

11. Emulator on your AAPS smartphone

V 2.6 / draft
screens too dark

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

[Available related case studies:](#)

11.1.1 Download QPython 3L

Case study 11.1: still missing

11.1.2 File structure on your internal memory

11.1.3 config und vdf files

11.1.4 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

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

Green writing
= drafted stuff,
best wait for update with the missing info

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

11.5 Exploring phone data further on PC

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).

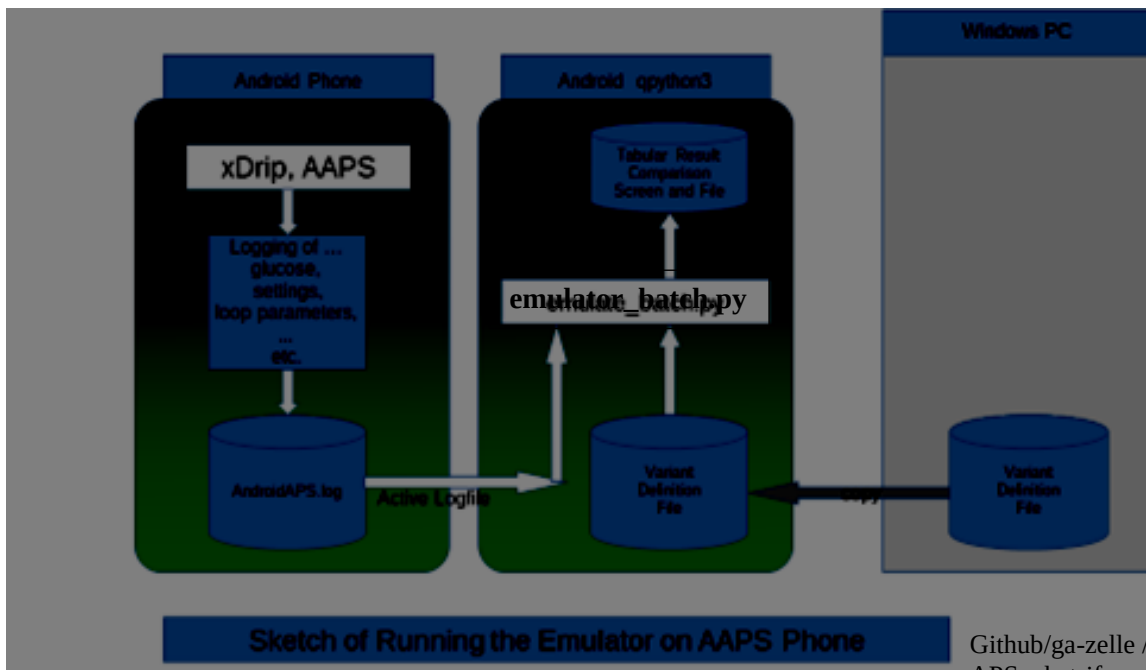
A very useful *additional* tool, is the emulator running **on the AAPS loop smartphone**

- 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](#)).

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

There you find the files needed to download on phone or PC, and the primary instructions.

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



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

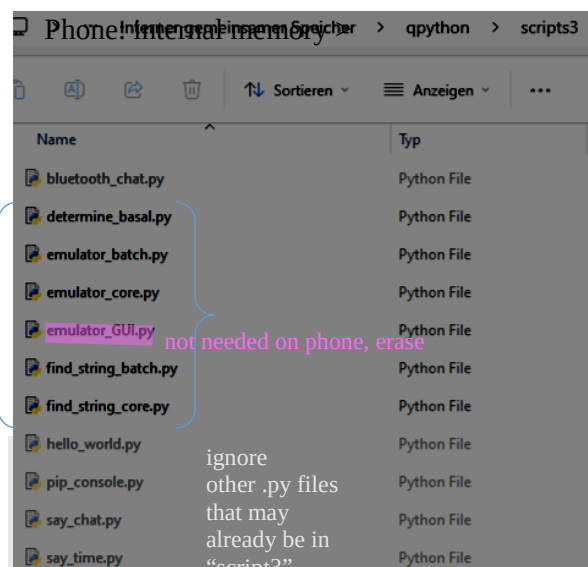
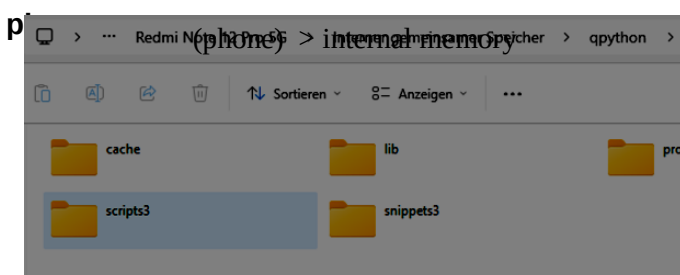
On your smartphone, go to Google Playstore and download the QPython 3L app. Put the app icon next to your other looping related app icons on the main screen of your smartphone.

