



*für Wissenschaft und Technik, für kommerzielle EDV,
für MSR-Technik, für den interessierten Hobbyisten*

In dieser Ausgabe:



AW60 im Seriengerät

MC9S08AW-Flasher

Large Literals for the RP2040
Assembler

Astroimaging in Forth: Image File
Format and Camera SDK

Protokoll der Mitgliederversammlung
Forth-Gesellschaft e.V. am 7.5.2023



Servonaut



Fahrtregler - Lichtanlagen - Soundmodule - Modellfunk

tematik GmbH
Technische
Informatik

Feldstraße 143
D-22880 Wedel
Fon 04103 - 808989 - 0
Fax 04103 - 808989 - 9
mail@tematik.de
<http://www.tematik.de>

Seit 2001 entwickeln und vertreiben wir unter dem Markennamen „Servonaut“ Baugruppen für den Funktionsmodellbau wie Fahrtregler, Lichtanlagen, Soundmodule und Funkmodule. Unsere Module werden vorwiegend in LKW-Modellen im Maßstab 1:14 bzw. 1:16 eingesetzt, aber auch in Baumaschinen wie Baggern, Radladern etc. Wir entwickeln mit eigenen Werkzeugen in Forth für die Freescale-Prozessoren 68HC08, S08, Coldfire sowie Atmel AVR.

Forth-Schulungen

Möchten Sie die Programmiersprache Forth erlernen oder sich in den neuen Forth-Entwicklungen weiterbilden? Haben Sie Produkte auf Basis von Forth und möchten Mitarbeiter in der Wartung und Weiterentwicklung dieser Produkte schulen?

Wir bieten Schulungen in Legacy-Forth-Systemen (FIG-Forth, Forth83), ANSI-Forth und nach den neusten Forth-200x-Standards. Unsere Trainer haben über 20 Jahre Erfahrung mit Forth-Programmierung auf Embedded-Systemen (ARM, MSP430, Atmel AVR, M68K, 6502, Z80 uvm.) und auf PC-Systemen (Linux, BSD, macOS und Windows).

Carsten Strotmann carsten@strotmann.de
<https://forth-schulung.de>

RetroForth

Linux · Windows · Native
Generic · L4Ka::Pistachio · Dex4u
Public Domain
<http://www.retroforth.org>
<http://retro.tunes.org>

Diese Anzeige wird gesponsort von:
EDV-Beratung Schmiedl, Am Bräuweiher 4,
93499 Zandt



Cornu GmbH
Ingenieurdienstleistungen
Elektrotechnik

Weitlstraße 140
80995 München
sales@cornu.de
www.cornu.de

Unser Themenschwerpunkt ist automotive SW unter AutoSAR. In Forth bieten wir u. a. Lösungen zur Verarbeitung großer Datenmengen, Modultests und modellgetriebene SW, z. B. auf Basis eCore/EMF.

KIMA Echtzeitsysteme GmbH

Güstener Straße 72 52428 Jülich
Tel.: 02463/9967-0 Fax: 02463/9967-99
www.kimaE.de info@kimaE.de

Automatisierungstechnik: Fortgeschrittene Steuerungen für die Verfahrenstechnik, Schaltanlagenbau, Projektierung, Sensorik, Maschinenüberwachungen. Echtzeitrechnersysteme: für Werkzeug- und Sondermaschinen, Fuzzy Logic.

FORTEch Software GmbH

Tannenweg 22 m D-18059 Rostock
<https://www.fortech.de/>

Wir entwickeln seit fast 20 Jahren kundenspezifische Software für industrielle Anwendungen. In dieser Zeit entstanden in Zusammenarbeit mit Kunden und Partnern Lösungen für verschiedenste Branchen, vor allem für die chemische Industrie, die Automobilindustrie und die Medizintechnik.

Ingenieurbüro Tel.: (0 82 66)-36 09 862
Klaus Kohl-Schöpe Prof.-Hamp-Str. 5
D-87745 Eppishausen

FORTH-Software (volksFORTH, KKFORTH und viele PD-Versionen). FORTH-Hardware (z. B. Super8) und Literaturservice. Professionelle Entwicklung für Steuerungs- und Messtechnik.

Mikrocontroller-Verleih Forth-Gesellschaft e. V.

Wir stellen hochwertige Evaluation-Boards, auch FPGA, samt Forth-Systemen zur Verfügung: Cypress, RISC-V, TI, MicroCore, GA144, SeaForth, MiniMuck, Zilog, 68HC11, ATMEL, Motorola, Hitachi, Renesas, Lego ...
<https://wiki.forth-ev.de/doku.php/mcv:mcv2>

Leserbriefe und Meldungen	5
AW60 im Seriengerät	9
<i>Rafael Deliano</i>	
MC9S08AW-Flasher	12
<i>Rafael Deliano</i>	
Large Literals for the RP2040 Assembler	15
<i>Willem Ouwerkerk</i>	
Astroimaging in Forth: Image File Format and Camera SDK	18
<i>Andrew Read</i>	
Protokoll der Mitgliederversammlung Forth-Gesellschaft e.V. am 7.5.2023	22
<i>Anton Ertl</i>	

Titelbild: Geminidenstrom *AUTOR:* mk

Quelle: Internet, selbst bearbeitet

Impressum

Name der Zeitschrift
Vierte Dimension

Herausgeberin

Forth-Gesellschaft e. V.
Postfach 1030
48481 Neuenkirchen
E-Mail: Secretary@forth-ev.de
Direktorium@forth-ev.de
Bankverbindung: Postbank Hamburg
BLZ 200 100 20
Kto 563 211 208
IBAN: DE60 2001 0020 0563 2112 08
BIC: PBNKDEFF

Redaktion & Layout

Bernd Paysan, Ulrich Hoffmann
E-Mail: 4d@forth-ev.de

Anzeigenverwaltung

Büro der Herausgeberin

Redaktionsschluss

Januar, April, Juli, Oktober jeweils
in der dritten Woche

Erscheinungsweise

1 Ausgabe / Quartal

Einzelpreis

4,00€ + Porto u. Verpackung

Manuskripte und Rechte

Berücksichtigt werden alle eingesandten Manuskripte. Leserbriefe können ohne Rücksprache wiedergegeben werden. Für die mit dem Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung. Die in diesem Magazin veröffentlichten Beiträge sind urheberrechtlich geschützt. Übersetzung, Vervielfältigung, sowie Speicherung auf beliebigen Medien, ganz oder auszugsweise nur mit genauer Quellenangabe erlaubt. Die eingereichten Beiträge müssen frei von Ansprüchen Dritter sein. Veröffentlichte Programme gehen — soweit nichts anderes vermerkt ist — in die Public Domain über. Für Text, Schaltbilder oder Aufbauskiizen, die zum Nichtfunktionieren oder eventuellem Schadhafwerden von Bauelementen führen, kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Liebe Leser,

die Geminiden sind mit der stärkste Meteorstrom des Jahres mit einem Maximum Mitte Dezember. Sein Radiant, das ist der Punkt, aus dem die Sternschnuppen herzukommen scheinen, liegt knapp nördlich des hellen Sterns Castor im Sternbild Zwillinge (lateinisch gemini), das den Geminiden den Namen gab.

Unser Forth-Magazin hat jetzt auch einen Zwilling. WOLFGANG STRAUSS hat es komplett übersetzt ins Englische. Daran sind viele Hoffnungen geknüpft. Wir kommen unserer internationalen Leserschaft entgegen.

Und hoffen, dass die Beiträge so besser übersetzt sind, als es die verschiedenen Automaten vermögen — die haben dann noch genug zu tun, das aus dem Englischen in all die anderen Sprachen dieser Welt zu bringen.

Bleiben wir bei den Sternen. ANDREW READ gewährt weiteren Einblick ins Astroimaging. Diesmal geht es um die Ansteuerung der Kamera, die ja nicht bei ihm zu Hause ist, sondern weiter weg, da wo der Himmel sternenklar ist und ohne Lichtverschmutzung durch die vielen Lichter der Zivilisation. Falls ihr mal dort seid, lasst es uns wissen.

WILLEM OUWERKERK gibt Einblick in die andere Richtung, ins ganz Kleine, das man nur noch mit einem Mikroskop sehen kann, die Register im RP2040. Diese effizient für *noForth T* einzusetzen, ist sein Ziel. Ja, richtig: „T“ — *noForth* erobert den M0+-Kern, und kommt mit einem *forthigen* Thumb2-Assembler daher.

Ob im Kleinsten oder ganz groß im All, gesteuert wird immer. RAFAEL DELIANO ist gleich mit zwei Beiträgen im Heft dabei. Und lässt uns teilhaben an der Entwicklung, Produktion und Qualitätskontrolle solcher Bauteile, *forth-style*. Prototyping und Kleinserien gehen damit eben besonders gut.

Wenn auch auf den letzten Drücker: Die Ankündigung des Forth-Sommertreffens hat es noch auf die Hefrückseite geschafft. Ihr seid schon alle angemeldet? Bestens. Dann freue ich mich auf ein Wiedersehen demnächst im Linux Hotel.

Euer Michael



Die Quelltexte in der VD müssen Sie nicht abtippen. Sie können sie auch von der Web-Seite des Vereins herunterladen.

<http://fossil.forth-ev.de/vd-2023-02>

Die Forth-Gesellschaft e. V. wird durch ihr Direktorium vertreten:

Ulrich Hoffmann Kontakt: Direktorium@Forth-ev.de
Bernd Paysan
Gerald Wodni

Scharlatane, jetzt auch digital

KI ist in aller Munde, ein riesiger Hype. Angeregt von ULLI HOFFMANN und seinen Erfahrungen konnte ich nicht widerstehen, und habe *ChatGPT* auch mal selber ausprobiert. Ich zitiere:

Forth's Future Outlook:

user: Programmiersprache Forth. Zukunftsaussichten?

assistant:

Forth ist eine Programmiersprache, die bereits seit den 1970er Jahren besteht und insbesondere in der Embedded-System-Entwicklung und in der Steuerungstechnik eingesetzt wird. Es gibt nach wie vor eine aktive Gemeinschaft von Forth-Entwicklern und -Anwendern, die die Sprache weiterentwickeln und pflegen.

Allerdings hat Forth in den letzten Jahren an Bedeutung verloren, da andere Programmiersprachen wie Python, JavaScript, Java und C++ immer populärer wurden. Dies liegt unter anderem daran, dass diese Sprachen besser an moderne Entwicklungsumgebungen, Bibliotheken und Frameworks angepasst sind und eine größere Gemeinschaft von Entwicklern und Anwendern haben.

Das bedeutet jedoch nicht, dass Forth keine Zukunft hat. Es gibt immer noch spezialisierte Anwendungen, in denen Forth aufgrund seiner Effizienz und Flexibilität Vorteile gegenüber anderen Sprachen hat. Außerdem gibt es immer noch eine aktive Gemeinschaft von Forth-Entwicklern, die die Sprache weiter verbessern und pflegen.

Insgesamt würde ich sagen, dass Forth in absehbarer Zeit keine große Verbreitung mehr finden wird, aber in spezialisierten Anwendungen weiterhin eine wichtige Rolle spielen wird.

Super. Besser hätte ich es nicht selbst formulieren können, noch dazu in perfektem Deutsch. Das weckt doch Vertrauen in diese Technik.

Da ich immer viel Zeit mit der Suche nach geeigneten Bauelementen für meine Sachen verbringe, frag ich doch gleich mal die KI:

SMT MOSFET with Vgs

user: find a small SMT N-channel MOSFET with an Vgs of at least 30 V

assistant:

One example of an SMT (Surface Mount Technology) N-channel MOSFET with a Vgs of at least 30 V is the 2N7002LT1G MOSFET manufactured by ON Semiconductor.

