7. Kids: Mastering additional challenges… *(NN)* V2.2

Please note that with autoISF 3.0 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

No medical advice
research paper

- - - - - - - call for contributions, chapter and case studies - - - - - - - - - - - - - - -

-

Points from section 1.6 that need to be detailed here :

To establish and maintain a FCL for kids brings about some extra challenges if:

- Lyumjev is not available or well tolerated
- Hourly basal rate is very low, providing a poor basis for big SMBs *=> Measures to get sufficient initial SMB sizes?*
- Diet is rich in sweet components. With the typical low blood volume of a small body, strong tendency towards very high bg spikes! *=> Provisions for erratic hi carb snacking/ the occasional sweet drink? (see* **case study 5.2** *- is this applicable ?)*
- Going through marked changes of insulin sensitivity or of circadian pattern makes it difficult to keep the FCL appropriately tuned.

  This problem is about the same in Hybrid Closed Looping. However, now you might expect miracles from the FCL. This is not going to happen. You still should try to set appropriate temp. changed profiles, that serve also as a basis for your autoISF FCL.
- Discipline is poor regarding keeping Bluetooth connectivity and infusion sites perfectly running *(=> alarms on parents phone, see section 1.3 Automation example ?)*
- Between kid and supervising parent it must be guaranteed, especially in the initial weeks, that an eye is kept on whether the FCL is working about as to be expected.

  FCL => implications for remote parent monitoring ? (may have to come at the end, when all potential problems were discussed)

- Extra caution needed re. elevating the **SMB delivery ratio ?**
  - The fixed 0.5 value in AAPS was installed also with a consideration on user/follower (parent) set up and limiting potential problems from a bolus being initiated from both phones in parallel
  - As in FCL neither phone should initiate a manual bolus, this precaution seems not really needed (dev – check)

The DIY Cockpit (see quoted below, from section 5.2.2.3) allows to have buttons dedicated to kindergarten hours etc that can in a time-uncritical manner activate special routines – **should be very helpful to custom program Automations for kids**!

---quoted section 5.2.2.3 ------------------------------------

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 "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 act 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) often too much to be balanced by just a snack getting absorbed.

For that reason, or just for increased comfort and safety, you might want to differentiate, and make use of what follows for the *sweet snack* example.

Tuning aggressiveness

Key is that a sweet snack likely benefits from even more aggressive initial FCL performance than the meals in your normal spectrum of diets require.
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 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

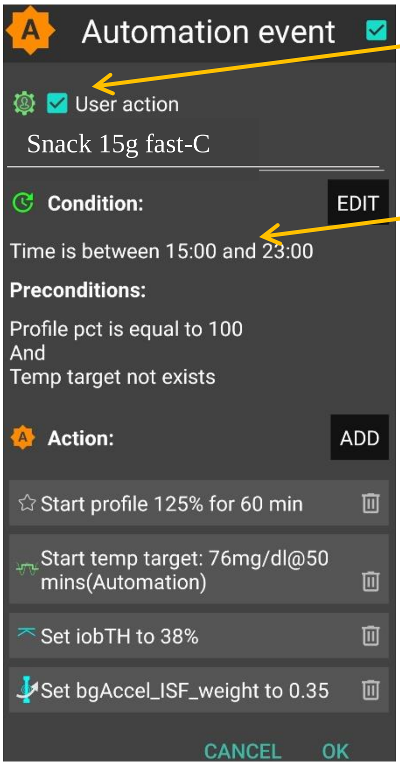
Note that “the last SMB” is allowed to overshoot the effective iobTH by 30%

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 glass of juice, you can lower the **iobTH_percent** accordingly.



Automation event ☑

☑ User action

Snack 15g fast-C

Condition: EDIT

Time is between 15:00 and 23:00

Preconditions:

Profile pct is equal to 100
And
Temp target not exists

Action: ADD

Start profile 125% for 60 min

Start temp target: 76mg/dl@50 mins(Automation)

Set iobTH to 38%

Set bgAccel_ISF_weight to 0.35

CANCEL OK

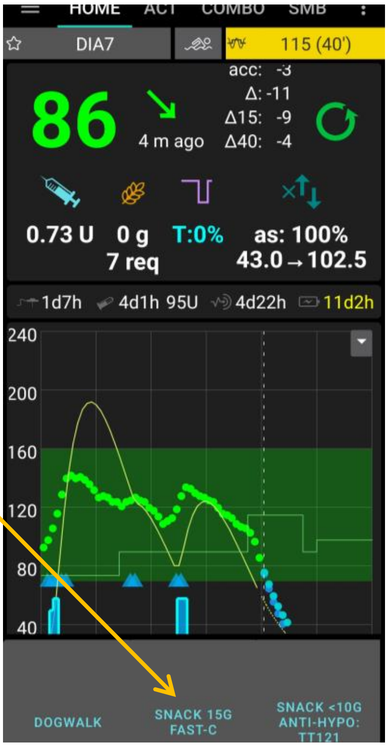
„User action“ is always ticked-on

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).