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 [Github/ga-zelle / APS what-if](#), or seek advice via <https://discord.gg/n3tD5eXExC>

11.1.2 Copy .py files from your PC into your phone's qpython folder:

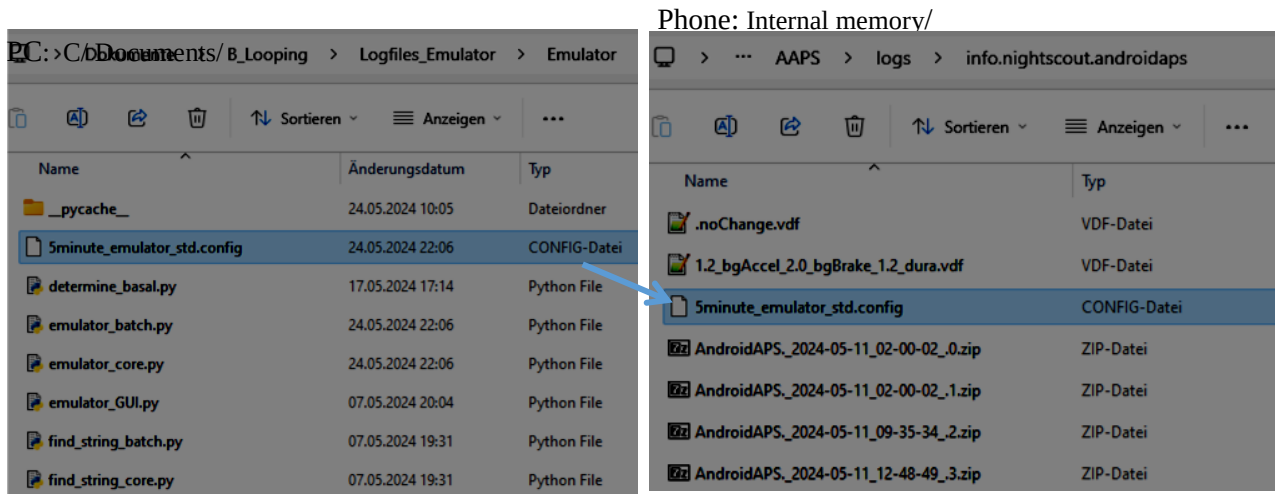
Connect your phone to your PC for data transfer and locate your .py files on your PC (in Emulator file, see next page).

Copy all Python related files **except emulator:GUI.py** into the internal memory/QPython/Scripts3 of your phone:



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

With your phone connected to your PC for data transfer, retrieve **5minute_emulator.std.config** (or 1minute_emulator.std.config if you use Libre3/1minute) in ...



...PC, (picture, left), and put a **copy into** the phone into the folder (picture, right):

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

This config file contains the programming as to which hours of day there can be speech messages. Default 07-23 h ("your" time, not "Z"scale). How to change, see in [section 11.4.3](#)

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

- C:\.....\Documents\ Looping\ Logfiles_Emulator\ Emulator_Studies

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

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

11.1.4 Customization of output table

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

Consult Github/ga-zelle/APS-what-if (newest branch): Instructions Emulator in case you see a need to customize .config files and output tables for you.