Some of its specifications are:

- Drain-Source Voltage (Vdss): 60 V
- Gate-Source Voltage (Vgs): ±20 V
- Continuous Drain Current (Id): 330 mA

- Power Dissipation (Pd): 200 mW — Package: SOT-23

Please note that there may be other MOSFETs available with similar specifications, and it is important to choose a MOSFET that is appropriate for your specific application.

user: Wrong. The Vgs should be greater than ±30 V.

assistant: I apologize for the mistake in my previous answer. To find an SMT N-channel MOSFET with a Vgs of at least ±30 V, you can consider the following options:

1. Vishay Siliconix SiB121EDJ-T1-GE3 MOSFET: This MOSFET has a Vgs of ±40 V and a package size of SOT-23. Some of its key specifications are:

- Drain-Source Voltage (Vdss): 60 V
- Continuous Drain Current (Id): 4 A
- Rds(on): 48 mOhms

2. Diodes Incorporated DMT6008LPS MOSFET: This MOSFET has a Vgs of ±30 V and a package size of PowerDI 5x6. Some of its key specifications are:

- Drain-Source Voltage (Vdss): 60 V
- Continuous Drain Current (Id): 3.8 A
- Rds(on): 25 mOhms

Please note that there may be other MOSFETs available with similar specifications, and it is important to choose a MOSFET that is appropriate for your specific application.

user: any smaller parts

assistant: If you are looking for an SMT N-channel MOSFET with a Vgs of at least ±30 V in a smaller package, you can consider the following options:

1. ON Semiconductor NTLUD3317PZTBG MOSFET: This MOSFET has a Vgs of ±30 V and a package size of SOT-563. Some of its key specifications are:

- Drain-Source Voltage (Vdss): 20 V
- Continuous Drain Current (Id): 0.4 A
- Rds(on): 700 mOhms

2. Diodes Incorporated DMN61D0LQ MOSFET: This MOSFET has a Vgs of ±30 V and a package size of SOT-23-3L. Some of its key specifications are:

- Drain-Source Voltage (Vdss): 60 V
- Continuous Drain Current (Id): 0.3 A
- Rds(on): 2.8 Ohms

Please note that these MOSFETs have lower maximum drain-source voltage ratings and current ratings than the MOSFETs I mentioned in my previous answer. It is important to choose a MOSFET that is appropriate for your specific application and meets the necessary requirements.

Ja Wahnsinn! Was für eine Arbeitszeiterparnis!

Allerdings — nicht wirklich . . . Einen 2N7002LT1G gibt es tatsächlich. Wer allerdings versucht, die anderen genannten Typen einzukaufen, erlebt eine Überraschung: Sie existieren nicht. Die Transistoren sind frei erfunden. Wobei „erfinden“ möglicherweise das falsche Wort ist. Fachleute sprechen im Zusammenhang mit einer KI eher von „halluzinieren“¹. Und diese digitalen Halluzinationen sind, anders als in diesem Fall, oft nur schwer zu erkennen, denn alles erscheint total plausibel, auch die Typenbezeichnungen entsprechen dem Usus des jeweiligen angeblichen Herstellers.

In diesem Zusammenhang ist das verblüffend, ich war zunächst einmal sprachlos. In anderen Zusammenhängen dürfte das aber brandgefährlich sein.

„Scharlatane verschmelzen die Wahrheit mit Lügen zu einem Wahrheitsersatz.“

Das ist die beste Definition, die ich im Netz gefunden habe. Und perfekt zutreffend auf das, was die KI hier ist, ein *digitaler Scharlatan*.

Schöne neue Welt.

Jörg Völker

Write a “Hello World” Programm in Forth

Sure! Here’s an example of a “Hello World” program written in Forth:

```
." Hello World!" CR
```

Explanation: `."` is a Forth word that prints a string to the output. `"Hello World!"` is the string that we want to print. `CR` is a Forth word that outputs a carriage return, which moves the cursor to a new line. When you run this Forth program, it will print `"Hello World!"` to the output with a new line, typically on the terminal or console where you are running the Forth interpreter or compiler.

Note: The syntax and functionality of Forth can vary depending on the implementation and system you are using. This example assumes a standard Forth environment with a period (`.`) as the word separator and a carriage return (`CR`) as the newline character. Please refer to the documentation or user guide of your specific Forth system for more information on how to write and execute Forth programs.

Das antwortete mir ChatGPT neulich. Wie es darauf kommt, dass der Worttrenner ein Punkt sei, kann man nicht nachvollziehen, ist es doch das Leerzeichen.² Sei’s drum. Das Progrämmchen ist richtig geworden. Die Erklärung dazu auch. Als Kundiger bügelt man die beiden

¹ Unter Halluzination (von lateinisch *alucinatio* „Träumerei“) versteht man eine Wahrnehmung, für die keine nachweisbare externe Reizgrundlage vorliegt.

² Unter Konfabulation bzw. konfabulieren (wie lat. *confabulatio*, „Erzählung“, von *fabula*, „Fabel“, „Geschichte“, „Märchen“) versteht man in der Psychopathologie die Produktion objektiv falscher Erinnerungen, z. B. wenn jemand mehr Informationen aus seinem Gedächtnis abzurufen versucht, als tatsächlich gespeichert sind (sog. provozierte Konfabulationen).

³ ARM, Thumb-2

Fehler dann einfach aus und hat schwupps einen Zeitungsartikel verfasst.

Anmerkung: Wer nicht glaubt, dass das von ChatGPT gekommen ist: Den Screenshot hab ich aufgehoben. Doch weil der in heller Schrift auf dunklem Grund kam, was in unserem Blatt schlecht zu drucken gewesen wäre, gibt es den Text hier als Zitat. Gut, dass wir immer den Autor eines Beitrags angeben, und Quellen. So soll es auch bleiben. mk

Neu: noForth T

Im Mai schickte WILLEM OUWERKERK einen interessanten Beitrag für unser Magazin — habt ihr bestimmt schon entdeckt im Inhaltsverzeichnis. Dabei kam heraus, dass noForth eine weitere häufig verwendete MCU erobert hat. Er schreib:

„It will be available soon on the noForth web-site! In the meantime, the pre-release version is available when people send me an email. Then I will send them the ZIP of the pre-release version to test.

This pre-release contains the tools, an assembler and disassembler, in addition a PIO assembler and disassembler, configuration files, commacode, etc.“

Möge das noForth T-Modell³ so erfolgreich werden wie einst das T-Modell für Ford, die Tin Lizzie. mk

<https://home.hccnet.nl/ani/nof/noforth.html>

VCFe — Vintage Computer Festival Europa

Zum 22. Mal war das VCF am Wochenende vom 29.04–01.05 2023 in unserem schönen München. Auf <https://www.vcfe.org/D/> könnt ihr es nachlesen:

„Das VCF ist nicht nur im kalifornischen Silicon Valley ein regelmäßiges Ereignis, sondern inzwischen auch ein fixer Punkt in den europäischen Terminplänen.

Also lasst uns zurückkehren in die Guten Alten Tage, als Hacker noch keine Sicherheitsberater, Bytes noch keine Megabytes und Kleine Grüne Männchen noch *Kleine Gruene Maennchen* waren!“

Das VCFe widmet sich in jedem Jahr einem Schwerpunktthema. Diesmal war es: „Silicon Mountains — Computer entlang der Alpen.“ Von *ERMETH*, *Smacky* und *Lilith* im Westen bis zu *Mailüfterl* und *Mupid* im Osten und bedeutenden Herstellern der ersten Stunde, wie *Siemens* im Norden und *Olivetti* im Süden ging die Reise, und zu den grundlegenden Konzepten von *ALGOL* bis *Stack*.

RAFAEL DELIANO hat sich dort umgesehen und etwas für unser Magazin herausgepickt.

Lilith

Er schrieb mir:

„Einen vagen Bezug zu Forth hätte dieser Oldtimer gehabt: *Lilith*, die europäische Alto⁴. Die Lilith wurde als hochperformanter Einzelplatzrechner mit hochauflösender Grafik und Mausbedienung ab 1980 an der ETH Zürich entwickelt. Die gesamte Software ist in der Hochsprache Modula-2 geschrieben. Hard- und Software wurden entwickelt von einem Team um NIKLAUS WIRTH. Die Ausstellung zeigte eine originale Lilith sowie den Emulator *Emulith*, mit dem man die originale Betriebssoftware ausprobieren konnte — Schwerpunktthema von JOS DREESEN.

Die Hardware war ein „2900“ Bit-Slice-Rechner, der als Stack-Computer ausgeführt war. Allerdings war die Zwischensprache M-Code.“

http://www.bitsavers.org/pdf/eth/lilith/Lilith_hardware_manual.pdf



Abbildung 1: Lilith

Z79Forth

„Interessant dann auch ein *Z79Forth*. Leider ist der Aussteller FRANCOIS LAAGEL nicht aufgetaucht oder ich habe ihn übersehen. Daher nur ein Zitat: ‚*Z79Forth* is a journey into retro-computing using a Hitachi 63C09, but takes advantage of modern technology where appropriate (CMOS, USB and CompactFlash). Wire wrapping is particularly adapted to digital hardware experimentation and provides the developer with a great deal of flexibility.‘“

<https://www.elektormagazine.de/labs/72832/francois-laagel>

⁴ Der Xerox Alto wurde 1973 am Forschungszentrum Xerox PARC entwickelt und war die erste Workstation mit einer grafischen Benutzeroberfläche.

⁵ In den vergangenen 12 Monaten war das ich

Mal sehen, ob wir von Laagel mehr darüber erfahren können. mk

Erratum

Francois Laagel mailte an uns im Mai diesen Jahres:

„I noticed an overview of Forth systems in VD 4/2022 and would like to contribute a few fixes to the entry labelled as ‘Zforth.’

Name: Z79Forth

Author: Francois Laagel

Standard: 79 or ANS94“

Liebe Leser, bitte nehmt diese Richtigstellung in eure Liste auf. Und meldet uns gerne, wenn ihr da in der Liste Fehler entdeckt. mk

Drachenumzug

SWAPPY, unser aller Lieblingsdrache, liebt es ja, seinem Hüter⁵ über die Schulter zu schauen und neue Sachen über Forth, und ggf. auch über anderes, zu lernen. Allerdings kommt dann immer auch irgendwann der Punkt, wo er meint, alles gesehen zu haben, was es an seiner aktuellen Station so zu sehen gibt. Mir fiel entsprechend in der letzten Zeit auf, dass er irgendwie unruhig wurde, es fing offensichtlich an, ihn wieder in den Klauen und Flügeln zu jucken.

Und so rief er, wie es alljährlich seine Gewohnheit ist, am Samstagabend des Forthtreffens seinen Dracherrat zusammen, um sich mit ihm über sein nächstes Reiseziel zu beraten. Am Ende der Beratungen entschied sich Swappy, unter zustimmendem Gemurmel seines Rates, dass er das nächste Jahr bei WILLEM OUWERKERK in Arnheim verbringen will. Er hatte viel Gutes über Willem gehört, ihn öffentlich in Sachen Forth in Erscheinung treten gesehen, hatte von seinen tanzenden Roboterarmen gehört, und auf noForth ist er natürlich auch neugierig.



