



## DOCUMENT INFORMATION

**TITLE:** IoT devices and embedded systems with uLisp

**AUTHOR:** Max-Gerd Retzlaff

**VERSION:** presentation version without demo slides and source code

**PUBLICATION DATE:** 2022-03-21 (original version), 2022-05-16 (source code/demo version)

**DESCRIPTION:** Slides for the demo at the European Lisp Symposium 2022 in Porto, Portugal on March 21th, 2022, from 14:30 till 15:00. More information on the conference web site at <https://www.european-lisp-symposium.org/2022/>.

**PUBLICATION LOCATION:** The document is published on <http://www.defsystem.net> and can be referenced there. The direct link is <http://www.defsystem.net/id/74678c84-e7e3-400f-9558-e4c5f271f3a0>.

**RIGHTS:** No redistribution from other locations, no sharing of the document itself, no derivative works. That is, not without explicit permission. Just link to it.

MAX-GERD RETZLAFF

I O T   D E V I C E S   A N D  
E M B E D D E D   S Y S T E M S  
W I T H   U L I S P



**M5STACK ESP32 BASIC CORE****~40 EUROS (INCL. VAT)**

- 240 MHz 32-bit dual-core ESP32 processor (Espressif ESP32-D0WDQ6-V3),
- 520 KB SRAM in total for programs and data,
- 4 or 16 MB Flash memory,
- 2.4 GHz Wi-Fi and Bluetooth with built-in antennae,
- USB-C socket,
- 2 inch TFT LCD display with 320x240 pixels and 262 K colours,
- SD card slot,
- 150 mAh battery,
- 1 W Speaker and microphone,
- 3 buttons

**ESP32 M5STACK ATOM LITE ESP32****~10 EUROS (INCL. VAT)**

- 240 MHz 32-bit dual-core ESP32 processor (Espressif ESP32-PICO-D4),
- 520 KB SRAM in total for programs and data,
- 4 MB Flash memory,
- 2.4 GHz Wi-Fi and Bluetooth with built-in antennae,
- USB-C socket,
- 1 RGB LED (compatible to Adafruit NeoPixel),
- Infrared Transmitter,
- 1 button

## DOIT ESP32 DEVKIT V1

~6-7 EUROS (INCL. VAT)



- 240 MHz 32-bit dual-core ESP32 processor (Espressif ESP-WROOM-32),
- 520 KB SRAM in total for programs and data,
- 4 MB Flash memory,
- 2.4 GHz Wi-Fi and Bluetooth with built-in antennae,
- USB-C socket

## ULISP

## AUTHOR

Authored by David Johnson-Davies from Cambridge, UK.



**IDENTIFONT****DAVID JOHNSON-DAVIES, 2000**

A directory of digital fonts, including features to allow you to identify fonts by appearance, find fonts by name, find picture or symbol fonts, and find fonts by designer or publisher, at [www.identifont.com](http://www.identifont.com).

The screenshot shows the Identifont website interface. At the top, there's a navigation bar with links: About us, Blog, Latest fonts, Popular fonts, Fontset, Tools, Free fonts, Feedback, Contact us, and Terms. Below the navigation bar are five tabs: Fonts by Appearance (highlighted in yellow), Fonts by Name, Fonts by Similarity, Fonts by Picture, and Fonts by Designer/Publisher.

**Identify result**

Close matches (most likely first):  
Show all matches  
1 of 10

**Futura (Berthold)**

ABCDEFGHIJKLMNO  
PQRSTUVWXYZÀÉÂÎ  
abcdefghijklmnopqr  
stuvwxyzàéâîõøü&1  
234567890(\$£€.,!?)

Designers: Paul Renner and Dieter Hofrichter  
Year: 1928-32, 2000  
Publisher: Berthold

Buy this font online from:  
Berthold:  
<https://www.bertholdtypes.com/>

**Your Fontset**

Add to Fontset  
Show Fontset

**Similar fonts**

Fonts most similar to 'Futura (Berthold)':  
Futura Now Head.  
Futura Now Text  
Futura Next  
Futura ND  
Architype Renner  
Futura PT  
Futura No. 2  
Twentieth Century  
Futura  
Futura

Show more similar fonts

The largest independent directory of typefaces organized into categories,  
at [www.fontscape.com](http://www.fontscape.com).

**FONTSCAPE®**  
typeface directory

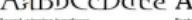
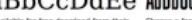
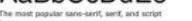
Welcome to Fontscape!

An independent directory of typefaces organized into categories.

**Topical**

- Spring and St Patrick's Day (March 17th)
- Spring (7)  Pictures for the Irish festival, St. Patrick's day, March 17th.
- St Patrick's Day (1)  Typefaces for the Irish festival, St. Patrick's day, March 17th.

**Categories**

- Application (199)  Typefaces classified according to their suitability for particular applications.
- Mood (59)  Typefaces that convey a particular mood or impression.
- Period (79)  Typefaces that capture a particular period or artistic movement.
- Simulation (447)  Typefaces that simulate the appearance of lettering created using a particular tool, such as a rubber stamp or typewriter.
- Classification (254)  The traditional typeface classifications, used by type historians and typographers.
- International (385)  Typefaces with a characteristic international flavour.
- Alternatives (56)  Suggested alternatives to the chosen typeface, often suggested by computers or printers, that have become overused and over-familiar.
- Award winners (806)  Award-winning typefaces.
- Free (311)  Fonts available for free download from their designers.
- Dimensions (68)  Choose a font with particular dimensions or metrics.
- Pictures (731)  Fonts containing collections of pictures.
- Picture Styles (203)  Fonts classified by drawing style.
- Symbols (605)  Fonts containing collections of symbols.
- Popular (60)  The most popular sans-serif, serif, and script fonts.

FONTIFIER

DAVID JOHNSON-DAVIES, 2003

Creates handwriting fonts from templates, at [www.fontifier.com](http://www.fontifier.com).



### Fontifier examples

A selection of fonts created by Fontifier users.

