



Das Forth-Magazin

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

In dieser Ausgabe:



Win32forth+visualFORTH
controllieren Node-MCU

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
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.

LEGO RCX-Verleih

Seit unserem Gewinn (VD 1/2001 S.30) verfügt unsere Schule über so ausreichend viele RCX-Komponenten, dass ich meine privat eingebrachten Dinge nun Anderen, vorzugsweise Mitgliedern der Forth-Gesellschaft e. V., zur Verfügung stellen kann.

Angeboten wird: Ein komplettes LEGO-RCX-Set, so wie es für ca. 230,-€ im Handel zu erwerben ist.

Inhalt:

1 RCX, 1 Sendeturm, 2 Motoren, 4 Sensoren und ca. 1.000 LEGO Steine.

Anfragen bitte an
Martin.Bitter@t-online.de

Letztlich enthält das Ganze auch nicht mehr als einen Mikrocontroller der Familie H8/300 von Hitachi, ein paar Treiber und etwas Peripherie. Zudem: dieses Teil ist „narrensicher“!

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

Ingenieurbüro

Klaus Kohl-Schöpe

Tel.: (0 82 66)-36 09 862

Prof.-Hamp-Str. 5

D-87745 Eppishausen

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

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

Entwicklungsbüro Dr.-Ing. Egmont Woitzel

Bergstraße 10 D-18057 Rostock
Tel.: +49 381 496800-0 Fax: +49 381 496800-29

PC-basierte Forth-Entwicklungswerkzeuge, comFORTH für Windows und eingebettete und verteilte Systeme. Softwareentwicklung für Windows und Mikrocontroller mit Forth, C/C++, Delphi und Basic. Entwicklung von Gerätetreibern und Kommunikationssoftware für Windows 3.1, Windows95 und WindowsNT. Beratung zu Software-/Systementwurf. Mehr als 15 Jahre Erfahrung.



Cornu GmbH
Ingenieurdienstleistungen
Elektrotechnik

Weitstraße 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.

Hier könnte Ihre Anzeige stehen!

Wenn Sie ein Förderer der Forth-Gesellschaft e.V. sind oder werden möchten, sprechen Sie mit dem Forth-Büro über die Konditionen einer festen Anzeige.

Secretary@forth-ev.de

Leserbriefe und Meldungen	5
Win32forth+visualFORTH kontrollieren Node-MCU	6
<i>Peter Minuth</i>	

Titelbild AUTOR Text

Quelle

Impressum

Name der Zeitschrift Vierte Dimension

Herausgeberin

Forth-Gesellschaft e. V.
Postfach 32 01 24
68273 Mannheim
Tel: ++49(0)6239 9201-85, Fax: -86
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 Aufbauskizzen, die zum Nichtfunktionieren oder eventuellem Schadhaftwerden 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,

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

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-2018-03>

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

Ulrich Hoffmann Kontakt: Direktorium@Forth-ev.de
Bernd Paysan
Carsten Strotmann



Minitipp: Bitgruppen

Neulich vermisste WOLFGANG S. eine einfache Möglichkeit beim Schreiben in Register mittels ! (store), im Quellcode dafür die einzelnen Bitgruppen deutlich erkennbar zu machen. Neben einem Überdefinieren von ?number oder dem Einsatz von Recognizern, kam mir diese Idee:

```
%000.111.11.10.01.00 drop REGISTER !
```

Hier wird ausgenutzt, dass in einigen (oder gar allen?) Forthdialekten doppelt genaue Zahlen durch eine beliebige Anzahl von Punkten (vulgo: dots) eingegeben werden können. Ich selbst habe das sofort in meine Programmierweise übernommen.