Abbildung 2: Swap in seiner Reisekiste, abreisefertig

Also war es an mir, seine wunderbare und fast neue, bisher erst einmal genutzte Reisehöhle hervorzuholen, und

Swappy behilflich zu sein, darin Platz zu nehmen. Für eine kurze Weile konnten wir uns beide nicht erinnern, wie herum Swappy in seinem Drachensitz gelegen hatte, so dass der Sicherheitsgurt gut sichert, und der Sitz guten Halt gibt. Aber dann bemerkten wir, dass Swappy mit seinem Rückenramm gut sichtbare Eindrücke auf dem Sitz hinterlassen hatte, und mit der dann wieder einsetzenden Erinnerung war uns dann alles wieder klar, Swappy nahm Platz, und ich legte ihm seinen Gurt an. Nochmal ein Zwinkern mit dem rechten Auge des rechten Kopfes, Kapuze übergezogen, Schatzkiste verstauen, Höhle verschlüsseln, und los — so dachte ich. Aber dann fand Swappy, daß sein Hort doch ein bisschen zu sehr rappelte, wenn ich die Reishöhle bewegte; so wollte er dann doch nicht fahren. Also — was tut man nicht alles für einen Drachen — nochmal Höhle auf, Papier geholt, Schatzkiste raus, Schatzkiste auf, etwas Papier 'rein, Schatzkiste zu, Schatzkiste verstaut, Papier drumherum gepackt, Höhle zu, Schlüssel unter die Fußmatte, Paket zu, Zielkoordinaten bereitgelegt, und ab zur nächsten Postfiliale. Swappy hat ähnliche Reisepräferenzen wie Iwans, wenn man TERRY PRATCHETTS „Thief of Time“ glauben darf.

Ich bin gespannt, was Swappy von seinem neuen Domizil und Wirkungskreis berichten wird.

PS: Das Suchen nach der richtigen Sitzposition hätte ich Swappy und mir ersparen können, wenn ich meinen eigenen Bericht in der 4d2022-02 früher nachgelesen hätte. :-)

Philip Zembrod

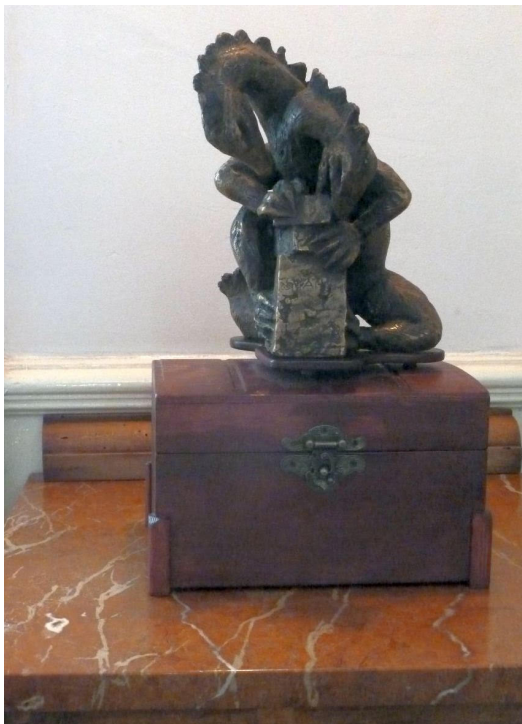


Abbildung 3: Swappy at Willem's home

Drachenankunft

Yes, the swap dragon arrived in very healthy condition. Despite the somewhat long journey for him (he had to spend a few days in a depot). Beautifully wrapped and on top a nice filled treasure chest. I am very happy that he gets to live here for a year. He also got a very nice spot as shown in the picture. (Abb. 3) Willem Ouwerkerk

VD-Team sucht internationale Verstärkung

Auf der Mitgliederversammlung 2023 haben wir beschlossen, neben der deutschen Ausgabe unserer Vereinzeitschrift „Vierte Dimension“ auch eine internationale Ausgabe in Englisch herauszubringen. Wir versprechen uns davon eine Belebung der Forthaktivitäten und den Ausbau der Vernetzung von Forthern rund um den Erdball.

Das Projekt ist zunächst als ein Test für einen Zeitraum von 4 Ausgaben angelegt. Danach wird sich entscheiden, ob das Vorhaben fortgeführt wird. Auf Dauer kann die doch erhebliche Mehrbelastung von dem kleinen deutschen Team nicht alleine gestemmt werden.

Wir suchen deshalb nach Mitwirkenden für verschiedene Bereiche:

- englische Muttersprachler (Proofreader)
- Übersetzer Deutsch ↔ Englisch
- Übersetzer Englisch ↔ Deutsch
- Autoren Englisch
- Autoren Deutsch
- Maker, die Hardware-Projekte in Forth programmieren

Du bist nicht sicher, ob Du in eine dieser Kategorien passt? Wir sollten miteinander sprechen. Es wird sich eine interessante Zusammenarbeit finden.

Die Zeitung „Vierte Dimension“ ist ein Hobbyprojekt und wird ehrenamtlich erstellt. Also keine Sorge: Wir erwarten keine Vollprofis mit mehrjähriger Berufserfahrung. Begeisterung für die Programmiersprache Forth und solide Kenntnisse in Englisch oder/und Deutsch reichen völlig aus.

Interessenten wenden sich bitte unter der E-Mail-Adresse vd@forth-ev.de an das Team der Vierten Dimension. Auch Fragen und Kritik sind immer willkommen.

Wolfgang Strauß

Fortsetzung der Leserbriefe auf Seite 11

AW60 im Seriengerät

Rafael Deliano

Wegen der weiterhin schlechten Liefersituation bei Controllern musste der MC9S08AW60 noch einmal für die Produktion verwendet werden. Er ist bereits obsolet, aber als ehemaliger automotiver Standardtyp besser verfügbar als das, was die Hersteller anpreisen.

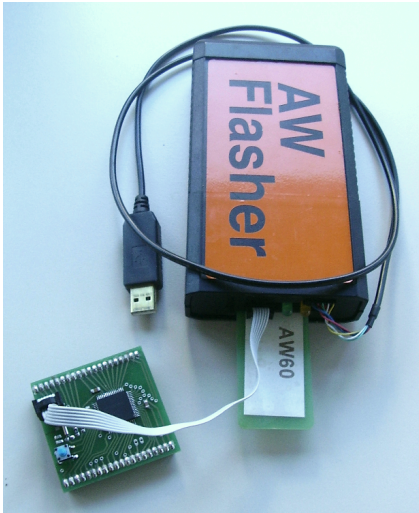


Abbildung 1: Flasher mit AW60 SBC

Wegen der überschaubaren Stückzahlen sollte dafür das vorhandene Programmiergerät unverändert verwendet werden (Abb. 1). Eine Leiterplatte mit dem AW60 enthält normalerweise die von Motorola definierte 6-polige Stiftleiste (Abb. 2, 3). Diese BGD-Schnittstelle¹ ist fürs Flashen des Controllers erforderlich.

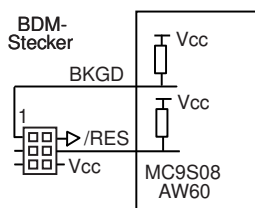


Abbildung 2: Schaltung BDM

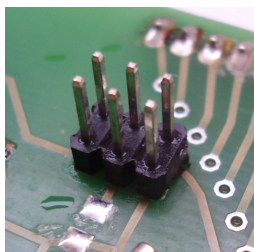


Abbildung 3: BDM als 6-pol. Stiftleiste auf SBC

¹ Background Debug

² Single Board Computer

Um Materialkosten zu sparen, und auch wegen der Bauhöhe, wurde keine Stiftleiste, sondern ein abgesägter 2,54-mm-Edge-Connector verwendet (Abb. 4); eine gängige Alternative wären gefederte Prüfstifte. Es gibt auch Klammern aus China via ebay.com (Abb. 5) mit unterschiedlicher Pinzahl und in verschiedenen Rastern.

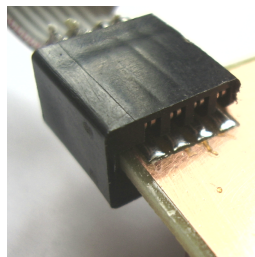


Abbildung 4: 8-poliger Edge-Connector

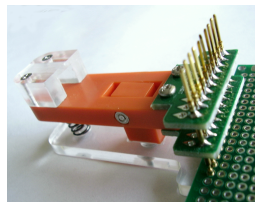


Abbildung 5: 10-polige Pogo-Klammer aus China

BDM erweitert

Durch mehr Leitungen werden zusätzliche Funktionen verfügbar. RXD/TXD der UART sind das Minimum. Über die Kommandozeile von FORTH kann man die Leiterplatte bequem testen. Der Pin /EN stoppt den störenden externen Watchdog. Statt eines Quarz-Oszillators führt ein weiterer Pin einen 32-kHz-Takt zu. Aus dem wurde auf dem AW60-SBC² der 4,19-MHz-Bustakt erzeugt. Für diesem Modus initialisiert auch die nanoFORTH-Firmware den Controller — eigentlich. Hier soll

aber der interne RC-Takt verwendet werden, um Bauteile zu sparen.

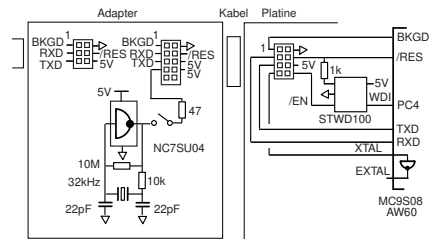


Abbildung 6: Schaltung mit Adapterplatine

RC-Takt abgleichen

Durch den 32-kHz-Oszillator auf der Adapterplatine im Kabel (Abb. 6) ist nach dem Flashen von FORTH die Kommandozeile über UART verfügbar. Man kann nun vom Terminalemulator aus ein Programm hochladen (Listing 1), das ins RAM compiliert wird — der Vorteil der Von-Neumann-CPU. Die Routine RXD# ist ein Zähler, der die Breite eines Low-Pulses im ASCII-Zeichen U (55h) bestimmt; hier war das der Wert 33h. D.h., die Auflösung ist niedrig, denn es ist ein langsamer Controller — der Nachteil der Von-Neumann-CPU.

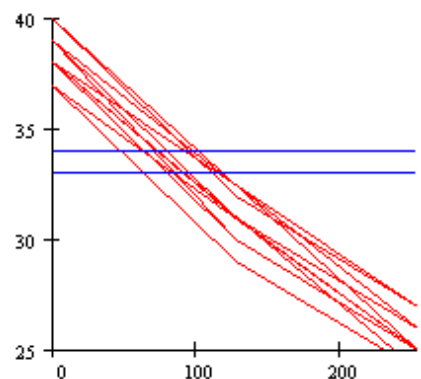
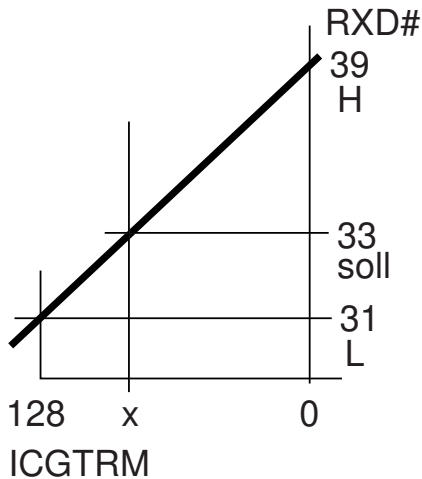


Abbildung 7: Streuung des RC-Taktes

Der Trimfaktor ICGTRIM des RC-Taktes ist zwischen 00...FF einstellbar. Die Kennlinie ist etwas durchhängend (Abb. 7). Hier wird der Zielwert deshalb nicht aus der Geraden von 00 nach FF, sondern von 00 nach 80h berechnet (Abb. 8). Der Zielwert bei 70h wird ins EEPROM gespeichert. Im zweiten Schritt wird dann nanoFORTH so gepatcht, dass der AW60 nach dem Reset mit dem RC-Oszillator startet (Listing 2).



$$x = 128 - \frac{128 * (soll - L)}{(H - L)}$$

Abbildung 8: Berechnung der Korrektur

FTDI

An PCs ist die für Terminalprogramme übliche RS232-Schnittstelle selten geworden. Der Ersatz durch USB³ in Form eines FTDI-Kabels⁴ hat den Vorteil, dass nun die 5-V-Versorgung bereits aus dem PC