**Thin felt pen**

The quick brown fox  
jumps over a lazy dog.

**Felt pen (Ralph Percival)**

The quick brown fox  
jumps over a lazy dog.

**Chicken scratch (Chris, MacUser UK)**

The quick brown fox  
jumps over a lazy dog

**Thin ball-point pen (Yva Williams)**

The quick brown fox  
jumps over a lazy dog.

**Thick felt pen, drying out**

The quick brown fox  
jumps over a lazy dog.

**Pictures A-Z (Nik, MacUser UK)**

ବ୍ୟୁତିରେ ଲାଗୁ ହୋଇଥିବା କବିତା  
ପାଦମାଳା ଏବଂ ଅନ୍ୟ ଚିତ୍ରାବଳୀ

**IDENTITREE****DAVID JOHNSON-DAVIES, 2015**

An expert system written in Lisp designed to enable users to identify trees from their visual characteristics, available online at [www.identitree.com](http://www.identitree.com).

**Identitree** [Trees by Name](#) [Trees by Feature](#) [Trees by Genus](#)  [Find](#)

[About Identitree](#) | [Terms of use](#) | [Contact Identitree](#)

**Trees by feature**

Find trees by a selection of popular features.

**Leaf type**  
What type are the leaves?



Needles      Broadleaved      Scale      Yellow

**Simple leaves with lobes**  
How many lobes do the leaves have?



Lobed      3 Lobes      7-11 Lobes

**Compound leaves with leaflets**  
How many leaflets do the leaves have?



3 leaflets      5 or 7 leaflets      9 to 11 leaflets      More than 11 leaflets

## CL-HTTP

The websites IDENTIFONT, FONTSCAPE, FONTIFIER, IDENTITREE, and also ULISP  
are all based on: CL-HTTP.

CL-HTTP is the Common Lisp Hypermedia Server, developed by John C. Mallory  
at the M.I.T. Computer Science & Artificial Intelligence Laboratory.

A CL-HTTP PRIMER by David Johnson-Davies is also served via CL-HTTP  
at <http://clhttp.plasticki.com/>.

**ULISP****NAME**

“I decided to settle on uLisp, pronounced ‘U’-lisp,  
rather than  $\mu$ Lisp, pronounced mu-lisp,  
for two reasons:

- It’s clearer what the URL for the website should be.
- The ‘ $\mu$ ’ character often doesn’t look very good on-screen on a page of text.

Also, I was able to trademark “ULISP” as a word.”

**ULISP****LISP FOR MICROCONTROLLERS**

- specifically designed for limited RAM
- main challenge of writing uLisp:  
have it run on the ATmega328 with only 2 kB of RAM  
(and 32 kB of flash program memory)
- 8-, 16-, 32- and 64-bit platforms
- from the Arduino Uno based on the ATmega328 up to the Teensy 4.0/4.1

**ULISP****IMPLEMENTATION**

- interpreter not compiler
- tail-optimization  
(efficient recursive functions)
- mark and sweep garbage collector  
(Under 1 msec on an Arduino Uno or under 3 msec on an Arduino Mega 2560.)

**ULISP****LANGUAGE**

- The language is generally a subset of Common Lisp, this is:  
uLisp programs should also run under Common Lisp.
- major difference: It's a Lisp-1.  
(One namespace for functions and variables.)

**ULISP****ORIGINAL OBJECTIVE**

“[...] a compact Lisp for small microcontrollers that would allow people to use a more advanced language than C for creating hardware-oriented projects. [...]”

There’s no point making uLisp into a full Common Lisp.

It won’t make uLisp more likely to be adopted by microcontroller users, and people with larger processor boards such as the Raspberry Pi can already get a full Common Lisp running under Unix (such as Clozure Common Lisp).

So for each proposed extension I have to think: will this make it more likely that people use uLisp for hardware projects? I think that the way to do this is to make it *appeal to people who haven’t encountered Lisp before*, rather than trying to appeal to experienced Common Lisp users.”

## ULISP VS. COMMON LISP

- major difference: It's a Lisp-1.  
One namespace for functions and variables.
- no macros
- no multiple-values,  
instead: use lists to represent multiple values.

NOTHING – symbol with no value, equivalent to (values) in Common Lisp,  
to suppress output from functions

- no DESTRUCTURING-BIND
- no condition system or exceptions,  
error brings you back to the top-level
- ; (semicolon) – comments end not at the end of the line  
but *before the next opening bracket*



[Getting started](#)

[Lisp for microcontrollers](#)

[Performance](#)

[Using uLisp](#)

[Using uLisp from a terminal](#)

[Debugging in uLisp](#)

[Using the program editor](#)

[SD card interface](#)

[I2C and SPI serial interfaces](#)

[Lisp Library](#)

[Download uLisp](#)

[uLisp Sensor Library](#)

[Forum](#)

[Self-contained Lisp computers](#)

[Tiny Lisp Computer](#)

[Lisp Badge](#)

[Pocket Op Amp Lab](#)

[8/16-bit platforms](#)

[Arduino Uno](#)

[Arduino Mega 2560](#)

[ATmega1284](#)

[ATmega4809 boards](#)

[AVR DA and DB series boards](#)

[32/64-bit platforms](#)

[Arduino M0 Boards](#)

## uLisp

[Home](#)

[Reference](#)

[Forum](#)

[Search](#)

# Lisp for microcontrollers

Lisp for Arduino, Adafruit M0/M4, Micro:bit, ESP8266/32, RISC-V, and Teensy 4.x boards.

## News!

### New ARM Version 4.1

Like the AVR Version 4.1 of uLisp released earlier this month, the ARM version now adds a register function to allow you to read the values of 32-bit processor registers, or write values to the registers. It allows you to control the peripherals in the ARM processor from a Lisp program, or interactively experiment with the peripherals by giving commands at the uLisp prompt.

For more information see: [ARM uLisp now also supports the register function](#).

uLisp® is a version of the Lisp programming language specifically designed to run on microcontrollers with a limited amount of RAM, from the Arduino Uno based on the ATmega328 up to the Teensy 4.0/4.1. You can use exactly the same uLisp program, irrespective of the platform.