Wen das drop stört, der kann ja etwas „syntactic sugar“ dazu nehmen:

```
: reg! ( dl dh adr -- )
  swap drop ! ;
  \ dl=bitmask dh=doublenumber high byte
  \ adr=register address
```

Obiges Beispiel sähe dann so aus:

```
%000.111.11.10.01.00 REGISTER reg!
```

Das Ganze geht auch im Hexadezimalmodus, solange die Bitgruppen sich schön an Nibbelgrenzen halten.

Martin

eForth 34.1, Naken Assembler Version

TING's eForth für den MSP430G2553, das *430eForth43*, gibt es bekanntlich seit diesem Jahr mit ausdrücklicher Zustimmung von Ting auch als Version für den Naken Assembler von MICHEAL KOHN. eForth ist damit aus

den Krallen solch aufgeblähter IDEs, wie dem des Code Composers Studio (CCS), befreit. Dieses eForth ist das minimalste komplette Forthsystem für diesen kleinen Chip. MANFRED MAHLOW (MM) hat sich damit näher befasst.¹

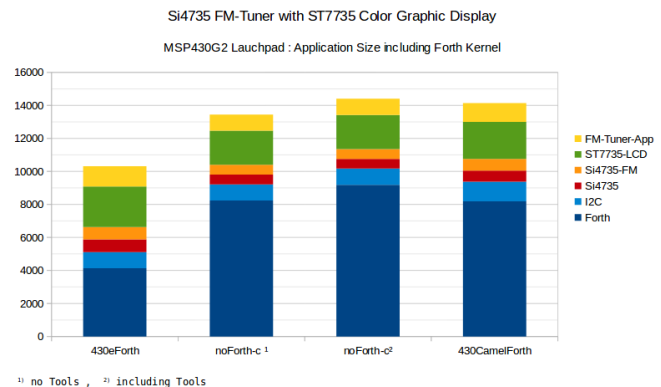


Abbildung 1: Vergleich der Speicherbelegungen

Klein, schnell und kompakt wie es ist, kann man damit aber auch in die *Reset-Falle des Flash* tappt. Wir hatten in unserem Magazin darüber berichtet.² Damit man bei der interaktiven Entwicklung auch im Flash-Speicher nach Herzenslust herum probieren kann, sind kleine Werkzeuge erforderlich, um das Flash immer wieder leeren zu können. MM hatte auf der Forthtagung 2017 schon vorgestellt wie das sehr platzsparend an das eForth angefügt werden kann, ohne dessen Kern aufzublähen, nämlich als nachladbares Tool. Und nun hat er das für die Naken-asm-Ausgabe des eForth aufbereitet. Als Tiny-Toolbox, zum nachladen. Das Projekt ist auf Github hinterlegt.

<https://github.com/mikalus/eForth43-msp430g2553-naken>

mk

¹ https://wiki.forth-ev.de/lib/exe/fetch.php/events:430eforth-tips_tools_tests2017.pdf

² Vierte Dimension 3/2017, Das RAM-ROM-Dilemma von interaktivem Forth in kleinen MCUs, M.Kalus; <https://wiki.forth-ev.de/lib/exe/fetch.php/vd-archiv:4d2017-03.pdf>



Win32forth+visualFORTH kontrollieren Node-MCU

Peter Minuth

Vor ca. 2 Jahren erwachte mein Interesse am IOT mit „PunyForth“, einem Forthdialekt für den ESP8266, geschrieben von ATTILA MAGYAR. Obwohl diese Implementierung sich sehr gut für meine Zwecke, das Internet of Things (IOT) eignete, entdeckte ich in der Anwendung einen Nachteil. Das Verbindungsinterface wurde in Python und nicht in Forth geschrieben.

Tatsächlich, wenn alle Tools von Expressif auf Python verfügbar sind, ist es für den Entwickler einfacher auf dem selben Weg zu bleiben. Aber für einen alten Forthliebhaber wie mich, bedeutete dies nicht nur Herzschmerzen, sondern auch eine unnötige Komplikation im Umgang mit zwei verschiedenen Sprachen. Kurz vor einem „Gehirnkurzschluss“ :) entschloss ich mich, das Interface auf Forth neu zu schreiben.

