**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Операционные системы»

**Курсовая работа**

Тема: Разработка lua скриптов для аппаратуры frsky taranis x-lite s/pro, работающая на операционной системе “OpenTx”.

Студент: Волков

Группа: 80-207

Преподаватель: Миронов Е. С.

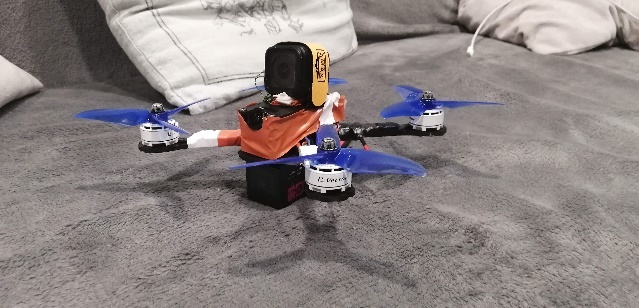
Дата:

Оценка:

1. **Постановка задачи**

Имеется квадрокоптер собственной сборки, работающий на ОС betaflight. Также имеется аппаратура frsky taranis x-lite s. Нужно разработать скрипты на языке lua, совместимые с системными вызовами OpenTX, которые способны:

* показывать телеметрию
* предупреждать о низком заряде батареи на квадрокоптере
* показывать заряд квадрокоптера
* изменять параметры квадрокоптера через совместимые скрипты betaflight

 **

Квадрокоптер. Изолента вокруг корпуса нужна, чтобы защитить электронику от снега.

Аппаратура.

Ап

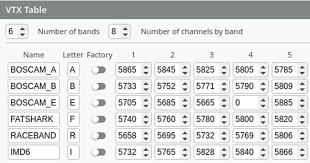
1. **Описание терминов и зачем скрипты нужны**

**Betaflight** – название прошивки полетного контроллера квадрокоптера. При помощи нее идет управление всем аппаратом. Betaflight можно назвать операционной системой квадрокоптера. Прошивка для полетного контроллера написана на си, так как все полетные контроллеры работают на stm32.

**OpenTX** – прошивка для аппаратуры управления. Полностью написана на си и с++ по той же причине, что и прошивка betaflight.

**Smart audio –** порт на видеопередатчике, при помощи которого можно изменять его параметры.

**Smart port –** специальный пин, при помощи которого идет связь между полетным контроллером (далее ПК) и приемником.

**VTX таблица –** это таблица соответствия определенного канала с частотой. 

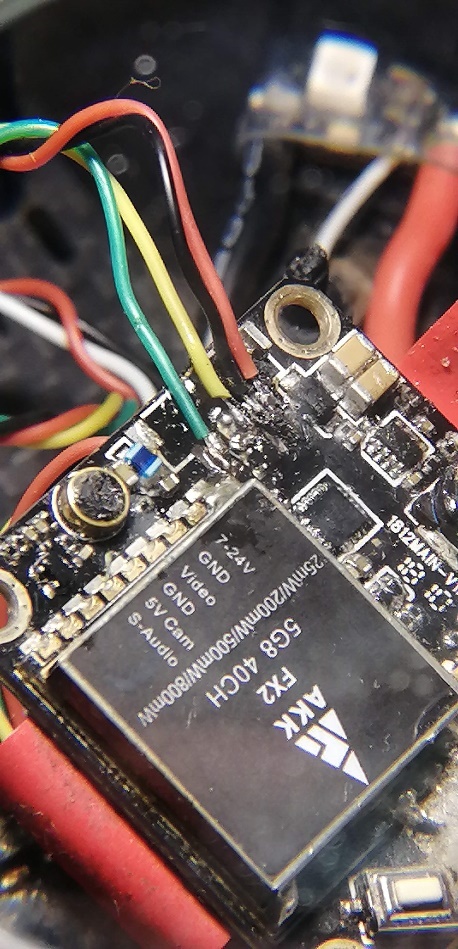
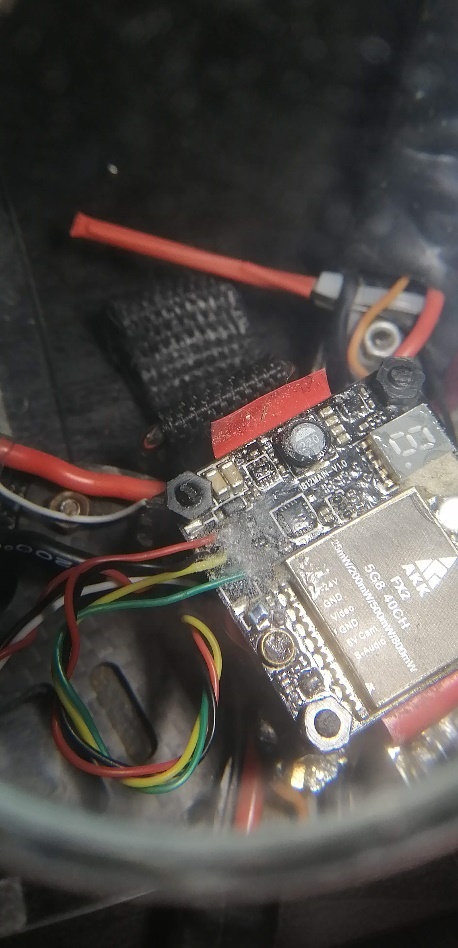
Итак, почему были выбран язык lua. Язык lua очень легко интегрируется в си подобный код. Именно поэтому разработчики выбрали именно его. Также язык lua легко учится.

