

## 11. Emulator on your AAPS smartphone

V 3.0

**Please note that with autoISF you are in an early-dev. environment,** where the user interface is **not optimized for safety** of users who stray away from intended ways to use. Good safety features exist, but these are only as good as the development-oriented user understands and implements them. This is not a medical product, refer to disclaimer in [section 0](#)



### 11.1 Installing the emulator on your smartphone

- 11.1.1 Download QPython 3L
- 11.1.2 .py files in phone internal memory
- 11.1.3 .config file
- 11.1.4 .vdf files
- 11.1.5 Customization of output table

### 11.2 Checking loop decisions on the smartphone

- 11.2.1 Principal purpose
- 11.2.2 Generating **results table**
- 11.2.3 Analyzing results
- 11.2.4 Console running in background

### 11.3 Options available on i-Phone (Trio or iAPS)

### 11.4 Real-time checking a „what-if“ question using **speech synthesis**

[Available related case studies:](#)

Case study 11.1: none available yet

*Skip what is in green writing:*

= Drafted fragments or not implemented ideas.  
Please contribute, or wait for update with the missing info

The emulator **on the PC** was presented in [section 10](#) as

- a very good tool for making your **initial** tuning for a meal spectrum (= when weighing different effects over the entire course of time after each meal, and for a variety of your meals).

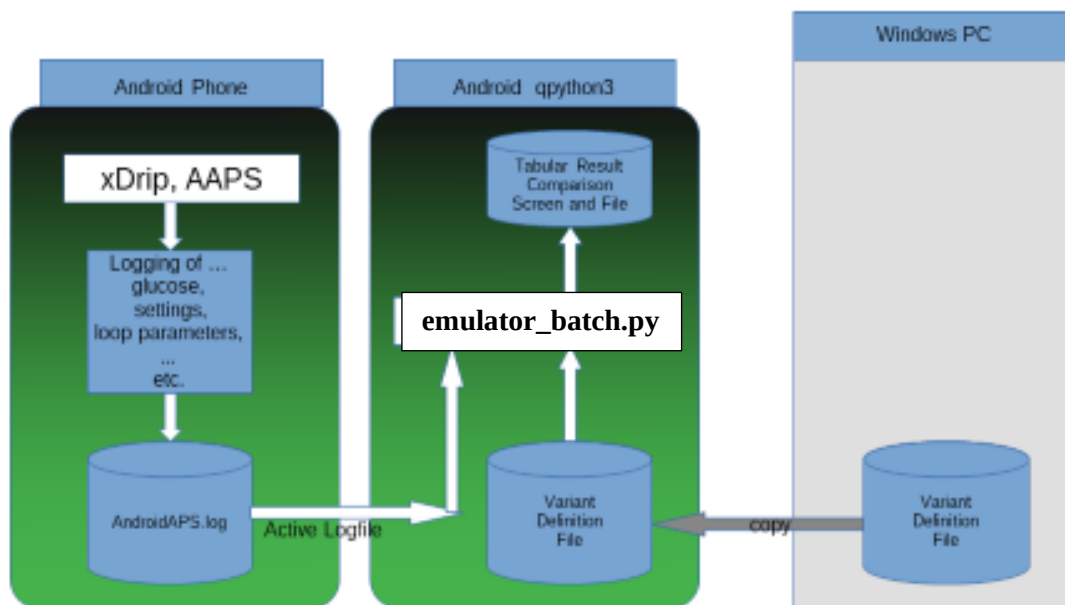
A very useful *additional* tool, is the emulator running **on the AAPS loop smartphone**

- for a quick look how autoISF triggers SMBs after starting a meal ([section 11.2](#)).
- and especially for real-time checking „what-if“ you implemented a specific change idea (developed on your PC, or after analyzing many SMB tabs) ([section 11.4](#)).

Regarding **i-Phone** options, see section 11.3

The emulator for the AAPS phone is described in <https://github.com/ga-zelle/APS-what-if>

Join <https://discord.gg/n3tD5eXExC> for seeking (and giving) help with the Emulator set-up or use, and to share experience.



Sketch of Running the Emulator on AAPS Phone

Github/ga-zelle /  
APS what-if

## 11.1 Installing the emulator on your Android smartphone

Note that iOS based autoISF variants cannot use the Emulator on an i-Phone. However, some tabular outputs of ISF-factors are available (see [section 11.3](#) to be provided by iAPS user)

### 11.1.1 Installing QPython 3L

On your smartphone, go to Google Playstore and download the QPython 3L app. Put the app icon next to your other looping related app icons on the main screen of your smartphone.



With many QPython 3L versions, phone and Android OS versions etc around, you might run into problems and may need to consult detailed installation instructions from [https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4\\_ai3.0.1/Documentation%20in%20English/Installation%20Guide.pdf](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/Installation%20Guide.pdf), or seek advice via <https://discord.gg/n3tD5eXExC>

By long pressing on the QPython 3L app icon, go into “app info” and make the settings like for all your other looping related apps, so they do not get killed all the time by energy savings routines.