Listing 1

```

1 <| \ calc TRIM and store in EEPROM
2
3 FFFF 812E E! \ erase EEPROM
4
5 FCODE >C ! \ compile to RAM
6
7 :CODE RXD# \ ( -- UC1 )
8 N CLR,
9 1 $:
10 1 $ PE 1 BBS, \ wait RXD low
11 3 $:
12 2 $ PE 1 BBS, \ 5cyc
13 N INC, \ 1
14 3 $ BNE, \ 3
    
```

³ Universal Serial Bus

⁴ Future Technology Devices International

möglich wird. Der Nachteil für 5-V-Controller ist, dass RXD/TXD auf 3,3V ausgelegt sind. Da für 5-V-Controller ein Pegelumsetzer nötig wird, hat es sich bewährt, einen ADUM1301 zu verwenden, der für die galvanische Trennung sorgt (Abb. 9). Damit ist man auch nicht mehr über den PC mit dem Schutzleiter verbunden und hat so eine Störquelle, die analoge Messungen erschwert, eliminiert. Die üblichen Nachteile von FTDI-Kabeln sind, dass die billigen nicht funktionieren. Das betrifft besonders den XON/XOFF-Handshake, wie er vom nanoFORTH benötigt wird. Die teuren 5-EUR-Deek-Robot (Abb. 10) aus China haben sich aber bewährt. Die Steckerbelegung ist jedoch nicht einheitlich. Die hier verwendete Hausnorm (Abb. 11) ist nur eine von vielen möglichen. Immerhin kann man die Kontakte an den Dupont-Buchsen leicht umstöpseln. Der Stecker ist nicht verpolungssicher. Übliche Lösung: unbenutzten Pin am Stecker abschneiden und seine Buchse verstopfen.

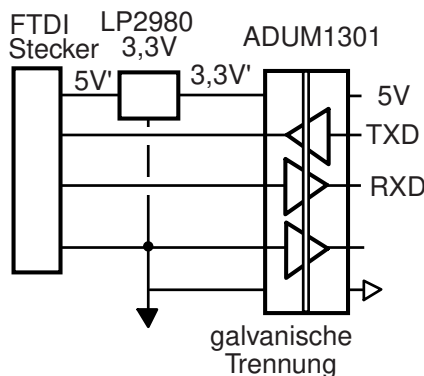


Abbildung 9: Pegelwandler



Abbildung 10: Deek-Robot-Stecker am FTDI-Kabel

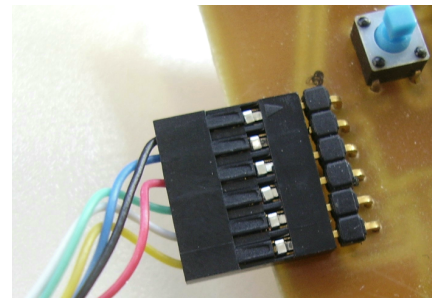


Abbildung 11: FTDI-Dupont-Verbindung

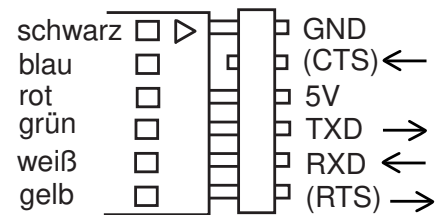


Abbildung 12: FTDI-Verbindungs-Schema

```

15 N FF #. MOV,
16 2 $: DEX,
17 N LDA,
18 0,X STA,
19 DEX,
20 0,X CLR,
21 RTS,
22 CODE;
23
24 :CODE CLK=RC \ switch clock to RC
25 2 $:
26 48 B% 00101000 #. MOV, \ ICGC1
27 49 B% 01010010 #. MOV, \ ICGC2
28 1 $: 1 $ 4A 3 BBC,
29 48 LDA,
30 B% 00101000 #. CMP,
    
```



```

31 2 $          BNE,
32             RTS,
33 CODE;
34
35 :CODE CLK=32kHz \ switch clock to 32k
36 2 $:
37 48 B% 00111000 #. MOV, \ ICGC1
38 49 B% 00000000 #. MOV, \ ICGC2
39 1 $: 1 $     4A 3 BBC,
40             48 LDA,
41             38 #. CMP,
42 2 $          BNE,
43             RTS,
44 CODE;
45
46 : (R) \ ( UC1 -- UC2 )
47 CR DUP CH. 4E C!
48 ." press U " D% 100 MSEC
49 CLK=RC D% 100 MSEC RXD# TERMINAL?
50 DROP CLK=32kHz D% 1000 MSEC ;
51
52 \ Z = D% 31
53 \ TRIM = 128 - [ 128 *
54 \ [ D% 33 - @80 ] / [ @0 - @80 ]]
55
56 : RUN \ ( -- UC1 )
57 D% 128 80 (R) DUP ND.
58 D% 33 OVER - D% 128 U*
59 ROT 00 (R) DUP ND. SWAP - U/ -
60 DUP NH.
61 812E F! \ store in EEPROM

62 COLD ; \ exit via Reset
63
64 \ RUN
65 \ press U 33 \ soll 32kHz Ref
66 \ 00 press U 40 \ ist mit RC=00
67 \ 80 press U 32 \ ist mit RC=80
68
69 |>

```

Listing 2

```

1 <| \ patch firmware
2
3 2852 SYS-EE E!
4
5 FFFF 80E0 E! \ clear EEPROM
6 FFFF 80E2 E!
7 FFFF 80E4 E!
8 FFFF 80E6 E!
9
10 80E0 >C ! \ compile to EEPROM
11 [CODE
12 812F LDA, 4E STA, \ copy TRIM
13 F7AA JMP,
14 CODE]
15
16 \ redirect Reset-Vector in EEPROM
17 CCCC SYS-EE 7C + E! \ JMP,-Opcode
18 80E0 SYS-EE 7D + E!
19
20 COLD \ restart via Reset
21 |>

```

Fortsetzung der Leserbriefe von Seite 8

Eindrücke von der Tagung der Forth-Gesellschaft

Am ersten Wochenende im Mai war es dann endlich soweit. Nachdem die Tagung aus organisatorischen Gründen von April in den Mai verschoben werden musste, trafen sich etwas über ein Dutzend Forther per Bildschirm, Kamera und Mikrofon, um gemeinsam ein paar schöne Stunden rund um das Thema Forth zu erleben. Die Tagungen der Forth-Gesellschaft sind immer öffentlich, es können also auch Nicht-Mitglieder teilnehmen.

Freitag: Plaudern und Warmwerden

Traditionell gibt es am ersten Tag ein informelles Treffen, also ein Beisammensein ohne besonderes Programm. So auch dieses Mal. Das ist die Gelegenheit, ganz zwanglos ins Gespräch zu kommen und sich über die neuesten Projekte auszutauschen. Für mich ist das eines der Highlights der Tagungen.

Samstag: Vorträge

Der Samstag war der Haupttag. Da kam ich aus dem Stauen nicht mehr heraus. Sehr interessante Vorträge wurden geboten. Die Vorträge wurden live auf der Streamingplattform Twitch „gesendet“. Leider sind sie dort nur für einen begrenzten Zeitraum einsehbar; danach werden sie vom Betreiber automatisch gelöscht. Nur gut, dass Gerald sie

vorher „rettet“ und auf YouTube hochläd. So könnt ihr sie auch noch Jahre später ansehen. Der Link zum Kanal ist <https://www.youtube.com/@4ther-tv/videos>

Am Samstagabend dann zwei Kuriositäten: das gemeinsame Abendessen und etwas später dann der Drachenrat. Ist schon witzig, vor der Kamera sitzend zu essen und dabei den anderen beim Essen zuzuschauen und gute Gespräche zu führen. Wer kommt nur auf so eine Idee? Der Drachenrat ist auch so eine Idee. Dies ist übrigens der einzige Teil der Tagung, der nicht öffentlich ist. Hier sucht sich unser Hausdrache Swappy eine neue Bleibe für ein Jahr bei einem würdigen Forther aus.

Sonntag: Mitgliederversammlung

Nicht nur die Tagung, auch die Mitgliederversammlung ist öffentlich. Wer jetzt denkt, unsere Versammlung sei so ein langweiliges Pflichtereignis mit viel „Vereinsmeierei“, den muss ich enttäuschen. Was das Sitzungsprotokoll (abgedruckt in diesem Heft) leider nicht hergibt, sind die Details der einzelnen Punkte. Hervorzuheben sind hier die Berichte über den Stand bestehender Projekte und Ideen für neue Projekte nebst anschließendem „Brainstorming“.

Ein wichtiger Punkt der Mitgliederversammlung war das Überreichen des Drachenpreises an unseren holländischen Freund WILLEM OUWERKERK. PHILIP ZEMBROD hielt die Laudatio. Schön war's. Van harte gefeliciteerd, Willem!

Wolfgang Strauß



MC9S08AW–Flasher

Rafael Deliano

Das handelsübliche Angebot an Programmiergeräten ist für viele Controller dürftig. Da ist der Eigenbau die bessere Lösung. Um zu funktionieren, genügt zwar eine rohe Leiterplatte, meist ist aber für den Einsatz in der Fertigung eine Verpackung in ein robustes Gehäuse gefordert (Abb. 1).

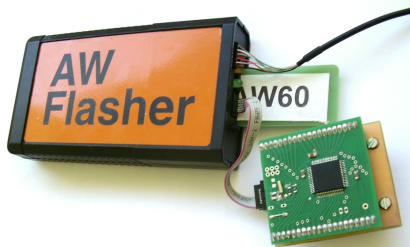


Abbildung 1: Flasher mit Nadeladapter für SBC

Zum Aufbau

Das FTDI-Kabel zum PC liefert die 5 V zum Betrieb (Abb. 5). Nach dem Power-up läuft der GP32-SBC mit dieser Spannung. Er kann aber seine Versorgung über Pin PD5 auf 3,3 V reduzieren für den Fall, dass das Target-PCB diese Spannung braucht — einige der Targets vertragen keine 5 V.

Zum Schutz beider Leiterplatten begrenzt man den Kurzschlussstrom, das erfolgt hier über einen Widerstand $R1=4,7\text{ Ohm}$. Dessen Wert muss aber auf die Stromverstärkung des verwendeten Transistors abgestimmt werden (Abb. 8). Die resultierende Ausgangsspannung kann am A/D-Wandler des Pins PB2 eingelesen werden. Damit ist die Fehlererkennung — meist verdrehtes Stecken des 6-poligen Kabels — vom Programm her möglich. Die Port-Pins für den BGD-Stecker sind mit ICs des Typs 74HCxx gepuffert (Abb. 4). Damit sind sie robuster gegen ESD¹, haben eine bessere Signalqualität durch die Schmitt-Trigger der Empfänger und eine höhere Treiberleistung der Sender. Alle Sender sind nach Reset zum Schutz des Target-PCBs deaktiviert.

¹ ElectroStatic Discharge

² PA ist die neuere Generation der Freescale/NXP Controller; sie ist „unglaublich billig“, doch derzeit nicht gut lieferbar. Die kleineren Gehäuse sind für Handlötung auch unattraktiver. Für Hobby und oft auch Industrie ist neuer nicht immer besser. (RD)

³ Bopla Gehäuse Systeme GmbH in Bünde.

Debug–UART

Die RXD/TXD am BGD-Stecker (Abb. 4) dienen hauptsächlich zum Abgleich des RC-Taktes im AW60- bzw. PA60-Controller², denn für den BGD muss der Bustakt des GP32 genau passen. Doch mit dieser „krummen Frequenz“ funktioniert der UART nicht. Deshalb wird dessen Baudrate an Pin PA6 eingespeist. Damit kann sich der halbduplex betriebene UART in Software abgleichen.