1. **Дневник разработки**

У моего проекта было несколько этапов:

1. Для начала нужно было подготовить квадрокоптер к передаче и принятию данных. Для этого нужно было припаять нужный порт. К счастью таковой у меня уже был. Далее нужно было припаять провод к порту smart audio. Так получилось, что площадка с этим портом оторвалась вместе с разъемом. Разъем оторвался при неудачном падении. Пришлось припаиваться к конденсатору, стоящий после этого пина. Затем нужно было залить все провода эпоксидкой, чтобы провода ничего не оторвали при очередном плачевном падении. На картинках ниже видны вышеописанные этапы.

Также мной был настроен порт управления камерой квадрокоптера издалека. Ранее мне нужно было использовать специальный пультик (на фотке ниже). Но теперь я могу настраивать свою камеру при помощи аппаратуры удаленно. Планировалось сделать еще и вывод параметров камеры, и ее настройку на экранчик аппаратуры, но дальнейший анализ показал, что никак данные с камеры получить у меня не получится.

1. Теперь нужно настроить smart port. Из [1] и [2] мы понимаем, что нам нужно зайти в betaflight configurator и применить нужные настройки. К сожалению, в ходе работы оторвался разъем micro usb, при помощи которого происходила связь с ПК. Поэтому картинки из betaflight показать не смогу.

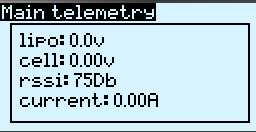
Чтобы проверить работоспособность моих пинов и правильность настройки, скачаем с гитхаба betaflight их стандартные скрипты. Попробуем изменить параметры видео передатчика. Так как недавно betaflight обновили свою прошивку, нужно было настроить **VTX таблицу.** Итак, заработало все не с первого раза, но оно работает. Данные квадрокоптеры очень капризные и все косяки чаще всего идут от его обладателя – слишком много факторов, которые способны испортить устройство.

1. Теперь нужно изучить язык программирования lua. К счастью, это оказался довольно простой язык, на изучение которого пришлось потратить всего несколько часов. Далее нужно разобраться с библиотеками и системными вызовами, которые добавили openTX и betaflight. К моему огромному сожалению документация openTX оказалась на столько ужасной: в ней не было примеров, а описание методов укладывается в несколько приложений – что мне пришлось связываться с разработчиками в дискорд чате. Благо они очень быстро отвечали. Для скриптов betaflight вообще нет документации. Пришлось разбирать код, который состоит из нескольких десятков файлов. Разобравшись во всем, я решил поменять UI данных скриптов для быстрой навигации по меню. Для этого тоже пришлось разобраться в логике программы и интегрировать свои фишки.
2. Последний этап был не менее сложный, чем несколько предыдущих. Нужно написать код. Для начала я решил поправить UI стандартных скриптов. Заодно разобрался почти полностью как работают скрипты. Дальнейшее развитие проекта читайте в 6-ом разделе. Теперь, поняв, как работает язык, методы и системные вызовы openTX, можно написать свою телеметрию.

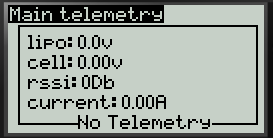
Для начала я подумал, что мне нужно звуковое оповещение низкого заряда моего БПЛА. Затем хорошо бы еще было знать состояние моего аккумулятора и потребляемый ток. Затем для отладки была использована программа

Firmware simulator от openTX. Эта программа способна эмулировать любую аппаратуру от frsky.

