# 5. Modulation of autoISF aggressiveness. V 4.0

**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/readme.md




Available *(related)* case studies:

Case study 5.2: Sweet snack.
Case study 5.3: Compression low

*Skip what is in **green writing***:
= Drafted fragments or not implemented ideas. Read only if you like to participate in discussion – Suggestions will probably be radically reduced because FCL *can and* should be done *without* most discussed extra features

Once the initial tuning according to section 4. is done, you are ready to use autoISF for your fully automated *meal management*.

## Challenges beyond managing main meals

There are up to four *other major challenges* you might have to manage:

1. Secure the technical pre-requisites for your FCL remain a given (see section 1):
    - Prevent (frequent cannula changes), recognize and manage (partial) occlusions
    - Secure "meaningfulness" of 100% all your CGM values

- Secure 24/7 BT connectivity of all your components

2. Deal with times when insulin given, after meal detection, must be restricted (e.g.:
   - a snack should not be “treated” as a meal
   - meal before exercise should not be treated “all the way” towards high iob.
3. Deal with times when the loop should be set "less aggressive" as a precaution, e.g.
   - during nights
   - in an exercise context
4. Deal with times when the loop should be set "more aggressive" e.g.
   - at sickness
   - at tempting hotel breakfast buffet

How big the remaining challenge really is, depends very much on your individual lifestyle.

## Hands-off FCL – an elusive goal?

To run the FCL “hands-off” around the clock, you now must analyze also **your** data in the times *outside* the meal blocks, and seek solutions to problems (if any).

With a technically well-functioning system, moderate meals, moderate or no exercise, moderate %TIR expectations and a bit of mindfulness it should be possible to go into **Full Closed Loop 24/7**, after working through, and observing, sections 1-4. See case study 4.3.

- In section 5.1 we explore avenues towards ***fully automated*** management that in daily life will require no user intervention at all.

Often it will be **your** choice whether you want to bother with researching in **your** data for, and defining, dedicated (“personalized”) Automations (see also section 5.1.6.). Or whether you accept instances, where you do manual steps:

- In section 5.2 (and later, related to exercise, in section 6.) we will look at solutions that involve an easy ***user interaction*** *like a data entry, or a button push.*

## Avenues for temporary modulation of the FCL aggressiveness

1. temporary shut-off SMBs (odd-numbered target)
2. temporary change bgAccel_ISF-weight

3. temporary change iobTH_percent
4. temporary change the set %profile
5. temporary set different bg target (especially in connection with exercise mode)t

All of these are also easy accessible via Automations in AAPS

This means: If, in **your** data, you find sets of conditions where any of the actions 1.-5. could help, you can make this aspect integrated part of **your** hands-off FCL!

Note that in Automations you can go beyond above mentioned 5 avenues, and also include e.g. setting a different bgAccel_ISF_weight; or you can come up with a nearly surgical definition as to from which iob on, and other criteria, some temporary modulation of aggressiveness shall happen.

## 5.1 **Fully automatic** modulation of FCL aggressiveness

The following subchapters describe set-ups you may want to use for allowing **completely hands-off FCL** in as many daily situations as possible (and potentially all the time, as in case study 4.3)

### 5.1.1 All autoISF ISF adaptations switched off outside of meal-time windows

If, aside from having to bolus for meals, your hybrid closed loop was running pretty well *without* other interventions from your side, you could continue to run in that mode, and just focus your new autoISF FCL on management of meals.

In your initial transitioning phase, this approach makes a lot of sense, and even by focusing autoISF on just a sub-set of meals, like only dinners.

Also in the long run this avenue is taken by many FCL users for the night times, “hanging on” to their well performing hybrid closed loop with standard oref(1) SMB+UAM

For this, you define Automations

- that set meal time windows in which “Enable ISF adaptation by glucose behavior” (autoISF) is turned on in AAPS preferences/OpenAPS SMB
- *or:* that turn *all* autoISF’s ISF modulations (*or* just *bgAccel_ISF* ) off in time windows in which surely no meal occurs.

For instance, you can go *for all nights* back into your Hybrid Closed Loop, as you had before.

Your temp. “autoISF shut-down” (exiting autoISF FCL = shutting off “Enable ISF adaptation by glucose behavior“ in AAPS Preferences / SMB) is meant to prevent problems from the loop *over-reacting* to bumps in the glucose curve in times of day (night), when standard oref(1) performance is sufficient.

A very good alternative to defining “meal-time windows”, or to resorting to night-time Hybrid Closed Loop, is letting the autoISF FCL run 24/7, and “taming” the FCL via a night time SMB shut-off (see next section 5.1.2).

> *Off topic*: ***Other* “early dev” AAPS variants** (see section 13.3), do require working with meal-time windows. The window is either set by time of day in the settings, or it always must be „set“ by the user. Trigger to set a meal time window could be a pre-bolus given by the user, a carb entry made, an EatingSoonTT set, or a meal-announcement-button pushed (none of these things are required in autoISF FCL).
> Outside of these time windows, these loops then run with less aggressive SMBs, just like oref(1) SMB+UAM in AAPS Master, with or without (modified versions of) dynamicISF. So, outside of the meal time windows, you would be in “your normal” hybrid closed loop.
> The term **Meal Announcement** (MA) is often used to label this closed looping mode.
> It is not really FCL, but an advance over traditional HCL.

### 5.1.2 Odd-numbered profile targets, to block SMBs

An alternative route of preventing the FCL loop from over-reacting to bumps in the glucose curve would be to make use of the option to temporarily shut down SMBs

> Ensure the even/odd logic in the settings is toggled on in Preferences> openAPS SMB> autoISF settings> smb delivery settings>: "Enable alternative activation of SMB depending on bg target”: ON.

In time blocks with an odd-numbered profile target you can prevent any SMBs being given by your loop. The (unchanged) aggressive settings then can only translate within the limits set by %TBR possible.

This will very much slow down any more insulin being given, and is an excellent solution for night times, especially if you occasionally experience compression lows.

Alternatively, you could use the new included options for Automation Conditions and temporarily tune your bgAccel_ISF_weight much lower (see section 5.1.4).

The same situation can be achieved if you generally operate with a mild bgAccel_ISF, and make your autoISF only really aggressive for meal-time slots (if you have similar enough times every day, or also can “employ” geo-fencing in your Automation (or middleware, in iAPS) conditions).

In these cases you would not need to have night profiles that disable SMBs: - Which is the better way would depend on a lot of personal factors relating to how high-carb the diet is, regularity of meals, snacking habit, CGM quality and incidence of compression lows, and probably more. - I would try both routes, or, as this is fairly complex to tune, just one, and stick with what is working good enough.

Yet another alternative was already presented (section 5.1.1) = to go into hybrid closed loop for the night.

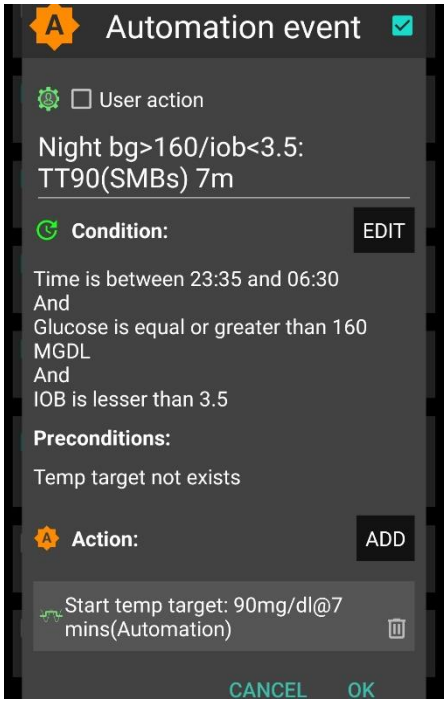
That is possible, with SMBs available (without them getting boosted via autoISF), and, for a long time, was the author’s favored solution for the nights.

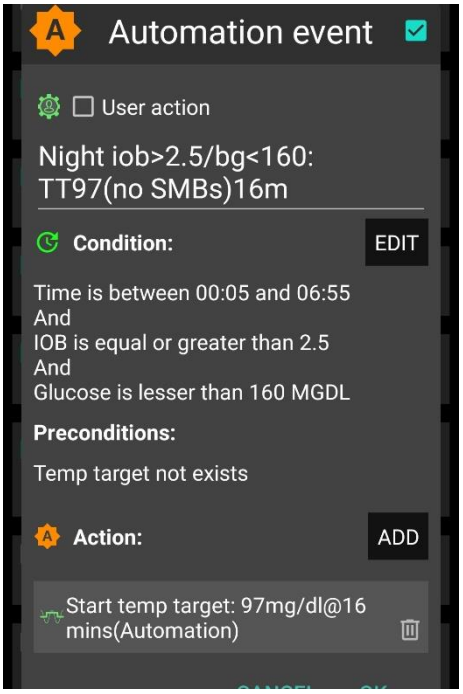
This solution is evidently similar to the prior discussed one, of having a mildly tuned autoISF 24/7, boosted to high aggressiveness only in meal-time slots.

### Adjunct Automations to allow a few SMBs, in nights with odd profile target SMB shut-out

My current **favorite** builds on the **method** of this section (5.1.2, odd profile target provides SMB shut out), but then allowing some, automatically triggered SMBs, when needed:

In case you occasionally do have nights that would benefit from a couple of SMBs (to treat temp. highs from a late fatty pizza, raclette and such): Define suitable Automations like the two „night“ ones in this *example:*




Caution: Never underestimate the „trickiness“ of getting your Automations „right“.

With your thought-out Automations in place, night data need to be analyzed to see

- whether the bg and iob limits, as defined in the given example, work sensibly four *your* data pattern
- whether the TT duration is chosen appropriately
- how swapping the sequence in which the automations appear in the Automation list would lead to different SMB impacts.

### 5.1.3 Odd-numbered temp. targets (TT) set via Automation, to block SMBs

A widely used Action that strongly modifies how fast your FCL can add more iob is setting an **odd**-numbered **temp. glucose target** which makes the loop operate without giving any SMBs (%TBR modulation only).

