

0. Introduction

V. 3.2



Exploring Full Closed Loop potential of-autoISF

Disclaimer – Important to read and understand

Authors are no medical professionals but T1 diabetics (or parents of a T1D child) who report their -limited - understanding and experience, in an effort to contribute to a growing body of knowledge, and to facilitate development of patient centered solutions.

Nothing in this site is medical advice, but meant to stimulate patient-driven self-responsible research, and is meant also to stimulate product developments by the medical industry. Anything you try to conclude for yourself you do on own risk. **This is by no means a medical product but what is offered is a toolset for participating in development.**

Never copy what others report to use, but **investigate and adjust to your data**. Neglecting safety instructions, and just using the “buttons” that are made available in a supposed “learning by doing” mode, would be very dangerous with the early development stage tools this research paper is about.

In case you choose to get deeper involved, **run the system disconnected**, parallel to your current glucose management, to learn its behavior before eventually considering (on own risk) to go any further. Please stay connected and share experiences, too.

Introduction

Full Closed Loop using Automations is represented in AAPS Master and in the related readthedocs since autumn 2023. (<https://androidaps.readthedocs.io/en/latest/Usage/Full-ClosedLoop.html>).

Pre-requisites and the principal function of a Full Closed Loop, *without the user ever giving a bolus and without entering any carb info* are explained, also in a couple of other languages, there (and also in our [section 1.](#))

autoISF is being developed as a much more **sophisticated alternative for FCL, aiming at higher %TIR performance and/or higher degree of daily „freedom“** than simpler approaches to FCL could provide.

However, this demands much higher degree of involvement by the user. **Setting up your FCL is a very serious multi-week project, and it is important that you follow us through the material in the sequence of suggested steps.**

33 With autoISF, and especially with the intention to use it for Full Closed Loop, you are in the early
34 development area. It is therefore important to observe the disclaimer given above, and the warn-
35 ings given in the e-book sections, as well as the hints given by the developers in the respective
36 manuals and readme files on their Github pages:

37

- 38 • For autoISF with **AAPS**, the main ones are <https://github.com/ga-zelle/autoISF/> and
39 <https://github.com/T-o-b-i-a-s/AndroidAPS/tree/3.2.0.4-ai3.0.1>

40 That version was in very minor ways updated in beginning of 2025:

- 41 • AAPS3.2.0.4-ai3.0.1_no-version-check is offered to eliminate the update request to Master
42 AAPS 3.3. (It will take many months to transform all autoISF code for AAPS 3.3; and “negoti-
43 ations” might not lead to acceptance of all features in Master).
- 44 • AAPS3.2.0.4-ai3.0.2 offers a tiny bit more re. info about iobTH in the SMB tab

45 **This e-book is valid for all these versions.**

46 Should you choose to upgrade to AAPS 3.3 (Master or dev) you also get (some or many) autoISF
47 features - but not all. and so this e-book will not fully apply (notably not [sections 3.4](#) and [6.6](#))

48

- 49 • Note there is **no** comparable FCL solution for **iOS Loop** because their algorithm depends
50 very much on carb inputs.

51

- 52 • However, theoref(1) algorithm (UAM+SMB as in AAPS) has been developed also for
53 **i-Phone** based systems:

- 54 ○ on the **Trio** platform <https://discord.gg/Rr37aAzWz9>, “**TAI**” (for Trio + autoISF) dev
55 variant with autoISF see <https://github.com/mountrcg/Trio#autoisf>

- 56 ○ on the **iAPS** platform, with autoISF ported into rapidly evolving early development
57 branches of iAPS: <https://github.com/mountrcg/iAPS/> readme.md.

58 dev_autoISF3.x_newUI is the default branch there for autoISF.

59 Note that setting up an autoISF FCL on i-Phone platform may be especially hard.
60 iAPS/Trio users are disadvantaged vs AAPS because 1) lack of some technical fea-
61 tures (Automations, emulator) 2) there is no e-book variant exactly for their system
62 3) many users did not go through the mandatory “education” via Objectives as in
63 AAPS, and might not have a good “vanilla HCL” starting point (“blueprint for your
64 FCL”, as I call it in [section 4.1](#))

65

66 Unless you are ready to do a lot of reading and loop data analysis to get your FCL running, please
67 stay away. Trial and error won't get you far: Yes; with 18 (!) additional parameters on board, you al-
68 ways can "trick" your loop to get one situation (e.g. pizza) look pretty good. But to find settings that
69 cover also other situations well (say, a salad with chicken) is not easy. It is possible, though, if you
70 "build" your FCL following the suggestions, notably in [sections 2](#) and [4](#).

71

72

73 If you currently can't commit yourself, or lack an important pre-requisite, you might want to con-
74 sider one of the following "easier" options:

- 75 • You could first try the **Full Closed Loop in a simpler form** with Automations (see AAPS
76 readthedocs and [section 13.1](#)):

77 Depending on the quality of their HCL tuning they are starting from, their expectations for
78 %TIR, and on rapid carb contents of their diet, an increasing number of people succeed in
79 making a respectable start the first time they try using AAPS in that much simpler Full
80 Closed Loop mode.

81 See also the first published medical study that included 16 patients using AAPS, who found,
82 on average, comparable %TIR performance when using a basic Full Closed Loop mode:

83 <https://pubmed.ncbi.nlm.nih.gov/36826996/>

84 Note that Trio and iAPS lack Automations, This makes it much more complicated if you try
85 similar implementations via so-called middleware.

- 86 • You could also opt for a **Meal Announcement** method, which is a significant step from
87 HCL *towards* FCL, but still **involves a pre-bolus**.

- 88 ○ For autoISF, this method is sketched in [section 7](#).

- 89 ○ Other early-DEV-variants of AAPS are mentioned in [section 13.3](#), which also un-
90 dergo permanent further development (Boost, AIMI, EatingNow, Tsunami).

- 91 ○ AIMI was also ported into Trio as an option, see [https://github.com/moun-](https://github.com/moun-trcg/Trio#aimi-b30)
92 [trcg/Trio#aimi-b30](https://github.com/moun-trcg/Trio#aimi-b30)

93 Note that all these "Meal Announcement" methods are far less well described than the two FCL
94 options, but they are easier to set up and give some of the benefits you may seek (notably, no
95 carb counting).

96

97

98