

Forthasia

**A Hardware Composition Based on Several
Forth Kernels**

By Franklin Amador

Forthasia's Purpose

- NASA's International Space App Challenge

<http://spaceappschallenge.org/>

- Low Cost Space Navigation System

- Linux OS
- Open Source Software
 - Human Machine Interface
 - Supervisor Control and Data Acquisition



Forthasia's Structure

1. Open Source Supervisory Control and Data Acquisition (SCADA) package.
 - ❖ Beremiz Python/C SCADA (<http://Beremiz.org>)
 - ❖ MBLLogic Python SCADA (<http://mblogic.sourceforge.net>)
2. Low Cost Linux computer
 - Plug Computer (<http://plugcomputer.org>)
 - ✓ Raspberry Pi (<http://www.raspberrypi.org>)
3. Integrating Input/Output devices
 - Several forth kernels
 - ✓ Gforth (<http://www.gnu.org/software/gforth>)
 - ✓ AMForth (<http://amforth.sourceforge.net>)
 - ✓ 4e4th (<http://4e4th.eu>)
 - ✓ eForth (<http://www.offete.com>)

Open Standards



...

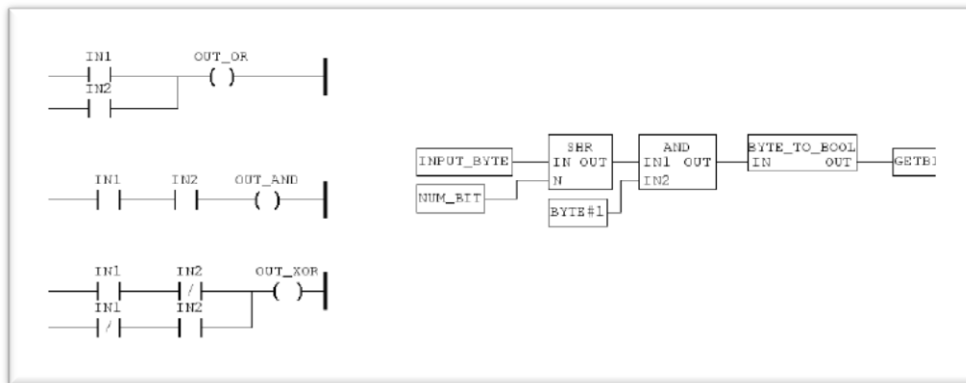
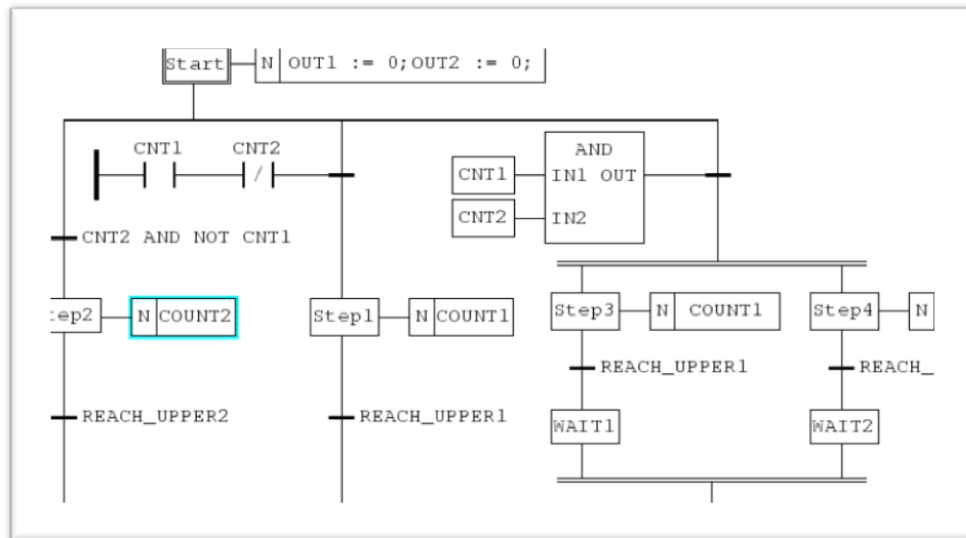
Modbus

CANopen

ETHERNET
POWERLINK

...

