

Экспериментальная лаборатория Летающего Кота



FSIOHUB

Справочное руководство

г. Краснодар
2020 г.

ОГЛАВЛЕНИЕ

ИСТОРИЯ ДОКУМЕНТА.....	3
ВВЕДЕНИЕ.....	4
1. УСТРОЙСТВА И ПРОТОКОЛ.....	4
1.1. СПЕЦИФИКАЦИЯ УСТРОЙСТВА.....	4
1.2. ФОРМАТ СОБЫТИЙ ВВОДА.....	4
1.3. ФОРМАТ ВЫВОДА.....	5
1.4. ПРОЦЕДУРА ОБНАРУЖЕНИЯ УСТРОЙСТВА ПРИЛОЖЕНИЕМ.....	5
2. ПРИЛОЖЕНИЕ FSIONUB.....	5
3. КОНФИГУРАЦИОННЫЙ ФАЙЛ.....	7
3.1. СТРУКТУРА КОНФИГУРАЦИОННОГО ФАЙЛА.....	8
3.2. СОСТАВЛЕНИЕ КОНФИГУРАЦИОННОГО ФАЙЛА ПРИ ПОМОЩИ FSIONUBCONFIGHELPER НА ПРИМЕРАХ.....	9
3.2.1 Секция для тумблера Landing Lights.....	9
3.2.2 Секция для кнопки BARO STD.....	11
3.2.3 Секция тумблера выпуска/уборки фар.....	12
3.2.4 Секция энкодера AP VERTICAL SPEED.....	13
3.2.5 Секция энкодера AP HDG.....	15
3.2.6 Секция энкодера AP CRS SET.....	16
3.2.7 Секция потенциометра закрылков.....	17
3.2.8 Секция потенциометра триммера руля высоты.....	18
3.2.9 Секция индикатора AP ALT.....	19
3.2.10 Секция индикатора AP HDG.....	20
3.2.11 Секция индикатора COM1 FREQ.....	21
3.2.12 Секция индикатора ADF FREQ.....	22
3.2.13 Секция индикатора AP MASTER.....	24
3.2.14 Секция лампы подсветки приборов.....	25
4. СПИСОК ЛИТЕРАТУРЫ.....	27

ИСТОРИЯ ДОКУМЕНТА

	Версия	Дата	Автор	Примечание
1	0.1	10.11.20	xflyingcat	Draft

ВВЕДЕНИЕ

FSIOHub представляет собой приложение, которое осуществляет взаимодействие с оборудованием ввода/вывода (устройствами) с симулятором полёта (симулятором).

В качестве симулятора подразумевается один из FS2004, FSX, версии P3D, FS2020.

Приложение присоединяется к симулятору посредством программного продукта FSUIPC, то есть, процедуры интерпроцесса и доступа к памяти симулятора осуществляет FSUIPC.

Приложение использует свой собственный протокол для взаимодействия с устройствами. Нет существенных ограничений на количество устройств, подключенных к приложению. В расчёте только ресурсы операционной системы, здравый смысл и влияние количества устройств на время отклика.

Устройства обнаруживаются автоматически, в конфигурационном файле не нужно прописывать имена последовательных портов.

1. УСТРОЙСТВА И ПРОТОКОЛ

1.1. Спецификация устройства

В простейшем случае могут быть подключены устройства, имеющие следующую спецификацию:

1. Физический интерфейс подключения RS232/RS422 или USB CDC (виртуальный порт).
2. Сигналов в виде событий до 256 (в расширенном варианте до 4096).
3. Устройств дискретного вывода до 64.
4. Устройств символьного вывода до 128 символов.

1.2. Формат событий ввода

	Описание	Элементы протокола				
		Старт	Тип события	Код события	Значение параметра	Стоп
1	Замыкание цепи (тумблер SPDT)	{	8	2 или 3 символа	-	}

2	Размыкание цепи (тумблер SPDT)	{	4	2 или 3 символа	-	}
3	Событие энкодера	{	2	2 или 3 символа	-	}
4	Событие переключателя или нажатия кнопки SPST	{	1	2 или 3 символа	-	}
5	Событие с параметром	{	A	2 или 3 символа	4 символа	}

Примечание 1. Код события и значение параметра имеют шестнадцатеричную кодировку. Соответственно, двухсимвольные коды имеют диапазон 00...FF (всего 256 событий), а трёхсимвольные 000...FFF (всего 4096 событий).

Значение параметра имеет диапазон 0000...FFFF (от нуля до 65535).

Примечание 2. SPST – Single Pole Single Throw – однополюсный ключ или кнопка
SPDT – Single Pole Double Throw – однополюсный переключатель на два положения

1.3. Формат вывода

	Описание	Элементы протокола			
		Старт	Преамбула	Данные	Стоп
1	Лента дискретного вывода	{	LED	64 символа	}
2	Лента символьного вывода	{	7SG	128 символов	}

Примечание 1. В ответ на принятые данные, устройство отвечает символом ! (восклицательным знаком) в качестве АСК. **В любом случае подтверждаются только пакеты, принятые от приложения.**

Примечание 2. Расположение и размер актуальных данных на ленте определяется на этапе разработки устройства и должно быть известно при составлении файла конфигурации приложения. Размеры лент фиксированные, то есть, даже если в устройстве будет один светодиод, приложение будет передавать ленту из 64-х символов.

Примечание 3. С содержимым ленты устройство может поступать по своему усмотрению: отправить на устройство индикации, подвергнув или нет дополнительному форматированию или преобразовать для использования численных значений в качестве управляющих параметров для дополнительных исполнительных устройств (сервомоторы, шаговые двигатели).

1.4. Процедура обнаружения устройства приложением

1. Приложение отправляет в линию связи запрос обнаружения {DET}
2. Устройство отвечает символом ! в качестве АСК
3. Устройство отправляет {Fxx}, где xx - шестнадцатеричный идентификатор устройства.

2. ПРИЛОЖЕНИЕ FSIOHub

Это исполняемый файл `fsiohub.exe`. Для его нормальной работы требуется размещение файла `fsiohub.ini` в той же самой директории.

На рисунке ниже показан процесс нормального старта приложения.

FSIOHUB 0.7.0

Config file loading

```
Total config items: 106
Total input items: 0
Total output items: 35
Config errors: 0
```

Config loaded with no errors

USB CDC I/O Hub for FSUIPC/XPUIOC rev.0.7.0

Getting list of serial ports

Подпапка или файл C:\Users\alex\AppData\Local\Temp\fsiohub уже существует.