Because uLisp is an interpreter you can type commands in, and see the effect immediately, without having to compile and upload your program. This makes it an ideal environment for learning to program, or for setting up simple electronic devices.

Lisp is also an ideal language for learning about fundamental programming concepts. It incorporates string handling, list processing, and garbage collection, and so is also an excellent language for expressing complex ideas, such as teaching a robot to solve mazes or finding the shortest route on a map. As well as supporting a core set of Lisp functions uLisp includes Arduino extensions, making it ideal as a control language for the Arduino.

You can download the current version of uLisp free from the [Download uLisp](#) page.

**ULISP****GREAT WEB SITE****uLisp****Home**
 
**Getting started**

- [Lisp for microcontrollers](#)
- [Performance](#)
- [Using uLisp](#)
- [Using uLisp from a terminal](#)
- [Debugging in uLisp](#)
- [Using the program editor](#)
- [SD card interface](#)
- [I2C and SPI serial interfaces](#)
- [Lisp Library](#)
- [Download uLisp](#)
- [uLisp Sensor Library](#)
- [Forum](#)

**Self-contained Lisp computers**

- [Tiny Lisp Computer](#)
- [Lisp Badge](#)
- [Pocket Op Amp Lab](#)

**8/16-bit platforms**

- [Arduino Uno](#)
- [Arduino Mega 2560](#)
- [ATmega1284](#)
- [ATmega4809 boards](#)
- [AVR DA and DB series boards](#)

**32/64-bit platforms**

- [Arduino M0 Boards](#)
- [Adafruit M0 boards](#)
- [Adafruit QT-Py and Seeduino XIAO](#)
- [Adafruit M4 boards](#)
- [Adafruit PyGamer and PyBadge](#)
- [Adafruit nRF52840 boards](#)
- [BBC Micro:bit](#)
- [Calliope mini](#)
- [Maxim MAX32620FTHR](#)
- [Teensy 4.0 and 4.1](#)
- [RP2040 boards \*\*New!\*\*](#)
- [ESP32 boards](#)
- [ESP8266 boards](#)
- [Sipeed MAIX RISC-V boards](#)

**Simple examples**

- [Blinking primes](#)
- [Tweetmaze](#)
- [Mood light](#)
- [LED 8x8 dot-matrix display](#)
- [Simon game](#)
- [I2C clock](#)
- [Data logging](#)
- [Ringing the changes](#)
- [Driving DotStar RGB LEDs](#)
- [LCD character display](#)
- [DDS signal generator](#)
- [Temperature sensor](#)
- [Simple data plotter](#)
- [Thermocouple interface](#)
- [Benchmarks](#)

**Games**

- [Animals](#)
- [Eliza chatbot](#)
- [Simple arcade game](#)
- [Bulls & Cows game](#)
- [Mini text adventure game](#)
- [Larger examples](#)
- [Route finder](#)
- [Calculating with fractions](#)
- [Infinite precision arithmetic](#)
- [Scrolling text display](#)
- [Dot-matrix clock](#)
- [Graphics display interface in Lisp](#)
- [Plotting to a colour TFT display](#)
- [Ray tracing with uLisp](#)
- [GPS mapping application](#)
- [Query language](#)
- [uLisp GSM server](#)
- [A LoRaWAN node using uLisp](#)
- [Solving resistor networks](#)
- [Fast Fourier Transform](#)
- [Sudoku solver](#)
- [Prime number spiral](#)
- [Wi-Fi examples](#)
- [Automatic uLisp to C converter](#)
- [Simple object system](#)

**Assemblers**

- [AVR assembler overview](#)
- [AVR assembler examples](#)
- [AVR NeoPixel driver using assembler](#)
- [ARM assembler overview](#)
- [ARM assembler instructions](#)
- [ARM assembler examples](#)
- [ARM NeoPixel driver using assembler](#)
- [RISC-V assembler overview](#)
- [RISC-V assembler instructions](#)
- [RISC-V assembler examples](#)
- [Mandelbrot set using assembler](#)
- [Returning a floating-point result](#)
- [Tutorials](#)
- [Getting started](#)
- [Lists](#)
- [Expressions](#)
- [Defining functions](#)
- [Variables](#)
- [Strings](#)
- [Arrays](#)
- [Read, Evaluate, Print](#)
- [Garbage collection](#)
- [Tail-call optimization](#)
- [Saving and loading images](#)
- [Assembler and defcode](#)
- [Debugging uLisp](#)
- [Adding your own functions](#)
- [Converting between C and uLisp](#)
- [Porting uLisp to a new platform](#)
- [uLisp Builder](#)
- [uLisp Zero](#)

**Reference**

- [Language reference](#)
- [Floating-point extensions](#)
- [Graphics extensions](#)
- [Wi-Fi extensions](#)
- [Programming ARM registers](#)
- [Programming AVR registers](#)
- [Error messages](#)
- [Adding useful functions to uLisp](#)
- [Implementation](#)
- [Implementation](#)
- [Objects](#)
- [Symbols](#)
- [Built-in symbols](#)
- [Strings](#)
- [Arrays](#)
- [Read, Evaluate, Print](#)
- [Garbage collection](#)
- [Tail-call optimization](#)
- [Saving and loading images](#)
- [Assembler and defcode](#)
- [Debugging uLisp](#)
- [Adding your own functions](#)
- [Converting between C and uLisp](#)
- [Porting uLisp to a new platform](#)
- [uLisp Builder](#)
- [uLisp Zero](#)

**Other information**

- [About me](#)
- [Contact](#)
- [Legal stuff](#)

**RSS Feed**

- [uLisp news!](#)