После долгой отладки и написания кода, родилась вот такая телеметрия.



Ранее вместо rssi было другое значение, которое не понимали что показывает даже разработчики. После очередного падения, я подумал, что неплохо бы было добавить значение rssi, чтобы было удобно искать свой квадрокоптер в траве. Потом я решил добавить немножко красоты и сделал рамочку. А также, когда rssi = 0, то есть нет сигнала, аппаратура мне выдавала соответствующую надпись.



На картинке не видно, но она мигает.

1. **Демонстрация работы**
2. **Листинг**

**t.lua –** код моей телеметрии.

**local** lastVal = {}  
  
**local function** getTelemetryId(name)  
 field = getFieldInfo(name)  
 **if** getFieldInfo(name) **then return** field.id **end  
 return** -1  
**end  
  
local** IDs = {}  
 IDs.a4 = getTelemetryId("A4")  
 IDs.vfas = getTelemetryId("VFAS")  
 IDs.curr = getTelemetryId("Curr")  
 IDs.fuel = getTelemetryId("Fuel")  
 IDs.rssi = getTelemetryId("RSSI")  
  
**local** data = {}  
 data.curr = 0  
 data.fuel = 0  
 data.vfas = 0  
 data.a4 = 0  
 data.rssi = getValue(IDs.rssi)  
  
  
**local function** getAllData()  
 ***data***.curr = getValue(***IDs***.curr) \* 100  
 ***data***.a4 = getValue(***IDs***.a4) \* 10  
 ***data***.vfas = getValue(***IDs***.vfas) \* 10  
 ***data***.fuel = getValue(***IDs***.fuel)  
 **if *data***.fuel > 100 **then *data***.fuel = 100 **end  
end  
  
local function** setAll(num)  
 ***data***.curr = num.curr  
 ***data***.fuel = num.fuel  
 ***data***.vfas = num.vfas  
 ***data***.a4 = num.a4  
**end  
  
local function** init()  
 ***getAllData***()  
 ***lastVal***.a4 = ***data***.a4  
 ***lastVal***.curr = ***data***.curr  
 ***lastVal***.vfas = 0  
 ***lastVal***.fuel = 0  
**end  
  
local** time = 0  
**local function** backgr()  
  
 ***data***.a4 = getValue(***IDs***.a4) \* 100  
  
 **if *data***.a4 < 355 **and *data***.a4 > 0 **then  
  
 if *time*** == 0 **then** playFile("Alert.wav")  
 ***time*** = ***time*** + 1  
 **elseif *time*** < 120 **then  
 *time*** = ***time*** + 1  
 **elseif *time*** == 120 **then  
 *time*** = 0  
 **end  
 else  
 *time*** = 0  
 **end  
end  
  
  
local function** run(event)  
 ***data***.rssi = getValue(***IDs***.rssi)  
 lcd.clear()  
 lcd.drawText(1,1,"Main telemetry",INVERS)  
 **local** X = 5  
 **local** Y = 10  
  
  
 lcd.drawRectangle(X,Y,120,50,1)  
 **if *data***.rssi == 0 **then  
 *setAll***(***lastVal***)  
 lcd.drawText(35,55,"No Telemetry",BLINK)  
  
 **else  
 *getAllData***()  
 ***lastVal***.a4 = ***data***.a4  
 ***lastVal***.curr = ***data***.curr  
 ***lastVal***.vfas = 0  
 ***lastVal***.fuel = 0  
 **end** lcd.drawText(X+5,Y+5,"lipo:",0)  
 lcd.drawNumber(lcd.getLastRightPos()+2,Y+5,***data***.vfas,PREC1)  
 lcd.drawText(lcd.getLastRightPos(),Y+5,"v",0)  
  
 lcd.drawText(X+5,Y+15,"cell:",0)  
 lcd.drawNumber(lcd.getLastRightPos()+2,Y+15,***data***.a4,PREC2)  
 lcd.drawText(lcd.getLastRightPos(),Y+15,"v",0)  
  
 lcd.drawText(X+5,Y+25,"rssi:",0)  
 lcd.drawNumber(lcd.getLastRightPos()+2,Y+25,***data***.rssi,0)  
 lcd.drawText(lcd.getLastRightPos(),Y+25,"Db",0)  
  
