Versatile, Trusted, Open

ArduPilot Lua Scripting

Yuri
April 2022

INTRODUCTION

- **Who I am:**
  - Retired military pilot, now working in the defense sector
  - 20+ year old BS in Computer Science
  - Avid robotics and autonomous vehicle enthusiast
  - A guy with too many hobbies!

    Electronics, physical computing, CNC machining, 3D design (CAD/CAM), anything with an engine
  - Fairly well versed in ArduPilot Lua Scripting

- **Who I'm not:**
  - An ArduPilot dev team member
  - A software engineer
  - Paid to do this
  - Fluent in ArduPilot Lua Scripted aerobatics

OVERVIEW

- Lua syntax/structures
- ArduPilot Lua
  - Documentation
  - Idiosyncrasies
  - VS Code / SITL setup
- Demos
  - Hello World!
  - Hello World! (recursive)
  - Hello World! (state machine)
  - RC Input + Servo / Relay Output
  - Acting on Sensor Input
  - Parameter Access
  - SD File Access
- Question / Answer Session

LUA IN A NUTSHELL

- Lightweight, embeddable
- Supports OO and functional paradigms
- Dynamically (loosely) typed

  nil, boolean, number, string, function, userdata, thread, table
- Keywords (always lowercase)

  `and, break, do, else, elseif, end, false, for, function, goto, if, in, local, nil, not, or, repeat, return, then, true, until, while`
- Operators / Tokens

  `+ - * / % ^ # & ~ | << >> // == ~= <= >= < > = ( ) { } [ ] :: ; : , . .. ...`

  (NOTE: ~ is the “not” operator)
- Format/indentation agnostic
  - Semantically interpreted
  - Capitalization matters
  - Whitespace kind of matters
- Lexically scoped
  - Variables exist within blocks (e.g., `keyword..end`)
    - Until they don’t!
    - Use `local` to write cleaner scripts
- Basic data structure: Tables
  - Akin to arrays
  - 1 indexed
    - Until they aren’t!

Someone will hate me for saying that Lua is a like a refined version of Visual Basic Scripting

Documentation

- https://www.lua.org/
- Online Demo (lua.org)
- Lua Scripts — (ardupilot.org Wiki)
    - (also exists for Copter and Plane)
- ardupilot/bindings.desc (github.com)
- AP_Scripting/examples (github.com)
- ardupilot/docs.lua (github.com)
- Messages (common) · MAVLink
- ardupilotmega.xml · MAVLink
    - (also, ArduPilot/mavlink/ardupilotmega.xml)

ARDUPILOT LUA

- Enabled with `SCR_ENABLE=1`
  - Not available on all autopilots (<2MB flash)
  - May need to increase `SCR_HEAP_SIZE`
  - Must reboot for changes to take effect
- Each script has its own “sandbox”
  - Each runs periodically, as allocated by a scheduler
  - Attempting too much script logic in one period will cause a (graceful) failure
- Syntax errors will simply fail and exit
- Other errors should result in script reloading

“If scripts run out of memory (or panic for any reason) all currently running scripts are terminated, and the scripting engine will restart, and reload all scripts from the disk. This is allowed to happen at all flight stages, even while the vehicle is armed and flying.”

- Bindings are selectively enabled direct access to C++ objects/methods
  - Generally, as immutable userdata objects

- KEY TAKEAWAYS:
  - Most ArduPilot scripts run in a pseudo-recursive loop
  - e.g., `return update, 500`
  - They do not create an ever-growing recursion stack
  - “Recursive” scheduling calls cannot be made to accept arguments – use outer scope variables to retain data from one iteration to the next

DEMO TIME!

- Hello World!
- Hello World! (recursive)
- Hello World! (state machine)
- RC Input + Servo / Relay Output
- Acting on Sensor Input
- Parameter Access
- SD File Access

DISCUSSION / QUESTIONS

ON A SHORT BREAK