My4TH light
Construction Manual

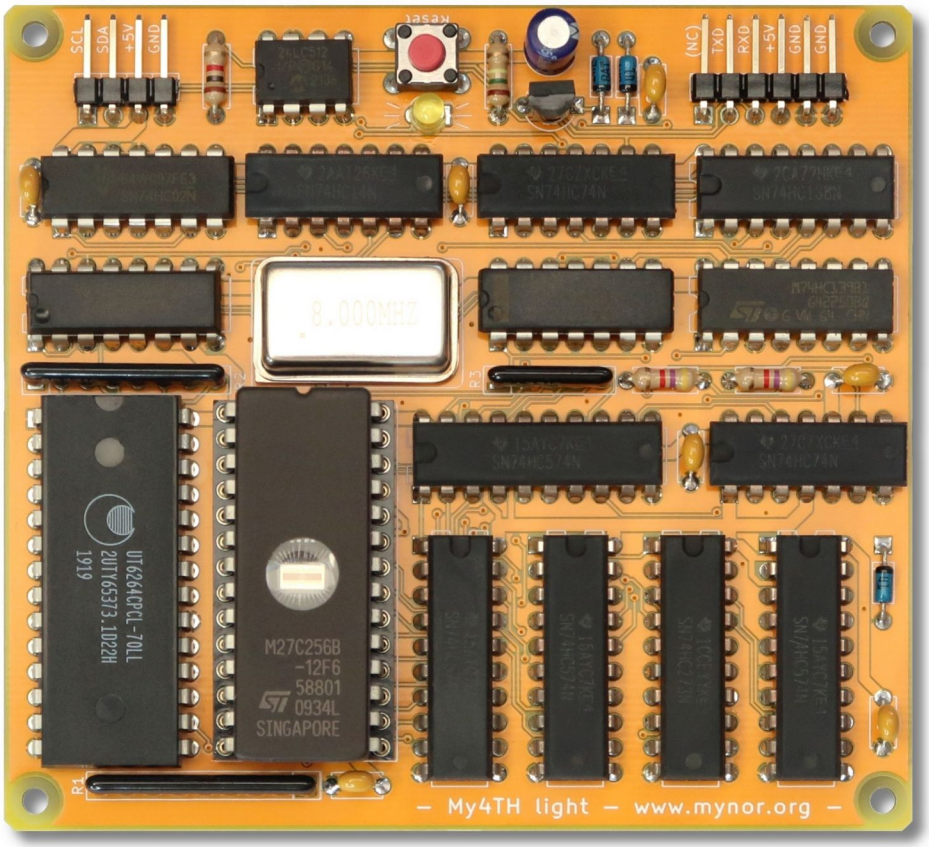
Dennis Kuschel
dennis_k@freenet.de
2023-11-23

My4TH light, together with My4TH XS, is a simplified version of the My4TH computer board. I developed My4TH specifically to run FORTH, an old but very efficient programming language for small computer systems. My4TH light demonstrates how few logic chips are needed to build a working FORTH computer system. The "CPU" consists entirely of discrete logic chips, and to simplify the ALU-less design, a single NOR gate performs all calculations.

This document will help you assemble your own My4TH light board. Visit www.mynor.org for more documentation.

My4TH light Specifications:

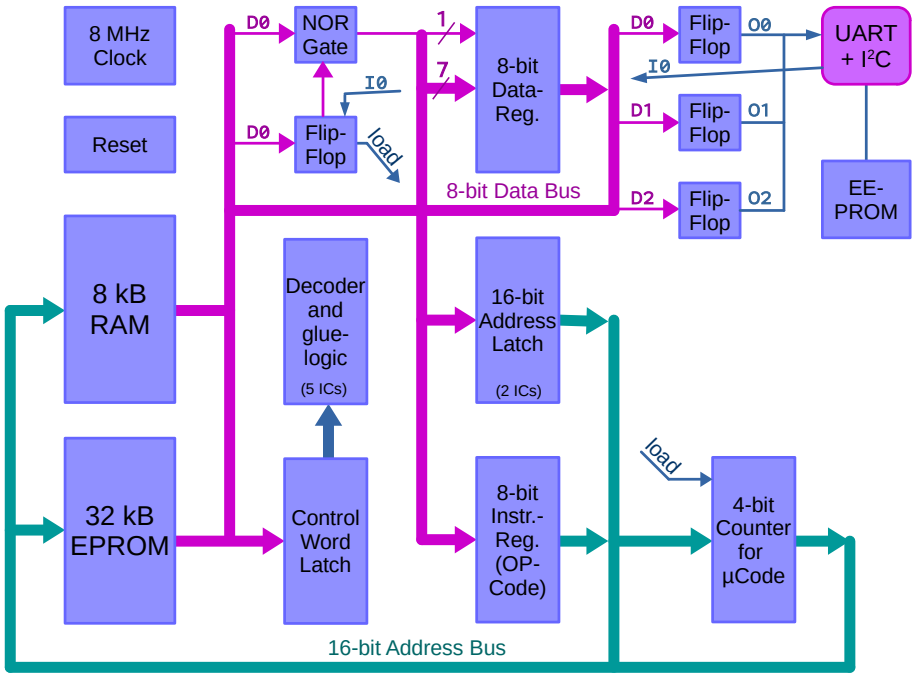
- ✔ 8-bit Computer System with Von-Neumann architecture, 8 MHz system clock
- ✔ 1-bit NOR gate logic unit (no “ALU”), 5200 8-bit additions per second @ 8 MHz
- ✔ 32 KB EPROM, 64 kB EEPROM and 8 KB SRAM
- ✔ Bit-banging serial port (4800 baud) and fast I^2C (up to 83 kHz at 8 MHz system clock)
- ✔ The power consumption is only 0.3 W at 8 MHz system clock
- ✔ Supports the FORTH 2012 Standard. Implements the Forth Core words, Core extension words, the Double-Number word set, the Block word set, Floating Point words, and many other words.



My4TH light, the light-weight version of My4TH!

System Overview

Below you can see a block diagram of My4TH light. The most important chips are the two memories, a 32 KB EPROM and a 8 KB RAM. The EPROM contains the microcode that controls the flow of data through the system, as well as the FORTH interpreter code. The RAM contains CPU registers, the stack memory, and the FORTH application code and data. As you can see, there is no CPU and no ALU. The computer is built from the simplest logic chips - the most complex chip in the system is the EEPROM, which serves as non-volatile memory for the FORTH code written by the user. All calculations requiring two operands, such as addition, subtraction, AND, OR and XOR, are performed using a single NOR gate, while other simple operations, such as rotating and inverting a byte, are performed using a look-up table. My4TH light has only one digital input pin, shared between the UART and the I²C bus. A simple mechanism prevents data on the I²C from being corrupted if data is sent from a serial terminal to the My4TH computer while an I²C transaction is in progress: The UART RXD line is disabled while the UART TXD line is set low. Thus, while the I²C is active, a series of long low pulses is sent over TXD, but these pulses are discarded by the receiver because they are classified as illegal data or so-called break signals.