Ensure the even/odd logic in the settings is toggled on in Preferences> openAPS SMB> autoISF settings> smb delivery settings>: "Enable alternative activation of SMB depending on bg target": ON.

Then, from patterns you find in *your* data, at times where you want your loop act differently, you need to carve out Conditions that describe the respective situations (and either *for how long* it typically lasts, or at which *other* Conditions you want your loop get back to default FCL operation).

An odd TT is often set for an *anti-hypo* snack *or sports* snack. In both instances, you do not want SMBs to quickly counter-act.

In case of *sweet "fun"* snacks, this is entirely different -> section, 5.2.1 or for regular snacks (*e.g.at school break*) see next section 5.1.4

### 5.1.4 Automatic differentiation of FCL aggressiveness using Automations (or middleware)

**Personalized Automations** tailor the loop exactly to ***your*** data so **fully automated** handling of situations with **different aggressiveness** of the loop can be made.

Automations are an integrated and very easy-to-use feature in AAPS.
(The i-Phone platforms Trio or iAPS lack this feature. However, so-called **middleware** has been developed as add-in to your code, see: https://github.com/macconnellk/RoboSurfer/tree/main )
.
From, autoISF 3.0 onwards, also the following parameters are provided as Condition and/or as Action for defining YOUR Automations:

- Enable ISF adaptations by glucose behavior => Allows temp. ON/OFF for the key ISF modulation parts of autoISF (and, as a result, will usually decrease loop aggressiveness)

- Trigger/set iobTH percent => Keeps default aggressiveness, but only until a iob threshold (that your Automation modifies) is surpassed (which is when any further SMBs will be blocked blocked)

- Trigger/set bgAccel_ISF_weight => Modifies the aggressiveness of just the acceleration component

To set up suitable Automations, you first must **analyze patterns** you find **in *your* data**, at times (or geo-location, or bg and iob patterns that point to a problem …) **where you want your loop act differently**, to carve out Conditions that describe the respective situations (and either for how long it typically lasts, or at which *other* Conditions you want your loop get back to default FCL operation).

A variant of this mode is to define several windows in which autoISF aggressiveness (bgAccel_ISF_weight) and/or iobTH are automatically set differently

- for **different meal time slots** of your day –

  (*Breakfast at home, school lunches, school intermission snacks, dinners at home* could for example all deserve special settings regarding ISF_weights and iobTH).

  Note: Circadian differences in insulin sensitivity between meal times are included via your ISF profile and should not be a reason for different _weights needed between meals!

- or even for a geo-location etc –

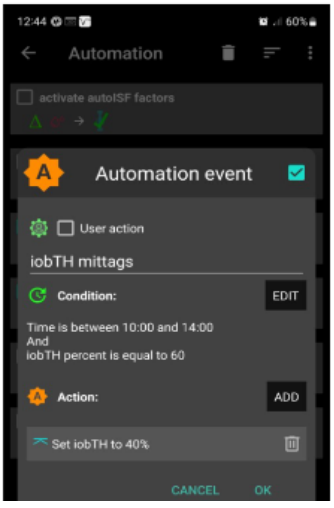
  (*School lunches, or mother-in-law visits, would be examples*).

An example for this was given in section 3 already:

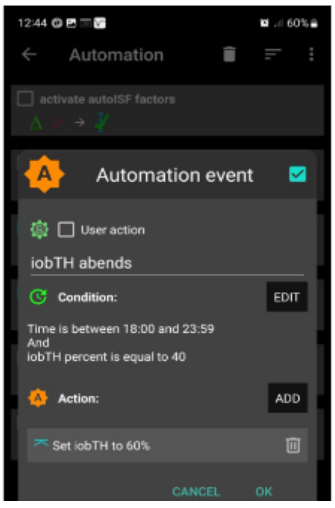
Here is an example set of automations to alternate between two values of iobTH:

I use two different values of *iob_threshold_percent* during a normal day. It is 40% for lunch time and 60% for dinner time. I have these two rules to switch by time of day and only if the current value equals the value from the earlier shift. Any other value is treated as a manual override for special occasions until I manually set it to its regular value. The time windows for switching are long enough to catch an opportunity to be processed and do not need to be actioned half a day each.

lunch




dinner

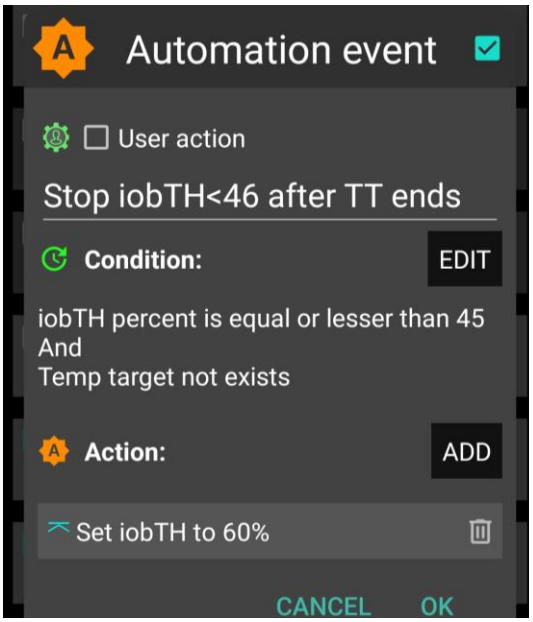


Unless your meals differ vastly in size and in fast carb content **all this may *not* be needed**.

Still, personalized Automations might help ease your initial job of setting the various ISF_weights, and a best-suitable iob_theshold_percent that would work “always”.

> Note: Only the main two parameters (bgAccel_ISF_weight for “initial aggressiveness”, and iobTH_percent for “where SMBs stop”) are available in Automations. So, finding your parameter sets *for each of* the time slots, will not be trivial. => **Spending more effort to set the .._weights so they accommodate *just one, broader* spectrum** (section 4.) **should be the first, and standard, approach.**
>
> An intermediate (maybe only temporary) approach could be to use a profile switch (for low carb meal, or eating half, setting %profile to 60% for instance, and only for the brief, less than an hour, initial meal period). See section 5.2.2 and 5.4.5

**Caution**: Setting a different iobTH% or bgAccel_ISF_weight can probably not be done with a duration attached. Then you **must** define a suitable **additional Automation** that must be active in tandem, to **restore the values you had set in /Preferences for your iobTH% or bgAccel-ISF_weight**. Else, once your Automation set in, it will *forever* shift these important parameter settings!




If for instance you have several Automations that, in combination with a set elevated TT also set a lower iobTH: Don’t be fooled, the duration only applies to the TT. You need an extra Automation for all of them.
I picked out the highest of the altered iobTH values that these Automations can set (45_percent), and then I can automatically restore my default desired 60% via this one Automation (see screenshot - - > )

### 5.1.5 Different FCL aggressiveness set by the Activity Monitor

With the autoISF variant of AAPS you can make use of your smartphone‘s **stepcounter** and use it to fully automatically adjust insulin sensitivity ratio to **activity level in the past minutes to one hour** time frame.

This feature comes with yet another little tuning opportunity, in which you study your body’s response to light exercise (like walking) or to not moving at all (like desk, couch), and select appropriate settings (in AAPS Preferences/OpenAPS SMB/Activity modifies sensitivity/ -> set **two scaling factors**).

This **will automatically adjust insulin delivery** (basal, ISF, and iobTH; see 1st screen of AAPS SMB tab (example in section 5.4.5)), to suit activity state of the past minutes (up to 1 hour).

This autoISF feature (new since V.3.0) is much quicker responding than Autosens or dynamicISF to adjust insulin sensitivity to your current „lifestyle state“ (if it is largely (in-)activity related, which often is the case).
For loopers who do not have huge variations in exercise levels in their everyday lives, this feature might fairly much close the gap towards being able to do a 24/7 hands-off FCL.

The Activity monitor is described in more detail on page 9 of the autoISF Quick guide (https://github.com/ga-zelle/autoISF) and in section 6.6 .

While the Activity Monitor takes automatic care of *light* deviations from an average activity level, exercise enthusiasts, or heavy workers, should make use of the “*heavier tools*” (which automatically supersede (shut off) Activity Monitor: *Exercise mode*, discussed in section 6.)

### 5.1.6 Pro/con completely hands-off Full Closed Loop

To stay **24/7** in a **completely „hands-off“ FCL** can be a realistic goal with autoISF, if besides meals also some special challenges, as discussed in this section 5.1, were analyzed and could be addressed. Example see in Case study 5.3.

Clearly it depends very much on your lifestyle, and how interested, willing, and capable you are to recognize, deal with, (and in the future avoid?) situations that get you outside of your desired %TIR on occasion.

> So, this is also about what %TIR you are aiming at, and can accept, as it averages out for the week, for instance.

Everybody must weigh for her/himself

- how much **upfront effort** to put into the setting up process, for getting it all 100% automatic
- **or** whether to take an **easier start, with a couple of situations left to take care of when and as they arise in daily life**

Even if a principal capability for a fully automatic “hands-off” running FCL is given, this still means that

- the user should be knowledgeable about what exactly is going on, and
- have a principal capability to „nudge“, or even to completely take over, in a manual mode.

In the sections that immediately follow, we present the options to nudge or temporarily take over from the AAPS home screen which will be serving as your **FCL cockpit:**

Section 5.2 describes how you can use available "buttons" from your AAPS home screen to manually "nudge" aggressiveness of your FCL.

Section 5.3 will show how you recognize FCL state (aggressiveness) and understand recent loop decisions

**No need to read** section 5.4, unless you are interested in contributing to define/design/program further improvements). **Ideas** for an improved FCL cockpit in the future (probably an over-design; I intend to reduce to a few "really nice to-have" features)

- 

## 5.2 Modulating aggressiveness **manually**, from the DIY-FCL-Cockpit*

* Like in the airplane cockpit: Cruising in full auto mode should involve having an eye on the instruments, and on potential disturbances ahead in the environment.

In section 4. we dealt with major meals.