95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133

11.2 Inspecting loop decisions on your smartphone

11.2.1 Principal purpose

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

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

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

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

11.2.2 Generating a results table re.recent 75 minutes loop decisions

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

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

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

On your main phone screen, press the QPython 3L app icon:

In case you forgot to erase emulator:GUI.py in 11.1.2, press Programs, then select emulator_batch.py from the list that pops up, and press Run

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

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

Press your selected number and <- (for enter). Then proceed in the same screen to make your "N" entry for Next + "<-" for enter

insert

134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162

insert screens

2) Now a next similar looking screen comes up for **selecting your ,vdf** underlying the kind of analysis you are trying to do.

Again, enter your selected number.

In case you have already more than just your (noChange).vdf file on board, this 2nd screen will ask you to make a selection between (noChange).vdf or any of the (yourChange).vdf you put in (see [section 11.4](#)).

Then proceed in the same screen to make your "N" entry for Next + "<-" for enter

3) Now a third screen come up for selecting a configuration file.
Press your selected number and <- (for enter). Then procees in the same screen to make your "N" entry for Next + "<-" for enter

4) It now gives you an info about width of the upcoming table (default 89)

Now turn your phone 90 degrees into landscape position, and press ">-" to continue

163

164

165

166

167

168

169

170 5) Now a

171 up detailing

172 decisions on

173 the last (15 *

174 minutes.

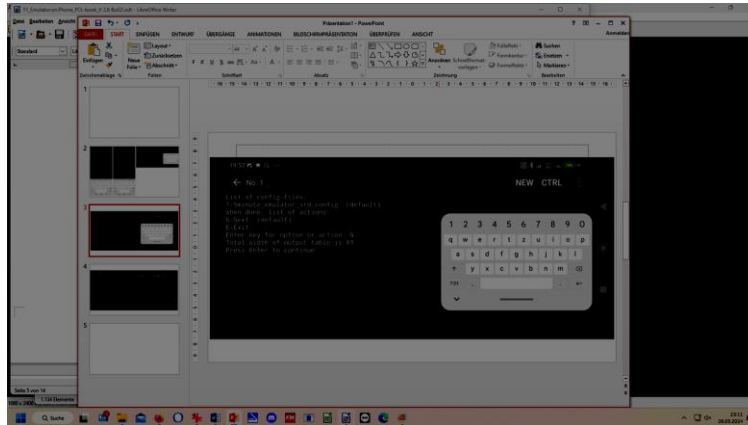


table comes

the loop

SMB size for

5 => 75

175

In case

you forgot to do

176

it in step 4), turn the phone now 90 degrees for landscape format; in this case, give it 5

177

minutes for the format to straighten out (after a new value has arrived)

178

If the default 89 characters width does not suit you, see [section 11.1.4](#). how to customize

179

180 The line at the bottom of the table says the time (hh:mm) when the next bg result and

181 loop decision are expected. Also it shows the (yourChange).vdf file investigated

182

183 The table consists default of the columns:

184 time(Z!)/bg/IOB/eff_jobTHthen in 7 following colums all ISFs:

185 bgAcce-/bg/_pp/_dura/_eff.ISF(noChange)/profile_ISF/emul.ISF(yourChange)...

186 ...then insulinRequ and SMB delivered (

A screenshot of a mobile phone screen displaying a table of data. The table has columns for time, UTC, eff., acce, bg, pp, dura, ISFs, insulin Req, and SMB. The data is presented in a grid format with rows of numerical values. The phone's status bar at the top shows the time 17:05 and various icons for connectivity and battery.

187

188

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

191

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

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

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

199 Note that the values reported in the **acce, bg, pp and dura ISF columns** are always for
200 the emul run.

201 • So in the first case, when using (yourChange).vdf, you see in our phone’s table exclusively
202 the ISFs that *would result-if* (yourChange) were made.

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

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

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

209

210 [11.2.3 Analyzing the results](#)

211

212 In above example (table), all given SMBs were driven by bgAccel_ISF, when glucose rose.
213 The biggest SMB in the time we are looking at (actual local time = Z + 2 hours, so we are
214 looking at late small rises, like 3 hours after lunch) was $1.7 \text{ U} = 0.74 \text{ SMB delivery rate} * 2.3 \text{ U insulinReq}$.