 lcd.drawText(X+5,Y+35,"current:",0)  
 lcd.drawNumber(lcd.getLastRightPos()+2,Y+35,***data***.curr,PREC2)  
 lcd.drawText(lcd.getLastRightPos(),Y+35,"A",0)  
  
  
 lcd.refresh()  
**end  
return** { background= backgr, run=run, init=init }

**ui.lua –** немного измененный код стандартных скриптов betaflight. Данный код был исправлен примерно на 30%.

**local** uiStatus =  
{  
 init = 1,  
 pages = 2,  
 mainMenu = 3,  
}  
  
**local** pageStatus =  
{  
 display = 2,  
 editing = 3,  
 saving = 4,  
 popupMenu = 5,  
}  
  
**local** uiMsp =  
{  
 reboot = 68,  
 eepromWrite = 250,  
}  
  
**local** uiState = uiStatus.init --статус, в котором находится программа  
**local** pageState = pageStatus.display  
**local** requestTimeout = 80  
**local** currentPage = 1  
**local** currentField = 1  
**local** saveTS = 0  
**local** saveTimeout = protocol.saveTimeout  
**local** saveRetries = 0  
**local** saveMaxRetries = protocol.saveMaxRetries  
**local** popupMenuActive = 1  
**local** killEnterBreak = 0  
**local** pageScrollY = 0  
**local** mainMenuScrollY = 0  
**local** PageFiles, Page, init, popupMenuList  
  
**local** backgroundFill = TEXT\_BGCOLOR **or** ERASE  
**local** foregroundColor = LINE\_COLOR **or** SOLID  
  
**local** globalTextOptions = TEXT\_COLOR **or** 0  
  
**local function** saveSettings()  
 **if *Page***.values **then  
 local** payload = ***Page***.values  
 **if *Page***.preSave **then** payload = ***Page***.preSave(***Page***)  
 **end** protocol.mspWrite(***Page***.write, payload)  
 ***saveTS*** = getTime()  
 **if *pageState*** == ***pageStatus***.saving **then  
 *saveRetries*** = ***saveRetries*** + 1  
 **else  
 *pageState*** = ***pageStatus***.saving  
 ***saveRetries*** = 0  
 **end  
 end  
end  
  
local function** invalidatePages()  
 ***Page*** = *nil* ***pageState*** = ***pageStatus***.display  
 ***saveTS*** = 0  
 collectgarbage()  
**end  
  
local function** rebootFc()  
 protocol.mspRead(***uiMsp***.reboot)  
 ***invalidatePages***()  
**end  
  
local function** eepromWrite()  
 protocol.mspRead(***uiMsp***.eepromWrite)  
**end  
  
local function** accCal()  
 ***invalidatePages***()  
 ***currentField*** = 1  
 ***Page*** = assert(loadScript("Pages/accelerometer.lua"))()  
 collectgarbage()  
**end  
  
local function** getVtxTables()  
 ***uiState*** = ***uiStatus***.init  
 ***PageFiles*** = *nil* ***invalidatePages***()  
 io.close(io.open("/BF/VTX/"..mcuId..".lua", 'w'))  
 **return** 0  
**end  
  
local function** createPopupMenu()  
 ***popupMenuList*** = {  
 { t = "save page", f = ***saveSettings*** },  
 { t = "reload", f = ***invalidatePages*** },  
 { t = "reboot", f = ***rebootFc*** },  
 { t = "acc cal", f = ***accCal*** },  
 }  
 **if** apiVersion >= 1.042 **then  
 *popupMenuList***[#***popupMenuList*** + 1] = { t = "vtx tables", f = ***getVtxTables*** }  
 **end  
end  
  
local function** processMspReply(cmd,rx\_buf)  
 **if not *Page* or not** rx\_buf **then  
 elseif** cmd == ***Page***.write **then  
 if *Page***.eepromWrite **then  
 *eepromWrite***()  
 **else  
 *invalidatePages***()  
 **end  
 elseif** cmd == ***uiMsp***.eepromWrite **then  
 if *Page***.reboot **then  
 *rebootFc***()  
 **end  
 *invalidatePages***()  
 **elseif** cmd == ***Page***.read **and** #rx\_buf > 0 **then  
 *Page***.values = rx\_buf  
 **for** i=1,#***Page***.fields **do  
 if** #***Page***.values >= ***Page***.minBytes **then  
 local** f = ***Page***.fields[i]  
 **if** f.vals **then** f.value = 0  
 **for** idx=1, #f.vals **do  
 local** raw\_val = ***Page***.values[f.vals[idx]] **or** 0  
 raw\_val = bit32.lshift(raw\_val, (idx-1)\*8)  
 f.value = bit32.bor(f.value, raw\_val)  
 **end** f.value = f.value/(f.scale **or** 1)  
 **end  
 end  
 end  
 if *Page***.postLoad **then  
 *Page***.postLoad(***Page***)  
 **end  
 end  
end  
  
