

```
--[[ pokeys_fsx.lua
```

```
lua interface between FSX and PoKeys USB / PoKeys Ethernet (PoKeys56E)
using lua5.1.dll and luacom.dll
```

```
Written by Terry Hall, terry.hall@afi.cc (October 5, 2011)
```

```
For this lua to work the following must be completed.
```

1. Copy lua5.1.dll to the FSX root directory
example: C:\Program Files\Microsoft Games\Microsoft Flight Simulator X
2. Install FSUIPC by Peter Dowson (<http://www.schiratti.com/dowson.html>)
3. Create a subdirectory called "lua" in "Modules" directory where your FSX is installed.

ex: "C:\Program Files\Microsoft Games\Microsoft Flight Simulator X\Modules\lua"
4. Copy luacom.dll into the directory created in step 3. I am using luacom version 1.4-1

```
In order for FSX to execute the pokeys_fsx.lua I added an entry in the FSUIPC lua called ipcready.lua. (See FSUIPC manuals for details about ipcready.lua)
```

```
ipc.runlua("pokeys_fsx")
```

```
Hint: If you have multiple poKeys devices make a copy of pokeys_fsx.lua into a new file name (ex: pokeys_1_fsx.lua) and run the new lua using the ipc.runlua() command.
```

```
ex: ipc.runlua("pokeys_1_fsx")
```

```
In this example I have the following setup
```

1. One toggle switch (Digital Input) connect to pin 10 - FSX Battery
2. One toggle switch (Digital Input) connect to pin 11 - FSX Landing Light
3. Encoder 1 connected to pin 1 and 2
4. Encoder 2 connected to pin 3 and 4
5. Pin 50 set to Digital Output (Nose Gear)
6. Pin 51 set to Digital Output (Right Gear)
7. Pin 52 set to Digital Output (Left Gear)
8. Pin 53 set to Digital Output (Flap Indicator)

```
The digital output (pin 50-53) triggers a UNL2803A circuit to turn the LED on or off.
```

```
For the Pokeys56E device I configured it at a fixed IP address of 192.168.1.120  
Also you should use the event.timer function to keep the poKeys56E network connection active otherwise after apx three seconds of no activity it will close the connection
```

```
*****
```

```
* IMPORTANT * Because the luacom routine is a wrapper for interfacing to a DLL  
***** it handles the way values are returned from DLL functions differently.
```

```
Example: Typical Code ---> cmd_check = device.GetInput(0, pinState)  
luacom Code ---> cmd_check, pinState = device.GetInput(0)
```

```
=====]]
```

```
-- =====
```

```
-- Initial Setup
```

```
-- =====
```

```
require "luacom"  
poKeys = luacom.CreateObject("PoKeysDevice_DLL.PoKeysDevice")
```

```
if poKeys == nil then
```

```

    ipc.log("Error: Unable to create PoKeysDevice_DLL object")
    ipc.exit()
end

flaps_led = 0
lastEncoder1 = -1
lastEncoder2 = -1

connect_usb = true    -- true = USB / false = ethernet

-- =====
-- Connect to PoKeys USB Device
-- =====

function connect_to_usb()

    -- Get Number of poKeys Devices Found
    no_devices = poKeys:EnumerateDevices()

    -- Connect to Device

    if (num_devices ~= 0) then
        connect_flag = poKeys:ConnectToDevice(0)
        if(connect_flag == false)then
            -- code to perform if ConnectToDevice fails
            ipc.log("Error connecting to poKeys USB Device 0")
            ipc.exit()
        end
    end
end

-- =====
-- Connect to PoKeys Ethernet Device
-- =====

function connect_to_ethernet()

    eth_con = poKeys:ConnectToNetworkDevice("192.168.1.120")

    if (eth_con == false) then
        -- connection failed
        ipc.log("Error: Unable to connect to poKeys56E at 192.168.1.120")
        ipc.exit()
    end
end

-- =====
-- Set Digital Output Devices
-- =====

function output_setup()

    poKeys:SetPinData(49,4)
    poKeys:SetPinData(50,4)
    poKeys:SetPinData(51,4)
    poKeys:SetPinData(52,4)
end

-- =====
-- check current physical switch setting and update FSX software switch to match
-- =====

function fsx_setup()

    -- This section checks each switch I have connected to the poKeys device
    -- and sets the fsx switch to match the physical switch position.

