11. Emulator on your AAPS smartphone   V 2.8

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
11.3 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 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.
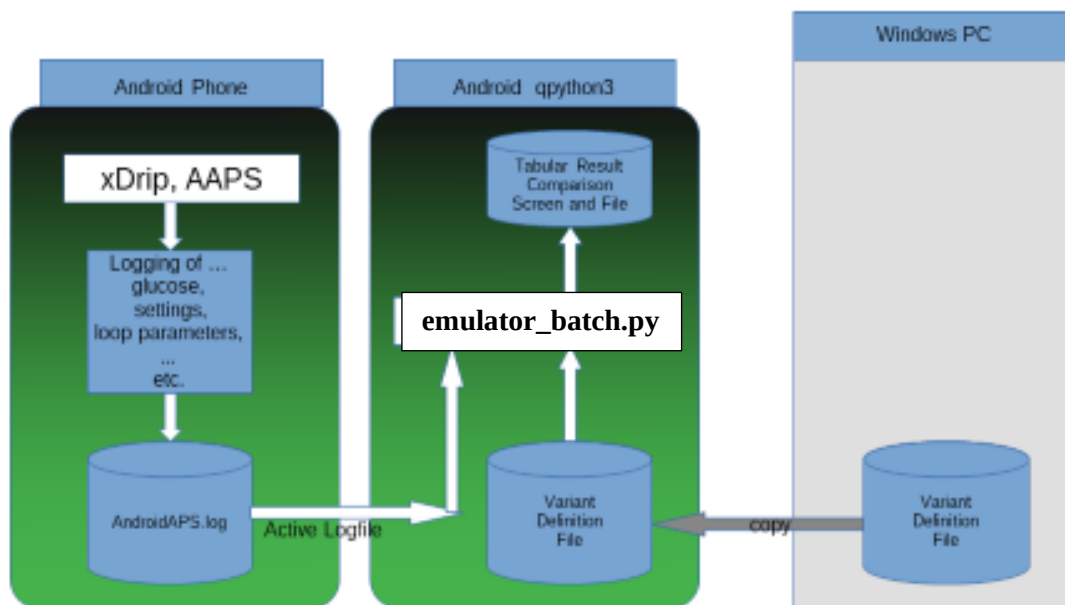
Sketch of Running the Emulator on AAPS Phone

Github/ga-zelle /
APS what-if

40

41

11.1 Installing the emulator on your Android smartphone

43

44   Note that iOS based autoISF variants cannot use the Emulator on an i-Phone. However, some

45   tabular outputs of ISF-factors are available (see to be provided by iAPS user)

46

11.1.1 Installing QPython 3L

48

49 On your smartphone, go to Google Playstore and download the QPython 3L app. Put the app icon

50 next to your other looping related app icons on the main screen of your smartphone.

51

52   With many QPython 3L versions, phone and Android OS versions etc

53   around, you might run into problems and may need to consult detailed

54   installation instructions from Github/ga-zelle / APS what-if, or seek advice via

55   https://discord.gg/n3tD5eXExC

56

57   By long pressing on the QPython 3L app icon, go into "app info" and make

58   the settings like for all your other looping related apps, so they do not get

59   killed all the time by energy savings routines.

60

61

62

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

64

65 Connect your phone to your PC for USB data transfer

66

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

68

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   **Copy** all Python related files except emulator:GUI,py from your PC over the internal memory / QPython

71   / **Scripts3** of your phone:

Phone: internal memory >   >   qpython   >   scripts3

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	

not needed on phone, erase

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

72

73

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   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), and put a

84   **copy** into the phone into the logfiles (not the QPython!) folder (picture, right):