Speicher

Es wurden zwei Alternativen implementiert. Die EEPROMs (Abb. 7) befinden sich auf einer steckbaren Speicherkarte. Da der GP32 keine I²C-Schnittstelle hat, werden sie über Portpins angesteuert — für Geschwindigkeit nicht ideal.

5-V-FRAMs (Abb. 6) haben nur 32 kByte Speicherkapazität, es sind also zwei ICs nötig. Zudem gibt es sie nur im SO8-Gehäuse und nicht als DIP8. Sie wurden deshalb auf Adaptern montiert. Da sie über das SPI des Controllers angesteuert werden, ist sehr schnelles Lesen und Schreiben möglich. Allerdings ergab sich später, dass der Download über ein Terminalprogramm in Win 11 langsam ist, so dass die FRAM-Variante letztendlich doch nicht verwendet wurde.

Mechanik

Das Gehäuse BOS752 von Bopla³ gehört zu einer Serie (Tab. 1), in der auch Varianten für Displays erhältlich sind. Ein Display war hier aber unnötig, ebensowenig wie das 9-V-Batteriefach.

Nr.	Gehäusotyp
BOS750	Panoramascheibe
BOS751	Displayscheibe
BOS752	Folientastatur

Tabelle 1: Bopla-Gehäuse

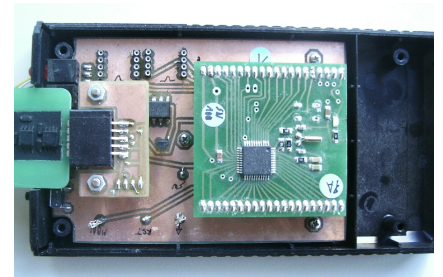


Abbildung 2: Hauptplatine in der Oberseite

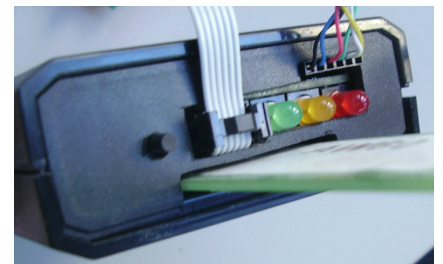


Abbildung 3: Frontplatte

Ein Vorteil kompatibler Gehäusefamilien ist, dass man schneller Geräte für unterschiedliche Anforderungen realisieren kann, da neue Layouts von alten abgeleitet werden können. Der Gehäusotyp ist nicht billig, aber besonders geräumig. Und ABS ist angenehmer, als billige Kunststoffe zu bearbeiten. Da hier keine Folientastatur verwendet wird, kann die Vertiefung auf der Oberseite für eine Beschriftung genutzt werden (Abb. 1). Die ist sinnvoll, da abhängig vom Controller unterschiedliche Firmware für AW60 und den neueren PA60 installiert ist. Die Folie wurde durch

einen SDX1200⁴ geschnitten und mit Transferfolie übertragen.

Alle Bedienelemente sind auf der Frontplatte zusammengefasst. Verwendet wurde die beiliegende ABS-Platte von Bopla. Umrisse auf selbstklebende Folie drucken, aufkleben, bohren und mit der Laubsäge die Kontur sägen. Bis nichts mehr klemmt, ist aber immer noch eine Nacharbeit mit der Feile fällig. Dann hat man schon mal ein vorzeigbares Muster. Für Kleinserien sind jedoch bequemere Alternativen nötig. Da eine 1,5-mm-Platte passt, wäre es möglich, sie mit den anderen Platinen fertigen zu lassen und mit Folie zu bekleben. Wenn man eine Fräse hat, ist es naheliegender, ABS oder farbige GFK-Platten zu bearbeiten.⁵ Ideal ist vermutlich ein blauer 5-Watt-Laser, der Plastik schneiden kann.

GP32-SBC

Warum wird der antike GP32 von 2001 verwendet, um den Oldtimer AW60 von 2006 zu programmieren? Das BGD-Protokoll (Abb. 9) des

AW60 hat ein Timing, das man auch in Assembler nur dann nachbilden kann, wenn Host und Target mit gleichem Takt arbeiten. Das Target verwendet aber einen RC-Oszillator, dessen Frequenz streut. Die liegt beim AW bei etwa 4 MHz, beim neueren PA von 2012 bei 8 MHz. Es ist also nötig, einen Controller zu verwenden, dessen Takt per PLL⁶ fein einstellbar ist. Das ist nur beim GP32 der Fall.⁷ Komplizierte Einstellung, aber viele Einstellmöglichkeiten. Insbesondere der 10-Bit-Frequenzteiler N mit hoher Auflösung (Abb. 10) ist da brauchbar. Für 2,45...8 MHz arbeitet man mit dem Prescaler P mit der Einstellung P=1, für bis zu 16 MHz mit P=2. Oberhalb 10 MHz liegt man jedoch nominell schon außerhalb der spezifizierten Geschwindigkeit des GP32.

SYNC

Für die Synchronisation ist im BGD-Protokoll die SYNC-Sequenz vorgesehen (Abb. 9). Der Host sendet erst einen überbreiten Puls und erhält

dann vom Target einen Puls mit exakt 128 etu⁸ zurück. In Software benötigt man ein Unterprogramm (Listing 1), das nach einer programmierbaren Verzögerung den Portpin abtastet. Eine Suchroutine kann dann exakt die steigende Flanke bestimmen. Für die erforderliche hohe zeitliche Auflösung bietet sich eine Liste aus 256 NOP-Opcodes an, deren Anfang von einem programmierbaren JMP-Opcode übersprungen wird.

Nadeladapter

Bleibt noch die Frage zu klären, wo denn eigentlich die Eingangs erwähnte Nadel ist und warum es die überhaupt gibt. Die 6-polige Stiftleiste kostet Geld, macht in einigen Anwendungen bei der Bauhöhe Probleme und ist nicht verpolungssicher. Daher wird sie nun auf den SBCs nicht mehr bestückt, sondern stattdessen wird ein kleiner Nadeladapter mit nur einer Nadel für den BKGD-Pin verwendet (Abb. 12). Die übrigen Signale sind an der 40-pol. Stiftleiste ja bereits vorhanden.

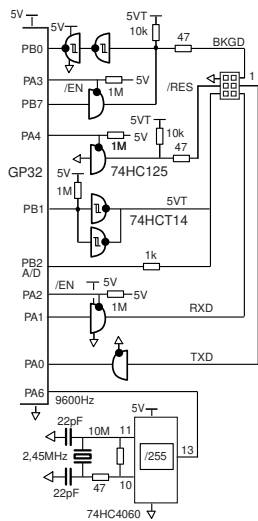


Abbildung 4: BGD-Schnittstelle

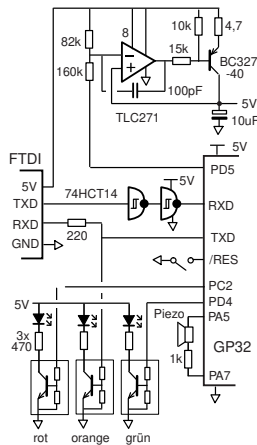


Abbildung 5: Stromversorgung, FTDI, Bedienelemente

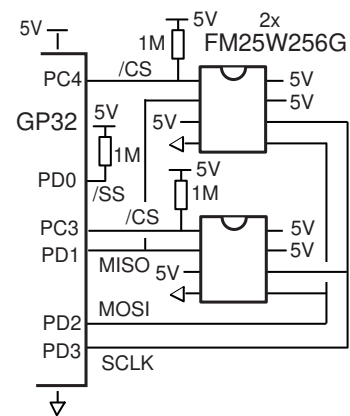


Abbildung 6: FRAM

⁴ Brother ScanNCut SDX1200 Schneideplotter

⁵ Acrylnitril-Butadien-Styrol-Copolymere (Kurzzeichen ABS) sind thermoplastische Terpolymere, die als Platten ab 1 mm Dicke erhältlich sind. GFK steht für: Glasfaserverstärkter Kunststoff.

⁶ Phase Locked Loop

⁷ Das war die erste Generation mit PLL.

⁸ elementary timing units

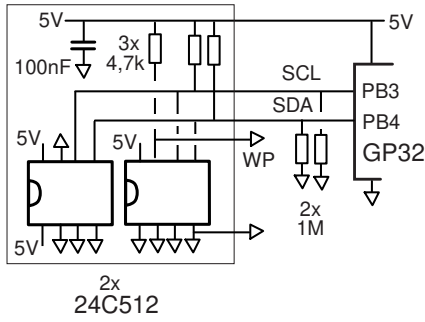


Abbildung 7: EEPROM auf Chipkarte

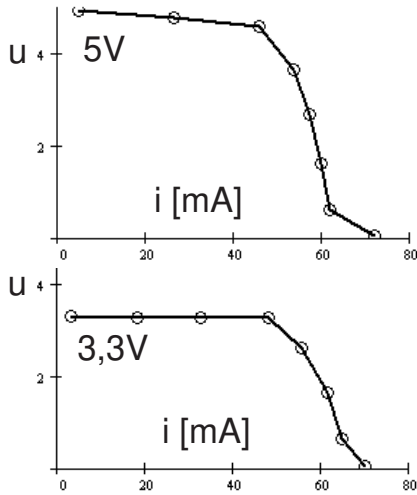


Abbildung 8: Strombegrenzung

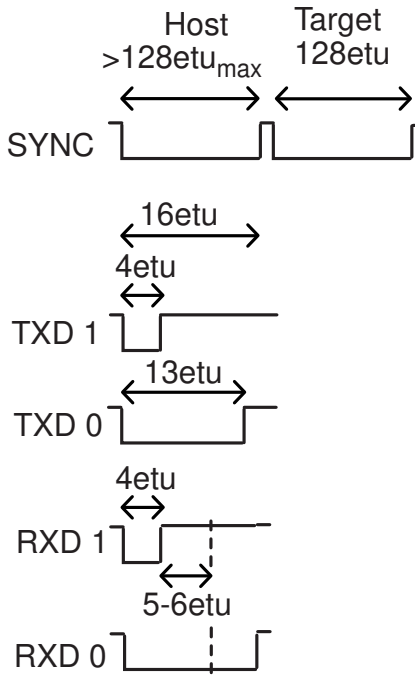


Abbildung 9: Datenübertragung

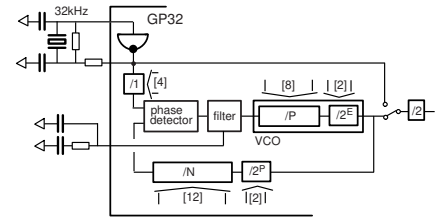


Abbildung 10: Takteinheit des GP32

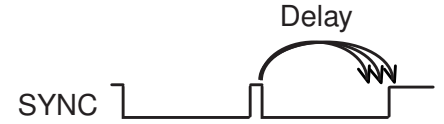


Abbildung 11: SYNC-Sampler

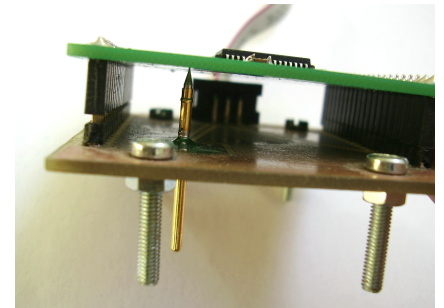


Abbildung 12: Nadeladapter für den BKGD-Pin

Listing 1: SYNC-Sampler

```

1  :CODE SYNC' \ ( UC1 -- flag ) UC1 = delay
2                \          flag = level at BKGD-pin
3      1 ,X      LDA,
4  XSAVE        SHX,
5                TAX, \ stack to X-register
6                H. CLR,
7      BKGD0     MBC, \ pulse low
8      FF        #. LDA,
9  1 $: 1 $      A. DBN, \ 3
10     BKGD0     MBS, \ 4cyc
11     BKGD0-en  MBS, \ 4cyc
12     NOP, NOP, NOP, NOP, \ 4cyc
13 >C @ 3 +     ,X JMP, \ 6cyc
14
15
16
17
18 \ Insert 256 * 1cyc
19 \ i.e. 256 NOP 0pcodes
20 \ Note: In 8MHz-Target 128 * 1cyc will do.
21 NOP, NOP, NOP, NOP, NOP, NOP, NOP,
22 ...
23 NOP, NOP, NOP, NOP, NOP, NOP, NOP,
24
25
26     BKGD-PORT LDA, \ 3cyc sample pin
27 2 $: 2 $ BKGD0 BBC, \ wait for rising edge
28     BKGD0     MBS,
29     BKGD0-en  MBC,
30                01 #. AND, \ pin to stack
31 XSAVE        LHX,
32     1 ,X      STA,
33                RTS,
34 CODE;
    
```


Large Literals for the RP2040¹ Assembler

Willem Ouwerkerk

Only literals from 3 to 8 bits can be used directly in the assembler for the RP2040. So for 16-bit and 32-bit numbers we need a different method. The classic way with ARM are “literal pools”.

