

# eLuaBrain

32-bit Cortex-M3 autonomous computer running eLua

- Motivation
- Overview
- Hardware design
- Software design
- Future development
- More information
- Questions

# Motivation

- Hardware/software educational platform
  - Learn about hardware and software in parallel
  - Learn to cope with low resource systems
  - Simple, easy to understand structure
  - Easily hackable and extendable
  - Fully integrated (compiler, editor, help system)
  - Powerful, multi layer API
  - Good community support
- Programmable automation controller
  - Lower cost/lower power than PC/embedded Linux
  - Easier to program than a PC/embedded Linux
  - Easy to interface and extend

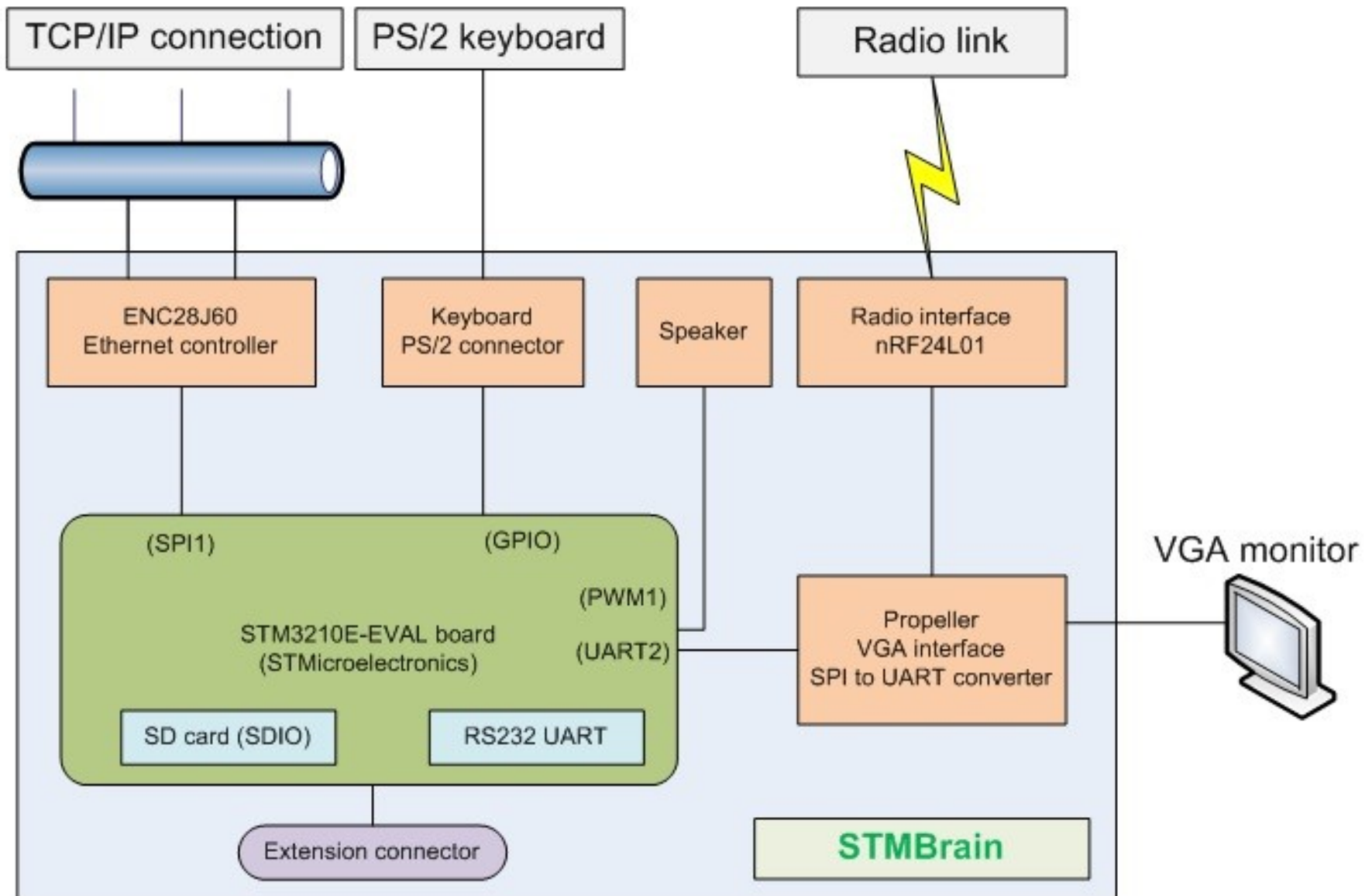
# Overview

- What is eLuaBrain?
  - 32-bit completely **autonomous computer**
  - Low cost and low power
  - **On-board Lua interpreter**
  - Open source (**MIT**)
  - Built around a Cortex-M3 MCU
  - Direct video output (**VGA**), PS/2 keyboard input
  - Connectivity: Ethernet, UART, CAN, 2.4GHz radio
  - Storage: SD card
- eLuaBrain is not:
  - A complete operating system
  - Another Arduino clone