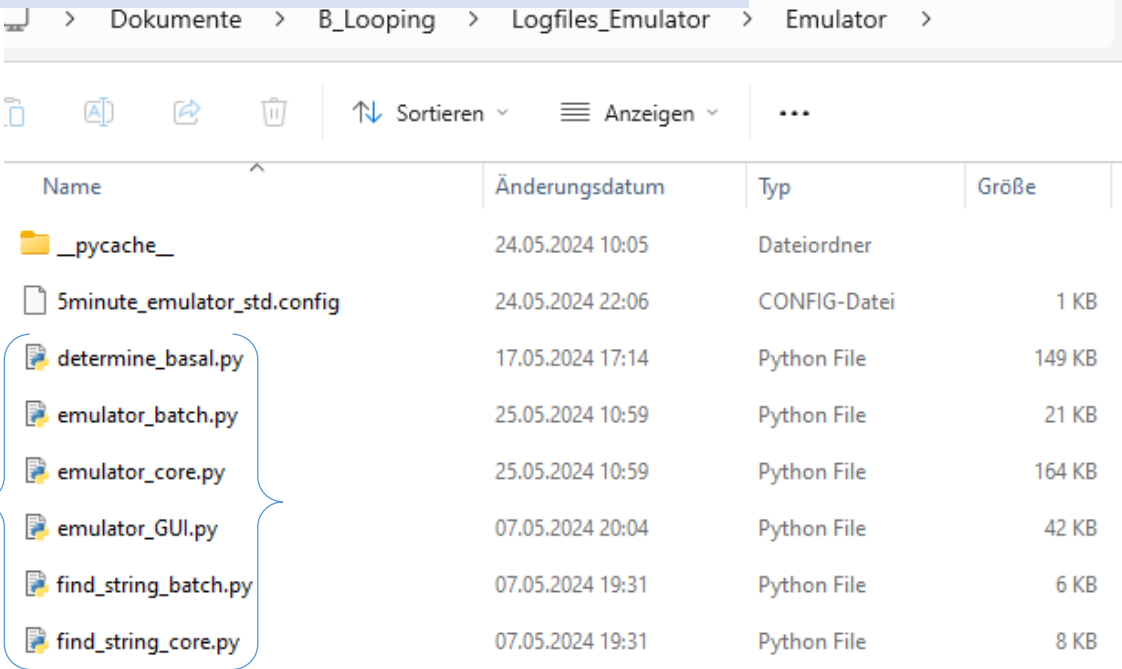
### 11.1.2 Copy .py files from your PC into your phone's QPython/skript3 folder:

1). Connect your phone to your PC for USB data transfer

67 2).Locate your .py files on your PC (in Emulator file).

68

PC: - your path to the Emulator files may differ -



Name	Änderungsdatum	Typ	Größe
__pycache__	24.05.2024 10:05	Dateiordner	
5minute_emulator_std.config	24.05.2024 22:06	CONFIG-Datei	1 KB
determine_basal.py	17.05.2024 17:14	Python File	149 KB
emulator_batch.py	25.05.2024 10:59	Python File	21 KB
emulator_core.py	25.05.2024 10:59	Python File	164 KB
emulator_GUI.py	07.05.2024 20:04	Python File	42 KB
find_string_batch.py	07.05.2024 19:31	Python File	6 KB
find_string_core.py	07.05.2024 19:31	Python File	8 KB

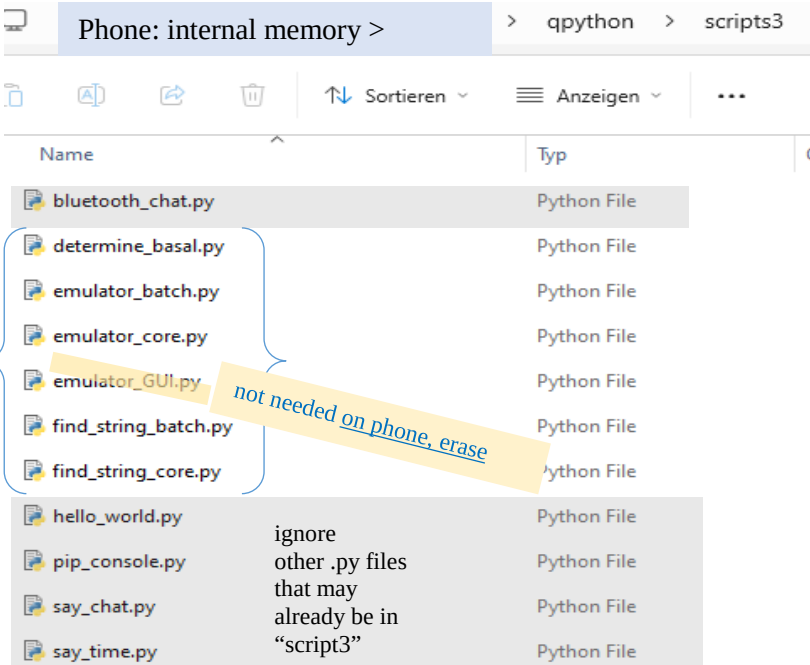
69

70

71 3).Copy all Python related files except emulator:GUI.py from your PC over the internal memory /

72 QPython / Scripts3 of your phone:

73



Name	Typ
bluetooth_chat.py	Python File
determine_basal.py	Python File
emulator_batch.py	Python File
emulator_core.py	Python File
emulator_GUI.py	Python File
find_string_batch.py	Python File
find_string_core.py	Python File
hello_world.py	Python File
pip_console.py	Python File
say_chat.py	Python File
say_time.py	Python File

ignore other .py files that may already be in "script3"

not needed on phone, erase

74

75

76

77

78

79

80 11.1.3 Put configuration file and noChange.vdf into the phone's AAPS logfiles folder

81

82 1). With your phone connected to your PC for data transfer, retrieve **5minute\_emulator.std.config**

83 (or 1minute\_emulator.std.config if you use Libre3/1minute) on your PC, (picture, left)

84

85 2). This config file contains **your "STANDARD"** programming as to which hours of day there can be speech

