and VD’s when to be executed.

**IMPORTANT NOTE!** YourGlobalVariableName from above MUST contain time in format "00:00" otherwise it will not work!!!

## - Scheduled global variable change setup example

From version 1.2.4 scheduled change of global variable values is added. This is almost same as scheduled VD button press except that it allows to change values of global variables at predefined times. Same rules apply to this option as for scheduled VD setup! Bellow example shows how to setup change of one global variable value at two different times of day:

|  |
| --- |
| -- SCHEDULED GLOBAL VARIABLE CHANGE SETUP -------------------------------  -- here you can setup global variable which value will be changed at  -- predefined time of day in format "00:00". You can set up more than one  -- time during day. Also you can enter flag for each day of the week when  -- global variable value will change or not. See more details bellow.  -- enter global variable name which value you want to change at scheduled  -- time separated by comma.  local chgGlobalSchedName = {"LampFlag"};  -- enter here what value will be set to global variable for each time that  -- is entered bellow  local chgGlobalValue = {{1, 0}};  -- enter at what time of day will global value change separated by comma.  -- You can define more than one time in format "00:00" for one VD  -- or even use local or global variables like sunsettime or sunrisetime  -- Example: {{"13:30", "19:20"}, {sunrisetime, "13:00", sunsettime}}  -- in above example first global variable value will be changed at 13:30  -- and another value at 19:20 while second global variable value will be  -- changed at sunrisetime, 13:00 and at sunsettime  local chgGlobalHour = {{"07:00", "09:00"}};  -- enter here flag for each day of the week for VD to press button  -- at above scheduled time. if flag 1 the VD button will be pressed and  -- if flag 0 then will not be pressed for that day. Remeber, Sunday is  -- first day of the week on HC2! Example for two VDs:  -- {{1, 0, 0, 0, 0, 0, 0}, {0, 1, 1, 1, 1, 1, 0}} - in this example  -- first VDs button will be pressed only on Sunday, and second VDs  -- button will be pressed on weekdays but not on weekends (Saturday  -- and Sunday).  local chgGlobalWeek = {{0, 1, 1, 1, 1, 1, 0}};  -- enter here flag for sending push message when VD buttons is pressed  -- 1 for sending message, 0 no messages  local chgGlobalPushFlag = {0};  -- enter here message content that will be sent to you when VD is  -- executed. If above flag is 0 for any of the VDs then just put  -- empty "" for it.  local chgGlobalPushMessage = {""}; |

Above example will change value of global variable "LampFlag" to value 1 at 7:00h in the morning and then to 0 at 9:00h every day of the week except on Saturday and Sunday and will not send push notification.

## – User GPS location tracking setup and usage

**NEW with version 1.2.3!**

Main scene FTBE from version 1.2.3 can read users current location and calculate distance from home. User can setup user IDs for whom location can be retrieved and distance calculated. Setup is done in following section:

|  |
| --- |
| -- GPS LOCATION AND USERS SETUP ------------------------------------------  -- define users for which you want to get GPS location. Enter users inside  -- braces separated by comma  local userID = {2, 45, 182}; |

At the moment only position of users and distance from home is calculated but there is no other code to use that data. User can write his own code to use this data which then must be added to function **extraUserCodeLast()**. Users position is contained in local variable **userposition[x]** where **x** is number of user position in variable **userID**. For example if the user set userID = {100, 150, 155} then position of userID 100 is userposition[1] while userID 155 position is userposition[3]. Distance for each user is contained in local variable **userdistance[x]** where **x** is same as for userposition[x]! The user can find user IDs by entering this link to your web browser:  *http://<YOUR\_HC\_IP>/docs/* and then clicking on **users**.

## – Push messages and mobile devices setup

**NEW with version 1.2.3!**

To use push notifications mobile devices ID’s must be set. This setup is in this section of the scene:

|  |
| --- |
| -- PUSH MESSAGES AND MOBILE DEVICE SETUP ---------------------------------  -- define mobile devices to send push messages. Enter devices inside  -- braces separated by comma  local iosDeviceID = {jT.ios.me, jT.ios.my\_ipad, jT.ios.wife, jT.ios.wife\_ipad};  -- flag for each user; if 1 then send notifications else if 0 do not  -- send notifications. You can add code in function extraUserCodeFirst()  -- where you can change user flags depending on some global variable.  local iosDeviceFlag = {1, 1, 1, 1}; |

Mobile devices IDs can be found by entering this link in web browser  *http://<YOUR\_HC\_IP>/docs/* and then clicking on **iosDevices**.