85

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

87

88   This config file contains the programming as to which hours of day there can be speech messages.

89   Default 07-23 h ("your" time, not "Z"scale). How to change, see in <u>section 11.4.3</u>

90

91

92
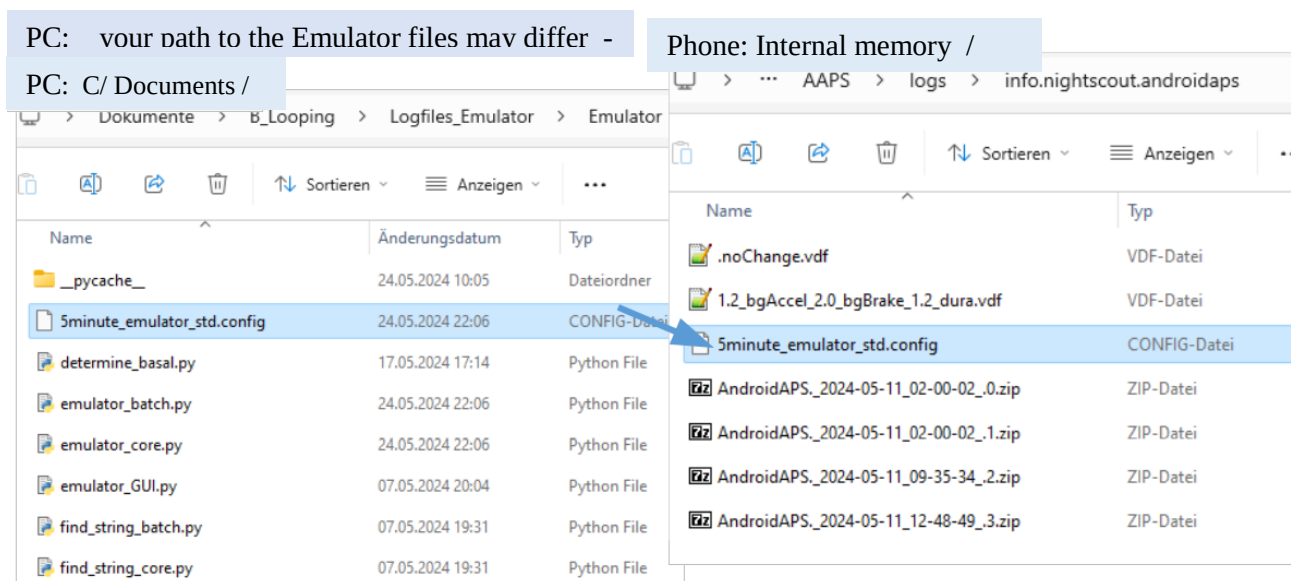
93   11.1.4   Put noChange.vdf into the phone's AAPS logfiles folder

94

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

96   •   C:\....\Documents\ Looping\ Logfiles_Emulator\ Emulator_Studies

97

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

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

100

101   Later, in <u>section 11.4.1</u>, you will add also (yourChange).vdf files into the same folder. One is

102   already included, line under noChange.vdf, in picture above, right.

103

104

105

106

11.1.5 Customization of output table

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

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.

137   11.2.2 Generating a results table (for last 75 minutes' loop decisions)

138

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

140   15 minutes is really too short to analyze what is going on, and Libre3 users probably should

141   mostly use the "what-if" part, see section 11.4. That part is principally not impacted (except,

142   cutting one 5 minute change into 5 very tiny changes, often will make it hard to see and

143   interpret "what-if" effects).

144

145   Display setting on your phone should be set for automatic switch between landscape and

146   upright viewing (depending how you hold your phone).

147   Reducing selected font size will *not* help to get more table info on your phone screen, or to

148   avoid broken/double lines. Go to 11.1.4 if, in the following, you are not happy with lay-out.