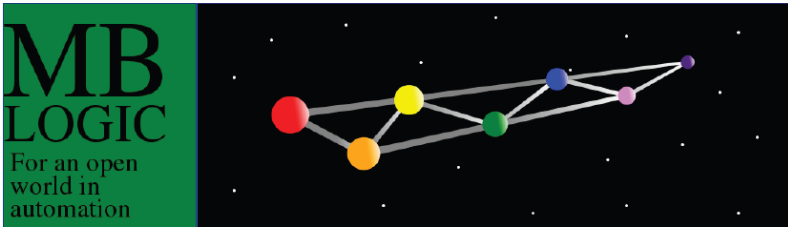
- Standard Software
 - Python
 - C
- Communication Protocol
 - Modbus
 - CANOpen
- IEC-61131-3
 - Sequential Function Charts
 - Ladder Logic
 - Function Blocks
 - Structure Text
 - Instruction List



```

200000000001
2  IF Collision THEN
3   Speed := 0;
4   Brakes := OK;
5  END_IF;
6
7  IF (Gate = CLOSED) AND
8   (Pump = OK) AND (Temp > 200.0) THEN
9   Control_State := Active;
10 ELSE
11  Control_State := Hold;
12  PumpSpeed := 10.0;
13 END_IF;
14
15 LD Y1
16 SUB Y2 (* Subtract Y2 from Y1 *)
17 ST Temp (* Store Y1-Y2 in Temp *)
18 MUL Temp (* Multiply by Temp to square *)
19 ADD X1 (* Defer ADD *)
20 SUB X2 (* Subtract X1 from X2 *)
21 ST Temp (* store X1-x2 in Temp *)
22 MUL Temp (* Multiply by Temp to square *)
23 )
24 SQR (* Call Square root fun *)
25 ST ILTest (* Setup function result *)
26 GT TMax (* Greater than TMax ? *)
27 JMP ERR (* Yes, Jump to Error *)
28 S ERROR (* Set ERROR *)
29 RET (* Normal return *)
30 ERR: RET (* Error return, END not set *)