**ULISP****GREAT WEB SITE****uLisp**

Home

 
**Getting started**

[Lisp for microcontrollers](#)  
[Performance](#)  
[Using uLisp](#)  
[Using uLisp from a terminal](#)  
[Debugging in uLisp](#)  
[Using the program editor](#)  
[SD card interface](#)  
[I2C and SPI serial interfaces](#)  
[Lisp Library](#)  
[Download uLisp](#)  
[uLisp Sensor Library](#)  
[Forum](#)

**Tutorials**

[Getting started](#)  
[Lists](#)  
[Expressions](#)  
[Defining functions](#)  
[Variables](#)  
[Strings](#)  
[Testing a result](#)  
[Manipulating lists](#)  
[Processing items in a list](#)  
[Recursion](#)  
[Returning functions](#)  
[Lisp for C programmers](#)  
[Thinking in a Lispy way](#)

**Reference**

[Language reference](#)  
[Floating-point extensions](#)  
[Graphics extensions](#)  
[Wi-Fi extensions](#)  
[Programming AVR registers](#)  
[Programming ARM registers](#)  
[Error messages](#)  
[Adding useful functions to uLisp](#)

**Implementation**

[Implementation](#)  
[Objects](#)  
[Symbols](#)  
[Built-in symbols](#)  
[Strings](#)  
[Arrays](#)  
[Read, Evaluate, Print](#)  
[Garbage collection](#)  
[Tail-call optimization](#)  
[Saving and loading images](#)  
[Assembler and defcode](#)  
[Debugging uLisp](#)  
[Adding your own functions](#)  
[Converting between C and uLisp](#)  
[Porting uLisp to a new platform](#)  
[uLisp Builder](#)  
[uLisp Zero](#)

ULISP

GREAT WEB SITE



uLisp

Home

## 8/16-bit platforms

[Arduino Uno](#)  
[Arduino Mega 2560](#)  
[ATmega1284](#)  
[ATmega4809 boards](#)  
[AVR DA and DB series boards](#)

## 32/64-bit platforms

[Arduino M0 Boards](#)  
[Adafruit M0 boards](#)  
[Adafruit QT-Py and Seeduino XIAO](#)  
[Adafruit M4 boards](#)  
[Adafruit PyGamer and PyBadge](#)  
[Adafruit nRF52840 boards](#)  
[BBC Micro:bit](#)  
[Calliope mini](#)  
[Maxim MAX32620FTHR](#)  
[Teensy 4.0 and 4.1](#)  
[RP2040 boards New!](#)  
[ESP32 boards](#)  
[ESP8266 boards](#)  
[Sipeed MAiX RISC-V boards](#)

## Assemblers

[AVR assembler overview](#)  
[AVR assembler examples](#)  
[AVR NeoPixel driver using assembler](#)  
[ARM assembler overview](#)  
[ARM assembler instructions](#)  
[ARM assembler examples](#)  
[ARM NeoPixel driver using assembler](#)  
[RISC-V assembler overview](#)  
[RISC-V assembler instructions](#)  
[RISC-V assembler examples](#)  
[Mandelbrot set using assembler](#)  
[Returning a floating-point result](#)

**Simple examples**

[Blinking primes](#)  
[Tweetmaze](#)  
[Mood light](#)  
[LED 8x8 dot-matrix display](#)  
[Simon game](#)  
[I2C clock](#)  
[Data logging](#)  
[Ringing the changes](#)  
[Driving DotStar RGB LEDs](#)  
[LCD character display](#)  
[DDS signal generator](#)  
[Temperature sensor](#)  
[Simple data plotter](#)  
[Thermocouple interface](#)  
[Benchmarks](#)

**Larger examples**

[Route finder](#)  
[Calculating with fractions](#)  
[Infinite precision arithmetic](#)  
[Scrolling text display](#)  
[Dot-matrix clock](#)  
[Graphics display interface in Lisp](#)  
[Plotting to a colour TFT display](#)  
[Ray tracing with uLisp](#)  
[GPS mapping application](#)  
[Query language](#)  
[uLisp GSM server](#)  
[A LoRaWAN node using uLisp](#)  
[Solving resistor networks](#)  
[Fast Fourier Transform](#)  
[Sudoku solver](#)  
[Prime number spiral](#)  
[Wi-Fi examples](#)  
[Automatic uLisp to C converter](#)  
[Simple object system](#)

**Self-contained Lisp computers**

[Tiny Lisp Computer](#)  
[Lisp Badge](#)  
[Pocket Op Amp Lab](#)

**Games**

[Animals](#)  
[Eliza chatbot](#)  
[Simple arcade game](#)  
[Bulls & Cows game](#)  
[Mini text adventure game](#)

**Other information**

[About me](#)  
[Contact](#)  
[Legal stuff](#)  
[RSS Feed](#)  
[uLisp news!](#)

## ULISP

## ACTIVE USER FORUM


[Home](#)   [Reference](#)   [Forum](#)


Do you want live notifications when people reply to your posts? [Enable Notifications.](#)


[all categories](#)   [Latest](#)   [Top](#)   [Categories](#)
[+ New Topic](#)

Topic	Category	Users	Replies	Views	Activity
<b>✉ Welcome to the uLisp Forum</b>					
The aim of this forum is to help you get the most out of uLisp, discuss what you'd like to see in future versions of uLisp, and share information about projects created with uLisp.			0	5.8k	May '16
Curious mention	General	  	2	411	5d
Adafruit nRF52840 Express + Sharp memory display	Platforms	 	1	59	14d
uLisp on the Teensy 4.0/4.1 now supports save-image	Platforms	 	1	86	28d
Assembler NeoPixel Driver for RP2040 Boards	Applications		0	65	28d
Improvements to uLisp for the RP2040	Platforms		0	102	30d
Adding QSPI ram to a Teensy 4.1 and uLisp Memory usage questions	Platforms	 	1	129	Feb 5
🔒 Enable Micro-SD Card on PyGamer	Platforms	 	5	122	Feb 2
🔒 Problems using uLisp with ESP32 Arduino core version 2.0.2	Bugs		2	127	Jan 30
🔒 uLisp ARM version 4.1 download page broken?	Site Feedback	 	2	90	Jan 29