149

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

151

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

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

154   upcoming black screen

155

156   All black screens have a keypad at the bottom:

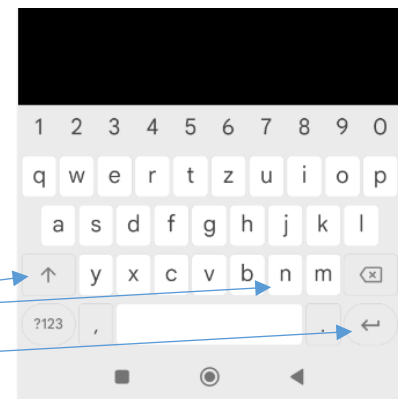157

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

159 3).Then proceed in the same screen

160   to make your capital-"N" entry   +

161   and finally " <- " for Next (see illustration)

162

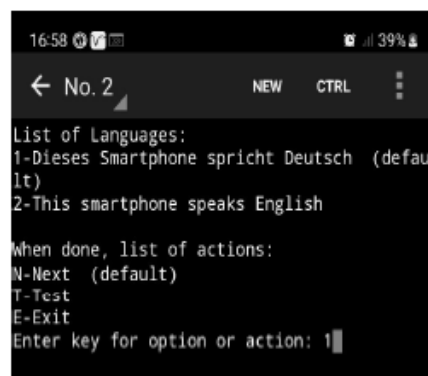163 In the following the instructions from the developer how to navigate through the screens:

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.

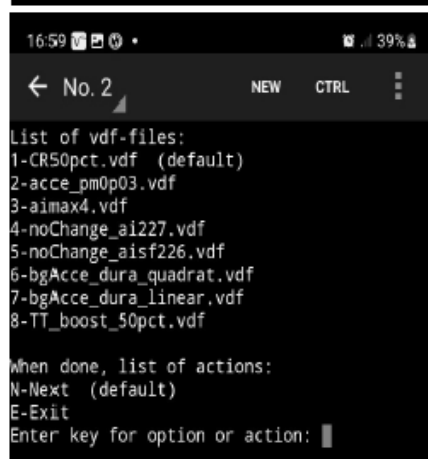Select „Test" to listen to a sample speech synthesis.

Select „Next" to proceed.


```
16:58          39%
← No. 2           NEW    CTRL

List of Languages:
1-Dieses Smartphone spricht Deutsch  (defau
lt)
2-This smartphone speaks English

When done, list of actions:
N-Next  (default)
T-Test
E-Exit
Enter key for option or action: 1
```
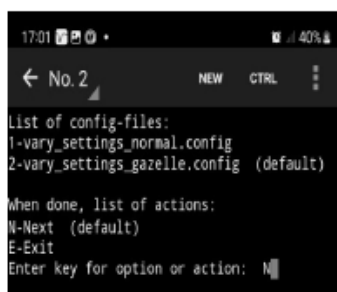
4-5). Repeat steps 2) and 3) with the next screen:

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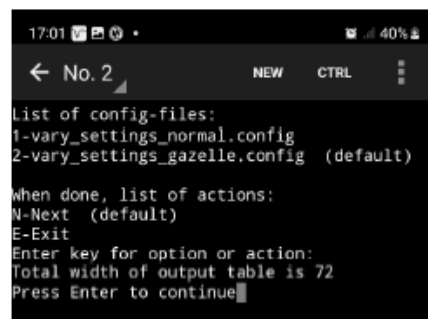


```
16:59          39%
← No. 2           NEW    CTRL

List of vdf-files:
1-CR50pct.vdf  (default)
2-acce_pm0p03.vdf
3-aimax4.vdf
4-noChange_ai227.vdf
5-noChange_aisf226.vdf
6-bgAcce_dura_quadrat.vdf
7-bgAcce_dura_linear.vdf
8-TT_boost_50pct.vdf

When done, list of actions:
N-Next  (default)
E-Exit
Enter key for option or action:
```

164

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.


```
17:01          40%
← No. 2           NEW    CTRL

List of config-files:
1-vary_settings_normal.config
2-vary_settings_gazelle.config  (default)

When done, list of actions:
N-Next  (default)
E-Exit
Enter key for option or action: N
```



```
17:01          40%
← No. 2           NEW    CTRL

List of config-files:
1-vary_settings_normal.config
2-vary_settings_gazelle.config  (default)

When done, list of actions:
N-Next  (default)
E-Exit
Enter key for option or action:
Total width of output table is 72
Press Enter to continue
```

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.