216 The insulinReq. Is proportionally amplified by the effective ISF, called sens in your SMB
217 tab, or ISF“orig” in this table. The amplification of $39 \text{ profile_ISF} / 20.9 \text{ effective_ISF} = 1.86$
218 happens to be the dominating bgAccel_ISF amplification factor

219

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

225

226 Depending on your selected “safety” settings, you might occasionally bump into
227 restrictions. **Tuning up factors that make the system bounce into restrictions is a**
228 **completely useless, and potentially even dangerous, exercise!**
229
230 Therefore, for your initial set-up of parameters ([section 2](#) and [section 4](#) of the FCL e-book)
231 it is advisable to **not** do this on the smartphone, but on the PC, where you can inspect the
232 complete info on each loop decision.
233 If for some reason you cannot do this on the PC, you must frequently take screenshots (in
234 very many decisive 5 minute segments), and analyze the data in there.
235
236 In your initial set-up of your FCL, you probably must “loop” a couple of times back into
237 [section 2](#) to adjust the safety settings made for max. SMB sizes.
238
239

11.3 Options available in Trio or iAPS

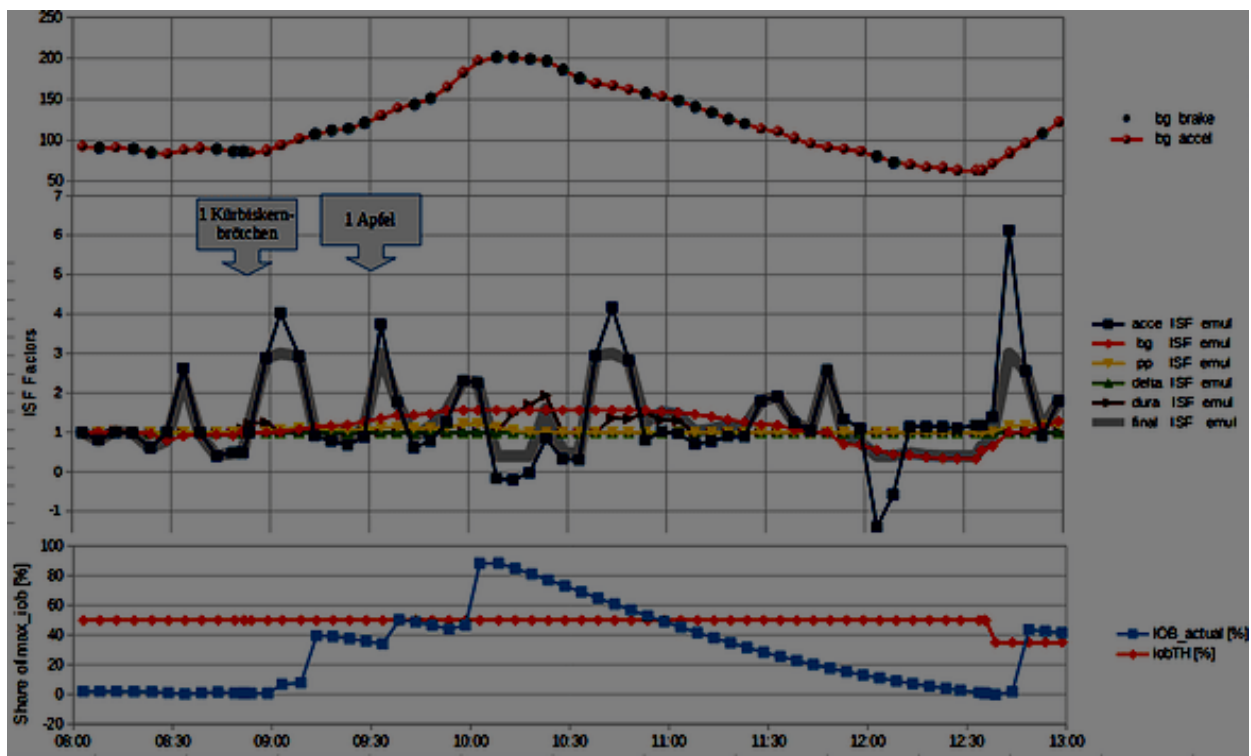
The emulator will not work in the iOS world.