```



Standard Software

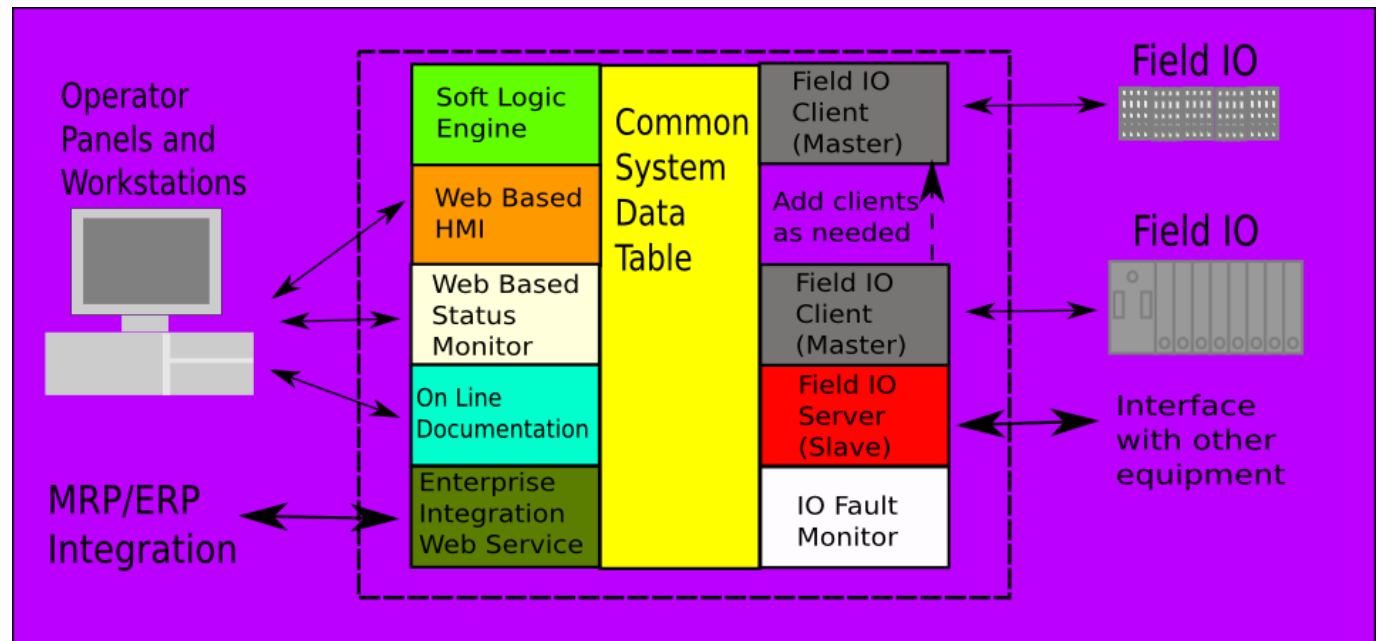
- Python

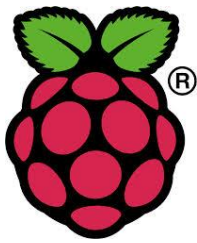
Communication Protocol

- Modbus

IEC-61131-3

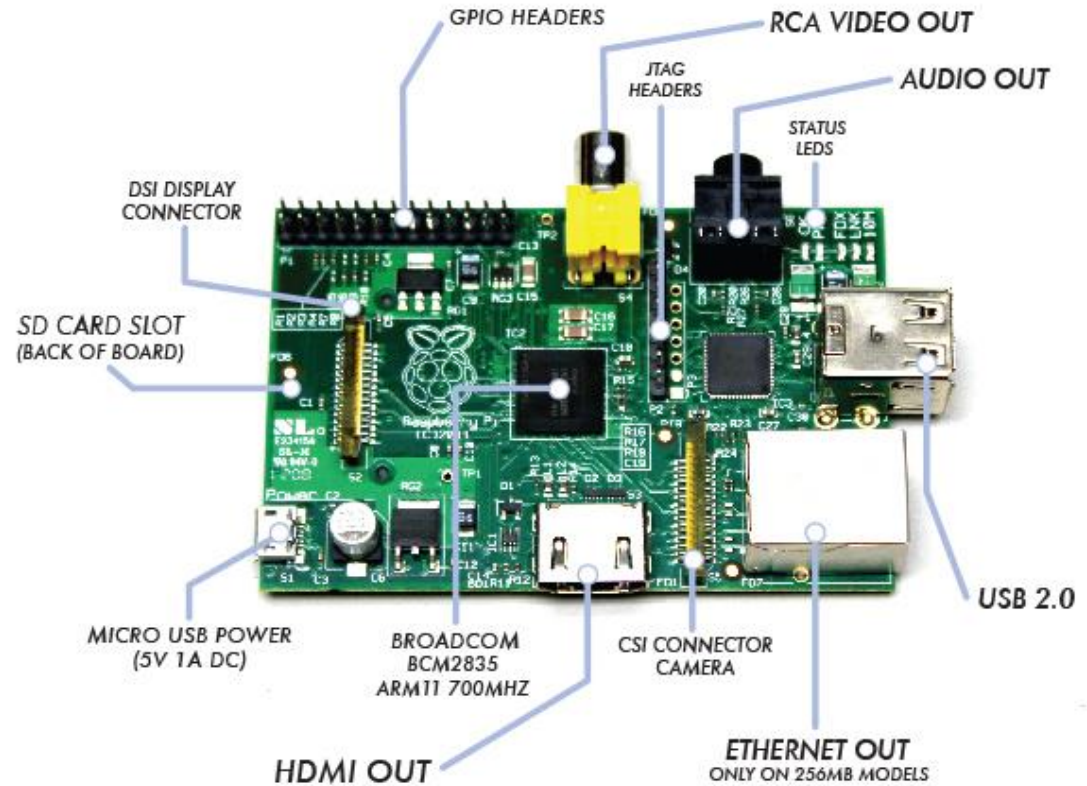
- Ladder Logic





Raspberry Pi

	Model A	Model B
Target Price	20,00 €	28,00 €
SoC	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM)	
CPU	700 Mhz ARM1176JZF-S core	
GPU	VideoCore IV, OpenGL ES 2.0, 1080p30 Full HD HP H.264	
Memory	128 MiB SDRAM	256 MiB SDRAM
USB 2.0 ports	1	2 (via integrated USB hub)
Video outputs	Composite RCA, HDMI	
Audio outputs	3.5 mm jack, HDMI	
Onboard storage	SD / MMC / SDIO card slot	
Low-level peripherals	GPIO pins, SPI, I ² C, UART	
Onboard network	none	10/100 wired Ethernet (RJ45)
Real-time clock	No clock or battery	
Power ratings	500 mA (2.5 Watt)	700 mA (3.5 Watt)
Power source	5 Volt via MicroUSB or GPIO header	
Size	85.60mm x 53.98mm	
Supported OS'es	Debian GNU/Linux, Fedora, Arch Linux	



gForth v7.0

- Downloaded
- Compiled
- 1st Time Working!