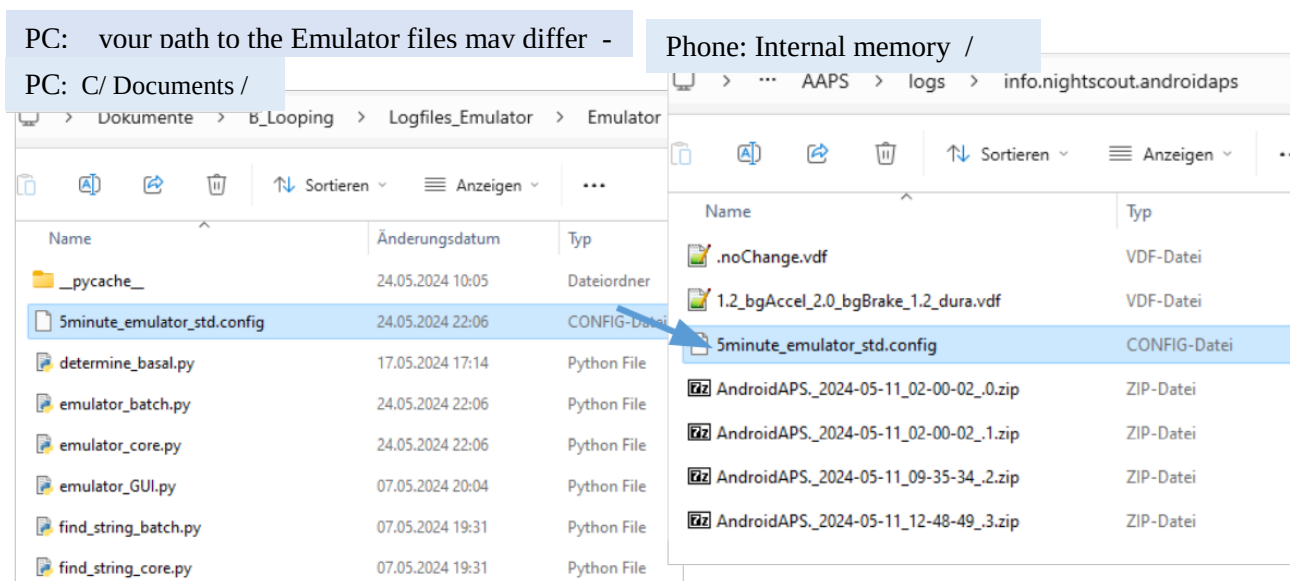
86 messages. Default 07-23 h ("your" time, not "Z"scale). How to change: see in [section 11.4.3](#)

87

88 3). Put a **copy** into the phone into the logfiles (not the QPython!) folder (picture, right):

89 • Internal memory/AAPS/logs/info.nightscout.androidaps

90



91

92 4). You have the option to produce more than just your „...std.config“ file.

93 For instance you could additionally define and load one, that remains silent at carb-related messages, and

94 gives you only insulin (SMB size) related “what-if” suggestions via speech synthesis ; name it for instance

95 „5m\_noCarbsAnnounced.config“ .

96 How to switch between the .config files in a run, see [section 11.2.2](#) step 6)

97

98 11.1.4 Put noChange.vdf into the phone's AAPS logfiles folder

99

100 Retrieve the **noChange.vdf**. on PC in the parallel Studies file

101 • C:\....\Documents\ Looping\ Logfiles\_Emulator\ Emulator\_Studies

102

103 ...and put it on the phone, also into the folder (picture, right):

104 • Internal memory/AAPS/logs/info.nightscout.androidaps

105

106 Later, in [section 11.4.1](#), you will add also (yourChange).vdf files into the same folder. One is  
107 already included, line under noChange.vdf, in picture above, right.

108

109

### 110 11.1.5 Customization of output table

111

112 The table should contain the most relevant information that can be displayed reasonably on most  
113 smartphone screens

114 Consult [https://github.com/ga-zelle/APS-what-](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf)

115 [if/blob/A3.2.0.4\\_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf)  
116 [phone.pdf](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf) in case you see a need to customize .config files and output tables for you.

117

118

119

120

121

## 122 11.2 Inspecting loop decisions on your smartphone

123

### 124 11.2.1 Principal purpose

125

126 The result table from the emulator on your smartphone allows you much easier insight  
127 than the SMB tab can offer into current and recent determinants of given SMB sizes (e.g.  
128 which of the 4 autoISF categories contributed).

129 So, if you would have benefitted from a bigger (or smaller) SMB at times where, say,  
130 bgAccel\_SF was the dominant factor, you would increase (or decrease) the associated  
131 weight. Before actually making such changes, look in other lines of the table to estimate  
132 how this would influence decisions in other time points.

133

134 Testing your hypothesis on the PC ([section 10.3](#)) would allow multi-day multi-(kind-of-)-meal  
135 judgement on feasibility of your tuning idea.

136

137 Before firmly deciding on a settings change, it is advised to run on your phone a what-if  
138 emulation ([section 11.3](#)) using the less aggressive settings for your active loop run, and the  
139 more aggressive defined in (yourChanges).vdf.

140

141

## 142 11.2.2 Generating a results table (for last 75 minutes' loop decisions)

143

144 Note: 1 minute **Libre3 users** will get data only on a significantly shorter time span.

145 15 minutes is really too short to analyze what is going on, and Libre3 users probably should  
146 mostly use the “what-if” part, see [section 11.4](#). That part is principally not impacted (except,  
147 cutting one 5 minute change into 5 very tiny changes, often will make it hard to see and  
148 interpret “what-if” effects).

149

150 Display setting on your phone should be set for automatic switch between landscape and  
151 upright viewing (depending how you hold your phone).

152 Reducing selected font size will *not* help to get more table info on your phone screen, or to  
153 avoid broken/double lines. Go to [11.1.4](#) if, in the following, you are not happy with lay-out.

154

155

156 1). On your main phone screen, press the QPython 3L app icon:

157

158 The first black screen then popping up asks to make a **language** selection

159 In case you don't get an alpha-numerical input field (with <- enter button), touch the  
160 upcoming black screen

161

162 All black screens have a keypad at the bottom:

163

164

165 2). Press your selected number and <- (for enter).

166

167 3). Then proceed in the same screen

168 to make your capital-”N” entry

169 and finally “ <- “ for Next (see illustration)

170

171

172

173 In the following the instructions from the developer how to navigate through the screens:



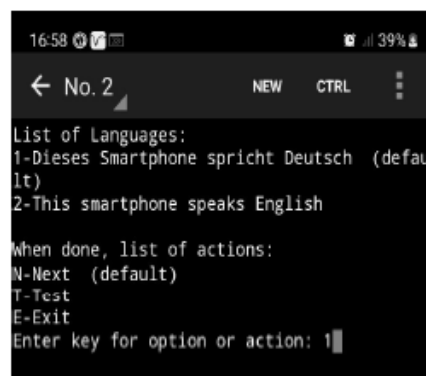
After some Android12 update the previously used GUI dialogues no longer worked and I had to create a keyboard based version similar to the typical telephone dialogues “for option X dial 2”. This dialog system consists of two parts:

- The top part has numerical keys for each option that can be selected. One of those options has **(default)** at the end of the line which indicates that this option is the current selection. If you enter a different number the dialog screen is redrawn and that indicator moved to your new selection. Once your intended selection is OK you focus on the ...
- bottom part, which has letters as keys for the action to be selected. Again, the **(default)** indicator highlights what would be done next if you just press enter without any digit or any letter. Those actions typically are Next, Test and Exit.