Patterns for USB port finding:

- ```
=====
1. USB\UID
2. FTDIBUS\UID
3. MOSCHIP\COMPOUND_SERIAL_INTERFACE
4. Electronic Team - USBC
```

Serial ports list:

- ```
=====
* 1. COM16 - Arduino LLC (www.arduino.cc) - USB\VID_2341&PID_0042\754393332353514152F1
2. COM4 - MosChip Semiconductor Technology Ltd - MF\PCI#VEN_9710&DEV_9835&SUBS_YS_001210000&REV_01\5&9BE9244&0&28A4#CHILD0000
3. COM3 - MosChip Semiconductor Technology Ltd - MF\PCI#VEN_9710&DEV_9835&SUBS_YS_001210000&REV_01\5&9BE9244&0&28A4#CHILD0001
* 4. COM6 - IAR Systems - USB\VID_FFFF&PID_0037\000000001
* 5. COM31->COM32 - Electronic Team - USBC9\DEVICES\0000
* 6. COM32->COM31 - Electronic Team - USBC9\DEVICES\0001
7. COM1 - (Стандартные порты) - ACPI\PNP0501\1
```

Serial port	Available	Device detected	Device ID
COM16	yes	yes	0
COM6	yes	yes	4
COM31->COM32	yes	no	-1
COM32->COM31	yes	no	-1

Waiting for FSUIPC

Thread for device:0 serial port: COM16 started

Thread for device:4 serial port: COM6 started

Request for time ok: FS clock = 11:49:22

Sim is P3D, FSUIPC Version = 4.923

Link established to FSUIPC

```
event_type:008 port_id:04 event_id:0037(0055) value:0
event_type:004 port_id:04 event_id:0037(0055) value:0
event_type:002 port_id:04 event_id:0080(0128) value:0
event_type:002 port_id:04 event_id:0080(0128) value:0
event_type:002 port_id:04 event_id:0080(0128) value:0
event_type:002 port_id:04 event_id:0081(0129) value:0
event_type:008 port_id:04 event_id:005C(0092) value:0
event_type:004 port_id:04 event_id:005C(0092) value:0
event_type:008 port_id:04 event_id:0011(0017) value:0
event_type:004 port_id:04 event_id:0011(0017) value:0
```

Сначала приложение загружает конфигурационный файл, затем сканирует последовательные порты, определяет кандидатов на дальнейшее сканирование по сигнатурам (образцам поиска, на картинке это список Patterns for USB port finding).

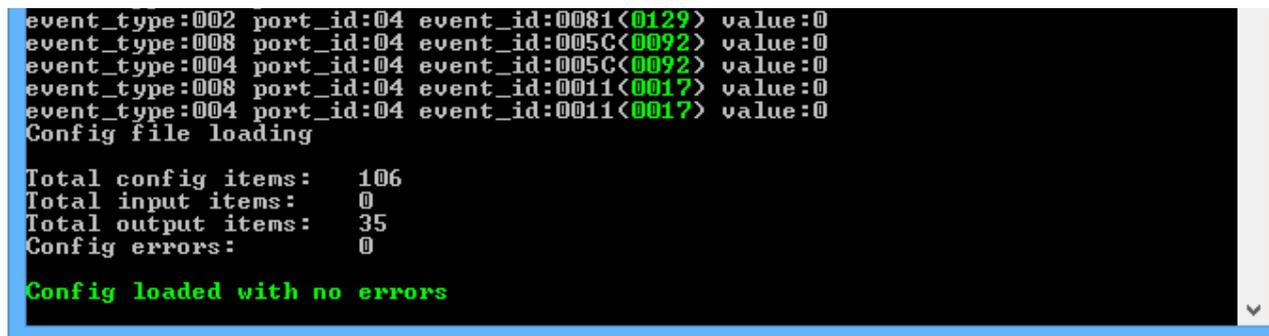
Затем приложение определяет доступность каждого порта на открытие и определяет присоединенное устройство, отвечающее по протоколу FSIOHUB на запрос {DET}

В итоге строится таблица доступных портов с указанием ID присоединенных устройств.

В процессе поступления событий выводится строка описания. Код события в десятичном представлении выделен зелёным цветом. Именно он фигурирует в имени секции описания в конфигурационном файле.

В любое время работы приложения можно перезагрузить конфигурационный файл, нажав на клавиатуре клавишу L при активном окне приложения.

На рисунке ниже состояние после нажатия клавиши L.



```
event_type:002 port_id:04 event_id:0081<0129> value:0
event_type:008 port_id:04 event_id:005C<0092> value:0
event_type:004 port_id:04 event_id:005C<0092> value:0
event_type:008 port_id:04 event_id:0011<0017> value:0
event_type:004 port_id:04 event_id:0011<0017> value:0
Config file loading

Total config items: 106
Total input items: 0
Total output items: 35
Config errors: 0

Config loaded with no errors
```

В том случае, если в конфигурационный файл добавлены описания для новых устройств (добавлены секции [device]), необходимо будет перезагрузить приложение.

3. КОНФИГУРАЦИОННЫЙ ФАЙЛ

Органы управления наделены смыслом, когда расположены в определённом месте и маркированы для удобства пользования. Устройство ввода преобразовывает действия с этими органами в коды и отправляет по протоколу в приложение, которое, исходя из принципов универсализма и формального подхода, должно отправить формально закодированные данные в правильные ячейки памяти симулятора. То есть, приложение

является формальным автоматом и само нуждается в программировании. В данном случае языком программирования является язык конфигурационного файла. Он скучный, невнятный, интуитивно непонятный, иными словами, не достоин отдельного изучения.

Для начала только о структуре файла, без этого никак.

3.1. Структура конфигурационного файла

Конфигурационный файл является единственным для всех устройств и имеет структуру:

```
[device]
port: 0

# описания для устройства, имеющего ID 0

[device]
port: 1

# описания для устройства, имеющего ID 1

[device]
port: 2

# описания для устройства, имеющего ID 2
```

В начале файла можно вставить опцию включения журналирования

```
[mode]
log=on
```

Конфигурационный файл fsiohub.ini должен располагаться в той же самой директории, что и исполняемый файл fsiohub.exe

3.2. Составление конфигурационного файла при помощи FSIOHubConfigHelper на примерах.

Во всех примерах подразумевается, что устройство запитано и подключено, FSIOHubConfigHelper.exe запущен, а пользователь успел освоиться с открытием нужного последовательного порта и видит отклик устройства в виде его идентификатора.

3.2.1 Секция для тумблера Landing Lights

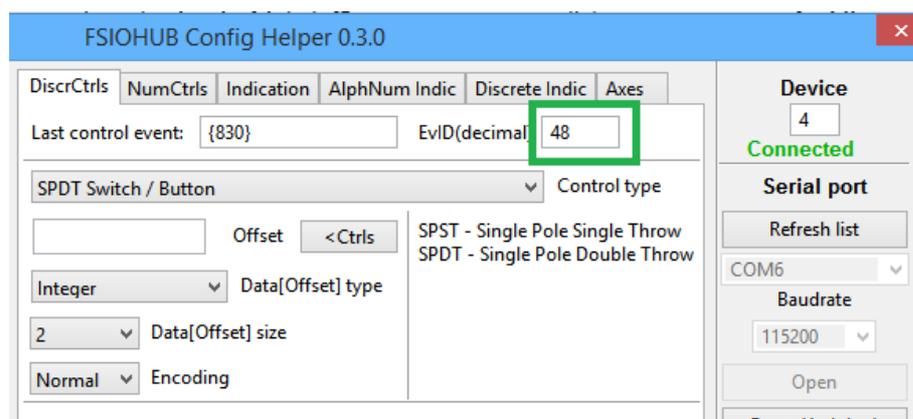
1. Исходные данные [1]

0D0C	2	Lights, a switch for each one (bits from lo to hi):
		0 Navigation
		1 Beacon
		2 Landing
		3 Taxi
		4 Strobes
		5 Instruments
		6 Recognition
		7 Wing
		8 Logo
		9 Cabin

2. Перевести тумблер в исходное положение, если это не так.

3. Включить тумблер Landing Lights.

4. Наблюдать появление кода события в окошке.