Im Zoo der Programmiersprachen

Als Ersatz für *Python*, die Schlange, könnte gut ein anderes Tier passen, ein echtes Mammut! Denn mit mehr als 5400 Worten kann man *Win32Forth* von TOM ZIMMER nicht besser bezeichnen. Ein Forth, das dank dieser Vielfalt immer allzeit bereit steht, um diverse Projekte zu verwirklichen. Nach einer Weile herum-geForth den Win32forth-Quellen, adaptierte ich ein USB Terminal, das sich prima an mein Projekt anpasste — der Mammut war fit und lebendig!

Die Kommunikation zwischen PC und Target lief vom ersten Moment an fehlerfrei. Als ich die Befehle im Win32console-Terminal eintippte, und das OK> von Punyforth zurückkam, fühlte ich mich wieder zuhause! ¹

Wer einen ESP8266² hat, will doch Wi-Fi!³ So war das nächste Ziel ganz klar (Abb.1). Aus der Fundgrube von Win32-Sourcen kam nochmals die Lösung: Der Folder *Internet* hat die komplette socket library von THOMAS DIXON. So steht das TCP-IP Protokoll gleich zur Verfügung ... Zu meinem Glück hatte auch Attila im Punyforth ein Beispiel vom TCP-IP REPL⁴, also die Interpreter Quelle wird statt aus dem UART vom TCP-IP Stack geholt, und beantwortet. Diesen TCP-IP REPL hatte ich schon im *TeraTerm*-Terminal getestet, und es lief perfekt. So entstand mit wenig Programmierarbeit die Verbindung von beiden Rechnern über Wi-Fi. Aber der eigentliche Treffer dieser Kombination ist, dass beide Systeme durch den eigenen Interpreter kommunizieren. Das heisst, unbeschränkte Möglichkeiten von Makros, also Wörter hin und her zu schicken und beliebig zu erweitern! Welche Sprache ausser Forth kann sowas?

¹ Punyforth hat als Default-Rückmeldung (**stack**), und ich tauschte es natürlich in den normalen OK> Prompt um.)

² NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

³ Wi-Fi bezeichnet sowohl ein Firmenkonsortium, das Geräte mit Funkschnittstellen zertifiziert, als auch den zugehörigen Markenbegriff.

⁴ Read-Eval-Print Loop (REPL)

Und das alles entstand mit den schon verfügbaren Forth-Tools plus Win32-Libraries, die ich nur anpassen musste.

In diesem Moment blickte ich auf dem Bildschirm, sah die beiden CPUs in lebendiger Wi-Fi-Kommunikation ... und sagte zu mir selbst „Willkommen Forth im 21. Jahrhundert!“

Dieses System hat mir die Möglichkeit gegeben, den ESP in verschiedenen Geräten meines Hauses als Controller für Steckdosen, Lampen, Wasserpumpen, usw. erfolgreich zu nutzen. Im Internet findet man tausende von Artikeln über IOT und Hausautomation auf Lua, Mpython, C, usw. Ich aber bin stolz darauf, dass bei uns zu Hause — mit einer richtigen IP — die Steckdosen mit einem herzlichen: „OK>“ grüssen.

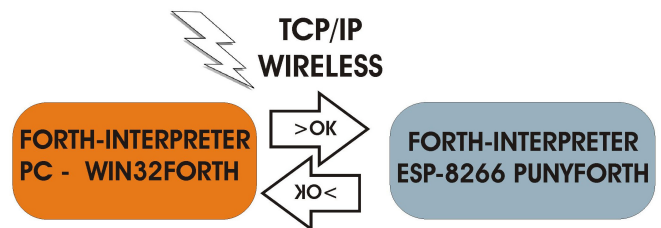


Abbildung 1: Schema der Kommunikation zwischen den Forth-Systemen

Wie das graphische User-Interface entstand

Eines Tages war ich entschlossen ein Interface in *visualFORTH* auszuprobieren (für ein anderes Programm). VisualForth (vF) ist eine Implementierung von DIRK BRUEHL auf Win32forth, das den *FormEditor* erweitert, und die Entwicklung von Programmen und GUI erleichtert.