The first dialogue is used to select the **language** for the speech synthesis.

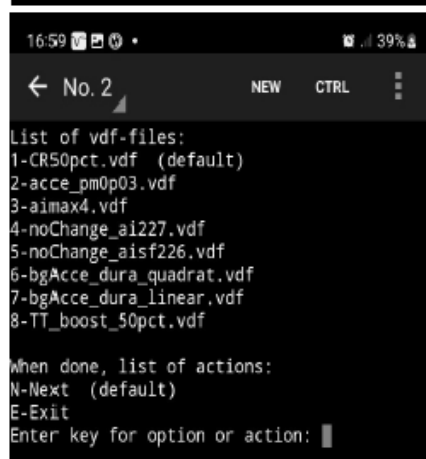
Select „Test“ to listen to a sample speech synthesis.

Select „Next“ to proceed.



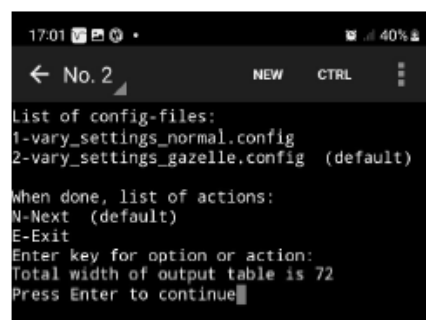
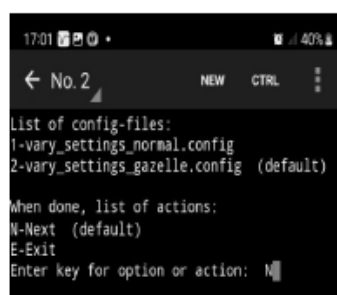
4-5). Repeat steps 2) and 3) with the next screen:

The next dialogue is for selecting your **variant definition file**. All vdf-files found in the logfile folder will be listed.



174 6-7). Repeat steps 2) and 3) with the next screen:

The last dialogue is used to select your favourite **configuration file** with the content discussed in the preceeding section. After you activate the selection you get an informational message of how many columns the selected tabular output will occupy.



8) Before activating the “N(ext)” selection now is a convenient time to rotate the phone to landscape to prepare for the multi column result table display.

Note: Above in step 6), you can also switch between different ...config files, e.g. to silence less important outputs. See also step 4) in [section 11.1.3](#),





178

179

180 9).Now a table comes up detailing the loop decisions on SMB size for the last (15 \* 5 =) 75  
 181 minutes.

182 In case you forgot to do it in step 4), turn the phone now 90 degrees for landscape format;  
 183 in this case, give it 5 minutes for the format to straighten out (after a new value has arrived)

184 In case you want other, or less, columns, see [section 11.1.4](#).

01:39 84 58%

← N

1

2

3

4

5

6

7

8

9

III

UTC

eff.

acce

bg

pp

dura

-----ISFs-----

insulin Req

---SMB---

time

bg

IOB

iobTH

ISF

ISF

ISF

orig

prof

emul

orig

emul

orig

emul

23:23Z

97

0.93

4.0

-0.28

0.65

1

1

375

150

375.0

0

0

0

0

23:24Z

94

0.9

4.0

-0.44

0.61

1

1

375

150

375.0

0

0

0

0

23:25Z

92

0.89

4.0

-0.21

0.58

1

1

375

150

375.0

0

0

0

0

23:26Z

92

0.85

4.0

0.25

0.58

1

1

375

150

375.0

0

0

0

0

23:27Z

91

0.84

4.0

0.38

0.7

1

1

375

150

375.0

0

0

0

0

23:28Z

89

0.82

4.0

0.4

0.67

1

1

372.8

150

372.8

0

0

0

0

23:29Z

88

0.79

4.0

0.43

0.65

1

1

348.7

150

348.7

0

0

0

0

23:31Z

86

0.77

4.0

0.54

0.61

1

1

275.3

150

275.3

0

0

0

0

23:32Z

85

0.74

4.0

0.63

0.6

1

1

248.2

150

248.2

0

0

0

0

23:33Z

90

0.72

4.0

0.63

0.68

1

1

239.8

150

239.8

0

0

0

0

23:34Z

91

0.69

4.0

3.53

0.71

1

1

59.9

150

59.9

0

0

0

0

23:35Z

87

0.67

4.0

0.78

0.64

1

1

234.1

150

234.1

0

0

0

0

23:37Z

85

0.65

4.0

1.08

0.61

1

1

229.7

150

229.7

0

0

0

0

23:38Z

86

0.63

4.0

0.79

0.62

1

1

241.5

150

241.5

0

0

0

0

23:38Z

88

0.62

4.0

1.2

0.65

1

1

192.1

150

192.1

0

0

0

0

Waiting 60sec for next loop at 01:40; Variant "noChange"

185

186 The line at the bottom of the table says the time (hh:mm) when the next bg result and loop  
 187 decision are expected. Also it shows the (yourChange).vdf file investigated

188

189 In sequence of the time(Z!), the table consists default of the columns with info on (1) bg (2)  
 190 IOB and eff.iobTH (3-6) the adaptation factors on ISF suggested by the 4 autoISF  
 191 categories (7) resulting ISF that was used, profile\_ISF, and emul ("what-if" ISF) (8)  
 192 insulinRequired calculated by your running loop, and "what-if" result (9) same for resulting  
 193 SMB

194



195 The columns marked “**orig**”.in the table of results shows the ISF actually used to  
196 determine insulinReq and SMB size in the actual run.

197

198 In the columns marked “**emul**” you find the calculated emulation results that is calculated  
199 every 5 minutes

200 • If a (yourChange).vdf was clicked (in step 2), see also bottom right of the table), the emul  
201 columns show the result, how the investigated changes *would have* changed SMB sizes (strictly  
202 always looking at the one decision, in that line of the table).