5. Выполнить последовательность действий, показанных на рисунке ниже

DiscrCtrls NumCtrls Indication AlphNum Indic Discrete Indic Axes

Last control event: {830} 1 EvID(decimal) 48

SPDT Switch / Button Control type

ODOC 2 Offset <Ctrls

Integer 3 Data[Offset] type

2 4 Data[Offset] size

Normal 5 Encoding

0: 0 6 Min:

1: 1 7 Max:

Mask: 4 Increment:

Cycling options

Loop (e.g. HDG, CRS)

Limits (e.g. IAS, ALT)

Add control section 11

Mask calculator

8

All clear

All set

Invert

9

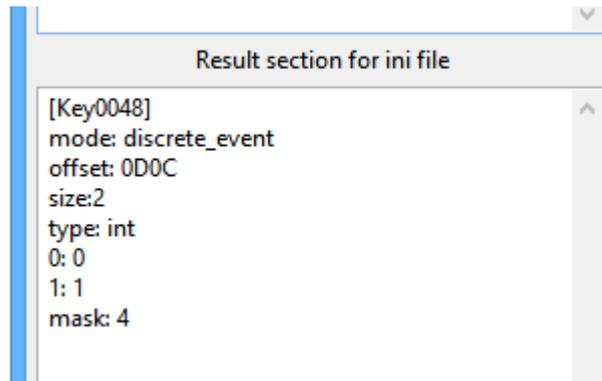
10

> Mask

< Mask

31	30	29	28	27	26	25	24
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
23	22	21	20	19	18	17	16
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
15	14	13	12	11	10	9	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
7	6	5	4	3	2	1	0
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

6. Наблюдать появление секции



3.2.2 Секция для кнопки BARO STD

1. Исходные данные представляют собой адрес универсальной ячейки памяти, через которую можно подавать простые команды и команды с параметрами [1].

3110	8	<p>Operates a facility to send any 'controls' to Flight simulator. This works with <i>all</i> versions of FS & CFS. Write all 8 bytes for controls which use a value (axes and all <code>_SET</code> controls), but just 4 will do for 'button' types.</p> <p>This is really two 32-bit integers. The first contains the Control number (normally 65536 upwards), as seen in my FS Controls lists. The second integer is used for the parameter, such as the scaled axis value, where this is appropriate. Always write all 8 bytes in one IPC block if a parameter is used, as FSUIPC will fire the control when you write to 3110.</p>
------	---	--

И код команды из списка Controls [2].

PRESSURIZATION PRESSURE DUMP SWITCH SET	66842
PRESSURIZATION RAM SWITCH	66843
PRESSURIZATION RAM SWITCH SET	66844
BAROMETRIC STD PRESSURE	66846
MP PAUSE SESSION	66847
VIEW IG VIEWGROUP SET	66848
VIEW CAMERA SELECT 1	66851
VIEW CAMERA SELECT 2	66852

2. Нажать на кнопку STD BARO

3. Наблюдать код события

AlphNum Indic	Discrete Indic	Axes
	EvID(decimal) <input type="text" value="88"/>	
Control type		

4. Выполнить действия по рисунку ниже

5. Наблюдать появление секции в окне **Result section for ini file**

6. Перенести результат в конфигурационный файл.

FSIOHUB Config Helper 0.3.0

Remarks: BAROMETRIC STD PRESSURE 66846

Result section for ini file:

```
#####
# BAROMETRIC STD PRESSURE 66846
#####
[Key0088]
mode: discrete_event
offset: 3110
size: 8
type: int
1: 66846
```

Configuration Panel:

- Last control event: {458}
- EvID(decimal): 88
- Control type: SPST Button, Switch / Encoder dir pulse/ Rotary switch pc
- Offset: 3110
- Data[Offset] type: <Ctrls
- Data[Offset] size: 8
- Encoding: Normal
- 0: [] Min: []
- 1: 66846 Max: []
- Mask: [] Increment: []
- Cycling options:
 - Loop (e.g. HDG, CRS)
 - Limits (e.g. IAS, ALT)
- Add control section

3.2.3 Секция тумблера выпуска/уборки фар

1. Исходные данные [1]

3110	8	<p>Operates a facility to send any 'controls' to Flight simulator. This works with <i>all</i> versions of FS & CFS. Write all 8 bytes for controls which use a value (axes and all <code>_SET</code> controls), but just 4 will do for 'button' types.</p> <p>This is really two 32-bit integers. The first contains the Control number (normally 65536 upwards), as seen in my FS Controls lists. The second integer is used for the parameter, such as the scaled axis value, where this is appropriate. Always write all 8 bytes in one IPC block if a parameter is used, as FSUIPC will fire the control when you write to 3110.</p>
------	---	--

И, собственно коды управления [2].

LANDING LIGHT UP	65862
LANDING LIGHT DOWN	65863
LANDING LIGHT LEFT	65864
LANDING LIGHT RIGHT	65865
LANDING LIGHT HOME	65866

2. Перевести тумблер выпуска/уборки фар в оппозитное положение
3. Наблюдать код события или ввести, если заранее известно
4. Выполнить последовательность действий по рисунку ниже

5. Наблюдать появление секции в окне **Result section for ini file**

3.2.4 Секция энкодера AP VERTICAL SPEED

1. Исходные данные [1]

07EC	4	Autopilot vertical speed hold
07F2	2	Autopilot vertical speed value, as ft/min
07F4	4	Autopilot RPM (NI) hold

2. Определить или ввести код вращения энкодера по часовой стрелке

3. Выполнить действия по рисунку ниже

FSIOHUB Config Helper 0.3.0

DiscrCtrls NumCtrls Indication AlphNum Indic Discrete Indic Axes

Last control event: {244} EvID(decimal) 68 1

Encoder 2 Control type

07F2 3 Offset <Ctrls SPST - Single Pole Single Throw
SPDT - Single Pole Double Throw

Integer Data[Offset] type 4

2 Data[Offset] size 5

Normal Encoding 6

0: Min: -9900 7

1: Max: 9900 8

Mask: Increment: 100 9

Cycling options

Loop (e.g. HDG, CRS)