Obwohl in der Hausautomation alles perfekt mit Punyforth funktionierte, fand ich die gekreuzte Benennung der verschiedenen IO-Pins ein wenig frustrierend. Denn die Pins haben eine andere Numerierung als die Ports.

Ich musste bei der Programmierung des ESP immer das Bild mit allen In-Out-Pins vor Augen haben, damit ich keinen Fehler verursache, der vielleicht die NodeMCU

zerstören würde. (Und aufpassen dass PunyForth case sensitive ist!) Es wurde für mich klar, das PunyForth noch ein kleiner Schub fehlte, bis es komplett „im 21. Jahrhundert“ landete.

Es was keine grosse Denkarbeit nötig: Wenn man visualFORTH in einem Fenster hat, Win32forth auf dem Fenster daneben über TCP IP verbunden mit der NodeMCU, und in der Mitte ein Bild mit allen IO-Pins zur Erinnerung ... Eureka! Problem und Lösung erschienen zusammen vor meinen Augen. (Abb.2 und Abb.5)

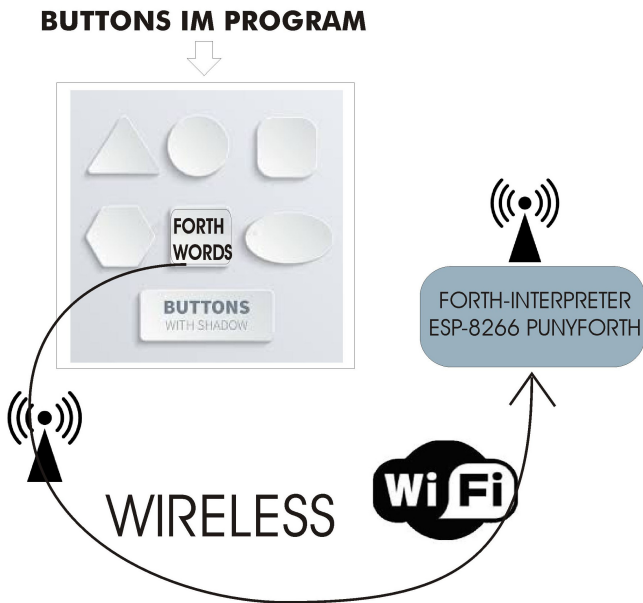


Abbildung 2: Forth-Kommunikation im IOT

VisualFORTH erklärt sich selbst bei der Benutzung

Der erste Schritt war es, das Bild vom ESP8266 mit den IO-Descriptions in einer Form vom vF einbinden. Man definiert einen Bitmap-Button (zusammen mit CTRL) und setzt dieses Photo als Hintergrund in der Mitte der Form. Dann setzt man an jedem Pin einen normalen Button. VisualFORTH kann über den Befehl *Properties* diesen Buttons Namen geben, und Code zuordnen. Dieser Code ist nichts anderes als „...GPIO-mode... etc“ im Falle man programmiert einen I/O, also Forth-Befehle wie *GPIO-read*, oder *GPIO-write* im Falle, dass man vom Pin lesen oder schreiben will, also einen Ausgang setzen. Natürlich kann *jedes* beliebige FORTH-Wort, das Punyforth kennt, ausgeführt werden, denn wir sprechen direkt zum Interpreter! Also ein Button kann ein Programm aufrufen, das Messungen am A/D-Port jede 50 ms ausführt, und der Messwert kommt einfach mit ⁵ im Win32forth an! So einfach ist der ganze Prozess. (Abb.3)

In visualFORTH habe ich im File-Menu die Option **TEST**, oder auch im Toolbar ein „Läufer“-Icon, das ist Synonym.