In section 5.1 we looked into ways to automate also a modified loop response to *foreseeable* situations (tied to a time of day, geo-location etc), or to those *the loop could recognize* (with enough time to react).

However: *Other* **disturbances** might come up, that:

- are not noticeable in-time, or foreseeable, by the loop (*e.g. your plan to start exercise in an hour or two*), but **that influence sensitivity dramatically,** and therefore require temporary modified settings in order to remain in-range, and/or
- **require** a different "starting point" regarding iob and bg, which translates into **a different iobTH** that should **temporarily** be set much lower (*in case of exercise*) or noticeably higher (*e.g. with very fast absorbing carbs in a sweet snack "sin"*) .

*In these scenarios*, you must find an easy way to

- call up a pre-programmed routine for automatic management, with adjusted aggressiveness, or:
- manually tweak a setting or two, to temporarily adjust the aggressiveness

- There may also arise a desire to just exit the FCL mode, and “be your own captain” for mastering a special situation.

Lastly, for peace of mind, to learn, and to stay informed (especially so in your initial tuning phase, or when your glucose curve goes in unexpected ways) we also must be able to…

- …find the key parameters that frame and drive the recent and upcoming loop decisions.

All this is facilitated within seconds right from the AAPS home screen, serving as a **FCL cockpit** after you built a couple of DIY cockpit features via Automations (as described below and in case studies 5.2 and 6.2):

### 5.2.1 Status recognition

Before considering any manual interventions into the ongoing FCL, you should be aware what the current mode of action is (refer to section 5.3), and hence how you might be able to “nudge” your loop in order to adjust to the disturbance that you see coming up.

### 5,2.2 Manual interventions from the (DIY-) FCL cockpit

Trouble with most of these is, not to forget to set back manually, too (=> better solutions in 5.3)

#### 5.2.2.1 Temporary tuning of FCL aggressiveness via temp. %profile or TT settings

The set **% profile** multiplies with both, the ISF resulting from autoISF, and also with the default iobTH you have set, so both are nicely modulated in a linear way with the % temporarily chosen

Just taking profile e.g. to 110% for an afternoon might be an easy way to explore whether you might benefit from 10% more “aggressiveness” in your core settings for lunches (like bgAccel_ISF_weight).
Make sure, though, that the extra 10% are not cut away by set safety limits.

A lowered (relative to profile glucose target) temporary **bg target** (**TT**) signals lowered sensitivity (more insulin need), and
An elevated TT (as often used with exercise) increases sensitivity and hence works in the direction of a lowered % profile to also reduce insulin given by the loop.

Moreover, the **exercise button** (top center on your AAPS home screen) can be activated (turns yellow, then). This will **further boost** how your set TT elevates the resulting ISF, and sharply lowers iobTH, as often desired for sports. (See section 6.1).

#### 5.2.2.2 Making temporary changes in settings made in AAPS/preferences/Open APS SMB

Going into AAPS/preferences/Open APS SMB allows to:

- set milder or stronger …_ISF_weights
- set different iob_threshold_percent (or iobMAX)
- elevate or lower the SMB_delivery_ratio
- limit or expand max. allowed SMB size
- change the the even <-> odd logic for SMB on/off

Doing temporary changes *in AAPS/preferences* should be the exception because

- they require multiple steps, including entering a password
- you will often forget to set everything *back to original* settings, a couple of hours (or already minutes) later.

#### 5.2.2.3 Triggered Automations: Grey extra DIY cockpit buttons for pre-programmed “responses”

Recognizing conditions for fully automatic handling by the loop may not be not possible, or come too late for the loop to act on. Examples would be

- *exercise:* Minimum an hour *before* starting exercise, “the loop should know” to be able to lower iob and elevate bg by the time exercise starts.

- *snacks*: High carb snacks, sweets, consuming ice cream or having a sweet drink comes with the problem of even steeper glucose rises, but overall a lesser insulin need, compared to major meals (for which we tuned our FCL according to section 4).

This not necessarily implies that snacks *need* different settings than a meal. After all, autoISF was designed to react to all available data, especially to where the developing glucose curve is headed. So, depending on your effort to set parameters for a broad variety of meals (notably: how well you avoid to invariably bounce fast against your iobTH), you might be able to accommodate low carb, snack, and major meals with *one* set of settings.

In FCL autoISF, this is a bit more difficult than in HCL autoISF applications, because FCL involves revving up iob supply (largely via big bgAccel_ISF-weights) sometimes too much, to be balanced by just a snack getting absorbed.

In case a snack did trigger a “full meal response”: (1) You probably must continue snacking to prevent a hypo from your initial FCL over-reaction. (2) For future days, analyze your data (and snacking habit) to define how to prevent this from happening often.

For increased comfort and safety, you might have to differentiate, and make use of what follows for the *sweet snack* example, case study 5.2.

Note that in the iPhone versions of autoISF (**Trio** and **iAPS)** there are no Automations . Instead you need so-called **Middleware**, like for instance suggested for %sensitivity (profile ISF) adaptation by one user here: https://discord.com/channels/953929437894803478/1025731124615458848/1238099464531611668

### Tuning aggressiveness

A sweet snack likely benefits from even more aggressive initial FCL performance than set for the meals in your normal spectrum of diets.
Therefore, you could set

- a higher **temp. profile%** and/or
- a temp.elevated **bgAccel_ISF-weight** (see screenshot of my Automation).
- a **low temp. target** (76 mg/dl for instance; this additionally helps maximize the first SMBs that will automatically be triggered at detection of acceleration)..

When first defining and testing this Automation, also check:

- that the safety limits as discussed in section 2 will not block the intended elevated aggressiveness
- SMBs will not get outrageously big, and iobTH sometimes exceeded by too much.

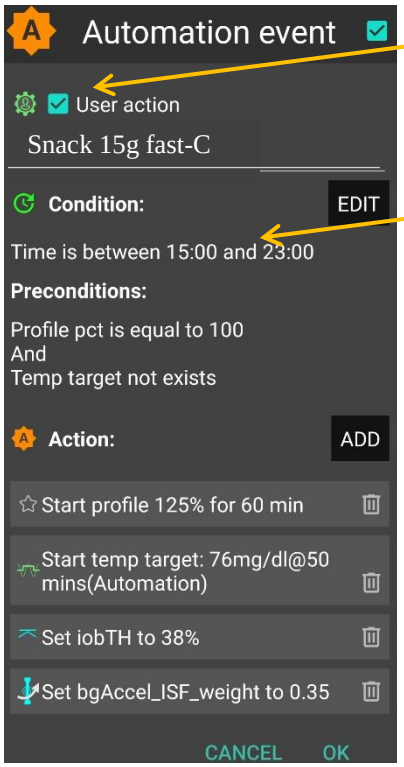
  Note that “the last SMB” is allowed to overshoot the effective iobTH by up to 30%, where it will be cut (or by up to 20% at even target> 100 mg/dl).

### Limiting iob

For “just a snack”, total insulin need will be lower than for a meal.

If you would just have your sweet drink, and your meal-oriented FCL would “attack”, iob likely would become too high, and a glucose rollercoaster would start, with you needing to consume more =>

If you just have a snack, or drink a small glass of juice, you can lower the **iobTH_percent** accordingly.



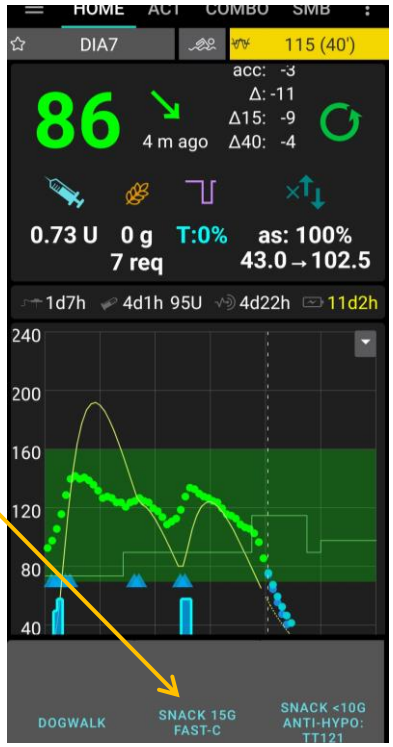

„User action“ is always ticked-on

This will, in the defined time space ..

..offer the “DIY cockpit” button..

..which I must press any time (~90…30 minutes) before my snack.

Note: Make sure that iobTH and bgAccel_ISF_weight revert to default afterwards




So, this can be a little **extra “project” when setting up your FCL**.

You need to research your snack habits (if any), and over time find out which settings in the snack-related Automations work well.

**In everyday life** you **then just** must **press the related button in your cockpit** (which is not time critical at all, except it should be clicked *latest* a couple of minutes after you took the drink or snack).

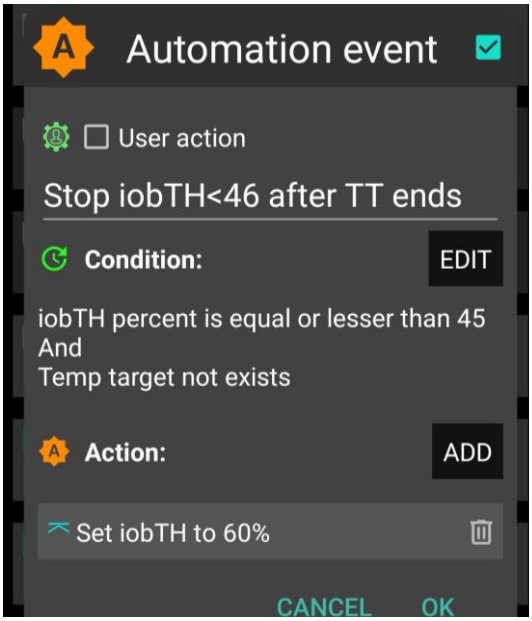
If you consume more, and also eat something with your sweet drink, this will more resemble a full meal… however, with unusual amounts of fast carbs.

**Caution:** Pressing your snack button *a second time* would ***not*** help because the lowered iobTH does not allow iob going high enough. So you are better off just letting your *normal* FCL meal routine run, after your snack mode expired.

