Der Lua Skripteditor

Lua in der Messtechnik

02.08.2014
Ingo Berg
berg@atvoigt.de
Automatisierungstechnik Voigt GmbH

Lua in der Messtechnik

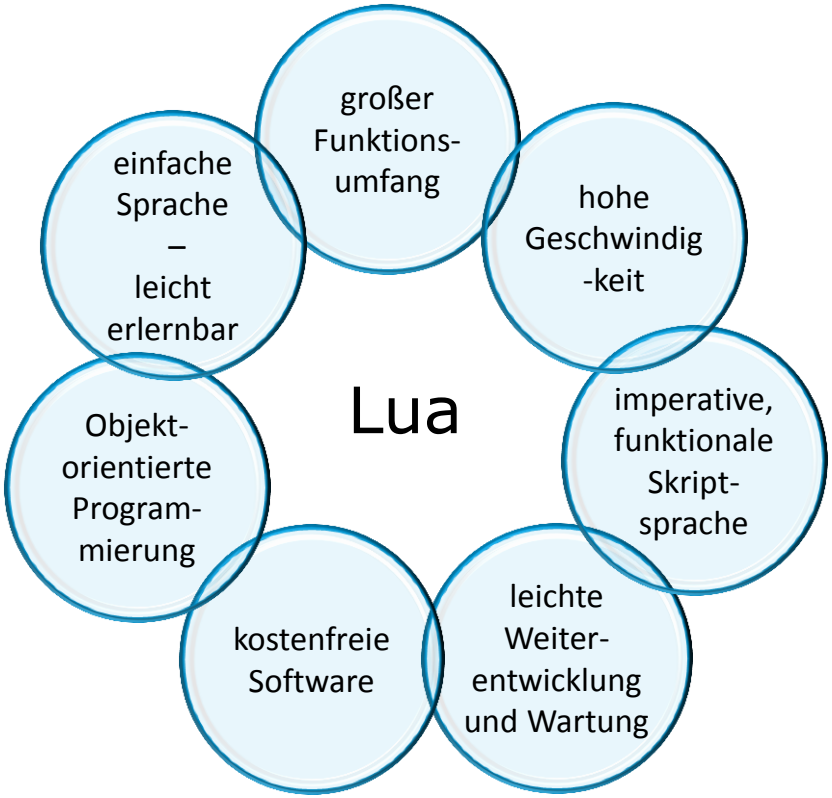
AutomatisierungsTechnik Voigt GmbH

Was ist Lua?

- freie Programmiersprache
- speziell entwickelt für eingebettete Systeme von *Computer Graphics Technology Group* der Päpstlich Katholischen Universität von Rio de Janeiro (Brasilien)

```
for i = 1, 100 do
  xval[i] = i-50

  if (sys.control_var==2) then
    yval[i] = 3 * math.sin(n/5) * math.sin(i/5 + n/3)
  elseif (sys.control_var==3) then
    yval[i] = 3 * math.sin(i/5 + n/3)
  elseif (sys.control_var==4) then
    yval[i] = 3 * math.tan(i/5 + n/3)
  end
end
```