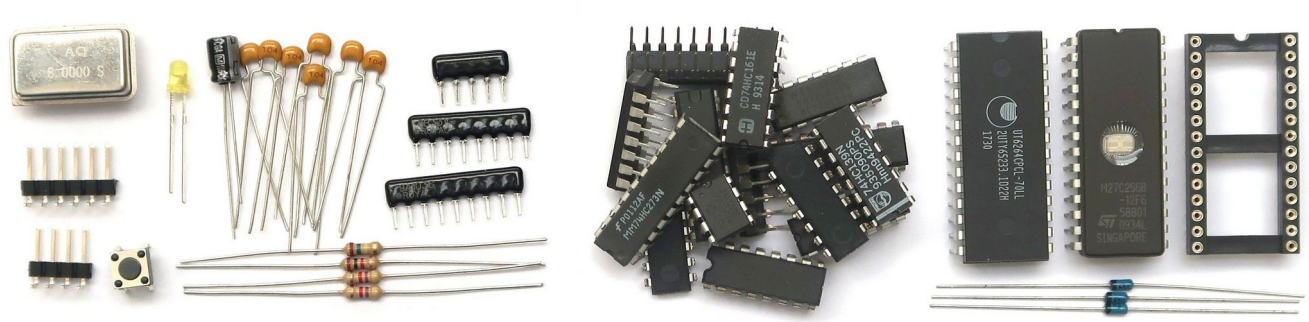


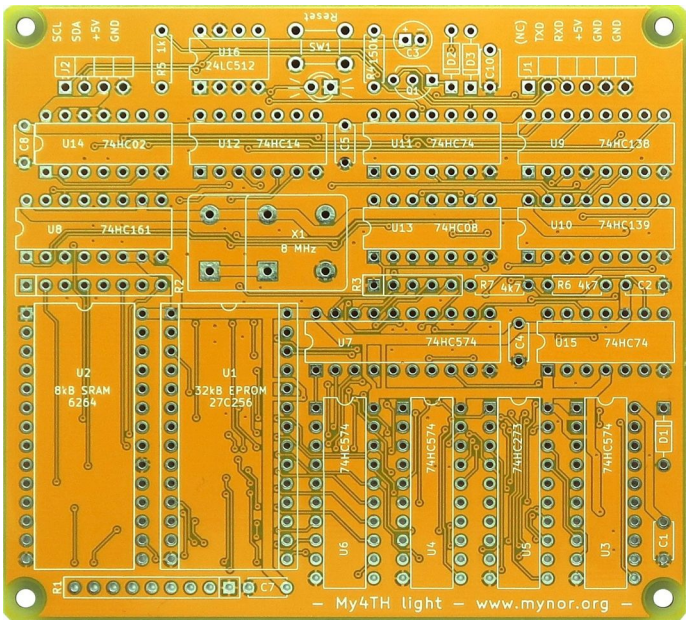
The following logic chips were used to construct My4TH light:

Chip	Description	Function in My4TH light
74HC02	Quad 2-input NOR gate	NOR-gate for calculations, glue-logic
74HC08	Quad 2-input AND gate	Glue-logic
74HC14	Hex inverter with Schmitt-trigger inputs	Clock buffer, reset generator, glue-logic
74HC74	Dual D-type flip-flop with set/reset	Latch for D0 on data bus, TXD/SCL/SDA signals
74HC138	3-to-8 line decoder / demultiplexer	Data target decoder
74HC139	Dual 2-to-4 line decoder / demultiplexer	Data source decoder
74HC161	Synchronous 4-bit binary counter	Micro instruction counter with parallel set input
74HC273	Octal D-type flip-flop with reset	Control signal buffer for inputs of 74HC138/139
74HC574	Octal D-type flip-flop	Various data registers

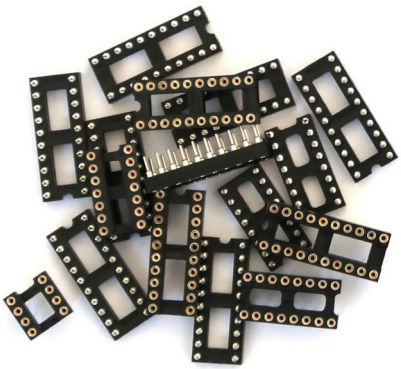
Required Components

Here is an overview of all the required components:



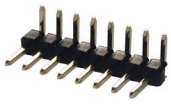
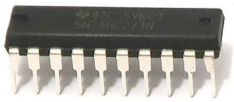
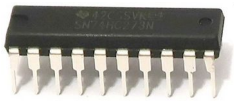


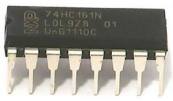
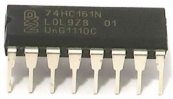
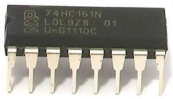
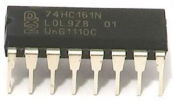
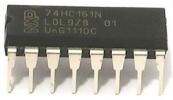
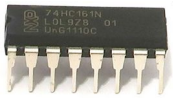
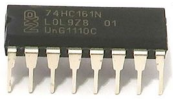
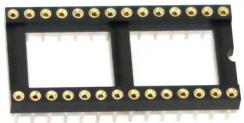
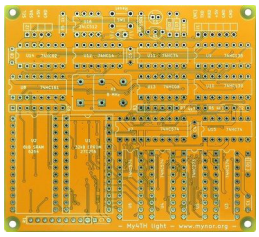
You may also use sockets for all of the ICs. This is optional, but it can simplify troubleshooting a lot:



The complete bill of material is listed in the table on the next page. Note that the table contains also the EPROM 27C256. When you are not able to „burn“ or „program“ it by yourself, please contact me, I can help you with that.

Bill of Material

Reference	Qty	Picture	Value	Mouser P/N www.mouser.com	Reichelt P/N www.reichelt.de
C1 C2 C4 C5 C6 C7 C8	7		100 nF (X7R / 5 mm)	SR215C104K	X7R-5 100N
C3	1		10µF	ECE-A1HKA100	AK 10U 50
D1 D2 D3	3		BAT41	BAT41-TAP	BAT 41
D4	1		LED, 3mm	151031YS06000	EVL 204-10UYD
J1	1		6 pin header, right angle	Molex 22-28-8060	SL 1X40W 2,54
J2	1		4 pin header, right angle	Molex 22-28-8040	SL 1X40W 2,54
Q1	1		BS170	BS170D27Z	BS 170
R1	1		8x 10k, 9 pins	4609X-101-103LF	SIL 9-8 10K
R2	1		4x 330, 8 pins	4608X-102-331LF	SIL 8-4 330
R3	1		4x 330, 5 pins	4605X-101-331LF	SIL 5-4 330
R4	1		150k	CFR-25JR-52150K	1/4W 150K
R5	1		1k	CFR-25JR-52-1K	1/4W 1,0K
R6	1		4k7	CFR-25JR-52-4K7	1/4W 4,7K
SW1	1		SW_Push	TL1105AF160Q	TASTER 3301
U1	1		27C256 EPROM 32KB (55 - 100 ns)	AT27C256R-70PU (OTP PROM)	27C256-100 (EPROM)
U2	1		6264 SRAM 8KB (55 - 70 ns)	AS6C6264-55PCN	6264-70
U3 U4 U6 U7	4		74HC574	SN74HC574N	74HC 574
U5	1		74HC273	SN74HC273N	74HC 273

U8	1		74HC161	SN74HC161N	74HC 161
U9	1		74HC138	SN74HC138N	74HC 138
U10	1		74HC139	SN74HC139N	(74HC 139, no more available)
U11, U15	2		74HC74	SN74HC74N	74HC 74
U12	1		74HC14	SN74HC14N	74HC 14
U13	1		74HC08 (see text)	SN74HC08N	74HC 08
U14	1		74HC02	SN74HC02N	74HC 02
U16	1		24LC512 24LC1026 (see text)	24LC512-I/P 24LC1026-I/P	24LC512-I/P
X1	1		8 MHz	MXO45-3C-8M0000	OSZI 8,000000
Socket	1		DIP28 for EPROM	110-47-628-41-001000	GS 28P
PCB Raw Card	1		Use provided gerber files (in zip file) and order the PCB at jlcpcb.com		

You can buy the components at mouser.com, digikey.com, reichelt.de and many other websites. Regarding the PCB raw card I made very good experiences with JLCPCB in China. Please note that the minimum order quantity is 5 boards, for a price of $2.00 plus shipping costs. That‘s really cheap!

Notes about certain components

EPROM

With the components listed in the BOM you should be able to run My4TH light at 8 MHz without any problems. If you cannot get an EPROM or OTP PROM with an access time of 100 ns or faster, you can work around this by shifting the timing in the system a bit. For example, if you use an EPROM with 120ns access time, this may also work, but it tends to cause instability. Try replacing the 74HC08 (U13) with a 74AC08. If this does not help, you can also try replacing the 74HC161 (U8) with its 74AC counterpart. I have found that the address lines are sensitive to touch. Try not to touch the pins on the left side of the SRAM. Touching the address pins will add a small amount of capacitance to the signals, which will shift the signal timing in the wrong direction.