Question to developers: Do we already have, or can we get, the option to automatically block out using the same User Action Automation a second time within, say, 2 hours?

Other options (when you just can't stop snacking) would require a manual modulation regarding %profile and/or bgAccel_ISF, but keeping the full default set iobTH_percent, or even elevating it (refer to section 5.2.3). If that happens often, define for yourself an extra User action Automation for a bigger snack (= another grey DIY cockpit button).

**Caution**: Setting a different iobTH or bgAccel_ISF_weight can not be done with a duration attached. Hence you **must** define a suitable **additional Automation, that** must be active in tandem, and **restores the iobTH or bgAccel-ISF_weight** in AAPS/Preferences. Else, once your Automation set in, it will *forever* shift these important parameter settings!




If for instance you have several Automations that, in combination with a set elevated TT also set a lower iobTH: Don't be fooled, the duration only applies to the TT. You need an extra Automation for all of them.
I picked out the highest of the altered iobTH values that these Automations can set (45_percent), and then I can automatically restore my default desired 60% via this one Automation (see screenshot - - > )

### Installing the DIY cockpit button

In the related Automation, just keep the "User action" box clicked at all times, and define in the Conditions when you want to see that button available for cockpit use (see screenshot above) => you will see that button offered.

Besides snacks, also any **other recurring special situations can be addressed via a DIY cockpit button, and receive different aggressiveness up to a suitable iobTH level**.

Over time you can have a big number of User action Automations, and keep them "shelved" rather invisibly (clicked in-active, top left box) in your long list of potential Automations. Even when active, they only show in your cockpit (bottom grey field of your AAPS home screen) *in the time slot you assigned* as potentially relevant.

Discussion

In case you do have a snack habit and …

- can not find settings, as in section 4. defined for your meals, also suit your snacks
- can not pin a time slot or other Condition to it for programming an Automation response as in section 5.1.4

.. then you minimum need a “snack announcement” for which the extra button in your DIY cockpit provides a time-uncritical 1-button-push solution.

This could be a good solution for kids in kindergarten, too. Make sure caregivers understand to use it *only once* for *one* snack. Continued snacking would require iob as for a meals. This is what the FCL loop takes care of automatically; **using the snack button several times in a row would limit iobTH at a too-low level!**

In a software update, we might try to automatically block usage of that type of Automation for 2 hours, after it was once used.

### 5.2.3 Temporarily exiting the FCL

The “last resort” alternative always is to **temporarily** leave the FCL mode, and handle any disturbance “the traditional way” in **hybrid closed loop**.

For this, we switch the automatic aggressive adaptations of ISF to the bg curve OFF that are only needed in FCL ….

(if in hybrid closed loop you like e.g. the dura_ISF adapation still, you alternatively could elect to just set bgAccel_ISF_weight temp. to zero, instead)

Do not forget that, before meal starts, giving a bolus will then be necessary again.

The suggested improved FCL cockpit user interface with an extra version of violet loop on the AAPS home screen (section 5.4.1 ) would facilitate this transition FCL < - > HCL, including automatic removal and re-appearance of the insulin button at the bottom of the APS home screen.

In case this feature is not yet available, you must:
Exit the FCL mode by going to AAPS/preferences/put in your password/OpenAPS SMB/scrolldown to autoISF settings and switch “Enable ISF adaptation..” OFF

(or, alternatively, set bgAccel_ISF_weight to zero).

**Caution:** Unfortunately, there is no way yet for your full closed loop with ISF adaptations to come automatically back on, after a selected time for instance. So **do not forget to switch your autoISF fully back on, later.**

As this will often be forgotten, it may be worth doing a “User action” Automation, for a “temp. FCL OFF” grey button (see section 5.2.2.3).
Caution though, there is very limited experience with this brand new feature. Make sure your Automation definition really applies a duration (or other condition) that will automatically terminate all settings changes it made. As we have seen e.g. in section 5.1.4, this is not always the case.

To recognize whether autoISF currently runs with **ISF adaptation** or not, please consult the **“ai: %” indicator** below the Autosens% **on the AAPS home screen.**

From autoISF 3.0.1 onwards, there is also a very easy way to see effective ISF and effective iobTH in the 1st screen of the **SMB tab**. At the same time, there you see the adaptation of sensitivity to:

- a set %profile change (or effect from Autosens, in case you have that activated)
- a set temporary target
- the Activity Monitor
- +/- exercise mode

So, occasionally (especially in your early set-up phase, after starting of a meal) it is a great idea to study the SMB tab to find out what is going on. See example given in section 5.3.6

## 5.3 Recognizing your loop state in the AAPS home screen

### 5.3.1 Modulated loop aggressiveness via the 3 top buttons in the AAPS home screen

3 Buttons (%profile; exercise; TT) allow temporary modulation of aggressiveness (sensitivity) as may be needed e.g.

- going into meals (EatingSoonTT => providing lower bg “starting point” and more pos. iob),
- or doing exercise (Exercise mode, limiting basal, iob etc for hypo prevention).

,

All buttons grey (GGG) indicates: The loop is running using un-modulated profile parameters

- Note that features like autoISF might still build on profile (regardless whether original, or a temp. modulated profile) for further modifications *(*See calculation cascade for used sensitivity (sens) in the SMB tab. Example see FCL e-book section ).

Any **yellow** (*y* or **Y**) button indicates you are running with a modified *(elevated*, or **lowered**) **profile sensitivity**.

*Actually, the middle button could be yellow without being active, see table below*

| | **Less aggressive** loop = more sensitive user needs temp. less insulin | Standard loop using the set profile | **More aggressive** loop = less sensitive user needs temp. more insulin |
|---|---|---|---|
| % profile | under 100 % (*y*) | = 100% (G) | above 100% (**Y**) |
| Exercise button | ON (*y*)<br>Note that it can | OFF (G) | OFF (G) |
| | | *If ON (y), it is not active but ready to automatically become active whenever any TT above target is set (e.g. by any Automation)* | |
| TT | above profile target (*y*) | = profile bg target (G) | below profile target (**Y**) |
| Example | Exercise mode with weakened %profile, *yyy* | | EatingSoonTT, GG**Y** |
| ..used e.g. in : | Case study 6.2 (*yyy*) | | Case study 6.2 (G_**Y** via Automation)<br>Case study 5.2 (**Y**G**Y**) |

Case study 6.2 uses extremes from both ends of this sensitivity modulation spectrum:

- *yyy* for reducing basal and iob, and getting weaker ISF, during exercise
- GG**Y** for temporary “EatingSoonTT” boost when, in FCL, a high carb meal had
- The even stronger **Y**G**Y** case is contained in Case study 5.2, where for a high carb snack an extra boost, using >100% profile sensitivity, is temp. added

### 5.3.2 Color scheme of top cockpit buttons tells kind of closed loop that is running

3 Buttons (%profile; exercise; TT), each in 2 states, yellow (for modified), else: grey (G)

Note that yellow could be less (*y*) or more (**Y**) aggressive than standard profile

GGG is the loop running with set profile. Any yellow setting modulates sensitivity

*or attempts to modulate, whenever another required condition may become true (e.g. also via an Automation)*.

Overall, 2 exp 3 = **eight principal FCL modes** are possible. They need to be **differentiated further**, based on whether the modulation…:

- …goes into the **more aggressive** (**Y,** higher % and/or lower TT), *or…*
- …the *less aggressive* (*y, lower % and/or higher TT and evtl exercise*) direction.
- Note the Exercise button works only one way, to make the loop less aggressive.

DEV idea: Should we make the **Y** to violet or orange (more aggressive) and keep the *y* yellow (less aggressive than profile)? Then beginners would easier recognize whether they go more or less aggressive …

| Color combinations | Lower aggressiveness | …….. | Higher aggressiveness |
|---|---|---|---|
| GGG | | profile | |
| G*yy* …. G*y***Y** | dynamic exercise mode (if TT > profile target) | | EatingSoonTT (as TT < profile target: middle *y* inactive, G*y***Y** = GG**Y** 2 ) |
| *y*G*y*-*y*G**Y**- **Y**G*y*-**Y**G**Y** | „traditional“ exercise mode (if <100% sens AND TT > profile target) | mixed cases 1) | boosted EatingSoonTT (if >100% sens AND TT < profile target) |
| G*y*G | | G*y*G = GGG 2) | |
| *y*GG …. **Y**GG | Loop adjusted for period of lower insulin need (e.g. 80% for night after exercise) | | Loop adjusted for period of increased insulin need (e.g. 120% during sickness) |
| GG*y* ….. GG**Y** | Loop running with temp elevated target e.g. for enhanced safety against going low (HypoTT) | | EatingSoonTT (if TT < profile target) |
| *yy*G …… **Y***y*G | same as *y*GG | *y* inactive 2) | same as **Y**GG |
| *yyy* - **Y***yy*- *yy***Y**- **Y***y***Y** | dynamic exercise mode further modulated by <100% profile | mixed cases 3) *yy***Y** = *y*G**Y**, 2) | Boosted EatingSoonTT (if >100% sens AND TT < profile target)( YyY = YGY, 2) |

1) *y*G**Y** would be the combination of setting a milder % profile, yet orienting the loop to a lower target. Usually we do not use this combination, but it would be OK for an instance where we want to start the next meal at a low target, but we want our loop to be careful and not rush getting there.

**Y**G*y* would be the combination of setting a more aggressive % profile, yet orienting the loop to a higher target. Usually we do not use this combination, but it would be OK if we believe we are in a state of enhanced insulin need, and want to get to a set elevated TT (which could be what we want to go max down to to have room for exercise, or just to be sure not to go into a hypoglycemia if we misjudged the symptoms about enhanced insulin need.

2) A yellow middle button *y* works really only in the combination ..*yy*. Any combination ending *y*G or *y***Y** will work the same as if it were ending GG or G**Y**. Allthough the yellow middle button does not make a difference in these cases, it might be "justified" by the loop showing its readiness to employ the exercise mode, whenever a TT > profile target comes around, e.g. via an Automation.