**iosDeviceFlag** is used to determine to which mobile devices push message will be sent. If flag is set to 1 then notification is sent otherwise not. To change flags for mobile devices depending on situation user can add chunk of code in function **extraUserCodeFirst()** to change them as needed. Following example uses global variables to change **iosDeviceFlag** depending if dad and mom are at home or away:

|  |
| --- |
| -- this part goes under GLOBAL VARIABLE SETUP  -- for all those daddies and momies who work away from home to lower number of  -- notifications while at work possible values: "Yes", "No"  local dadAway = "DadAway"  local dadAwayMapping = {Yes="Yes", No="No"}  local momAway = "MomAway"  local momAwayMapping = {Yes="Yes", No="No"}  ---------------------------------------------  function extraUserCodeFirst()  -- DEFINE FLAGS - in this section add code to change users flags -----------------  if dadAway ~= "" then  if fibaro:getGlobalValue(dadAway) == dadAwayMapping.Yes then  iosDeviceFlag[1] = 0  iosDeviceFlag[2] = 0  else  iosDeviceFlag[1] = 1  iosDeviceFlag[2] = 1  end  end  if momAway ~= "" then  if fibaro:getGlobalValue(momAway) == momAwayMapping.Yes then  iosDeviceFlag[3] = 0  iosDeviceFlag[4] = 0  else  iosDeviceFlag[3] = 1  iosDeviceFlag[4] = 1  end  end  end |

If the user don't want to use this feature then can just add users that will receive push messages and set **iosDeviceFlag** for each of them to 1.

## – Change Darkness with light measuring device

**New with version 1.3.0**

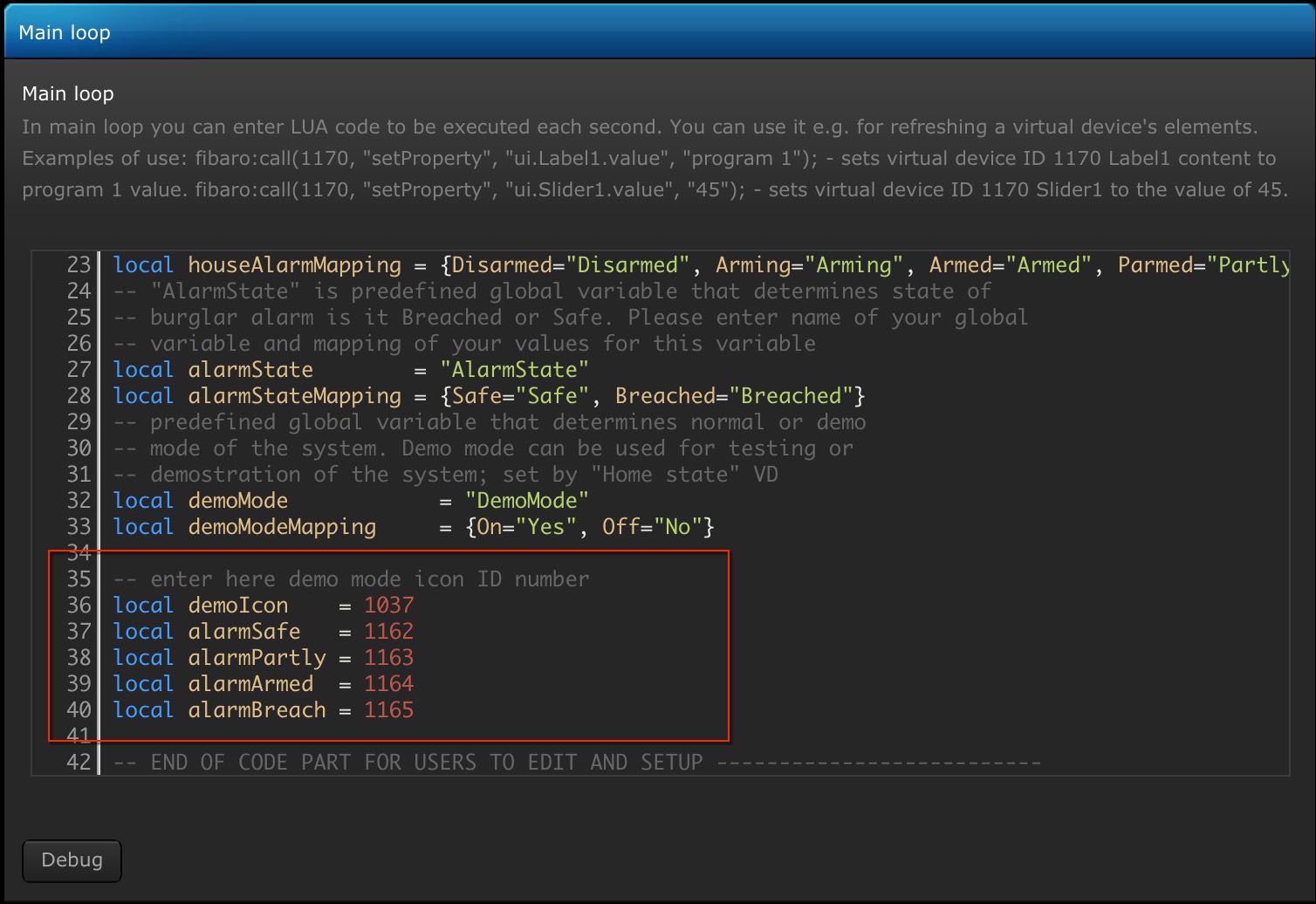
With new version it is possible to setup light measuring devices or global variables which contain value of measured light intensity to change Darkness global variable value. Here is example of using light measuring device and global variables:

|  |
| --- |
| -- PART OF CODE FOR USERS TO EDIT AND SETUP  -- DARKNESS LIGHT SENSOR SETUP -- NEW!  -- if you setup light sensor here then scene will change global Darkness  -- according to sensor setup instead of using sunrise and sunset times  -- for example {400,320}  local luxID = {38,412}  -- or you can setup global variables names that are set by other light sensors  -- for example {"LuxGarden","LuxShade"}  local luxGlo = {"LuxGarden", "LuxShade"}  local luxLevelmin = 100  local luxLevelmax = 300  -- luxDelayTime is set in minutes and is used to avoid frequent change of Darkness  -- default setting is 5 minutes  local luxDelayTime = 5  -- do not change this variables  local luxDelay=(60\*luxDelayTime)  local lastChange=os.time()-luxDelay  -- END DARKNESS LIGHT SENSOR SETUP |

In above example two light sensors are setup with ID’s **38** and **412** and also two global variables **LuxGarden** and **LuxShade** that are checked every minute and if their average light level (lux) is bellow **luxLevelmin** then **Darkness** will be set to 1 or Dark. If measured lux is between **luxLevelmin** and **luxLevelmax** then no changes will be made and of course if lux is above **luxLevelmax** setting then **Darkness** will change to 0 or Light. To avoid frequent changes of Darkness **luxDelayTime** is defined in minutes in which next change will not occur. User can adjust settings of **luxLevelmin**, **luxLevelmax** and **luxDelayTime** as needed. Also any number of light sensors and global variables can be setup and scene will calculate average light level (lux).