165

166

167

168

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

170 minutes.

171   In case you forgot to do it in step 4), turn the phone now 90 degrees for landscape format;

172   in this case, give it 5 minutes for the format to straighten out (after a new value has arrived)

173   In case you want other, or less, columns, see <u>section 11.1.4</u>.

174

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

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

177

178 In sequence of the time(Z!), the table consists default of the columns with info on (1) bg (2)

179 IOB and eff.iobTH (3-6) the adaptation factors on ISF suggested by the 4 autoISF

180 categories (7) resulting ISF that was used, profile_ISF, and emul ("what-if" ISF) (8)

181 insulinRequired calculated by your running loop, and "what-if" result (9) same for resulting

182 SMB

183

184 The columns marked "**orig**".in the table of results shows the ISF actually used to

185 determine insulinReq and SMB size in the actual run.

186

187 In the columns marked "**emu**l" you find the calculated emulation results that is calculated

188 every 5 minutes

189 •   If a (yourChange).vdf was clicked (in step 2), see also bottom right of the table), the emul

190 columns show the result, how the investigated changes *would have* changed SMB sizes (strictly

191 always looking at the one decision, in that line of the table).

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

193 columns.

194

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

196

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

198 •   So when using (yourChange).vdf, you see in your phone's table exclusively the ISFs that

199 *would result-if* (yourChange) were made.

200 We get to "what-if" projects later, in section 11.4.

201 •   Only in the noChange scenario, the values there would be the *orig ones, corresponding to*

202 *what could be seen also in the SMB tab* at the times

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

204

205 orig.ISF is called **sens** in the SMB tab, a couple of lines below "end autoISF". It is the ISF

206 that replaces, for that one decision you are looking at, the profile ISF (called profile.sens in

207 the SMB tab),

208   We get back to this topic at end of section 11.4.2 "Understanding how the ISF is emulated by

209   (yourChange) and how SMB or TBR would differ"

210   "

211

212

11.2.3 Analyzing the results

UTC time	bg	IOB	iobTH	eff. ISF	acce ISF	bg ISF	pp ISF	dura ISF	ISFs orig	ISFs prof	ISFs emul	insulin Req orig	insulin Req emul	SMB orig	SMB emu
13:53:42Z	116	2.19	6.0	1.54	1.01	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	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	28.5	39	28.5	0	0	0	0	
14:08:43Z	113	1.81	6.0	1.18	1.0	1	1.08	33	39	33.0	0	0	0	0	
14:13:41Z	115	1.64	6.0	1.11	1.01	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	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	27	39	27.0	0.26	0.26	0.1	0.1	

215 In above example (table), all given SMBs were driven by bgAccel_ISF, when glucose rose.

216 The biggest SMB in the time we are looking at (actual local time = Z + 2 hours, so we are

217 looking at late small rises, like 3 hours after lunch) was 1.7 U = 0.74 SMB delivery rate *

218 2.3 U insulinRequ.

219 The insulinRequ. Is proportionally amplified by the effective ISF, called sens in your SMB

220 tab, or ISF"orig" in this table. The amplification of 39 profile_ISF / 20.9 effective_ISF = 1.86

221 happens to be the dominating bgAccel_ISF amplification factor

222

223 Note **not** always just the biggest ISF factor "wins". Consult flowchart in <u>LINK</u>, and

224 occasionally read the real-time explanations in your SMB tab as to which other factors are

225 contributing to the amplification resulting from profile_ISF to effective_ISF ("orig", or sens).

226 For instance, the Activity Monitor, or a set %temp. profile, or TT, could contribute, or also

227 the question whether glucose already decreases.