However, iAPS / Open iAPS offer in their autoISF variants also access to a tabular as well as graphical representation of **autoISF contributions to resulting SMB sizes**

Note that on the iPhone, so far, the what-if emulation and speech synthesis (see next [section 11.4](#) for AAPS) are currently not possible. Also not the PC analysis as in section 10.

(insert example of such table and graph)

... is this an example from i-APS ?



How to get table and graphics (double click on statistics button ?)

Screenshots and description must be provided by iAPS / Trio user please

266

267 **11.4 Real-time checking a „what-if“ question using speech synthesis**

268

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

271

272 **11.4.1 Put a (yourChange).vdf into the phone's AAPS logfiles folder**

273

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

276

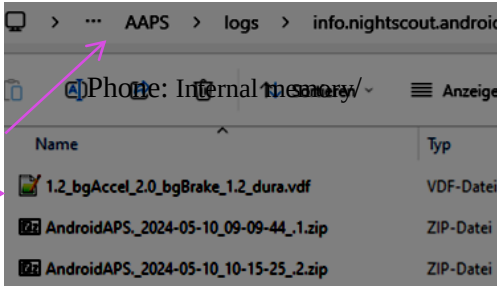
277 Produce or retrieve **(yourChange).vdf**. on PC from one of your studies files, see end of [section](#)
278 [10.2.1](#) :

- 279
- C:\....\Documents\ Looping\ Logfiles_Emulator\ Emulator_Studies\ Study_n

280

281 and put a copy **on the phone** into the folder :

- 282
- Internal memory/AAPS/logs/info.nightscout.androidaps



283

284

285 **11.4.2 Inspect emulated results**

286

287 Now, whenever you run QPython 3L emulation (section 11.2.2) you additionally get the emul.
288 results filled in the resulting table, showing in which time points your change would lead to adapted
289 insulinRequ, and SMB size data.

290

291

No. 1														
UTC		eff.		acce	bg	pp	dura	-----ISFs-----			insulin Req		---SMB--	
time	bg	I0B	iobTH	ISF	ISF	ISF	ISF	orig	prof	emul	orig	emul	orig	emu
20:41:16Z	112	1.93	6.0	1	1.0	1	1	40.9	41	40.9	0	0	0	
20:46:16Z	111	1.77	6.0	1.14	1.0	1	1.03	36	41	36.0	0	0	0	

etc

292

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	

293 In the marked 21;21 Z line, dura_ISF was the dominant factor. The (yourChange)vdf would apply a
294 factor of 1.2 and lead to further strengthening the ISF: profile 43 -> orig.(noChange) 41.9 -> emul
295 (yourChange) 39.5

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

299

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

304 below are "orig."ISFs:

305

	UTC		eff.	acce	bg	pp	dura	-----ISFs-----			insulin Req		---SMB--		
-	time	bg	I0B	iobTH	ISF	ISF	ISF	ISF	orig	prof	emul	orig	emul	orig	emu
1															

306

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	
0															

307

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

309

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

312 @ 21:21 Z / Script Debug -----

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

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

315 -----

316 start autoISF 3.0.1

317 -----

318 Loop allows APS power level; SMB enabled due to enableSMB_always

319 acce_ISF adaptation is 0.95

320 bg_ISF adaptation is 1

321 pp_ISF adaptation is 1

322 dura_ISF adaptation is 1.08 because ISF 43 did not do it for 30 m

323 strongest autoISF factor 1.08 weakened to 1.03 as bg decelerates already

324 final ISF factor is 1.03

325 -----

326 end autoISF

327 -----

328 profile.sens: 43 sens: 41.9

Note: 43 / 1.026 = 41.9

329

330

331

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

333

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

337 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	41,1	0	0,16
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

339

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

344

345 Note that the orig. values for the four autoISF components (acce, bg, pp, dura "ISF orig")
346 are **not** contained in this csv table (only, in Q30, the final amplification factor for final_ISF in
347 the noChange scenario).