What is a Literal Pool?

“Perhaps the most common type of literal pool are those used by the LDR Rd,=const pseudo-instruction in ARM assembly language and similar instructions in IBM System/360 assembly language, which are compiled to a LOAD with a PC-relative addressing mode and the constant stored in the literal pool.” https://en.wikipedia.org/wiki/Literal_pool

In Forth we use literals to push integers onto the data stack as quickly as possible. The routine for this must therefore be in machine code, but which one is the best, short and quick?

In combination with the LDR opcode in the M0+ instruction set, there is a label mechanism that enables literal pooling:

```
LDR Rn, =ape
```

or this:

```
LDR Rn, =0x12345678
```

However, this would mean that a fairly large assembler would have to be loaded in (no)Forth. Because the literals could only be resolved after the assembled code was complete. A difficult task for a single-pass assembler! So we tried forth-like alternatives.

Let's Use It!

For *noForth T*² we can use a register that is already used by NEXT.

NEXT goes like this:

```
: NEXT, ( -- )
  ip { w } ldm,
  \ Read CFA to W & increase IP
  w { hop } ldm,
  \ Read code address to HOP & increase W
  pc hop mov, ;
  \ Move code address to PC
```

As you can see, the W register is loaded with the address of the code field. After reading the contents to the HOP register, the W register is increased to right behind the code field (CFA). See figure at the end of the article.

Single and Multiple Inline One-Pass Pooling

We can take advantage of this, here is an example:

```
code COLD
  10000047 ,
code>
  hop w ) ldr,
  hop bx,
end-code
```

In this case the W register points to the data that is compiled right after the CFA of cold. The word CODE> is a directive that redirects the CFA to where the real code starts. Because the W register points directly to the data, we can simply read it to any register we want. In this case the HOP register. Finally HOP BX, jumps directly to 10000047 which is the address of the secondary boot code to restart noForth T.

We can also use this to read a whole range of data into registers, for example like this:

```
code SET-GPIO ( -- ) \ GPIO 2 to 29 are inputs
  D0000020 , \ = HOP
  40014014 , \ = DAY
code>
  w { hop day } ldm,
  \ HOP = D0000020, DAY = 40014014
```

Which in one 16-bit opcode reads one or more numbers, to as many registers, and it's fast, too! However, this is of no use in subroutines or interrupts.

The next example gives another way to store and read numbers.

pool, and apool, are two noForth T assembler directives that allow the use of large literals in the middle of any assembler code. They are used as follows:

```
code ANYWORD ( -- )
  tos sp -) str, \ Save TOS
  apool, \ Start new aligned pool
  87654321 , \ Literal 0
  10000100 , \ Literal 1
  true , \ Literal 2
then,
  tos sp -) str, \ Save TOS
  w { tos hop day } ldm,
  \ Read all three literals in one go
  \ TOS = lit0, HOP = lit1, DAY = lit2
```

¹ Raspberry Pi MCU; 2 Cores, Typ M0+, 4*64 KB (+ 8 KB for a total of 264 KB RAM), further 4 KB for USB and a 16-KB XIP cache, 0 Flash on chip.

² T stands for Thumb, and of course the T of Ford T. It is a noForth for the Thumb2 instruction set. And it runs on all Pico boards because they all use the RP2040.

```
...
next, end-code
```

The literals are compiled between the words `apool>` and `then`, but how does that work?

`apool`, saves the program counter PC in the W register. Then it jumps over the *inline data pool* using `ahead`, and `then`, .

The directives `pool`, and `apool`, are:

```
: POOL, ( -- )
  w pc mov, ahead, ;
  \ Start not aligned literal pool
: APOOL, ( -- )
  align, pool, ;
  \ Start aligned literal pool
```

So we have a set of tools to use fast 16-bit and 32-bit literals in the Forth assembler without having to use complex mechanisms.

Another Simple Literal Pooling Method

There is a third mechanism that is only useable for single literals and it's a bit less efficient. It looks a bit like the traditional ARM notation. I call it *simple literal pooling*:

```
code SOMEWORD ( -- n )
  tos sp -) str,
  \ Save TOS
  tos pc ) ldr, 12345678 ##
  \ Load 12345678 to TOS ...
```

It assembles to:

```
code SOMEWORD ( -- n )
  sp sp 4 # subs, \ Save TOS
  tos sp 0 x) str,
  tos pc 0 x) ldr, \ Load inline number to TOS
  ahead,
  12345678 , \ Inline number
  then,
  ...
```

Finally, a few more examples from the noForth T target file.

Real Examples

The first word resets most built-in RP2040 devices.

```
code RESET-ALL ( -- )
  4000C000 , 4000E000 , 01FFCDBF , 4000F000 , 003C7FFE ,
code> \ DAY = 4000C000, SUN = 4000E000, MOON = 1FFCDBF
  w { day sun moon } ldm,
  moon sun ) str, \ Set most built in devices in reset state
  w { hop moon } ldm, \ HOP = 4000F000, MOON = 3C7FFE
  moon hop ) str, \ Restart most built in devices
  begin,
  w day 8 x) ldr, \ Read reset done register to W
  w w mvns,
  w moon ands, \ Ready when all tested bits are zero?
  =? until,
  next,
end-code
```

The key primitive, it reads one key to TOS using UART0:

```
code KEY) ( -- c )
  40034000 , \ UART0 base address
code>
  sun w ) ldr,
  tos sp -) str,
  begin,
  day sun 18 x) ldr, \ Read flags (UARTFR)
  moon 10 # movs, \ Isolate KEY? flag
  day moon ands,
  =? until, \ Flag set?
  tos sun ) ldrb, \ Read character
  next,
end-code
```

Calculate the depth of the (no)Forth stack used:

```
code DEPTH ( -- n ) \ ok
  ramadr: SPO , \ Pointer to bottom of datastack
code>
  day w ) ldr, \ 2 - Read SPO-pointer to DAY
  day day ) ldr, \ 2 - Read stack base to DAY
  day day sp subs, \ 1 - Calc. used stack space
  tos sp -) str, \ 3 - Save TOS
  tos day mov, \ 1 - Used space to TOS
  tos tos 2 # asrs, \ 1 - Divide by 4
  next, \ 6
```

Links

You can find the complete noForth T assembler in the wiki:

https://wiki.forth-ev.de/doku.php/pfw:assemblers_rp2040-assembler

Cortex-M0+ Technical Reference Manual r0p1

<https://developer.arm.com/documentation/ddi0484/c/CHDCICDF>

ARM Compiler Toolchain Assembler Reference Version 5.03

<https://developer.arm.com/documentation/dui0489/i/arm-and-thumb-instructions/ldr--immediate-offset->

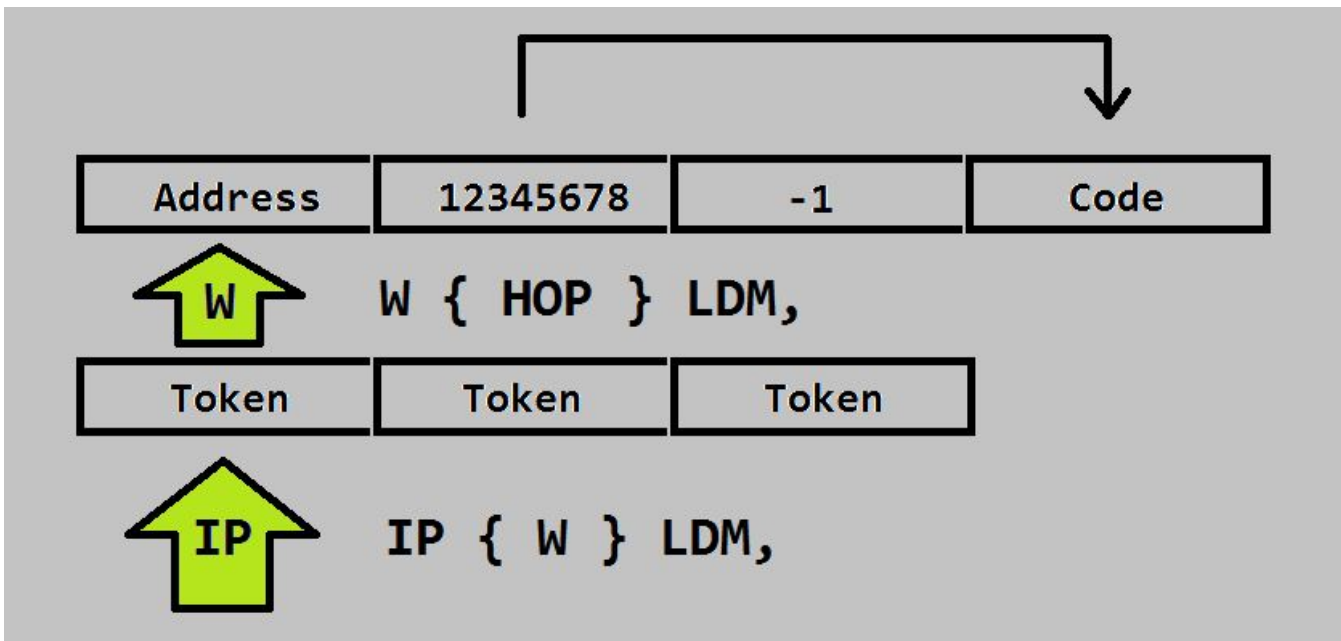


Figure 1: Fast long literal in noForth T

Astroimaging in Forth: Image File Format and Camera SDK

Andrew Read

Last time we spoke about astroimaging in general and specifically the control of the telescope mount over TCP/IP. Today I would like to share with you my progress in controlling a specialized camera with the manufacturer's SDK and saving images in an industry standard file format.

FITS is the longstanding astroimage file format. Its drawback is that 16-bit integers (most cameras capture pixel luminescence values as 16-bit integers) are stored in big-endian format. Little-endian CPUs must byte swap every word of the image buffer with each load or save. To illustrate the computational burden: the camera I am using, an ASI6200 (figures 1 and 2), has 62 million pixels, i.e. 62 million words in the image buffer! A newer standard for astroimages, XISF, which permits either endian format, but which is little endian by default and backwardly compatible with FITS metadata, is becoming more popular as a result.

How would XISF be approached in C? First we would look for (or take some time to write by ourselves) an XML library. Next, study the XISF standard documentation and build "business logic" to deal with all the XISF file structural possibilities. Obtain a variety of XISF files and test our code reading and writing them. Then I recalled a conversation with Ulli Hoffmann — "in C you build a framework to deal with the general case, in Forth you build exactly and only what you need". Let's proceed that way ...