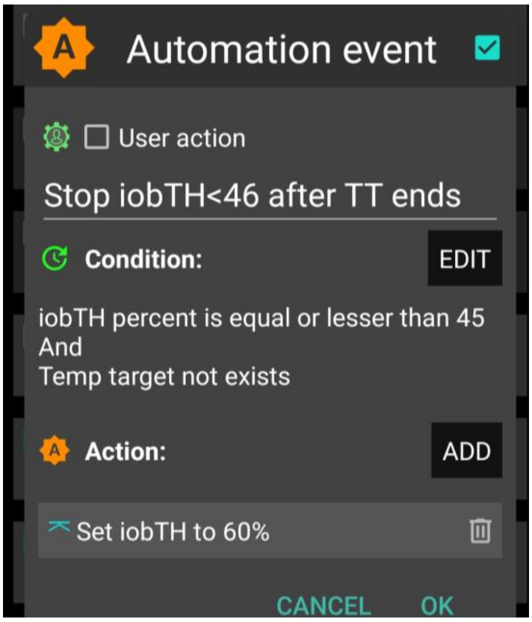
If you consume more, and also eat something with your sweet drink, this will more resemble a full meal… however, with unusual amounts of fast carbs.

Note: 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.

Other options when snacks keep extending 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: If (as in autoISF 3.0) setting a different iobTH or bgAccel_ISF_weight can not be done temporarily (i.e. with a duration attached) you **must** define a suitable **additional Automation, that** must be active in tandem, that **restores the default set iobTH or bgAccel-ISF_weight again**. Else, once your Automation set in, it will *forever* shift this important parameter setting!

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 - - >)



Automation event ☑

☐ User action

Stop iobTH<46 after TT ends

Condition: EDIT

iobTH percent is equal or lesser than 45
And
Temp target not exists

Action: ADD

Set iobTH to 60%

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.

---------------------------------------end of quote from chapter 5.2.2.3 -------------------------------

End with something like:

There are a couple of parents and kids pioneering this area, see also Case studies 7.x.

We highlighted areas that would require some minimum compliance. In the end it comes down to compare the achieved ease in daily use and achieved %TIR to how it was when hybrid closed looping.

If users pre-bolus for kids …..always, or sometimes? How(much)? =>

Make sure to give **details** …

…. and possibly a **warning**, regarding implications for settings (I could imagine bgAccel_ISF-weight tuning should be factor 5 if not 10 milder in that case = … and not be good if sometimes you pre-bolus a bit, and sometimes not ????)

Re. pre-requisites, CGM, for kids: what works well ??

If not G6 is used: (re CGMs, observe latest developments re 1 minute vs 5 minute readings, see github/ga-zelle pls)

Pls include implications of (from wiki aaps):

Dexcom G6 or Dexcom ONE with xDrip+ Companion Modeℑ

The noise-level data is not shared with AAPS using this method. Therefore ‘Enable SMB always’ and ‘Enable SMB after carbs’ are disabled.

xDrip+ with FreeStyle Libreℑ

None of the FreeStyle Libre systems (FSL1, FSL2, or FSL3) broadcast any information about the level of noise detected in the readings, and therefore ‘Enable SMB always’ ‘Enable SMB after carbs’ are disabled for all setups using the FreeStyle Libre.

In addition, many people have reported the FreeStyle Libre often produces noisy data. In xDrip+ there are a few options to help with this:

Smooth Sensor Noise. In xDrip+ Settings > xDrip+ Display Settings ensure that Smooth Sensor Noise is turned on. This attempts to apply smoothing to noisy data.

Smooth Sensor Noise (Ultrasensitive). If you are still seeing noisy data in xDrip+ you can apply more aggressive smoothing using the Smooth Sensor Noise (Ultrasensitive) setting. This will attempt to apply smoothing even on very low levels of detected noise. To do this, first enable engineering mode in xDrip+. Then navigate to Settings > xDrip+ Display Settings and turn on Smooth Sensor Noise (Ultrasensitive).