# HOME STATUS VD SETUP

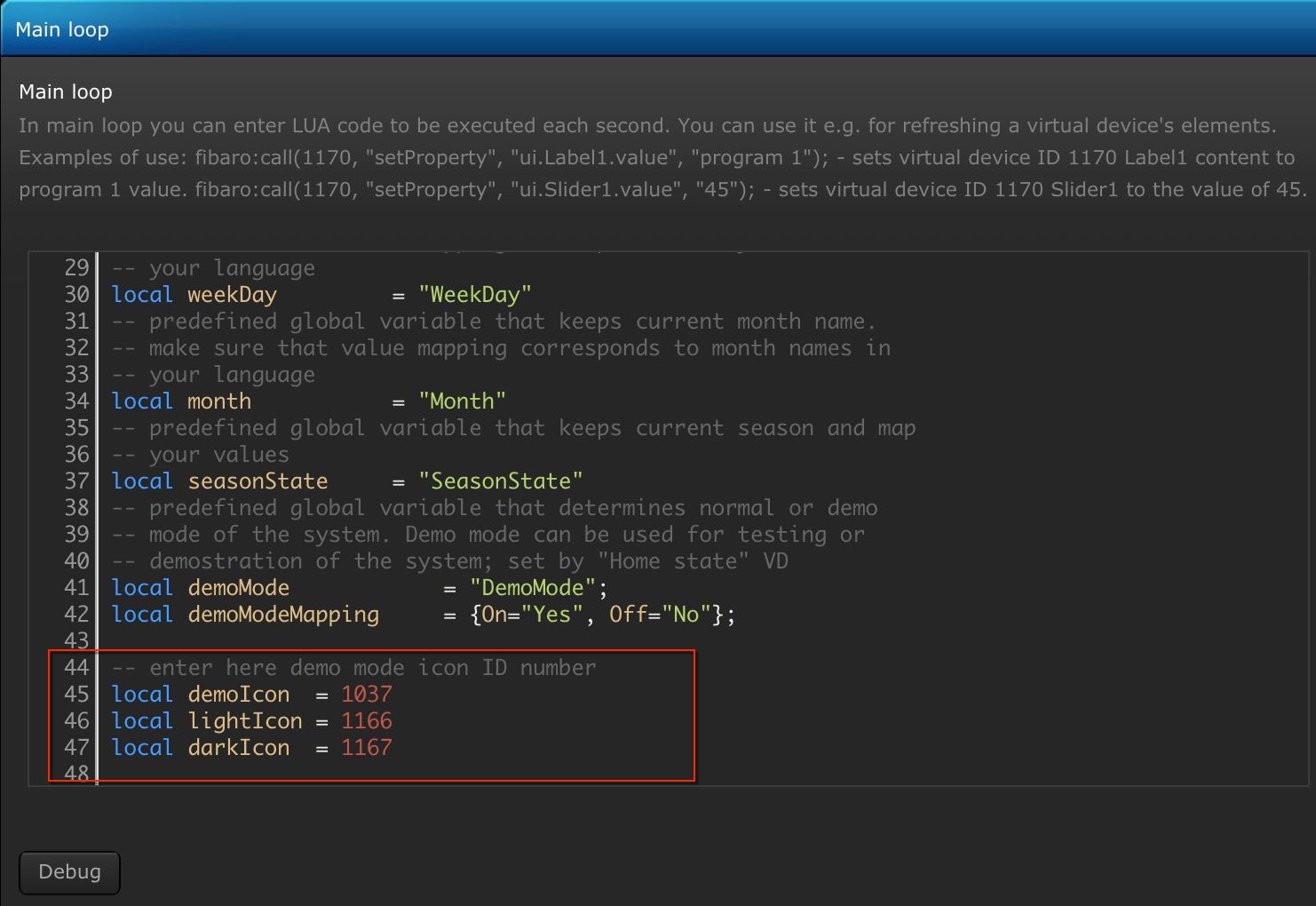
After importing Home Status VD global variable **HomeStatIcon** will be added to global variable panel that remembers VD main icon ID that was selected and is used for function that sets same icons on all VD buttons. As for Sunrise & Sunset VD it is important to setup global variables names and map values in user preferred language in main loop and all buttons code. It is also possible to setup other icon ID’s such as demoIcon and alarm state icons in main loop code:



If any of the global variables are not added to global variable panel then VD will show on labels that is missing.

# HOME TIMERS VD SETUP

After importing Home Timers VD global variable **HomeTimersIcon** will be added to global variable panel that remembers VD main icon ID that was selected and is used for function that sets same icons on all VD buttons. As for Sunrise & Sunset VD it is important to setup global variables names and map values in user preferred language in main loop and all buttons code. It is also possible to setup other icon ID’s such as demoIcon and Darkness state icons in main loop code:



If any of the global variables are not added to global variable panel then VD will show on labels that is missing.