EEPROM

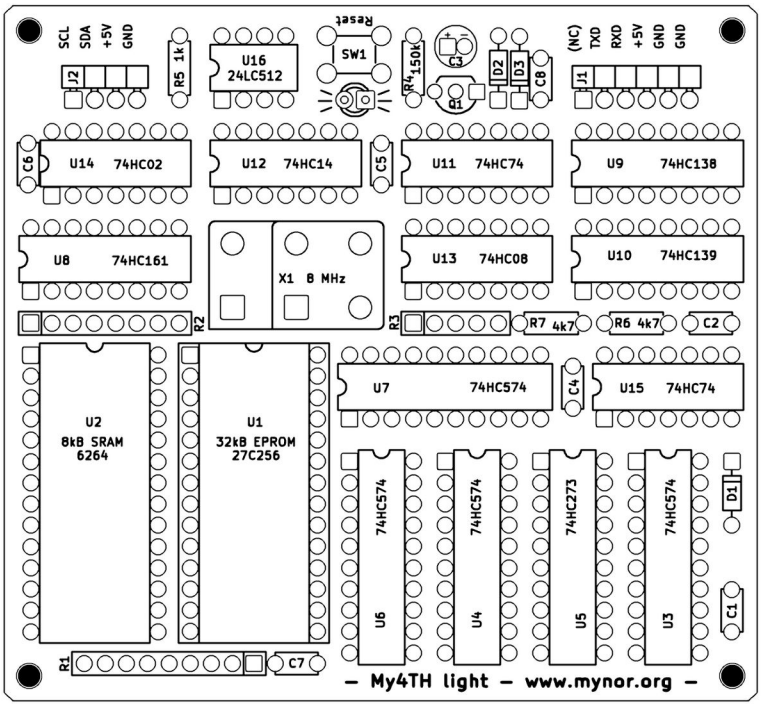
The board can be equipped with an EEPROM for Forth block storage. This is similar to storing data in sectors on a hard disk. This EEPROM is optional, but I recommend to use at least a small 24LC512 EEPROM. My4TH light automatically detects the available EEPROM memory as long as an EEPROM of the following type is installed: 24LC512, 24AA512, 24LC1026 or 24AA1026.

You can add more memory by connecting more EEPROMs to the external I^2C bus J2. Make sure that each EEPROM has a unique device address. This can be difficult because the 24xx1026 EEPROMs occupy two addresses (from the bus side of few, they contain two 24LC512 chips in one package).

The maximum memory configuration is 8x 24LC512 (8-bit I^2C addresses 0xA0, 0xA2, 0xA4, 0xA6, 0xA8, 0xAA, 0xAC and 0xAE), resulting in 512 KB of non-volatile blocks.

Board Assembly

The picture below shows the position of each part. Start with soldering the low components, in the following order: Header J1 and J2, IC sockets, ICs without socket and resistors. After that, continue with the resistor networks R1 - R3, the capacitors, the transistor and the LED. At last, mount the the switch and the clock oscillator.



First Test

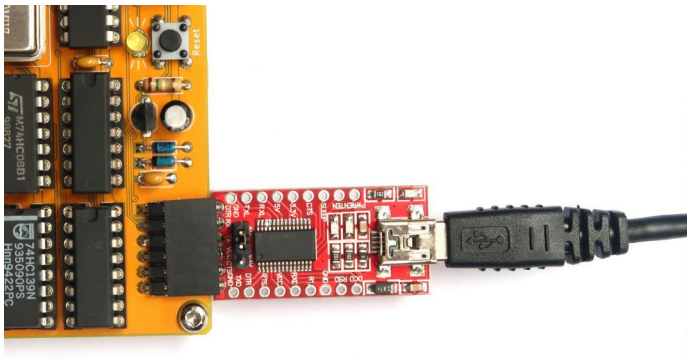
1. Review your work
Before you apply power to the My4TH light board for the first time, please check all components on the board to make sure they are the correct ones and that their orientation is correct. Check that the EPROM is correctly inserted into the socket and that no pin is bent and thus misses the associated receptacle. Also check the solder joints for missing connections or short circuits.