# Overview

- Optimized for developing
  - Integrated Lua compiler and interpreter
  - Integrated text editor and help system
  - Integrated shell for simple operations
  - Multiple file systems (ROM, FAT, network (UDP))
  - Rich set of APIs
    - Low level access to all memory
    - High level APIs for peripherals
- TCP/IP stack and API optimized for low resource systems
- Very short (less than one second) boot time
- Easily portable to other microcontrollers

# Block diagram



# The hardware

- Base board: STM3210E-EVAL from ST Microelectronics
  - MCU: STM32F103ZET6 (Cortex-M3 @ 72 Mhz)
  - Internal memory: 512k Flash/64k RAM
  - Serial bootloader (**WIN!**)
  - External memory: **1M RAM**/64M NAND/16M NOR
  - Headers for all CPU pins (another **WIN!**)
  - Loads of peripherals (2xUART, USB, CAN, DAC, LCD)
- Drawbacks:
  - No Ethernet :(
  - Unable to act as a VGA video generator
  - Expensive (but OK for a prototype)
  - Too many peripherals. Really.

# The hardware

- Specific eLuaBrain hardware on a 'daughterboard'
- Wired Ethernet connection
  - ENC28J60 Ethernet controller with SPI interface
  - Relatively easy to use
  - 10Mbps only, but good enough
- Radio interface:
  - nRF24L01(+) from Nordic Semiconductor
  - Low power 2.4GHz transceiver chip
  - On-chip packet handling
  - Ideal for wireless automation
- Basic PWM sound generation

# The hardware

- Video interface
  - Propeller (P8X32A) based (can easily output VGA)
  - 640x480, 80x30 chars (8x16 chars), 16 colors
  - Original IBM PC CP437 codepage (ASCII art!)
  - Needs to be fast
    - Full screen editor needs to be fluid
    - Serial terminal emulators are too slow
- Solution: emulate video RAM (VRAM)
  - Propeller uses double buffering for video RAM
  - A buffer is displayed, the other one is read from STM32
  - STM32 outputs video data via DMA
- Easy Propeller firmware upgrade from the eLua shell



# The hardware

- Sound generator
  - Very basic (PWM output)
  - On-board I2S DAC not used
- Extension connector
  - Easy/fast expansion
  - Access to the main peripherals (SPI, I2C, timers, UART, ADC, GPIO)
- PS/2 decoding with GPIO pins
- Propeller converts SPI to UART for radio interface

# The software

- Core: eLua
  - Already has a well supported STM32 port
  - Fast and light
- It lacks some features needed by the Brain
  - No integrated editor or help system
  - Very basic TCP/IP stack, no UDP
  - No support for the ENC28J60 Ethernet controller
  - Output only over UART and TCP/IP (telnet)
  - No PS/2 support
- Fork the eLua tree
  - Implement new features as generic as possible
  - Improve existent features and push them back to eLua

# The software

- The editor
  - Simple, but functional
  - Supports scrolling, long lines, go to line, block copy/paste
  - Memory friendly (custom allocator)
  - Integrated with the help system
- The help system
  - Global, per module or per function help
  - Generated from the same source **as the official docs!**
  - Accesible from anywhere (shell, editor, Lua interpreter)
  - Memory friendly (multi level indexes)
- Simple PS/2 protocol decoder, platform independent

# The software

- TCP/IP stack: major rewrite
  - Basic and buggy in eLua
  - Optional per socket buffers
  - Callbacks for 'new data' events
  - Memory friendly 'expect'-like interface
  - UDP support
  - Restartable (cable inserted/removed callbacks)
  - Platform independent ENC28J60 driver
- VGA console and terminal
  - Uses the internal VRAM (fast, but not portable)
  - Color support with ANSI escape sequence interpreter
  - Vastly improved **term** module

# The software

- Propeller firmware
  - Video generation objects adapted from Parallax forums
  - 'vram' object implemented from scratch
    - Handles double buffering
    - Loop unrolled SPI routines for speed
- Other changes
  - Remote file system (RFS) over UDP
  - Filemasks in shell (/mmc/\*.lua)
  - New 'rm' command
  - More Lua patches:
    - Bit operations using the C syntax
    - Read-only strings (memory optimization)

# The future

- Own PCB instead of daughterboard
- Most likely move to a Cortex-M4 core
- WiFi connectivity
- USB keyboard instead of PS/2
- Graphic mode VGA output (not just text)
- Video output to TV
- Loadable binary modules
- Better sound output
- Interactive debugger
- Improved editor
- Community support (docs, tutorials, reference code)

# More information

- eLuaBrain blog:

<http://eluabrain.blogspot.com>

- EluaBrain public repository:

```
git clone git://github.com/bogdanm/eLuaBrain.git
```

- eLua homepage:

<http://www.eluaproject.net>

- eLua mailing list:

<https://lists.berlios.de/mailman/listinfo/elua-dev>

- eLua wiki:

<http://wiki.eluaproject.net>

**The end**

Questions?