⁵ „dot“, also dem PRINT im Forth.

Form Copy-to-Clipboard

In einem anderem Fenster habe ich die Win32forth-IDE, also den Editor. Dann mache ich *File : NEW FILE* und einfach ein *Paste* um den Code zu kopieren. Die Win32Forth-IDE hat einen Button zum Kompilieren. So kann man alles gleich testen. (Abb.3)

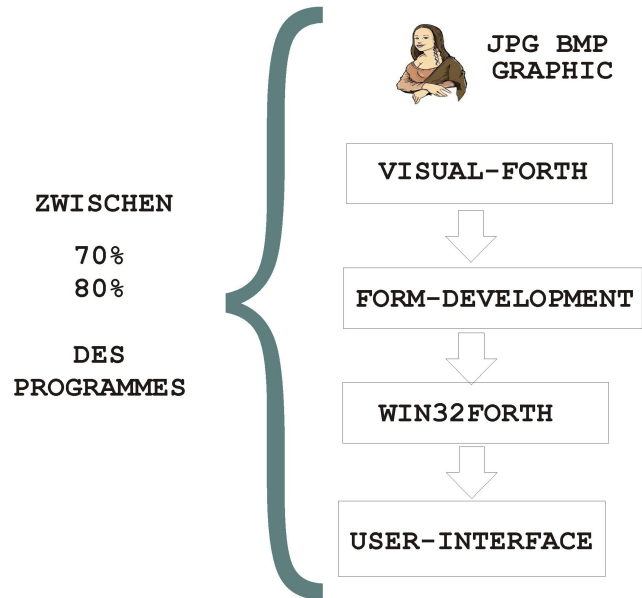


Abbildung 3: Programmiercyclus in Win32Forth + visualFORTH

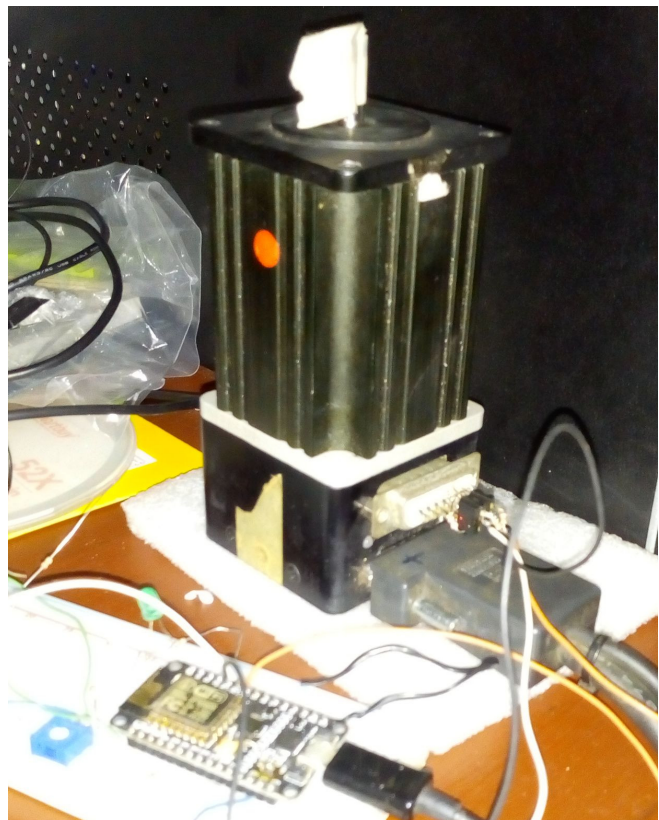


Abbildung 4: Einer meiner Tests: Servomotor ansteuern.

Wie funktioniert der Kontroll Panel?