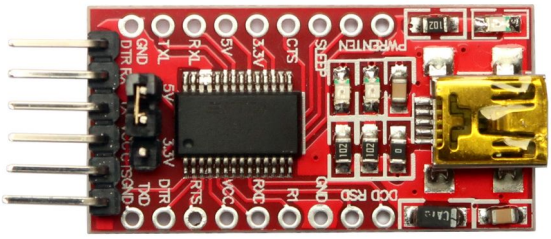
2. Power Supply
My4TH light can be powered via the +5V/GND - pins of the headers J1 or J2. I recommend using a „real“ 5V power supply (mains adapter). Tests have shown that the cheap USB chargers for smartphones often do not work properly. They are made to charge batteries, not for supplying devices directly. I have also tested USB power-banks. Some power-banks work, some not. You have to find out for yourself. The pinout of J1 matches the pinout of the common FTDI USB to UART bridge PCB breakout boards, so it is also possible to power the My4TH light board via a USB port.

3. Apply Power
Now switch on the power supply. Observe the LED. If everything is correct, the LED should now flicker a little and then light up. If you press the reset button, it flickers again and then stays on.

If the LED does not flicker, check all components and your soldering. Try running the board at a lower frequency, e.g. with a 4 MHz oscillator (the serial baud rate will then drop to 2400 baud). If the computer works now, you have a serious timing problem with some of the installed components. Please check the SRAM and the EPROM first. To compensate for a slow EPROM, try replacing U8 and U13 with their 74AC counterparts.

4. RS232 Terminal
My4TH light provides a serial interface with TTL level, i.e. 0 to +5V. Therefore it is not possible to connect My4TH light directly to the RS232 port of a PC. Instead you need some kind of level shifter, for example the good old MAX232. Or, in modern times, you can use a USB to serial converter chip like the FTDI FT232RL, which already supports the correct voltage levels. Fortunately, USB-to-serial converter breakout boards are widely available, such as the "FTDI Basic Breakout 5V" board and clones. I bought a clone board from Amazon, shown in the pictures below. The pinout matches the pins of J1, so the My4TH light board can also be powered from this board. On your PC, set your terminal program to 4800 baud, 8 bits, 1 stop bit and no handshake, and you should be able to communicate with your My4TH light board.





The picture shows a board that can be configured for 3.3V or 5V operation.
The jumper must be set to the 5V position.

Example FORTH Programs

Hello world:

```
: hello .” Hello world!” ;
hello
```

Hello world in a loop:

```
: hello 10 0 do cr .” Hello world!” loop ;
hello
```

Print square numbers:

```
: squares 182 1 do I dup * . loop ;
squares
```

Pascal’s triangle:

```
: PascTriangle cr dup 0 ?do 1 over 1- i - 2* spaces i 1+ 0
  ?do dup 4 .r j i - * i 1+ / loop cr drop loop drop ;
13 PascTriangle
```

99 bottles of beer:

```
:noname dup . ." bottles" ;
:noname       ." 1 bottle" ;
:noname       ." no more bottles" ;
create bottles , , ,
: .bottles  dup 2 min cells bottles + @ execute ;
: .beer     .bottles ."  of beer" ;
: .wall     .beer ."  on the wall" ;
: .take     ." Take one down, pass it around" ;
: .verse    .wall cr .beer cr 1- .take cr .wall cr ;
: verses    begin cr .verse ?dup 0= until ;
99 verses
```

FORTH Quick Start:

If you are not yet familiar with FORTH, I highly recommend reading “Starting FORTH”:
https://www.forth.com/starting-forth/

My4TH light has a built-in text editor to edit FORTH screens. For example, enter

```
5 edit
```

to edit screen number 5. To load, compile and execute the screen, enter

```
5 load
```

Further Readings

FORTH language reference for My4TH : My4TH-light_Forth_Glossary.pdf

How to exchange data with the PC : My4TH_Data_Exchange.pdf