```

```

-- I have included samples when using the poKeys USB device configured as
-- joystick buttons and reading the poKeys device via the DLL

-- =====
-- Check Battery Switch - Pin 10
-- =====

-- =====
-- poKeys USB device using Joystick Button Mapping
-- =====

if ipc.testbutton(10,0) == true then
    ipc.writeUD(0x281C, 1) -- set FSX battery switch ON
else
    ipc.writeUD(0x281C, 0) -- set FSX battery switch OFF
end

-- =====
-- poKeys USB device using Joystick Button Mapping
-- sample of using ipc.control and fsx toggle
-- switches that toggle instead of separate on
-- and off ipc.control values
-- =====

if ipc.testbutton(10,0) == true then
    -- physical switch in ON
    if ipc.readUD(0x281C) == 0 then -- FSX switch is Off
        ipc.control(66241) -- toggle FSX switch to ON
    end
else
    -- physical switch in OFF
    if ipc.readUD(0x281C) == 1 then -- FSX switch is On
        ipc.control(66241) -- toggle FSX switch to OFF
    end
end

-- =====
-- poKeys switch status via DLL interface
-- =====

chk_pin_10 = false
chk_pin_10, onoff_pin_10 = poKeys:GetInput(9)

if (chk_pin_10 == false) then
    -- GetInput Failed... Most likely not configured properly
    -- Put code here to perform for failure
else
    -- We got results from the switch GetInput Function
    if (onoff_pin_10 == true) then
        -- Switch at Pin 10 is ON
        -- Put code here you wish FSX to perform
        ipc.writeUD(0x281C, 1) -- set FSX battery switch ON
    else
        if (onoff_pin_10 == false) then
            -- Switch at Pin 10 is OFF
            -- Put code here you wish FSX to perform
            ipc.writeUD(0x281C, 0) -- set FSX battery switch OFF
        else
            -- Could not determine status of Switch at Pin 10
            -- Put code here you wish FSX to perform
        end
    end
end
end -- chk_pin_10

-- =====
-- Check Landing Light Switch - Pin 11
-- =====

```

```

-- =====
-- poKeys USB device using Joystick Button Mapping
-- =====

if ipc.testbutton(1,11) == true then
    ipc.setbitsUW(0x0d0c, 4) -- set FSX Landing Light ON
else
    ipc.clearbitsUW(0x0d0c, 4) -- set FSX Landing Light OFF
end

-- =====
-- poKeys switch status via DLL interface
-- =====

chk_pin_11 = false
chk_pin_11, onoff_pin_11 = poKeys:GetInput(10)

if (chk_pin_11 == false) then
    -- GetInput Failed... Most likely not configured properly
    -- Put code here to perform for failure
else
    -- We got results from the switch GetInput Function
    if (onoff_pin_11 == true) then
        -- Switch at Pin 11 is ON
        -- Put code here you wish FSX to perform
        ipc.setbitsUW(0x0d0c, 4) -- set FSX Landing Light ON
    else
        if (onoff_pin_11 == false) then
            -- Switch at Pin 11 is OFF
            -- Put code here you wish FSX to perform
            ipc.clearbitsUW(0x0d0c, 4) -- set FSX Landing Light OFF
        else
            -- Could not determine status of Switch at Pin 11
            -- Put code here you wish FSX to perform
        end
    end
end -- chk_pin_11

end

-- =====
-- check current encoder / switch status

-- This function is trigger by the event.timer

-- You can adjust in milliseconds how often to check the status of the
-- encoders and switches you have connected

-- I found that a timing of 50 to 100 milliseconds works ok.

-- NOTE: If you are using a poKeys USB device it is much more efficient and
-- simpler to use the Joystick / Keyboard emulation features of the
-- poKeys device to interface to FSX via FSUIPC

-- =====

function check_switches()

    -- =====
    -- Get poKeys Internal Tick Counter
    -- =====

    -- If you are using the poKeys ethernet device and you are not monitoring
    -- any switches you need to at least read the poKeys tick counter
    -- every second or two so that the device does not disconnect for
    -- no activity