local function** incMax(val, inc, base)  
 **return** ((val + inc + base - 1) % base) + 1  
**end  
  
function** clipValue(val,min,max)  
 **if** val < min **then** val = min  
 **elseif** val > max **then** val = max  
 **end  
 return** val  
**end  
  
local function** incPage(inc)  
 ***currentPage*** = ***incMax***(***currentPage***, inc, #***PageFiles***)  
 ***currentField*** = 1  
 ***invalidatePages***()  
**end  
  
local function** getWeel (min, max)  
 **local** s2Val = getValue('s2') + 1024  
 **local** result = math.floor(s2Val\*(max - min + 1)/2048) + 1  
  
 **if** result > max **then** result = max **end  
 return** result  
**end  
  
local function** incField()  
 ***currentField*** = ***getWeel***(1, #***Page***.fields)  
**end  
  
local function** incMainMenu(inc)  
 ***currentPage*** = clipValue(***currentPage*** + inc, 1, #***PageFiles***)  
**end  
  
local function** incPopupMenu(inc)  
 ***popupMenuActive*** = clipValue(***popupMenuActive*** + inc, 1, #***popupMenuList***)  
**end  
  
local function** requestPage()  
 **if *Page***.read **and** ((**not *Page***.reqTS) **or** (***Page***.reqTS + ***requestTimeout*** <= getTime())) **then  
 *Page***.reqTS = getTime()  
 protocol.mspRead(***Page***.read)  
 **end  
end  
  
local function** drawScreenTitle(screenTitle)  
 **if** radio.highRes **then** lcd.drawFilledRectangle(0, 0, LCD\_W, 30, TITLE\_BGCOLOR)  
 lcd.drawText(5,5,screenTitle, MENU\_TITLE\_COLOR)  
 **else** lcd.drawFilledRectangle(0, 0, LCD\_W, 10, FORCE)  
 lcd.drawText(1,1,screenTitle,INVERS)  
 **end  
end  
  
local function** drawScreen()  
 **local** yMinLim = radio.yMinLimit  
 **local** yMaxLim = radio.yMaxLimit  
 **local** currentFieldY = ***Page***.fields[***currentField***].y  
 **local** textOptions = radio.textSize + ***globalTextOptions* if** currentFieldY <= ***Page***.fields[1].y **then  
 *pageScrollY*** = 0  
 **elseif** currentFieldY - ***pageScrollY*** <= yMinLim **then  
 *pageScrollY*** = currentFieldY - yMinLim  
 **elseif** currentFieldY - ***pageScrollY*** >= yMaxLim **then  
 *pageScrollY*** = currentFieldY - yMaxLim  
 **end  
 for** i=1,#***Page***.labels **do  
 local** f = ***Page***.labels[i]  
 **local** y = f.y - ***pageScrollY* if** y >= 0 **and** y <= LCD\_H **then** lcd.drawText(f.x, y, f.t, textOptions)  
 **end  
 end  
 local** val = "---"  
 **for** i=1,#***Page***.fields **do  
 local** f = ***Page***.fields[i]  
 **local** valueOptions = textOptions  
 **if** i == ***currentField* then** valueOptions = valueOptions + INVERS  
 **if *pageState*** == ***pageStatus***.editing **then** valueOptions = valueOptions + BLINK  
 **end  
 end   
 if** f.value **then  
 if** f.upd **and *Page***.values **then** f.upd(***Page***)  
 **end** val = f.value  
 **if** f.table **and** f.table[f.value] **then** val = f.table[f.value]  
 **end  
 end  
 local** y = f.y - ***pageScrollY* if** y >= 0 **and** y <= LCD\_H **then  
 if** f.t **then** lcd.drawText(f.x, y, f.t, textOptions)  
 **end** lcd.drawText(f.sp **or** f.x, y, val, valueOptions)  
 **end  
 end  
 *drawScreenTitle***("BF / "..***Page***.title.." field "..***getWeel***(1,#***Page***.fields))  
**end  
  