203 • If only a noChange.vdf was run, the emul columns contain the same results as the orig.  
204 columns.

205

206 **Caution when interpreting the values in the acce, bg, pp and dura ISF columns (3-6)**

207

208 The factors given there are always those for the emul run.

209 • So when using (yourChange).vdf, you see in your phone's table exclusively the ISFs that  
210 *would result-if* (yourChange) were made.

211 We get to “what-if” projects later, in [section 11.4](#).

212 • Only in the noChange scenario, the values there would be the *orig ones, corresponding to*  
213 *what could be seen also in the SMB tab* at the times

214 You easy recognize whether you run the noChange: it says so at the bottom of the table.  
215

216 orig.ISF is called **sens** in the SMB tab, a couple of lines below “end autoISF”. It is the ISF  
217 that replaces, for that one decision you are looking at, the profile ISF (called profile.sens in  
218 the SMB tab),

219 We get back to this topic at end of [section 11.4.2](#) “Understanding how the ISF is emulated by  
220 (yourChange) and how SMB or TBR would differ”

221 “

222

223

## 224 11.2.3 Analyzing the results

UTC	bg	IOB	iobTH	eff. ISF	acce ISF	bg ISF	pp ISF	dura ISF	-----ISFs-----	insulin Req	---SMB---
time	bg			ISF	ISF	ISF	ISF	ISF	orig prof emul	orig emul	orig emu
13:53:42Z	116	2.19	6.0	1.54	1.01	1	1	1	26.1 40 26.1	0 0	0 0
13:58:42Z	120	2.0	6.0	2.08	1.01	1.12	1.04	1	19.2 40 19.2	0.16 0.16	0.1 0.1
14:03:57Z	117	1.97	6.0	1.37	1.01	1	1.06	1	28.5 39 28.5	0 0	0 0
14:08:43Z	113	1.81	6.0	1.18	1.0	1	1.08	1	33 39 33.0	0 0	0 0
14:13:41Z	115	1.64	6.0	1.11	1.01	1	1.1	1	35.2 39 35.2	0 0	0 0
14:18:42Z	125	1.47	6.0	1.86	1.02	1.3	1	1	20.9 39 20.9	2.3 2.3	1.7 1.7
14:23:46Z	131	3.19	6.0	1.44	1.03	1.18	1	1	27 39 27.0	0.26 0.26	0.1 0.1

225 In above example (table), all given SMBs were driven by bgAccel\_ISF, when glucose rose.  
 226 The biggest SMB in the time we are looking at (actual local time = Z + 2 hours, so we are  
 227 looking at late small rises, like 3 hours after lunch) was 1.7 U = 0.74 SMB delivery rate \*  
 228 2.3 U insulinRequ.

230 The insulinRequ. Is proportionally amplified by the effective ISF, called sens in your SMB  
 231 tab, or ISF“orig” in this table. The amplification of 39 profile\_ISF / 20.9 effective\_ISF = 1.86  
 232 happens to be the dominating bgAccel\_ISF amplification factor

233  
 234 Note **not** always just the biggest ISF factor “wins”. Consult flowchart in [LINK](#), and  
 235 occasionally read the real-time explanations in your SMB tab as to which other factors are  
 236 contributing to the amplification resulting from profile\_ISF to effective\_ISF (“orig”, or sens).  
 237 For instance, the Activity Monitor, or a set %temp. profile, or TT, could contribute, or also  
 238 the question whether glucose already decreases.

239  
 240 Depending on your selected “safety” settings, you might occasionally bump into  
 241 restrictions. **Tuning up factors that make the system bounce into restrictions is a**  
 242 **completely useless, and potentially even dangerous, exercise!**

243  
 244 Therefore, **for your initial set-up** of parameters ([section 2](#) and [section 4](#) of the FCL e-  
 245 book) it is **advisable to not do this analysis** on the smartphone, but **on the PC**, where  
 246 you can inspect the complete info on each loop decision (see [section 10](#))..

247 If for some reason you cannot do this on the PC, you must frequently take screenshots (in  
 248 very many decisive 5 minute segments), and analyze **more complete data**, than the table  
 249 on our smartphone could offer, **in the SMB tab)**

250

251 In your initial set-up of your FCL, you probably must “loop” a couple of times back into  
252 [section 2](#) to adjust the safety settings made for max. SMB sizes.  
253  
254 **11.2.4 Console running in background**  
255  
256 When you are finished analyzing real-time coming up tables, you can either close the  
257 emulation program, or (in a box “Run console in background”) you can opt for letting it run  
258 in the background.  
259  
260 This is required if you want the speech synthesis give information at times when a bigger  
261 SMB would have been suggested by the (yourChange).vdf emulation. Also you might hear  
262 suggestions by when you should eat how many g of carbs (more see in [section 11.4.3](#)).

263

264 **11.3 Options available in Trio or iAPS**

265

266

267 iAPS / Trio offer in their autoISF variants also

268 access to a tabular representation of

269 **autoISF contributions to resulting SMB sizes:**

270

271

272

273

274

275 The emulator will not work in the iOS world.

276

277 Note that on the iPhone, so far, the what-if emulation

278 and speech synthesis (see next [section 11.4](#) for AAPS)

279 are currently **not** possible. .

280

281

282

283

284

285

286

287 **More description must be provided by iAPS / Trio user please**

288

289

290

291

292

autoISF History								Sulje	
30.5.2024 13.00 2 hours ↕									
Time	BG	ISF factors		bg	pp	dura	Insulin		req.
		final	acce				SMB	TBR	
12.16	6,5	0,5	0,1	1	1	1	0	0	0
12.11	7,2	1,07	1	1,01	1	1,07	0	0	0
12.06	7,3	1,07	1,07	1,01	1,07	1,05	0	0	0
12.00	7,1	1,01	1	1	1	1	0	0	0
11.54	7,5	1,18	1	1,02	1	1,18	0	0	0
11.48	7,9	1,22	1	1,04	1	1,22	0	0	0
11.40	7,9	1,18	1	1,04	1	1,18	0	0	0
11.30	7,9	1,12	1	1,04	1	1,12	0	0	0
11.22	8,1	0,5	0,1	1,05	1	1,07	0	0	0
11.17	8,3	0,75	0,71	1,06	1,02	1	0	0	0

## 11.4 Real-time checking a „what-if“ question using speech synthesis

The emulator on your **Android** smartphone can help clarify "what if..." you implemented a considered change.

