

Vorwort

Entwicklungen von Geräten zum Messen-Steuern-Regeln und in unserem Fall der Laborautomatisierung zeichnen sich heute durch einen hohen Grad an Komplexität aus.

Die in der Laborautomatisierung eingesetzten Geräteplattformen sind in der Regel komplexe mechatronische Systeme mit zahlreichen Aktoren zur Ausführung von Steuerungsfunktionen, Sensoren zur Prozessüberwachung und integriertem Steuerungsrechner mit grafischem Userinterface (Touchscreen), welche für Labornutzer in den Bereichen Forschung, angewandte Testverfahren und molekularer Diagnostik eine völlig neue Dimension der Probenhandhabung und eine erhebliche Zeit- und Kostenersparnis sowie eine massive Leistungssteigerung erlauben.

Die gesamte Entwicklung solcher Plattformen erfolgt häufig an verschiedenen Standorten. In die Entwicklung waren die folgenden Disziplinen involviert:

- Biologen – für die Entwicklung der zu automatisierenden Prozesse und die Prozessintegration
- Kunststoffingenieure – für die Entwicklung der so genannten Labware (Tips etc.)
- Maschinenbauingenieure – für die Entwicklung der Mechanik der gesamten Geräteplattform
- Elektronikingenieure – für die Entwicklung der Steuerungselektronik incl. Sensorik und Antrieben
- Softwareingenieure – für die Entwicklung der hardwarenahen Schichten der Software
- Softwareingenieure und Informatiker – für die Entwicklung der prozessnahen Schichten (die eigentliche Applikation) und des grafischen User Interfaces (GUI)

Dieses Szenario ist durchaus auf andere Disziplinen erweiterbar und soll die Ausgangssituation einer Entwicklung beschreiben.

Die Basisfunktionalität des zu entwickelnden Geräts wurde in Funktionsmustern und Prototypen implementiert, bevor die Applikation und das GUI für Inbetriebnahme und Tests zur Verfügung standen. Dadurch entstand der Bedarf nach einem Tool, mit dem der Test der Basisfunktionalität möglich wurde.

Am Anfang des Projektes war klar, dass das Gerät neben der eigentlichen Applikation einen einfachen Skript Interpreter haben soll, mit dem beliebige Aktionen ausgeführt werden können.

Um den Entwicklungsaufwand für einen spezifischen Interpreter zu vermeiden, wurde nach verfügbaren Lösungen gesucht. Berichte mit guten Erfahrungen haben uns auf das Ausprobieren eines bereits existierenden, ausgereiften Interpreters neugierig gemacht und der Lua Interpreter wurde versuchsweise in die Low-Level-Testsoftware integriert. Ein Nachmittag reichte, um Lua zu integrieren und das erste Kommando zur Ansteuerung des Geräts zu implementieren.

Nach einigen weiteren Erfahrungen mit der Lua Integration wurde die Idee des spezifischen Interpreters verworfen und ganz auf Lua gesetzt. Dem Test der Low-Level Steuerung des Geräts folgten schnell Lua Anbindungen für weitere Komponenten. Heute haben wir für alle Schichten unterhalb des grafischen User Interfaces (GUI) spezifische Lua Packages.

Später wurde Lua intensiv zur Verifikation einzelner Module und zur Kalibrierung des Geräts eingesetzt.

Um für solche Fälle die Kommandozeilen-Schnittstelle zu verbergen, folgte die Entwicklung einer einfachen grafischen Bibliothek, basierend auf VT102 Kommandos und einer Touchscreen Anbindung. Daraus entstanden weitere Applikationen, die als Fertigungshilfsmittel dienen oder in Test- und Programmiersoftware eingesetzt werden.

Mit dieser Broschüre wollen wir an Hand unserer Erfahrungen die Leistungsmerkmale von Lua verdeutlichen und die Erweiterungsfähigkeit mit zwei Beispielen demonstrieren.

Im ersten Beispiel werden wir den Lua Interpreter auf einem PC mit einer DLL erweitern, die die Ansteuerung eines über USB angeschlossenen AD-DA-Subsystems ermöglicht.

Im zweiten Beispiel werden wir Lua in eine Anwendung auf einem Embedded System auf Basis eines Intel386™ EX Prozessors mit ROM-DOS (kompatibel zu MS-DOS 6.22) einbetten und zeigen, dass Lua auch in Systemen mit knappen Ressourcen eingesetzt werden kann.