Thank you for using Main scene FTBE and wish you many pleasant moments with your home automation

Author

# APPENDIX 1

## Version History

**UPDATE (23 Nov 2018)**

* **Main scene FTBE 1.3.0** - cleaned some bugs, added some more checks to prevent scene from stop running due to errors. Optimized and compacted code, removed unnecessary comments. DemoMode only stops changing timers while user code and scheduled events will continue to execute. Darkness can now be changed by measured lux level or global variable that is set to lux value measured by light measuring devices placed outside. Scene is now running on setTimeout function instead on sleep for better timing and also aligns to 0 seconds or exact minute.
* **Sunrise & Sunset VD v1.4.0** - Modified main loop code to better handle global variables. Variables are now automatically repaired and are not reset to default settings after HC2 reboot. Added variable SunriseIcon to store VD icon ID and now when main icon for VD is set also all button icons are set automatically.
* **Home Status VD v1.1** - Added variable HomeStatIcon to store VD icon ID and now when main icon for VD is set also all button icons are set automatically. Added label that shows most important values on VD when closed.
* **Home Timers VD v1.1** - Added variable HomeTimersIcon to store VD icon ID and now when main icon for VD is set also all button icons are set automatically. Added label that shows most important values on VD when closed.

**UPDATE (20 Jan 2017)**

* **Main scene for time based events 1.2.5** - since there was some misunderstanding with SleepState usage and TimeOfDay setting, useSleepState is now added and by default set to false. When useSleepState is set to false then Night will change to Morning at surnise time adjusted by marginSunrise so if user wakeup before sunrise then will still be at Night. If useSleepState is set to true then Night will change to Morning when SleepState is set to Awake if it happen before sunrise time. Some minor bugs cleaning.
* **Sunrise & Sunset VD v1.3.1** - thanks to [@Tor Magnus](https://forum.fibaro.com/index.php?/profile/11444-tor-magnus/) replaced time calculation functions and additionally extended adjustment times for Day time and Night time, code cleaning.

**UPDATE (17 Jan 2017)**

* **Sunrise & Sunset VD v1.3**- Corrected main loop code to properly initialize variables if get corrupted by using SAVE button in variable panel. If this happen then it is not needed to delete global variables any more. Just open Sunrise & Sunset VD to edit and click on save button to regenerate global variable values. Corrected marginBlinds range so now it is possible to set from 00:00h till 12:59h. Extended marginSunrise and marginSunset range to +/- 59 minutes

**UPDATE (1 Jan 2017)**

* **Main scene for time based events v1.2.4** - replaced functions for time calculation with ones provided by Fibaro international forum member [@Tor Magnus](https://forum.fibaro.com/index.php?/profile/11444-tor-magnus/). Thank you!!! Added function to change global variable values by scheduled time. Added possibility to define different buttons of one VD to be pressed at different scheduled times. See example bellow.

**UPDATE (21 Dec 2016)**

* **Main scene for time based events v1.2.3** - now reading location of your HC from Longitude and Latitude global variables and properly sets season according to your location. Also added function to setup users for which you want to have GPS location and distance from home calculated. To avoid any mistakes local variables to setup mobile devices for sending push messages are now renamed to **iosDeviceID** and **iosDeviceFlag** since **userID** is now used to setup users which GPS location will be retrieved and distance from home calculated.
* **VD "Sunrise & Sunset" v1.2** - added two more global variables that are automatically added by VD to global variable panel. These are Longitude and Latitude variables that are updated automatically from location of your Home Center after first run of VD main loop. Also added labels to display difference between real sunrise and sunset times and ones user adjust. Labels for Longitude and Latitude also added with button to refresh values if you happen to update your location.

**UPDATE (19 Dec 2016)**

* **VD "Sunrise & Sunset" v1.1** is now adding global variables for you after importing and setting them to default settings! Please first import "Sunrise & Sunset" VD, then add bellow mentioned global variables and you are ready to run Main scene for time based events! For those users who already installed previous version of "Sunrise & Sunset" VD please delete it and replace it with this new version. If you have problems also with time formatting then delete following global variables: MarginSunrise, MarginSunset, MarginDay, DayTime, MarginNight, NightTime, MarginBlinds, OpenBlindTime BEFORE importing new version of "Sunrise & Sunset" VD so that can be regenerated and set properly to default settings by VD.
* **Main scene for time based events v1.2.2** - scene code is also corrected and some bugs cleaned. I corrected code to properly calculate all times and follow changes of the sunrise and sunset. Please copy and paste complete new code due to changes in part for users to edit. I apologize for any inconvenience.
* **VD "Timers" v1.0.1** - corrected bug which prevented from changing Darkness.