### 11.4.1 Put a (yourChange).vdf into the phone's AAPS logfiles folder

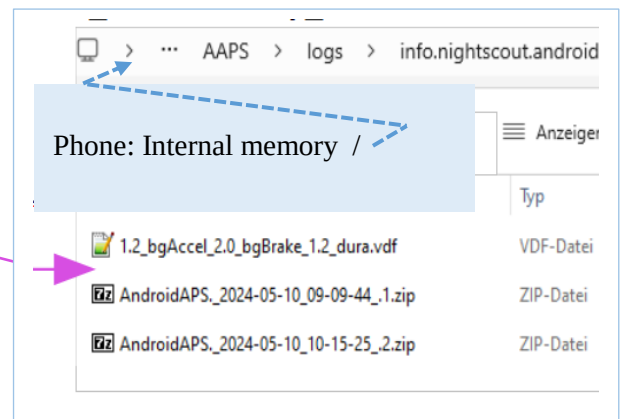
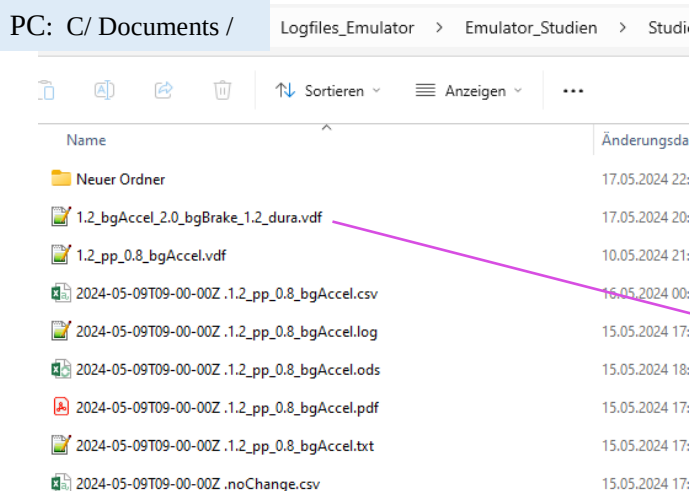
In running the emulator on the phone, you can define in the .vdf file of the emulator, which setting(s) you would like to be differently aggressive than in the active AAPS.

How to write .vdf files was already explained in the section "Emulator on PC". See in end of [section 10.2.1](#). You could also pull a vdf file example from the developer's Github that you could customize further: Access see [section 3.8 /5](#).

Produce or retrieve **(yourChange).vdf**. on PC from one of your studies files.

- C:\....\Documents\ Looping\ Logfiles\_Emulator\ Emulator\_Studies\ Study\_n

PC: your path to the Emulator files may differ -



... and put a **copy** of that .vdf into the **smartphone folder with the AAPS logs** :

- Phone: Internal memory/AAPS/logs/info.nightscout.androidaps

### Switch between python scripts running at the same time

In case you have **more than one** (yourChange).vdf to investigate, you can look at the data of your currently running loop (last 15\*5 minutes) by just switching between the related vdf files used for emulation.

322 Details see [section 3.8, 9](#)) – or try to go direct via [https://github.com/ga-zelle/APS-what-](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf)  
323 [if/blob/A3.2.0.4\\_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf)  
324 [phone.pdf](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf) and there p.5, under above sub-headline “.Stop the emulator, or switch...”

325

326 [Stop the emulator](#)

327 (see in paper as above)

328

329 [11.4.2 Inspect emulated results](#)

330

331 Now, whenever you run QPython 3L emulation (following the steps as described in [section 11.2.2](#))  
332 you additionally get the emul. results filled in in the resulting table, showing in which time points  
333 your change would lead to adapted insulinReq, and SMB size data.

334

23:57

<

etc.

21:11:17Z	108	1.06	6.0	1.12	1.0	1	1.08	38.3	43	38.3	0	0	0	
21:16:18Z	109	0.94	6.0	1.1	1.0	1.03	1.07	39.1	43	39.1	0.1	0.1	0	
21:21:20Z	109	0.87	6.36	0.95	1.0	1	1.08	41.9	43	39.5	-0.1	-0.08	0	
21:26:20Z	108	0.76	6.36	1	1.0	1	1	43	43	40.6	0	0	0	

335

336 etc

337 In the marked 21;21 Z line, dura\_ISF was the dominant factor. The (yourChange).vdf would apply a  
338 factor of 1.2 and lead to further strengthening the ISF: profile 43 -> orig.(noChange) 41.9 -> emul  
339 (yourChange) 39.5

340 In this case, late after a meal, and bg=109 mg/dl, the loop saw in the orig. (noChange) case 0.1 U  
341 insulin too much; and as the (yourChange) emul case asks for typically more insulin (all weights in  
342 the (yourChange).vdf are > 1), now only 0.08 U are seen as too much (a 20% difference).