AMForth 5.0 IO Device

Specifications:

Microcontroller	ATmega328P
Clock Speed	16 MHz
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Operating Voltage	5V
Input Voltage	612 V
Digital I/O Pin Count	14 (including 6 for PWM output)
Analog Input Pin Count	8
Other Connections	Mini-USB Serial communication (requires header) ICSP (requires header)
Dimensions	1.73 x 0.71 x 0.31 inches (44.0 x 18.0 x 8.0 mm)
Power Source	Mini-USB



AMForth 5.0

- Downloaded
- Compiled
- Flashed
- Finally Working!
- \$35

4e4th IO Device

Specifications:

MCU	Arduino/Atmega	4e4th/MSP430G2553
features	328P	
Architecture	8bit – RISC – Harvard	16bit – RISC – Von Neumann
Power supply (typ.)	5V	3.3V
Flash	32KB	16KB
PINs	28	20
Timers	2	2
PWM pins	6	3
RAM	2KB	512B
clock	8 MHz int. or 16 MHz ext.	16 MHz int. + optional low speed ext.

Flash	8	KB
Serial	TimerSerial	

- ### 4e4th v.34

 - Downloaded
 - Compiled
 - Flashed
 - 1st Time Working!
 - Was \$5, Now \$10

LaunchPad with MSP430G2452
Revision 1.5



+3.3V				1
RED_LED	A0	P1_0		2
	TXD	A1	P1_1	3
	RXD	A2	P1_2	4
PUSH2	A3	P1_3		5
	A4	P1_4		6
	SCK (B0)	A5	P1_5	7
	CS (B0)		P2_0	8
			P2_1	9
			P2_2	10

Hardware	
Pin number	
PC	
TimerSerial	
SPI	
analogRead()	
digitalRead() and digitalWrite()	
digitalRead(), digitalWrite() and analogWrite()	

20					GROUND
19	P2_6				XIN
18	P2_7				XOUT
17					TEST
16					RESET
15	P1_7	A7	SCA	MISO (B0)	
14	P1_8	A6	SCL	MOSI (B0)	GREEN_LED
13	P2_5				
12	P2_4				
11	P2_3				

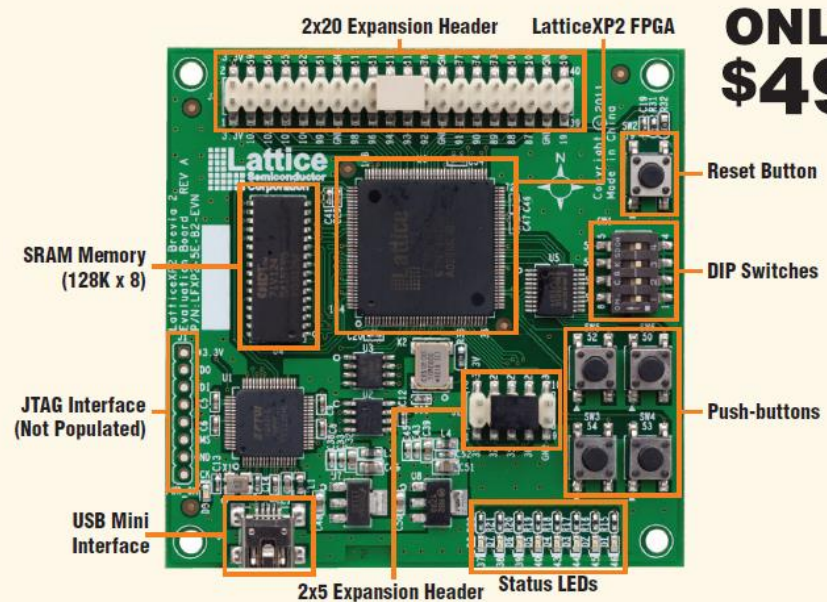
eP32 Forth (CPU) IO Device

LatticeXP2 LFXP2-5E-6TN144C Device
2 Mb SPI Flash Memory
1 Mb SRAM Memory
On-board USB Controller for
JTAG Programming (FTDI-FT232H)
2x20 and 2x5 Expansion Headers
Push-buttons for General Purpose I/O and Reset
4-bit DIP Switch for User Defined Inputs
8 Status LEDs for User Defined Outputs
USB Mini Cable