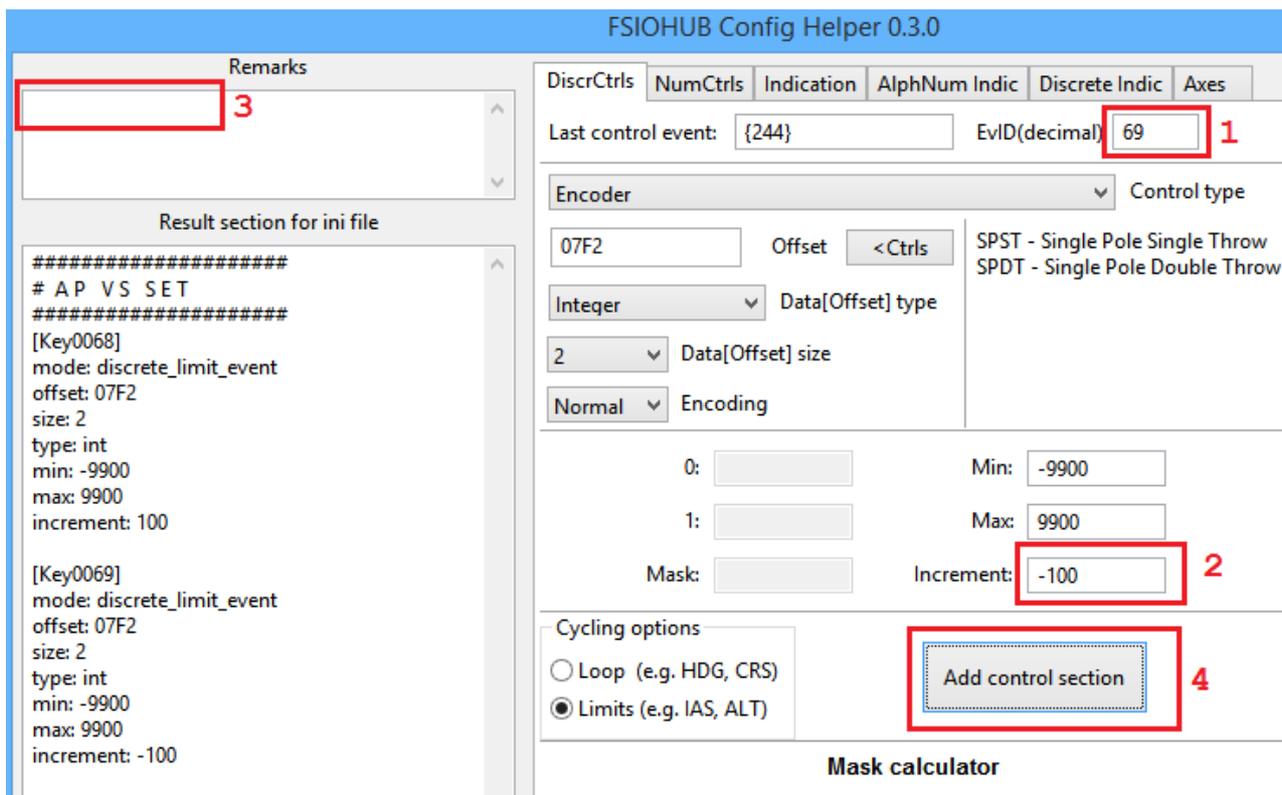
Limits (e.g. IAS, ALT) 10

Add control section 12

4. Сменить направление вращения для получения нового кода события

5. Выполнить действия по рисунку ниже. Красным цветом помечены изменённые

поля.



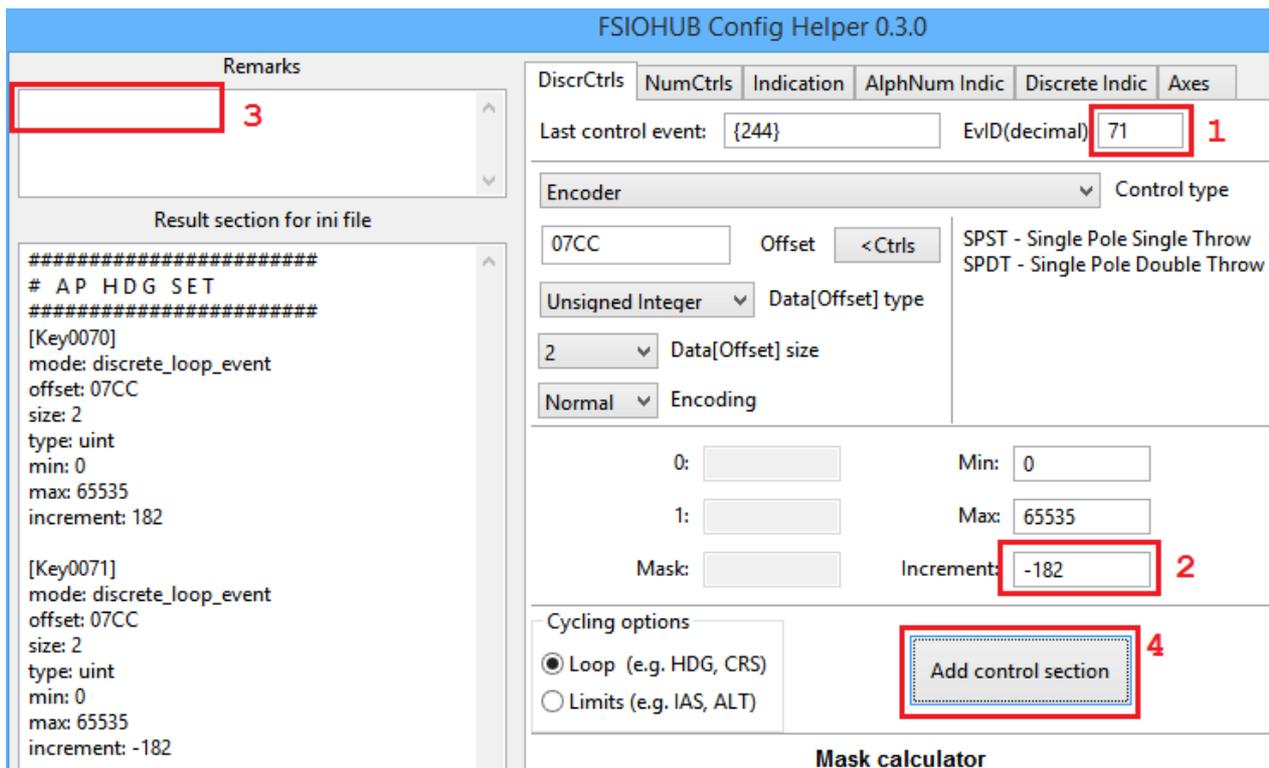
6. Наблюдать появление новой секции, соответствующей противоположному направлению вращения.

3.2.5 Секция энкодера AP HDG

1. Исходные данные [1]

07C4	4	Autopilot NAV1 lock
07C8	4	Autopilot heading lock
07CC	2	Autopilot heading value, as degrees*65536/360
07D0	4	Autopilot altitude lock
07D4	4	Autopilot altitude value, as metres*65536

2. Выполнить действия по рисунку ниже.



3.2.6 Секция энкодера AP CRS SET

1. Исходные данные [1]

3110	8	<p>Operates a facility to send any 'controls' to Flight simulator. This works with <i>all</i> versions of FS & CFS. Write all 8 bytes for controls which use a value (axes and all _SET controls), but just 4 will do for 'button' types.</p> <p>This is really two 32-bit integers. The first contains the Control number (normally 65536 upwards), as seen in my FS Controls lists. The second integer is used for the parameter, such as the scaled axis value, where this is appropriate. Always write all 8 bytes in one IPC block if a parameter is used, as FSUIPC will fire the control when you write to 3110.</p>
------	---	---

И коды управления для данного случая [2].

CLOCK MINUTES INC	65660
CLOCK SECONDS ZERO	65661
VOR1 OBI DEC	65662
VOR1 OBI INC	65663
VOR2 OBI DEC	65664
VOR2 OBI INC	65665

2. Выполнить вращение энкодера по часовой стрелке.
3. Выполнить действия по рисунку ниже.