228

229 Depending on your selected "safety" settings, you might occasionally bump into

230 restrictions. **Tuning up factors that make the system bounce into restrictions is a**

231 **completely useless, and potentially even dangerous, exercise!**

232

233 Therefore, **for your initial set-up** of parameters (<u>section 2</u> and <u>section 4</u> of the FCL e-

234 book) it is **advisable to** not **do this analysis** on the smartphone, but **on the PC**, where

235 you can inspect the complete info on each loop decision (see <u>section 10</u>)..

236   If for some reason you cannot do this on the PC, you must frequently take screenshots (in

237   very many decisive 5 minute segments), and analyze **more complete data**, than the table

238   on our smartphone could offer, **in the SMB tab**)

239

240 In your initial set-up of your FCL, you probably must "loop" a couple of times back into
241 to adjust the safety settings made for max. SMB sizes.
242
243

244

11.3 Options available in Trio or iAPS

246

247 The emulator will not work in the iOS world.

248

249 However, iAPS / Open iAPS offer in their autoISF variants also access to a tabular as well as

250 graphical representation of **autoISF contributions to resulting SMB sizes**

251

252   Note that on the iPhone, so far, the what-if emulation and speech synthesis (see next <u>section 11.4</u> for

253   AAPS) are currently not possible. Also not the PC analysis as in section 10.

254

255 (insert example of such table and graph)

256

257

258 … is this an example from i-APS ?

259

260

261
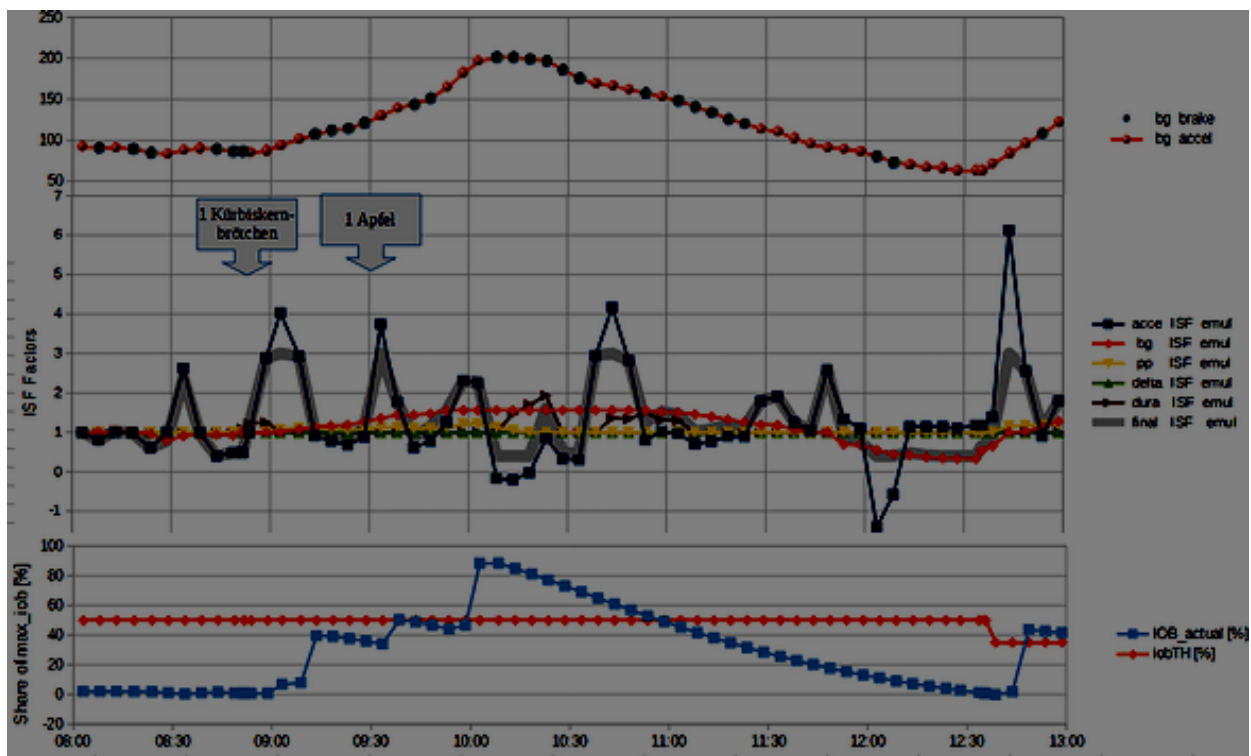
262 How to get table and graphics   (double click on statistics button ?)