eForth v.34

- Downloaded
- Compiled
- Flashed
- 1st Time Working!
- \$49

LatticeXP2 Brevia2 Evaluation Board – Top View



**ONLY
\$49**

Miscellaneous IO Devices

ABElectronics UK

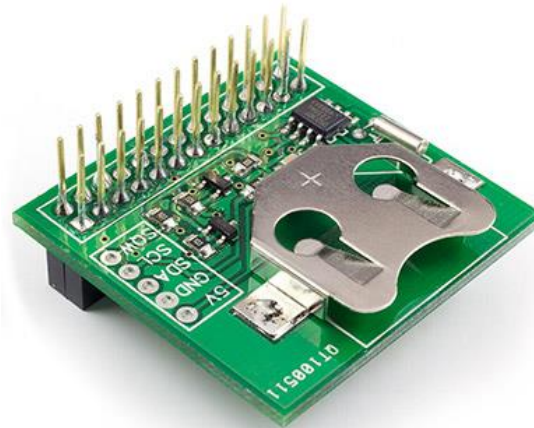
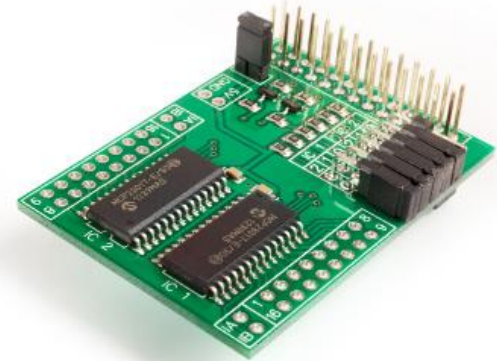
<http://www.abelectronics.co.uk>

RTC

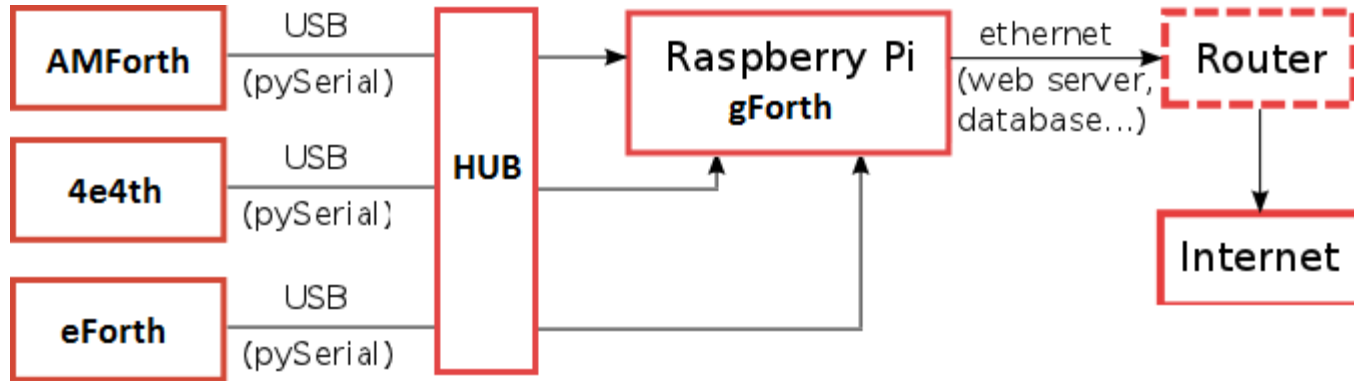
32 I/O

8 ADC

3.75 (18 bit), 15 (16 bit),
60 (14 bit) or 240 (12 bit)
samples per second



Overall Picture



- Common protocol between Forth devices
Modbus or CANOpen

Command Line Tools

Debian Linux

- Sudo apt-get update
- Sudo apt-get install gForth

Serial Session

- Launch Pad (4e4th)
 - `screen /dev/ttyACM0 9600`
- Osepp Nano (AMForth)
 - `screen /dev/ttyUSB2 9600`
- eP32 (eForth)
 - `screen /dev/ttyUSB1 115200 -tn -U`

Screen Command Lines

- Screen
- Ctrl-a + c New Session
- Ctrl-a + A Session Title
- Ctrl-a + k Kill Session
- Ctrl-a + n Next Window
- Ctrl-a + Shift-s Split Horizontal
- Ctrl-a + Shift-| Split Vertical
- Ctrl-a + Tab Change Split Screen
- Ctrl-a + Shift-x Kill Split
- Ctrl-a + i Serial Port Status
- Ctrl-a + d Detach Screen
- Ctrl-a + “ Review Screens

References

- Java on Raspberry Pi (<http://www.oracle.com/technetwork/articles/java/raspberrypi-1704896.html>)
- ABElectronics UK Code Examples (<http://www.abelectronics.co.uk/codesamples/info.aspx>)
- Adafruit learning system (<http://learn.adafruit.com>)
 - Raspberry Pi Category (<http://learn.adafruit.com/category/raspberry-pi>)
 - Raspberry Pi WEB IDE (<http://learn.adafruit.com/webide/overview>)
 - Debugging with the Raspberry Pi WebIDE (<http://learn.adafruit.com/debugging-with-the-raspberry-pi-webide>)
 - Raspberry Pi Educational Linux Distro (<http://learn.adafruit.com/adafruit-raspberry-pi-educational-linux-distro>)

Thank You

Q&A

Brief Example