```

```

-- tick_count = poKeys:GetTickCounter()

-- =====
-- Check Values of Encoder 1 at Pin 1 and 2
-- =====

-- FYI: The encoder returns a value between 0 and 255. As you
-- turn the encoder clockwise when the value reaches 255 it
-- resets to zero (0). The reverse is true when turning counter
-- clockwise. In the encoder sample below I did not include any
-- coding to monitor this transition. What happens is that as
-- you turn clockwise at the point the value changes from 255
-- to 0 (or 0 to 255) the indicator moves in the opposite
-- direction for that one click.

chk_encoder_1 = false
chk_encoder_1, curEncoder1 = poKeys:GetEncoderValue(0)

if (chk_encoder_1 == false) then
  -- GetEncoderValue Failed... Most likely not configured properly
  -- Put code here to perform for failure
else
  if (lastEncoder1 ~= curEncoder1) then
    if (curEncoder1 > lastEncoder1) then
      ipc.control(65663) -- Increment VOR1 OBI Indicator
    else
      ipc.control(65662) -- Decrement VOR1 OBI Indicator
    end
    lastEncoder1 = curEncoder1
  end
end

-- =====
-- Check Values of Encoder 2 at Pin 3 and 4
-- =====

chk_encoder_2 = false
chk_encoder_2, curEncoder2 = poKeys:GetEncoderValue(1)

if (chk_encoder_2 == false) then
  -- GetEncoderValue Failed... Most likely not configured properly
  -- Put code here to perform for failure
else
  if (lastEncoder2 ~= curEncoder2) then
    if (curEncoder2 > lastEncoder2) then
      ipc.control(65665) -- Increment VOR2 OBI Indicator
    else
      ipc.control(65664) -- Decrement VOR2 OBI Indicator
    end
    lastEncoder2 = curEncoder2
  end
end

-- =====
-- Check Switch Position Connect to Pin 10
-- =====

chk_pin_10 = false
chk_pin_10, onoff_pin_10 = poKeys:GetInput(9)

if (chk_pin_10 == false) then
  -- GetInput Failed... Most likely not configured properly
  -- Put code here to perform for failure
else
  -- We got results from the switch GetInput Function
  if (onoff_pin_10 == true) then
    -- Switch at Pin 10 is ON
    -- Put code here you wish FSX to perform

```

```

    ipc.writeUD(0x281C, 1) -- set FSX battery switch ON
else
    if (onoff_pin_10 == false) then
        -- Switch at Pin 10 is OFF
        -- Put code here you wish FSX to perform
        ipc.writeUD(0x281C, 0) -- set FSX battery switch OFF
    else
        -- Could not determine status of Switch at Pin 10
        -- Put code here you wish FSX to perform
    end
end
end -- chk_pin_10

-- =====
-- Check Switch Position Connect to Pin 11
-- =====

chk_pin_11 = false
chk_pin_11, onoff_pin_11 = poKeys:GetInput(10)

if (chk_pin_11 == false) then
    -- GetInput Failed... Most likely not configured properly
    -- Put code here to perform for failure
else
    -- We got results from the switch GetInput Function
    if (onoff_pin_11 == true) then
        -- Switch at Pin 11 is ON
        -- Put code here you wish FSX to perform
        ipc.setbitsUW(0x0d0c, 4) -- set FSX Landing Light ON
    else
        if (onoff_pin_11 == false) then
            -- Switch at Pin 11 is OFF
            -- Put code here you wish FSX to perform
            ipc.clearbitsUW(0x0d0c, 4) -- set FSX Landing Light OFF
        else
            -- Could not determine status of Switch at Pin 11
            -- Put code here you wish FSX to perform
        end
    end
end
end -- chk_pin_11

end -- check_switches

-- =====
-- check flap position
--
-- This routine is triggered from FSX when the flap position changes.
--
-- I use this to turn an LED ON to indicate that the flaps are not in the
-- Full Up position
-- =====

function flap_check(off, val)

    if (val == 16383) then -- Flaps Full Down
        if (flaps_led == 0) then
            set_results = poKeys:SetOutput(52,1) -- Turn LED On
            if (set_results == false) then
                -- code to perform if SetOutput fails
                ipc.log("Error: Unable to Set Output for Pin 53")
            else
                -- Perform additional task you may want to do after turning LED On
                flaps_led = 1
            end
        end
    end
else
    if (val == 0) then -- Flaps Full Up