## ULISP

## ACTIVE GITHUB

**m-g-r / ulisp-esp-m5stack** Public  
forked from [technoblogy/ulisp-esp](#)

[Pin](#) [Watch 1](#) [Fork 28](#) [Star 3](#)

[Code](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [...](#)

Pulse
Contributors
Community
Traffic
Commits
Code frequency
Dependency graph
Network
Forks

**forks**

- technoblogy / ulisp-esp
- bkmit / ulisp-esp
- braechnov / ulisp-esp
- brown131 / ulisp-esp
- danja / ulisp-esp
- davidbroughsmith / ulisp-esp
- dsmvwld / ulisp-esp
- elbow / ulisp-esp
- goncha / ulisp-esp
- hemml / ulisp-esp
- inquisitor / ulisp-esp
- interpreters-compilers / ulisp-esp
- jig / ulisp-esp
- jmbruno / ulisp-esp
- jurov / ulisp-esp
- Kaef / ulisp-esp
- kwccoin / ulisp-esp
- lupyuen / ulisp-bl602
- m-g-r / ulisp-esp-m5stack
- cycle-uber-space / ulisp-esp-m5stack
- madand / ulisp-esp
- nelsonov / ulisp-esp
- pluizer / ulisp-esp
- sake / ulisp-esp
- shreknal / ulisp-esp
- simon2e / ulisp-esp
- theearthur / ulisp-esp
- VaclavSynacek / ulisp-esp-32c3-riscv
- xrossco / ulisp-esp

## CONNECT TO ULISP

Linux with screen:

```
screen /dev/ttyUSB0 115200
```

macOS:

```
screen /dev/tty.usbserial-55D2D232BE 115200
```

Emacs:

```
(defun ulisp-terminal-start ()
  (serial-term "/dev/ttyUSB0" 115200)
  (term-line-mode))
```

```
M-: (ulisp-terminal-start)
```

D E M O   T I M E

**M5STACK STAND-ALONE ULISP COMPUTER****MAX-GERD RETZLAFF, MARCH 2021**

Batteries are included, which makes it self-contained and take-along.



**GPS-BASED SPEEDOMETER AND CLOCK****MAX-GERD RETZLAFF, MAY 2021**

A replacement for the speedometer clock of my Vespa ET4  
which often fails as it just uses a tiny battery.



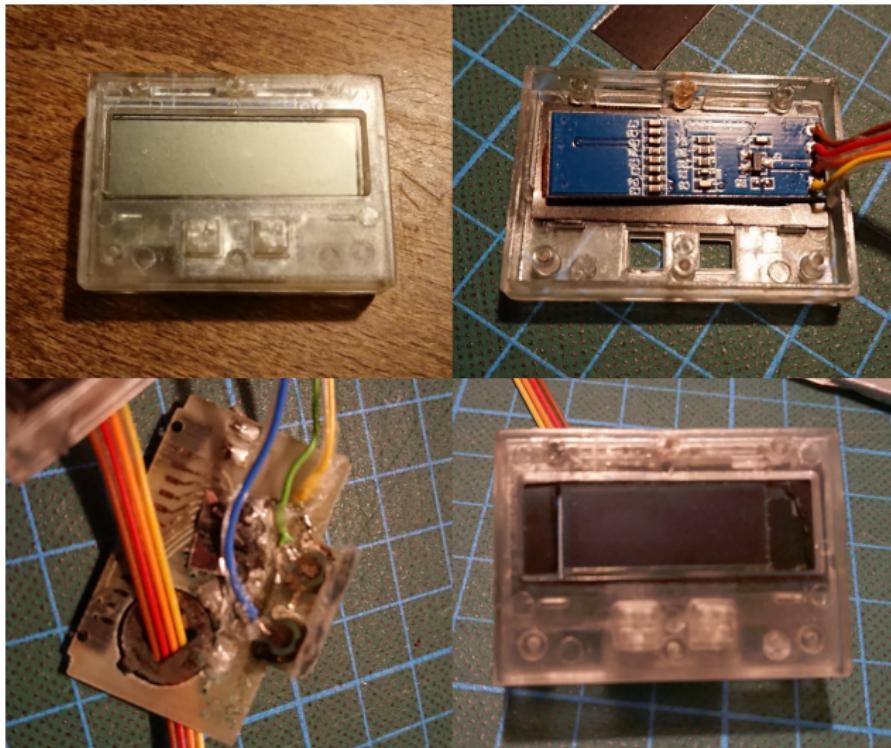
## GPS-BASED SPEEDOMETER AND CLOCK

MAX-GERD RETZLAFF, MAY 2021



## GPS-BASED SPEEDOMETER AND CLOCK

MAX-GERD RETZLAFF, MAY 2021



## GPS-BASED SPEEDOMETER AND CLOCK

MAX-GERD RETZLAFF, MAY 2021



PRESENTER FOR “NEXT PAGE”

MAX-GERD RETZLAFF, MARCH 2022

On the plane trip to ELS 2022 . . . :-)