263

264 Screenshiots and description must be provided by iAPS / Trio user please

265

266

267

268

269

270

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

272

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

considered change.

275

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

277

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.

280

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).
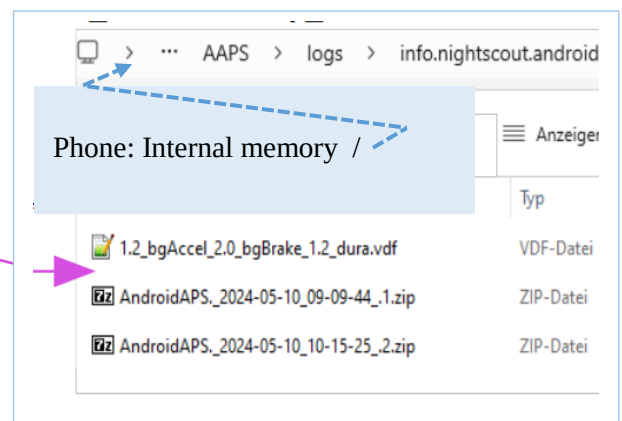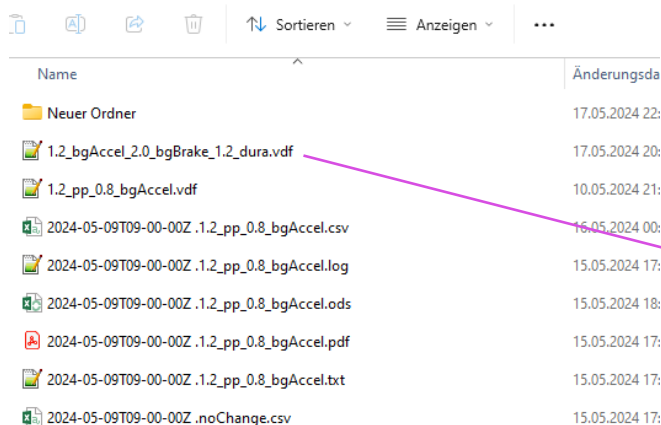
284

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

286

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

288

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

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

292

293

294   Switch between python scripts running at the same time

295

In case you have more than one (yourChange).vdf to investigate, you look at the data of

your currently running loop (last 15*5 minutes) by just switching between the related vdf

files used for emulation.

299 Details see section 3.8, 9) – or try to go direct via https://github.com/ga-zelle/APS-what-

300 if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-

301 phone.pdf and there p.5, under above sub-headline ".Stop the emulator, or switch…"

302

303 Stop the emulator

304 (see in paper as above)

305

306 11.4.2 Inspect emulated results

307

308 Now, whenever you run QPython 3L emulation (following the steps as described in section 11.2.2)

309 you additionally get the emul. results filled in in the resulting table, showing in which time points

310 your change would lead to adapted insulinRequ, and SMB size data.

311

etc.

313 etc

314 In the marked 21;21 Z line, dura_ISF was the dominant factor. The (yourChange)vdf would apply a

315 factor of 1.2 and lead to further strengthening the ISF: profile 43 -> orig.(noChange) 41.9 -> emul

316 (yourChange) 39.5

317 In this case, late after a meal, and bg=109 mg/dl, the loop saw in the orig. (noChange) case 0.1 U

318 insulin too much; and as the (yourChange) emul case asks for typically more insulin (all weigts in

319 the (yourChange).vdf are > 1), now only 0.08 U are seen as too much (a 20% difference).