Same as GGG, only *if* (e.g. an Automation) sets *a TT>profile target*, automatically a softer response due to additional exercise mode kicking in *then*.

3) *yy***Y** would essentially working like *y*G**Y** (discussed under 1), because the middle button, whether y or G, makes no difference with an elevated TT.

Again, this the combination of setting a milder % profile, yet orienting the loop to a lower target. Usually we do not use this combination, but it would be OK for an instance where we want to start the next meal at a low target, but we want our loop to be careful and not rush getting there.

**Y***yy* would be the combination of setting a more aggressive % profile, yet orienting the loop to a higher target. As opposed to the **Y**G*y* case (discussed under 1), in **Y***yy* the Exercise mode is active and strongly reduces aggressiveness. It is hard to imagine, why this combination this combination of settings could make sense; maybe if a "long waved" state of reduced insulin sensitivity meets a case where normally G*yy* would be sought (see there).

### 5.3.3 Active Automations (or middleware in Trio, iAPS)

It is important to be aware which of your Automations (see listed in your AUTO tab, or via the top left burger menue, in AAPS) is active to eventually "kick in".

The ones where you clicked “User Action” should also show as an extra grey button on the bottom of your AAPS screen.

Note that an **Automation might not be permitted** to change settings by **“killing” another still running Automation** (always consider that, when putting the duration into your Automations!). For instance, you cannot switch from 130% profile to 110%. Either the 130% times out, or you **need an extra “in-between” Automation that terminates** the 130% under described conditions (example see in Case study 6.2).

This “design” is for a good reason: The assumption here is, that your 1st Automation (the 130% in the example) is designed well and runs for a reason. It should either “get finished” when the job might be done (and kick in again, if not), or, in exceptional cases, it should be consciously terminated by another well thought through 2nd Automation (describing the conditions in which you would find that other Automation more important than “finishing up” the one that was already running). That “in-between” Automation makes the loop return to base profile, which is a signal *to all Automations*, to now check whether any conditions exist, to activate a 3rd Automation (as in example of Case study 6.2).

Advice: Try to stay away from Automations that also aim at temp. modifying aggressiveness (e.g. temp. setting different bgAccel_ISF_weight). For the reason just given in above note, they often will not kick in anyways.
Generally, it also is no good idea to double up sub-algorithms for tweaking loop behaviors (“loop inside a loop”).

### 5.3.4 FCL related indicator fields in the AAPS home screen

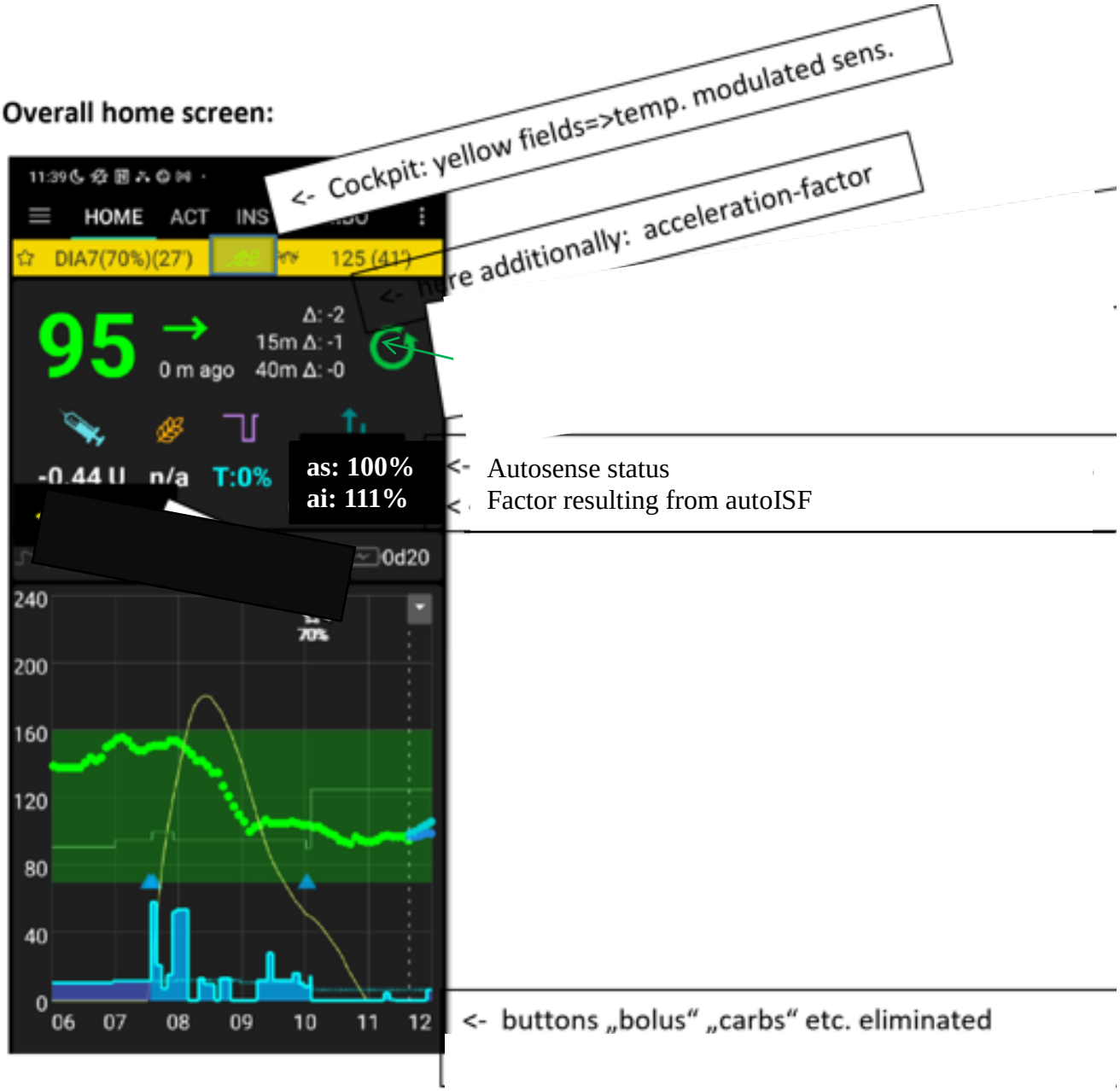
In extra data fields of the AAPS main screen you can always see (not change) the key „aggressiveness“ parameters your loop currently operates operates with (see also home screen example below):

- To recognize whether autoISF currently runs with **ISF adaptation** or not, please consult the **“ai: %” indicator** below the Autosens% **on the AAPS home screen.**
- Details for every loop decision see result/debug section of the **SMB tab**.
- The AAPS home screen additionally shows, above the deltas, the current **acceleration**

  Having a look at that can be valueable. For instance, when glucose is relatively low and still falling, a positive (and getting more positive) acceleration indicates that bg will swing back up, rather than crash low. This will give info about necessary snack size, and hence help avoid both, unnecessary calories, and going on a bg roller coaster.

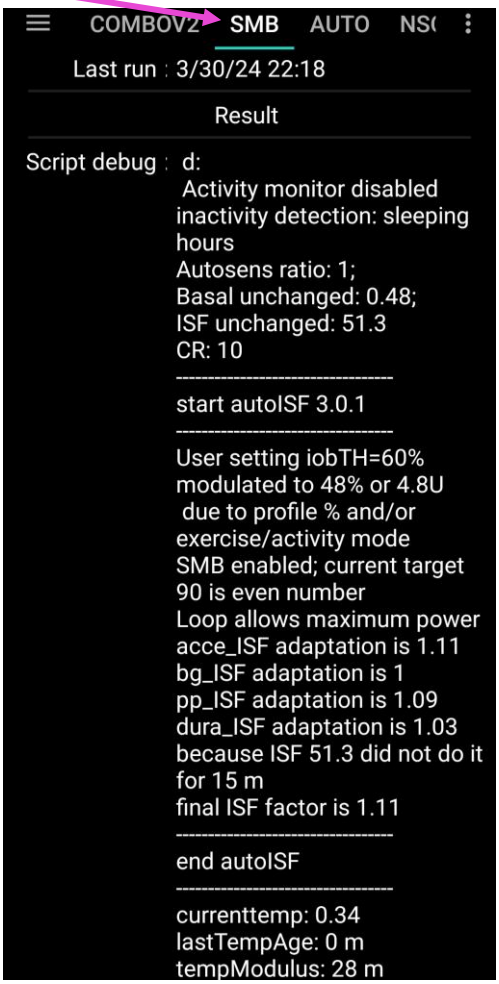
- 5.3.5 Overall home screen: *(do a new picture*

**Overall home screen:**



### 5.3.6 Info given every 5 minutes in the SMB tab

(If **tab** not shown -> Config.builder/APS: tick box on the right)




When clicking on the SMB tab, you see how your standard and temporary settings, as well as the latest bg and iob status, influenced the last decision of your FCL.

Example 1: A 80% temp. profile modulates 60% iobTH to 80% of 60% = 48%:


The profile ISF of 41 mg/dl/U got modified by the set 80% temp. profile to 41/0.8 = 51.3 mg/dl/U, called “ISF unchanged” (before “start autoISF”).

autoISF applies the 4 sub-categories (acce, bg, pp and dura_ISF), and depending on the bg curve form suggests various ISF changes.

The final resulting factor “sens” (see flowcharts in section 3) is 1.11 (in our case, driven by bgAccel_ISF). This changes the 51.3 “unchanged” ISF to 51.3/1.11 = 42.4 mg/dl/U

Further down in the SMB tab, you can see how this ISF is applied to define the SMB size to be given, and whether any limitations – notably by autoISFmax, max possible SMB size, or maxIOB – cut the amount.