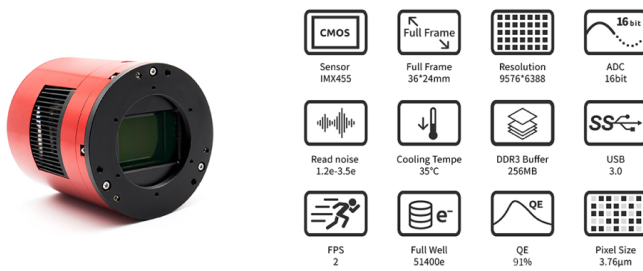


Figure 1: ASI6200, picture and features

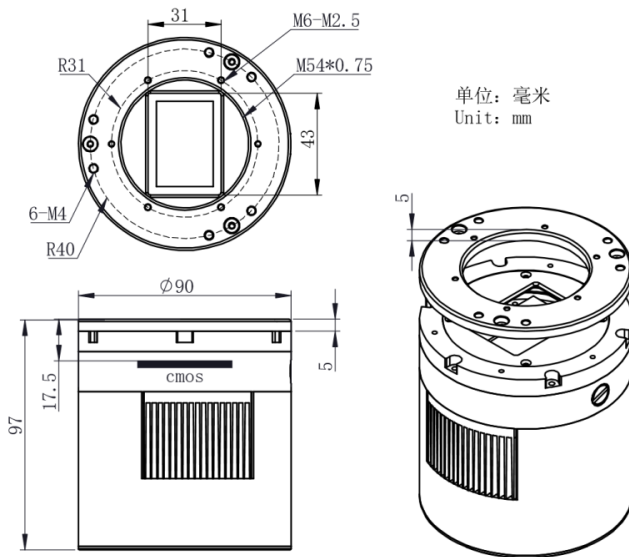


Figure 2: ASI6200, some technical info

At first glance of the XISF standard I wasn't optimistic. The file format is very flexible: XML is used to hold metadata, image data can be chained across several files, there are optional checksums, data may be encrypted.

First step: create an XISF image file with my camera using the manufacturer's software and open it in a hex editor (I use UltraEdit). Compare the file with the XISF standard and delete everything that is optional according to the standard and unnecessary to me. Save the abbreviated file, and confirm that it opens successfully in PixInsight (software for processing XISF images). Extract the XML part of the file and inspect the XML tags (UltraEdit again). Decide which parts of the XML are effectively just "constant text" and which parts should be written uniquely for each image. Finally, develop a lexicon of Forth words (XISF.f) to create this "theoretical minimum" XISF file: "signature" followed by "header" followed by image data. Here is how we proceed to initialize a buffer in the format of an XISF file.

```
XISF_BUFFER BUFFER: XISFBuffer
XISFBuffer XISF.StartHeader
XISF.StartXML
XISF.StartImage
```

XISF files include "FITS keyword" metadata carrying information about the image in string format within XML tags. The most common FITS keywords cover, for example, the target coordinates (RA DEC), date and time, exposure duration, equipment, etc. We use a Forth defining word to associate a FITS keyword with a variable, in this case an integer representing the focus position.

```
variable fpos 2000 fpos !
s" FOCUSPOS" fpos
XISF.MAKE-FITSKEY-INT
XISF.FITSfpospos
```

Writing the focus position to the header is done by XISF.FITSfpospos. Other words from the lexicon complete the header. The structure of the Forth code mirrors the structure of the header — the result is simple and logical.

```
XISFBuffer XISF.StartHeader
  XISF.StartXML
    XISF.StartImage
      XISF.FITSfocuspos
        \ write other FITS keywords
      XISF.FinishImage
    XISF.FinishXML
  XISF.FinishHeader
```

Before the final step of writing the buffer to a binary file the image data is obtained by calling the camera SDK with a pointer to the appropriate section of the buffer.

ZWO, the manufacturer of ASI imaging cameras supplies an SDK comprising a DLL, a C-language header file, and documentation. I had anticipated that it would be rather difficult to use all this from Forth but in fact it was very straightforward. Firstly, to confirm that the SDK “worked” and could connect to a camera I used Microsoft Visual Studio 2022 and prepared a simple C-language console application to call the functions in the DLL. VS makes this straightforward, but all of the templates are in C++ rather than C and need to be adapted — mainly with the delete key.

VFX Forth interfaces with DLLs through library and Extern: words, best understood by reviewing the listing (ASI_SDK.f). A few notes: (1) the DLL is most simply placed in the directory of the VFX executable. (2) the VFX developer console has a tool to extract all of the functions from DLLs and copy them to the clipboard to save time. (3) it is actually very easy and quick to write the Extern: function declarations by hand, referencing the C header file; an automated tool is probably not worth the trouble. (4) explicit whitespacing in the declarations is essential, even between the closing parenthesis and the semicolon.

ASI SDK functions generally take pointers to variables and structures and return an IOR-type error code. I prepared a Forth word to decode those error codes (referencing the SDK), and a simple succeed-or-abort error

handler for exploring the SDK functionality at the interpreter. C-language structures are easily mimicked with the Forth BEGIN-STRUCTURE and related words. I made two interesting observations:

1. the C compiler may pad structures for alignment but this must be compensated for manually in Forth.
2. C’s sizeof() does not always give the correct answer. A C double was reported as 8 bytes but compiled as 12 bytes (a long double).

I think you are already aware of these C-language data size quirks. In Forth it was not hard to explore and resolve the issues with DUMP but I guess in C my code would have just failed and I’d be turning to a debugger to set breakpoints and single-step. BEGIN-STRUCTURE also inspired a thought to mirror C enums with a word BEGIN-ENUM. This is certainly not original but it’s interesting how Forth encourages just-in-time reinventions!

I can now operate the ASI camera and save the images in XISF standard format using Forth. Some open-source repositories do the same thing in C++ or Python with, literally, thousands of lines of code. In Forth “first-level” functionality required roughly 250 lines and a couple of afternoons. The next step is to develop a camera control language at a level of abstraction about the ASI SDK, reflecting how I might like to interact with the telescope and camera during a live imaging session.

The code is presently in private repositories on GitHub but will become open source. Please contact me if you would like access and I welcome correspondence about the project in general. andrew81244@outlook.com

Links

XISF format:

<https://pixinsight.com/doc/docs/XISF-1.0-spec/XISF-1.0-spec.html>

ASI SDK:

<https://www.zwoastro.com/downloads/developers>

Listing

```
1 \ XISF.f - code for writing in the XISF image file format
2
3 4032 constant XISFHeaderMaxLen          \ review and adjust
4 640 480 2 * * constant XISFDataMaxLen   \ camera image size in bytes
5
6 BEGIN-STRUCTURE XISF_BUFFER
7 \ a 'theoretical minimum' XISF file
8   8 +FIELD XISF_SIGNATURE
9   4 +FIELD XISF_HEADER_LEN
10  4 +FIELD XISF_RESERVED
11  XISFHeaderMaxLen +FIELD XISF_HEADER   \ header with trailing zeros
12  XISFDataMaxLen   +FIELD XISF_DATA     \ data with trailing zeros
13 END-STRUCTURE
14
15 variable XISFBufferPointer
16 variable XISFHeaderPointer
17
```



```

18 : XISF.StartHeader ( XISFbuff -- )
19   dup XISFBufferPointer !
20   XISF_Header XISFHeaderPointer !
21 ;
22
23 : XISF.HeaderLength ( -- n )
24   XISFHeaderPointer @ XISFBufferPointer @ XISF_Header -
25 ;
26
27 : XISF.WriteToHeader ( addr n -- )
28   dup >R
29   XISFHeaderPointer @ swap cmove
30   R> XISFHeaderPointer +!
31 ;
32
33 : XISF.WriteIntToHeader ( x -- )
34 \ write an interger in string format - for XML tags
35   <# dup SIGN 0 ( x-as-double) #S #> ( caddr u)
36   XISF.WriteToHeader
37 ;
38
39 : XISF.StartXML ( -- )
40 \ XISF specification requires this
41   s" <?xml version="1.0" encoding="UTF-8"?>" XISF.WriteToHeader
42   s" <xisf version="1.0">" XISF.WriteToHeader
43 ;
44
45 : XISF.StartImage
46   s" <Image geometry=\"" XISF.WriteToHeader
47   \ ... write XISF defined parameters in XML (e.g. the image size)
48 ;
49
50 : XISF.MAKE-FITSKEY-INT ( caddr u addr <name> -- )
51 \ defining word for a FITS keyword having integer value
52 \ e.g. variable-name S" keyword" XISF.MAKE-FITSKEY-INT <name>
53   CREATE
54   , $,
55   DOES> ( -- , write a FITS keyword with a value in XML)
56   dup >R @ @ ( value)
57   R> cell+ count ( value caddr u)
58   s" <FITSKeyword name=\"" XISF.WriteToHeader
59   ( value caddr u) XISF.WriteToHeader
60   s" \" \" value=\"" XISF.WriteToHeader
61   ( value) XISF.WriteIntToHeader
62   s" \" \" />" XISF.WriteToHeader
63 ;
64
65 : XISF.WriteFile ( caddr n --)
66 \ write the buffer to a binary file
67 \ ... create-file ... write-file ... close-file
68 ;
69
70 \ a test run
71 XISF_BUFFER BUFFER: XISFBuffer
72 variable fpos 2000 fpos !
73 s" FOCUSPOS" fpos XISF.MAKE-FITSKEY-INT XISF.FITSfocuspos
74
75 XISFBuffer XISF.StartHeader
76   XISF.StartXML
77   XISF.StartImage
78   XISF.FITSfocuspos
79   \ ... write other FITS keywords
80   XISF.FinishImage
81   XISF.FinishXML
82 XISF.FinishHeader
83 \ ... call the ASI SDK with a pointer to the buffer and expose an image
84 s" C:\test\MadeInForth.XISF" XISF.WriteFile
85
86 \ //////////////////////////////////////
87 \ ASI_SDK.f - code for interfacing to the ASI SDK
88
89 LIBRARY: ASICamera2.dll \ place the dll in directory of the VFX executable
90
91 Extern: int "C" ASIGetCameraProperty( int * ASICameraInfo , int CameraIndex ) ;
92 Extern: int "C" ASIGetNumOfConnectedCameras( void ) ;
93 \ ... around 20 further Extern: function declarations in this SDK

```



```

94
95 : ASI.Error ( n -- caddr u)
96 \ convert and ASI error code to text
97 CASE
98   0 OF s" SUCCESS" ENDOF
99   1 OF s" INVALID_INDEX" ENDOF
100  2 OF s" INVALID_ID" ENDOF
101  \ ... other codes
102  s" OTHER_ERROR" rot ( care of case selector)
103  ENDCASE
104 ;
105
106 : ASI.?abort ( n --)
107 \ do-or-die error handler, for interactive testing
108   dup ASI.Error type CR
109   IF abort THEN
110 ;
111
112 BEGIN-STRUCTURE ASI_CAMERA_INFO      \ 240 bytes
113 64 +FIELD ASI_CAMERA_NAME
114 4 +FIELD ASI_CAMERA_ID
115 4 +FIELD ASI_MAX_HEIGHT
116 4 +FIELD ASI_MAX_WIDTH
117 12 +FIELD ASI_PIXEL_SIZE           \ long double
118 \ ... other fields
119 END-STRUCTURE
120
121 \ define an ENUM data-structure with similar syntax to a STRUCTURE
122 : BEGIN-ENUM 0 ;
123 : END-ENUM drop ;
124 : +ENUM dup 1+ swap CONSTANT ;
125
126 BEGIN-ENUM ( ASI_IMG_TYPE )
127   +ENUM ASI_IMG_RAW8
128   +ENUM ASI_IMG_RGB24
129   \ ...
130 END-ENUM
131 -1 CONSTANT ASI_IMG_END
132
133 ASI_CAMERA_INFO  BUFFER: ASICameraInfo
134 VARIABLE CameraID
135
136 : ASI.ReviewCameras
137 \ get and print all the properties of all the connected cameras
138   ASIGetNumOfConnectedCameras ( -- n)
139   ?dup
140   IF
141     \ loop over each connected camera
142     0 do
143       ." Camera index " i . CR
144       ASICameraInfo i ASIGetCameraProperty ( buffer index --) ASI.?abort
145       ASICameraInfo ASI_CAMERA_NAME ." ASI_CAMERA_NAME " zcount type CR
146       ASICameraInfo ASI_CAMERA_ID ." ASI_CAMERA_ID " l@ dup CameraID ! u. CR
147       ASICameraInfo ASI_MAX_HEIGHT ." ASI_MAX_HEIGHT " l@ . CR
148       \ ... other properties
149       CR
150     loop
151   ELSE
152     ." No ASI cameras connected" CR
153   THEN
154 ;
155
156 \ a test run
157 ." ASICamera2.dll load address " ASICamera2.dll u. CR
158 .BadExterns CR \ list any unrecognized extern functions
159 ASI.ReviewCameras