320

321 The table on your phone is too reduced to show each emulated ISF component. If you need to see

322 more details on how (yourChange).vdf would make a change in a point of time that you like to

323 analyze deeper: Note down interesting Z times, and later look it up in more detail in the Emulator

324 on the PC:

325

below are "orig."ISFs:

UTC time	bg	IOB	iobTH	eff. ISF	acce ISF	bg ISF	pp ISF	dura ISF	ISFs orig	prof	emul	insulin Req orig	emul	SMB 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		

326

327

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

329

330 To fully understand how acce, bg, pp and dura_ISF determined the orig.ISF, we do an **emulator**

331 run **on the PC,** and look up the **.txt** result:

332 @ 21:21 Z / Script Debug --------------------

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

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

335 ----------------------------------

336 start autoISF 3.0.1

337 ----------------------------------

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

339 acce_ISF adaptation is 0.95

340 bg_ISF adaptation is 1

341 pp_ISF adaptation is 1

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

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

344 final ISF factor is 1.03

345 ----------------------------------

346 end autoISF

347 ----------------------------------

348 profile.sens: 43 sens: 41.9   Note: 43 / 1.026 = 41.9

349

350

351

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

353

354 While the result for ISF_emul (and for the SMB size) is given in the table on your phone, the

355 details behind it, and also for finer effects in %TBR, can only be inspected from the .csv table from

356 the (yourChange),vdf **emulator run** on the **PC**:

357

358

359

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

		bg	bg		final	dura		acce	bg	pp	delta	dura	final				TBR	TBR
UTC	AAPS	accel	brake		ISF	min-	dura	ISF	ISF	ISF	ISF	ISF	ISF	ISF	ISF	ISF	TBR	TBR
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*

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

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

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

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

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

…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

Apology: The above example was not well chosen to see relevant effects. The author is struggling to 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.

385 Principal limitation of any of your emulations

386

387 Note that always the first biggest change regarding insulinRequ and SMB size in emul vs. orig. is

388 the most relevant. This is because:

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

395

396 This is one of the reasons why setting up your FCL ([section 4](#)) should be a iterative process,

397 seeking solutions (with mainly bgAccel_ISF) for the first rise (and a range of different meals) first,

398 AND NOT concurrently already "tuning" the dura_ISF.

> Both, bg peak height, and pattern of insulin activity from the present iob (that is only sluggishly
> further adjustable driven by dura_ISF and evtl. by bg_ISF) depend on how the first bg rise stage was
> managed by autoISF (your bgAccel_ISF_weight setting, and others).

402

403 Another conclusion you might draw, is to limit use of the emulator *on the PC* largely to analyzing

404 the (in FCL extremely important) first rise, to seek bgAccel_ and pp_ISF_weights.

405

406 Then to test these, and many other like ideas for changed settings "in real life" emulated *on the*

407 *smartphone*: A synthetic voice will announce to you how (if) each actual decision would differ. See

408 next section!

409

410

411 11.4.3 Real time speech synthesized treatment suggestions

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

414

415 At time points when the (yourChange) setting would result in smaller or greater difference in SMB

416 insulin delivery (compared to the real "noChange" run), you can get a real-time notification **via**

417 **speech synthesis**, and you can assess the situation in real-time yourself.

418 If for instance a suggested extra, or bigger, SMB makes sense, **you can add this portion**

419 **manually*** ….**and observe**, for this meal, whether this bolus was OK and you should switch in

420 direction of the different setting you were investigating (which would automatically give you that

421 extra in the future).