FSIOHUB Config Helper 0.3.0

Remarks

VOR1 OBI INC 65663

Result section for ini file

```
#####
# VOR1 OBI INC 65663
#####
[Key0072]
mode: discrete_event
offset: 3110
size: 8
type: int
1: 65663
```

DiscCtrls NumCtrls Indication AlphNum Indic Discrete Indic Axes

Last control event: {244} EvID(decimal) 72

SPST Button, Switch / Encoder dir pulse/ Rotary switch pc Control type

3110 Offset <Ctrls SPST - Single Pole Single Throw
SPDT - Single Pole Double Throw

Integer Data[Offset] type

8 Data[Offset] size

Normal Encoding

0: Min:

1: 65663 Max:

Mask: Increment:

Cycling options

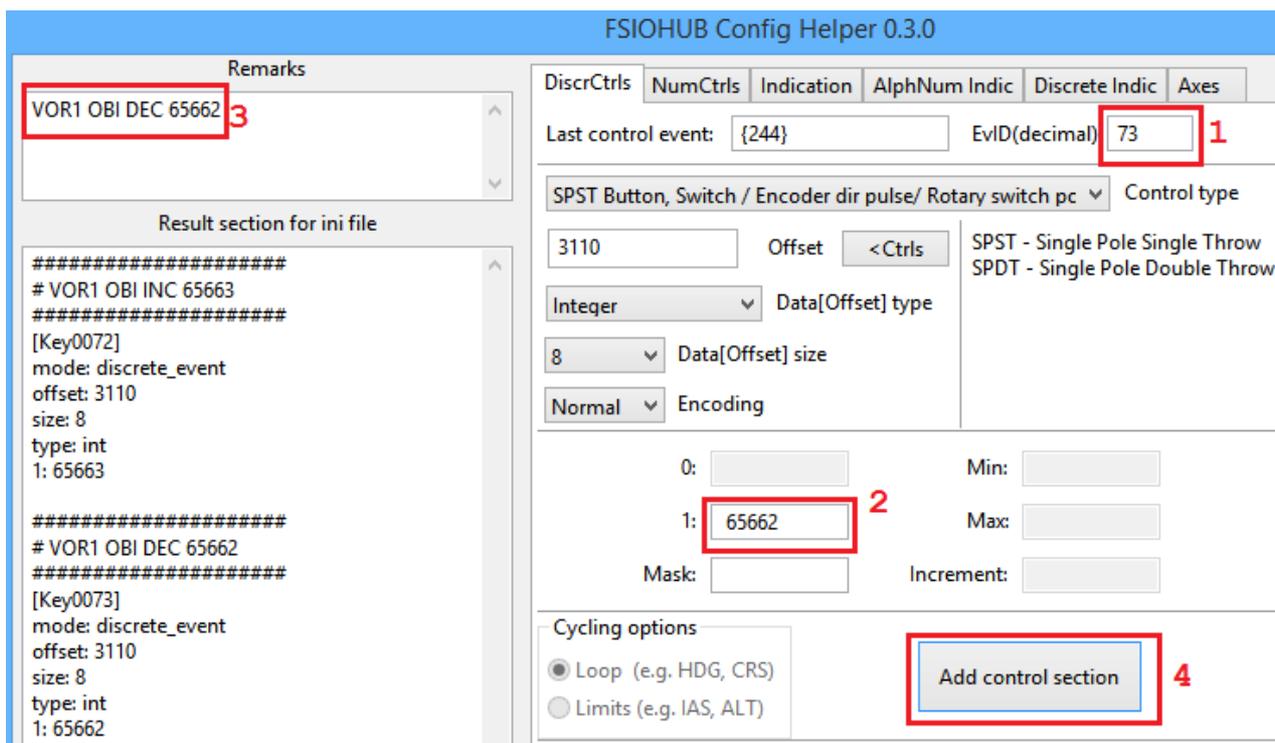
Loop (e.g. HDG, CRS)

Limits (e.g. IAS, ALT)

Add control section

4. Выполнить вращение энкодера в противоположном направлении.
5. Выполнить действия по рисунку ниже. Красным цветом обозначены изменённые поля.

поля.



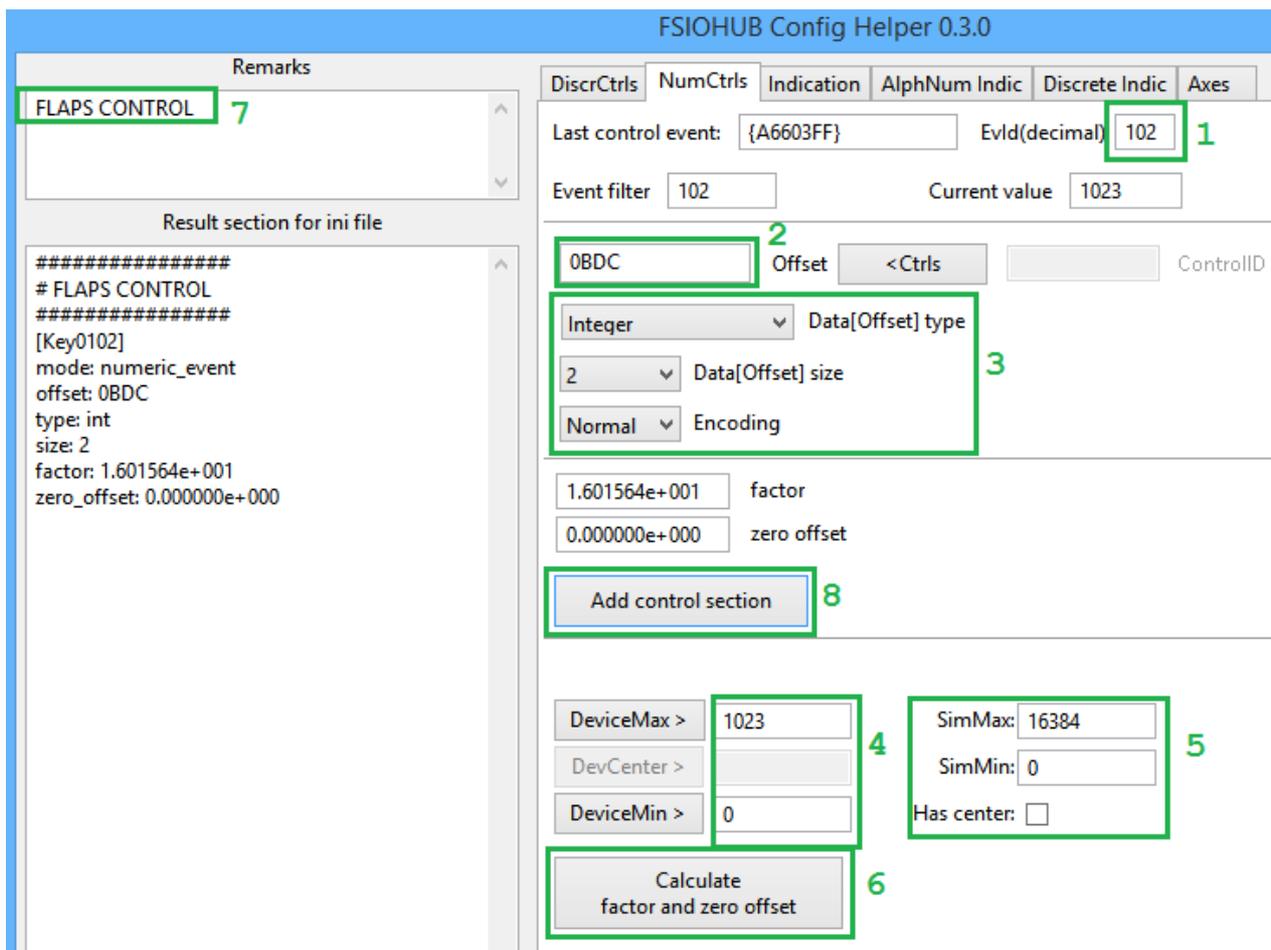
3.2.7 Секция потенциометра закрылков

1. Исходные данные [1].