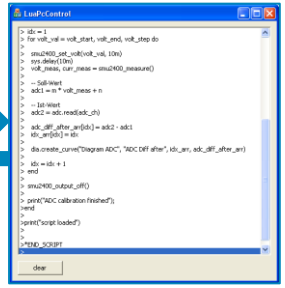


Lua in der Messtechnik

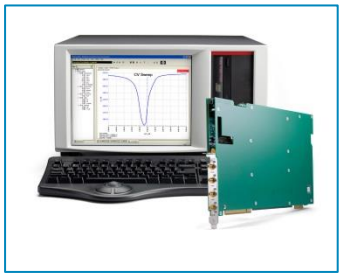
atv AutomatisierungsTechnik Voigt GmbH

Lua Script Editor

TCP/IP



Run Time Engine



Keithley 4200

Lua Script Editor

TCP/IP

MeasLab Applikation

Lua Script Editor

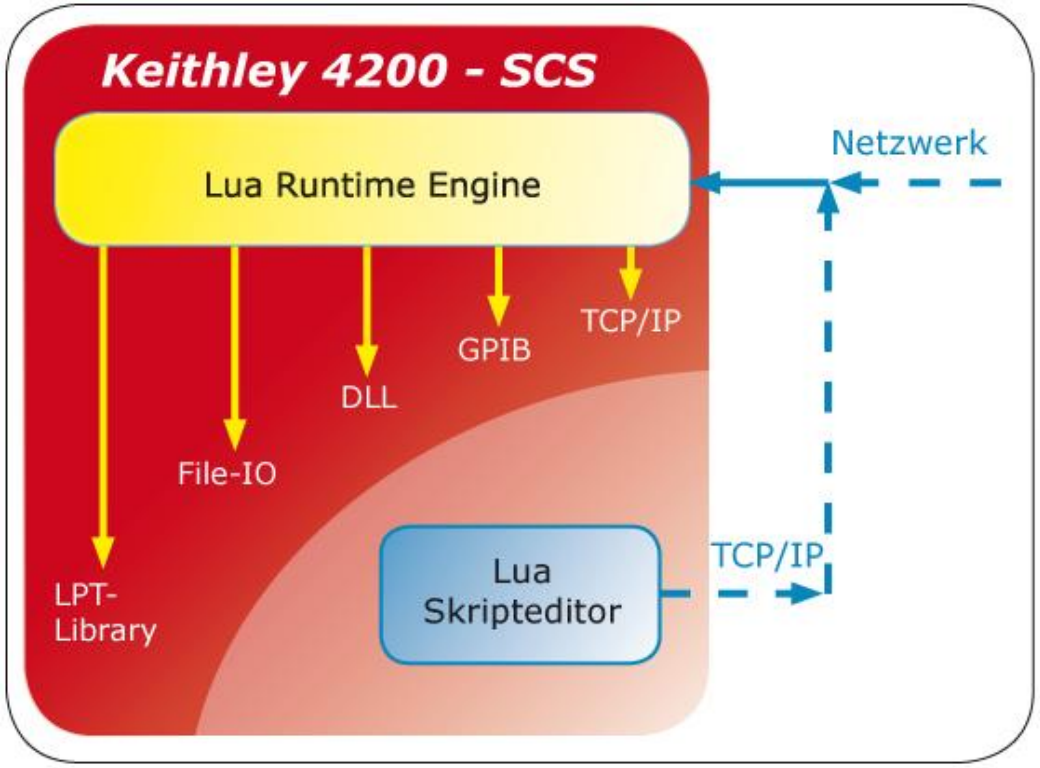
TCP/IP

Keithley 2600; Keithley 3700

Lua in der Messtechnik

AutomatisierungsTechnik Voigt GmbH

- Gerätefirmware selbst programmierbar (z.B. leistungsbegrenzter Sweep)
- Kein C-Programmieren nötig
- Kein Kompilieren mehr nötig
- Kommunikation über Kommandozeile
- Debugging
- Datenvisualisierung
- Integrierte graphische Benutzeroberfläche (GUI)



- TCP/IP – Steuerbarkeit, Aufruf von Funktionen aus dem Skript
- Leichtes Einbinden von externen Geräten
- Volle Anwendungsbreite aller 4200-LPT-Commands

Lua Sprachkonstrukte

AutomatisierungsTechnik Voigt GmbH

- For Schleifen; While Schleifen

```
-- for Schleifen
for i=1,5 do
  print("Hallo welt!")
end

-- While schleifen
local i=0
while i<5 do
  print("Hallo welt!")
  i=i+1
end
```