343

344 The table on your phone is too reduced to show each emulated ISF component. If you need to see  
345 more details on how (yourChange).vdf would make a change in a point of time that you like to  
346 analyze deeper: Note down interesting Z times, and later look it up in more detail in the Emulator  
347 on the PC:

348

below are "orig."ISFs:

UTC		eff.	acce	bg	pp	dura	-----ISFs-----			insulin Req		---SMB--		
time	bg	IOB	iobTH	ISF	ISF	ISF	ISF	orig	prof	emul	orig	emul	orig	emu
21:21:20Z	109	0.87	6.36	0.95	1.0	1	1.08	41.9	43	39.5	-0.1	-0.08	0	

### Understanding ISF orig.(the sensitivity used to determine insulinRequ)

To fully understand how acce, bg, pp and dura\_ISF determined the orig.ISF, we do an **emulator** run **on the PC**, and look up the **.txt** result:

@ 21:21 Z / Script Debug -----

Activity monitor disabled inactivity detection: sleeping hours; Autosens ratio: 1;

Basal unchanged: 0.45; ISF unchanged: 43 CR: 9

-----

start autoISF 3.0.1

-----

Loop allows APS power level; SMB enabled due to enableSMB\_always

acce\_ISF adaptation is 0.95

bg\_ISF adaptation is 1

pp\_ISF adaptation is 1

dura\_ISF adaptation is 1.08 because ISF 43 did not do it for 30 m

strongest autoISF factor 1.08 weakened to 1.03 as bg decelerates already

final ISF factor is 1.03

-----

end autoISF

-----

profile.sens: 43 sens: 41.9

Note: 43 / 1.026 = 41.9

### Understanding how the ISF is emulated by (yourChange) and how SMB or TBR would differ

While the result for ISF\_emul (and for the SMB size) is given in the table on your phone, the details behind it, and also for finer effects in %TBR, can only be inspected from the .csv table from the (yourChange),vdf **emulator** run on the **PC**:



383 The following table is an extraction of the most relevant data from the (yourChange).csv:

	B	C	E	F	L	Q	R	S	AA	AB	AC	AD	AE	AF	AG	AH	AI	AP	AQ
1			bg	bg		final	dura		acce	bg	pp	delta	dura	final					
2	UTC	AAPS	accel	brake		ISF	min-	dura	ISF	ISF	ISF	ISF	ISF	ISF	ISF	ISF	ISF	TBR	TBR
3	time	time			iob	orig	utes	avg.	emul	emul	emul	emul	emul	emul	orig	prof	emul	orig	emul
29	21:16:18	23:16	109		0,94	1,1	25	108,2	1,12	1	1,03	1	1,08	1,12	39,1	43	38,4	0,65	0,65
30	21:21:20	23:21	109	109	0,87	1,03	30	108,3	0,9	1	1	1	1,1	1,04	41,9	43	41	0	0,46
31	21:26:20	23:26	108		0,76	1	5	108,5	1	1	1	1	1	1	43	43	40,6	0	0
32	C=B+n/24 @ n=+2 h time diff.										If there were bigger orig->emul effects, we would show SMB columns, too								

384

385 It shows that the final (noChange) ISF factor of 1.03 (box Q30) becomes 1.04 (box AF30) in the  
 386 (yourChange) emulated case. Given that, so late after dinner (box C30), we sit near the 90 mg/dl  
 387 target (box E30), the difference the changes in the emulated case are minor. Still zero SMB, but  
 388 the minor changes reflect in elevating the temp. basal rate for fine adjustment (box AP30 -> AQ30).

389

390 Note that the orig. values for the four autoISF components (acce, bg, pp, dura "ISF orig")  
 391 are **not** contained in this csv table (only, in Q30, the final amplification factor for final\_ISF in  
 392 the noChange scenario).

393 However, you can fetch the autoISF details for 21:20 Z in the actual "noChange" run:

- 394 • from the txt result file (as shown above); or
- 395 • you also see them on the phone:

396

orig	orig	orig	orig	ISF
acce	bg	pp	dura	-----
ISF	ISF	ISF	ISF	orig
0.95	1.0	1	1.08	41.9

397

398 ...while the emul\_ISFs come from .csv results @ (yourChange).vdf run on the PC:

AA	AB	AC	AE	AI
acce	bg	pp	dura	
ISF	ISF	ISF	ISF	ISF
emul	emul	emul	emul	emul
1,12	1	1,03	1,08	38,4
0,9	1	1	1,1	41,2

399

400

401 Apology: The above example was not well chosen to see relevant effects. The author is struggling to  
 402 put this chapter first time together, quickly for the V.3.0.1 launch, and just picked from his phone what  
 403 was available at the moment, I might patch it over with a better example in a later update, or I (or  
 404 maybe you?) provide an adjunct case study.

405

406

407

## 408 Principal limitation of any of your emulations

409

410 Note that always the first biggest change regarding insulinRequ and SMB size in emul vs. orig. is  
411 the most relevant. This is because:

- 412       ○ Doing that change would change, by the same amount, the iob basis for the next  
413       following loop decision
- 414       ○ Doing that change would certainly change the course of the bg curve about half an hour  
415       afterwards (to be precise: for the duration of DIA, for the extra insulin), so then it is  
416       anybody's guess, for instance how difficult of a job presents itself to the loop to "attack"  
417       with duraISF.

418

419 This is one of the reasons why setting up your FCL ([section 4](#)) should be a iterative process,  
420 seeking solutions (with mainly bgAccel\_ISF) for the first rise (and a range of different meals) first,  
421 AND NOT concurrently already "tuning" the dura\_ISF.

422       Both, bg peak height, and pattern of insulin activity from the present iob (that is only sluggishly  
423       further adjustable driven by dura\_ISF and evtl. by bg\_ISF) depend on how the first bg rise stage was  
424       managed by autoISF (your bgAccel\_ISF\_weight setting, and others).

425

426 Another conclusion you might draw, is to limit use of the emulator *on the PC* largely to analyzing  
427 the (in FCL extremely important) first rise, to seek bgAccel\_ and pp\_ISF\_weights.

428

429 Then to test these, and many other like ideas for changed settings "in real life" emulated *on the*  
430 *smartphone*: A synthetic voice will announce to you how (if) each actual decision would differ. See  
431 next section!

432

433

### 434 11.4.3 Real time speech synthesized treatment suggestions

435       Unfortunately this great feature is **not available on i-Phone**. Look in (updates of) [section 11.3](#) for  
436       eventual alternatives.

437

438 At time points when the (yourChange) setting would result in smaller or greater difference in SMB  
439 insulin delivery (compared to the real "noChange" run), you can get a real-time notification **via**  
440 **speech synthesis**, and you can assess the situation in real-time yourself.

441

442       Also if just the noChange.vdf is running (and even in AAPS without autoISF) , the speech synthesis could alert  
443       you to "carbs required" messages, for instance.