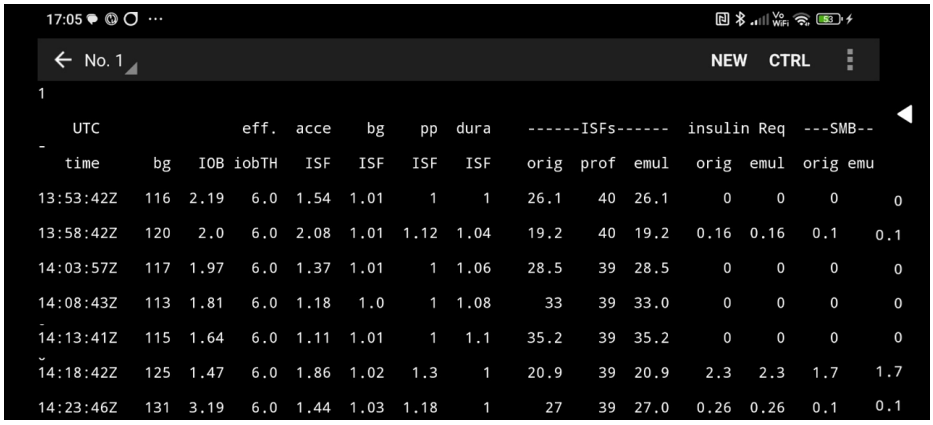
| Message | Condition | What does it affect? |
|---|---|---|
| Loop allows maximum power | even target < 100 | increase in bg limited to 30%, otherwise no SMB;<br>actual SMB delivery ratio is max of fixed smb_delivery_ratio and linearly growing ratio |
| Loop allows medium power | even target >= 100 | increase in bg limited to 20%, the AAPS default, otherwise no SMB; actual SMB delivery ratio is either fixed smb_delivery_ratio or linearly growing ratio |
| Loop allows minimal power | odd target | no SMB, only TBR available for action |
| Loop power level temporarily capped | IOB > effective iobTH | Last SMB capped to stay below iob threshold + 30% overrun;<br>IOB getting above user defined iobTH, potentially modulated by exercise mode, activity monitor and profile percent |
| Loop allows AAPS power level | no even/odd target option active | SMB enabled/disabled according to standard AAPS rules and settings;<br>no iobTH threshold is active |

Note that in the SMB tab you can– in “real time” - capture and analyze *only one* decision.

### 5.3.7 Info about last 15 autoISF decisions

Refer to section 11.2.2 for an option that enables extended analysis of the on-going ISF modulations from autoISF. (To do this on your **Android** loop phone requires QPython and a logfile emulator).

You get tables like this …




| 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 |
| 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 |

… which gives you a quick orientation about recent loop decisions, and relative contributions of the various autoISF contributors (ace, pp, bg, dura).

A similar table is available also on the **iPhone** for Trio and iAPS users of their autoISF variants.

### 5.3.8 SMB tab info when operating in 1-minute mode with Libre3

*Users: Please supply text and screenshot*

### 5.3.9 Summary: Your personal FCL cockpit (for maneuvering through disturbances)

A lot of avenues were shown that could help you or your loop maneuver through a variety of "disturbances". You should not have to try out many of them, and (like the author), you should find a way to narrow it down to what really helps in **your** everyday T1D management.

Try to **keep things as simple and clear as possible.**

Especially, do not pre-maturely rush into setting up Automations as quick over-patches for what you may not like to see. Limit the number of Automations, and further limit which ones from your list are every-day active (vs. switched off, and only ticked active for special days).

## 5.4 Modulating aggressiveness manually from the improved FCL-cockpit

***Skip this section 5.4 (next ~8 pages. Continue with section 6)*** *unless you are deeper interested in discussing further user interface upgrades. Actually, some suggestions made are probably an "over-design". After trying a lot of options for refinements out, the author returned pretty much to a "keeping-it-simple" route.*

*My main suggestion is to get that violet loop button (sections 5.4.1-5.4.2), something I think many would use - very handy certainly in the setting-up stage, too, for easy switching between the "old" HCL, and new territory in FCL.*

autoISF is an early dev variant of AAPS, and as user you are participating in an on-going development. Of note, autoISF 3.0.x was launched without many of the cockpit features that are suggested below in green font color.

> Only what is written in black is currently of some relevance for using autoISF.
> **No need to read** any of the green lines, unless you are interested in contributing to define/design/program further improvements.
> This is also an open invitation for you to contact us in case you could help program a module for one of the suggested user interface extras.
> For future integration into AAPS Master, an eye should be kept also on the question which *other* modes (like FCL using Automations and others mentioned in section 13; and maybe also HCL) might benefit from some of the extra features.

Keep in mind, that the **goal should be to interfere with the loop as little as possible**. Under certain conditions, it can run **fully automatically** without any user interaction, as described in the preceding section 4. + section 5,1.

Just like in the airplane **cockpit**, also cruising in full auto mode should involve having an eye on the instruments, and on potential disturbances ahead in the environment.

> *E.g.: storm ahead => instruct your plane to climb to another flight height.*
> *Anology: exercise ahead => setting an exercise TT, or => pressing a button that activates a sequence of instructions (some of them probably hinging on conditions, like actual iob), how to manage through that exercise situation).*

So, **for the occasional „disturbance" coming up**, you should find an easy way to

- call up a pre-programmed routine for automatic management, with auto-adjusted aggressiveness, or:
- tweak a setting or two, to temporarily adjust the aggressiveness

- There may also arise a desire to just exit the FCL mode, and “be your own captain” for mastering a special situation.

All this is facilitated within seconds right from the AAPS home screen’s **cockpit features** to the extent they are already incorporated, or to the extent you can build alike DIY cockpit features via Automations, as described in section 4.1.3 and case studies 5.2 and 6.2):

- The button that is integrated into the **violet FCL icon** serves as emergeny off button, to quickly stop FCL, or to at least to immediately stop any more SMBs (…just for a couple of minutes, or for the remaining meal time: pick from the options offered with just one keystroke).

  Via the violet FCL icon on your AAPS home screen, you also can access a temp. switch-off button for SMBs (see section that next follows below).

- The **three top fields** (%profile, exercise, TT) provide access to temp. tuning of core parameters, and/or to some pre-programmed routines.

Taken together with some **new indicator fields** about your loop state (section 5.3.3 and 5.3.4), and the **grey DIY cockpit buttons** (section 5.2.2.3) this makes the AAPS home screen your **cockpit** for Full Closed Looping.

Let us look on each of these suggested cockpit elements in some detail:

### 5.4.1 Violet FCL icon and underlying buttons

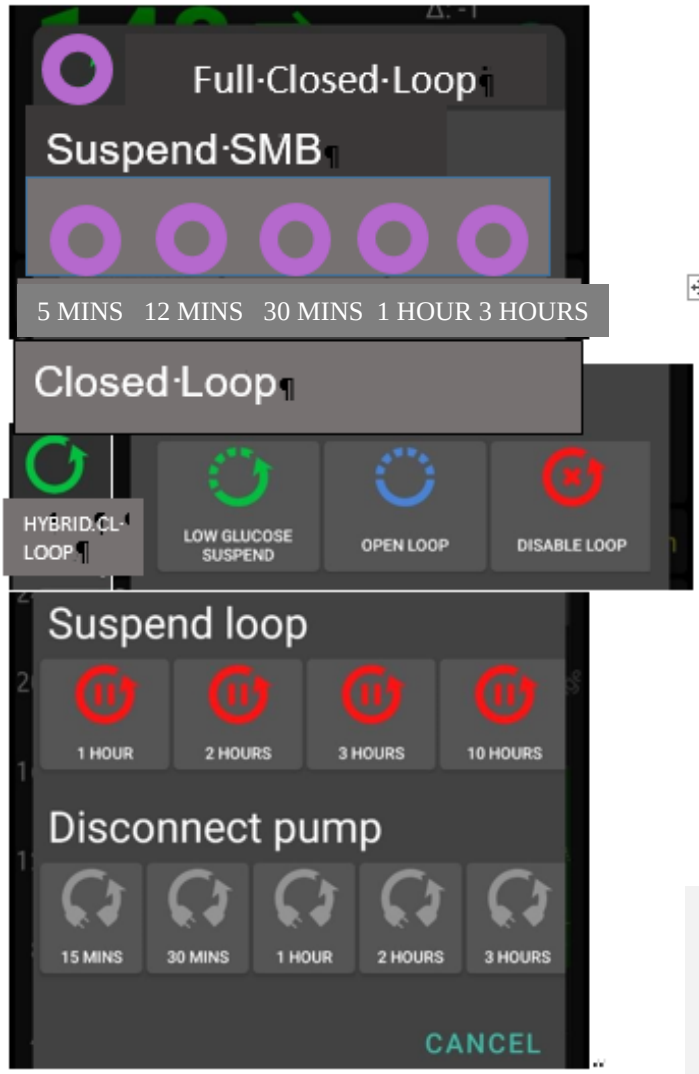
Novices to FCL, or really anyone running into a very special situation, may appreciate that the new closed loop icon on the AAPS home screen in pink (for FCL) has buttons to quickly shut off getting more SMBs (1st row), or to enter other loop modes (second row).

It functions very much as the other ones that you know from HCL already, and in fact you get offered some of the same options (for instance, to switch the (full) closed loop off for 15 minutes for going to take a shower)

Note that in FCL you leave all BG regulation, notably against meal spikes, to the loop. So, try not to disconnect in phases when your FCL must ramp up your iob.

The required insulin would still be supplied *after* you reconnect. However, without the user pre-bolussing, the delay would be more of an issue in FCL than it had been in HCL.

Just pressing on the FCL icon,  a dialogue box comes up:




<-add arrow, form like like Cl.Loop

<-dotted + arrow, form like like LGS

DEV: when in Hybrid Closed Loop = top headline + green circle), there is no Suspend SMB part but it starts right with „Closed Loop". In that screen. There, the 1st element must be option to pick

.....pink circle, FULL Cl. LOOP

where here the green HYBRID CL. LOOP stands

For loopers who did not set up FCL, a feedback must come up ~ „FCL not installed" if they press on that button ¤

*Skip*:
= This User Interface is a not implemented idea only.
Please contribute, or wait for update with the missing info

Pressing „**Suspend SMB**"provides fast and easy „emergency braking" regarding delivery of more SMBs:
Select the one with the desired number of minutes:  5 or 12 for just blocking the potential next SMB(s), and up to 3 hours to manage the entire rest of this meal with %TBR from then on.