0BDC	4	Flaps control, 0=up, 16383=full. The “notches” for different aircraft are spaced equally across this range: calculate the increment by $16383/(\text{number of positions}-1)$, ignoring fractions. See also offset 3BFA below.
------	---	---

2. Выполнить вращение оси потенциометра для определения значений АЦП в крайних положениях.

3. Выполнить действия по рисунку. Иметь в виду, что пункт 4 рисунка может быть выполнен путём считывания значений АЦП при помощи соответствующей кнопки в соответствующее поле. Иметь в виду, что калькулятор коэффициента и смещения может выдавать результат в так называемом «научном виде» (его ещё называют приведённым и экспоненциальным). Гуглить это дело самостоятельно или не обращать внимания. При расчёте коэффициентов самостоятельно можно использовать обычный вид представления вещественных чисел.



4. Наблюдать результат действий.

3.2.8 Секция потенциометра триммера руля высоты

1. Исходные данные [1] – адрес ворот в пещеру Али-Бабы

3110	8	<p>Operates a facility to send any 'controls' to Flight simulator. This works with <i>all</i> versions of FS & CFS. Write all 8 bytes for controls which use a value (axes and all _SET controls), but just 4 will do for 'button' types.</p> <p>This is really two 32-bit integers. The first contains the Control number (normally 65536 upwards), as seen in my FS Controls lists. The second integer is used for the parameter, such as the scaled axis value, where this is appropriate. Always write all 8 bytes in one IPC block if a parameter is used, as FSUIPC will fire the control when you write to 3110.</p>
------	---	---

и код управления 65766. Этот код предполагает, что по следующему адресу 3114 будет записано значение АЦП, которое можно масштабировать при помощи коэффициента

и смещения. Ходят слухи, что если Control имеет в своем имени слово SET, то всегда по адресу 3114 нужно записывать значение, но это не точно.

Код события с параметром [2].

AXIS AILERONS SET	65763
AXIS RUDDER SET	65764
AXIS THROTTLE SET	65765
AXIS ELEV TRIM SET	65766
PROP PITCH SET	65767
PROP PITCH LO	65768

2. Выполнить действия по рисунку

The screenshot shows the FSIOHUB Config Helper 0.3.0 interface. On the left, the 'Remarks' section contains a list of events, with 'ELEVATOR TRIM CONTROL' and 'AXIS ELEV TRIM SET 65766' highlighted in green and labeled '7'. Below it, the 'Result section for ini file' shows the generated configuration text, including the event name, mode, offset (3110), type (int), size (8), control (65766), factor (3.203128e+001), and zero_offset (5.110000e+002). The main configuration area on the right has several fields and buttons highlighted in green and labeled with numbers 1 through 8: 1. 'Evid(decimal)' set to 103; 2. 'Event filter' set to 103; 3. '<Ctrls' button and 'ControllID' set to 65766; 4. 'DeviceMax' set to 1023; 5. 'SimMax' (16384) and 'SimMin' (-16384) fields with 'Has center' checked; 6. 'Calculate factor and zero offset' button; 7. 'Add control section' button.

3. Наблюдать появление новой секции.

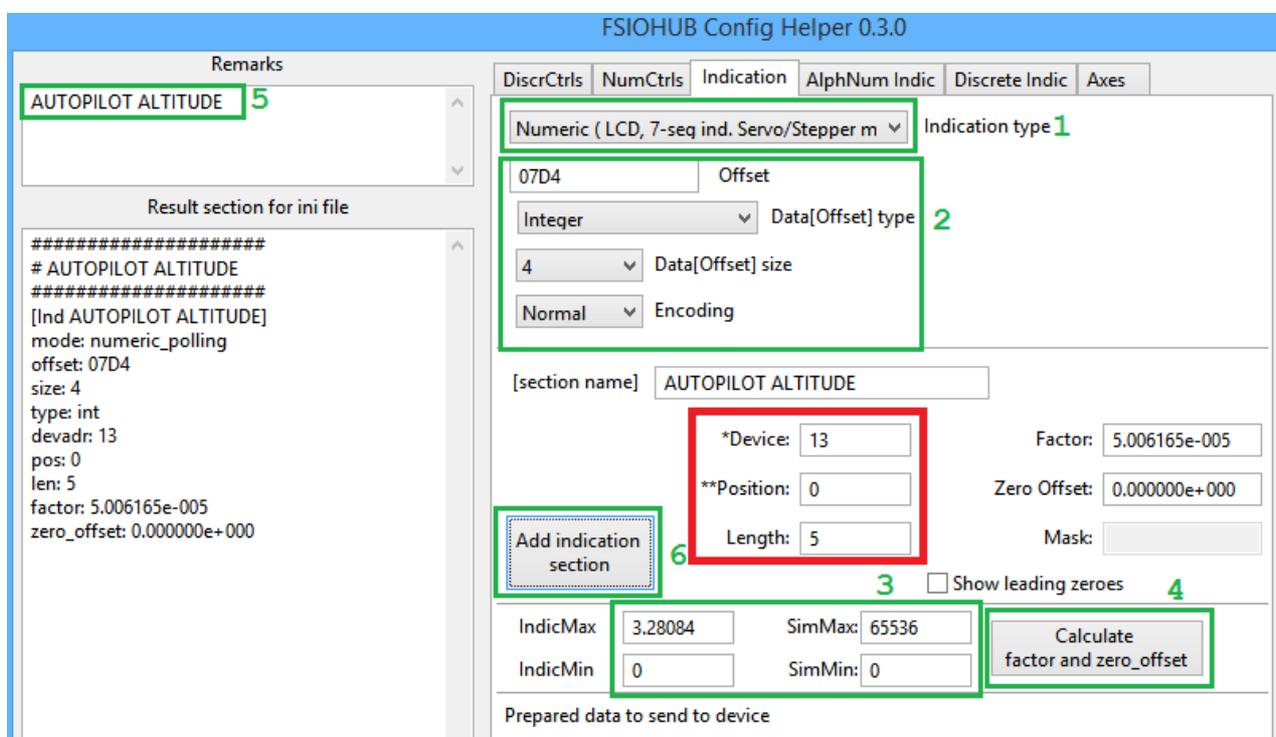
3.2.9 Секция индикатора AP ALT

1. Исходные данные [1].

Нужны бесподобные преобразования, чтобы получить нужный результат, имея в виду, что один дискрет высоты имеет значение 1/65536 метра. Если нужны только метры, можно обойтись вычитыванием двух байт по смещению 07D6, а если таки футы, то масштабировать всё 32-битное число.

07CC	2	Autopilot heading value, as degrees*65536/360
07D0	4	Autopilot altitude lock
07D4	4	Autopilot altitude value, as metres*65536
07D8	4	Autopilot attitude hold
07DC	4	Autopilot airspeed hold

2. Выполнить действия по рисунку.



2. Значения для заполнения полей в красном прямоугольнике, нужно получить у разработчика устройства, каким угодно способом, да хоть из той же документации на изделие.

3. Наблюдать результат формирования конфигурационной секции.