422 Likewise, you might choose **not** to intervene, but regret it an hour or so later, seeing the further

423 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++
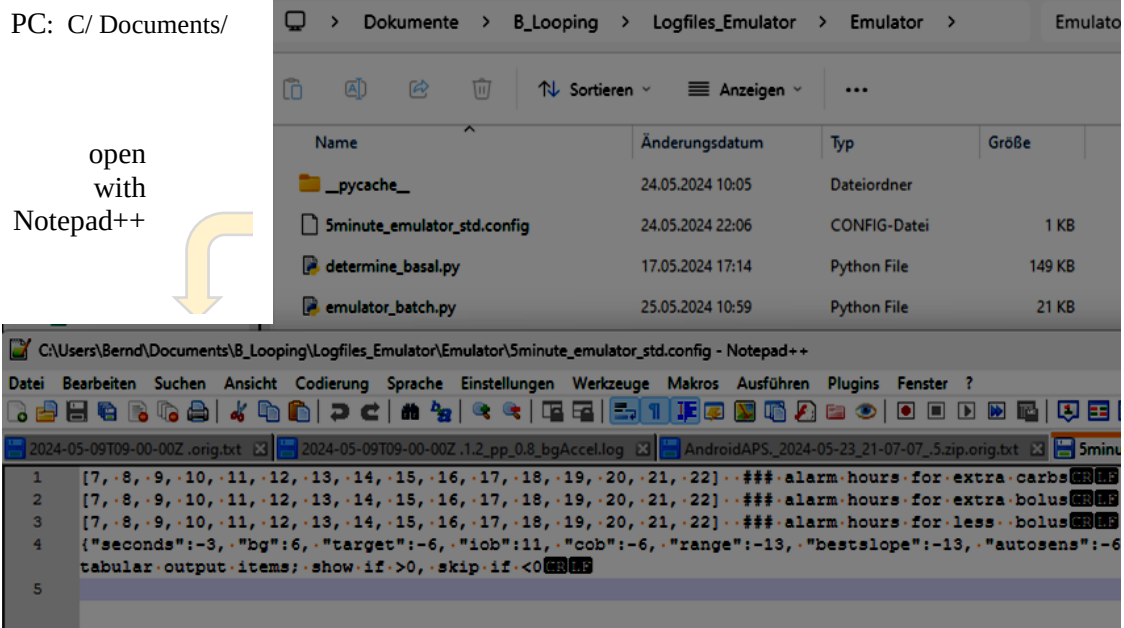
Save the changes, and copy the file also into your **phone** at Internal memory/AAPS/logs

/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)

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

452

453 (3) "**Kill" (and later resume) the "what-if" emulation**. This could be done by de-

454 selecting the (yourChange).vdf in step   . However, this stops (or interrupts, until you

455 re-start) the entire emulation and you will have no tabular data later for the silenced time.

456

457 (4) **De-activate Qpython 3L app** (temporarily?): Press on Qpython 3L app icon, force

458 close or remove necessary permission – re-activate (or need to re-start emulation then ??)

459 when you want to hear again ..   However, this stops (or interrupts, until you re-

460 start) the entire emulation and you will have no tabular data later for the silenced time

461

462 (5)   Probably better than the 2 afore mentioned options would be to **run (for the**

463 **intended silencing period) exclusively the noChange.vdf,**

464 Then you will not get any speech outputs (because you are NOT investigating a what-if

465 question, in that case). But you get all data (the un-interrupted noChange actual run) and

466 later on the PC still can investigate any "what-if" scenarios

467   How to change the .vdf reference during a run, see section 3.8, 9) – or go direct to the

468   instructions in the Github repo at: https://github.com/ga-zelle/APS-what-

469   if/blob/A3.2.0.4_ai3.0.1/Documentation%20in%20English/How-to-run-the-emulator-on-the-

470   phone.pdf and there p.5, under above sub-headline ".Stop the emulator, or switch…"

471

472 Note that this area (silencing) has not been researched much, and good answers are likely

473 lifestyle and phone specific.

474

475

476 Please share your experiences with the emulator in Discord / Full-Closed-Looping / HOW

477 TO /_emulate-aaps, at: https://discord.gg/n3tD5eXExC

478

479

480