Whenever, and why-ever, your FCL is in „no SMBs allowed" mode (e.g. automatically after surpassing an iobTH , or triggered by a set odd TT), the FCL icon will turn into a dotted one.
Instead of remaining **duration to end time** it indicates in the middle „the condition", „**iob**"  or „**TT**
Add an indication **if** suspend SMB comes from an Automation, e.g. add an **„ (A) „ underneath** the #minutes, iob, or TT in the middle of the dotted violet field.
So, as in other (already in HCL existing) cases, those icons show in the middle the minutes left that they will be running, or the condition which would have to go away for this temp. setting to stop.
It always auto-reverts into the FCL state and FCL icon, when time (or other condition) has elapsed.

Pressing „**HYBRID CL. LOOP**" or other buttons from the 2nd row  provides fast and easy „emergency **exit" into other modes**.

This enables beginners an easy „temp. escape“ into their well-known HCL (green) at any point of time. bgAccel_ISF_weight is set to zero when going FCL->HCL. HCL can run with autoISF (for instance dura_ISF) uninhibited otherwise. (check implications for HCL users of autoISF ?? ).

Note: These options from row 2 have no time limit. Loop will **not** by itself go back to FCL. You see the different loop icon as a reminder to manually revert, when ready.

### 5.4.2 Buttons „Insulin“, „Calculator“ etc at bottom of AAPS home screen

These buttons are **not useful any longer in FCL**, and automatically disappear whenever in FCL mode (also in Suspend SMB state), and re-appear when leaving FCL. This applies also when an Automation or technical system failure shut off FCL.

Users who, maybe in the beginning phase, feel better having those buttons, can override the removal (of the insulin button, or any other) by going into /preferences/overview/buttons and forcing them on. They only remain on until the next re-entry into FCL mode, when auto-off happens again.

The reason why we do this: It really is important to let the loop loop, and not interfere more than absolutely needed. Any bolus the user gives will sure distort the bg curve, on which autoISF, especially when aggressively tuned for FCL, builds a lot of its decisions!

### 5.4.3. Three top fields (%profile, exercise, TT)

Depending on the variedness of lifestyle, the desired %TIR, and the initial tuning effort put in, the user may want occasionally **to „tweak“ the aggressiveness of her/his FCL**.

The top 3 fields (grey in default mode, **yellow when temp. in mode with changed aggressiveness**) serve as quick and easy entry points to make temp. switches (as users will be used to for %profile switches, or for setting an EatingSoonTT in HCL, .. which they still can do in FCL … but more:)

Expert FCL users might need this feature rarely, but probably at least to manage activity after meals: Each require opposite aggressiveness, and the switch has to come in a certain point in time that would be difficult to capture. (More see section 6.4 )

#### Information printed on the top buttons

The yellow TT field shows the currently valid TT (and further duration):

(profile) stands for the abbreviation you labeled your selected running profile


In the special case of settings for meal preceding sports, the field will look slightly differently:


…and …

… when iobTH is first time exceeded, this automatically switches to:


Likewise, if on the AAPS main screen just an **EatingSoonTT** is set (e.g.72), this is entered with the desired duration. Afterwards, it automatically reverts to profile target and the display turns grey again (can also turn green first, for a short transition period), with the profile target (e.g. 90) on it. (No time limit, then, for the profile value as set in preferences).

Without sports context, the middle field remains grey.

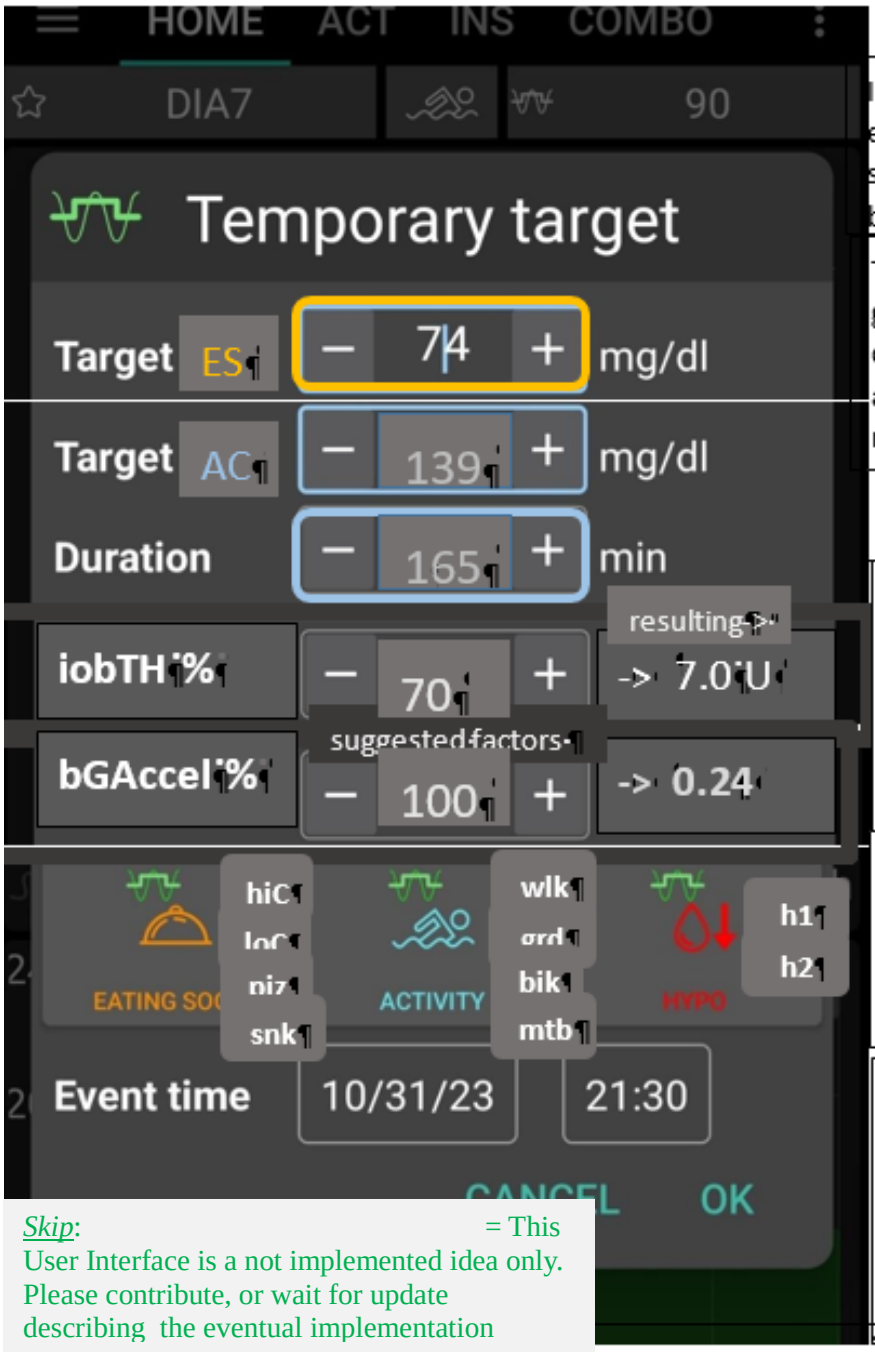

Independently from setting a TT, the user can choose to set a **%profile in the left top field**, for an independent number of minutes, e.g. 70% in this screen example: Also, or additionally, this will influence the resulting ISF and sensitivity%

The % might change and turn yellow also in context of making TT inputs in the related dialogue box (see chapter TT dialogue field, above). Still, the % (or the length of time the profile switch shall be active) can be independently overriden in the top left field, if so desired.

If an **Automation** sets a %profile, and/or a TT *(e.g. automatic detection of meal start  at condition e.g. when delta >10*) , this would automatically show in respective field(s) turning yellow and showing the temp. setting. To show the set parameter comes from an Automation,  „ **(A)** „ is added in the end of button text.

#### 5.4.3.1 TT dialoque field ( Currently not available in the pictured form and function ! )

The TT field (top right of AAPS home screen) is a primary daily interface, and a dialogue field opens when pressing on it




Initially, the form can be entirely empty re. TT inputs and just show the default iobTH and bgAccel parameters.

Two lines appear for **target** and get labeled ES, AC or HY depending on what was pushed at the bottom buttons. (HY => red frame; evtl 1 empty => grey)

TT entries can be made or overridden.

**iobTH** calculates from Target AC and other settings shifts away from default set in /preferences (here 0.7) but could be overridden here 0...200 % is allowed

**bgAccel** ISF_weight can be modulated here, too Note: It can change again if %profile is also changed. 0...200 % is allowed

**CANCEL** allows to start fresh (select one or two of the square buttons, ES, AC or HY

.

**OK** needed to use the settings

**Duration** input is made in minutes. In the exceptional case that both, ES and AC targets are defined, the duration input is for AC and framed blue. (This is because the preceding AC mode is automatically determined in length by the loop observing when iobTH is exceeded

*Skip*: = This User Interface is a not implemented idea only. Please contribute, or wait for update describing the eventual implementation

This looks complicated but only because it allows 4 different modes of use. Each user will primarily use her/his preferred one.

(1) Who is happy with the initially well tuned FCL and does not have huge variations in daily eating and moving around, will **not use** the TT **at all**. FCL is possible without an intervention via the

TT button in your cockpit. Actually 4 of 8 modes (GGG …YYY permutations, list see section 5.4.1) are not making use of TT.

(2) Super easy is also, to just input **any odd-TT** (odd-numbered temporary target) that will shut out any SMBs for the set duration. *That can be a good idea when having a small snack, for instance*.

Super quick access to stop SMBs is possible also via the loop icon (section 5.3.1).

Specifically, an **EatingSoon TT** can be activated here (limited relevance see section 2.5). It is time-un-critical, can be manually set, or come up via an Automation.

The cockpit enables you to set the iobTH differently (override) for the current meal.

Alternatively, iobTH can be temporarily changed in /preferences or using an Automation.