local function** incValue(inc)  
 **local** f = ***Page***.fields[***currentField***]  
 **local** scale = f.scale **or** 1  
 **local** mult = f.mult **or** 1  
 f.value = clipValue(f.value + inc\*mult/scale, (f.min **or** 0)/scale, (f.max **or** 255)/scale)  
 f.value = math.floor(f.value\*scale/mult + 0.5)\*mult/scale  
 **for** idx=1, #f.vals **do  
 *Page***.values[f.vals[idx]] = bit32.rshift(math.floor(f.value\*scale + 0.5), (idx-1)\*8)  
 **end  
 if** f.upd **and *Page***.values **then** f.upd(***Page***)  
 **end  
end  
  
local function** drawPopupMenu()  
 **local** x = radio.MenuBox.x  
 **local** y = radio.MenuBox.y  
 **local** w = radio.MenuBox.w  
 **local** h\_line = radio.MenuBox.h\_line  
 **local** h\_offset = radio.MenuBox.h\_offset  
 **local** h = #***popupMenuList*** \* h\_line + h\_offset\*2  
  
 lcd.drawFilledRectangle(x,y,w,h,***backgroundFill***)  
 lcd.drawRectangle(x,y,w-1,h-1,***foregroundColor***)  
 lcd.drawText(x+h\_line/2,y+h\_offset,"Menu:",***globalTextOptions***)  
  
 **for** i,e **in** ipairs(***popupMenuList***) **do  
 local** textOptions = ***globalTextOptions* if *popupMenuActive*** == i **then** textOptions = textOptions + INVERS  
 **end** lcd.drawText(x+radio.MenuBox.x\_offset,y+(i-1)\*h\_line+h\_offset,e.t,textOptions)  
 **end  
end  
  