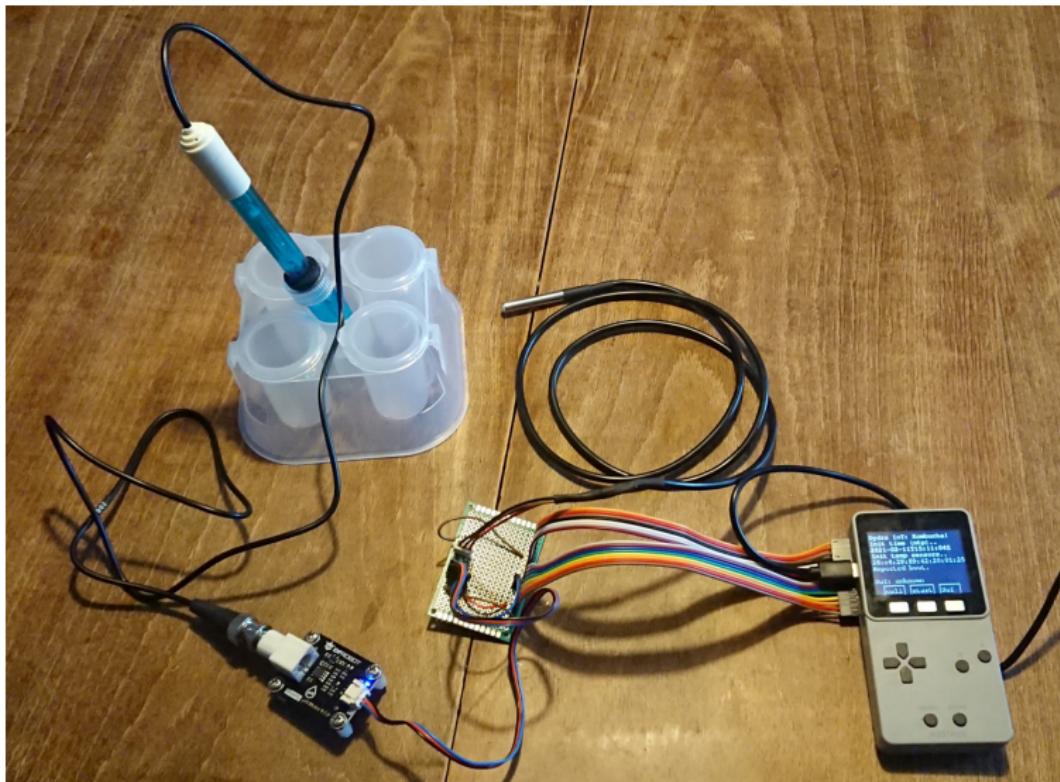
## ULISP WITH ANDROID

MARCH 2022



## KOMBUCHA SENSOR DEVICE

MAX-GERD RETZLAFF, FEB. 2021



## KOMBUCHA SENSOR DEVICE

MAX-GERD RETZLAFF, FEB 2021



## KOMBUCHA SENSOR DEVICE

MAX-GERD RETZLAFF, FEB 2021



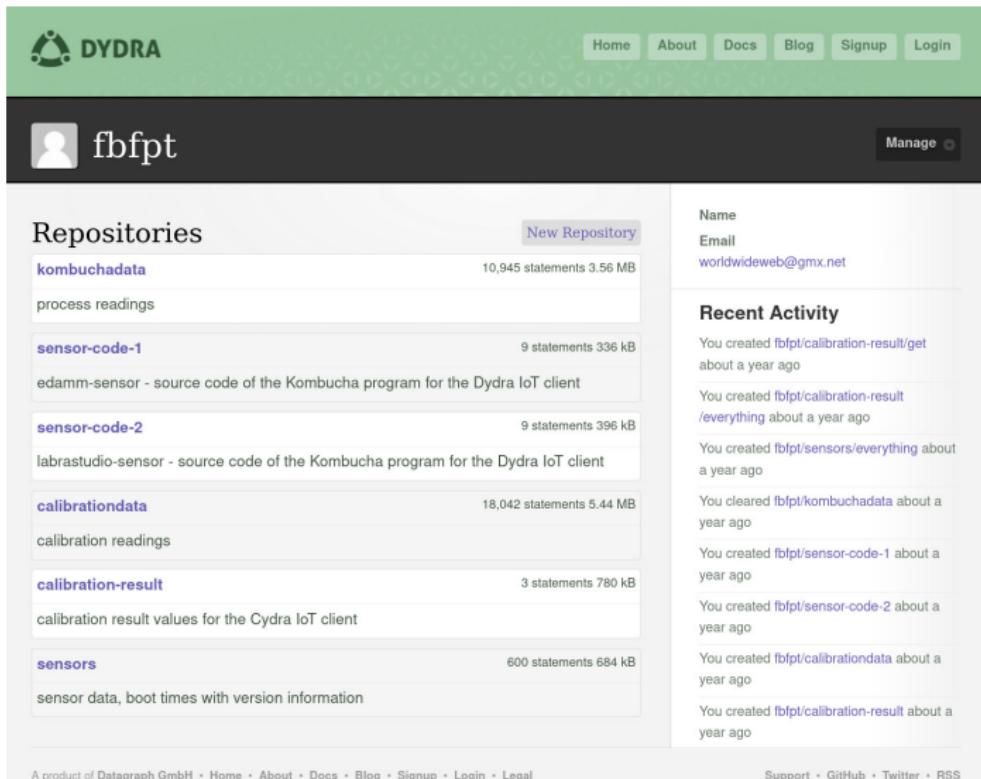
## KOMBUCHA SENSOR DEVICE

MAX-GERD RETZLAFF, FEB 2021



## KOMBUCHA SENSOR DEVICE

MAX-GERD RETZLAFF, FEB. 2021



The screenshot shows a user interface for managing repositories on the Dydra platform. At the top, there is a navigation bar with links for Home, About, Docs, Blog, Signup, and Login. Below the navigation bar, the user's profile picture and name "fbfpt" are displayed, along with a "Manage" button.

**Repositories**

Repository Name	Description	Statements	Size
kombuchadata	process readings	10,945 statements	3.56 MB
sensor-code-1	edamm-sensor - source code of the Kombucha program for the Dydra IoT client	9 statements	336 kB
sensor-code-2	labrastudio-sensor - source code of the Kombucha program for the Dydra IoT client	9 statements	396 kB
calibrationdata	calibration readings	18,042 statements	5.44 MB
calibration-result	calibration result values for the Cydra IoT client	3 statements	780 kB
sensors	sensor data, boot times with version information	600 statements	684 kB

**Name**  
Email  
worldwideweb@gmx.net

**Recent Activity**

- You created [fbfpt/calibration-result/get](#) about a year ago
- You created [fbfpt/calibration-result/everything](#) about a year ago
- You created [fbfpt/sensors/everything](#) about a year ago
- You cleared [fbfpt/kombuchadata](#) about a year ago
- You created [fbfpt/sensor-code-1](#) about a year ago
- You created [fbfpt/sensor-code-2](#) about a year ago
- You created [fbfpt/calibrationdata](#) about a year ago
- You created [fbfpt/calibration-result](#) about a year ago