Eigentlich gibt es nicht nur eine einzige Art- und Weise diesen Kontroll Panel (Abb.5) zu benutzen. Es dient als ein Beispiel das erweitert oder für spezifische oder ganz andere Zwecke adaptiert werden kann. Ich sehe interessante Anwendungen dafür: Lehrer von technischen Schulen könnten diese Idee in eine wunderbare Experimentierplattform weiterentwickeln. Für einen Physiker oder Biologe im Labor könnte die NodeMCU Instrumente oder Prozesse überwachen und steuern, und per Wi-Fi zum PC für weitere Überarbeitung senden. Auch Maker, die auf Steckplatinen⁶ Probeschaltungen mit Forth kontrollieren wollen, können hier Inspiration finden. (Abb.4)

Danksagung

Zum Schluß möchte ich mich herzlichst bei allen Autoren bedanken, die dieses phantastische Tool geschrieben haben. Danke also an TOM ZIMMER für Win32forth, an EZRABOYCE für den FormEditor, DIRK BRUEHL für sein visualFORTH und ATTILA MAGYAR für sein Punyforth.

Links

<https://sourceforge.net/projects/win32forth/>
<http://www.visualforth.org/>
<https://github.com/zeroflag/punyforth>

Listings

... gibt es beim Autor. Die Änderungen im Win32Forth, um das hier Vorgestellte zu verwirklichen, sind verstreut in den Quellen. Und im visualFORTH ist der Quellcode in den Forms enthalten. Das lässt sich nicht so ohne weiteres abdrucken. Aber angehängt sind zwei kleine Code-Beispiele, die zeigen, wie das Ganze funktioniert.

Für den Blinker gibt es außerdem ein Video auf YouTube. Es werden zwei Node-MCUs angesprochen.