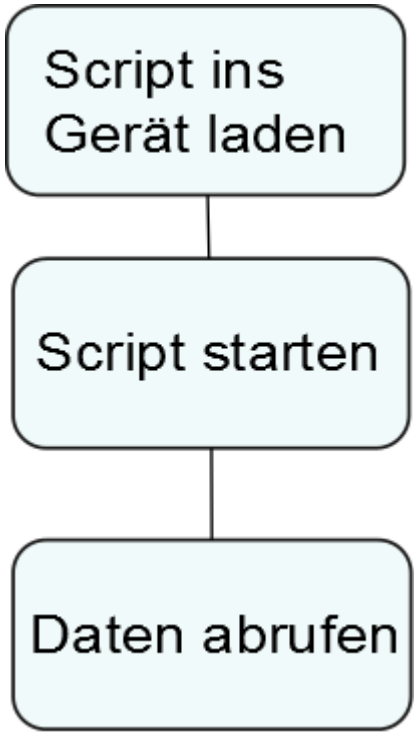
Lua Sprachkonstrukte

AutomatisierungsTechnik Voigt GmbH

- Funktionen, Verzweigungen und Rekursion

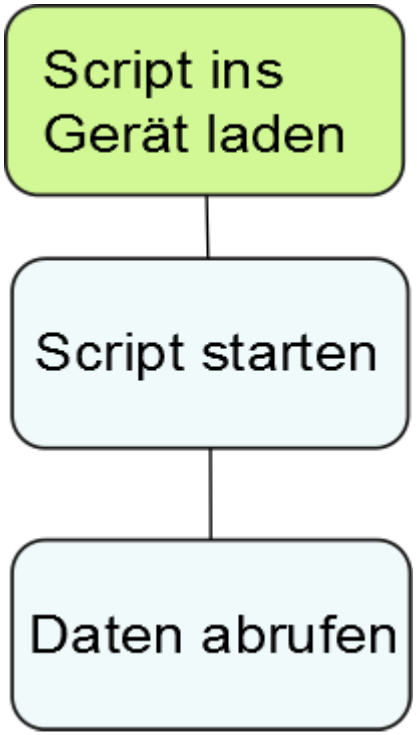
```
-- Berechnung der Fakultät
function factorial(n)
  if n == 0 then
    return 1
  else
    return n * factorial(n - 1)
  end
end
```

Messablauf



- Anbindung über TCP/IP bzw. GPIB
- Messdaten werden auf dem Gerät in Lua gewonnen und mittels selbstdefinierter Protokolle an den PC übertragen

Messablauf



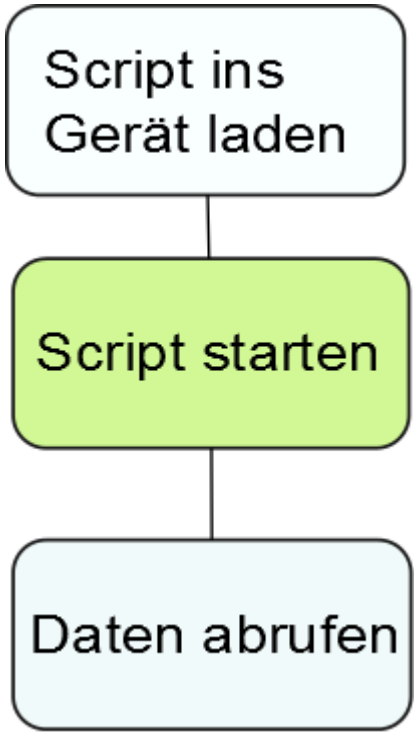

```
function kennlinie1(nsteps)
  smua.source.func   = smua.OUTPUT_DCVOLTS
  smua.source.limiti = 0.01
  smua.measure.nplc  = 0.1
  smua.source.output = smua.OUTPUT_ON

  vmin = -2    -- Minimalspannung
  vmax = 2     -- Maximalspannung
  vs   = (vmax-vmin)/nsteps

  for i=1, nsteps do
    smua.source.levelv = vmin + i*vs
    print(string.format("%d, %2.2f, %2.2f", i,
     smua.source.levelv, math.abs(smu.measure.i())))
  end

  smua.source.output = smu.OUTPUT_OFF
end
```