A product of Datagraph GmbH • Home • About • Docs • Blog • Signup • Login • Legal

Support • GitHub • Twitter • RSS

## KOMBUCHA SENSOR DEVICE

MAX-GERD RETZLAFF, FEB. 2021



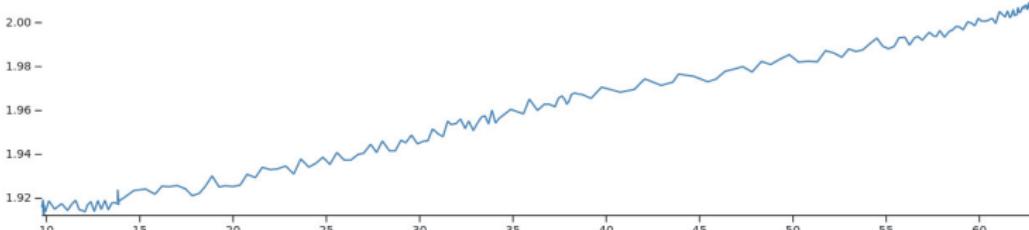
By m-g-r You Edited Mar 20 Fork of ph sensor calibration runs

## ph sensor calibration runs

All data is stored in: <https://nl4.dydra.com/fbfpt/calibration> See query:  
[https://nl4.dydra.com/fbfpt/calibration/@query#calibration\\_all\\_get](https://nl4.dydra.com/fbfpt/calibration/@query#calibration_all_get)

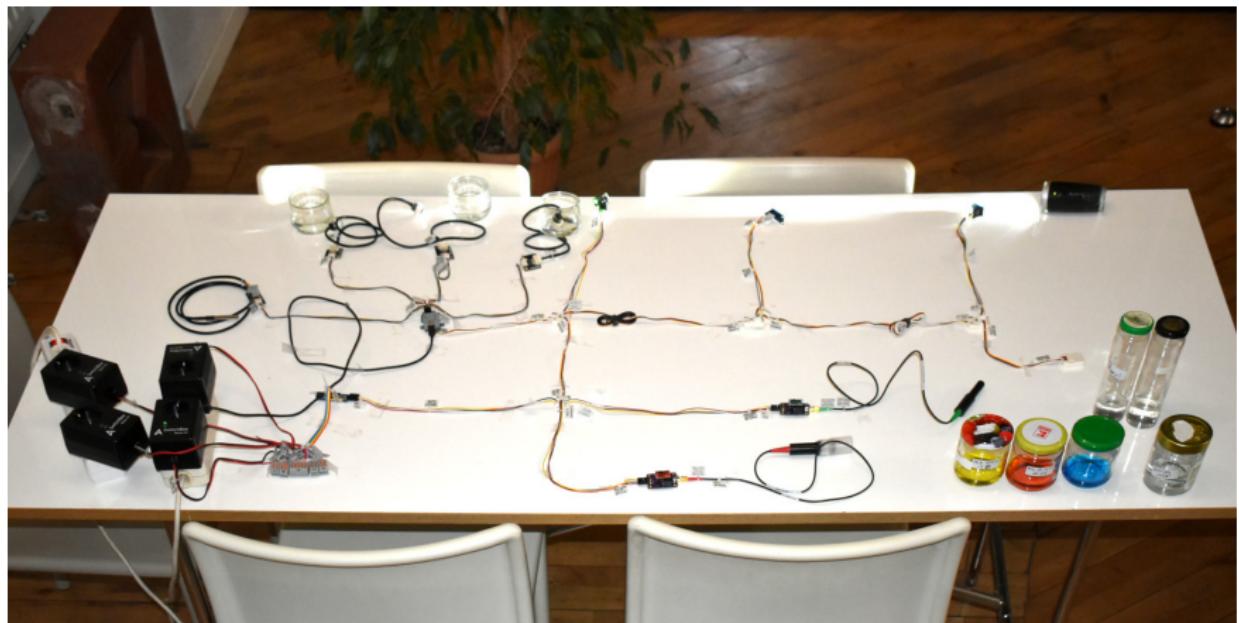
### interactive visualization

```
showfile = ► FileAttachment {name: "ph4.01-red01.csv"}  
  
// Apera Instruments calibration solutions:  
//showfile = ph7g  
showfile = ph4r  
//showfile = ph10b  
//showfile = ph7g2  
// dfrobot calibration solutions:  
//showfile = ph4  
//showfile = ph7
```



## SENSOR DEVICE FOR AN AUTOMATED IOT DEVICE

MAX-GERD RETZLAFF, NOV. 2021



## SENSOR DEVICE FOR AN AUTOMATED IOT DEVICE

MAX-GERD RETZLAFF, NOV. 2021

- ten environmental sensors
- four controllable power sockets to activate environmental control measures
- Wi-Fi provisioned via Bluetooth (BLE)
- communication with a REST backend over Wi-Fi via JSON and HTTPs
- JSON protocol based on RelaxNG schemas (json-rnc)
- sends sensor readings to backend
- retrieves commands from a controlling smartphone application (from a different party) to control the actors or to calibrate the more complicated sensors
- liquid temperature sensor connected via one wire bus
- most sensors and actors connected via I2C bus  
with libraries completely written in uLisp

# DISCUSSION



THANK YOU





# APPENDIX

**SOLAR CLOCK****DAVID JOHNSON-DAVIES, 2021**

A low-power LCD clock, with an average current consumption as about  $9\mu\text{A}$  at 3.3V, designed to run from a solar cell allowing indefinite off-grid operation.