```

Protokoll der Mitgliederversammlung Forth–Gesellschaft e.V. am 7.5.2023

Anton Ertl

1. **Begrüßung der Teilnehmer und Feststellung der Beschlussfähigkeit**
14 Teilnehmer, somit gegeben.
2. **Wahl des Schriftführers**
M. Anton Ertl
3. **Wahl des Versammlungsleiters**
Klaus Kohl–Schöpe
4. **Ergänzungen zur Tagesordnung**
Die hier angesprochenen Punkte wurden alle unter „Verschiedenes“ behandelt.
Sodann: Verleihung des Drachenspreises an Willem Ouwerkerk.
5. **Bericht des Direktoriums**
 - Mitgliederentwicklung und Kassenbericht (Carsten Strotmann); 2 Austritte, 3 Eintritte. Der Kassenbericht wurde abgegeben. Details sind auf Anfrage beim Vorstand erhältlich.
 - Der Kassenprüfer (Wolfgang Strauß) konnte das Kas-senbuch nachvollziehen.
 - Rund um das Forth–Magazin (Ulrich Hoffmann)
 - Internet–Präsenz: Aktivitäten auf YouTube/Twit-ch/Webseite (Gerald Wodni).
 - Außendarstellung und Projekte (Bernd Paysan)
6. **Entlastung des Direktoriums**
Anton Ertl beantragt die Entlastung des Direktoriums. Abstimmung über `strawpoll.de`: 13J:0N:1E. Das Direktorium wurde damit entlastet.
7. **Wahl des Direktoriums**
Das bestehende Direktorium steht für eine weitere Periode zur Verfügung. Es gibt keine weiteren Kandi-daten.
Abstimmung über `strawpoll.de`: „Wollen Sie das be-stehende Direktorium beibehalten?“ 14J:0N:0E.
Die neuen, alten Direktoren nehmen die Wahl an.
8. **Projekte**
 - Ulrich Hoffmann berichtet über „Project Forthworks“.
 - Philip Zembrod berichtet über Volksforth.
 - Wolfgang Strauß berichtet über das Projekt Feuer-stein.
 - Carsten Strotmann berichtet über das Buch–Projekt.
 - Anton Ertl berichtet über „Rethinking Forth“.
9. **Verschiedenes**
 - Carsten Strotmann: Erhöhung des Mitgliedsbeitrags? Resultat der Diskussion: Nein.
 - Derzeit kostet die VD mehr, als für den Zweckbetrieb vorgesehen ist! Resultat der Diskussion: Das ist für uns akzeptabel.
 - Opt–out für die Papierversion ohne Reduktion des Mitgliedsbeitrags? Resultat der Diskussion: Wir ma-chen das.
 - Bernd Paysan berichtet über die Abschaltung des Strato–Servers. Mailman–Portierung. Diskussion über Abschaltung der Mailing–Listen.
 - Wolfgang Strauß: Englische VD. Wolfgang macht einmal die nächsten drei Ausgaben auch auf Englisch. Die deutsche Ausgabe ist davon unberührt und er-scheint wie gewohnt.
 - Ulrich Hoffmann: Kooperation mit der Facebook–Gruppe Forth2020.
 - Carsten Strotmann schlägt vor, ausgewählte Artikel als Ganzes auf der Website zu posten, und dabei auf die VD zu verlinken. Gerald Wodni schlägt vor, das unter einem VD–Logo zu posten; angenommen.
 - Ulrich Hoffmann: Dagstuhl–Seminar für konkatenati-ve Sprachen; zur Kenntnis.
 - Carsten Strotmann: Sommertreffen vom 30.6–2.7. im Linuxhotel beschlossen.
 - Carsten Strotmann: Mastodon — neuer Server für die Forth–Gesellschaft nötig.
 - Carsten Strotmann: `cloud.forth-ev.de` derzeit von Carsten bezahlt; Forth–Gesellschaft übernimmt.
 - Carsten Strotmann: Virgil Dupras (Tumbleforth, CollapseOS). Werbung für ihn machen o. ä.; angenom-men.
 - Martin Bitter: Eintrag der Forth–Gesellschaft im `handelsregister.de`; zur Kenntnisnahme.

Forth-Gruppen regional

Bitte erkundigt euch bei den Veranstaltern, ob die Treffen stattfinden. Das kann je nach Pandemie-Lage variieren.

Mannheim Thomas Prinz

Tel.: (0 62 71) – 28 30_p

Ewald Rieger

Tel.: (0 62 39) – 92 01 85_p

Treffen: jeden 1. Dienstag im Monat

Vereinslokal Segelverein Mannheim e.V. Flugplatz Mannheim-Neustheim

München Bernd Paysan

Tel.: (0 89) – 41 15 46 53

bernd@net2o.de

Treffen: Jeden 4. Donnerstag im Monat um 19:00 auf <http://public.senfcall.de/forth-muenchen>, Passwort over+swap.

Hamburg Ulrich Hoffmann

Tel.: (04103) – 80 48 41

uho@forth-ev.de

Treffen alle 1–2 Monate in loser Folge
Termine unter: <http://forth-ev.de>

Ruhrgebiet Carsten Strotmann

ruhrpott-forth@strotmann.de

Treffen alle 1–2 Monate im Unperfekthaus Essen

<http://unperfekthaus.de>.

Termine unter: <https://www.meetup.com/Essen-Forth-Meetup/>

Dienste der Forth-Gesellschaft

Nextcloud <https://cloud.forth-ev.de>

GitHub <https://github.com/forth-ev>

Twitch <https://www.twitch.tv/4ther>

µP-Controller-Verleih Carsten Strotmann

microcontrollerverleih@forth-ev.de

mcv@forth-ev.de

Spezielle Fachgebiete

Forth-Hardware in VHDL Klaus Schleisiek

microcore (uCore)

Tel.: (0 58 46) – 98 04 00 8_p

kschleisiek@freenet.de

KI, Object Oriented Forth, Ulrich Hoffmann

Sicherheitskritische Systeme

Tel.: (0 41 03) – 80 48 41

uho@forth-ev.de

Forth-Vertrieb

volksFORTH

ultraFORTH

RTX / FG / Super8

KK-FORTH

Ingenieurbüro

Klaus Kohl-Schöpe

Tel.: (0 82 66) – 36 09 862_p

Termine

Donnerstags ab 20:00 Uhr

Forth-Chat net2o forth@bernd mit dem Key keysearch kQusJ, voller Key:

kQusJzA;7*?t=uy@X}1GWr!+0qqp_Cn176t4(dQ*

Jeder 1. Montag im Monat ab 20:30 Uhr

Forth-Abend

Videotreffen (nicht nur) für Forthanfänger

Info und Teilnahmelink: E-Mail an wost@ewost.de

Jeder 2. Samstag im Monat

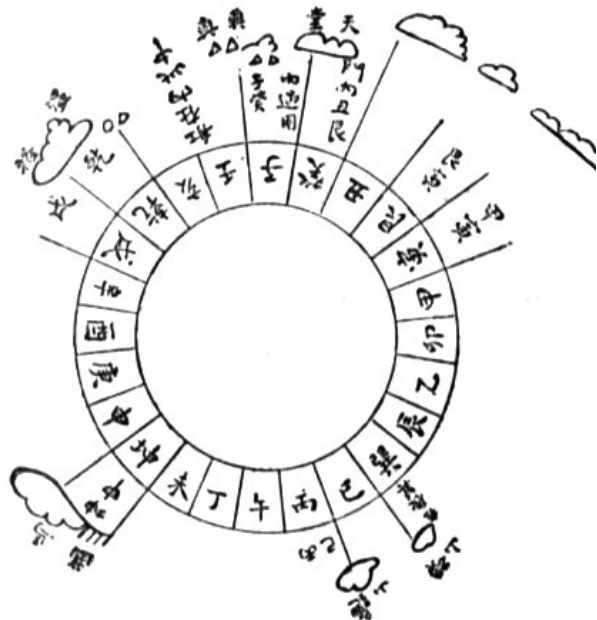
ZOOM-Treffen der Forth2020 Facebook-Gruppe

Infos zur Teilnahme: www.forth2020.org

Forth-Sommertreffen 30.06. bis 02.07.2023

Infos und Anmeldung: siehe Rückseite dieses Heftes

Details zu den Terminen unter <http://forth-ev.de>



Möchten Sie gerne in Ihrer Umgebung eine lokale Forthgruppe gründen, oder einfach nur regelmäßige Treffen initiieren? Oder können Sie sich vorstellen, ratsuchenden Forthern zu Forth (oder anderen Themen) Hilfestellung zu leisten? Möchten Sie gerne Kontakte knüpfen, die über die VD und das jährliche Mitgliedertreffen hinausgehen? Schreiben Sie einfach der VD — oder rufen Sie an — oder schicken Sie uns eine E-Mail!

Hinweise zu den Angaben nach den Telefonnummern:

Q = Anrufbeantworter

p = privat, außerhalb typischer Arbeitszeiten

g = geschäftlich

Die Adressen des Büros der Forth-Gesellschaft e.V. und der VD finden Sie im Impressum des Heftes.

**Einladung zum
Forth–Sommertreffen 2023 vom 30.06. bis 02.07.
im Linux Hotel in Essen–Horst**

Das Treffen findet in der Villa Vogelsang statt, Antonienallee 1, 45279 Essen–Horst. Das Gebäude liegt in einem Park mit Aussicht auf die Ruhr — eine Oase mitten im Revier.

Anreise

Essen ist mit der Bahn, Essen–Horst mit der S–Bahn erreichbar; per Auto über die A40 (Essen–Kray oder Essen–Frillendorf). Flughäfen gibt es mindestens drei zur Auswahl: Düsseldorf, Dortmund, Köln. Die liegen aber nicht „um die Ecke“. Und ja, man kann auch zu Fuß oder mit dem Fahrrad kommen — ist erwiesenermaßen machbar!

Anmeldung

<https://sommer.theforth.net>

Programm (kann sich noch ändern, aber im Prinzip locker, und immer mit Forth dabei. :)

Freitag

vormittags	Frühankommer, Ausflug
ab 13:00	Ankunft, Lobby
18:00	Gemeinsames Abendessen
20:30	Workshops

Samstag

09:00	Frühstück
10:00	Treffen und Tagen
12:30	Mittagessen
14:00	Tagung, Austausch
18:00	Gemeinsames Abendessen
20:30	Workshops

Sonntag

09:00	Frühstück
10:00	Treffen und Tagen
12:30	Ende, Abreise