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

- 349
- from the txt result file (as shown above); or
 - you also see them on the phone:
- 350

below are "orig." ISFs:				
acce	bg	pp	dura	-----
ISF	ISF	ISF	ISF	orig
0,95	1,0	1	1,08	41,9

351

352 ...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

353

354

355 Apology: The above example was not well chosen to see relevant effects. The author is struggling to
356 put this chapter first time together, quickly for the V.3.0.1 launch, and just picked from his phone what

was available at the moment, I might patch it over with a better example in a later update, or I (or maybe you?) provide an adjunct case study.

11.4.3 Real time speech synthesized treatment advice

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

If for instance a suggested extra, or bigger, SMB makes sense, **you can add this portion manually*and observe**, for this meal, whether this bolus was OK and you should switch in direction of the different setting you were investigating (which would automatically give you that extra in the future).

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

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

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

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

Activating and silencing emulator suggestions

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

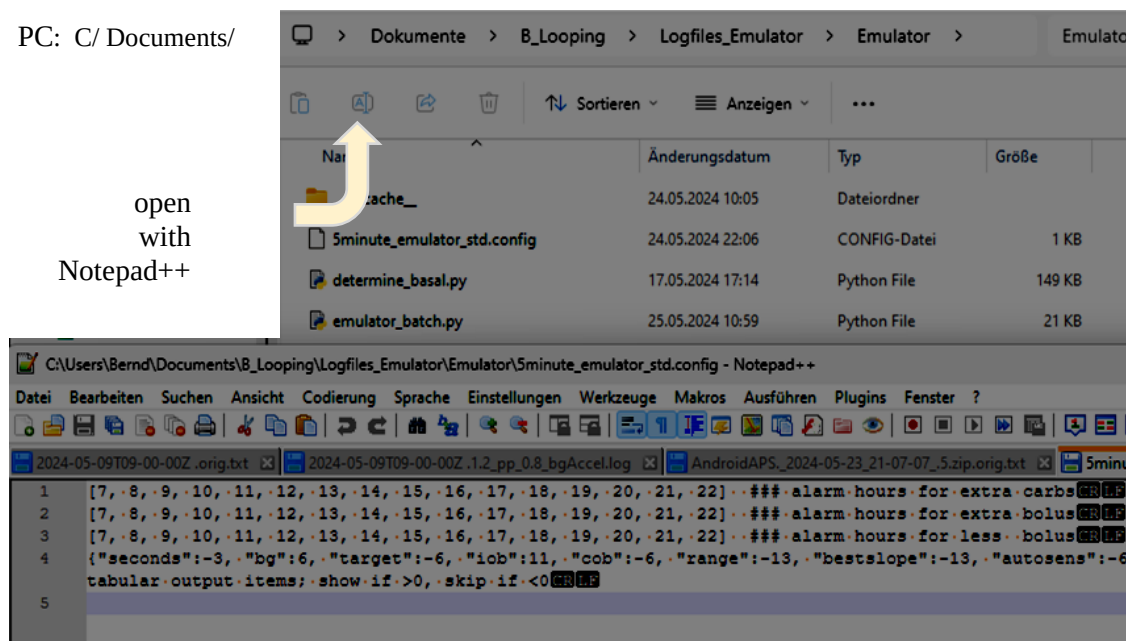
(1) Change **principal settings** what shall be announced (e.g. only if bigger SMB size is suggested, or also warnings about carbs eventually needed?), and in **which hours of day**, to make any announcements via speech synthesis .

These are set in the (1 or) 5minute_emulator_std.config file:

Go on **PC** into the config. file (see illustration). Open it with Notepad++ and edit the hours there for when you would want (no) announcements regarding: extra carb 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) Note that when still, and exclusively, running the noChange.vdf, you will not get any speech outputs (because you are NOT investigating a what-if question, in that case). But you can look up the table as shown in [section 11.2.2](#)

(5) ? - Note that this area has not been researched much, and good answers will be lifestyle and phone specific.

Please share experiences in Discord/FullClosed Looping

419 11.5 Exploring phone data further on PC

420

421

422 Add later:

423

- 424 • Evaluation of AAPS logfile autoISF FCL after accepting some emulator “what-if”
425 suggestions (small manual boli to complement triggered SMBs)

426

427

428