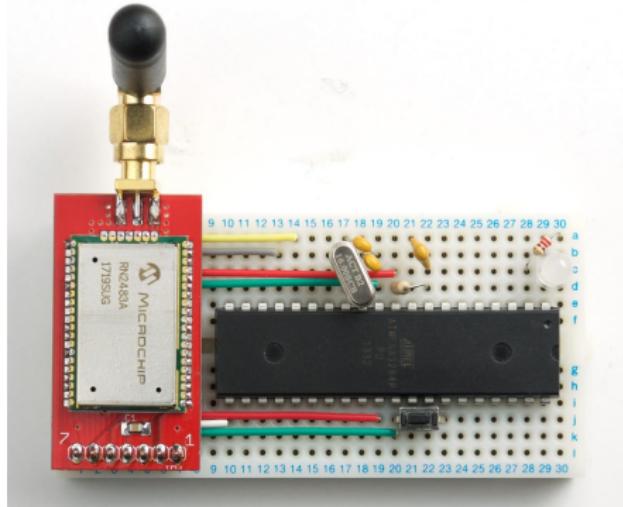


## LORAWAN NODE USING ULISP

DAVID JOHNSON-DAVIES, 2017

A LoRaWAN node, based on a Microchip RN2483A interfaced to an ATmega1284, and controlled by a program written in uLisp, capable of sending data to The Things Network.

See also <http://www.ulisp.com/show?1XWL>.



**LISP BADGE****DAVID JOHNSON-DAVIES, 2019**

A self-contained computer with its own display and keyboard, based on an ATmega1284 microcontroller, that can be programmed in uLisp, a subset of Common Lisp.

See also <http://www.ulisp.com/show?2LOC>.



**ULISP****IMPLEMENTATION ACKNOWLEDGEMENTS**

Nurullah Akkaya's article "A micro-manual for LISP Implemented in C" in early stages of designing the interpreter.

Bob Nystrom's article "Baby's First Garbage Collector" for the garbage collector.

Section "A Properly Tail-Recursive Interpreter" in Peter Norvig's classic book "Paradigms of Artificial Intelligence Programming" for help with making the interpreter tail-recursive.

User clawson on AVR Freak (<https://www.avrfreaks.net/>) when designing a procedure name lookup table in program memory.

# B I B L I O G R A P H Y

## R E F E R E N C E S

DOIT 2018

DOIT, Shenzhen Sibo Zhilian Technology Co Ltd: *Picture of DOIT ESP32 DEVKIT V1.* Promotional picture, 2018. <http://doit.am>.

Johnson-Davies 2014a

JOHNSON-DAVIES, David: *uLisp implementation page.* Web page, 2014-2022. <http://www.ulisp.com/show?1AWG>.

Johnson-Davies 2014b

JOHNSON-DAVIES, David: *uLisp specification page.* Web page, 2014-2022. <http://www.ulisp.com>.

Johnson-Davies 2017

JOHNSON-DAVIES, David: *Picture of a LoRaWAN Node using uLisp by David Johnson-Davies*. Photography, 2017. <http://www.ulisp.com/show?1XWL>.

Johnson-Davies 2019

JOHNSON-DAVIES, David: *Picture of the Lisp Badge by David Johnson-Davies*. Photography, 2019. <http://www.ulisp.com/show?2LOC>.

Johnson-Davies 2021a

JOHNSON-DAVIES, David: *Design objective of uLisp*. E-mail, 2021. Personal correspondence, Feb 5th, 2021.

Johnson-Davies 2021b

JOHNSON-DAVIES, David: *On the name of uLisp*. E-mail, 2021. Personal correspondence, May 6th, 2021.

Johnson-Davies 2021c

JOHNSON-DAVIES, David: *Picture of David Johnson-Davies*. Photography, 2021. Personal correspondence.

Johnson-Davies 2021d

JOHNSON-DAVIES, David: *Picture of the Solar Clock by David Johnson-Davies*. Photography, 2021. Personal correspondence.

M5Stack 2020

M5STACK: *Picture of M5Stack Atom Lite ESP32*. Promotional picture, 2020.

Retzlaff 2021a

RETZLAFF, Max-Gerd: *Picture of a Kombucha sensor device*. Photography, 2021.

Retzlaff 2021b

RETZLAFF, Max-Gerd: *Picture of a sensor device for an automated IoT device*. Photography, 2021.

Retzlaff 2021c

RETZLAFF, Max-Gerd: *Picture of assembly of the Scooter Lisp computer.* Photography, 2021.

Retzlaff 2021d

RETZLAFF, Max-Gerd: *Picture of M5Stack ESP32 Basic Core.* Photography, 2021.

Retzlaff 2021e

RETZLAFF, Max-Gerd: *Picture of M5Stack stand-alone uLisp computer.* Photography, 2021.

Retzlaff 2021f

RETZLAFF, Max-Gerd: *Picture of replaced speedometer clock with Scooter Lisp.* Photography, 2021.

Retzlaff 2021g

RETZLAFF, Max-Gerd: *Picture of replaced speedometer clock with Scooter Lisp #2.* Photography, 2021.

Retzlaff 2021h

RETZLAFF, Max-Gerd: *Picture of the replacement of the display of the speedometer clock.* Photography, 2021.

Retzlaff 2021i

RETZLAFF, Max-Gerd: *Screenshot of an Observable notebook by Datagenous.* Screenshot picture, 2021. <http://corp.datageno.us>.

Retzlaff 2021j

RETZLAFF, Max-Gerd: *Screenshot of repository overview of the Dydra graph database by Datagraph.* Screenshot picture, 2021. <http://www.dydra.com>.

Retzlaff 2021k

RETZLAFF, Max-Gerd: *Six pictures of the display of the Kombucha sensor device.* Photography, 2021.

Retzlaff 2022a

RETZLAFF, Max-Gerd: *Picture of a presenter device.* Photography, 2022.

Retzlaff 2022b

RETZLAFF, Max-Gerd: *Picture of twelve defsystem atoms.* Photography, 2022.

Retzlaff 2022c

RETZLAFF, Max-Gerd: *Picture of using an Android smartphone to connect to an M5Stack Atom Lite ESP32 (while on an air plane).* Photography, 2022.

– End of Document –