local function** run\_ui(event)  
 **if *uiState*** == ***uiStatus***.init **then** lcd.clear()  
 ***drawScreenTitle***("Betaflight Config")  
 ***init*** = ***init* or** assert(loadScript("ui\_init.lua"))()  
 **if not *init***() **then  
 return** 0  
 **end  
 *init*** = *nil* ***createPopupMenu***()  
 ***PageFiles*** = assert(loadScript("pages.lua"))()  
 ***invalidatePages***()  
 ***uiState*** = ***uiStatus***.mainMenu  
 **elseif *uiState*** == ***uiStatus***.mainMenu **then  
 if** event == EVT\_VIRTUAL\_EXIT **then  
 return** 2  
 **elseif** event == EVT\_VIRTUAL\_NEXT **then  
 *incMainMenu***(1)  
 **elseif** event == EVT\_VIRTUAL\_PREV **then  
 *incMainMenu***(-1)  
 **elseif** event == EVT\_VIRTUAL\_ENTER **then  
 *uiState*** = ***uiStatus***.pages  
 **end** lcd.clear()  
 **local** yMinLim = radio.yMinLimit  
 **local** yMaxLim = radio.yMaxLimit  
 **local** lineSpacing = 10  
 **if** radio.highRes **then** lineSpacing = 25  
 **end  
 local** currentFieldY = (***currentPage***-1)\*lineSpacing + yMinLim  
 **if** currentFieldY <= yMinLim **then  
 *mainMenuScrollY*** = 0  
 **elseif** currentFieldY - ***mainMenuScrollY*** <= yMinLim **then  
 *mainMenuScrollY*** = currentFieldY - yMinLim  
 **elseif** currentFieldY - ***mainMenuScrollY*** >= yMaxLim **then  
 *mainMenuScrollY*** = currentFieldY - yMaxLim  
 **end  
 for** i=1, #***PageFiles* do  
 local** attr = ***currentPage*** == i **and** INVERS **or** 0  
 **local** y = (i-1)\*lineSpacing + yMinLim - ***mainMenuScrollY* if** y >= 0 **and** y <= LCD\_H **then** lcd.drawText(6, y, ***PageFiles***[i].title, attr)  
 **end  
 end  
 *drawScreenTitle***("Betaflight Config")  
 **elseif *uiState*** == ***uiStatus***.pages **then  
 if *pageState*** == ***pageStatus***.saving **then  
 if *saveTS*** + ***saveTimeout*** < getTime() **then  
 if *saveRetries*** < ***saveMaxRetries* then  
 *saveSettings***()  
 **else  
 *pageState*** = ***pageStatus***.display  
 ***invalidatePages***()  
 **end  
 end  
 elseif *pageState*** == ***pageStatus***.popupMenu **then  
 if** event == EVT\_VIRTUAL\_EXIT **then  
 *pageState*** = ***pageStatus***.display  
 **elseif** event == EVT\_VIRTUAL\_PREV **then  
 *incPopupMenu***(-1)  
 **elseif** event == EVT\_VIRTUAL\_NEXT **then  
 *incPopupMenu***(1)  
 **elseif** event == EVT\_VIRTUAL\_ENTER **then  
 if *killEnterBreak*** == 1 **then  
 *killEnterBreak*** = 0  
 **else  
 *pageState*** = ***pageStatus***.display  
 **return *popupMenuList***[***popupMenuActive***].f() **or** 0  
 **end  
 end  
 elseif *pageState*** == ***pageStatus***.display **then  
 if** event == EVT\_VIRTUAL\_PREV\_PAGE **then  
 *incPage***(-1)  
 killEvents(event) -- X10/T16 issue: pageUp is a long press  
 **elseif** event == EVT\_VIRTUAL\_NEXT\_PAGE **then  
 *incPage***(1)  
 **elseif** event == EVT\_VIRTUAL\_PREV **or** event == EVT\_VIRTUAL\_PREV\_REPT **then  
 *incField***()  
 **elseif** event == EVT\_VIRTUAL\_NEXT **or** event == EVT\_VIRTUAL\_NEXT\_REPT **then  
 *incField***()  
 **elseif** event == EVT\_VIRTUAL\_ENTER **then  
 if *Page* then  
 local** f = ***Page***.fields[***currentField***]  
 **if** f.onClick **then** f.onClick(***Page***)  
 **end  
 if *Page***.values **and** f.vals **and *Page***.values[f.vals[#f.vals]] **and not** f.ro **then  
 *pageState*** = ***pageStatus***.editing  
 **end  
 end  
 elseif** event == EVT\_VIRTUAL\_ENTER\_LONG **then  
 *popupMenuActive*** = 1  
 ***killEnterBreak*** = 1  
 ***pageState*** = ***pageStatus***.popupMenu  
 **elseif** event == EVT\_VIRTUAL\_EXIT **then  
 *invalidatePages***()  
 ***currentField*** = 1  
 ***uiState*** = ***uiStatus***.mainMenu  
 **return** 0  
 **end  
 elseif *pageState*** == ***pageStatus***.editing **then  
 if** event == EVT\_VIRTUAL\_EXIT **or** event == EVT\_VIRTUAL\_ENTER **then  
 if *Page***.fields[***currentField***].postEdit **then  
 *Page***.fields[***currentField***].postEdit(***Page***)  
 **end  
 *pageState*** = ***pageStatus***.display  
 **elseif** event == EVT\_VIRTUAL\_INC **or** event == EVT\_VIRTUAL\_INC\_REPT **then  
 *incValue***(1)  
 **elseif** event == EVT\_VIRTUAL\_DEC **or** event == EVT\_VIRTUAL\_DEC\_REPT **then  
 *incValue***(-1)  
 **end  
 end  
 if not *Page* then  
 *Page*** = assert(loadScript("Pages/"..***PageFiles***[***currentPage***].script))()  
 collectgarbage()  
 **end  
 if not *Page***.values **and *pageState*** == ***pageStatus***.display **then  
 *requestPage***()  
 **end** lcd.clear()  
 ***drawScreen***()  
 **if *pageState*** == ***pageStatus***.popupMenu **then  
 *drawPopupMenu***()  
 **elseif *pageState*** == ***pageStatus***.saving **then  
 local** saveMsg = "Saving..."  
 **if *saveRetries*** > 0 **then** saveMsg = "Retrying"  
 **end** lcd.drawFilledRectangle(radio.SaveBox.x,radio.SaveBox.y,radio.SaveBox.w,radio.SaveBox.h,***backgroundFill***)  
 lcd.drawRectangle(radio.SaveBox.x,radio.SaveBox.y,radio.SaveBox.w,radio.SaveBox.h,SOLID)  
 lcd.drawText(radio.SaveBox.x+radio.SaveBox.x\_offset,radio.SaveBox.y+radio.SaveBox.h\_offset,saveMsg,DBLSIZE + ***globalTextOptions***)  
 **end  
 end  
 if** protocol.rssi() == 0 **then** lcd.drawText(radio.NoTelem[1],radio.NoTelem[2],radio.NoTelem[3],radio.NoTelem[4])  
 **end** mspProcessTxQ()  
 ***processMspReply***(mspPollReply())  
 **return** 0  
**end  
  
return** run\_ui

1. **Дальнейшее развитие проекта**

В будущем планируется сделать следующее:

* подсчет в mah емкость, потраченная на полет
* отображение в виде индикатора батарейки заряд батареи на квадрокоптере
* построение графика рейтов, как в betaflight (картинка ниже)



1. **Вывод**

Благодаря проделанной работе, я настроил свой БПЛА так, что теперь мне не нужно каждый раз заходить в betaflight configurator. Также узнал про язык lua. Научился делать скрипты для своей аппаратуры. Это открыло мне новые возможности при полете и поиске своего квадрокоптера.

Также в ходе выполнения работы я понял на сколько важно делать качественную библиотеку с примерами. Иначе разработчикам будет очень сложно делать необходимые дополнения для какой-то вещи.

**Список литературы**

1. Руководство: меняем PID’ы Betaflight и настройки видео передатчика прямо с Тараниса *— RCDetails Blog*

URL:<https://blog.rcdetails.info/rukovodstvo-menyaem-pid-y-betaflight-i-nastrojki-video-peredatchika-pryamo-s-taranisa/>

1. Распиновка полетного контроллера OMNIBUS F4 Fireworks v2

URL:<https://store.myairbot.com//media/iverve/uploadpdf/1521998647_OMNIBUSF4Fireworks_V2.pdf>

1. Issues betaflight lua script – *github*

URL:<https://github.com/betaflight/betaflight-tx-lua-scripts/issues>

1. OpenTX downloads

URL:<https://www.open-tx.org/downloads>

1. Index of /2.3/release/sdcard/opentx-xlite/

URL:<https://downloads.open-tx.org/2.3/release/sdcard/opentx-xlite/>

1. betaflight lua script – *github*

URL:[https://github.com/betaflight/betaflight-tx-lua-scripts](https://github.com/betaflight/betaflight-tx-lua-scripts/issues)

1. Руководство: как использовать Betaflight SoftSerial – *RCDetails Blog*

URL:<https://blog.rcdetails.info/rukovodstvo-kak-ispolzovat-betaflight-softserial/>

1. Что такое порт softserial? Настройка и как сделать дополнительный UART порт – *profpv*

URL:<https://profpv.ru/chto-takoe-softserial-opisanie-i-nastrojka/>

1. Форум написания скриптов для OpenTX

URL:<http://forum.rcdesign.ru/f4/thread367143.html>

1. LUA 501 – Introduction to Lua Scripts – *open-txu*

URL:<http://open-txu.org/home/grad-school/lua/foundations-of-lua/>

1. Lua scripting in OpenTX – *github*

URL:<https://github.com/opentx/opentx/wiki/Lua-scripting-in-OpenTX>

1. Еще один форум по lua скриптингу

URL:<https://www.rcgroups.com/forums/showthread.php?2180477-LUA-scripting-Technical-discussion>

1. OpenTX - LUA Reference Guide

URL:<https://docs.google.com/document/d/1ILh9lSW4d05qJD2rSKxep0WnJUfNCcAPCafKZeESweE/mobilebasic?pli=1#h.5tqx7jv6yan5>

1. OpenTX 2.1 Lua Reference Guide – *gitbook*

URL:<https://opentx.gitbooks.io/opentx-lua-reference-guide/content/>

1. OpenTX 2.3 Lua Reference Guide – *gitbook*

URL:<https://opentx.gitbooks.io/opentx-2-3-lua-reference-guide/content/>

1. Lua 5.3 Руководство – *lua.org*

URL:<https://lua.org.ru/contents_ru.html>

1. Betaflight telemetry – *github*

URL:<https://github.com/betaflight/betaflight/blob/master/docs/Telemetry.md>

1. How to make your Taranis read out low voltage – *youtube*

URL:<https://www.youtube.com/watch?v=2xt623VAMEk>

1. The A4 telemetry sensor is the one thing FrSky does better than Crossfire (what even is that) – *youtube*

URL:<https://www.youtube.com/watch?v=Ek-E_rVu59U>

1. Lua Tutorial – *youtube*

URL:<https://www.youtube.com/watch?v=iMacxZQMPXs&t=1037s&ab_channel=DerekBanas>

1. Еще огромное количество видео малой важности на канале **joshua bardwell** на youtube
2. Чат разработчиков дискорд сервера OpenTX