Messablauf

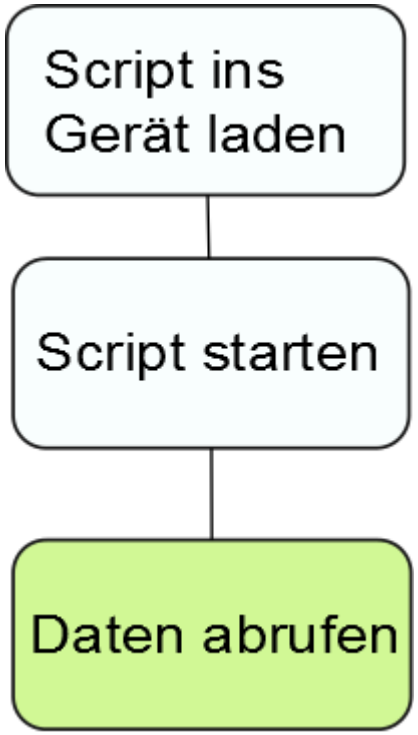


Aufrufen des Skriptes:

"kennlinie1(10)"

via TCP/IP oder GPIB an das Gerät senden

Messablauf



Datenübertragung auf Basis von ASCII, wie im Script definiert:

```
1, -1.60, 0.00
2, -1.20, 0.00
3, -0.80, 0.00
4, -0.40, 0.00
5, 0.00, 0.00
6, 0.40, 0.00
7, 0.80, 0.00
8, 1.20, 0.00
9, 1.60, 0.00
10, 2.00, 0.00
```

10 Zeilen, Werte durch Komma getrennt. Abruf via TCP/IP oder GPIB

Fehlerbehandlung

AutomatisierungsTechnik Voigt GmbH

- Fehler werden in der Fehlerqueue gespeichert.
- Diese sollte regelmäßig abgefragt werden.
- Zum Beispiel mit folgendem Lua-Code:

```
function check_errorqueue()
  while errorqueue.count>0 do
    local errc, msg, sev, node
    errc, msg, sev, node = errorqueue.next()
    print("Fehler: "..msg)
  end
end
```

- Das übergeordnete Programm (z.B. Labview, TestPoint) muss die Daten auf Fehlerzeilen prüfen!

Eingriff in die Ablaufsteuerung

AutomatisierungsTechnik Voigt GmbH

Problem:

Wie greife ich von „außen“ in den Ablauf eines Lua-Skriptes ein?

Bei KEITHLEY Geräten nur indirekt möglich:

- Über eine RS232 Verbindung
- Über Digitalen I/O Kanäle

Bei ATV-Measlab Geräten gibt es eine Kontrollvariable

- Kontrollvariable kann via TCP/IP gesetzt und im Lua-Script abgefragt werden (“*CTRL_VAR 1“)

Eingriff in die Ablaufsteuerung

AutomatisierungsTechnik Voigt GmbH

Beispiel 3: „Start of Test“ über die serielle Schnittstelle

```
function wait_for_start_of_test()
  serial.baud = 9600
  serial.databits = 8
  serial.flowcontrol = serial.FLOW_NONE
  serial.parity = serial.PARITY_NONE

  local cmd = ""
  while (cmd~="start") do
    delay(1)                -- 1 Sekunde warten
    cmd = serial.read(100) -- Daten über RS232 einlesen
  end
end
```

Auswertung von Messdaten

- Daten müssen an den PC Übertragen werden und dort gespeichert werden.

Kontakt

AutomatisierungsTechnik Voigt GmbH

Haben Sie Fragen?

Kontaktieren Sie uns:

Automatisierungstechnik Voigt GmbH
Löbtauer Straße 67
01159 Dresden

Tel.: + 49 351 213 86 40
Fax: + 49 351 213 86 50
E-Mail: atv@atvoigt.de