444

445 If for instance a suggested extra, or bigger, SMB makes sense, **you can add this portion**  
446 **manually\* ....and observe**, for this meal, whether this bolus was OK and you should switch in

447 direction of the different setting you were investigating (which would automatically give you that  
448 extra in the future).

449 Likewise, you might choose **not** to intervene, but regret it an hour or so later, seeing the further  
450 development **without** implementing the supposed improvement.

451

452 *\*In Full Closed Loop, you don't need any buttons at the bottom of the AAPS main screen. But for such*  
453 *test phases it is practical to re-install the insulin button at the bottom of the AAPS main screen*  
454 *(Preferences/Overview/Buttons/Insulin -> ON).*

455

456 After a couple of days, you will get a feel for whether you want to incorporate your  
457 investigated change (or a gradual step towards it) into your active AAPS settings.

458

459 **Warning:** Your settings must always work for a variety of meals. Do not put too  
460 much effort into optimizing one situation! (See [case study 8.2](#)).

461

## 462 **Activating and silencing emulator suggestions**

463

464 For silencing the suggestions from voice synthesis you have the following options:

465

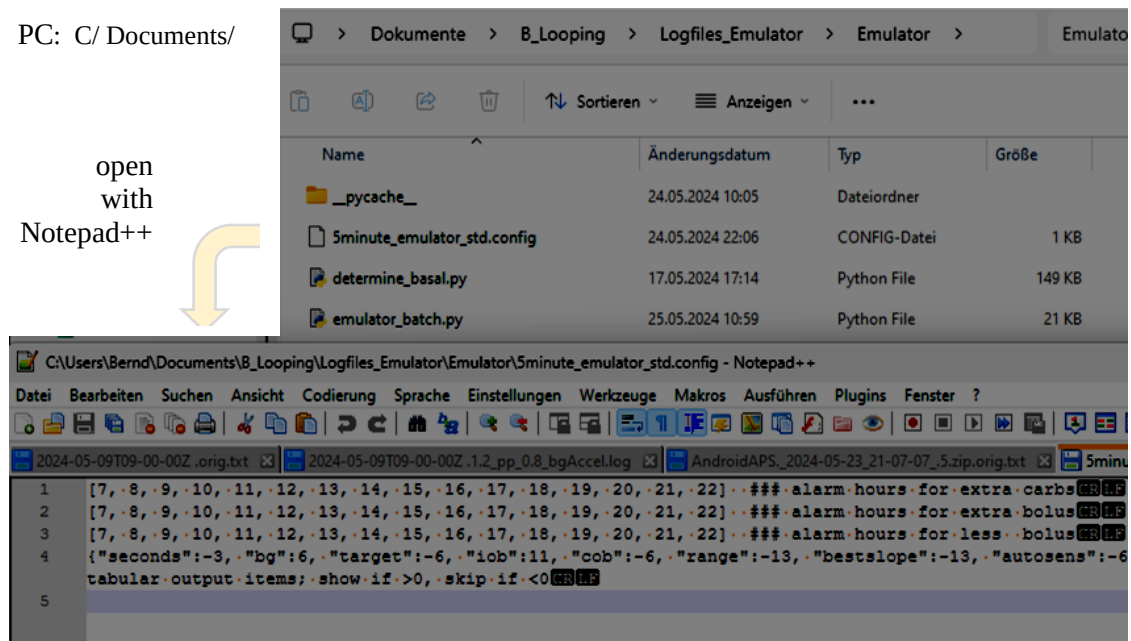
466 (1) Change **principal settings** what shall be announced (e.g. only if bigger SMB size  
467 is suggested, or also warnings about carbs eventually needed?), and in **which hours of**  
468 **day**, to make any announcements via speech synthesis .

469 These are set in the (1 or) 5minute\_emulator\_std.config file:

470 Go on **PC** into the config. file (see illustration). Open it with Notepad++ and edit the  
471 hours there for when you would want (no) announcements regarding: extra carb  
472 need (line 1), extra bolus need (line 2), or less bolus (line 3):

PC: C:/ Documents/

open  
with  
Notepad++



Save the changes, and copy the file also into your **phone** at Internal memory/AAPS/logs /info.nightscout.androidaps (see [11.1.3](#)) over the 5minute\_emulator\_std.config

(2) Turn off **phone volume** (silence media + switch on do not disturb)

Of course, this also shuts off many other potential alerts that you might not want to shut off.

(3) **“Kill” (and later resume) the “what-if” emulation.** This could be done by de-selecting the (yourChange).vdf in step . However, this stops (or interrupts, until you re-start) the entire emulation and you will have no tabular data later for the silenced time.

(4) **De-activate Qpython 3L app** (temporarily?): Press on Qpython 3L app icon, force close or remove necessary permission – re-activate (or need to re-start emulation then ??) when you want to hear again .. However, this stops (or interrupts, until you re-start) the entire emulation and you will have no tabular data later for the silenced time

(5) ☐ Probably better than the 2 afore mentioned options would be to **run (for the intended silencing period) exclusively the noChange.vdf**, Then you will not get any speech outputs (because you are NOT investigating a what-if question, in that case). But you get all data (the un-interrupted noChange actual run) and later on the PC still can investigate any “what-if” scenarios

How to change the .vdf reference during a run, see [section 3.8, 9](#) – or go direct to the instructions in the Github repo at: [https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4\\_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf](https://github.com/ga-zelle/APS-what-if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-phone.pdf) and there p.5, under above sub-headline “Stop the emulator, or switch...”

499 (6) A variation of option (5) would be to silence all but the (less frequently occurring but  
500 most important) one line in your config definition (done according to [section 11.1.5](#))  
501 Then go through step 6) in [section 11.2.2](#), you can also switch between different ...config files, e.g.  
502 to silence the less important outputs. See also step 4) in [section 11.1.3](#),

503

504 Note that this area (silencing) has not been researched much, and good answers are likely  
505 lifestyle and phone specific.

506

507

508 Please share your experiences with the emulator in Discord / Full-Closed-Looping / HOW  
509 TO /\_emulate-aaps, at: <https://discord.gg/n3tD5eXExC>

510

511

512