<https://www.youtube.com/watch?v=NvZHxFj0pt8>

Kontakt

eMail: peter4th2017@gmail.com
Facebook: Peter Forth

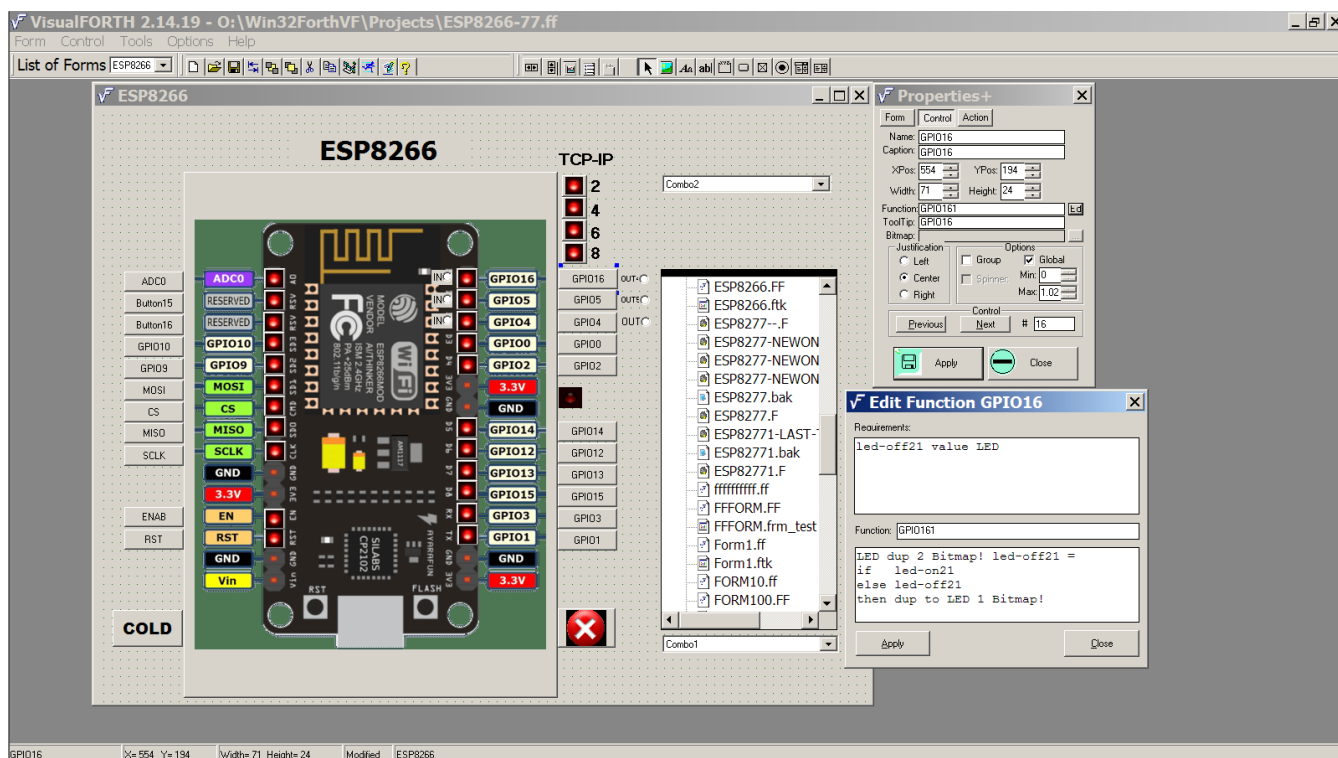


Abbildung 5: Das ESP-Kontrollpanel

Zwei kleine Beispiele

TELETYPE an Node-MCU

„Hallo world“-Blinker, Werte auslesen, Befehle im Node definieren, Funktionen an- und ausschalten ...

```
: telcr ( -- )
  13 telsend 10 telsend ; ( CR ans Telnet )
: telypeCR ( string -- )
```

⁶ den „breadboards“


```

teltype telcr ; ( teltype ist ein type ueber telnet)
: boton15
  s" 2 blink " teltype telcr ; ( led 2 blinkt)
: boton16
  s" 0 blink " teltype telcr ; ( led 0 blinkt)

\ read adc0 channel
: read-adc0 ( --n ) ( der ADC vom Nodemcu bekommt man im Win32forth stack)
  s" adc-read . " teltypeCR ;
' read-adc0 SetFunc: ADC0 ( ADC0 ist der Button am Bildschirm , entstand mit VForth )

: definput4 ( --n1...n100) ( 100 Messwerte lesen, aufpassen, alles erfolgt ueber Stack)
  s" : gg1 4 gpio-read ; " teltypeCR ( Port 4 wird gelesen )
  s" : gg 100 0 do gg1 . 1000 ms loop ; " teltypeCR ; ( test 100 mal lesen, je 1000 ms)

: blueledoff ( --) s" 2 GPIO_HIGH gpio-write " teltypeCR ;
\ 2 GPIO_LOW gpio-write
\ 2 GPIO_HIGH gpio-write
: blo blueledoff ;
' blo SetFunc: MOSI
\ ' MISO SetFunc: MISO
\ ' SCLK SetFunc: SCLK
' boton15 SetFunc: MISO ( alle Pins vom Nodemcu werden gleicherweise programmiert )
' boton16 SetFunc: SCLK