```

```

    set_results = poKeys:SetOutput(52,0) -- Turn LED Off
    if (set_results == false) then
        -- code to perform if SetOutput fails
        ipc.log("Error: Unable to Set Output for Pin 53")
        ipc.exit()
    else
        -- Perform additional task you may want to do after turning LED Off
        flaps_led = 0
    end
end
else
    -- Flaps Deployed but not in Full Down position
    if (flaps_led == 0) then
        set_results = poKeys:SetOutput(52,1) -- Turn LED On
        if (set_results == false) then
            -- code to perform if SetOutput fails
            ipc.log("Error: Unable to Set Output for Pin 53")
            ipc.exit()
        else
            -- Perform additional task you may want to do after turning LED Off
            flaps_led = 1
        end
    end
end
end
end
end
end

```

```

-- =====
-- check nose gear
-- =====

```

```

function gear_nose(off, val)
    if (val == 16383) then
        set_results = poKeys:SetOutput(49,1) -- Turn LED Pin 50 On
        if (set_results == false) then
            -- code to perform if SetOutput fails
            ipc.log("Error: Unable to Set Output for Pin 50")
        else
            -- Perform additional task you may want to do after turning LED On
        end
    end
else
    if (val == 0) then
        set_results = poKeys:SetOutput(49,0) -- Turn LED Pin 50 Off
        if (set_results == false) then
            -- code to perform if SetOutput fails
            ipc.log("Error: Unable to Set Output for Pin 50")
        else
            -- Perform additional task you may want to do after turning LED Off
        end
    end
end
end
end
end

```

```

-- =====
-- check right gear
-- =====

```

```

function gear_right(off, val)
    if (val == 16383) then
        set_results = poKeys:SetOutput(50,1) -- Turn LED Pin 51 On
        if (set_results == false) then
            -- code to perform if SetOutput fails
            ipc.log("Error: Unable to Set Output for Pin 51")
        else
            -- Perform additional task you may want to do after turning LED On
        end
    end
else
    if (val == 0) then
        set_results = poKeys:SetOutput(50,0) -- Turn LED Pin 51 Off
    end
end
end

```

```

        if(set_results == false)then
            -- code to perform if SetOutput fails
            ipc.log("Error: Unable to Set Output for Pin 51")
        else
            -- Perform additional task you may want to do after turning LED Off
        end
    end
end
end
end

```

```

-- =====
-- check left gear
-- =====

```

```

function gear_left(off, val)
    if (val == 16383) then
        set_results = poKeys:SetOutput(51,1) -- Turn LED Pin 52 On
        if(set_results == false)then
            -- code to perform if SetOutput fails
            ipc.log("Error: Unable to Set Output for Pin 52")
        else
            -- Perform additional task you may want to do after turning LED On
        end
    else
        if (val == 0) then
            set_results = poKeys:SetOutput(51,0) -- Turn LED Pin 52 Off
            if(set_results == false)then
                -- code to perform if SetOutput fails
                ipc.log("Error: Unable to Set Output for Pin 52")
            else
                -- Perform additional task you may want to do after turning LED Off
            end
        end
    end
end
end
end

```

```

-- =====
-- Initial Script Startup
-- =====

```

```

if (connect_usb == true) then
    connect_to_usb()
else
    connect_to_ethernet()
end

```

```

output_setup() -- set digital output pins as required
fsx_setup()    -- check physical switch and configure fsx software switch

```

```

-- =====
-- Event Section
-- =====

```

```

-- Gear up / down event
event.offset(0x0BEC, "UD", "gear_nose")
event.offset(0x0BF0, "UD", "gear_right")
event.offset(0x0BF4, "UD", "gear_left")
event.offset(0x0BDC, "UD", "flap_check")
event.timer(100,"check_switches")

```