Temp. iobTH will always revert to default when the TT expires. If another TT immediately follows, like in the example of the screen above, it will calculate, (then) show and use a new temp. iobTH.

(3) The third way is to **use the input mask** (*if already ncluded in your software version* see picture above) **to freely modulate the loop aggressiveness** for a declared number of minutes. Click the bottom big square(s): Either HYPO, or ACTIVITY, or EATING SOON, or ACTIVITY and EATING SOON (*example in the pictured screen above*). Make or override entries in the offered fields. Press OK.

(4) The fourth way is to exclusively use one of the 4+4+2 little buttons seen in the bottom part of the TT dialogue box *(if already included in your software version)*. They provide a set of settings (as will immediately show in all input fields above) that the user has set up in Preferences/SMB/autoISF/FullLoop (refer to section 6.3), and can freely label there. *For instance „hiC“ at high carb EatingSoon, „piz“ for Pizza/fatty meals, „grd“ for garden work, „mtb“ for mountain biking …*

**Capturing good settings for not-everyday situations in /preferences** *(if already included)* **allows calling them up within 1 second**, from your cockpit on the AAPS home screen (…and won't ruin the FCL experience at all , especially because in most cases it is not time-critical, how long before the intended exercise the buttons are pressed).

Case study 6.2 demonstrates that nearly the same performance and comfort can be reached via the **DIY FCL cockpit** with the grey extra buttons appearing at the bottom of the AAPS home screen, based on Automations with User action (see also section 5.2.2.3).

The example picture given above, and also case study 6.2, is the most complicated (but also most useful) case, **when exercise follows after a sizeable meal.** It is then that you need (a) aggressive FCL initial performance at the meal, but, *exactly when* (!) a (for the intended sport already temp.lowered) *iobTH is exceeded*, you need (b) to have SMBs automatically switched off and go into the „milder“ mode, as defined for the exercise (with *high* instead of the immediately prior *low*TT, that automatically significantly reduces iobTH again, and insulin sensitivity(resistance) settings too).

Pressing exercise related buttons will automatically also light the **exercise button** on the main screen yellow.

To summarize, the TT dialogue field offers easy but powerful ad-hoc modulation of loop aggressiveness for FCL (if already included).

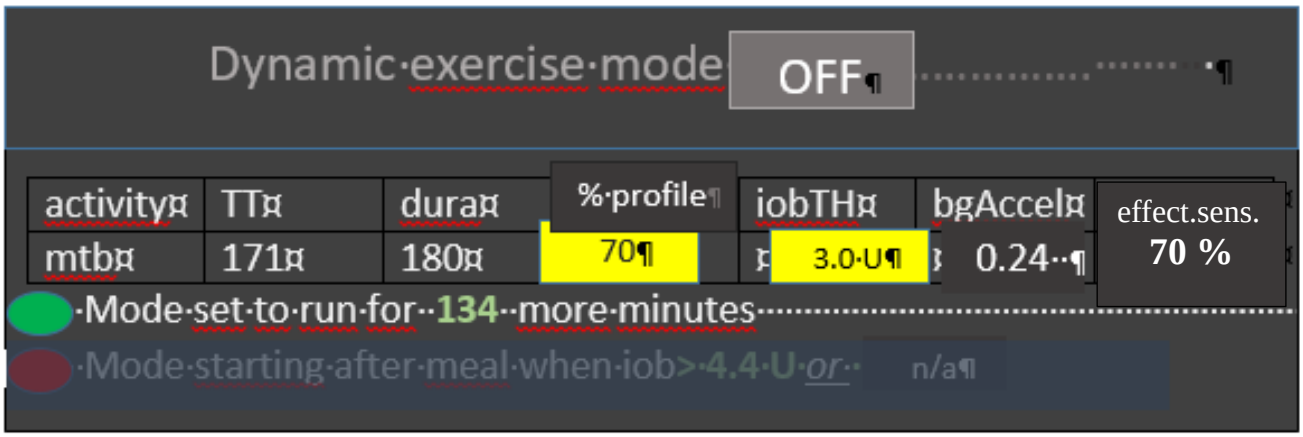
5.4.3.2 Exercise button (see more in section 6.)

The exercise button automatically lights yellow when exercise related TTs are activated in the TT dialogue box.

4 of 8 principal FCL modes (section 5.4.1) are making use of the exercise button.

If pressing on the exercise button, a dialogue box appears *(if extended design for FCL cockpit is already launched)* with info on exercise setting first (and opportunity to override), plus below the activity monitor (experimental for auto-tracking of lighter movement during the day, and effects on sensitivity that may have. *See* section 4.5).

So, first the exercise settings (as set under TT) are there to read. Example :



The exercise (here mtb) is selected in the dialogue box of the neigboring TT field, and there auto-filled with settings made in the set-up and tuning stage by the user under preferences. They are reported also under the exercise button here, and TT, duration, and % sens (the temp. profile sensitivity that also shows on the %profile field on the left side of the exercise button) can be temp. changed there.
iobTH, bgAccel_ISF and overall resulting effective sensitivity ratio (effect.sens. %) is given in the other fields.
The **middle field** of the table, **„% profile“** either picks up the % set under the %profile button, or an input can be made here, in the exercise button domain, which will:

- turn the neighboring %profile button on yellow and show that inputted % on it, too
- be multiplied with the result from the exercise mode settings per se, and change the % overall, accordingly.

So, if this middle field of above table (dialogue box of sports button) contains a figure other than 100, input field becomes yellow,and you are operating with a combination of traditional PLUS new exercise mode (with all three top buttons of your FCL cockpit yellow). This maximally will soften aggressiveness, for which you get an idea by the last calculated figure.

The mode is either running already (for another number of minutes, as probably also shown in the yellow TT field anyways). Or it is scheduled to run, after insulination for a started meal reaches iobTH (as in table). Or, no exercise is scheduled (both points red, no entries.

The lower part of the exercise dialogue box (not pictured above, but see in section 6.5) is dedicated to the Activity Monitor

#### 5.4.3.3 Profile button

The profile button can still be used to set a different profile, or profile%, for instance to adjust for days with sickness (as you are used to from hybrid closed looping). 4 of 8 modes are not making use of the profile button.

Any inputs made here will be used to modify profile_ISF on which all further changes are made on (multiplied with).

The profile field remains grey if standard profile is applied.
It turns yellow, displaying a %figure relating to any altered loop overall aggressiveness:

- When no inputs (changes from 100% profile) are made here, but inputs in the TT field, e.g. for exercise, automatically lead to different effective sensitivity ratio

- when% is changed by input in the profile button itself, it will be multiplied with with profile_ISF and be used in place of profile_ISF *by the algorithm*.

However, for exercise (sports) you no longer must make an entry here, because reasonable %reductions should be automatically provided, driven by your set TT (and half-basal exercise target), see section 6.

### 5.4.4 FCL related indicator fields in the AAPS home screen

In extra data fields of the AAPS main screen you can always see (not change) the key „aggressiveness“ parameters your loop currently operates operates with (see also home screen example below):

- how profile sensitivity (**ISF**) adjusts by the %profile input, by autoISF, and/or a set exerciseTT, resulting in an effective sensitivity (ISF that is used to determine insulinRequired. Details for every loop decision see result/debug section of the SMB tab).

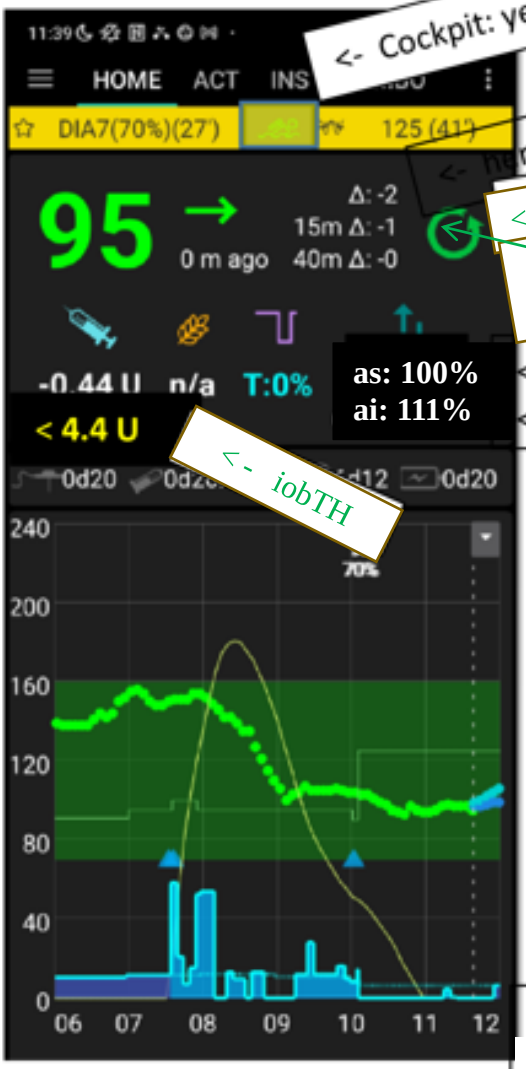
- next to current available iob number is an indication of your **valid iobTH** (the iob above which no more SMBs will be given)

- The AAPS home screen additionally shows, above the deltas, the current **acceleration**

  Having a look at that can be valueable. For instance, when glucose is relatively low and still falling, a positive (and getting more positive) acceleration indicates that bg will swing back up, rather than crash low. This will give info about necessary snack size, and hence help avoid both, unnecessary calories, and going on a bg roller coaster.

### 5.4.5 Overall home screen with discussed further features:

**Overall home screen:**




<- Cockpit: yellow fields=>temp. modulated sens.

<- here additionally: acceleration-factor

<- Violet < - > green circle for FCL < - > HCL dotted if SMB off
in the middle: minutes counting down if temp.set;
below: (A) if coming from an Automation that is running

<- Autosense status
<- Factor resulting from autoISF

<- iobTH

Note: iobTH is currently not shown here, but can easy be found in SMB tab, see next page

<- buttons „bolus“ „carbs“ etc. eliminated
(auto- re-appearing when violet –> green loop)