```

Schritt- oder Servomotor-Steuerung

Richtungs- und Positionssignale uebergeben an den Node.

```

: stepper-load ( --)
  cr ." >> loading stepper program " cr
  s" : gw gpio-write ; " teltypeCR 1000 ms
  s" : pul 2 1 gw 10 ms 2 0 gw ; " teltypeCR 1000 ms
  s" : tes2 200 0 do pul 3 ms loop ; " teltypeCR 1000 ms
  s" : tes 100 0 do pul 10 ms loop ; " teltypeCR 1000 ms ;
: tes
  cr ." *** stepping **** " cr
  s" tes " teltypeCR ; \ call the stepper prog
: tes2
  cr ." *** stepping fast**** " cr
  s" tes2 " teltypeCR ; \ call the stepper prog
' stepper-load SetFunc: GPIO3 \ Buttons werden mit Schrittmotor Kommandos geladen
' tes SetFunc: GPIO1
' tes2 SetFunc: GPIO15
\ : tes2 1000 0 do pul 3 ms loop ;

\ -----
: motright ( --) cr ." *** right **** " cr s" 0 0 gw " teltypeCR ;
: motleft ( --) cr ." *** left **** " cr s" 0 1 gw " teltypeCR ;
: switchLED00 ( -- ) \ Anzeige am Bildschirm, eine LED zeigt Rechts-/Linkslauf vom Motor.
  LED dup 68 Bitmap! led-off21 =
    if led-on21 motright
    else motleft led-off21
    then dup to LED 68 Bitmap! ;
' switchLED00 SetFunc: GPIO0
\ -----

```

[Zum Autor: Peter Minuth lebt heute in Sao Paulo, Brasilien. In Argentinien geboren wuchs er zweisprachig auf, Deutsch ist seine Vatersprache. Forth nutzt er seit vielen Jahren. Im Studium der Astrophysik und später in der professionellen Arbeit bei der CNC Maschinenherstellung — von einfachen Werkzeugwechsler bei Leiterplattenroboter bis complettem CAD/CAM für die Luftfahrtindustrie. Hobbys: Sternbeobachtung, Tauchen, Fliegen und Bluesmusik.]





Forth-Gruppen regional

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.paysan@gmx.de
Treffen: Jeden 4. Donnerstag im Monat
um 19:00 in der Pizzeria La Capannina,
Weitlstr. 142, 80995 München (Feldmo-
chinger Anger).

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 Freitags im Un-
perfekthaus Essen
<http://unperfekthaus.de>.
Termine unter: <http://forth-ev.de>

Gruppengründungen, Kontakte

Hier könnte Ihre Adresse oder Ihre Rufnummer stehen — wenn Sie eine Forthgruppe gründen wollen.

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 75 45) – 94 97 59 3_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 **Ingenieurbüro**
volksFORTH **Klaus Kohl-Schöpe**
ultraFORTH Tel.: (0 82 66) – 36 09 862_p
RTX / FG / Super8
KK-FORTH

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*

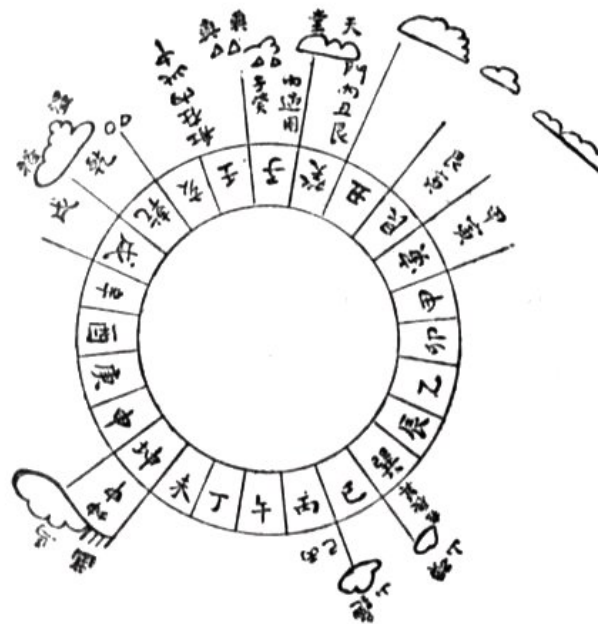
Datum
Forth-Tagung in ???
<https://tagung.forth-ev.de>

Datum
Maker Faire Hannover
<http://www.makerfairehannover.com>

Datum
EuroForth
<http://www.euroforth.org>

27.–30.12.20xx
3Xc3 in Hamburg

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.