В п.3 рисунка калькулятор использован не совсем обычно, по сути, это просто расчёт соотношения между количеством футов в метре и количеством дискрет в метре, что даст

и простое деление 3.28084/65536. За то, не пришлось доставать другой калькулятор.

3.2.10 Секция индикатора AP HDG

1. Исходные данные [1].

07C4	4	Autopilot NAV1 lock
07C8	4	Autopilot heading lock
07CC	2	Autopilot heading value, as degrees*65536/360
07D0	4	Autopilot altitude lock
07D4	4	Autopilot altitude value, as metres*65536
07D8	4	Autopilot attitude hold
07DC	4	Autopilot airspeed hold

2. Выполнить действия по рисунку

3. Наблюдать результат

3.2.11 Секция индикатора COM1 FREQ

1. Исходные данные [1].

Сразу видно, что в частоте имеется недостача целых 100 мегагерц, что в коде представления составляет 0x10000, а в десятичном виде 65536. Это значение будет необходимо добавить с использованием поля zero_offset

034C	2	ADF1 Frequency: main 3 digits, in Binary Coded Decimal. See also offset 0356. A frequency of 1234.5 will have 0x0234 here and 0x0105 in offset 0356.
034E	2	COM1 frequency, 4 digits in BCD format. A frequency of 123.45 is represented by 0x2345. The leading 1 is assumed.
0350	2	NAV1 frequency, 4 digits in BCD format. A frequency of 113.45 is represented by 0x1345. The leading 1 is assumed.

2. Выполнить действия по рисунку

3. Наблюдать результат. Обратите внимание, что выбирается двоично-десятичная кодировка (она же BCD, она же двоично-десятичная упакованная), как указано в описании оффсета.

3.2.12 Секция индикатора ADF FREQ

1. Исходные данные [1]. Интересный случай, значение частоты собирается из двух датарефов 034C: сотни, десятки и единицы килогерц

034C	2	ADF1 Frequency: main 3 digits, in Binary Coded Decimal. See also offset 0356. A frequency of 1234.5 will have 0x0234 here and 0x0105 in offset 0356.
------	---	--

и 356: тысячи килогерц и десятые доли килогерц,

0356	2	Extended ADF1 frequency. The high byte contains the 1000's digit and the low byte the fraction, so, for a frequency of 1234.5 this offset will contain 0x0105.
------	---	--

причём, десятые доли килогерц на практике не актуальны, поэтому будем трактовать этот офсет несколько иначе
 смещение: 0357
 размер: 1
 содержимое: единицы мегагерц

Задача: наложить содержимое этих офсетов на ленту символьного вывода в необходимой последовательности.

Например, устройство ожидает индикацию частоты ADF со следующими параметрами
 dev:8
 pos:0
 len:4

это можно разбить на два вывода
 dev:8
 pos:0
 len:1

и
 dev:8
 pos:1
 len:3

2. Выполнить настройку вывода для сотен, десятков и единиц по рисунку

Remarks

ADF FREQUENCY 5

Result section for ini file

```
#####
# ADF FREQUENCY
#####

[Ind ADF1 FREQ 100s 10s 1s]
mode: numeric_polling
offset: 034C
size: 2
type: int
devadr: 8
pos: 1
len: 3
encoding: bcd
opt: 1
```

FSIOHUB Config Helper 0.3.0

DiscrCtrls NumCtrls Indication AlphNum Indic Discrete Indic Axes

Numeric (LCD, 7-seg ind. Servo/Stepper m) Indication type 1

034C Offset

Integer Data[Offset] type 2

2 Data[Offset] size

BCD Encoding

[section name] ADF1 FREQ 100s 10s 1s 3

*Device: 8 Factor:

**Position: 1 Zero Offset:

Length: 3 Mask:

6 Add indication section

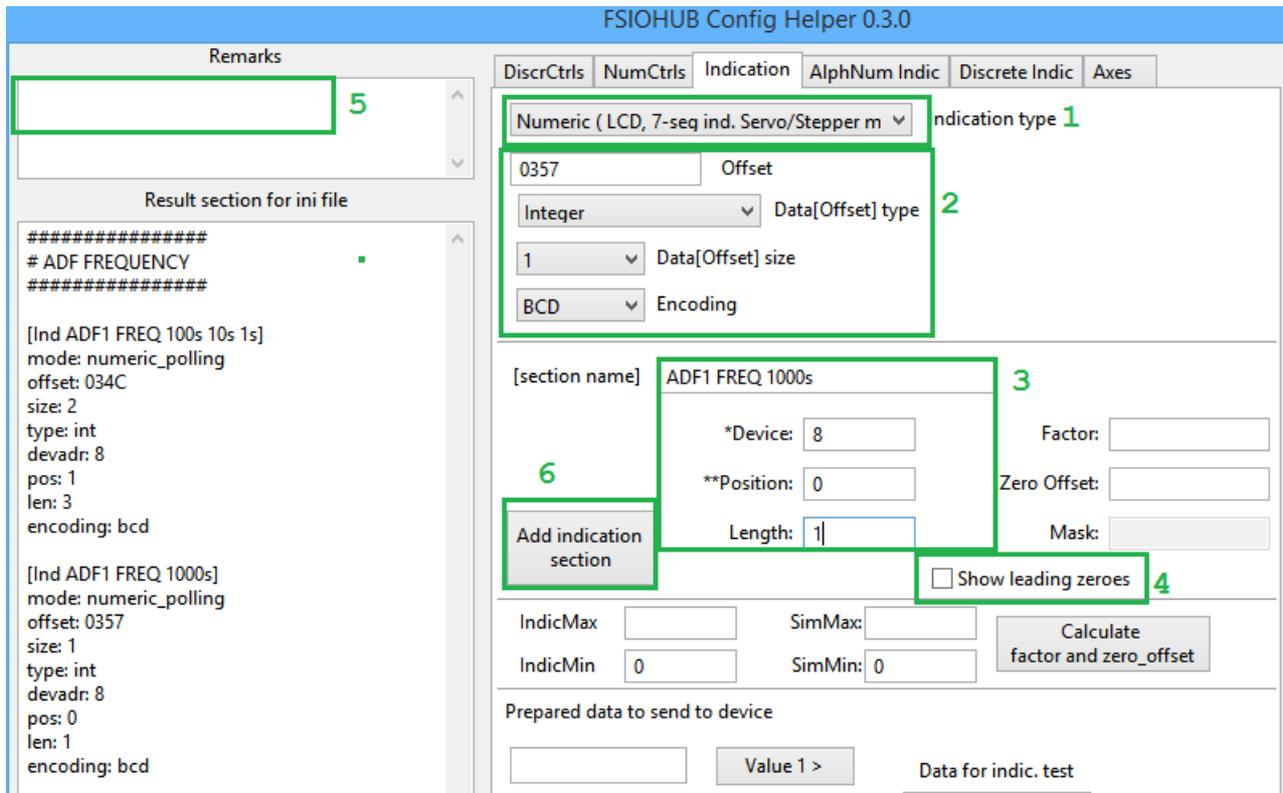
Show leading zeroes 4

IndicMax SimMax:

IndicMin 0 SimMin: 0

Calculate factor and zero_offset

3. Выполнить настройку вывода для единиц мегагерц по рисунку



4. Наблюдать результат, анализировать

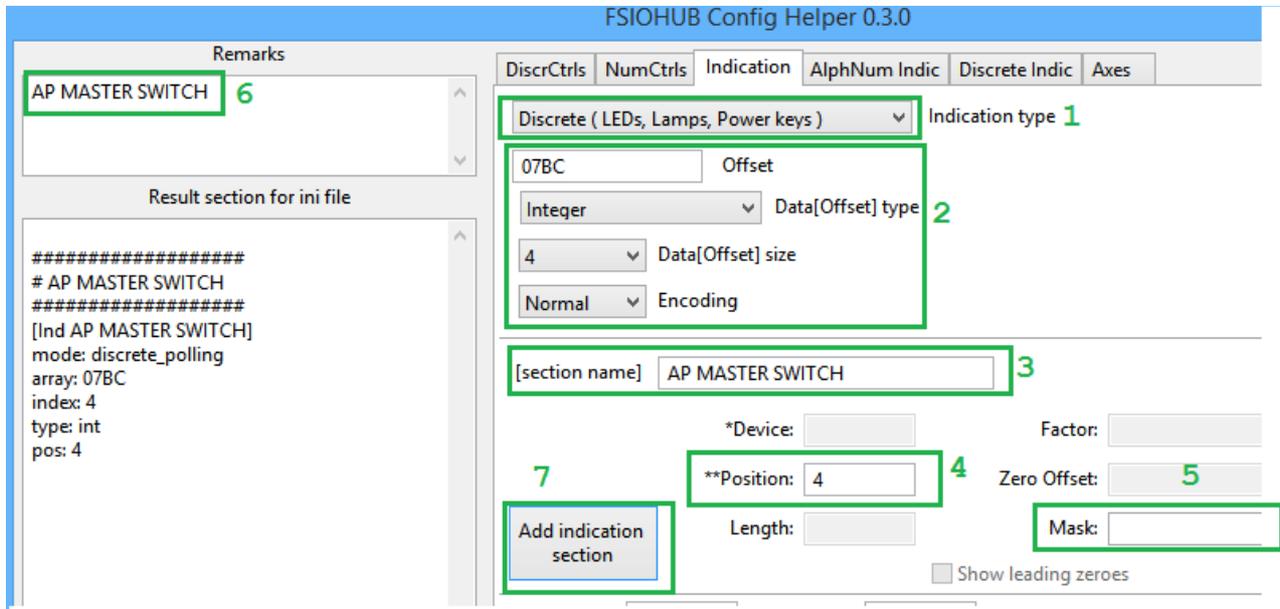
3.2.13 Секция индикатора AP MASTER

1. Исходные данные [1].

07BB	1	Fly by wire SEC computer failed flag
07BC	4	Autopilot Master switch
07C0	4	Autopilot wing leveller

Интересный момент, для хранения одного бита используется целое 32-битное число

2. Выполнить действия по рисунку



Данные для заполнения поля в действии 4 получить у разработчика устройства.

3. Наблюдать результат и анализировать.

3.2.14 Секция лампы подсветки приборов

1. Исходные данные [1].

0D0C	2	Lights, a switch for each one (bits from lo to hi):
		0 Navigation
		1 Beacon
		2 Landing
		3 Taxi
		4 Strobes
		5 Instruments
		6 Recognition
		7 Wing
		8 Logo
		9 Cabin

По данному оффсету находится 16 битное значение. Необходимо определить состояние только одного бита для этого на число нужно наложить маску, которую можно рассчитать на закладке дискретного ввода.

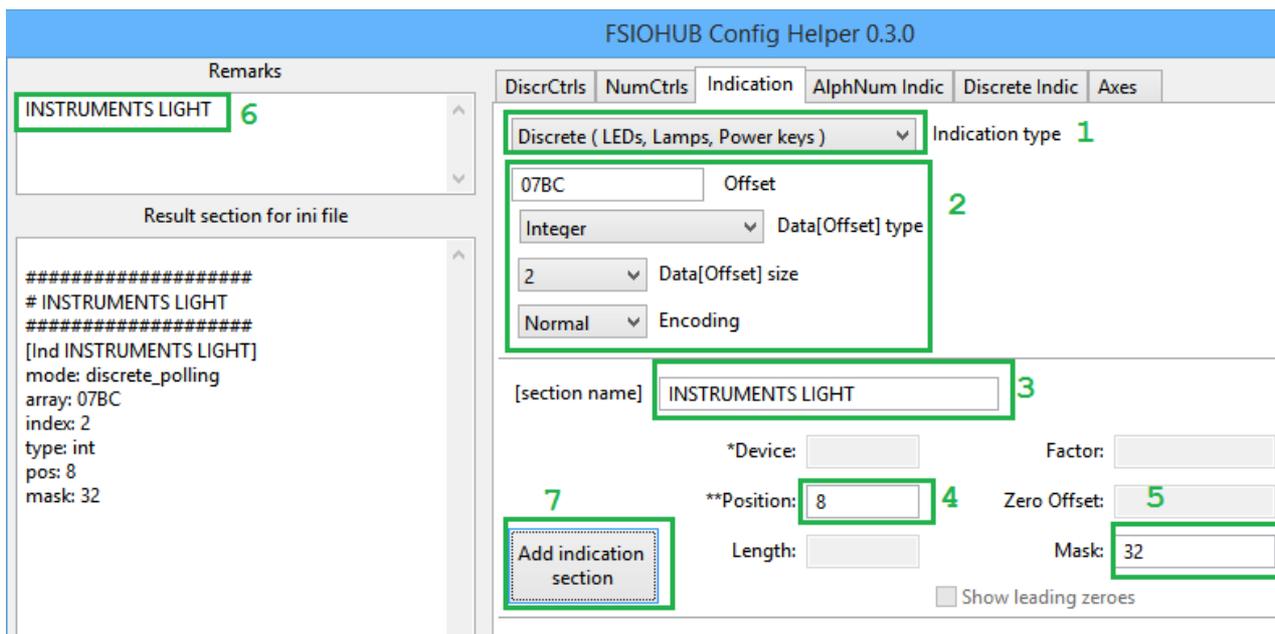
2. Рассчитать маску на закладке дискретного ввода и перенести на закладку вывода.

The screenshot shows a software interface with the following elements:

- DiscrCtrls** tab selected.
- Last control event:
- EvID(decimal):
- SPDT Switch / Button: Con
- Offset: <Ctrls
- SPST - Single Pole Si
- SPDT - Single Pole D
- Data[Offset] type: Integer
- Data[Offset] size: 2
- Encoding: Normal
- 0: Min:
- 1: Max:
- Mask: Increment:
- Cycling options:
 - Loop (e.g. HDG, CRS)
 - Limits (e.g. IAS, ALT)
- Add control section
- Mask calculator**
 - Buttons: All clear (1), All set, Invert
 - Mask grid (bits 31-0):

31	30	29	28	27	26	25	24
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23	22	21	20	19	18	17	16
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	14	13	12	11	10	9	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	6	5	4	3	2	1	0
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
 - Buttons: > Mask (3), < Mask

3. Выполнить действия по рисунку.



4. Наблюдать результат, анализировать, перенести в конфигурационный файл.

4. СПИСОК ЛИТЕРАТУРЫ

1. FSUIPC4 Offsets Status.pdf by Pete Dowson
2. The 2016 List of FSX and P3D Controls.pdf by Pete Dowson